**ООО НПО «Рапира»**
Россия, 125438, Москва, ул. Автомоторная, дом 7, офис 221
Тел./факс (495) 980-88-74
support@nporapira.ru http://www.nporapira.ru

# Радиомаршрутизатор WIFIBIRD

## Руководство пользователя

# Оглавление

Быстрая настройка ... 8
Настройка в режиме прозрачного моста с использованием веб-интерфейса ... 8
Настройка маршрутизатора с использованием командной строки ... 12
Настройка в режиме прозрачного моста ... 13
Настройка в режиме маршрутизации ... 16
Базовая станция ... 16
Клиентская станция ... 18
Конфигурационная утилита WinBox ... 18
Запуск Winbox ... 18
Часто задаваемые вопросы по WinBox ... 21
Настройка FTP-сервера ... 22
Спецификации ... 22
Параметры файла ... 22
Описание команд ... 22
Терминальная консоль ... 24
Стандартные консольные функции ... 24
Номера и названия элементов ... 25
Быстрый набор команд ... 26
Описание основных команд ... 27
Встроенная справка ... 30
Безопасный режим ... 30
Режим «HotLock» ... 32
Руководство по Firewall ... 33
Прохождение пакета ... 33
Правила файервола ... 34
Фильтрация трафика ICMP ... 34
Тип обслуживания (QOS) ... 35
Peer-to-Peer ... 36
Цепочки правил ... 36
Применение межсетевого экрана ... 37
Защита клиентской сети ... 39
Пример настройки маскарадинга ... 41
Пример настройки destination NAT ... 41
Настройка фильтров в Firewall ... 42

Быстрая настройка фильтров ... 42
Общее описание ... 42
Основные принципы фильтрации ... 43
Фильтрующие цепочки ... 43
Описание параметров ... 44
Примеры применения ... 51
Настройка Mangle ... 54
Описание параметров ... 54
Просмотр статистики ... 63
Примеры использования ... 64
Маркировка по MAC-адресам ... 65
Изменение MSS ... 66
Настройка NAT ... 67
NAT ... 67
Недостатки NAT ... 67
Перенаправление и маскарадинг ... 67
Описание параметров ... 68
Просмотр статистики ... 75
Примеры использования ... 76
Пример использования SRC-NAT (маскарадинг) ... 76
Пример использования DST-NAT ... 77
Пример переадресации данных на другой порт (Port mapping) ... 77
Пример проброса портов на FTP-сервер в локальной сети ... 77
Пример отображения одной сети в другую (один к одному) ... 78
Списки адресов ... 79
Описание параметров ... 79
Протокол L7 (layer7) ... 80
Описание параметров ... 80
Примеры использования ... 80
Обработка пакетов в зависимости от скорости соединения ... 82
Отслеживание установленных соединений ... 84
Механизм определения состояний (conntrack) ... 85
Описание параметров ... 85
Службы, протоколы и порты ... 86
Описание параметров ... 86

Список служб ... 86
SOCKS (прокси-сервер) ... 89
Описание параметров ... 89
Список доступа ... 90
Описание параметров ... 90
Просмотр активных соединений ... 90
Описание параметров ... 90
Пример создания FTP-соединения через SOCKS-сервер ... 91
Описание параметров ... 93
Типы метрик ... 95
Просмотр статуса ... 95
OSPF-зоны ... 96
Описание параметров ... 96
Просмотр статуса ... 97
Межзональное суммирование ... 97
Настройка параметров сети ... 98
Описание параметров ... 98
Настройка дополнительных параметров ... 98
Описание параметров ... 98
Просмотр статуса ... 100
Смежность в сетях NBMA ... 100
Описание параметров ... 100
Виртуальные соединения ... 101
Описание параметров ... 101
Объявление о состоянии канала (LSA) ... 102
Описание параметров (доступны только для чтения) ... 102
Перечень соседних маршрутизаторов ... 102
Описание параметров (доступны только для чтения) ... 103
Список пограничных маршрутизаторов ... 103
Описание параметров (доступны только для чтения) ... 104
Просмотр параметров маршрута ... 104
Описание параметров (доступны только для чтения) ... 104
Прозрачный мост ... 104
Описание параметров ... 105
Настройка основных параметров прозрачного моста ... 107

Описание параметров ..... 107
Настройка параметров портов прозрачного моста ..... 107
Описание параметров ..... 107
Мониторинг прозрачного моста ..... 108
Описание параметров ..... 108
Мониторинг портов прозрачного моста ..... 109
Описание параметров ..... 109
Мониторинг устройств, входящих в состав прозрачного моста ..... 110
Описание параметров (доступны только для чтения) ..... 110
Прозрачный мост и firewall ..... 111
Описание параметров ..... 112
Фильтр пакетов прозрачного моста ..... 116
Описание параметров ..... 116
Прозрачный мост и NAT ..... 117
Основные параметры беспроводного интерфейса ..... 118
Описание параметров ..... 118
Подержка RTS/CTS ..... 130
Настройка списка доступа (ACL) ..... 132
Описание параметров ..... 132
Юстировка ..... 134
Описание параметров ..... 134
Описание команд ..... 134
Список подключений ..... 135
Описание параметров ..... 136
Примеры использования списка подключений ..... 136
Ручная настройка мощности сигнала на беспроводном интерфейсе ..... 137
Описание параметров ..... 138
Таблица регистрации клиентских станций ..... 138
Описание параметров (доступны только для чтения) ..... 138
Утилита «Sniffer» ..... 141
Описание параметров ..... 141
Утилита «Snooper» ..... 142
Основные параметры интерфейса Ethernet ..... 143
Описание параметров ..... 143
Описание параметров (доступны только для чтения) ..... 145

Описание команд ..... 147
Мониторинг проводного интерфейса ..... 147
Описание параметров ..... 147
Диагностика Ethernet-кабеля ..... 149
Просмотр суммарной статистики проводного интерфейса ..... 150
Описание скриптового языка ..... 151
Синтаксис консольных команд ..... 151
Формат строк скрипта ..... 152
Комментарии ..... 152
Объединение строк ..... 152
Пробелы между частями команды ..... 153
Области видимости переменных ..... 154
Глобальная область видимости ..... 154
Локальная область видимости ..... 154
Ключевые слова ..... 154
Разделители ..... 155
Типы данных ..... 155
Escape-последовательности ..... 155
Операторы ..... 156
Арифметические операторы ..... 156
Операторы сравнения ..... 156
Логические операторы ..... 157
Битовые операции ..... 157
Исключающее ИЛИ (XOR) ..... 157
Конкатенация ..... 157
Переменные ..... 158
Команды ..... 159
Глобальные команды ..... 159
Основные команды меню ..... 162
Циклы ..... 163
Функции ..... 164
Обработка ошибок времени выполнения ..... 165
Работа с массивами ..... 166
Доступ к ключам и значениям элементов массива ..... 166
Изменение значения элемента массива ..... 166

Хранилище скриптов ..... 166
Описание параметров ..... 167
Описание параметров (доступны только для чтения) ..... 167
Окружение ..... 168
Описание параметров (доступны только для чтения) ..... 168
Задачи ..... 168
Описание параметров (доступны только для чтения) ..... 168
Планировщик ..... 169
Описание параметров ..... 169
Мониторинг сети ..... 171
Описание параметров ..... 172
Просмотр статуса хоста ..... 172
Мониторинг трафика ..... 174
Описание параметров ..... 174

# Быстрая настройка

### Настройка в режиме прозрачного моста с использованием веб-интерфейса

Данное описание может быть полезно пользователям, которым требуется быстрая настройка ***основных параметров*** оборудования для организации беспроводного канала связи.

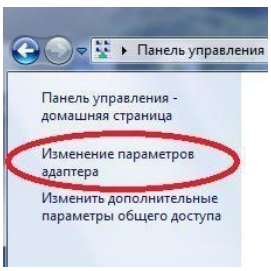
Предположим, требуется организация беспроводного канала в сети с диапазоном ip-адресов: **192.168.0.0/24**

Для этого настроим оба устройства в режиме прозрачного моста и присвоим им следующие адреса: **192.168.0.1** и **199.168.0.2**

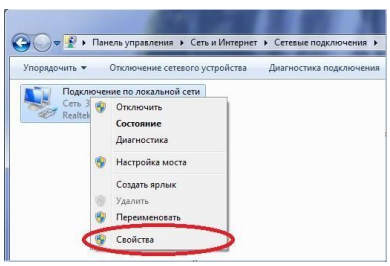
---

*Перед настройкой убедитесь, что компьютер, с которого происходит настройка, находится в сети **192.168.0.0/16**.*
*Процесс настройки сетевых параметров компьютера на примере ОС «Windows 7» описан ниже.*

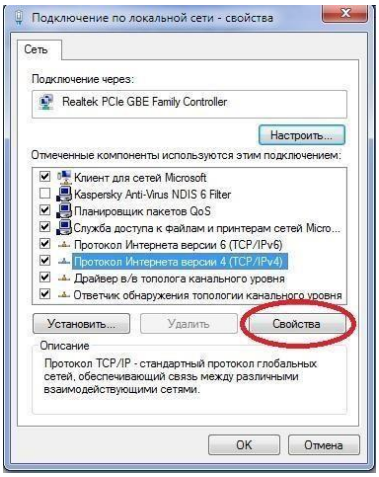
---



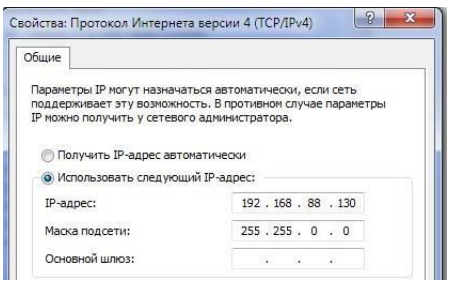

Откройте Панель управления → Сеть и Интернет → Центр управления сетями и общим доступом, в левом меню выберите пункт «Изменение параметров адаптера».




Нажмите на сетевом подключении правой кнопкой мыши и выберите пункт «Свойства».




В данном окне выберите пункт «Протокол интернета версии 4 (TCP/IPv4) и нажмите на кнопку «Свойства».




Укажите IP-адрес и маску подсети, после чего закройте окна настройки сетевых параметров, нажимая на кнопки «ОК».

После выполнения всех необходимых подключений**,** для настраиваемого устройства выполните следующее:

В командной строке браузера укажите стандартный адрес **192.168.88.1** и нажмите ENTER.

В появившемся окне приглашение в качестве логина введите **admin,** поле пароля оставьте пустым и нажмите кнопку **Login.**


В окне Quick Set установите следующие параметры:

1. Режим работы: **PTP Bridge**
2. Wireless Bridge Mode: **Client/CPE** (при настройке первого устройства) и **Server/AP** (при настройке второго устройства)
3. IP Address: **192.168.0.1** (при настройке первого устройства) и

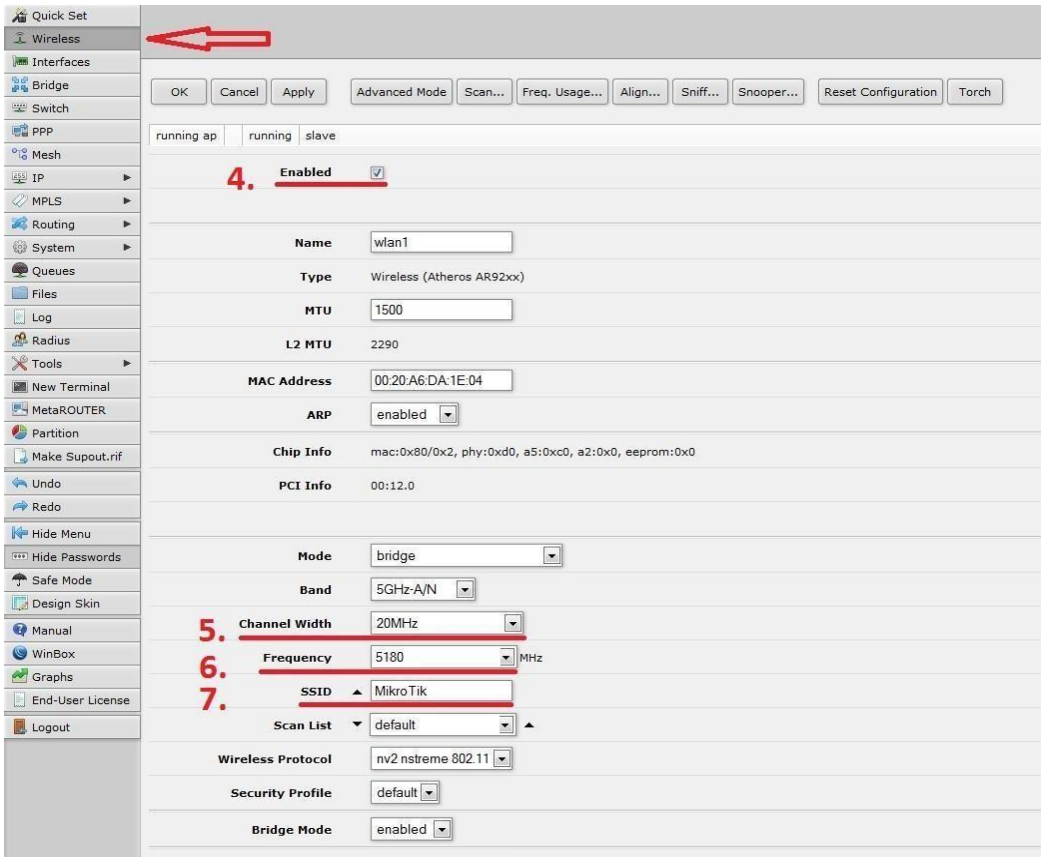
   **192.168.0.2** (при настройке второго устройства)


После чего нажмите кнопку **Apply Configuration** и заново войдите в систему, указав в адресной строке браузера новые адреса:

**192.168.0.1** (для первого устройства) и
**192.168.0.2** (для второго устройства)

4. Теперь необходимо включить беспроводные интерфейсы. Для этого необходимо перейти в окно **Wireless,** выбрать в таблице беспроводной интерфейс (см. рисунок ниже) и у обеих устройств отметить опцию **Enabled.**

5. При необходимости в этом же окне можно настроить и дополнительные параметры: ширину радиоканала, используемую для вещания частоту и SSID (пункты 5-7)



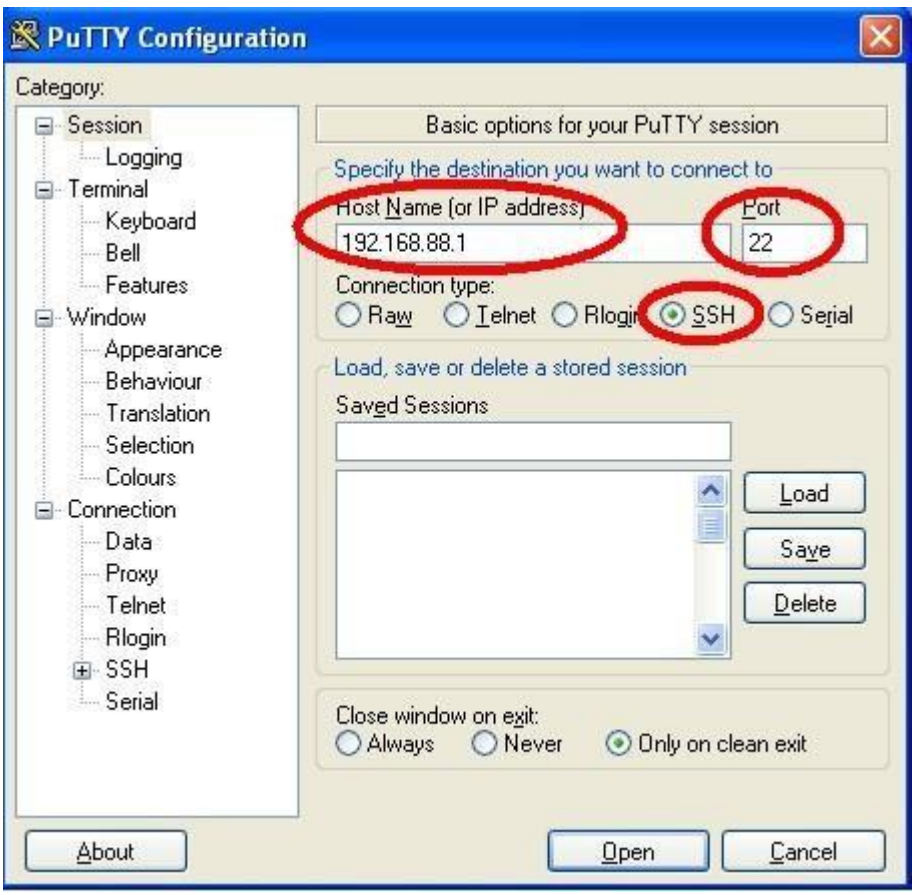

*Расположение элементов в веб-интерфейсе может отличаться от указанных в зависимости от версии ПО. Данная информация представлена для ПО RouterOS версии 6.10*

**Настройка маршрутизатора с использованием командной строки**

Ниже представлены примеры настроек на основе командной строки. Для этой цели мы предлагаем воспользоваться утилитой

PuTTY, скопировав её с прилагаемого компакт-диска или загрузив с адреса: http://putty.org.ru/

Запустите программу, на главной странице укажите IP-адрес **192.168.88.1** и нажмите на кнопку «Open», в открывшемся окне введите логин **admin**. Ввод пароля НЕ ТРЕБУЕТСЯ.




Для настройки основных параметров можно также использовать веб-интерфейс, введя в командной строке браузера адрес

192.168.88.1 и нажав клавишу Enter.

**Внимание!** В целях безопасности не оставляйте поле пароля пустым, укажите новое значение пароля при помощи команды ***/password***

В качестве примера ниже приводится быстрая настройка устройств в двух вариантах:

- ✓ в режиме прозрачного моста
- ✓ в режиме маршрутизации

**Настройка в режиме прозрачного моста**

**Настройка базовой станции в режиме бриджа**

1. Настройте параметры беспроводного интерфейса:

Установка необходимого IP-адреса на проводном интерфейсе, например 192.168.1.250:

```
/ip address set address=192.168.1.250  interface=ether1 0
```

После выполнения данной команды запустите еще одну копию программы PuTTY и подключитесь к устройству, используя только что указанный ip-адрес.

Установка SSID:

```
/interface wireless set wlan1 ssid=test
```

Выбор рабочей частоты радиоканала, например, 5805 МГЦ:

```
/interface wireless set wlan1 frequency=5805
```

Установка режима работы устройства:

```
/interface wireless set wlan1 mode=bridge
```

Установка беспроводного протокола:

```
/interface wireless set wlan1 wirelessprotocol=nv2
```

***2.*** Настройка интерфейса бриджа

**Внимание!** Как правило, на базовой станции по умолчанию создан бридж и оба интерфейса (проводной и беспроводной) уже помещены в группу бриджа, если этот так, то при выполнении команд ***/interface bridge print*** и ***/interface bridge port print*** вы увидите нечто подобное:

```
[admin@MikroTik] > /interface bridge print
Flags: X - disabled, R - running
 0  R name="bridge1" mtu=1500 l2mtu=65535 arp=enabled
      mac-address=00:00:00:00:00:00 protocol-mode=none priority=0x8000
      auto-mac=yes admin-mac=00:00:00:00:00:00 max-message-age=20s
      forward-delay=15s transmit-hold-count=6 ageing-time=5m
```

```
[admin@MikroTik] > /interface bridge port print
Flags: X - disabled, I - inactive, D - dynamic
 #    INTERFACE            BRIDGE             PRIORITY  PATH-COST    HORIZON
 0    ether1-local         bridge1                0x80         10       none
 1    wlan1-gateway        bridge1                0x80         10       none
```

Если указанные выше условия выполнены, то выполнение нижеследующих команд пункта «2» **НЕ ТРЕБУЕТСЯ**.

Создаем интерфейс bridge1:

```
/interface bridge add
```

Помещаем проводной и беспроводной интерфейсы в группу бриджа:

```
/interface bridge port add bridge=bridge1 interface=wlan1
/interface bridge port add bridge=bridge1 interface=ether1
```

### Настройка клиентской станции в режиме бриджа

Настройка клиентской точки производится аналогично базовой, с той лишь разницей, что в качестве режима работы устройства необходимо указать **station-bridge** (см. ниже)

1. **Настройка параметров беспроводного интерфейса**:

Установка необходимого IP-адреса на проводном интерфейсе, например **192.168.1.251:**

```
/ip address set address=192.168.1.251  interface=ether1-local 0
```

После выполнения данной команды запустите еще одну копию программы PuTTY и подключитесь к устройству, используя только что указанный ip-адрес.

Установка SSID:

```
/interface wireless set wlan1 ssid=test
```

Выбор рабочей частоты радиоканала, например, 5805 МГЦ:

```
/interface wireless set wlan1 frequency=5805
```

Установка режима работы устройства:

```
/interface wireless set wlan1 mode=stationbridge
```

Установка беспроводного протокола:

```
/interface wireless set wlan1 wirelessprotocol=nv2
```

### 2. Настройка интерфейса бриджа

Создаем интерфейс bridge1:

```
/interface bridge add
```

Помещаем проводной и беспроводной интерфейсы в группу бриджа:

```
/interface bridge port add bridge=bridge1 interface=wlan1-gateway
/interface bridge port add bridge=bridge1 interface= ether1-local
```

### Настройка в режиме маршрутизации

На следующем рисунке представлена топология сети, на примере которой будет приведена настройка сетевых параметров.


| Маршрутизатор 1 | Маршрутизатор 2 |
|---|---|

**Описание топологии сети:**

Маршрутизатор №1 (база) подключен к сети с диапазоном ip-адресов: **192.168.0.0/24**, а маршрутизатор №2 (клиент) в диапазоне: **192.168.1.0/24**. Для беспроводной сети выбрана сеть **10.0.0.0/24**.

### Базовая станция

Добавление необходимого ip-адреса на проводном интерфейсе базового маршрутизатора:

```
/ip address add address=192.168.0.251 netmask=255.255.255.0 interface=ether1
```

После выполнения данной команды запустите еще одну копию программы PuTTY и подключитесь к устройству, используя ip-адрес, указанный на проводном интерфейсе.

**Внимание!** Как правило, на базовой станции по умолчанию создан бридж и оба интерфейса (проводной и беспроводной) уже помещены в группу бриджа, если этот так, то при выполнении команд ***/interface bridge print*** и ***/interface bridge port print*** вы увидите нечто подобное:

```
[admin@MikroTik] > /interface bridge print
Flags: X - disabled, R - running
 0  R name="bridge1" mtu=1500 l2mtu=65535 arp=enabled
      mac-address=00:00:00:00:00:00 protocol-mode=none priority=0x8000
      auto-mac=yes admin-mac=00:00:00:00:00:00 max-message-age=20s
      forward-delay=15s transmit-hold-count=6 ageing-time=5m
```

```
[admin@MikroTik] > /interface bridge port print
Flags: X - disabled, I - inactive, D - dynamic
 #     INTERFACE            BRIDGE             PRIORITY  PATH-COST    HORIZON
 0     ether1-local         bridge1               0x80         10       none
 1     wlan1-gateway        bridge1               0x80         10       none
```

Если указанные выше условия выполнены, то необходимо ввести следующие команды:

**Удаление интерфейсов из группы бриджа:**

**/interface bridge port remove 1**

**/interface bridge port remove 0**

Добавление необходимого IP-адреса на беспроводном интерфейсе базового маршрутизатора:

**/ip address add address=10.0.0.1/24 interface=wlan**

Установка ip-адреса шлюза по умолчанию на базовом маршрутизаторе:

**ip route add gateway=10.0.0.2**

Добавление маршрута на базовом маршрутизаторе:

**ip route add dst-address=192.168.1.0/24 gateway=10.0.0.1 distance=2**

### Клиентская станция

Установка необходимого IP-адреса на проводном интерфейсе клиентского маршрутизатора:

```
/ip address set address=192.168.1.251 netmask=255.255.255.0 interface=ether1-local 0
```

После выполнения данной команды запустите еще одну копию программы PuTTY и подключитесь к устройству, используя только что указанный IP-адрес.

Добавление необходимого IP-адреса на беспроводном интерфейсе клиентского маршрутизатора:

```
/ip address add address=10.0.0.2/24 interface=wlan1-gateway
```

Установка ip-адреса шлюза по умолчанию на клиентском маршрутизаторе:

```
/ip route add gateway=10.0.0.1
```

# Конфигурационная утилита WinBox

### Описание

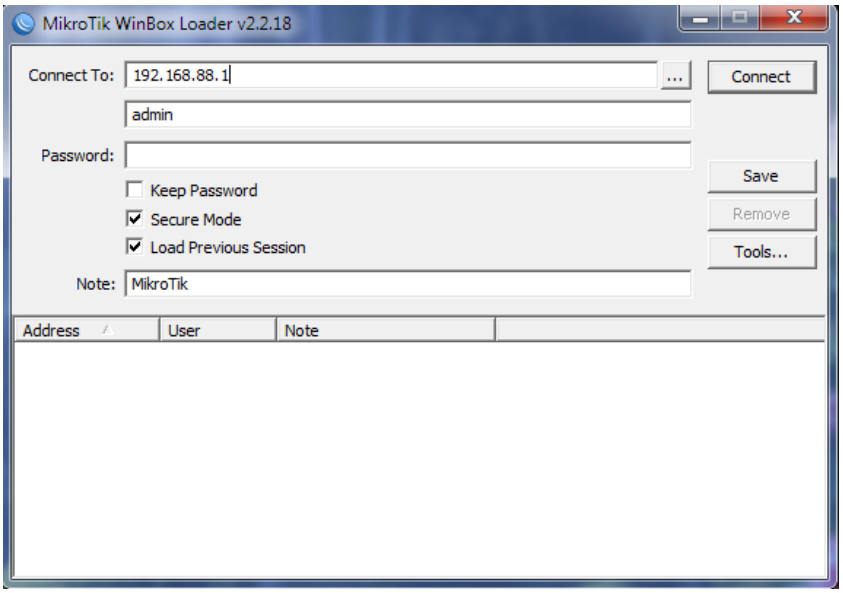
Winbox используется для быстрой и удобной настройки радиомаршрутизатора при помощи графического интерфейса (GUI). Все функции Winbox максимально приближенны к традиционной консоли.
Утилита winbox.exe доступна по адресу http://ip_адрес/winbox/winbox.exe, где ip_адрес соответствует IP-адресу настраиваемого маршрутизатора. Используйте любой браузер для доступу к файлу утилиты.

Все утилиты RouterOS, загруженные через браузер, кэшируются на жесткий диск и при повторном обращении загружаются уже с него.

### Запуск Winbox

При подключении через браузер к маршрутизатору (осуществляется по http, 80 порт по умолчанию), отобразится страница приветствия.

При нажатии на иконку Winbox начнется закачка исполняемого файла winbox.exe. Сохраните winbox.exe на диск и запустите ее. При запуске программы открывается окно входа в систему:




Описание кнопок:

... – Поиск и отображение MNDP (MikroTik Neighbor Discovery Protocol) или CDP (Cisco Discovery Protocol) устройства.

Connect – Подключение к радиомаршрутизатору по выбранному IP (и номеру порта, если вы изменили это значение. По умолчанию используется **80 порт**) или MAC-адресу (если маршрутизатор находится в другой подсети), указанному имени пользователя и паролю.

Save – Сохранение текущего сеанса в списке ниже. Для запуска кликните два раза на выбранном элементе.

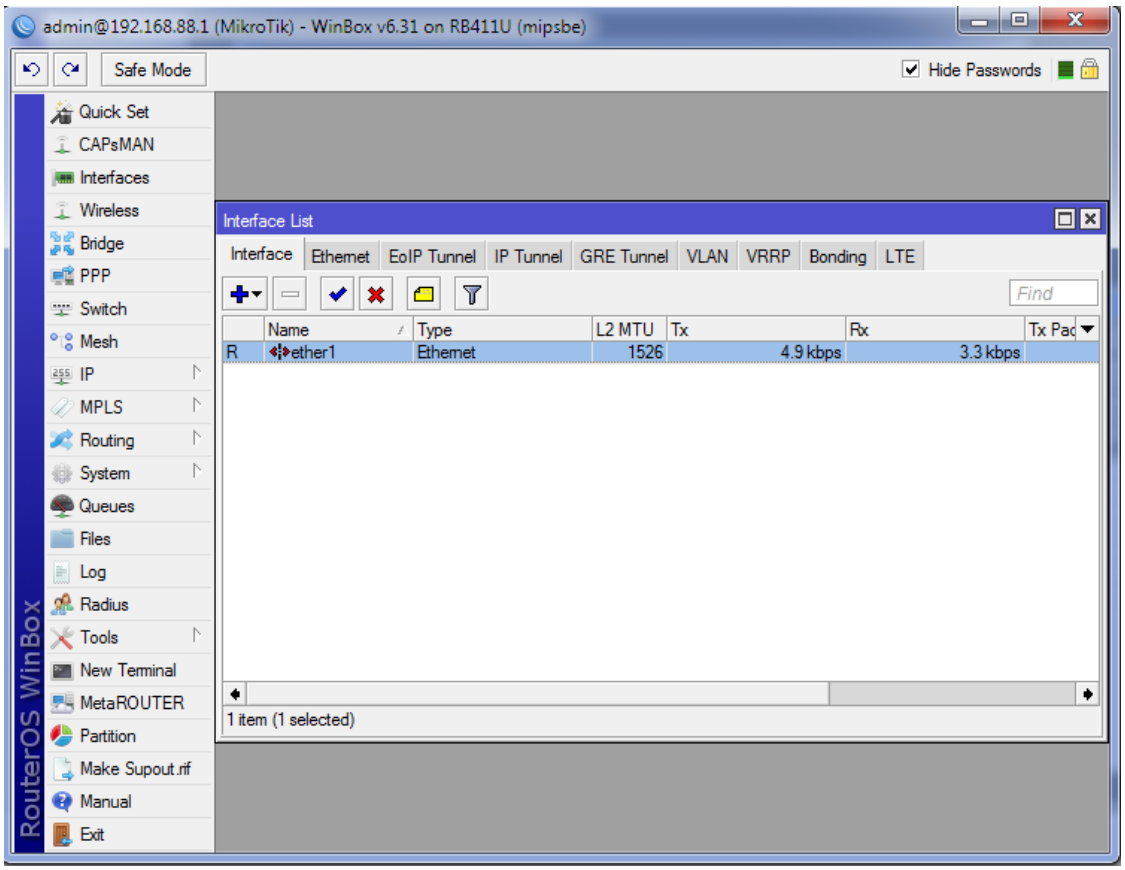
Remove - Удаление выбранного элемента из списка

Tools... - Удаление всех элементов из списка, очистка кэша на локальном диске, импортирование адресов из wbx-файла или экспортирование их в wbx-файл.

- Secure Mode – Обеспечивает секретность и целостность данных в сеансе подключения посредством TLS (Transport Layer Security)
- Keep Password – Сохраняет пароль как обычный текст на локальном жестком диске. ВНИМАНИЕ: хранение паролей в обычном тексте позволит любому пользователю, имеющему доступ к файлу с паролем воспользоваться им.

**Внешний вид Winbox**

Консоль WinBox использует TCP-порт 8291. После подключения к маршрутизатору вы можете работать с его конфигурацией через WinBox и выполнять те же самые задачи, что и в обычной консоли.




**Краткий обзор общих функций**

Используйте меню слева для навигации по функционалу маршрутизатора. Двойным щелчком на каком-либо элементе списка открывается окно настройки выбранного элемента. Ниже показаны некоторые элементы управления консоли WinBox.

Для открытия требуемого окна, кликните один раз на соответствующем пункте меню.

- Добавить новый элемент
- Удалить существующий элемент
- Включить элемент
- Отключить элемент

- Создать или редактировать комментарий
- Обновить окно
- Отменить действие
- Повторить действие
- Отключиться от WinBox

**Часто задаваемые вопросы по WinBox**

**Могу ли я запустить WinBox на Linux?**

Да вы можете запустить WinBox и подключиться к RouterOS, используя Wine.

**Я не могу открыть консоль WinBox?**

Проверьте порт и адрес для **www**-сервиса в списке **/ip service print**.
Убедитесь, что адрес, к которому вы подключаетесь, находится в доступной подсети и корректно указан порт загрузчика WinBox. Команда **/ip service set www port=80 address=0.0.0.0/0** восстановит настройки по умолчанию.

Консоль WinBox использует порт 8291. Убедитесь в том, что доступ не блокируется файерволом.

# Настройка FTP-сервера

### Спецификации

Требуемые пакеты: **system**

Стандарты и технологии: FTP (RFC 959)

Аппаратное обеспечение: не принципиально

Уровень подменю: **/file**

### Описание

RouterOS оснащена стандартным FTP-сервером. Для связи с другими хостами в сети используются 20 и 21 порты.

Через меню **file** можно получить доступ к загруженным файлам, а также к экспортируемой конфигурации или резервным файлам. Здесь же можно удалить неиспользуемые файлы. Для авторизации на FTP используйте пользовательский аккаунт (имя и пароль, заведенные на маршрутизаторе).

### Параметры файла

**creation-time** (read-only: time) – дата и время создания файла

**name** (read-only: name) – имя файла

**size** (read-only: целое значение) – размер файла в байтах

**type** (read-only:file | directory | unknown | script | package | backup) – тип файла

### Описание команд

**print –** показать список сохраненных файлов

Список параметров команды **print**

- **detail –** показать содержимое файлов (если размер не превышает 4Кбайт)
- **edit [item] contents –** редактирование содержимого файлов с помощью редактора
- **set contents=[content] –** присваивание файлу **[item]** содержимого **contents**

# Терминальная консоль

## Стандартные консольные функции

Для настройки маршрутизатора в консоли используются текстовые команды. Хотя структура консольных команд напоминает Unix shell, рекомендуется получить дополнительную информацию о структуре команд в разделе «**Описание скриптового языка**» на стр. 151. Далее будет описаны наиболее популярные команды, организованные в виде иерархического меню (уровней). Название уровня отражает настройки, доступные в соответствующем разделе.

Для примера выполните команду **/ip route print**:

```
/ip route print
Flags: A – active, X – disabled, I – invalid, D – dynamic, C – connect, S – static, r – rip, b – bgp, o – ospf, d – dynamic
#            DST-ADDRESS      G GATEWAY          DISTANCE INTERFACE
ADC 1.1.1.0/24         isp2
A S 2.2.2.0/24          r 1.1.1.2   0              isp2
ADC 3.3.3.0/24         bonding1
ADC 10.1.0.0/24         isp1
A S 0.0.0.0/0           r 10.1.0.1  0              isp1
```

Вместо того, чтобы каждый раз набирать полный путь /**ip route** для выполнения команды **print**, вы можете просто перейти на соответствующий уровень и уже из него выполнять необходимую команду. Пример ниже демонстрирует эту возможность:

```
ip route
ip route> print
Flags: A – active, X – disabled, I – invalid, D – dynamic,C – connect, S – static, r – rip, b – bgp, o – ospf, d – dynamic
#            DST-ADDRESS      G GATEWAY          DISTANCE INTERFACE
ADC 1.1.1.0/24         isp2
A S 2.2.2.0/24          r 1.1.1.2   0              isp2
ADC 3.3.3.0/24         bonding1
ADC 10.1.0.0/24         isp1
A S 0.0.0.0/0           r 10.1.0.1  0              isp1
```

**Обратите внимание**: в результате вы перейдете на уровень ниже в иерархии меню. Для перехода в корень меню выполните «/»

```
/ip route
ip route> /
>
```

Для перехода на один уровень вверх выполните «**..**»

```
ip route> ..
ip>
```

Вы также можете использовать символы «/» и «**..**» для выполнения команд из другого уровня меню без смены текущего уровня, например:

```
ip route> /ping 10.0.0.1
10.0.0.1 ping timeout
2 packets transmitted, 0 packets received, 100% packet loss

ip firewall nat> .. service-port print
Flags: X – disabled, I – invalid
# NAME
PORTS
ftp           21
tftp           69
irc
6667
X h323
quake3
mms
gre
pptp
```

### Номера и названия элементов

Некоторые уровни команд содержат несколько элементов, представленные в виде списков, например: interfaces, routes, users и т.д.

Каждый элемент в подобном списке имеет свой уникальный номер и перечень параметров, соответствующих данному элементу. Для изменения параметра элемента используется команда **set** с последующим указанием имени или номера элемента.

#### Названия элементов

Как правило каждому элементу в списке кроме номера присвоено уникальное имя, например, на уровне команд **interface** или **user**. Работая с такими элементами, вы можете использовать названия вместо порядкового номера.

Названия элементов, в отличие от номеров, не назначаются автоматически, а являются свойствами элементов. Как правило названия элементов более «стабильны» и информативны чем номера, поэтому при написании консольных сценариев предпочтительно использовать именно их.

#### Номера элементов

Номера элементам присваиваются автоматически и присвоенные номера не являются константами: вполне возможно, что вызванная два раза подряд команда **print** отсортирует элементы списка в другом порядке. Однако результат последних команд **print** запоминается, таким образом, номера элементов могут быть использованы с командами **add**, **remove** и **move** (в прошивке версии 3 и выше команда **move** не перенумеровывает элементы списка). Номера элементам присваиваются в каждой новой рабочей сессии, и они останутся такими же, пока вы не закроете консоль не выполните очередную команду **print**. Обратите внимание, что номера назначаются индивидуально для каждого элемента списка, поэтому команда **IP-адрес print** не изменит нумерацию в списке интерфейсов.

При выполнении ряда команд возможно перечислить номера или названия сразу нескольких элементов списка, что позволяет упростить настройку конфигурации, например:

```
interface print
Flags: X – disabled, D – dynamic, R – running
 #    NAME              TYPE          MTU
 0  R ether1            ether         1500
 1  R ether2            ether         1500
 2  R ether3            ether         1500
 3  R ether4            ether         1500

interface set 0,1,2 mtu=1460
interface print
Flags: X – disabled, D – dynamic, R – running
 #    NAME              TYPE          MTU
 0  R ether1            ether         1460
 1  R ether2            ether         1460
 2  R ether3            ether         1460
 3  R ether4            ether         1500
```

### Быстрый набор команд

Возможно использование двух алгоритмов, помогающих и намного ускоряющих ввод команд в консоли: используя клавишу [Tab] и вводя укороченные команды. Автозавершение команд реализовано подобно **bash** shell в UNIX. Если вы нажмете клавишу [Tab] после набора части команды, консоли попытается найти соответствующую команду, начинающуюся с части набранного слова. Если найдено только одно совпадение, будет автоматически добавлена оставшаяся часть команды с последующим пробелом:

**/int[Tab]_ → /interface _**

Если найдено более одного совпадения, будет выведена максимально совпадающая часть слова, без добавления пробела в конец:

**/interface set e[Tab]_ → /interface set ether_**

Если вы набрали максимально совпадающую часть слова и нажали клавишу [Tab] дважды, будут отображены все возможные конечные варианты:

```
interface set e[Tab]_
interface set ether[Tab]_
interface set ether[Tab]_ ether1 ether5
interface set ether_
```

Клавиша [Tab] может быть использована в другом контексте, где консоль может найти ключи к возможным значениям – имена команд, имена аргументов, аргументы, имеющие несколько возможных значений (названия элементов списков или названия протоколов в firewall и NAT правилах). Вы не можете выполнять быстрый набор цифр, IP-адресов и других подобных значений.

Другой путь уменьшения нажатия клавиш — это сокращение команд и аргументов. Вы можете набирать только начало команды, и, если оно (имя команды) не полно, консоль примет ее как полное имя команды. Так набору:

**pi 10.1 c 3 si 100**

будет соответствовать команда

**ping 10.0.0.1 count 3 size 100**

Быстрый набор команды работает даже если слово введено не сначала, но включает в себя точную часть подстроки: если не найдено точного совпадения, то консоль начинает просмотр выражений содержащих введенную строку как первое слово в выражении, или просто как подстроку всей строки в том же самом выражении. Если найдено совпадение, то в слово будет дополнено. Например:

```
interface x[TAB]_
interface export _

interface mt[TAB]_
interface monitor-traffic _
```

### Описание основных команд

Здесь описаны некоторые команды, обычно используемые на всех уровнях меню, а именно, **print, set, remove, add, find, get, export, enable, disable, comment, move.** Эти команды ведут себя одинаково на всех уровнях меню.

#### PRINT

- Отображает всю информацию, доступную в текущем уровне меню. Таким образом, команда **/system clock print** показывает системную дату и время, **ip route print** показывает все маршруты и т.д. Если на текущем уровне меню используется список элементов и у них не установлено свойство «только для чтения» (как, например, список элементов **/system history**, показывающий историю выполненных действий), то можно изменять/удалять данные элементы.

Также команда **print** автоматически назначает номера элементам списка, которые используются командами при работе с элементами списка.

**Основные параметры команды print**

- **from** – отображает указанные элементы списка. Может быть использован в сочетании с параметром **find**, например:

```
interface print from=[find l2mtu>1500]
Flags: D – dynamic, X – disabled, R – running, S – slave
 #    NAME                TYPE       ACTUAL-MTU L2MTU  MAX-L2MTU
 0  R  ether1              ether          1500  1526
 1  X  wlan1               wlan           1500  1600
interface print  from=1
Flags: D – dynamic, X – disabled, R – running, S – slave
 #    NAME                TYPE       ACTUAL-MTU L2MTU  MAX-L2MTU
 0  X  wlan1               wlan           1500  1600
interface print  from=1
no such item
```

**Обратите внимание**: каждый раз при выполнении команды **print** происходит ***перенумерация*** списка отображаемых элементов. В данном примере был найден только один элемент (wlan1) с присвоенным ранее номером (1) и команда **print** присвоила ему номер «0». Элемента с номером «1» в списке уже не существует, и попытка обратиться к элементу «1» сразу после предыдущей команды окончилась неудачей.

- **as-value –** отображает элементы в виде массива параметров и их значений
- **brief** – отображает элементы списка в табличном виде
- **count-only** – отображает количество элементов в списке
- **detail** – отображает элементы списка в виде «параметр=значение»
- **file** – перенаправление вывода в указанный файл. Данный файл будет доступен в меню /file
- **follow –** отображает все элементы списка единым списком, пока не будет нажато сочетание ctrl+c. Удобно для просмотра содержимого файла журнала.
- **follow-list –** отображает только новые элементы списка в реальном времени, пока не будет нажато сочетание ctrl+c. Удобно для просмотра содержимого файла журнала.
- **from –** отображает параметры только указанного элемента
- **interval** – постоянно отображает список элементов с обновлением экрана через указанный интервал времени
- **oid –** отображает идентификаторы объектов (OID), доступных через SNMP
- **terse** – отображает информацию в компактном виде

- **value-list** – отображает каждый параметр на отдельной строке
- **where –** отображает элементы, соответствующие указанным критериям. Синтаксис аналогичен команде **find**
- **without-paging –** печатает вывод команды полностью, без постраничного отображения

**SET**

Применяется для изменения значений главных параметров или параметров элементов. В качестве аргумента передаётся имя изменяемого параметра. Используйте команду «**?**» или двойное нажатие клавиши [Tab] для просмотра списка доступных аргументов. Если команда применяется к списку элементов, то команда **set** имеет один аргумент – номер элемента (или список номеров) которые вы изменяете. Эта команда не выводит на экран результат своего выполнения.

**ADD**

Данная команда как правило имеет все те же аргументы что и команда **set**, кроме номера элемента. Команда добавляет новый элемент с указанными значениями параметров, обычно в конец списка элементов. Часть параметров необходимо указывать в обязательном порядке (например, название интерфейса для вводимого IP-адреса), часть параметров могут автоматически принимать значения по умолчанию, если вы явно не указали необходимые значения.

**Основные параметры команды add**

- **copy-from** – создание нового элемента на основе существующего элемента. Значения по умолчанию наследуются от родительского элемента. При необходимости можно сразу указать необходимые значения для новых параметров. Если родительский элемент помимо номера имеет и название, то для нового элемента необходимо задать новое название.
- **place-before –** размещение создаваемый элемент перед указанным элементом. В этом случае вам не придется использовать команду **move** после добавления элемента в список.
- **disabled –** управление состоянием отключено/включено добавляемых элементов.
- **comment** – добавление комментария к добавляемому элементу

Возвращаемые значения команды **add**

При выполнении команда **add** выводит номер добавленного элемента.

**REMOVE**

Команда удаляет указанные (по номеру или названию) элемент(ы) из списка.

**MOVE**

Команда изменяет порядок следования элементов в списке.

**Параметры команды move**

- Первый аргумент определяет перемещаемый элемент(ы)
- Второй аргумент определяет, перед каким элементом будет размещен перемещаемый(ые) элемент(ы). Если второй аргумент опущен, элемент добавляется в конец списка.

**FIND**

Команда имеет те же аргументы что и команда **set**, а также дополнительные аргументы **disabled** и active, которые могут принимать значения yes или **no.** Для просмотра всех аргументов и их значений, см. «Основные параметры команды print» на стр. 28. Команда **find** возвращает номера всех элементов, найденных по указанному значению аргумента.

**EDIT**

Команда работает так же, как и команда set, но её удобнее использовать для редактирования больших объемов текста, например, скриптов.

**Встроенная справка**

В консольном режиме можно отобразить краткую справочную информацию по вводимым командам, нажав клавишу «?» после веденного элемента (аналогично двойному нажатию клавиши **[Tab],** но в данном случае информация выводится в более детальном варианте)

**Безопасный режим**

Этот режим делает невозможным изменить конфигурацию маршрутизатора кроме как с терминальной консоли. В обычном режиме после случайной потери связи с маршрутизатором не существует возможности отката только что настроенной конфигурации. Безопасный режим минимизирует подобные риски при настройке.

Переход в безопасный режим осуществляется нажатием клавиш **[Ctrl]+[X]**. Для выхода из безопасного режима **с сохранением** всех настроек повторно нажмите **[Ctrl]+[X].**

Для выхода из безопасного режима **без сохранения** введённых настроек нажмите **[Ctrl]+[D]**.

При переходе в безопасный режим в приглашении командной строки появляется суффикс **<SAFE>**

```
ip route>[Ctrl]+[X]
[Safe Mode taken]
ip route <SAFE>
```

Теперь все изменения конфигурации (для **всех** рабочих сессий) автоматически отменяются, если рабочая сессия завершается некорректно. Все отменяемые изменения можно просмотреть в истории команд: они помечаются флагом «**F**»:

```
/ip route<SAFE> add gateway=10.0.0.1
/ip route<SAFE> /system history print
Flags: U – undoable, R – redoable, F – floating-undo
 ACTION                    BY          POLICY
F route added              admin       write
```

Теперь, если соединение с маршрутизатором будет внезапно потеряно, все изменения, выполненные ранее в безопасном режиме, будут отменены (приблизительно через 9 минут). Закрытие сеанса комбинацией клавиш **[Ctrl]+[D]** также отменят все изменения, выполненные в безопасном режиме.

Если другие пользователи попытаются параллельно подключится в безопасном режиме, то они увидят следующее сообщение:

**Hijacking Safe Mode from someone – unroll/release/don't take it [u/r/d]:**

**Где:**
**[u] – отмена** всех изменений, сделанных ранее в безопасном режиме, и перевод текущего сеанса в безопасный режим
**[r] – сохранение** всех изменений, сделанных ранее в безопасном режиме, и перевод текущего сеанса в безопасный режим Предыдущий владелец безопасного режима будет уведомлен об этом:

**ip firewall rule input [Safe mode released by another user]**

**[d] –** оставить все изменения «как есть».

Если во время безопасного режима было сделано слишком много изменений, и они не могут быть сохранены в истории команд (текущая версия истории команд может хранить до **100** записей), то сеанс автоматически выходит из безопасного режима, без автоматической отмены изменений. Таким образом, работая в безопасном режиме, лучше изменять конфигурацию постепенно. При необходимости вы легко отмените действия, произведенные в безопасном режиме, выполнив двойное нажатие клавиш **[Ctrl]+[X]**.

### Режим «HotLock»

В данном режиме включено автодополнение вводимых команд.

Для входа в режим нажмите сочетание [**CTRL]+[V].**

```
/IP-адрес> [CTRL]+[V]
/IP-адрес>>
```

Суффикс «≫» в конце строки приглашения означает, что данный режим включён.

Например, при вводе **`/in e,`** вы получите

```
/IP-адрес>> /interface ethernet
```

# Руководство по Firewall

### Спецификация

Требуемые пакеты: **system**

Уровень лицензии: Level1(Фильтрация P2P ограничено 1), Level3

Уровень подменю: **/ip firewall**

Стандарты и технологии: IP

Аппаратные требования: Увеличиваются по мере увеличения количества правил фильтрации

### Описание

Межсетевые экраны позволяют отделять локальную сеть от внешней сети, делая ее менее уязвимой. Всегда есть вероятность что кто-нибудь извне вторгнется в вашу локальную сеть (ЛВС). Такие взломы могут привести к потере данных и прочим проблемам.
Брандмауэры (иначе – файервол) используются как средство предотвращения или уменьшения рисков вторжения в вашу ЛВС. В RouterOS также реализована такая функция как маскарадинг, который позволяет скрывать инфраструктуру вашей ЛВС.

### Прохождение пакета

### Описание

RouterOS упрощает создание и развертывание сложных политик безопасности.

Фактически вы можете легко создать простой фильтр для трансляции адресов, даже не задумываясь о том, как пакет обрабатывается маршрутизатором.

Но если вы хотите развернуть более сложную политику безопасности, необходимо знать некоторые детали процесса. Прохождение пакета через маршрутизатор показано на схеме ниже.

Пакет может поступить на обработку двумя путями. Первый пакет прибыл с сетевого интерфейса, второй пакет был порожден самим маршрутизатором. Аналогично пакет может покинуть обработку т.е. или уйти через сетевой интерфейс наружу или, если пакет предназначен локальным приложениям, он будет отдан им.

Когда пакет прибывает на сетевой интерфейс, правила файервола отрабатываются в следующем порядке:

Сначала отрабатываются правила NAT. Файервольные правила цепочки input и решение о маршрутизации отрабатываются после прохождения правил NAT.

Если пакет должен быть отправлен через маршрутизатор(forward), то следующим за NAT отрабатываются правила forward.

Когда пакет покидает интерфейс, первыми отрабатываются правила цепочки output, затем NAT и очереди.

Дополнительные стрелки от IPSec показывают обработку зашифрованных пакетов (сначала пакет проходит шифрацию/дешифрацию, а затем обрабатывается как обычный пакет).

### Правила файервола

Уровень подменю: **/ip firewall rule**

#### Описание

Правило – это определенное выражение, которое говорит маршрутизатору что делать с конкретным пакетом. Правило состоит из двух логических частей: установки соответствия и установки действия. Для каждого пакета вам необходимо определить правило соответствия и действие.

Управление правилами файервола может быть выполнено путем выбора желаемой цепочки. Если вы используете консоль WinBox, выберите желаемую цепочку и для открытия окна со списком правил укажите выпадающий список правил (справа вверху).

#### Фильтрация трафика ICMP

Для защиты маршрутизатора и стоящей за ним частной сети, вам необходимо сконфигурировать файервол для отклонения большей части ICMP-трафика. Однако некоторые ICMP-пакеты необходимы для поддержки бесперебойной работы сети и диагностики неисправностей.

Ниже представлен список значений ICMP TYPE:CODE в нормальных пакетах. Эти типы ICMP трафика лучше разрешить.

**Ping**

- 8:0 – echo request (эхо запрос)
- 0:0 – echo reply (эхо ответ)

**Trace**

- 11:0 – TTL exceeded (превышен интервал)
- 3:3 – Port unreachable (порт недоступен)

**Path MTU descovery**

- 3:4 – Fragmentation-DF-Set (набор флагов фрагментации)

**Общие рекомендации по фильтрации ICMP-трафика**

- Разрешить исходящий ping ICMP Echo-Request и входящие сообщение Echo-reply
- Разрешить traceroute TTL-Exceeded и Port-Unrechable входящие сообщения
- Разрешить path MTU-ICMP Fragmentation-DF-Set входящие сообщения
- Все остальное блокировать

### Тип обслуживания (QOS)

Поскольку сеть сама по себе не наделена знаниями об оптимизации пути выбранного приложением или пользователем, протокол IP обеспечивает возможность взаимодействия с протоколами верхнего уровня, которые передают подсказки интернет-уровню как должны происходить обмены пакетами для данного соединения. Это средство называют Type of Service (типом сервиса).

Тип обслуживания – это стандартное поле TCP-пакета, используемое многими сетевыми приложениями и аппаратными средствами для определения наилучшего пути прохождения пакета.

RouterOS в полной мере использует поле ToS (1 байт). Зарезервированные биты в этом байте не обрабатываются. Это означает что устройство работает как с метками DiffServ (Differentiated Services Codepoint, DSCP определено в RFC2474), так и с расширением ECN (Explicit Congestion Notification, определено в RFC3168), использующих аналогичные поля заголовка IP-пакета. Обратите внимание: это не означает, что RouterOS поддерживает DiffServ или ECN, просто есть возможность изменения меток, использованных в данных протоколах.

В RFC1349 определены следующие стандартные значения:

- normal – нормальное обслуживание (ToS=0)
- low-cost – минимальная стоимость (ToS=2)

- max-reliability – максимальная надежность (ToS=4)
- max-troughput – максимальная пропускная способность (ToS=8)
- low-delay – минимальная задержка (ToS=16)

**Peer-to-Peer**

RouterOS обеспечивает возможность фильтрации трафика для многих пиринговых программ, использующих протоколы P2P. В частности, поддерживаются следующие протоколы:

- **Fasttrack** (Kazaa, KazaaLite, Diet Kazaa, Grokster, iMesh, giFT, Poisoned, mlMac)
- **Gnutella** (Shareaza, XoLoX, , Gnucleus, BearShare, LimeWire (java), Morpheus, Phex, Swapper, Gtk-Gnutella (linux), Mutella (linux), Qtella (linux), MLDonkey, Acquisition (Mac OS), Poisoned, Swapper, Shareaza, XoloX, mlMac)
- **Gnutella2** (Shareaza, MLDonkey, Gnucleus, Morpheus, Adagio, mlMac)
- **DirectConnect** (DirectConnect (AKA DC++), MLDonkey, NeoModus Direct Connect, BCDC++, CZDC++ )
- **eDonkey** (eDonkey2000, eMule, xMule (linux), Shareaza, MLDonkey, mlMac, Overnet)
- **Soulseek** (Soulseek, MLDonkey)
- **BitTorrent** (BitTorrent, BitTorrent++, Shareaza, MLDonkey, ABC, Azureus, BitAnarch, SimpleBT, BitTorrent.Net, mlMac)
- **Blubster** (Blubster, Piolet)
- **WPNP** (WinMX)

**Цепочки правил**

**Уровень подменю: /ip firewall**

**Описание**

Правила фильтрации группируются в цепочки. Это позволяет разрешить прохождение пакетов, объединив их общими критериями внутри одного правила и затем передать в другую цепочку, объединив их другими критериями. Предполагается, что, например, пакеты собираются для конкретных IP-адресов и портов. Соответственно, если используются только IP-адреса, то указывать протокол и порт не обязательно. С другой стороны, если указывать протокол и порт, то использовать IP-адрес необязательно.

Существуют три преопределенных цепочки, которые не могут быть удалены:

- Цепочка **input** используется для обработки входящих на маршрутизатор пакетов, и предназначенных именно ему. Пакеты, проходящие транзитом через маршрутизатор, в цепочку **input** не попадают.
- Цепочка **forward** используется для обработки пакетов, идущих транзитом через маршрутизатор.
- Цепочка **output** используется для обработки исходящих от маршрутизатора пакетов. Пакеты, проходящие транзитом через маршрутизатор, в цепочку **output** не попадают.

При обработке цепочки правила обрабатываются сверху вниз. Если пакет соответствует критерию правила, то для этого пакета выполняется соответствующее действие, указанное в правиле, и движение пакета по цепочке прекращается (если в правило не добавлено действие **passthrough**). Если пакет не соответствует ни одному правилу, то к нему применяется политика по умолчанию.

**Доступные политики по умолчанию:**

- **accept –** принять пакет
- **drop –** отклонить пакет (без отправки ICMP-сообщения об отклонении)

Обычно пакет должен соответствовать нескольким критериям. Общие правила фильтрации могут группироваться в отдельной цепочке. Чтобы прогнать пакет по правилам дополнительных цепочек, действие jump должно использовать адрес правил в пределах другой цепочки.

Цепочки не могут быть удалены, если они содержат правила.

Примечание

В виду того что правила NAT обрабатываются первыми, важно помнить об этом при построении межсетевого экрана, поскольку пакет уже может быть изменен NATом.

Пакеты, следующие через маршрутизатор транзитом, не обрабатываются правилами input или output.

Будьте осторожны при изменении политики по умолчанию для цепочек input или output! Вы можете потерять связь с маршрутизатором.

**Применение межсетевого экрана**

**Описание**

В этой секции будут рассмотрены некоторые примеры использования файервола.

## Базовые принципы построения файервола.

Предположим, у нас есть маршрутизатор, соединяющий клиентскую сеть с Интернетом. Базовые принципы настройки файервола могут быть сформулированы следующим образом.

- **Защита маршрутизатора от несанкционированного доступа**

При подключении к маршрутизатору производить проверку IP-адреса. Разрешить доступ к маршрутизатору только с определенных хостов на определенные порты. Это может быть достигнуто путем настройки правил цепочки input на соответствие пакетов с адресом назначения маршрутизатора на всех интерфейсах.

- **Защита пользовательских хостов**

Должен быть разрешен доступ только к определенным хостам и службам. Это может быть достигнуто путем настройки цепочки forward на проверку соответствия пакетов, проходящих транзитом через маршрутизатор в клиентскую сеть.

- **Использование маскарадинга для «маскировки» частной сети одним внешним адресом**

Все подключения с приватных адресов замаскарадены, и выходят во внешнюю сеть с адресом маршрутизатора. Это можно сделать путем включения маскарадинга.

- **Настройка политики использования Интернета клиентской сетью**

Подключения из клиентской сети должны проверяться. Это можно сделать путем настройки правил цепочки forward, и/или включением маскарадинга только для разрешённых подключений.

Фильтрация оказывает влияние на быстродействие маршрутизатора. Для снижения нагрузки на устройство правила, отслеживающие состояние соединения, должны находится сверху.

Ниже приводятся распространённые примеры настройки файервола**.**

**Пример настройки фильтров**

Допустим, адрес локальной сети 192.168.0.0/24, выход в Интернет осуществляется с ether1. Необходимо защитить маршрутизатор от несанкционированного доступа извне, разрешив доступ только из локальной сети. Также необходимо разрешить пакеты ICMP на всех интерфейсах, так, чтобы можно было пинговать роутер из внешней сети.

```
/ip firewall filter> add chain=input connection-state=invalid action=drop \
comment="Drop Invalid connections"
/ip firewall filter> add chain=input connection-state=established action=accept \
comment="Allow Established connections"
/ip firewall filter> add chain=input protocol=icmp action=accept \
comment="Allow ICMP"
/ip firewall filter> add chain=input src-address=192.168.0.0/24 action=accept \
in-interface=!ether1
/ip firewall filter> add chain=input action=drop comment="Drop everything else"
```

**Защита клиентской сети**

Для защиты клиентской сети нам необходимо проверять все пакеты, проходящие через маршрутизатор, пропуская нужные и уничтожая ненужные пакеты. Для icmp, tcp и udp-трафика мы создадим цепочки правил, где будут уничтожаться ненужные пакеты.

```
/ip firewall filter
add chain=forward protocol=tcp connection-state=invalid \
        action=drop comment="drop invalid connections"
add chain=forward connection-state=established action=accept \
        comment="allow already established connections"
add chain=forward connection-state=related action=accept \
        comment="allow related connections"
```

Блокируем служебный трафик:

```
add chain=forward src-address=0.0.0.0/8 action=drop
add chain=forward dst-address=0.0.0.0/8 action=drop
add chain=forward src-address=127.0.0.0/8 action=drop
add chain=forward dst-address=127.0.0.0/8 action=drop
add chain=forward src-address=224.0.0.0/3 action=drop
add chain=forward dst-address=224.0.0.0/3 action=drop
```

Добавляем переходы на новые цепочки правил:

```
add chain=forward protocol=tcp action=jump jump-target=tcp
add chain=forward protocol=udp action=jump jump-target=udp
add chain=forward protocol=icmp action=jump jump-target=icmp
```

Создаем цепочку «**tcp**» и блокируем в ней некоторые tcp-порты:

```
add chain=tcp protocol=tcp dst-port=69 action=drop \
        comment="deny TFTP"
add chain=tcp protocol=tcp dst-port=111 action=drop \
        comment="deny RPC portmapper"
add chain=tcp protocol=tcp dst-port=135 action=drop \
        comment="deny RPC portmapper"
add chain=tcp protocol=tcp dst-port=137-139 action=drop \
        comment="deny NBT"
add chain=tcp protocol=tcp dst-port=445 action=drop \
        comment="deny cifs"
add chain=tcp protocol=tcp dst-port=2049 action=drop comment="deny NFS"
add chain=tcp protocol=tcp dst-port=12345-12346 action=drop comment="deny NetBus"
add chain=tcp protocol=tcp dst-port=20034 action=drop comment="deny NetBus"
add chain=tcp protocol=tcp dst-port=3133 action=drop comment="deny BackOriffice"
add chain=tcp protocol=tcp dst-port=67-68 action=drop comment="deny DHCP"
```

Создаем цепочку «**udp**» и блокируем в ней udp-порты:

```
add chain=udp protocol=udp dst-port=69 action=drop comment="deny TFTP"
add chain=udp protocol=udp dst-port=111 action=drop comment="deny PRC portmapper"
add chain=udp protocol=udp dst-port=135 action=drop comment="deny PRC portmapper"
add chain=udp protocol=udp dst-port=137-139 action=drop comment="deny NBT"
add chain=udp rotocol=udp dst-port=2049 action=drop comment="deny NFS"
add chain=udp protocol=udp dst-port=3133 action=drop comment="deny BackOriffice"
```

Создаем цепочку «**icmp**» и разрешаем в ней прохождение некоторых типов icmp-пакетов:

```
add chain=icmp protocol=icmp icmp-options=0:0 action=accept \
        comment="echo reply"
add chain=icmp protocol=icmp icmp-options=3:0 action=accept \
        comment="net unreachable"
add chain=icmp protocol=icmp icmp-options=3:1 action=accept \
        comment="host unreachable"
add chain=icmp protocol=icmp icmp-options=3:4 action=accept \
        comment="host unreachable fragmentation required"
add chain=icmp protocol=icmp icmp-options=4:0 action=accept \
        comment="allow source quench"
add chain=icmp protocol=icmp icmp-options=8:0 action=accept \
        comment="allow echo request"
add chain=icmp protocol=icmp icmp-options=11:0 action=accept \
        comment="allow time exceed"
add chain=icmp protocol=icmp icmp-options=12:0 action=accept \
        comment="allow parameter bad"
add chain=icmp action=drop comment="deny all other types"
```

**Пример настройки маскарадинга**

Если вы хотите скрыть клиентскую сеть 192.168.0.0/24 за адресом маршрутизатора 10.5.8.217, вам необходимо произвести трансляцию исходящих адресов (маскарадинг). Маскарадинг будет подменять исходящий адрес и порт всех пакетов, идущих из сети 192.168.0.0/24 на адрес маршрутизатора 10.5.8.217 при прохождении пакетов через маршрутизатор.

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=Public
```

Теперь все исходящие пакеты в качестве адреса-источника будут иметь адрес 10.5.8.217 маршрутизатора. Доступ в локальную сеть из Интернета будет закрыт. Если вы хотите дать доступ к ресурсам внутри сети, необходимо использовать destination NAT.

**Пример настройки destination NAT**

Предположим, что необходимо сконфигурировать маршрутизатор так, чтобы локальный сервер был доступен из внешней сети. В свою очередь сервер также должен иметь доступ во внешнюю сеть.

Сервер, к которому надо дать доступ, имеет локальный IP-адрес 192.168.0.169.
Публичный IP-адрес маршрутизатора: 10.5.8.200

Добавляем публичный IP-адрес на интерфейс маршрутизатора:

```
/IP-адрес add address=10.5.8.200/32 interface=Public
```

Добавляем правило, разрешающее доступ к локальному серверу из внешней сети:

```
/ip firewall nat add chain=dstnat dst-address=10.5.8.200 action=dst-nat to-addresses=192.168.0.109
```

Добавляем правило, разрешающее доступ локального сервера во внешнюю сеть:

```
/ip firewall nat add chain=srcnat src-address=192.168.0.109 action=src-nat to-addresses=10.5.8.200
```

### Настройка фильтров в Firewall

Выполняя фильтрацию пакетов, файервол обеспечивает безопасное прохождение трафика через маршрутизатор. В сочетании с NAT данная функция представляет собой мощный инструмент, предотвращая несанкционированный доступ к локальным сетям / маршрутизатору и фильтруя исходящий трафик.

### Быстрая настройка фильтров

Для добавления правила, которое будет удалять все **TCP**-пакеты с портом назначения **135,** проходящие через маршрутизатор, используйте следующую команду:

```
/ip firewall filter add chain=forward dst-port=135 protocol=tcp action=drop
```

Для запрета доступа к маршрутизатору через Telnet (протокол TCP, порт 23) выполните следующую команду:

```
/ip firewall filter add chain=input protocol=tcp dst-port=23 action=drop
```

### Спецификация

Требуемые пакеты: **system**

Уровень подменю: **/ip firewall filter**

### Общее описание

Файервол защищает данные, находящиеся в локальной сети (ЛВС) от внешней угрозы. При объединении нескольких сетей в одну общую локальную сеть существует угроза вторжения извне. Такое вторжение может привести к краже или даже потере ценных данных. Файерволы используются как средство предотвращения или уменьшения рисков вторжения извне. Правильно настроенный файервол играет ключевую роль в эффективном развертывании безопасной инфраструктуры ЛВС. В RouterOS реализован мощный файервол со следующими ключевыми особенностями:

- полная фильтрация пакетов
- фильтрация по протоколу peer-to-peer
- классификация трафика по:
  - исходящему MAC-адресу
  - IP адресу/списку адресов и типам адресов (широковещательный, локальный, мультикаст, уникаст)
  - портам или диапазонам портов
  - IP-протоколам
  - опциям протоколов (различные типы ICMP-пакетов, TCP-флаги, IP-опции и MSS)
  - типу трафика (входящий/исходящий)

- внутреннему потоку и меткам соединений
- ToS (DSCP)
- лимитированию количества пакетов и состоянию счетчика
- размеру пакета
- времени поступления пакетов и т.д.

**Основные принципы фильтрации**

Все операции выполняются в соответствии с указанными правилами. Правила имеют определённый синтаксис и указывают маршрутизатору что именно сделать с текущим IP-пакетом. Каждое правило состоит из двух частей. Первая часть – это образец, который используется для сравнения с проходящим через маршрутизатор траффиком. Вторая часть – это действие, в котором определяется, что необходимо сделать в том случае, если проходящий пакет соответствует образцу. Для более удобного управления правила организованы в цепочки.

Существуют три стандартные цепочки: **input**, **forward** и **output,** эти цепочки отвечают за входящий, транзитный и исходящий трафик соответственно. При необходимости могут быть добавлены дополнительные пользовательские цепочки. Поскольку эти цепочки не имеют образцов для сопоставления с проходящим трафиком, в одну или более стандартных цепочек необходимо добавить правила, содержащие действия перехода на пользовательские цепочки (**jump** и **jump-target)**.

**Фильтрующие цепочки**

Как упоминалось выше, правила группируются в цепочки, что позволяет сравнить пакет сразу с несколькими правилами в каждой цепочке, и затем передать его на обработку другой цепочке. Например, пакет необходимо сравнить с парами **IP-адрес:порт**.
Конечно можно было бы добавить в цепочку **forward** несколько правил с образцами IP-**адрес:порт**, но более правильно было бы разделить это правило на две части: добавить одно правило для сопоставления с текущим IP-адресом, например:

**/ip firewall filter add src-address=1.1.1.2/32 jump-target=mychain**

и в случае совпадения передать управление IP-пакетом другой цепочке, в данном примере цепочке **mychain**. И затем, уже в отдельном правиле цепочки **mychain** провести сравнение портов, без проверки IP-адреса.

Ниже описываются три стандартные цепочки, которые не могут быть удалены:

- **input –** используется для обработки входящих на маршрутизатор пакетов, и предназначенных именно ему. Пакеты, проходящие транзитом через маршрутизатор, в цепочку **input** не попадают.
- **forward –** используется для обработки пакетов, идущих транзитом через маршрутизатор

- **output –** используется для обработки исходящих от маршрутизатора пакетов. Пакеты, проходящие транзитом через маршрутизатор, в цепочку **output** не попадают.

При прохождении цепочки правила, описанные в ней, рассматриваются сверху вниз. Если пакет соответствует критерию правила, то к нему применяется определенное ранее действие, и пакет прекращает свое движение по цепочке т.е. к нему более не применяется ни одного правила из этой цепочки (исключение составляет действие **passthrough**). Если для пакета не найдено совпадений в цепочке, то он принимается.

**Описание параметров**

**action** (accept | add-dst-to-address-list | add-src-to-address-list | drop | jump | log | passthrough | reject | return | tarpit; по умолчанию: **accept**) – действие, применяемое к пакету, если найдено соответствие в правиле.

- **accept –** принять пакет. Не предпринимать более никаких действий, то есть пакет принят и никакие правила к нему больше не применяются
- **add-dst-to-address-list –** добавить адрес назначения IP-пакета в список адресов, определенный в параметре **address-list**
- **add-src-to-address-list –** добавить адрес источника IP пакета в список адресов, указанный в параметре **address-list**
- **drop –** удалить пакет без отправки ICMP-сообщения об отклонении

  пакета

- **jump –** перейти в цепочку, указанную в параметре **jump-target**

- **log** – фиксировать каждое соответствие в системном журнале

- **passthrough –** игнорировать текущее правило и перейти к следующему

- **reject –** отклонить пакет и отправить ICMP-сообщение об отклонении

- **return –** вернуть контроль в то место родительской цепочки, откуда был совершен переход
- **tarpit –** захват и удержание входящих TCP соединений (отвечать SYN/ACK на входящий TCP SYN пакет)

**address-list** (название) – название списка адресов для сбора IP-адресов из правил, указанных в действиях **add-dst-to-address-list** или **add-src-to-address-list**. Эти списки адресов могут быть позже использованы в правилах.

**address-list-timeout** (время; по умолчанию: **00:00:00**) – временной интервал, после которого адрес будет удален из списка адресов **address-list.** Используется совместно с действиями **add-dst-to-address-list** или **add-src-to-address-list**

- **00:00:00 –** оставлять адрес в списке навсегда

**chain** (forward | input | output | название) – цепочка, в которой прописываются соотвтветствующие правила. Поскольку траффик проходит через различные правила, то будьте внимательны при выборе цепочки при добавлении нового правила. Если указанное в правиле название цепочки не будет соответствовать названию уже существующей цепочки, то будет создана новая цепочка.

**comment** (текст) – комментарий к правилу. Удобно использовать в скриптах для описания правил.

**connections-bytes** (целое значение-целое значение) – проверять пакеты, только если через установленное соединение прошло указанное количество байт.

- **0** – означает бесконечность, например: **connection-bytes=2000000-0** означает, что правило будет отрабатывать только в случае, если через соединение прошло более 2 000 000 байт.

**connection-limit** (целое значение, маска) – ограничение количества соединений по указанному адресу или диапазону адресов

**connection-mark** (название) – проверка пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle.** Если будет указан параметр **no-mark**, то будет проверяться соответствие правила всем немаркированным соединениям

**connection-rate** (целое значение 0..4294967295) – для RouterOS версии 3.3 и выше – проверка пакетов с зависимости от скорости соединения (см. раздел «Обработка пакетов в зависимости от скорости соединения» на стр. 82), например, пропускаем транзитный трафик при скорости соединения не более 500 кбайт:

```
/ip firewall filter
add action=accept chain=forward connection-rate=0-500k protocol=tcp
add action=accept chain=forward connection-rate=0-500k protocol=udp
```

**connection-state** (estabilished | invalid | new | related) – определение состояния соединений (трассировка соединений)

- **estabilished** – пакет, принадлежащий уже установленному соединению, например, когда узел передал пакет и получил на него ответ
- **invalid** – пакет, который не может быть опознан по каким-либо причинам. Например, при нехватке памяти или при получении *ICMP*-сообщения об ошибке, которое не соответствует какому-либо известному соединению. В подобном случае к пакету советуют применить действие **drop**.
- **new** – первый пакет в соединении (первый пакет, полученный трассировщиком).
- **related** – пакет, связанный с другим, уже установленным соединением, когда новое соединение инициировано из уже установленного соединения, имеющего признак **established**, например, соединение FTP-data, которое является связанным с портом *FTP* control, или DCC соединение, запущенное из *IRC*.

**connection-type (ftp | gre | h323 | irc | mms | pptp | quake3 | tftp) –** определение типа подключения, базирующееся на использовании данных, полученных при трассировке соединения. Если хосты расположены за маршрутизатором с настроенным NATом, то для прохождения трафика должен быть включён соответствующий хелпер командой /ip firewall service-port, например:

```
/ip firewall> service-port enable ftp
```

**content** (текст) – для соответствия правилу пакет должен содержать указанный текст

**dst-address** (IP-адрес/маска | IP-адрес | IP-адрес) – диапазон IP-адресов назначения. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**dst-address-list** (название) – соответствие адреса назначения, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**dst-address-type** (unicast | local | broadcast | multicast) – соответствие адреса назначения одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**dst-limit –** (целое значение | время | целое значение | адрес получателя | порт получателя | адрес источника | время) ограничение количества принимаемых в секунду пакетов, лимитируется по указанному IP-адресу назначения или по портам назначения. Каждый IP-адрес и порт назначения имеет свой лимит. Опции показаны ниже (в порядке появления)

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике
- **Mode** – указание IP-адресов/портов пакетов для лимитирования
- **Expire –** временной интервал, по истечении которого записи IP адресов/портов будут удалены

**dst-port** (целое значение: 0..65535- целое значение: 0..65535) – порт назначения или диапазон портов

**fragment** (yes | no) – соответствие правил для фрагментированных пакетов. Первый (стартовый) фрагмент не учитывается. Если включён механизм определения состояний

(conntrack), то правило неактуально, поскольку система будет автоматически собирать фрагментированные пакеты.

**hotspot** ( from-client | auth | local-dst | http) – соответствие правил для пакетов, полученных

от клиентов через различные хот-споты. Все значения могут быть инвертированы.

- **from-client –** истина, если пакет пришел от клиента HotSpot
- **auth –** истина, если пакет пришел от авторизованного клиента
- **local-dst –** истина, если пакет имеет локальный IP-адрес назначения
- **http** – истина, если это TCP-пакет от клиента + включен прозрачный прокси на 80 порту или на клиенте сконфигурирован адрес прокси и этот адрес равен адресу:порту IP-пакета

**icmp-options** (целое значение | целое значение) – соответствие полям ICMP Type:Code

**in-bridge-port** (название) – соответствие порту маршрутизатора, *принимающего* пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**in-interface** (название) – интерфейс, с которого пакет поступил в маршрутизатор

**ingress-priority –** соответствие приоритету пакета. Соответствующие биты приоритезации могут быть выставлены через VLAN / WMM.

**ipsec-policy –** соответствие используемой в IpSec политике. Значения записываются в формате: *направление, политика*. Направление используется для соответствия политике, используемой для декапсуляции или политике, которая будет использоваться для инкапсуляции.

- **in** – направление для цепочек PREROUTING, INPUT и FORWARD
- **out** – направление для цепочек POSTROUTING, OUTPUT и FORWARD

- **ipsec** – соответствие передачи пакета в туннельном режиме
- **none** – соответствие передачи пакета в транспортном режиме

**ipv4-options** (any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp) соответствие опциям в заголовке пакета IPv4

- **any –** соответствие хотя бы одной опции в заголовке пакета
- **loose-source-routing –** соответствие пакетов с опцией «Loose Source Routing».
- Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута может быть достигнут за **любое** количество шагов (хопов).

- **no-record-route –** соответствие пакетов с опцией «no record route» – с отсутствием информации о маршрутизации.
- **no-router-alert –** соответствие пакетов с опцией «No router alert»
- **no-source-routing** – соответствие пакетов с опцией «No sources routing»
- **no-timestamp –** соответствие пакетов с опцией отсутствия временной метки
- **record-route –** соответствие пакетов с опцией наличия записи о маршрутизации
- **router-alert –** соответствие пакетов с опцией «No router alert», когда маршрутизатор может перехватывать пакеты, не адресованные непосредственно ему, без значительного падения производительности
- **strict-source-routing –** соответствие пакетов с опцией «Strict Source Routing». Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута должен быть достигнут **строго** за 1 шаг (хоп)
- **timestamp** – соответствие пакетов с опцией наличия временной метки

**jump-target** (forward | input | output | название) – название целевой цепочки, в которую осуществляется переход, если используется действие **jump**

**layer7-protocol** (название) – протокол, обеспечивающий поиск указанных шаблонов в потоке пакетов (см. стр. 80).

**limit** (целое значение | время | целое значение) – лимитирование потока пакетов. Используется для уменьшения количества сообщений в логах

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике

**log-prefix** (текст) – все сообщения, записывающиеся в системный журнал, будут содержать указанный здесь префикс. Используется совместно с действием **log**

**nth** (целое значение | целое значение|: 0..15 | целое значение) – совпадение с правилом каждого n-ного пакета. Для подсчета пакетов может быть использован один из 16 доступных счетчиков.

- **Every –** соответствует каждому **Every+1th** пакету. Например, если **Every=1,** тогда правило должно соответствовать каждому второму пакету

- **Counter –** определяет какой счетчик использовать. Счетчик увеличивается на единицу каждый раз, когда правило находит соответствие в пакете

- **Packet –** соответствие правилу пакета с указанным номером. Номер должен быть в интервале от 0 до **Every**. Если эта опция используется для конкретного счетчика, поэтому должно быть по крайней мере **Every+1** правило с этой опцией, перекрывающее все значения между **0** и **Every**.

**out-bridge-port –** соответствие порту маршрутизатора, *отправляющего* пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**out-interface** (название) – интерфейс, с которого пакет покидает маршрутизатор.

**p2p** (all-p2p | bit-torrent | blubster | direct-connect | edonkey | fasttrack | gnutella | soulseek | warez | winmx) – соответствие пакетов протоколам peer-to-peer (P2P)

**packet-mark** (текст) – соответствие пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle.**

**packet-size** (целое значение: 0..65535| целое значение: 0..65535) – соответствие пакета определенному размеру или размеру, указанному в диапазоне значений, заданному в байтах.

- **Min –** нижняя граница диапазона или отдельно взятого значения
- **Max –** верхняя граница диапазона

**per-connection-classifier –** соответствие разделению трафика на одинаковые потоки, с размещением пакета с определёнными параметрами (src-address, src-port, dst-address, dst-port) в отдельном потоке.

**port** (целое значение [-целое значение]: 0..65535) – соответствие указанному порту (или диапазону портов) источника или получателя. Допустимые типы протоколов: TCP или UDP.

**protocol** (ddp | egp | encap | ggp | gre | hmp | icmp | idrp -cmtp | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | iso-tp4 | ospf | pup | rdp | rspf | st | tcp | udp | vmtp | xns -idp | xtp | целое значение) -соответствие определенному имени или номеру IP-протокола.

**psd** (целое значение|время| целое значение| целое значение) – попытка определить сканирование TCP- и UDP-портов. Рекомендуется назначать небольшой вес пакета непривилегированным портам (>1024) для уменьшения количества ложных срабатываний, например, при пассивной передаче через FTP.

- **WeightThreshold** – весовое значение для последовательности TCP/UDP-пакетов, в заголовке которых указаны различные порты назначения, при этом сами пакеты поступили от одного и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.
- **DelayThreshold** – задержка между пакетами, в заголовке которых указаны различные порты назначения, при этом сами пакеты

поступили от одного и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.

- **LowPortWeight** – весовое значение пакетов при сканировании привилегированных (<=1024) портов
- **HighPortWeight** – весовое значение пакетов при сканировании непривилегированных (>1024) портов

**random** (целое значение 1..99) – соответствие взятым наугад пакетам

**reject-with** (icmp-admin-prohibited | icmp-echo-reply | icmp-host-prohibited | icmp-hostunreachable | icmp-net-prohibited | icmp-network-unreachable | icmp-port-unreachable | icmpprotocol-unreachable | tcp-reset | целое значение) изменение отправляемого пакета (с указанием причины отклонения) при действии **reject**

**routing-mark** (название) – соответствие пакетов, помеченных ранее меткой «routing mark» при прохождении таблицы **mangle**

**src-address** (IP-адрес| маска | IP-адрес | IP-адрес) – диапазон IP-адресов источника. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**src-address-list** (название) – соответствие адреса источника, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**src-address-type** (unicast | local | broadcast | multicast) – соответствие адреса источника одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**src-port** (целое значение: 0..65535- целое значение: 0..65535) – порт источника или диапазон портов

**src-mac-address** (MAC-адрес) – MAC-адрес источника

**tcp-flags** (ack | cwr | ece | fin | psh | rst | syn | urg) – соответствие следующим tcp-флагам:

- **ack** (Acknowledgement field is significant) – используется для подтверждения получения данных
- **cwr** (Congestion Window Reduced) – окно перегрузки уменьшено – используется отправителем, и указывает, что получен пакет с установленным флагом ECE

- **ece** (ECN-echo) – используется для указания, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю информации о перегрузках в сети
- **fin** (Final) – указывает на завершение соединения
- **psh** (Push function) используется для информирования получателя о том, чтобы протолкнуть данные, накопившиеся в приемном буфере приложения пользователя
- **rst** (Reset the connection) – сброс соединения/очистка буфера
- **syn** (Synchronize sequence numbers) – синхронизация номеров последовательности (установка нового соединения)
- **urg** (Urgent pointer field is significant) – указатель важности данных

**tcp-mss** (целое значение: 0..65535) – соответствие IP-пакета значению TCP MSS

**time** (время | время | sat | fri | thu | wed | tue | mon | sun) – применяется для создания фильтра, основанного на времени и дате прибытия пакета или (для локально созданных пакетов) на времени и дате отправки пакета

**ttl** (целое значение: 0..255) – соответствие значению TTL

**Специальные команды**

**reset-counters** (id) – сброс статистики для указанных правил

**reset-counters-all** – сброс статистики для всех правил

**Примечание**

Поскольку правила NAT применяются первыми, важно помнить об этом при написании правил файервола: учтите, что оригинальный пакет мог быть уже изменен при помощи NAT.

**Примеры применения**

**Защита маршрутизатора**

Для защиты маршрутизатора недостаточно только смены пароля администратора, также необходимо организовать фильтрацию пакетов. Все пакеты, предназначенные маршрутизатору, проходят через цепочку **input**. Обратите внимание: пакеты, следующие транзитом через маршрутизатор, не обрабатываются в цепочке input.

```
/ ip firewall filter
add chain=input connection-state=invalid action=drop \
comment="Drop Invalid connections"
add chain=input connection-state=established action=accept \
comment="Allow Established connections"
add chain=input protocol=udp action=accept \
comment="Allow UDP"
add chain=input protocol=icmp action=accept \
comment="Allow ICMP"
add chain=input src-address=192.168.0.0/24 action=accept \
comment="Allow access to router from known network"
add chain=input action=drop comment="Drop anything else"
```

**Защита локальной сети**

Для защиты сети необходимо проверять весь трафик, проходящий через маршрутизатор, и блокировать нежелательный контент. Для icmp, udp и tcp-трафика создадим цепочки, в которых будем удалять все нежелательные пакеты

```
/ip firewall filter
add chain=forward protocol=tcp connection-state=invalid \
        action=drop comment="drop invalid connections"
add chain=forward connection-state=established action=accept \
        comment="allow already established connections"
add chain=forward connection-state=related action=accept \
        comment="allow related connections"
```

Блокируем служебный трафик:

```
add chain=forward src-address=0.0.0.0/8 action=drop
add chain=forward dst-address=0.0.0.0/8 action=drop
add chain=forward src-address=127.0.0.0/8 action=drop
add chain=forward dst-address=127.0.0.0/8 action=drop
add chain=forward src-address=224.0.0.0/3 action=drop
add chain=forward dst-address=224.0.0.0/3 action=drop
```

Добавляем переходы на новые цепочки правил:

```
add chain=forward protocol=tcp action=jump jump-target=tcp
add chain=forward protocol=udp action=jump jump-target=udp
add chain=forward protocol=icmp action=jump jump-target=icmp
```

Создаем цепочку «tcp» и блокируем в ней некоторые tcp-порты:

```
add chain=tcp protocol=tcp dst-port=69 action=drop \
        comment="deny TFTP"
add chain=tcp protocol=tcp dst-port=111 action=drop \
        comment="deny RPC portmapper"
add chain=tcp protocol=tcp dst-port=135 action=drop \
        comment="deny RPC portmapper"
add chain=tcp protocol=tcp dst-port=137-139 action=drop \
        comment="deny NBT"
add chain=tcp protocol=tcp dst-port=445 action=drop \
        comment="deny cifs"
add chain=tcp protocol=tcp dst-port=2049 action=drop comment="deny NFS"
add chain=tcp protocol=tcp dst-port=12345-12346 action=drop comment="deny NetBus"
add chain=tcp protocol=tcp dst-port=20034 action=drop comment="deny NetBus"
add chain=tcp protocol=tcp dst-port=3133 action=drop comment="deny BackOriffice"
add chain=tcp protocol=tcp dst-port=67-68 action=drop comment="deny DHCP"
```

Создаем цепочку «**udp**» и блокируем в ней udp-порты:

```
add chain=udp protocol=udp dst-port=69 action=drop comment="deny TFTP"
add chain=udp protocol=udp dst-port=111 action=drop comment="deny PRC portmapper"
add chain=udp protocol=udp dst-port=135 action=drop comment="deny PRC portmapper"
add chain=udp protocol=udp dst-port=137-139 action=drop comment="deny NBT"
add chain=udp rotocol=udp dst-port=2049 action=drop comment="deny NFS"
add chain=udp protocol=udp dst-port=3133 action=drop comment="deny BackOriffice"
```

Создаем цепочку «**icmp**» и разрешаем в ней прохождение некоторых типов icmp-пакетов:

```
add chain=icmp protocol=icmp icmp-options=0:0 action=accept \
        comment="echo reply"
add chain=icmp protocol=icmp icmp-options=3:0 action=accept \
        comment="net unreachable"
add chain=icmp protocol=icmp icmp-options=3:1 action=accept \
        comment="host unreachable"
add chain=icmp protocol=icmp icmp-options=3:4 action=accept \
        comment="host unreachable fragmentation required"
add chain=icmp protocol=icmp icmp-options=4:0 action=accept \
        comment="allow source quench"
add chain=icmp protocol=icmp icmp-options=8:0 action=accept \
        comment="allow echo request"
add chain=icmp protocol=icmp icmp-options=11:0 action=accept \
        comment="allow time exceed"
add chain=icmp protocol=icmp icmp-options=12:0 action=accept \
        comment="allow parameter bad"
add chain=icmp action=drop comment="deny all other types"
```

## Настройка Mangle

### Спецификации

Требуемые пакеты: **system**
Уровень подменю: **/ip firewall mangle**
Стандарты и технологии: IP
Аппаратное обеспечение: Зависит от количества правил mangle

### Mangle

Уровень подменю**: /ip firewall mangle**

### Описание

Основное назначение функционала таблицы Mangle – это пометка пакетов специальными метками для последующей обработки. Многие средства в RouterOS используют помеченные пакеты, например, VLAN, NAT, маршрутизация. Они дифференцируют пакеты, исходя из меток и обрабатывают их соответствующим образом. Помеченные пакеты используются только в границах маршрутизатора, они не передаются далее в сеть.

Дополнительный функционал Mangle – это изменение полей в заголовке пакета, например: TOS (DSCP), TTL и т.д.

### Описание параметров

**action** (accept | add-dst-to-address-list | add-src-to-address-list | change-dscp | change-mss | change-ttl | jump | log | mark-connection | mark-packet | mark-routing | passthrough | return | setpriority | strip-ipv4-options; по умолчанию: **accept**) – предпринимаемое действие, если пакет соответствует правилу

- **accept** – принять пакет. Не предпринимать более никаких действий, то есть пакет принят и никакие правила к нему больше не применяются
- **add-dst-to-address-list** – добавить адрес назначения IP-пакета в список адресов, определенный в параметре address-list
- **add-src-to-address-list –** добавить адрес источника IP пакета в список адресов, указанный в параметре **address-list**
- **change-dscp** – изменение поля DSCP (Differentiated Services Code Point – точка кода дифференцированных услуг) в соответствии со значением параметра **new-dscp**
- **change-mss** – изменение поля MSS (Maximum segment size – максимальный размер полезного блока данных в байтах) в соответствии со значением параметра **new-mss**
- **change-ttl –** изменение поля MSS (TTL – время жизни пакета) в соответствии со значением параметра **new-ttl**

- **change-ttl –** изменение поля MSS (TTL – время жизни пакета) в соответствии со значением параметра **new-ttl**
- **clear-df –** сброс флага запрета фрагментации, фактически разрешение фрагментации пакета
- **jump –** перейти в цепочку, указанную в параметре **jump-target**

- **log** – фиксировать каждое соответствие в системном журнале, включая следующие данные: in-interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port, размер пакета. После отработки данного действия происходит переход к следующему правилу (также как при отработке действия **passthrough)**
- **mark-connection** – маркировка всех пакетов текущего соединения в соответствии со значением параметра **new-connection-mark**, если есть соответствие текущему правилу
- **mark-packet –** маркировка пакета в соответствии со значением параметра **new-packet-mark**, если есть соответствие текущему правилу
- **mark-routing** маркировка пакета в соответствии со значением параметра new-**routing-mark,** метка учитывается при маршрутизации пакета.
- **passthrough –** игнорировать текущее правило и перейти к следующему

- **return –** вернуть контроль в то место родительской цепочки, откуда был совершен переход
- **set-priority –** установить необходимый приоритет на проходящих пакетах в соответствии со значением параметра **new-priority** при использовании VLAN или WMM на беспроводном интерфейсе
- **sniff- tzsp** – генерация потока данных, который может быть перенаправлен на сторонний сниффер, например, Wireshark
- **sniff-pc –** генерация потока данных, который может быть переправлен на другой маршрутизатор с RouterOS с установленным программным пакетом CALEA (Communications Assistance for Law Enforcement Act**)**
- **sniff-id** – уникальный идентификатор для дифференцировки потоков трафика (например, между различными пользователями или между клиентским и серверным **устройством)**
- **sniff-target –** IP-адрес устройства с установленным сниффером
- **sniff-target-port** – UDP-порт сниффера (например, порт 37008 для Wireshark)
- **strip-ipv4-options –** очистка IPv4-полей из заголовка IP-пакета

**address-list** (название) – название списка адресов для сбора IP-адресов из правил, указанных в действиях **add-dst-address-list** или **add-src-address-list**. Эти списки адресов могут быть позже использованы в правилах.

**address-list-timeout** (время; по умолчанию: **00:00:00**) – временной интервал, после которого адрес будет удален из списка адресов **address-list**. Используется совместно с действиями **add-dst-to-address-list** или **add-src-to-address-list**

- **00:00:00 –** оставлять адрес в списке навсегда

**chain** (forward | input | output | название) – цепочка, в которой прописываются соотвтветствующие правила. Поскольку траффик проходит через различные правила, то будьте внимательны при выборе цепочки при добавлении нового правила. Если указанное в правиле название цепочки не будет соответствовать названию уже существующей цепочки, то будет создана новая цепочка.

**comment** (текст) – комментарий к правилу. Удобно использовать в скриптах для описания правил.

- **0** – означает бесконечность, например: **connection-bytes=2000000-0** означает, что правило будет отрабатывать только в случае, если через соединение прошло более 2 000 000 байт.

**connection-limit** (целое значение, маска) – ограничение количества соединений по указанному адресу или диапазону адресов

**connection-mark** (название) – проверка пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle.** Если будет указан параметр **no-mark**, то будет проверяться соответствие правила всем немаркированным соединениям

**connection-rate** (целое значение 0..4294967295) – для RouterOS версии 3.3 и выше – проверка пакетов с зависимости от скорости соединения (см. раздел «Обработка пакетов в зависимости от скорости соединения» на стр. 82), например, пропускаем транзитный трафик при скорости соединения не более 500 кбайт:

```
/ip firewall filter
add action=accept chain=forward connection-rate=0-500k protocol=tcp
add action=accept chain=forward connection-rate=0-500k protocol=udpdd action=accept chain=forward connection-rate=0-500k protocol=tcp
add action=accept chain=forward connection-rate=0-500k protocol=udp
```

**connection-state** (estabilished | invalid | new | related) – определение состояния соединений (трассировка соединений)

- **estabilished** – пакет, принадлежащий уже установленному соединению, например, когда узел передал пакет и получил на него ответ
- **invalid** – пакет, который не может быть опознан по каким-либо причинам. Например, при нехватке памяти или при получении *ICMP*-сообщения об ошибке, которое не соответствует какому-либо известному соединению. В подобном случае к пакету советуют применить действие **drop**.
- **new –** первый пакет в соединении (первый пакет, полученный трассировщиком).
- **related –** пакет, связанный с другим, уже установленным соединением, когда новое соединение инициировано из уже установленного соединения, имеющего

признак **established**, например, соединение FTP-data, которое является связанным с портом *FTP* control, или DCC соединение, запущенное из *IRC.*

**connection-type (ftp | gre | h323 | irc | mms | pptp | quake3 | tftp) –** определение типа подключения, базирующееся на использовании данных, полученных при трассировке соединения. Если хосты расположены за маршрутизатором с настроенным NATом, то для прохождения трафика должен быть включён соответствующий хелпер командой /ip firewall service-port, например:

```
/ip firewall> service-port enable ftp
```

**content** (текст) – для соответствия правилу пакет должен содержать указанный текст

**dscp** (целое значение: 0..63) – соответствие полям DSCP в заголовке IP-пакета

**dst-address** (IP-адрес/маска | IP-адрес | IP-адрес) – диапазон IP-адресов назначения. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**dst-address-list** (название) – соответствие адреса назначения, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**dst-address-type** (unicast | local | broadcast | multicast) – соответствие адреса назначения одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**dst-limit –** (целое значение | время | целое значение | адрес получателя | порт получателя | адрес источника | время) ограничение количества принимаемых в секунду пакетов, лимитируется по указанному IP-адресу назначения или по портам назначения. Каждый IP-адрес и порт назначения имеет свой лимит. Опции показаны ниже (в порядке появления)

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике
- **Mode** – указание IP-адресов/портов пакетов для лимитирования
- **Expire –** временной интервал, по истечении которого записи IP адресов/портов будут удалены

**dst-port** (целое значение: 0..65535- целое значение: 0..65535) – порт назначения или диапазон портов

**fragment** (yes | no) – соответствие правил для фрагментированных пакетов. Первый (стартовый) фрагмент не учитывается. Если включён механизм определения состояний (conntrack), то правило неактуально, поскольку система будет автоматически собирать фрагментированные пакеты.

**hotspot** (from-client | auth | local-dst | http) – соответствие правил для пакетов, полученных

от клиентов через различные хот-споты. Все значения могут быть инвертированы.

- **from-client –** истина, если пакет пришел от клиента HotSpot
- **auth –** истина, если пакет пришел от авторизованного клиента
- **local-dst –** истина, если пакет имеет локальный IP-адрес назначения
- **http** – истина, если это TCP-пакет от клиента + включен прозрачный прокси на 80 порту или на клиенте сконфигурирован адрес прокси и этот адрес равен адресу:порту IP-пакета

**icmp-options** (целое значение | целое значение) – соответствие полям ICMP Type:Code

**in-bridge-port** (название) – соответствие порту маршрутизатора, принимающего пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**in-interface** (название) – интерфейс, с которого пакет поступил в маршрутизатор

**ingress-priority –** соответствие приоритету пакета. Соответствующие биты приоритезации могут быть выставлены через VLAN / WMM.

**ipsec-policy –** соответствие используемой в IpSec политике. Значения записываются в фомате: *направление, политика.* Направление используется для соответствия политике, используемой для декапсуляции или политике, которая будет использоваться для инкапсуляции.

- **in** – направление для цепочек PREROUTING, INPUT и FORWARD
- **out** – направление для цепочек POSTROUTING, OUTPUT и FORWARD

- **ipsec** – соответствие передачи пакета в туннельном режиме
- **none** – соответствие передачи пакета в транспортном режиме

**ipv4-options** (any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp) соответствие опциям в заголовке пакета IPv4

- **any –** соответствие хотя бы одной опции в заголовке пакета

- **loose-source-routing –** соответствие пакетов с опцией «Loose Source Routing».
- Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута может быть достигнут за **любое** количество шагов (хопов).
- **no-record-route –** соответствие пакетов с опцией «no record route» – с отсутствием информации о маршрутизации.
- **no-router-alert –** соответствие пакетов с опцией «No router alert»
- **no-source-routing** – соответствие пакетов с опцией «No sources routing»
- **no-timestamp –** соответствие пакетов с опцией отсутствия временной метки
- **record-route –** соответствие пакетов с опцией наличия записи о маршрутизации
- **router-alert –** соответствие пакетов с опцией «No router alert», когда маршрутизатор может перехватывать пакеты, не адресованные непосредственно ему, без значительного падения производительности
- **strict-source-routing –** соответствие пакетов с опцией «Strict Source Routing». Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута должен быть достигнут **строго** за 1 шаг (хоп)
- **timestamp** – соответствие пакетов с опцией наличия временной метки

**jump-target** (forward | input | output | название) – название целевой цепочки, в которую осуществляется переход, если используется действие **jump**

**layer7-protocol** (название) – протокол, обеспечивающий поиск указанных шаблонов в потоке пакетов (см. стр. 80).

**limit** (целое значение | время | целое значение) – лимитирование потока пакетов. Используется для уменьшения количества сообщений в логах

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике

**log-prefix** (текст) – все сообщения, записывающиеся в системный журнал, будут содержать указанный здесь префикс. Используется совместно с действием **log**

**new-connection-mark** (название) – значение метки, используемой в действии **mark-connection**
**new-dscp** (целое значение: 0..63) – значение метки, используемой при изменении полей DSCP в действии **change-dscp**

**new-mss** (целое значение) – значение метки, используемой в действии **change-mss**

**new-packet-mark** (название) – значение метки, используемой в действии **action-mark-packet**

**new-priority** (целое значение) – значение метки, используемой в действии **set-priority** для указания приоритета интерфейса

- **from-dscp –** установить приоритет пакета, взятого из значения поля DSCP
- **from-dscp-high-3-bits –** установить приоритет пакета, взятого из старших 3 бит поля DSCP
- **from-ingress –** установить приоритет пакета, взятого из INGRESS (при использовании VLAN или WMM на беспроводном интерфейсе; **0** если не установлено)

**new-routing-mark** (название) – значение метки, используемой в действии **mark-routing**

- max-reliability – максимальная надежность (ToS=4)
- max-throughput – максимальная пропускная способность (ToS=8)
- min-cost – минимальная стоимость (ToS=2)
- min-delay – минимальная задержка (ToS=16)
- normal – нормальное обслуживание (ToS=0)

**new-ttl** (decrement | increment | set | целое значение) – определение нового значения поля TTL используется в паре с **action=change-ttl**

- **decrement –** значение поля TTL будет уменьшено на указанное значение
- **increment –** значение поля TTL будет увеличено на указанное значение
- **set:** значение поля TTL будет установлено в указанное значение

**nth** (целое значение | целое значение|: 0..15 | целое значение) – совпадение с правилом каждого n-ного пакета. Для подсчета пакетов может быть использован один из 16 доступных счетчиков.

- **Every –** соответствует каждому **Every+1th** пакету. Например, если **Every=1,** тогда правило должно соответствовать каждому второму пакету
- **Counter –** определяет какой счетчик использовать. Счетчик увеличивается на единицу каждый раз, когда правило находит соответствие в пакете
- **Packet –** соответствие правилу пакета с указанным номером. Номер должен быть в интервале от 0 до **Every**. Если эта опция используется для конкретного счетчика, поэтому должно быть по крайней мере **Every+1** правило с этой опцией, перекрывающее все значения между **0** и **Every**.

**out-bridge-port –** соответствие порту маршрутизатора, отправляющего пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**out-interface** (название) – интерфейс, с которого пакет покидает маршрутизатор.

**p2p** (all-p2p | bit-torrent | blubster | direct-connect | edonkey | fasttrack | gnutella | soulseek | warez | winmx) – соответствие пакетов протоколам peer-to-peer (P2P)

**packet-mark** (текст) – соответствие пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle.**

**packet-size** (целое значение: 0..65535| целое значение: 0..65535) – соответствие пакета определенному размеру или размеру, указанному в диапазоне значений, заданному в байтах.

- **Min –** нижняя граница диапазона или отдельно взятого значения
- **Max –** верхняя граница диапазона

**per-connection-classifier –** соответствие разделению трафика на одинаковые потоки, с размещением пакета с определёнными параметрами (src-address, src-port, dst-address, dst-port) в отдельном потоке.

**port** (целое значение [-целое значение]: 0..65535) – соответствие указанному порту (или диапазону портов) источника или получателя. Допустимые типы протоколов: TCP или UDP.

**protocol** (ddp | egp | encap | ggp | gre | hmp | icmp | idrp -cmtp | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | iso-tp4 | ospf | pup | rdp | rspf | st | tcp | udp | vmtp | xns -idp | xtp | целое значение) -соответствие определенному имени или номеру IP-протокола.

**psd** (целое значение|время| целое значение| целое значение) – попытка определить сканирование TCP- и UDP-портов. Рекомендуется назначать небольшой вес пакета непривилегированным портам (>1024) для уменьшения количества ложных срабатываний, например, при пассивной передаче через FTP.

- **WeightThreshold** – весовое значение для последовательности TCP/UDP-пакетов, в заголовке которых указаны различные порты назначения, при этом сами пакеты поступили от одного и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.
- **DelayThreshold** – задержка между пакетами, в заголовке которых указаны различные порты назначения, при этом сами пакеты поступили от одного и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.
- **LowPortWeight** – весовое значение пакетов при сканировании привилегированных (<=1024) портов
- **HighPortWeight** – весовое значение пакетов при сканировании непривилегированных (>1024) портов

**random** (целое значение 1..99) – соответствие взятым наугад пакетам

**reject-with** (icmp-admin-prohibited | icmp-echo-reply | icmp-host-prohibited | icmp-hostunreachable | icmp-net-prohibited | icmp-network-unreachable | icmp-port-unreachable |

icmpprotocol-unreachable | tcp-reset | целое значение) изменение отправляемого пакета (с указанием причины отклонения) при действии **reject**

**routing-mark** (название) – соответствие пакетов, помеченных ранее меткой «routing mark» при прохождении таблицы **mangle**

**src-address** (IP-адрес| маска | IP-адрес | IP-адрес) – диапазон IP-адресов источника. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**src-address-list** (название) – соответствие адреса источника, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**src-address-type** (unicast | local | broadcast | multicast) – соответствие адреса источника одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**src-port** (целое значение: 0..65535- целое значение: 0..65535) – порт источника или диапазон портов

**src-mac-address** (MAC-адрес) – MAC-адрес источника

**tcp-flags** (ack | cwr | ece | fin | psh | rst | syn | urg) – соответствие следующим tcp-флагам:

- **ack** (Acknowledgement field is significant) – используется для подтверждения получения данных
- **cwr** (Congestion Window Reduced) – окно перегрузки уменьшено – используется отправителем, и указывает, что получен пакет с установленным флагом ECE
- **ece** (ECN-echo) – используется для указания, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю информации о перегрузках в сети
- **fin** (Final) – указывает на завершение соединения
- **psh** (Push function) используется для информирования получателя о том, чтобы протолкнуть данные, накопившиеся в приемном буфере приложения пользователя
- **rst** (Reset the connection) – сброс соединения/очистка буфера
- **syn** (Synchronize sequence numbers) – синхронизация номеров последовательности (установка нового соединения)
- **urg** (Urgent pointer field is significant) – указатель важности данных

**tcp-mss** (целое значение: 0..65535) – соответствие IP-пакета значению TCP MSS

**time** (время | время | sat | fri | thu | wed | tue | mon | sun) – применяется для создания фильтра, основанного на времени и дате прибытия пакета или (для локально созданных пакетов) на времени и дате отправки пакета

**ttl** (целое значение: 0..255) – соответствие значению TTL

### Просмотр статистики

```
/ip firewall filter print stats
```

**Параметры**:

**Bytes** – количество байт, соответствующих правилу
**Packets** – количество пакетов, соответствующих правилу

По умолчанию эта команда эквивалентна **print static** и отображает только статические правила:

```
/ip firewall mangle> print stats
Flags: X – disabled, I – invalid, D – dynamic
 #   CHAIN          ACTION          BYTES        PACKETS
 0   prerouting     mark-routing     17478158     127631
 1   prerouting     mark-routing     782505       4506
```

Для отображения динамических правил используйте **print dynamic:**

```
/ip firewall mangle> print stats dynamic
Flags: X – disabled, I – invalid, D – dynamic
 #   CHAIN          ACTION          BYTES        PACKETS
 0 D forward        change-mss      0            0
 1 D forward        change-mss      0            0
 2 D forward        change-mss      0            0
 3 D forward        change-mss      132444       2079
```

Для отображения всех правил используйте **print all**:

```
/ip firewall mangle> print all stats
Flags: X – disabled, I – invalid, D – dynamic
 #   CHAIN          ACTION          BYTES        PACKETS
 0   prerouting     mark-routing     17478158     127631
 1   prerouting     mark-routing     782505       4506
 2 D forward        change-mss      0            0
 3 D forward        change-mss      0            0
 4 D forward        change-mss      0            0
 5 D forward        change-mss      129372       2031
```

**Специальные команды**

**reset-counters** (id) – сброс статистики для указанных правил

**reset-counters-all** – сброс статистики для всех правил

**Примечание**

Если вы хотите промаркировать пакет, соединение или маршрут и завершить обработку события (другими словами – промаркировать пакет и принять его), то вместо создания двух правил вы можете отключить установленный по умолчанию параметр **passthrough** у промаркированного правила.

Как правило маркировка маршрута не требуется для P2P-соединений, поскольку P2P-трафик всегда направляется на шлюз по умолчанию.

**Примеры использования**

Ниже показаны некоторые примеры использования средств mangle.

**Маркировка пакетов**

**Необходимо:**

- Пометить все tcp-пакеты, с адресом получателя, находящимся в первом списке адресов, за исключением пакетов, приходящих на 80 порт,
- Пометить все udp-пакеты с адресом получателя, находящимся во втором списке адресов

Первый вариант:

```
/ip firewall mangle  add chain=forward protocol=tcp port=!80 dst-address-list=first action=mark-packet new-packet-mark=first
/ip firewall mangle  add chain=forward protocol=udp dst-address-list=second action=mark-packet new-packet-mark=second
```

Настройка выглядит достаточно простой и в общем-то будет работать в небольших сетях. Но если количество подобных правил исчисляется сотнями, а сеть перегружена трафиком, то процессор будет достаточно сильно загружен. Причина заключается в том, что *каждое* правило будет анализировать заголовок *каждого* IP-пакета, на что потребуется значительное количество системных ресурсов.

Но если включён механизм определения состояний (conntrack), то из создавшейся ситуации есть выход: использовать метки соединения для оптимизации наших настроек.

Второй вариант:

```
/ip firewall mangle add chain=forward protocol=tcp port=!80 dst-address-list=first connection-state=new \
action=mark-connection new-connection-mark=first
 /ip firewall mangle add chain=forward connection-mark=first action=mark-packet new-packet-mark=first passthrough=no

 /ip firewall mangle add chain=forward protocol=udp dst-address-list=second connection-state=new \
action=mark-connection new-connection-mark=second
/ip firewall mangle add chain=forward connection-mark=second action=mark-packet new-packet-mark=second passthrough=no
```

Теперь первое правило будет искать соответствие в заголовке только первого пакета соединения, когда соединение находится в состоянии NEW, и в случае совпадения добавит необходимую метку. Следующие правила уже не будут проверять заголовок каждого пакета: будет проверяться лишь метка соединения, что значительно снизит нагрузку на процессор. Дополнительно отключен параметр passthrough, что еще больше снижает нагрузку на CPU.

### Маркировка трафика Peer-to-Peer

Для гарантированного улучшения качества работы сетевых соединений, интерактивный трафик (например, передача голоса или видео) должен быть приоритетен по отношению к не интерактивному, такому как peer-to-peer. QOS осуществляет маркировку трафика различного типа, и затем помещает его в очереди с различными приоритетами.

В представленном ниже примере P2P-трафик получит не более 1 Мбит/с от всей полосы, а всему остальному трафику разрешается использовать всю доступную полосу:

```
/ip firewall mangle add chain=forward p2p=all-p2p action=mark-connection new-connection-mark=p2p_conn
/ip firewall mangle add chain=forward connection-mark=p2p_conn action=mark-packet new-packet-mark=p2p
/ip firewall mangle add chain=forward connection-mark=!p2p_conn action=mark-packet new-packet-mark=other

/ip firewall mangle print
Flags: X – disabled, I – invalid, D – dynamic
0 chain=forward p2p=all-p2p action=mark-connection new-connection-mark=p2p_conn
1 chain=forward connection-mark=p2p_conn action=mark-packet new-packet-mark=p2p
2 chain=forward packet-mark=!p2p_conn action=mark-packet new-packet-mark=other

/queue tree add parent=Public packet-mark=p2p limit-at=1000000 max-limit=100000000 priority=8
/queue tree add parent=Local packet-mark=p2p limit-at=1000000 max-limit=100000000 priority=8
/queue tree add parent=Public packet-mark=other limit-at=1000000 max-limit=100000000 priority=1
/queue tree add parent=Local packet-mark=other limit-at=1000000 max-limit=100000000 priority=1
```

### Маркировка по MAC-адресам

Для маркировки трафика по известным MAC-адресам, поступающим к маршрутизатору (или проходящих через маршрутизатор), выполните следующее:

```
/ ip firewall mangle add chain=prerouting \
src-mac-address=00:01:29:60:36:E7 action=mark-connection new-connection-mark=known_mac_conn
/ ip firewall mangle add chain=prerouting \
connection-mark=known_mac_conn action=mark-packet new-packet-mark=known_mac
```

### Изменение MSS

Как известно, в VPN-соединении пакеты имеют меньший размер из-за инкапсуляции. Большие пакеты с MSS, превышающим допустимые размеры блока данных в VPN-соединении, для успешной пересылки должны быть предварительно фрагментированы. Однако, если в пакете установлен флаг DF (запрет фрагментации), то такой пакет не может быть фрагментирован и будет удален. В соединениях, не поддерживающих технологию PMTUD (Path MTU discovery), это может создать ряд проблем, включая проблемы с передачей данных по протоколам FTP и HTTP, и работу с почтовыми сервисами.

Уменьшение размера блока данных IP-пакета (MSS), проходящего через VPN-соединение, может решить эту проблему. Приведенный ниже пример демонстрирует как можно уменьшить значение MSS средствами mangle:

```
/ip firewall mangle add out-interface=pppoe-out protocol=tcp tcp-flags=syn action=change-mss new-mss=1300 chain=forward

/ip firewall mangle print
Flags: X – disabled, I – invalid, D – dynamic
0 chain=forward out-interface=pppoe-out protocol=tcp tcp-flags=syn
action=change-mss new-mss=1300
```

## Настройка NAT

### Спецификации

Требуемые пакеты: **system**
Уровень подменю: **/ip firewall nat**
Стандарты и технологии: IP, RFC1631, RFC2663

### NAT

Преобразование сетевых адресов (NAT) – это механизм, позволяющим устройствам при взаимодействии в рамках локальной сети использовать одни IP-адреса, а для выхода в интернет – другие. Сеть, использующая NAT, обозначается как внутренняя. Для работы NAT необходим соответствующий шлюз (маршрутизатор с поддержкой NAT), обеспечивающий преобразование IP-адресов пакетов, выходящих из внутренней сети во внешнюю и наоборот.

Существует два типа NAT:

- Набор правил NAT для исходящего трафика или **srcnat.** Преобразует пакеты, выходящие из внутренней сети. Маршрутизатор заменяет локальный адрес источника в заголовке IP-пакетов, выходящих во внешнюю сеть, на новый публичный IP-адрес. Для входящих пакетов выполняется обратное преобразование
- Набор правил NAT для входящего трафика или **dstnat.** Преобразует пакеты, поступающие из внешней сети во внутреннюю. В основном применяется в тех случаях, когда необходимо обеспечить доступ из внешней сети к устройствам внутренней сети. Здесь маршрутизатор заменяет адрес получателя в заголовке IP-пакетов, поступающих из внешней сети, на новый локальный IP-адрес.

### Недостатки NAT

Один из недостатков NAT – невозможность установить реальное соединение типа точка-точка между хостами, находящимися во внешней и внутренней сети. Поэтому некоторые Интернет-протоколы могут не работать через NAT, например, протокол аутентификации AH из набора IPSec. RouterOS решает эту проблему за счет поддержки технологии Universal Plug & Play.

### Перенаправление и маскарадинг

Перенаправление и маскарадинг – это специальные формы **dstnat** и **srcnat**, соответственно. Перенаправление можно сравнить с dstnat, в то время как маскарадинг – с srcnat, но без указания параметра **to-address,** поскольку IP-адрес исходящему интерфейсу присваивается автоматически. Та же ситуация и с перенаправлением: параметр **to-addresses** не указывается,

вместо него автоматически подставляется IP-адрес входящего интерфейса. Обратите внимание, что параметр **to-ports** остается актуальным для правил перенаправления, поскольку здесь указывается номер порта (или диапазона портов), который обрабатывает соответствующие запросы (например, web-proxy).
Когда над пакетом производится действие dstnat (не важно, будь то **action=nat** или **action=redirect**), адрес назначения изменяется. Информация о преобразовании адресов (включая настоящий адрес назначения) содержится в специальных таблицах маршрутизатора. Прозрачный web-прокси маршрутизатора (когда web-запросы перенаправляется на порт-прокси маршрутизатора) имеет доступ ко встроенным таблицам маршрутизатора, и, соответственно, извлекает из них необходимый адрес web-сервера. Если вы выполняете действие dstnat для других прокси серверов, то не существует возможности получить адрес web-сервера из заголовка IP-пакета (потому что адрес назначения IP-пакета, который изначально был адресован web-серверу, был изменен на адрес прокси-сервера). Начиная с версии HTTP/1.1 в запросе HTTP существует специальный заголовок, передающий адрес web-сервера, таким образом, прокси-сервер может использовать именно его, вместо адреса назначения IP-пакета. Если специальный заголовок в HTTP не используется (например, в старых версиях HTTP-клиентов), прокси-сервер не имеет возможности определить адрес web-сервера и не сможет работать.

Это означает, что невозможно правильно прозрачно перенаправлять HTTP-траффик от маршрутизатора к некоторым другим прозрачным прокси. Единственный способ сделать это – поднять прозрачный прокси на самом маршрутизаторе, прописав в его настройках реальный прокси как родительский. В этой ситуации реальный прокси уже не будет прозрачным, прозрачный прокси на маршрутизаторе будет перенаправлять проксированные запросы (эти запросы включают всю необходимую информацию о web-сервере) к реальному прокси.

**Описание параметров**

**action** (accept | add-dst-to-address-list | add-src-to-address-list | dst-nat | jump | log | masquerade | netmap | passthrough | redirect | return | same | src -nat; по умолчанию: **accept**) – предпринимаемое действие если пакет соответствует правилу.

- **accept** – принять пакет. Не предпринимать более никаких действий, то есть пакет принят и никакие правила NAT к нему больше не применяются
- **add-dst-to-address-list** – добавить адрес назначения IP-пакета в список адресов, определенный в параметре address-list
- **add-src-to-address-list –** добавить адрес источника IP пакета в список адресов, указанный в параметре address-list

- **dst-nat –** подменить адрес назначения и/или порт IP-пакета на значение, указанное в параметрах **to-address** и **to-ports**
- **jump –** перейти в цепочку, указанную в параметре **jump-target**

- **log –** фиксировать каждое соответствие в системном журнале, включая следующие данные: in-interface, out-interface, src-mac, protocol, src-ip:port->dst-ip:port, размер пакета. После отработки данного действия происходит переход к следующему правилу (также как при отработке действия **passthrough)**
- **masquerade** – автоматически заменить адреса источника IP-пакета на адрес, определенный средствами маршрутизации
- **netmap** – отображение одного адреса на другой (Чаще всего используется для раздачи реальных IP-адресов хостам локальной сети)
- **passthrough –** игнорировать текущее правило и перейти к следующему

- **redirect –** заменить адреса назначения IP-пакета на один из локальных адресов маршрутизатора
- **return –** вернуть контроль в то место родительской цепочки, откуда был совершен переход
- **same –** выдать клиенту для каждого соединения IP-адрес источника/назначения из выделенного диапазона адресов. Чаще всего используется службами, ожидающими одинаковые клиентские IP-адреса при множественных подключениях от одного клиента
- **src-nat –** подменить адрес источника и/или порт IP-пакета на значение, указанное в параметрах to-address и to-ports

**address-list** (название) – название списка адресов для сбора IP-адресов из правил, указанных в действиях **add-dst-address-list** или **add-src-address-list**. Эти списки адресов могут быть позже использованы в правилах.

**address-list-timeout** (время; по умолчанию: **00:00:00**) – временной интервал, после которого адрес будет удален из списка адресов **address-list**. Используется совместно с действиями **add-dst-to-address-list** или **add-src-to-address-list**

- **00:00:00 –** оставлять адрес в списке навсегда

**chain** (dstnat | srcnat | название) – цепочка, в которой прописываются соотвтветствующие правила. Поскольку траффик проходит через различные правила, то будьте внимательны при выборе цепочки при добавлении нового правила. Если указанное в правиле название цепочки не будет соответствовать названию уже существующей цепочки, то будет создана новая цепочка.

- **dstnat –** правило из этой цепочки применяется **до маршрутизации**. Это правило подменяет адрес назначения-IP пакета.
- **srcnat –** правило из этой цепочки применяется **после маршрутизации**. Это правило подменяет адрес источника-IP пакета.

**comment** (текст) – комментарий к правилу. Удобно использовать в скриптах для описания правил.

- **0** – означает бесконечность, например: **connection-bytes=2000000-0** означает, что правило будет отрабатывать только в случае, если через соединение прошло более 2 000 000 байт.

**connection-limit** (целое значение, маска) – ограничение количества соединений по указанному адресу или диапазону адресов

**connection-mark** (название) – проверка пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle**. Если будет указан параметр **no-mark**, то будет проверяться соответствие правила всем немаркированным соединениям

**connection-type (ftp | gre | h323 | irc | mms | pptp | quake3 | tftp) –** определение типа подключения, базирующееся на использовании данных, полученных при трассировке соединения. Если хосты расположены за маршрутизатором с настроенным NATом, то для прохождения трафика должен быть включён соответствующий хелпер командой /ip firewall service-port, например:

```
/ip firewall> service-port enable ftp
```

**content** (текст) – для соответствия правилу пакет должен содержать указанный текст

**dscp** (целое значение: 0..63) – соответствие полям DSCP в заголовке IP-пакета

**dst-address** (IP-адрес/маска | IP-адрес | IP-адрес) – диапазон IP-адресов назначения. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**dst-address-list** (название) – соответствие адреса назначения, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**dst-address-type** (unicast | local | broadcast | multicast) – соответствие адреса назначения одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**dst-limit –** (целое значение | время | целое значение | адрес получателя | порт получателя | адрес источника | время) ограничение количества принимаемых в секунду пакетов, лимитируется по указанному IP-адресу назначения или по портам назначения. Каждый IP-адрес и порт назначения имеет свой лимит. Опции показаны ниже (в порядке появления)

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике
- **Mode** – указание IP-адресов/портов пакетов для лимитирования

- **Expire –** временной интервал, по истечении которого записи IP адресов/портов будут удалены

**dst-port** (целое значение: 0..65535- целое значение: 0..65535) – порт назначения или диапазон портов

**fragment** (yes | no) – соответствие правил для фрагментированных пакетов. Первый (стартовый) фрагмент не учитывается. Если включён механизм определения состояний (conntrack), то правило неактуально, поскольку система будет автоматически собирать фрагментированные пакеты.

**hotspot** (from-client | auth | local-dst | http) – соответствие правил для пакетов, полученных от

клиентов через различные хот-споты. Все значения могут быть инвертированы.

- **from-client –** истина, если пакет пришел от клиента HotSpot
- **auth –** истина, если пакет пришел от авторизованного клиента
- **local-dst –** истина, если пакет имеет локальный IP-адрес назначения
- **http** – истина, если это TCP-пакет от клиента + включен прозрачный прокси на 80 порту или на клиенте сконфигурирован адрес прокси и этот адрес равен адресу:порту IP-пакета

**icmp-options** (целое значение | целое значение) – соответствие полям ICMP Type:Code

**in-bridge-port** (название) – соответствие порту маршрутизатора, принимающего пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**in-interface** (название) – интерфейс, с которого пакет поступил в маршрутизатор

**ingress-priority –** соответствие приоритету пакета. Соответствующие биты приоритезации могут быть выставлены через VLAN / WMM.

**ipsec-policy –** соответствие используемой в IpSec политике. Значения записываются в формате: *направление, политика*. Направление используется для соответствия политике, используемой для декапсуляции или политике, которая будет использоваться для инкапсуляции.

- **in** – направление для цепочек PREROUTING, INPUT и FORWARD
- **out** – направление для цепочек POSTROUTING, OUTPUT и FORWARD

- **ipsec** – соответствие передачи пакета в туннельном режиме
- **none** – соответствие передачи пакета в транспортном режиме

**ipv4-options** (any | loose-source-routing | no-record-route | no-router-alert | no-source-routing | no-timestamp | none | record-route | router-alert | strict-source-routing | timestamp) соответствие опциям в заголовке пакета IPv4

- **any –** соответствие хотя бы одной опции в заголовке пакета
- **loose-source-routing –** соответствие пакетов с опцией «Loose Source Routing».
- Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута может быть достигнут за **любое** количество шагов (хопов).
- **no-record-route –** соответствие пакетов с опцией «no record route» – с отсутствием информации о маршрутизации.
- **no-router-alert –** соответствие пакетов с опцией «No router alert»
- **no-source-routing** – соответствие пакетов с опцией «No sources routing»
- **no-timestamp –** соответствие пакетов с опцией отсутствия временной метки
- **record-route** – соответствие пакетов с опцией наличия записи о маршрутизации
- **router-alert –** соответствие пакетов с опцией «No router alert», когда маршрутизатор может перехватывать пакеты, не адресованные непосредственно ему, без значительного падения производительности
- **strict-source-routing –** соответствие пакетов с опцией «Strict Source Routing». Эта опция используется для направления дейтаграмм по маршруту, предопределенного в адресе-источнике, где очередной пункт требуемого маршрута должен быть достигнут **строго** за 1 шаг (хоп)
- **timestamp** – соответствие пакетов с опцией наличия временной метки

**jump-target** (forward | input | output | название) – название целевой цепочки, в которую осуществляется переход, если используется действие **jump**

**layer7-protocol** (название) – протокол, обеспечивающий поиск указанных шаблонов в потоке пакетов (см. стр. 80).

**limit** (целое значение | время | целое значение) – лимитирование потока пакетов. Используется для уменьшения количества сообщений в логах

- **Count –** максимальное среднее количество пакетов в секунду (pps), проходящее за время **Time**
- **Time –** интервал времени, в течение которого замеряется количество проходящих пакетов
- **Burst –** количество пакетов, проходящих в пике

**log-prefix** (текст) – все сообщения, записывающиеся в системный журнал, будут содержать указанный здесь префикс. Используется совместно с действием **log**

**nth** (целое значение | целое значение|: 0..15 | целое значение) – совпадение с правилом каждого n-ного пакета. Для подсчета пакетов может быть использован один из 16 доступных счетчиков.

- **Every –** соответствует каждому **Every+1th** пакету. Например, если **Every=1,** тогда правило должно соответствовать каждому второму пакету
- **Counter –** определяет какой счетчик использовать. Счетчик увеличивается на единицу каждый раз, когда правило находит соответствие в пакете
- **Packet –** соответствие правилу пакета с указанным номером. Номер должен быть в интервале от 0 до **Every**. Если эта опция используется для конкретного счетчика, поэтому должно быть по крайней мере **Every+1** правило с этой опцией, перекрывающее все значения между **0** и **Every**.

**out-bridge-port –** соответствие порту маршрутизатора, отправляющего пакеты, если он добавлен в bridge. Работает только в том случае, если включена настройка **use-ip-firewall** в параметрах прозрачного бриджа (/interface bridge settings set use-ip-firewall=yes).

**out-interface** (название) – интерфейс, с которого пакет покидает маршрутизатор.

**packet-mark** (текст) – соответствие пакетов, помеченных ранее определённой меткой при прохождении таблицы **mangle.**

**packet-size** (целое значение: 0..65535| целое значение: 0..65535) – соответствие пакета определенному размеру или размеру, указанному в диапазоне значений, заданному в байтах.

- **Min –** нижняя граница диапазона или отдельно взятого значения
- **Max –** верхняя граница диапазона

**per-connection-classifier –** соответствие разделению трафика на одинаковые потоки, с размещением пакета с определёнными параметрами (src-address, src-port, dst-address, dst-port) в отдельном потоке.

**port** (целое значение [-целое значение]: 0..65535) – соответствие указанному порту (или диапазону портов) источника или получателя. Допустимые типы протоколов: TCP или UDP.

**protocol** (ddp | egp | encap | ggp | gre | hmp | icmp | idrp -cmtp | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | iso-tp4 | ospf | pup | rdp | rspf | st | tcp | udp | vmtp | xns -idp | xtp | целое значение) -соответствие определенному имени или номеру IP-протокола

**psd** (целое значение|время| целое значение| целое значение) – попытка определить сканирование TCP- и UDP-портов. Рекомендуется назначать небольшой вес пакета непривилегированным портам (>1024) для уменьшения количества ложных срабатываний, например, при пассивной передаче через FTP.

- **WeightThreshold** – весовое значение для последовательности TCP/UDP-пакетов, в заголовке которых указаны различные порты назначения, при этом сами пакеты поступили от одного и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.
- **DelayThreshold** – задержка между пакетами, в заголовке которых указаны различные порты назначения, при этом сами пакеты поступили от одного

и того же хоста – такие последовательности рассматриваются как попытки сканирования портов.

- **LowPortWeight** – весовое значение пакетов при сканировании привилегированных (<=1024) портов
- **HighPortWeight** – весовое значение пакетов при сканировании непривилегированных (>1024) портов

**random** (целое значение 1..99) – соответствие взятым наугад пакетам

**routing-mark** (название) – соответствие пакетов, помеченных ранее меткой «routing mark» при прохождении таблицы **mangle**

**same-not-by-dst** (да|нет) – менять или не менять адрес назначения при выборе нового IP-адреса источника для пакетов, отобранных по правилам **action=same**

**src-address** (IP-адрес| маска | IP-адрес | IP-адрес) – диапазон IP-адресов источника. Обратите внимание: консоль автоматически преобразует неправильно введенный сетевой адрес в правильный, например: **1.1.1.1/24** будет преобразован в **1.1.1.0/24**

**src-address-list** (название) – соответствие адреса источника, указанного в заголовке пакета, адресу, находящемуся в **address-list**

**src-address-type** (unicast | local | broadcast | multicast) – соответствие адреса источника одному из типов IP-пакетов:

- **unicast –** IP адрес использован для соединения типа «точка –точка». В данном случае у пакета только один отправитель и один получатель
- **local** – адрес пакета соответствует одному из адресов, указанных на сетевых интерфейсах маршрутизатора
- **broadcast** – пакет, отправленный сразу всем устройствам подсети
- **multicast** – пакет, отправленный от одного отправителя нескольким получателям

**src-port** (целое значение: 0..65535- целое значение: 0..65535) – порт источника или диапазон портов

**src-mac-address** (MAC-адрес) – MAC-адрес источника

**tcp-flags** (ack | cwr | ece | fin | psh | rst | syn | urg) – соответствие следующим tcp-флагам:

- **ack** (Acknowledgement field is significant) – используется для подтверждения получения данных
- **cwr** (Congestion Window Reduced) – окно перегрузки уменьшено – используется отправителем, и указывает, что получен пакет с установленным флагом ECE

- **ece** (ECN-echo) – используется для указания, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю информации о перегрузках в сети
- **fin** (Final) – указывает на завершение соединения
- **psh** (Push function) используется для информирования получателя о том, чтобы протолкнуть данные, накопившиеся в приемном буфере приложения пользователя
- **rst** (Reset the connection) – сброс соединения/очистка буфера
- **syn** (Synchronize sequence numbers) – синхронизация номеров последовательности (установка нового соединения)
- **urg** (Urgent pointer field is significant) – указатель важности данных

**tcp-mss** (целое значение: 0..65535) – соответствие IP-пакета значению TCP MSS

**time** (время | время | sat | fri | thu | wed | tue | mon | sun) – применяется для создания фильтра, основанного на времени и дате прибытия пакета или (для локально созданных пакетов) на времени и дате отправки пакета

**to-addresses** (IP-адрес | IP-адрес) – подмена IP-адреса пакета (или диапазона адресов) на указанный

**to-ports** (целое значение: 0..65535 | целое значение: 0..65535 ) – подмена оригинального порта на указанный при обработке действий dst-nat, redirect, netmap, same и src-nat

**ttl** (целое значение: 0..255) – соответствие значению TTL

### Просмотр статистики

```
/ip firewall nat print stats
```

**Параметры**:

**Bytes** – количество байт, соответствующих правилу
**Packets** – количество пакетов, соответствующих правилу

По умолчанию эта команда эквивалентна **print static** и отображает только статические правила:

```
/ip firewall nat> print stats
Flags: X – disabled, I – invalid, D – dynamic
 #   CHAIN       ACTION          BYTES      PACKETS
 0   dstnat      dst-nat         0          0
 1   srcnat      src-nat         0          0
```

Для отображения динамических правил используйте **print dynamic,** а для отображения всех правил используйте **print all stats**

**Специальные команды**

**reset-counters** (id) – сброс статистики для указанных правил

**reset-counters-all** – сброс статистики для всех правил

### Примеры использования

Ниже показаны некоторые примеры использования NAT.

Предположим, необходимо, чтобы маршрутизатор делал следующее:

- "скрывал" локальную сеть за одним адресом
- предоставлял реальный IP-адрес локальному серверу
- создавал отображение 1:1 одной сети в другую

### Пример использования SRC-NAT (маскарадинг)

Если необходимо "скрыть" локальную сеть 192.168.0.0/24 за реальным IP-адресом 10.5.8.109, то можно воспользоваться функцией маскарадинга. Маскарадинг будет подменять IP-адрес и порт пакета, исходящего из сети 192.168.0.0/24, на адрес маршрутизатора 10.5.8.109 при прохождении пакета через него. Для этого добавим следующее правило:

```
/ip firewall nat add chain=srcnat action=masquerade out-interface=Public
```

В этом случае все исходящие соединения из сети 192.168.0.0/24 в качестве адреса источника будут иметь адрес маршрутизатора 10.5.8.109 и порт выше 1024. Доступ к локальной сети из Интернета будет невозможен. Если необходимо обеспечить доступ к локальной сети из внешней сети, то необходимо выполнить трансляцию адреса назначения (см. ниже).

**Обратите внимание:** маскарадинг лучше использовать в том случае, когда адрес источника заранее неизвестен, например, меняется динамически при подключении по DHCP. Если же в подключении используется статический IP-адрес, то оптимальным будет использование **srcnat**, это значительно снизит нагрузку на маршрутизатор.

### Пример использования DST-NAT

Перенаправление трафика из внешней сети во внутреннюю сеть

Допустим, необходимо обеспечить прохождение пакета из внешней сети с адресом 10.5.8.200 на хост в локальной сети с адресом 192.168.0.109 – в этом случае необходимо использовать трансляцию адреса назначения. А если необходимо обеспечить доступ локального хоста во внешнюю сеть, то дополнительно необходимо выполнить трансляцию адреса источника.

Для начала добавим реальный IP-адрес на внешний интерфейс маршрутизатора:

```
/IP-адрес add address=10.5.8.200/32 interface=Public
```

Затем добавим правило, разрешающее доступ к локальному хосту из внешней сети:

```
/ip firewall nat add chain=dstnat dst-address=10.5.8.200 action=dst-nat to-addresses=192.168.0.109
```

И в заключение добавим правило для трансляции исходящего адреса локального хоста во внешний адрес маршрутизатора:

```
/ip firewall nat add chain=srcnat src-address=192.168.0.109 action=src-nat to-addresses=10.5.8.200
```

### Пример переадресации данных на другой порт (Port mapping)

Допустим, необходимо перенаправлять все запросы из внешней сети, поступающие на порт 1234, на определённый порт хоста 192.168.1.1 в локальной сети:

```
/ip firewall nat add chain=dstnat dst-port=1234 action=dst-nat protocol=tcp to-address=192.168.1.1 to-port=5678
```

### Пример проброса портов на FTP-сервер в локальной сети

Допустим, FTP с адресом 192.168.0.109 работает **в пассивном режиме** за файерволом с адресом 10.5.8.200, таким образом между клиентом и сервером устанавливаются два соединения, и то и другое – со стороны клиента: поток управления (порт 21 сервера) и поток данных (случайный порт сервера), но только поток управления может быть проброшен при помощи dstnat. Поток данных связан с потоком управления и имеет статус **related,** что и мы и обозначаем во втором правиле:

```
/ip firewall nat add chain=dstnat dst-address=10.5.8.200 dst-port=21 protocol=tcp action=dst-nat to-addresses=192.168.0.109
/ip firewall filter add chain=forward connection-state=established, related action=accept
```

**Обратите внимание:** для обработки соединений со статусом **related** необходимо запустить службу FTP-хелпера:

```
/ip firewall> service-port enable ftp
```

**Обратите внимание:** если FTP работает **в активном режиме**, то между клиентом и сервером устанавливаются два соединения: поток управления (порт 21 сервера) и поток данных (порт 20 сервера), но первое соединение устанавливается со стороны клиента, второе – уже со стороны сервера. В таком (активном) режиме FTP может и не работать, если не только сервер, но и клиент находится за отдельным маршрутизатором с настроенным файерволом или NATом, поскольку второе соединение (поток данных) устанавливается сервером и не может получить прямой доступ к клиенту. Если клиент находится за маршрутизатором – также не забудьте проверить, что служба FTP-хелпера запущена:

```
/ip firewall service-port> print
Flags: X – disabled, I – invalid
 #   NAME                                PORTS
 0   ftp                                 21
 1   tftp                                69
 2   irc                                 6667
 3   h323
 4   sip                                 5060
                                         5061
 5   pptp
```

**Пример отображения одной сети в другую (один к одному)**

Допустим, необходимо отобразить внешнюю сеть 11.11.11.0/24 на сеть 2.2.2.0/24. В этом случае необходимо использовать трансляцию адреса источника и адреса назначения в паре с действием **action=netmap**

```
/ip firewall nat add chain=dstnat dst-address=11.11.11.0/24 action=netmap to-addresses=2.2.2.0/24

/ip firewall nat add chain=srcnat src-address=2.2.2.0/24 action=netmap to-addresses=11.11.11.0/24
```

## Списки адресов

### Спецификация

Требуемые пакеты: **system**
Стандарты и технологии: IP
Аппаратное обеспечение: не принципиально

Данный функционал позволяет создавать списки IP-адресов и группировать их под общим названием. Таблицы filter, mangle и NAT могут в дальнейшем использовать эти списки адресов для проверки соответствия пакетов прописанным правилам.

Записи в списке адресов могут обновляться автоматически через действия **add-src-to-address-list** или **add-dst-to-address-list,** используемые в таблицах NAT, mangle и filter.

### Описание параметров

**address** (IP-адрес/маска | IP-адрес-IP-адрес) – IP-адрес или диапазон адресов, добавляемый в список. Введенный диапазон адресов, например, 192.168.0.0-192.168.1.255, будет автоматически сконвертирован в адрес с маской: 192.168.0.0/23

**list** (название) – название списка IP-адресов

Пример

Приведенный ниже пример создает список адресов, подключающихся к 23 порту (telnet) маршрутизатора, после чего сбрасывает весь исходящий от них трафик в течение 5 минут. Дополнительно список адресов будет содержать один статический адрес 192.0.34.166/32 (www.example.com).

```
/ip firewall address-list add list=drop_traffic address=192.0.34.166/32
/ip firewall mangle add action=add-src-to-address-list address-list=drop_traffic \
   address-list-timeout=5m chain=prerouting dst-port=23 protocol=tcp
/ip firewall filter add action=drop chain=input src-address-list=drop_traffic
```

```
/ip firewall address-list print
Flags: X – disabled, D – dynamic
 #   LIST         ADDRESS
 0   drop_traffic 192.0.34.166
 1 D drop_traffic 1.1.1.1
 2 D drop_traffic 10.5.11.8
```

Как видно из просмотра результатов команды **print,** в списке появились две новых динамических записи **(**с префиксом D).
Это IP-адреса хостов, которые пытались подключиться к маршрутизатору через telnet. Соединения были сброшены в соответствии с прописанными выше правилами.

### Протокол L7 (layer7)

Данный протокол обеспечивает поиск указанных шаблонов в потоках ICMP/TCP/UDP.

Протокол анализирует первые 10 пакетов (либо первые 2 килобайта) соединения, чтобы определить, по какому протоколу осуществляется передача данных. Если поиск соответствий не увенчался успехом, то анализ трафика прекращается, занятая оперативная память освобождается и протокол помечается как unknown **(нераспознанный).** Имейте в виду, что большое количество соединений значительно увеличивают использование памяти и процессора. Во избежание подобной ситуации необходимо как можно больше сократить объем данных, передаваемых фильтрам протокола, оформив правила в виде регулярных выражений.
**Дополнительное требование**: протокол должен иметь возможность анализировать оба потока: и входящий, и исходящий. Для этого правила должны быть прописаны в цепочке forward**.** Если же правила прописаны в цепочке input**/prerouting**, то эти же правило необходимо продублировать и в цепочке **output/postrouting**, в противном случае полученный результат может значительно отличаться от желаемого.

Большое количество готовых L7-шаблонов может быть загружено с этой страницы: http://l7-filter.sourceforge.net/protocols

**ВНИМАНИЕ**: Если RouterOS не сможет обработать указанное в правиле регулярное выражение, то в файле журнала (под заголовком firewall) появится сообщение с описанием проблемы.

#### Описание параметров

**name** (строка) – название шаблона, используемое в правилах файервола
**regexp** (строка) – POSIX-совместимое регулярное выражение

#### Примеры использования

**Пример 1**: прописываем регулярное выражение для поиска rdp-пакетов:

```
/ip firewall layer7-protocoladd name=rdp regexp="rdpdr.*cliprdr.*rdpsnd"
```

И используем созданный шаблон в правилах файервола:

```
/ip firewall filter

# add few known protocols to reduce mem usage
add action=accept chain=forward comment="" disabled=no port=80 protocol=tcp
add action=accept chain=forward comment="" disabled=no port=443 protocol=tcp

# add l7 matcher
add action=accept chain=forward comment="" disabled=no layer7-protocol= rdp protocol=tcp
```

**Обратите внимание**: несколько правил объединены выше в одном шаблоне, что значительно сокращает объем использованной памяти

**Пример2:** поиск соединений, установленных с маршрутизатором по протоколу telnet:

```
/ip firewall layer7-protocol add comment="" name=telnet regexp="^\\xff[\\xfb-\\xfe].\\xff[\\xfb-\\xfe].\\xff[\\xfb-\\xfe]"
```

**Обратите внимание –** правила прописываются в обеих цепочках, поскольку необходимо анализировать и входящий, и исходящий трафик:

```
/ip firewall filter
add action=accept chain=input comment="" disabled=no layer7-protocol=telnet   protocol=tcp
add action=passthrough chain=output comment="" disabled=no layer7-protocol=telnet   protocol=tcp
```

**Пример3:** правило для соединений, установленных с YouTube:

```
/ip firewall layer7-protocol
add name=youtube regexp="(GET \\/videoplayback\\\?|GET \\/crossdomain\\.xml)"
```

**Обратите внимание:** соединение должно быть незашифрованным, HTTPS-соединение не может быть проанализировано.

### Обработка пакетов в зависимости от скорости соединения

Параметр **connection-rate** фильтрующих цепочек файервола (доступно в RouterOS версии 3.3 и выше) позволяет прописывать правила фильтрации в зависимости от скорости установленного соединения. После установления соединения параметру **connections-bytes** присваивается количество переданных байт (с учетом и входящего и исходящего трафика), включая размер заголовков пакетов. На основе зафиксированных в **connections-bytes** данных, параметр **connection-rate** ежесекундно высчитывает текущую скорость соединения.

Для расчета скорости анализируется только TCP- или UDP-трафик, поэтому в правиле должен быть обязательно прописан тип протокола.

В connection-rate можно прописать диапазон значений скоростей (допустимо целое значение от 0 до 4294967295), при которых будет отрабатывать установленное правило. Например, в следующем варианте будет обрабатываться трафик при скорости соединения менее 100 кб/сек:

```
/ip firewall filter
add action=accept chain=forward connection-rate=0-100k protocol=tcp
add action=accept chain=forward connection-rate=0-100k protocol=udp
```

### Пример настройки приоритезации трафика

Параметр connection-rate может быть задействован в различных ситуациях, но наиболее популярное его использование – определение т.н. «тяжелых соединений» (загрузки по FTP, P2P и т.д, где требуется передать большой объем трафика за относительно продолжительный период времени) и установка им соответствующего приоритета.

Указанный в примере метод может быть использован параллельно с другими методами приоритезации трафика.

Для начала нам необходимо определить ту границу, за которой наше соединение будет определяться как «тяжёлое». Предположим, что обычное HTTP-соединение не будет потреблять более 500 кб трафика, а VOIP не потребует скорости соединения более 200 кб/сек. (это все лишь пример, для ваших «боевых» условий подберите соответствующие значения connections-bytes и connection-rate).

В данном примере допустим, что провайдер обеспечивает пропускную способность канала не более 6 Мбит/с в обе стороны.

```
/ip firewall mangle
add chain=forward action=mark-connection connection-mark=!heavy_traffic_conn \
   new-connection-mark=all_conn
add chain=forward action=mark-connection connection-bytes=500000-0 \
   connection-mark=all_conn connection-rate=200k-100M \
   new-connection-mark=heavy_traffic_conn protocol=tcp
add chain=forward action=mark-connection connection-bytes=500000-0 \
   connection-mark=all_conn connection-rate=200k-100M \
   new-connection-mark=heavy_traffic_conn protocol=udp
add chain=forward action=mark-packet connection-mark=heavy_traffic_conn \
   new-packet-mark=heavy_traffic passthrough=no
add chain=forward action=mark-packet connection-mark=all_conn \
   new-packet-mark=other_traffic passthrough=no

/queue tree
add name=upload parent=public max-limit=6M
add name=other_upload parent=upload limit-at=4M max-limit=6M \
   packet-mark=other_traffic priority=1
add name=heavy_upload parent=upload limit-at=2M max-limit=6M \
   packet-mark=heavy_traffic priority=8

add name=download parent=local max-limit=6M
add name=other_download parent=download limit-at=4M max-limit=6M \
   packet-mark=other_traffic priority=1
add name=heavy_download parent=download limit-at=2M max-limit=6M \
   packet-mark=heavy_traffic priority=8
```

Объяснение

В таблице mangle нам необходимо разделить все соединения на две группы и затем пометить пакеты из каждой группы соответствующими метками. Логичнее всего маркировать трафик в цепочке forward.

Имейте в виду, что как только «тяжелое» соединение получит низкий приоритет и очередь уменьшит пропускную способность согласно параметру max-limit (максимально возможная скорость), скорость соединения сразу уменьшится. Значение connection-rate будет минимальным, что приведет к повышению приоритета и кратковременному увеличению трафика. Что опять же увеличит значение connection-rate и приведет к уменьшению приоритета. Чтобы избежать подобной ситуации, мы должны все появляющиеся «тяжелые» соединения сразу же помечать соответствующими метками.

Следующее правило гарантирует, что все «тяжелые» соединения останутся таковыми. Все остальные соединения помечаются дефолтной меткой **all_conn.**

```
/ip firewall mangle
add chain=forward action=mark-connection connection-mark=!heavy_traffic_conn \
   new-connection-mark=all_conn
```

Эти два правила помечают соединения как «тяжелые» в соответствии с необходимыми критериями, а именно: при превышении трафика каждого

соединения более чем на 500 кб и при скорости данного соединения более чем 200 кб/сек:

```
add chain=forward action=mark-connection connection-bytes=500000-0 \
   connection-mark=all_conn connection-rate=200k-100M \
   new-connection-mark=heavy_traffic_conn protocol=tcp
add chain=forward action=mark-connection connection-bytes=500000-0 \
   connection-mark=all_conn connection-rate=200k-100M \
   new-connection-mark=heavy_traffic_conn protocol=udp
```

Последние два правила помечают «тяжелые» и остальные соединения соответствующими метками:

```
add chain=forward action=mark-packet connection-mark=heavy_traffic_conn \
   new-packet-mark=heavy_traffic passthrough=no
add chain=forward action=mark-packet connection-mark=all_conn \
   new-packet-mark=other_traffic passthrough=no
```

Далее идет обработка помеченных пакетов в *queue tree* согласно дисциплине очереди HTB. К интерфейсу HTB «public» подключена сеть провайдера, к интерфейсу HTB «local» – локальная сеть. Если имеется более одного интерфейса local или public, то необходимо обрабатывать каждое направление трафика отдельно и помещать queue tree в родительскую очередь global-out.

```
/queue tree
add name=upload parent=public max-limit=6M
add name=other_upload parent=upload limit-at=4M max-limit=6M \
   packet-mark=other_traffic priority=1
add name=heavy_upload parent=upload limit-at=2M max-limit=6M \
   packet-mark=heavy_traffic priority=8
add name=download parent=local max-limit=6M
add name=other_download parent=download limit-at=4M max-limit=6M \
   packet-mark=other_traffic priority=1
add name=heavy_download parent=download limit-at=2M max-limit=6M \
   packet-mark=heavy_traffic priority=8
```

### Отслеживание установленных соединений

Просмотр возможен как через графический интерфейс, например, в WINBOX (IP→Firewall→Connections), так и в консоли командой:

```
/ip firewall connection print
```

**Механизм определения состояний (conntrack)**

Начиная с 6 версии RouterOS данный механизм включается автоматически, как только в правило файервола добавляется как минимум одно правило.

**Описание параметров**

**enabled** (yes | no | auto) – включение или выключение механизма определения состояний. Рекомендуется отставить данный параметр в режиме **auto.** Отключение приведет к неработоспособности следующих функций файервола:

- connection-bytes
- connection-mark
- connection-type
- connection-state
- connection-limit
- connection-rate
- layer7-protocol
- p2p
- new-connection-mark
- tarpit

Следующие параметры устанавливают таймауты для соответствующих типов IP-пакетов:

**tcp-syn-sent-timeout** (временной интервал, по умолчанию: **5 сек**)
**tcp-сек.yn-received-timeout** (временной интервал, по умолчанию: **5 сек**)
**tcp-established-timeout** (временной интервал, по умолчанию: **1 день**)
**tcp-fin-wait-timeout** (временной интервал, по умолчанию: **10 сек**)
**tcp-close-wait-timeout** (временной интервал, по умолчанию: **10 сек**)
**tcp-last-ack-timeout** (временной интервал, по умолчанию: **10 сек**)
**tcp-time-wait-timeout** (временной интервал, по умолчанию: **10 сек**)
**tcp-close-timeout** (временной интервал, по умолчанию: **10 сек**)
**udp-timeout** (временной интервал, по умолчанию: **10 сек**)
**udp-stream-timeout** (временной интервал, по умолчанию: **3 мин**)
**icmp-timeout** (временной интервал, по умолчанию: **10 сек**)

**generic-timeout** (временной интервал, по умолчанию: **10 мин**) **–** таймаут для остальных соединений

**Параметры, доступные только для чтения**

**max-entries** (целое значение) – максимально возможное количество параметров, отслеживаемых механизмом определения состояний. Зависит от объемов установленной и доступной оперативной памяти.

**total-entries** (целое значение) – текущее количество параметров, отслеживаемых механизмом определения состояний.

# Службы, протоколы и порты

Уровень меню: **/ip service**

В данном разделе рассматриваются протоколы и порты, используемые различными сервисами RoterOS, рассказывается о необходимых действиях, позволяющих запретить/разрешить доступ к тому или иному сервису.

### Описание параметров

**address** (IP-адрес/маска | IPv6/0..128;) – список IP-адресов, с которых разрешен доступ к службам

**certificate** (имя | по умолчанию:**none**) – имя сертификата, используемого службой (для служб, требующих наличия сертификата – www-ssl, api-ssl)

**name** (название) **–** название службы

**port** (целое значение: 1..65535) – порт, который слушает служба

Пример:

Настроим доступ по протоколу telnet только с указанного диапазона адресов IPv6:

```
/ip service set telnet address=10.5.101.0/24,2001:db8:fade::/64

 /ip service print
Flags: X – disabled, I – invalid
 #   NAME      PORT ADDRESS                          CERTIFICATE
 0   telnet     23 10.5.101.0/24
                2001:db8:fade::/64
 1   ftp        21
 2   www        80
 3   ssh        22
 4 X www-ssl    443                                  none
 5 X api        8728
 6   winbox     8291
 7   api-ssl    8729                                 none
```

### Список служб

В таблице ниже представлены протоколы и порты, используемые службами RouterOS.

| Порт/Протокол | Описание |
|---|---|
| 20/tcp | FTP (поток данных) |
| 21/tcp | FTP (поток управления) |
| 22/tcp | SSH |
| 23/tcp | Telnet |
| 53/tcp | DNS |
| 53/udp | DNS |
| 67/udp | Bootstrap или DHCP-сервер |
| 68/udp | Bootstrap или DHCP-клиент |
| 80/tcp | HTTP |
| 123/udp | NTP |
| 161/udp | SNMP |
| 179/tcp | BGP |
| 443/tcp | Secure HTTP |
| 500/udp | IKE |
| 521/udp | RIP |
| 521/udp | RIP |
| 646/tcp | LDP (transport session) |
| 646/udp | LDP (hello) |
| 1080/tcp | SOCKS proxy |
| 1701/udp | L2TP |
| 1698/udp<br>1699/udp | Туннели RSVP TE |
| 1723/tcp | Туннели PPTP |
| 1900/udp | Universal Plug and Play (uPnP) |
| 2828/tcp | Universal Plug and Play (uPnP) |
| 1966/udp | MME originator message traffic |
| 1966/tcp | MME gateway |
| 2000/tcp | Bandwidth-test server |
| 5246,5247/udp | CAPsMan |
| 5678/udp | MikroTik™ Neighbor Discovery Protocol |
| 6343/tcp | Стандартный порт OpenFlow |
| 8080/tcp | HTTP Web proxy |
| 8728/tcp | Application Programmable Interface (API) |
| 8729/tcp | API-SSL |

| | |
|---|---|
| 8291/tcp | Winbox |
| 20561/udp | MAC winbox |
| /1 | ICMP – Internet Control Message Protocol |
| /2 | Multicast \| IGMP |
| /4 | IP IP (инкапсуляция) |
| /41 | IPv6 (инкапсуляция) |
| /46 | Туннели RSVP TE |
| /47 | GRE – General Routing Encapsulation (для туннелей PPTP и EoIP) |
| /50 | ESP – Encapsulating Security Payload для IPv4 |
| /51 | AH – Authentication Header для IPv4 |
| /89 | OSPF |
| /103 | Multicast \| PIM |
| /112 | VRRP |

# SOCKS (прокси-сервер)

### Спецификация

Требуемые пакеты: **system**
Уровень подменю: **/ip socks**
Стандарты и технологии: SOCKS version 4
Аппаратное обеспечение: не принципиально

### Описание

SOCKS – это прокси-сервер, позволяющий проходить TCP-пакетам приложений прозрачно через файерволл, даже если он блокирует пакеты. SOCKS-протокол не зависит от конкретных протоколов уровня приложений (7-го уровня модели OSI) и оперирует на уровне TCP-соединений (4-й уровень модели OSI), таким образом он может быть использован многими службами, такими как WWW, FTP, TELNET и др.

Протокол работает следующим образом: клиентское приложение подключается к SOCKS-прокси серверу, который в свою очередь просматривает свой список доступа и проверяет, разрешен ли доступ клиента к удаленному серверу или нет. Если разрешен, то прокси-сервер пересылает пакет серверу и создает соединение между серверным и клиентским приложениями.

### Примечание

Поддерживается только 4 версия протокола (без поддержки аутентификации). Вам необходимо защитить SOCKS-прокси от внешнего доступа, используя списки доступа и/или файервол. Нарушение безопасности прокси-сервера может плохо отразиться на безопасности вашей сети, и может предоставить спамерам возможность рассылки почтовых сообщений через маршрутизатор.

### Описание параметров

**connection-idle-timeout** (временной интервал, по умолчанию: 2 мин) – время, по истечении которого происходит сброс соединения.
**enabled** (yes | no; по умолчанию: **no**) – включение/отключение SOCKS-прокси
**max-connections** (целое значение: 1..500; по умолчанию: 200) – максимальное количество одновременных подключений
**port** (целое значение: 1..65535; по умолчанию: 1080) – TCP-порт, на котором SOCKS-сервер слушает запросы.

### Список доступа

Уровень подменю: **/ip socks access**

В списке доступа SOCKS вы можете добавлять правила, контролирующие доступ к SOCKS-серверу. Этот список аналогичен спискам доступа файервола.

#### Описание параметров

**action** (allow | deny; по умолчанию: **allow**) – действие, которое будет выполняться при выполнении правила

**allow –** принять пакеты, соответствующие правилу и отправить их для дальнейшей обработки

**deny –** запретить доступ пакетам, соответствующих правилу

**dst-address** (IP-адрес/маска:порт) – адрес назначения пакета

**dst-port** (порт) – порт назначения пакета

**src-address** (IP-адрес/маска:порт) – исходящий адрес пакета

**src -port** (порт) – порт источника пакета

### Просмотр активных соединений

Уровень подменю: **/ip socks connections**

Список активных соединений показывает все установленные TCP-соединения, проходящие через SOCKS-сервер.

#### Описание параметров

**dst-address** (только для чтения: IP-адрес) – IP-адрес назначения (адрес серверного приложения)

**rx** (только для чтения: целое значение) – количество полученных байт

**src-address** (только для чтения: IP-адрес) – IP-адрес источника (адрес клиентского приложения)

**tx** (только для чтения: целое значение) – количество отправленных байт

**type** (только для чтения: in | out | unknown) – тип соединения

- in – входящее соединение
- out – исходящее соединение
- unknown – инициализация соединения

Для просмотра текущих TCP соединений выполните следующее:

```
/ip socks connections> print
# SRC-ADDRESS     DST-ADDRESS        TX      RX
0 192.168.0.2:3242  159.148.147.196:80  4847    2880
1 192.168.0.2:3243  159.148.147.196:80  3408    2127
2 192.168.0.2:3246  159.148.95.16:80    10172   25207
3 192.168.0.2:3248  194.8.18.26:80      474     1629
4 192.168.0.2:3249  159.148.95.16:80    6477    18695
5 192.168.0.2:3250  159.148.95.16:80    4137    27568
6 192.168.0.2:3251  159.148.95.16:80    1712    14296
7 192.168.0.2:3258  80.91.34.241:80     314     208
8 192.168.0.2:3259  80.91.34.241:80     934     524
9 192.168.0.2:3260  80.91.34.241:80     930     524
10 192.168.0.2:3261 80.91.34.241:80     312     158
11 192.168.0.2:3262 80.91.34.241:80     312     158
```

### Пример создания FTP-соединения через SOCKS-сервер

Допустим, есть сеть 192.168.0.0/24, доступная через маршрутизатор с включенным маскарадингом и с IP-адресами: внешним – 10.1.0.104/24 и локальным – 192.168.0.1/24. Где-нибудь во внешней сети расположен FTP-сервер с IP-адресом 10.5.8.8. Мы хотим разрешить доступ к этому FTP-серверу одному из клиентов локальной сети с адресом 192.168.0.2/24.

Для начала проверим настройку маскарадинга для локальной сети:

```
/ip firewall nat> print
Flags: X – disabled, I – invalid, D – dynamic
 0   chain=srcnat action=masquerade src-address=192.168.0.0/24
ip firewall nat>
```

Проверим, закрыт ли доступ из локальной сети ко внешнему FTP:

```
/ip firewall filter> print
Flags: X – disabled, I – invalid, D – dynamic
 0   chain=forward action=drop src-address=192.168.0.0/24 dst-port=21 protocol=tcp
```

Теперь включаем SOCKS-сервер:

```
/ip socks> set enabled=yes

/ip socks> print
                enabled: yes
                   port: 1080
  connection-idle-timeout: 2m
        max-connections: 200
```

Добавляем в список доступа клиента с IP-адресом 192.168.0.2/32, разрешая для него передачу данных от FTP-сервера (открываем удаленные порты с 1024 по 65535 для любых IP-адресов) и запрещая все остальные соединения:

```
/ip socks access> add src-address=192.168.0.2 dst-port=21 action=allow
/ip socks access> add dst-port=1024-65535 action=allow
/ip socks access> add action=deny

/ip socks access> print
Flags: X – disabled
 0   src-address=192.168.0.2 dst-port=21 action=allow
 1   dst-port=1024-65535 action=allow
 2   action=deny
```

На этом настройку SOCKS-сервера для поставленной задачи можно считать завершённой. Для просмотра активных соединений и полученных/отправленных данных выполните следующее:

```
/ ip socks connections> print
# SRC-ADDRESS    DST-ADDRESS      TX        RX
0 192.168.0.2:1238   10.5.8.8:21      1163      4625
1 192.168.0.2:1258   10.5.8.8:3423    0         3231744
```

**Обратите внимание:** в случае использования SOCKS-прокси сервера необходимо на FTP-клиенте прописать IP-адрес и порт маршрутизатора. В данном случае это IP-адрес 192.168.0.1 (локальный адрес маршрутизатора) и порт 1080.

OSPF

**Спецификация**

Требуемые пакеты: **routing**
Уровень подменю: **/routing ospf**
Стандарты и технологии: OSPF
Аппаратное обеспечение: не принципиально

**Описание**

**Open Shortest Path First** (первый доступный кратчайший путь) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала. Для построения маршрута протокол использует алгоритм проверки состояния и вычисляет кратчайший путь ко всем доступным конечным точкам. Для вычисления кратчайшего пути используется алгоритм Дейкстры. OSPF распространяет информацию о маршрутах между маршрутизаторами, принадлежащими общим автономным системам (АС). Под АС подразумевается группа маршрутизаторов, обменивающихся маршрутной информацией через общий протокол маршрутизации.

Уровень подменю:**/routing ospf**

**Описание параметров**

**distribute-default** (never | if-installed-as-type-1 | if-installed-as-type-2 | always-as-type-1 | always-as-type-2; по умолчанию: **never**) – определяет распределение маршрута по умолчанию. Может быть использован для пограничного маршрутизатора (area border router, ABR) или для пограничного маршрутизатора автономной системы (AS boundary router, ASBR).

- **never –** не рассылать собственный маршрут по умолчанию другим маршрутизаторам
- **if-installed-as-type-1 –** рассылать маршрут по умолчанию с типом метрики 1 только если он существует (статический маршрут по умолчанию, или маршрут, добавленный через DHCP, PPP т.д.)
- **if-installed-as-type-2 –** рассылать маршрут по умолчанию с типом метрики 2 только если он существует (статический маршрут или маршрут, добавленный через DHCP, PPP т.д.
- **always-as-type-1 –** всегда рассылать маршрут по умолчанию с типом метрики 1

- **always-as-type-2 –** всегда рассылать маршрут по умолчанию с типом метрики 2

**domain-id** (Hex|адрес;) – параметр, относящийся к MPLS. Определяет домен экземпляра OSPF. По умолчанию используется нулевое значение, как это описано в стандарте RFC 4577.

**domain-tag** (целое значение: 0..4294967295;) – если параметр определен, то он используется при перераспределении маршрута (добавляется в виде тега при создании маршрутизатором объявления о состоянии канала (LSA), а также при расчете маршрута (все объявления LSA, содержащие подобный тег, игнорируются all external LSAs having this route tag are ignored). Параметр необходим для совместимости с устаревшим Cisco. По умолчанию данный параметр не определен и не используется.

***in-filter*** (строка) – название цепочки фильтра для входящих пакетов указанной подсети.

**metric-bgp** (целое число | auto; по умолчанию: **20**) – указать стоимость маршрута, полученного от протокола граничного шлюза (BGP). Если указать параметр **auto**, то будет использован атрибут MED (MULTIEXIT DISCRIMINATOR) от BGP, если атрибут MED не установлен, то будет использовано значение по умолчанию – 20

**metric-connected** (целое число; по умолчанию: **20**) – указать стоимость маршрута к ближайшим подключенным сетям (сети находящейся прямо за маршрутизатором)

**metric-default** (целое число; по умолчанию: **1**) – указать стоимость маршрута по умолчанию

**metric-other-ospf** (целое число | auto; по умолчанию: **20**) – указать стоимость маршрута, полученного из другого экземпляра OSPF. Если указать параметр **auto**, то будет указана стоимость, полученная от предыдущего экземпляра OSPF.

**metric-rip** (целое число; по умолчанию: **20**) – указать стоимость маршрута, полученного через протокол RIP.

**metric-static** (целое число; по умолчанию: **20**) – указать стоимость статических маршрутов

**mpls-te-area** (строка) – зона используется для управления трафиком (MPLS Traffic Egineering). В данной зоне создаются специальные типы LSA – TE Opaque LSA. Только один экземпляр OSPF может иметь настроенную зону mpls-te-area.

**mpls-te-router-id** (ip;) – loopback-интерфейс, с которого передается IP-адрес, используемый в качестве Router-ID в MPLS TE Opaque LSA

**out-filter** (строка) – название цепочки фильтра для исходящих пакетов указанной подсети.

**redistribute-bgp** (as-type-1 | as-type-2 | no; по умолчанию: **no**) – передавать информацию о маршрутах по протоколу BGP

**redistribute-connected** (as-type-1 | as-type-2 | no; по умолчанию: **no**) – передавать информацию о всех активных маршрутах, т.е. маршруты к непосредственно доступным сетям

**redistribute-ospf** (as-type-1 | as-type-2 | no; по умолчанию: **no**) – передавать информацию о маршрутах через другие экземпляры протокола OSPF.

**redistribute-rip** (as-type-1 | as-type-2 | no; по умолчанию: **no**) – передавать информацию о маршрутах по протоколу RIP.

**redistribute-static** (as-type-1 | as-type-2 | no; по умолчанию: **no**) – раздать заново информацию о всех статических маршрутах, добавленных в базу данных маршрутизации

**router-id** (IP-адрес; по умолчанию: **0.0.0.0**) – идентификатор маршрутизатора. Если не указан, OSPF использует в качестве такового наименьший IP-адрес на активном интерфейсе

**routing-table** (название таблицы) – название таблицы маршрутизации, с которой работает данный экземпляр OSPF

**use-dn** (yes | no;) – использование или неиспользование Down-бита (DN-bit). Используется в некоторых сценариях CE PE (Customer Edge Router – Provider Edge Router) для добавления внутризонового маршрута в виртуальный маршрутизатор (VRF). Если параметр не установлен, то Down-бит используется согласно спецификации RFC 4576. Доступно в RouterOS с версии 6RC12.

### Типы метрик

Протокол OSPF поддерживает два типа метрик:

- **type1** – метрика равна сумме стоимости внутреннего маршрута OSPF и стоимости внешнего маршрута
- **type2 –** метрика равна стоимости внешнего маршрута

### Просмотр статуса

Для просмотра статуса текущего экземпляра OSPF используется команда

```
/routing ospf monitor
```

Для просмотра статуса всех экземпляров:

```
/routing ospf   instance print status
```

### OSPF-зоны

Уровень подменю: **/routing ospf area**

OSPF работает с группами маршрутизаторов, называемыми **зонами**. Рассылая объявления внутри одной OSPF-зоны, все маршрутизаторы строят идентичную базу данных состояния каналов маршрутизатора. Это означает, что каждая зона имеет свою собственную базу данных состояний, на основе которой каждый маршрутизатор строит кратчайший путь к каждому известному пункту назначения с собой в качестве корня— так называемое дерево кратчайших путей.

Структура каждой конкретной зоны не доступна из других зон. Подобная изоляция позволяет протоколу быть более масштабируемым при использовании большого количества зон, значительно сократить нагрузку на процессор при работе с таблицей маршрутизации и уменьшить объем маршрутизируемого трафика.

Тем не менее, подобная мультизональность накладывает и определённые ограничения: не рекомендуется делить сеть на зоны с менее чем 50 маршрутизаторами в каждой. Максимальное количество маршрутизаторов в каждой зоне зависит от производительности процессора, работающего с таблицей маршрутизации.

#### Описание параметров

**area-id** (IP-адрес; по умолчанию: 0.0.0.0) – идентификатор зоны OSPF. Если маршрутизатор обслуживает сети, находящиеся более чем в одной зоне, то как минимум один интерфейс такого маршрутизатора обязательно должен иметь идентификатор магистральной (нулевой) зоны – 0.0.0.0. Такой маршрутизатор называется **магистральным (backbone router)** и формирует ядро сети. Магистральный маршрутизатор связан со всеми пограничными маршрутизаторами **(area border router, ABR)** и отвечает за маршрутизацию трафика между немагистральными зонами. Магистральная зона должна быть непрерывной, т.е. не должна содержать нерабочих сегментов. Однако пограничные маршрутизаторы не обязательно должны напрямую соединяться с магистральным – соединение с магистральной зоной может быть установлено и с помощью виртуальных каналов.

**default-cost** (целое значение; по умолчанию: **1**) – стоимость маршрута по умолчанию для маршрутизаторов тупиковой зоны (stub area). Если маршрутизаторам из тупиковой зоны необходимо передавать информацию за границу автономной системы, то они используют данный маршрут.

**name** (название; по умолчанию: "") – название зоны OSPF.

**translator-role** (translate-always | translate-candidate | translate-never; по умолчанию: translate-candidate) – параметр указывает, какой из пограничных маршрутизаторов (ABR) преобразует type 7 LSA в type 5 LSA. Применимо только для зон NSSA.

- translate-always – указанный маршрутизатор всегда будет использоваться для преобразования
- translate-never – указанный маршрутизатор никогда не будет использоваться для преобразования
- translate-candidate – указанный маршрутизатор может быть автоматически выбран для преобразования протоколом OSPF

**type** (default | nssa | stub; по умолчанию: **default**) – тип зоны

### Просмотр статуса

Уровень подменю: **/routing ospf area print status**

Просмотр дополнительных параметров маршрутизации

**interfaces** (целое число) – количество интерфейсов в зоне

**active-interfaces** (целое число) – количество активных интерфейсов в зоне

**neighbors** (целое число) – количество соседей в зоне (маршрутизаторы, имеющие интерфейсы в общей сети)

**adjacent-neighbors** (целое число) – количество соседей в зоне, находящихся в состоянии смежности

### Межзональное суммирование

Уровень подменю: **/routing ospf area range**

Суммирование используется для объединения информации о маршрутизации пограничных маршрутизаторов. Обычно пограничные маршрутизаторы создают объявление о состоянии канала (LSA) для **каждого** маршрута в зоне и передают информацию соседним маршрутизаторам (состояние смежности). Суммирование позволяет создавать одно общее объявление LSA для нескольких маршрутов с дальнейшей отправкой его соседним маршрутизаторам

**advertise** (yes | no; По умолчанию: yes) – использовать суммирование и передавать общее объявление в смежные зоны

**area** (строка) – название OSPF-зоны, в которой применяется суммирование

**cost** (целое значение | default; По умолчанию: default) – стоимость общего объявления LSA при суммировании

- default – использовать максимальную стоимость всех используемых маршрутов

**range** (подсеть) – адрес подсети

**Настройка параметров сети**

Уровень подменю: **/routing ospf network**

Для запуска протокола OSPF, необходимо определить:

- сети, где протокол будет запущен
- зоны для каждой из этих сетей

**Описание параметров**

**area** (строка; по умолчанию: **backbone**) – зона OSPF, связанная с неким диапазоном адресов.

**network** (IP-подсеть) – подсеть, связанная с зоной. Протокол OSPF будет включён на всех сетевых интерфейсах, на которых прописан хотя бы один адрес из указанной подсети.

**Настройка дополнительных параметров**

Уровень подменю: **/routing ospf interface**

Настройка дополнительных параметров OSPF.

**Описание параметров**

**authentication** (none; simple; md5; по умолчанию: none) – метод аутентификации в сообщениях протокола OSPF

- **none** – не использовать аутентификацию
- **simple** – открытая текстовая аутентификация
- **md5** – md5-аутентификация

**authentication-key** (строка; по умолчанию: **""**) – ключ аутентификации, используется при md5- аутентификации

**authentication-key-id** (целое значение; по умолчанию: **""**) – – идентификатор ключа, число от 1 до 255, используется только при md5- аутентификации

**cost** (целое значение: 1..65535; по умолчанию: **1**) – стоимость интерфейса в виде метрики состояния соединения

**dead-interval** (время; по умолчанию: **40 сек.**) – интервал, по истечении которого соседний маршрутизатор классифицируется как «мертвый». Значение рассылается в пакете hello. Это значение должно быть одинаково для всех маршрутизаторов сети, иначе не будет установлено состояние смежности

**hello-interval** (время; по умолчанию **10 сек.**) – интервал между пакетами hello, которые маршрутизатор рассылает с сетевого интерфейса. Чем меньше интервал, тем быстрее будут обнаружены изменения в топологии сети, но тем больше увеличится трафик. Это значение должно быть одинаково для всех маршрутизаторов сети, иначе не будет установлено состояние смежности

**interface** (строка | all; по умолчанию: **all**) – интерфейс, на котором будет запущен протокол OSPF

- **all** – протокол включен на всех сетевых интерфейсах

**network-type** (broadcast | nbma | point-to-point | ptmp*;* по умолчанию: **broadcast**) – тип сети интерфейса. Обратите внимание: если интерфейс не настроен, то тип сети равен 'point-to-point' для интерфейса типа «точка-точка» и , 'broadcast' для других типов интерфейсов.

- **broadcast** – широковещательный тип сети
- **nbma** – нешироковещательная сеть со множественным доступом. Пакет направляется на сетевой адрес соседнего маршрутизатора. Необходимо дополнительное указание адреса соседнего маршрутизатора
- **point-to-point** – точка-точка, сеть, состоящая из двух устройств Дополнительное указание адреса соседнего маршрутизатора не требуется
- **ptmp** – точка-многоточка. Настраивается проще, чем nbma, поскольку не требуется вручную указывать данные соседнего маршрутизатора. Это самый надежный и наиболее подходящий тип сети для беспроводного вещания

**passive** (yes | no; по умолчанию: **no**) **–** не получать и не отправлять OSPF-трафик при включённом параметре

**priority** (целое значение: 0..255; по умолчанию:**1**) – приоритет маршрутизатора. Параметр определяет выделенный маршрутизатор (designated router, DR) сети. Маршрутизатор с наивысшим значением priority имеет максимальный приоритет. Нулевой приоритет означает, что маршрутизатор не может использоваться в качестве выделенного (DR) или резервного выделенного (BDR) маршрутизатора

**retransmit-interval** (время; по умолчанию: **5сек.**) – интервал, по истечении которого повторно рассылается информация о состоянии канала (пакеты LSA). Когда маршрутизаторы обмениваются LSA-пакетами с соседними маршрутизаторами, обратно приходят подтверждения о получении пакета (Link State Update). Если в течении

указанного интервала подтверждение Link State Update не получено, то LSA-пакеты рассылаются повторно.

**transmit-delay** (время; по умолчанию: **1сек**) – интервал, в течении которого предполагается получить подтверждение о получении пакета (Link State Update).

**Просмотр статуса**

Уровень подменю: **/routing ospf print status**

Просмотр дополнительных параметров интерфейса

**ip-address** (IP-адрес) – IP-адрес интерфейса

**state** (backup | designated-router | point-to-point | passive) **–** текущий статус интерфейса

**instance** (название) **–** название экземпляра OSPF, используемого на данном интерфейсе

**area** (название) – название зоны, к которой принадлежит данный интерфейс

**neighbors** (целое число; – количество соседей в зоне (маршрутизаторы, имеющие интерфейсы в общей сети)

**adjacent-neighbors** (целое число; – количество соседей в зоне, находящихся в состоянии смежности

**designated-router** (IP-адрес) – IP-адрес выделенного маршрутизатора (DR)

**backup-designated-router** (IP-адрес) **–** IP-адрес резервного выделенного маршрутизатора (BDR)

**Смежность в сетях NBMA**

Уровень подменю:**/routing ospf nbma-neighbor**

Ручная настройка смежности в нешироковещательной сети со множественным доступом. Производится только в том случае, если тип сети (network-type) указан как nbma.

**Описание параметров**

**address** (IP-адрес) – IP-адрес соседнего маршрутизатора

**poll-interval** (время; по умолчанию: 2мин.) – интервал, определяющий как часто будут рассылаться hello-пакеты соседним маршрутизаторам, находящимся в состоянии «down»

**priority** (целое значение: 0..255; по умолчанию: 0) - предполагаемый приоритет соседних маршрутизаторов, находящихся в состоянии «down»

### Виртуальные соединения

Уровень подменю: **/routing ospf virtual-link**

#### Описание

В RFC к OSPF описано, что магистральная зона должна быть единой, не разделённой на отдельные зоны. Тем не менее, можно организовать различные зоны, разделённые между собой транзитными зонами, в виде единой магистральной зоны. Для этого должны быть настроены т.н. виртуальные каналы (ВК). ВК может настроен между пограничными (ABR) маршрутизаторами, один из которых должен входить в магистральную зону. OSPF воспринимает два маршрутизатора, соединённых виртуальным каналом так, как будто они работают в сети типа точка-точка

#### Описание параметров

**authentication** (none; simple; md5; по умолчанию: none) – метод аутентификации в сообщениях протокола OSPF

- **none** – не использовать аутентификацию
- **simple** – открытая текстовая аутентификация
- **md5** – md5-аутентификация

**authentication-key** (строка; по умолчанию: **""**) – ключ аутентификации, используется при md5- аутентификации

**authentication-key-id** (целое значение; по умолчанию: **""**) – – идентификатор ключа, число от 1 до 255, используется только при md5- аутентификации

**neighbor-id** (IP-адрес; по умолчанию: **0.0.0.0**) – идентификатор соседнего маршрутизатора.

**transit-area** (название; по умолчанию: **(unknown)**) – транзитная область, через которую связываются два пограничных маршрутизатора

**Примечание**

Виртуальные каналы должны быть настроены на обеих маршрутизаторах.

Виртуальные каналы не могут быть настроены через тупиковую зону.

**Пример**

Для создания виртуального канала между пограничными маршрутизаторами 10.0.0.200 и 10.0.0.201 через транзитную зону область *trans*, выполним следующее:

На маршрутизаторе 10.0.0.200:

```
/routing ospf virtual-link add transit-area=trans neighbor-id=10.0.0.201
```

На маршрутизаторе 10.0.0.201:

```
/routing ospf virtual-link add transit-area=trans neighbor-id=10.0.0.200
```

### Объявление о состоянии канала (LSA)

Уровень подменю: **/routing ospf lsa**

**Описание параметров (доступны только для чтения)**

**area** (строка) – название зоны

**type** (строка) – тип сети

**id** (IP-адрес) – идентификатор LSA-записи

**originator** (IP-адрес) – IP-адрес устройства, создавшего LSA-запись

**sequence-number** (строка) – сколько раз LSA-запись была обновлена

**age** (целое значение) – возраст текущей LSA-записи в секундах

### Перечень соседних маршрутизаторов

Уровень подменю: **/routing ospf neighbor**

**Описание**

Данное подменю отображает список OSPF-соседей, то есть маршрутизаторов, смежных с текущим маршрутизатором, и обменивающихся с ним статистикой.

**Описание параметров (доступны только для чтения)**

**router-id** (IP-адрес) – ID соседнего маршрутизатора

**address** (IP адрес) – IP-адрес соседнего маршрутизатора

**interface** (строка) – интерфейс текущего маршрутизатора, к которому подключен соседний маршрутизатор

**priority** (целое значение) – приоритет, установленный на соседнем маршрутизаторе

**dr-address** (IP-адрес) – IP-адрес выделенного (DR) маршрутизатора

**backup-dr-address** (IP-адрес) – IP-адрес резервного выделенного (BDR) маршрутизатора

**state** (read-only: Down | Attempt | Init | 2-Way | ExStart | Exchange | Loading | Full) – состояние соединения маршрутизатора:

- **Down** – hello-пакеты от соседнего маршрутизатора не получены
- **Attempt** – применимо только к облакам NBMA. Маршрутизатор недавно рассылал hello-пакеты, но ответа получено не было
- **Init** – hello-пакет получен от соседа, но двухстороннее соединение еще не установлено
- **2-Way** – Двухстороннее соединение установлено, в течение этого состояния происходят выборы DR и BDR, маршрутизаторы устанавливают смежность (соседство) с DR и BDR, устанавливается соединение типа точка-точка или виртуальный канал
- **ExStart** – маршрутизаторы пытаются установить последовательность ID, которая будет использована при обмене пакетами. Маршрутизатор с самым большим значением ID назначается главным и начинает обмен пакетами
- **Exchange** – маршрутизаторы обмениваются пакетами с описанием базы данных (DD)
- **Loading** – пакет запроса состояния канала (Link State Request) отправляется соседям, чтобы запросить новые LSA, обнаруженные в течение состояния Exchange
- **Full** – состояние смежности установлены, базы данных состояний соединений полностью синхронизированы. Маршрутизаторы отправляют информацию о состоянии канала (LSA) только DR- и BDR-маршрутизаторам (исключение – соединения типа точка-точка)

**state-changes** (целое значение) – суммарное количество изменений состояний OSPF при установке смежности

**adjacency** (время) – время, за которое было установлено состояние смежности

**Список пограничных маршрутизаторов**

Уровень подменю: **/routing ospf ospf-router**

**Описание параметров (доступны только для чтения)**

**area** (строка) – зона, к которой принадлежит маршрутизатор

**router-id** (IP-адрес) – ID маршрутизатора

**state** (строка) - состояние соединения маршрутизатора

**gateway** (IP-адрес) – адрес шлюза

**cost** (целое значение) – стоимость интерфейса

**Просмотр параметров маршрута**

Уровень подменю: **/routing ospf route**

**Описание параметров (доступны только для чтения)**

**instance** (строка) – экземпляр OSPF, к которому принадлежит маршрутизатор

**dst-address** (IP-сеть) – адрес подсети получателя

**state** (intra-area | inter-area | ext-1 | ext-2 | imported-ext-1 | imported-ext-2) – статус, отображающий происхождение маршрута

**gateway** (IP-адрес) – адрес шлюза

**interface** (строка) – используемый интерфейс

**cost** (целое значение) – стоимость маршрута

**area** (external | backbone | <other area>) – зона OSPF, к которой принадлежит маршрут

# Прозрачный мост

Компьютерные сети, работающие по протоколу Ethernet (Ethernet, Ethernet over IP, беспроводные роутеры, работающие в режимах *ap-bridge* или *bridge*, WDS, VLAN), могут быть скоммутированы между собой при помощи технологии сетевых мостов, работающих на 2 уровне модели OSI. Сетевые мосты, и, в частности, прозрачные мосты (объединяющие сети с едиными протоколами канального и физического уровней), позволяют объединять устройства, работающие в отдельных сетях так, как будто они все работают в единой локальной сети. Прозрачные мосты (ПМ) можно сравнить с «виртуальным проводом»: хосты, в него входящие, никак не будут отображаться при трассировке маршрута и для внешних утилит не будет никаких

различий между хостами, работающими в различных LAN, если эти сети объединены прозрачными мостами, хотя задержки с скорость передачи в таких сетях могут варьироваться, в зависимости от того, как именно эти сети скоммутированы между собой.

В сетях со сложной топологией могут появляться (не всегда преднамеренно) т.н. сетевые петли, мешающие нормальному функционированию сети вследствие лавинообразного увеличения количества пакетов. ПМ поддерживает специальные алгоритмы, предотвращающие появление подобных петель, в частности алгоритм STP (Spanning Tree Protocol – канальный протокол остовного дерева) и его более современный аналог RSTP (Rapid STP). При работе подобного алгоритма все возможные избыточные соединения блокируются, но при разрыве связи могут быть вновь задействованы для возобновления работы сети. Коммутаторы периодически обмениваются специальными сообщениями BPDU (Bridge Protocol Data Unit) для поддержания информации о топологии сети в актуальном состоянии. Протокол (R)STP выбирает корневое устройство (корневой мост), основываясь на BPDU с минимальным значением Bridge ID, которое и будет отвечать за топологию сети, открывая и закрывая порты на других коммутаторах.

Уровень меню: **/ interface bridge**

Для объединения нескольких сетей в единый прозрачный мост необходимо:

- Создать виртуальный интерфейс – прозрачный мост (**ПМ**)
- Пометить желаемые интерфейсы в качестве портов созданного ПМ, при этом всем помеченным интерфейсам назначается общий мак-адрес (из всех адресов автоматически выбирается наименьший)

**Описание параметров**

**admin-mac** (MAC-адрес; По умолчанию: ) статический MAC-адрес ПМ (только если *auto-mac=no*)

**ageing-time** (время; По умолчанию: **00:05:00**) – в течении какого времени информация будет храниться в базе данных ПМ

**arp** (disabled | enabled | proxy-arp | reply-only; По умолчанию: **enabled**) – параметры протокола ARP:

- disabled – интерфейс не использует ARP
- enabled – интерфейс использует ARP
- proxy-arp – интерфейс использует проксирование запросов ARP
- reply-only – интерфейс будет отвечать только на запросы, которые соответствуют только статической комбинации IP-адрес/MAC-адрес из таблицы arp.

Соответственно, для установления связи в таблице должны присутствовать статические значения.

**auto-mac** (yes | no; По умолчанию: **yes**) – автоматическое присваивание наименьшего мас-адреса в качестве мас-адреса ПМ

**forward-delay** (время; По умолчанию: **00:00:15**) – пауза, необходимая для инициализации интерфейса ПМ (при включении роутера или поднятии интерфейса)

**l2mtu** (целое значение; только для чтения) – максимальный размер пакета (**БЕЗ** учета размера мас-заголовка), который может быть передан интерфейсом

**max-message-age** (время; По умолчанию: **00:00:20**) – сколько времени хранить сообщение «Hello», получаемое от других ПМ

**mtu** (целое значение; По умолчанию: **1500**) – максимальный размер пакета, передаваемый интерфейсом

**name** (текст; По умолчанию: **bridgeN, где N – порядковый номер**) – название интерфейса ПМ

**priority** (целое значение: 0..65535 или шестнадцатеричное значение 0x0000-0xffff; По умолчанию: **32768 / 0x8000**) - приоритет протокола STP для интерфейса ПМ. Мост с наименьшим значением Bridge ID становится корневым мостом. Bridge ID состоит из двух частей: приоритета и мас-адреса интерфейса моста. При сравнении двух Bridge ID сначала сравниваются значения приоритета и если приоритеты совпадают, то сравниваются мас-адреса.

**protocol-mode** (none | rstp | stp; По умолчанию: **rstp**) – выбор протокола для устранения возможных петель в топологии сети. Протокол RSTP как более современный обеспечивает меньшее время сходимости при изменении топологии сети и более высокую устойчивость.

**transmit-hold-count** (целое значение: 1..10; По умолчанию: **6**) – установка максимального значения счетчика задержки передачи на порту

**Пример создания/включения интерфейса прозрачного моста**

```
/interface bridge> add
/interface bridge> print
Flags: X – disabled, R – running
 0  R name="bridge1" mtu=1500 l2mtu=65535 arp=enabled
     mac-address=00:00:00:00:00:00 protocol-mode=none priority=0x8000
     auto-mac=yes admin-mac=00:00:00:00:00:00 max-message-age=20s
     forward-delay=15s transmit-hold-count=6 ageing-time=5m
```

### Настройка основных параметров прозрачного моста

Уровень меню: **/ interface bridge settings**

#### Описание параметров

**allow-fast-path** (yes | no*;* По умолчанию: **yes**) – разрешает пересылку пакетов без их дополнительной обработки. Включенный параметр значительно увеличивает скорость пересылки пакетов рассчитанные

**use-ip-firewall** (yes | no*;* По умолчанию: **no**) – дополнительная обработка передаваемых в ПМ пакетов в цепочках файервола. Параметр не обрабатывает маршрутизируемый трафик.

**use-ip-firewall-for-pppoe** (yes | no; По умолчанию: **no**) - дополнительная обработка незашифрованного трафика PPPoE в цепочках файервола (параметр *use-ip-firewall* должен быть включен)

**use-ip-firewall-for-vlan** (yes | no*;* По умолчанию: **no**) дополнительная обработка трафика VLAN в цепочках файервола (параметр *use-ip-firewall* должен быть включен)

### Настройка параметров портов прозрачного моста

Уровень меню: **/ interface bridge port**

#### Описание параметров

**auto-isolate** (yes | no; По умолчанию:**no**) – предотвращение блокирования порта протоколом STP при его ошибочном переводе в состояние пересылки

**bridge** (название; По умолчанию: **none**) – помещение соответствующего интерфейса в прозрачный мост

**edge** (auto | no | no-discover | yes | yes-discover; По умолчанию: **auto**) – пометить порт как пограничный. Пограничные порты подсоединены к LAN, не содержащей других прозрачных портов. Если порт настроен как yes-discover, и на пограничный порт приходит пакет BPDU – порт перестает считаться пограничным.

**external-fdb** (auto | no | yes; По умолчанию: **auto**) – использование таблицы регистрации беспроводных клиентов для ускорения обучения прозрачного моста

**horizon** (none | целое значение 0..429496729; По умолчанию: **none**) – использовать расщепление горизонта для предотвращения образования сетевых петель

**interface** (название; По умолчанию: **none**) – название интерфейса

**path-cost** (целое значение: 0..65535; По умолчанию: **10**) – стоимость пути до интерфейса, используется протоколом STP для определения наилучшего пути

**point-to-point** (auto | yes | no; По умолчанию: **auto**) – использование режима «точка-точка»

**priority** (целое значение: 0..255; По умолчанию: **128**) – установка приоритета интерфейса

**Пример:**

Помещаем интерфейсы **ether1** и **ether2** в ранее созданный прозрачый мост **bridge1**

```
/interface bridge port> add bridge=bridge1 interface=ether1
/interface bridge port> add bridge=bridge1 interface=ether2
/interface bridge port> print
Flags: X – disabled, I – inactive, D – dynamic
 #   INTERFACE          BRIDGE          PRIORITY PATH-COST  HORIZON
 0   ether1             bridge1         0x80     10         none
 1   ether2             bridge1         0x80     10         none
```

**Мониторинг прозрачного моста**

Уровень меню: **/ interface bridge monitor**

**Описание параметров**

**current-mac-address** (MAC-адрес) - текущий MAC-адрес прозрачного моста

**designated-port-count** (целое значение) – количество назначенных портов прозрачного моста

**port-count** (целое значение) – количество портов прозрачного моста

**root-bridge** (yes | no) – является ли прозрачный мост корневым при использовании STP

**root-bridge-id** (текст) - идентификатор корневого моста (по сути – MAC-адрес моста), м.б. использован для определения приоритета моста

**root-path-cost** (целое значение) – стоимость пути до корневого моста

**root-port** (название) – порт, к которому подключён корневой

**state** (enabled | disabled) – статус прозрачного моста

**Пример**:

```
/interface bridge> monitor bridge1
              state: enabled
  current-mac-address: 00:0C:42:52:2E:CE
        root-bridge: yes
     root-bridge-id: 0x8000.00:00:00:00:00:00
     root-path-cost: 0
         root-port: none
        port-count: 2
designated-port-count: 0
```

### Мониторинг портов прозрачного моста

Уровень меню: **/ interface bridge port monitor**

### Описание параметров

**edge-port** (yes | no) – является ли порт пограничным

**edge-port-discovery** (yes | no) – может ли порт автоматически определять пограничные порты

**external-fdb** (yes | no) – используется ли таблица регистрации беспроводных клиентов вместо стандартной базы данных пересылки (forwarding database)

**forwarding** (yes | no) – статус порта

**learning** (yes | no) - статус порта

port-number (целое значение 1..4095) – номер порта

**point-to-point-port** (yes | no) – порт работает в режиме «точка-точка»

**role** (designated | root port | alternate | backup | disabled) алгоритм протокола (R)STP, применяемый на порту:

- **Disabled port** – отключение порта вручную, параметр не является частью протокола STP
- **Root port** – порт для соединения корневого моста с некорневым
- **Alternative port** – альтернативный путь к корневому мосту. Путь отличается от значения, используемого на root port.

- **Designated port** – порт, назначенный каждому сегменту сети
- **Backup port** – запасной путь к сегменту сети, к которому подключён другой порт прозрачного моста

**sending-rstp** (yes | no) – рассылает ли порт пакеты BPDU

**status** (in-bridge | inactive) – помещён ли порт в прозрачный мост

**Пример**:

```
/interface bridge port> monitor 0
            status: in-bridge
       port-number: 1
              role: designated-port
         edge-port: no
edge-port-discovery: yes
point-to-point-port: no
      external-fdb: no
      sending-rstp: no
          learning: yes
        forwarding: yes
```

### Мониторинг устройств, входящих в состав прозрачного моста

Уровень меню: **/ interface bridge host**

#### Описание параметров (доступны только для чтения)

**age** (время) – время, прошедшее с момента получения от хоста последнего пакета

**bridge** (название) – прозрачный мост, к которому принадлежит хост

**external-fdb** (флаг) – используется ли таблица регистрации беспроводных клиентов

**local** (флаг) – обозначение моста, используемого на локальном хосте

**mac-address** (MAC-адрес) – MAC-адрес хоста

**on-interface** (название) – интерфейс, используемый в бридже

**Пример:**

```
/interface bridge host> print
Flags: L – local, E – external-fdb
 BRIDGE      MAC-ADDRESS       ON-INTERFACE      AGE
 bridge1     00:11:32:1D:11:F8      ether1       41s
 bridge1     00:1D:60:34:15:E9      ether1       2m31s
 bridge1     00:80:48:56:DD:EC      ether1       0s
 bridge1     00:C0:26:AB:20:BB      ether1       41s
L bridge1    4C:5E:0C:8A:61:02      ether1       0s
 bridge1     5C:D9:98:F5:B2:0E      ether1       42s
 bridge1     60:A4:4C:24:87:9D      ether1       0s
 bridge1     90:2B:34:D5:0F:03      ether1       0s
 bridge1     90:2B:34:D5:BE:5A      ether1       0s
```

## Прозрачный мост и firewall

Уровень меню: **/interface bridge filter, /interface bridge nat**

Файервол прозрачного моста осуществляет фильтрацию проходящих через мост пакетов, обеспечивая функции безопасности и управления.

Трафик, проходящий через прозрачный мост, может быть пропущен через таблицы файервола, и проверен соответствующими настроенными правилами (см. **use-ip-firewall** на стр. **107**).

В данном случае используются две таблицы:

- **Filter** – с тремя предопределёнными цепочками:
  - **Input** – фильтрует пакеты, получателем которых является прозрачный мост (включая пакеты, подлежащие дальнейшей маршрутизации, поскольку они в любом случае проходят через мост)
  - **Output** – фильтрует пакеты, прошедшие через прозрачный мост (включая пакеты, среди прочего прошедшие и обычную маршрутизацию)
  - **Forward** – фильтрует пакеты от интерфейсов, находящихся в бридже (правила не распространяются на пакеты, подлежащие маршрутизации, правила работают только с пакетами, проходящими через порты текущего бриджа)
- **NAT** – используется для изменения MAC-адресов пакетов, проходящих через прозрачный мост и включает две цепочки:

- **srcnat** – используется для сокрытия хоста или сети за различными MAC-адресами. Цепочка работает с пакетами, проходящими через интерфейс прозрачного моста маршрутизатора
- **dstnat** – используется для перенаправления отдельных пакетов к другим получателям

На пакеты, проходящие через прозрачный мост могут быть установлены метки, аналогичные меткам, используемых в /ip firewall mangle (см. стр. **54**)

Основные параметры файервола прозрачного моста (аналогичные для таблиц Filter и NAT) перечислены ниже.

**Описание параметров**

**802.3-sap** (целое значение) – DSAP (Destination Service Access Point – Адрес точки входа сервиса назначения) и SSAP (Source Service Access Point – Адрес точки входа сервиса источника), два однобайтовых поля, позволяющие указать, какой сервис верхнего уровня пересылает данные с помощью этого кадра. Значения обоих байтов всегда одинаковы. В этом параметре указываются две шестнадцатеричные цифры.

**802.3-type** (целое значение) – тип протокола Ethernet. Работает только в том случае, если параметр 802.3-sap указан как 0xAA (SNAP – Sub-Network Attachment Point header – протокол доступа к подсети, использующийся для инкапсуляции дейтаграмм IP и запросов ARP в сетях IEEE 802). Например, протокол AppleTalk может быть указан как SAP со значением 0xAA плюс SNAP со значением 0x809B.

**arp-dst-address** (IP-адрес) – ARP IP-адрес получателя

**arp-dst-mac-address** (MAC-адрес) – ARP MAC-адрес получателя

**arp-gratuitous** (yes | no ) – Соответствие самообращённым ARP-запросам

**arp-hardware-type** (целое значение; По умолчанию: 1) – номер канального протокола передачи, для Ethernet номер =1

**arp-opcode** (arp-nak | drarp-error | drarp-reply | drarp-request | inarp-reply | inarp-request | reply | reply-reverse | request | request-reverse) ARP opcode (packet type)

- **arp-nak** – негативный ответ ARP (используется крайне редко, в основном в сетях ATM)
- **drarp-error** – код ошибки для динамического RARP, сообщает, что IP-адрес по указанному MAC-адресу не может быть получен
- **drarp-reply** – ответ динамического RARP reply, сообщающий IP-адрес хоста
- **drarp-request** – запрос динамического RARP на получение IP-адреса по указанному MAC-адресу
- **inarp-reply** – ответ InverseARP
- **inarp-request** – запрос InverseARP

- **reply** – стандартный ответ ARP, сообщающий MAC-адрес
- **reply-reverse** – ответ обратного ARP (RARP), сообщающий IP-адрес
- **request** – стандартный запрос ARP на получение MAC-адреса по известному IP-адресу
- **request-reverse** – запрос обратного ARP (RARP) на получение IP-адреса по известному MAC-адресу (используется хостами для определения собственного IP-адреса, по аналогии с сервисом DHCP)

**arp-packet-type** (целое значение: 0..65535 или в шестнадцатеричном виде 0x0000-0xffff) – код сетевого протокола

**arp-src-address** (IP-адрес) – ARP, IP-адрес источника

**arp-src-mac-address** (MAC-адрес) – ARP, MAC-адрес источника

**chain** (текст) – Цепочка таблицы файервола прозрачного моста (стандартная или определённая пользователем)

**dst-address** (IP-адрес) – IP-адрес получателя (только если MAC-протокол установлен в IPv4)

**dst-mac-address** (MAC-адрес) – MAC-адрес получателя

**dst-port** (целое значение 0..65535) – Порт или диапазон портов получателя (только для протоколов TCP или UDP)

**in-bridge** (название) – Название виртуального интерфейса прозрачного моста, на который поступают пакеты

**in-interface** (название) – Название физического интерфейса (т.н. порт бриджа), на который поступают пакеты

**ingress-priority** (целое значение 0..63) – приоритет пакета, может быть получен из VLAN, WMM или через биты EXP в MPLS-метке.

**ip-protocol** (ddp | egp | encap | etherip | ggp | gre | hmp | icmp | icmpv6 | idpr-cmtp | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | ipv6 | ipv6-frag | ipv6-nonxt | ipv6-opts | ipv6-route | iso-tp4 | l2tp | ospf | pim | pup | rdp | rspf | rsvp | st | tcp | udp | vmtp | vrrp | xns-idp | xtp) IP-протокол (только если MAC-протокол установлен в IPv4)

- **ddp** – протокол передачи дейтаграмм
- **egp** – протокол внешнего шлюза
- **encap** – ip-инкапсуляция
- **etherip** – туннелирование кадров Ethernet в IP-дейтаграммах
- **ggp** – межшлюзовой протокол
- **gre** – общая инкапсуляция маршрутов
- **hmp** – протокол мониторирования ЭВМ
- **icmp** – протокол межсетевых управляющих сообщений (IPv4)
- **icmpv6** – протокол межсетевых управляющих сообщений (IPv6)
- **idpr-cmtp** – idpr control message transport

- **igmp** – протокол управления групповой (мультикастовой) передачей данных в сетях
- **ipencap** – инкапсуляция IP в IP
- **ipip** – инкапсуляция IP
- **ipsec-ah** – протокол IPsec AH
- **ipsec-esp** – протокол IPsec ESP
- **ipv6**
- **ipv6-frag**
- **ipv6-nonxt**
- **ipv6-opts**
- **ipv6-route**
- **iso-tp4** – транспортный протокол, class 4
- **l2tp** – протокол туннелирования второго уровня
- **ospf**
- **pim** – мультикастинг, не зависящий от протокола
- **pup**
- **rspf**
- **rsvp**
- **rdp** (reliable datagram protocol)
- **st** (st datagram mode)
- **tcp**
- **udp**
- **vmtp**
- **vrrp**
- **xns-idp**
- **xtp** (xpress transfer protocol)

**jump-target** (название) – если указано действие jump, то происходит переход на пользовательскую цепочку таблицы файервола для дальнейшей обработки пакета

**limit** (целое значение/время, целое значение) – ограничение прохождения пакетов в соответствии со следующими критериями:

- **count** – среднее кол-во пакетов в секунду, если не указан параметр **time**
- **time** – временной интервал, в течение которого происходит подсчет кол-ва пакетов
- **burst** – количество пакетов, проходящих в пике

**log-prefix** (текст) – текстовый префикс, печатаемый в каждой строке файла журнала

**mac-protocol** (802.2 | arp | ip | ipv6 | ipx | length | mpls-multicast | mpls-unicast | pppoe | pppoe-discovery | rarp | vlan или целое значение в диапазоне: 0..65535 или в шестнадцатеричном виде 0x0000-0xffff) – протокол MAC-уровня

- **802.2**
- **arp** – тип 0x0806
- **ip** – тип 0x0800
- **ipv6** – тип 0x86dd
- **ipx** – тип 0x8137 (Internetwork Packet Exchange)

- **length**
- **mpls-multicast** – тип 0x8848
- **mpls-unicast** – тип 0x8847
- **ppoe** – тип 0x8864
- **ppoe-discovery** – тип 0x8863
- **rarp** – тип 0x8035
- **vlan** – тип 0x8100 (802.1Q tagged VLAN)

**out-bridge** (название) – исходящий интерфейс прозрачного моста

**out-interface** (название) – название исходящего физического интерфейса

**packet-mark** (название) – сравнение пакета с соответствующей меткой

**packet-type** (broadcast | host | multicast | other-host) – тип пакета:

- **broadcast** – широковещательный
- **host** – пакет назначен прозрачному мосту
- **multicast** – мультикастовый
- **other-host** – пакет назначен не прозрачному мосту, а другому получателю

**src-address** (IP-адрес) – IP-адрес источника (только если MAC-протокол установлен в IPv4)

**src-mac-address** (MAC-адрес) – MAC-адрес источника

**src-port** (целое значение 0..65535) – порт источника или диапазон портов (только для протоколов TCP и UDP)

**stp-flags** (topology-change | topology-change-ack) – флаг BPDU (Bridge Protocol Data Unit)

- **topology-change** – конфигурационный флаг, устанавливается как заявка на возможность устройства стать корневым коммутатором, на основании которой происходит определение активной конфигурации сети
- **topology-change-ack** – флаг уведомления о реконфигурации, устанавливается в случае события, которое требует проведения реконфигурации сети – отказ линии связи, отказ порта, изменение приоритетов коммутатора или портов

**stp-forward-delay** (время 0..65535) – таймер задержки пересылки

**stp-hello-time** (время 0..65535) – интервал рассылки HELLO-пакетов в STP

**stp-max-age** (время 0..65535) – максимальное время жизни сообщения

**stp-msg-age** (время 0..65535) – время жизни сообщения

**stp-port** (целое значение 0..65535) – номер порта STP

**stp-root-address** (MAC-адрес) – MAC-адрес корневого моста

**stp-root-cost** (целое значение 0..65535) – стоимость корневого моста

**stp-root-priority** (целое значение 0..65535) – приоритет корневого моста

**stp-sender-address** (MAC-адрес) – MAC-адрес отправителя сообщения STP

**stp-sender-priority** (целое значение 0..65535) – приоритет отправителя сообщения STP

**stp-type** (*config* | *tcn*) – тип BPDU:

- **config** – конфигурационный
- **tcn** – уведомление о реконфигурации

**vlan-encap** (802.2 | arp | ip | ipv6 | ipx | length | mpls-multicast | mpls-unicast | pppoe | pppoe-discovery | rarp | vlan или целое значение в виде: 0..65535 или в шестнадцатеричном формате 0x0000-0xffff) – тип прококола, инкапсулированного в кадре VLAN

**vlan-id** (целое значение 0..4095) – идентификатор VLAN

**vlan-priority** (целое значение 0..7) – приоритет VLAN

### Фильтр пакетов прозрачного моста

Уровень меню: **/interface bridge filter**

#### Описание параметров

**action** (accept | drop | jump | log | mark-packet | passthrough | return | set-priority)

- **accept** – принять пакет. Без выполнения каких–либо правил в цепочках таблиц
- **drop** – удалить пакет (без отправки сообщения ICMP об отклонении пакета)
- **jump** – перейти на цепочку, указанную в аргументе **jump-target**
- **log** – прописать информацию о пакете в файле журнала

- **mark** – пометить пакет меткой с целью использования поставленной метки в дальнейшем
- **passthrough** – игнорировать текущее правило и перейти к следующему. То же самое, что и выключение правила, за исключением возможности подсчёта пакетов
- **return** – вернуть контроль в то место родительской цепочки, откуда был совершен переход
- **set-priority** – установить приоритет согласно параметру **new-priority** (при настроенных VLAN или WMM на беспроводном интерфейсе)

### Прозрачный мост и NAT

Уровень меню: **/interface bridge nat**

**action** (accept | drop | jump | mark-packet | redirect | set-priority | arp-reply | dst-nat | log | passthrough | return | src-nat)

- **accept** – принять пакет. Без выполнения каких–либо правил в цепочках таблиц
- **arp-reply** – отправка ответа на запрос ARP (любые другие пакеты будут проигнорированы данным правилом) с соответствующим MAC-адресом (применимо только в цепочке dstnat)
- **drop** – удалить пакет (без отправки сообщения ICMP об отклонении пакета)
- **dst-nat** – изменение MAC-адреса получателя пакета (применимо только в цепочке dstnat)
- **jump** – перейти на цепочку, указанную в аргументе **jump-target**
- **log** – прописать информацию о пакете в файле журнала
- **mark** – пометить пакет меткой с целью дальнейшего использования поставленной метки
- **passthrough** – игнорировать текущее правило и перейти к следующему. То же самое, что и выключение правила, за исключением возможности подсчёта пакетов
- **redirect** – перенаправление пакета на тот же мост (применимо только в цепочке dstnat)
- **return** – вернуть контроль в то место родительской цепочки, откуда был совершен переход
- **set-priority** – установить приоритет согласно параметру **new-priority** (при настроенных VLAN или WMM на беспроводном интерфейсе)
- **src-nat** – изменить в пакете MAC-адрес источника (применимо только в цепочке srcnat)

**to-arp-reply-mac-address** (MAC-адрес) – MAC-адрес источника, помещаемый в кадр Ethernet, если выбрано действие **arp-reply**

**to-dst-mac-address** (MAC-адрес) – MAC-адрес получателя, помещаемый в кадр Ethernet, если выбрано действие **dst-nat**

**to-src-mac-address** (MAC-адрес) – MAC-адрес источника, помещаемый в кадр Ethernet, если выбрано действие **src-nat**

# Основные параметры беспроводного интерфейса

RourtOS обеспечивает полную поддержку стандартов 802.11a, 802.11b, 802.11g, 802.11n и 802.11ac. В дополнение к основным параметрам обеспечивается поддержка шифрования радиоканала (WEP, WPA, AES), прозрачного бриджинга (WDS), динамического выбора частоты (DFS), виртуальной базовой станции (virtual AP); поддержка проприетарных протоколов Nstreme и NV2.

Беспроводной интерфейс поддерживает несколько режимов работы, основными из которых являются: клиентская станция, базовая станция и прозрачный мост. Только данные режимы поддерживаются протоколом NV2.

Уровень меню: **/ interface wireless**

**Описание параметров**

**adaptive-noise-immunity** (ap-and-client-mode | client-mode | none; По умолчанию: **none**) Работает только на радиочипсете Atheros.

**allow-sharedkey** (yes | no; По умолчанию: **no**) Разрешить подключаться клиентам с включенным WEP Shared Key. **Обратите внимание**: таким клиентам не требуется аутентификация, они сразу подключаются к базовой станции (если это разрешено в списках доступа)

**antenna-gain** (целое значение [0...4294967295]; По умолчанию: **0**) – усиление антенны, измеряется в dBi, используется для расчета максимально разрешенной мощности передатчика, в соотвтетствии с правилами той страны, в которой эксплуатируется оборудование

**antenna-mode** (ant-a | ant-b | rxa-txb | txa-rxb;)– выбор антенн для приема и передачи сигнала

- **ant-a** – использовать только антенну 'a'
- **ant-b** – использовать только антенну 'b'
- **txa-rxb** – использовать антенну 'a' для передачи, антенну 'b' – длбя получения сигнала
- **rxa-txb** – использовать антенну 'b' для передачи, антенну 'a' – длбя получения сигнала

**area** (строка) – возможность разделения беспроводной сети на определенные группы. Данным параметром можно указать группу, к которой будет принадлежать настраиваемая базовая станция, с целью дальнейшего разграничения доступа через параметр area-prefix (см. стр. 136), настроенного в connect-list (см. стр. 135).

**arp** (disabled | enabled | proxy-arp | reply-only; По умолчанию: **enabled**) – режим использования протокола ARP

**band** (2ghz-b | 2ghz-b/g | 2ghz-b/g/n | 2ghz-onlyg | 2ghz-onlyn | 5ghz-a | 5ghz-a/n | 5ghz-onlyn | 5ghz-a/n/ac | 5ghz-only-AC;)– выбор частотного диапазона и протокола работы беспроводного интерфейса

**basic-rates-a/g** (12Мбит/с | 18Мбит/с | 24Мбит/с | 36Мбит/с | 48Мбит/с | 54Мбит/с | 6Мбит/с | 9Мбит/с; По умолчанию: **6Мбит/с**) – указание скорости передачи для протоколов 802.11a и 802.11g

**basic-rates-b** (11Мбит/с | 1Мбит/с | 2Мбит/с | 5.5Мбит/с; По умолчанию: **1Мбит/с**) – указание скорости передачи для протоколов 802.11b и 802.11g

Клиенткая станция может подсоединиться к базовой только в случае поддержки клиентом всех скоростей, заявленных базовой станцией. Базовая станция может подсоединиться к другой базовой станции в режиме прозрачного моста только в случае поддержки первой базой всех скоростей, заявленных второй базовой станцией.

Параметр актуален только в режиме базовой станции, при настроенном параметре rate-set (см. стр. **126**).

**bridge-mode** (disabled | enabled; По умолчанию: **enabled**) – поддержка режима station-bridge.

**burst-time** (целое значение| disabled; По умолчанию: **disabled**) – время в микросекундах, в течение которого данные будут передаваться безостановочно. В это время другие станции НЕ МОГУТ передавать данные. Параметр акуален только для радиочипсетов AR5000, AR5001X, и AR5001X+.

**channel-width** (10mhz | 20/40mhz-ht-above | 20/40mhz-ht-below | 20mhz | 40mhz-turbo | 5mhz; По умолчанию: **20mhz**) – ширина спектра сигнала, суффиксы ht-above и ht-below указывают на использование дополнительных 20MHz, используемых выше или ниже основной частоты. Задействование дополнительных частот увеличивает пропускную способность радиоканала при использовании протокола 802.11n.

**comment** (строка) - краткое описание интерфейса

**compression** (yes | no; По умолчанию: **no**) – использование аппаратного сжатия данных, если чипсет поддерживает данную функцию. Связь будет работать даже в том случае, если другие устройства не поддерживают данную функцию.

**country** (страна | no_country_set; По умолчанию: **no_country_set**) – ограничение ширины спектра сингала, частоты и мощности передатчика в зависимости от правил указанной страны. Также меняются стандартные значения параметра scan-list (см. стр. 127).

**default-ap-tx-limit** (целое значение [0...4294967295]; По умолчанию: **0**) – значение параметра ap-tx-limit (см. стр. 132) для клиентов, не указанных в access-list (см. стр. 132). 0 – отсутствие ограничний по мощности.

**default-authentication** (yes | no; По умолчанию: **yes**) – параметр базовой станции, указывающий тип **аутентификации** для склиентстких станций, не подпадающих под правила в access-list. Для клиентских станций – значение параметра **connect** базовых станций, не подпадающих под правила в connect-list (см. стр. 135).

**default-client-tx-limit** (целое значение [0....4294967295]; По умолчанию: **0**) – значение параметра client-tx-limit (см. стр. 133) для клиентов, не указанных в access-list. 0 – отсутствие ограничний по мощности.

**default-forwarding** (yes | no; По умолчанию: yes) – перенаправление клиентов, не указанных в access-list (см. стр. 132).

**dfs-mode** (no-radar-detect | none | radar-detec; По умолчанию: **none**) – настройка DFS (динамического выбора частоты).

- **none** – DFS отключен
- **no-radar-detect** – выбор и использование канала из списка **scan-list** с наименьшим номером из обнаруженных. Не работает в режиме 'wds-slave'.
- **radar-detect** – выбор и использование канала из списка **scan-list** с наименьшим номером из обнаруженных, если он остается свободен на протяжении 60 сек. Иначе происходит поиск следующего доступного канала. Работает только в режиме 'AP mode'.

**disable-running-check** (yes | no; По умолчанию: **no**) – при включеном параметре интерфейс всегда значится как активный (флаг 'R'). В противном случае флаг устанавливается только в случае, если база ассоциирована с одними или несколькими клиентами, либо если клиент ассоциирован с базой.

**disabled** (yes | no; По умолчанию: **yes**) – включение /отключение интерфейса

**disconnect-timeout** (время [0…15] сек; По умолчанию: **3 сек**) – временной интервал в секундах, по истечении которого соединение будет считаться утерянным. Рассчитывается по формуле 3 * (hw-retries + 1). В течении данного интервала будут предприняты повторные попытки отправки пакетов в соответствии с параметром on-fail-retry-time (см. стр. 125). Если в течении интервала **diconnect-timeout** пакет так и не будет

получен – соединение закрывается и в системный журнал добавляется запись "extensive data loss". При восстановлении соединения таймер параметра сбрасывается.

**distance** (целое значение| dynamic | indoors; По умолчанию: **dynamic**) – от данного параметра зависит как долго ожидать пакета подствержденния (ACK) перед тем, как предпринять повторную отправку пакетов. При значении 'dynamic' базовая станция автоматически выбирает наименьшее рабочее значение и использует его при соединении со всеми клиентами. Параметр не используется в протоколе Nstreme.

**frame-lifetime** (целое значение [0...4294967295]; По умолчанию: **0**) – удаление пакетов из очереди на отправку, если они находятся в ней дольше, чем указано в данном параметре. По умолчанию используется ноль, что означает, что пакеты будут удалаться только после закрытия соединения.

**frequency** (целое значение [0...4294967295];)– частота вещания базовой станции в МГц. Доступные значения зависят от выбранного частотного диапазона, указанной страны и характеристик радиокарты. Параметр не актуален для режимов **station** и **wds-slave**, а также при включеном параметре dfs-mode (см. стр. 120).

**Обратите внимание**: При использовании режима "superchannel" будут испольоваться ЛЮБЫЕ выставленные частоты, поддерживаемые радиокартой, но все нестандартные частоты должны быть предварительно настроены в scan-list (см. стр. 127), иначе они не будут сканироватиься. В утилите Winbox все частоты, настроенные через scan-list, выделены жирным шрифтом, остальные значения требуют настройки в scan-list на клиентских устройствах.

**frequency-mode** (manual-txpower | regulatory-domain | superchannel; По умолчанию: **manual-txpower**) – частотные режимы, доступны в трех вариантах

- **regulatory-domain** – ограничение доступных частотных каналов и мощности передатчика в зависимости от указанной страны
- **manual-txpower** – то же, что и выше, но без ограничения мощности передатчика
- **superchannel** – доступны все частотные каналы, поддерживаемые радиочипсетом (список доступных частотных каналов для каждого радиодиапазона можно посмотреть в меню **/wireless info print**. Данный режим позволяет протестировать ВСЕ доступные частотные каналы, все зависимости от стандартных настроек scan-list и/или указанной страны. Используйте этот режим на свой собственный страх и риск. Начиная с версии RouterOS 4.3 этот режим доступен всем желающим, без необходимости приобретения каких-либо дополнительных лицензий).

**frequency-offset** (целое значение [-2147483648…2147483647]; По умолчанию: **0**) – смещение рабочей частоты на указанное значение, если используется встроенный частотный конвертер. Например, если радиокарточка работает на частоте 4000 Мгц, но RouterOS отображает 5000 МГц, установите смешение в 1000 МГц для корректного отображения значения частоты. Значения могут быть как положительными, так и отрицательными.

**hide-ssid** (yes | no; По умолчанию: **no**) – скрытие идентификатора сети

- **yes** – базовая станция не включает SSID в кадры биконов, и не отвечает на запросы с широковещательными SSID
- **no** – базовая станция включает SSID в кадры биконов, и отвечает на запросы с широковещательными SSID

Параметр актуален только для базовых станций. Включение данного параметра может привести к удалению отображения базовой станции в ПО некоторых клиентских станций. Включение данного параметра не повышает безопасность беспроводной сети, поскольку SSID остается доступен в других кадрах базовой станции.

**ht-ampdu-priorities** (целое значение [0...7]; По умолчанию: **0**) – приоритеты пакетов, которые будут отправлены с использованием механизма AMPDU (Aggregated Mac Protocol Data Unit). Использование AMPDU увеличивает пропускную способность канала, но может увеличить время задержки и нежетателен для мультимедийного трафика.

**ht-amsdu-limit** (целое значение [0...8192]; По умолчанию: **8192**) - максимальный размер агрегированного пакета AMSDU (Aggregated Mac Service Data Unit). Использование AMSDU может значительно увеличить пропускную способность канала при использовании пакетов небольшого размера, но может и увеличить время задержки в случае потери агрегированного пакета и его повторной отправки. AMSDU также увеличивет нагрузку на CPU.

**ht-amsdu-threshold** (целое значение [0...8192]; По умолчанию: **8192**) – максимальный размер кадра, который может быть включен в пакет AMSDU

**ht-basic-mcs** (список значений (mcs-0 | mcs-1 | mcs-2 | mcs-3 | mcs-4 | mcs-5 | mcs-6 | mcs-7 | mcs-8 | mcs-9 | mcs-10 | mcs-11 | mcs-12 | mcs-13 | mcs-14 | mcs-15 | mcs-16 | mcs-17 | mcs-18 | mcs-19 | mcs-20 | mcs-21 | mcs-22 | mcs-23); По умолчанию: **mcs-0; mcs-1; mcs-2; mcs-3; mcs-4; mcs-5; mcs-6; mcs-7**) – кодовые схемы модуляци, должны поддерживаться всеми клиентсткими станциями, желающими подключиться к базовой станции по протоколу 802.11n.

**ht-guard-interval** (any | long; По умолчанию: **any**) – настройка защитного интервала. Параметр **any** позволяет использовать либо короткий, либо увеличенный интервал в зависимости от передаваемых данных. Параметр **long** позволяет использовать только увеличенный интервал.

**ht-rxchains** (целые значения [0...2]; По умолчанию: **0**) – антенны, используемые для получения данных

**ht-supported-mcs** (list of (mcs-0 | mcs-1 | mcs-2 | mcs-3 | mcs-4 | mcs-5 | mcs-6 | mcs-7 | mcs-8 | mcs-9 | mcs-10 | mcs-11 | mcs-12 | mcs-13 | mcs-14 | mcs-15 | mcs-16 | mcs-17 | mcs-18 | mcs-19 | mcs-20 | mcs-21 | mcs-22 | mcs-23); По умолчанию: **mcs-0; mcs-1; mcs-2; mcs-3; mcs-4; mcs-5; mcs-6; mcs-7; mcs-8; mcs-9; mcs-10; mcs-11; mcs-12; mcs-13; mcs-14; mcs-15; mcs-16; mcs-17; mcs-18; mcs-19; mcs-20; mcs-21; mcs-22; mcs-23**) – кодовые схемы модуляци, поддерживаемые устройством, (см. описание протокола 802.11n).

**ht-txchains** (целые значения [0...2]; По умолчанию: **0**) – антенны, используемые для передачи данных

**hw-fragmentation-threshold** (целое значение [256…3000] | **disabled**; По умолчанию: **0**) – максимальный размер фрагмента пакета в байтах при передаче его по беспроводному каналу для увеличения вероятности успешной передачи; передача нефрагментированного пакета более эффективна с точки зрения потребления сетевых ресурсов

**hw-protection-mode** (cts-to-self | none | rts-cts; По умолчанию: **none**) – использование режимов «RTS/CTS» или «CTS to self» для решения проблемы скрытого узла

**hw-protection-threshold** (целое значение [0...65535]; По умолчанию: **0**) – размер пакета при использовании параметра **hw-protection-mode**

**hw-retries** (целое значение [0...15]; По умолчанию: **7**) – количество повторных отправок потерянных пакетов. После трёх последовательных неудачных отправок пакетов отправка приостанавливается на время, указанное в параметре on-fail-retry-time (см. стр. 125). После этого пакет отправляется повторно до тех пор, пока передача не закончится, либо пока клиент не отключится из-за превышения значения параметра disconnect-timeout (см. стр. 120). Пакет также будет отброшен, если в течении указанного интервала будет превышего значение параметра frame-lifetime (см. стр. 121).

**l2mtu** (целое значение [0...65536]; По умолчанию: **1600**) – максимальный размер пакета без учета размера MAC-заголовка, который может быть отправлен через указанный интерфейс

**mac-address** (MAC-адрес) – MAC-адрес беспроводного интерфейса

**master-interface** (строка) – название беспроводного интерфейса, работающего в режиме виртуальной базовой станции. Интерфейс виртуальной базовой станрции работает только в том случае, если master-interface настоен в режиме **ap-bridge, bridge** или **wds-slave**. Параметр актуален только для интерфейсов виртуальной базовой станции.

**max-station-count** (целое значение [1...2007]; По умолчанию: **2007**) – максимально допустимое количество клиентских станций, ассоциированных с базовой станцией. Устройства, работающие в режиме прозрачного моста, также учитываются.

**mode** (station | station-wds | ap-bridge | bridge | alignment-only | nstreme-dual-slave | wds-slave | station-pseudobridge | station-pseudobridge-clone | station-bridge; По умолчанию: station) – выбор режима работы радиомаршрутизатора

Режимы работы клиентской станции:

- **station** – стандартный режим, поис и подключение к доступным БС. Find and connect to acceptable AP.
- **station-wds** – подключение к БС в режиме WDS. В настройках БС должно быть указано, что к ней разрешено подключение клиентов в данном режиме
- **station-pseudobridge** – то же, что и **station**, но с трансляцией MAC-адресов всех пакетов. Интерфейсы могут работать в режиме прозрачного моста.

- **station-pseudobridge-clone** – то же, что и **station-pseudobridge**, но с использование параметра station-bridge-clone-mac (см. стр. 127) для подключения к БС.

Режимы работы базовой станции (БС):

- **ap-bridge** – стандартный режим работы БС.
- **bridge** – то же, что и **ap-bridge**, но возможна ассоциация не более чем с одним клиентом.
- **wds-slave** – поиск и установка wds-соединения с другой БС с аналогичным SSID. Если параметр dfs-mode (см. стр. 120) установлен в значение **radar-detect**, то базовые станции с включённым параметром hide-ssid (см. стр. 121) не будут обнаружены в процессе поиска.

Специальные режимы работы:

- **alignment-only** – интерфейс находится в режиме постоянной передачи, используется для юстировки устройства на удаленную антенну
- **nstreme-dual-slave** – использование интерфейса при настройке nstreme-dual

При трансляции MAC-адресов в режиме **pseudobridge** создается таблица соответствий IP- и MAC-адресов. Все пакеты, отправляемые на БС, имеют MAC-адрес **pseudobridge**, а MAC-адреса получаемых пакетов восстанавливаются из таблицы трансляции адресов.
**Обратите внимание**: на даный момент протокол IPv6 не работает в режиме **pseudobridge.**
Виртуальные интерфейсы БС не поддерживают данный параметр, они наследуют режтмы работы основного интерфейса.

**mtu** (целое значение [0...65536]; По умолчанию: **1500**) – размер пакета в байтах

**multicast-helper** (default | disabled | full; По умолчанию: **default**) Параметр может быть включен только на базовых станциях, при этом клиенты должны работать в режиме **station-bridge**. Параметр доступен в прошивках версии 5.15 и выше.

- **disabled** – выключение хелпера и отправка мультикастовых пакетов с мультикастовыми MAC-адресами получателей
- **full** – все мультикастовые MAC-адреса пакетов перед отправкой заменяются на уникастовые
- **default** – параметр по умолчанию, на данный момент установлен в **disabled**. Значение данного параметра может измениться в будущих релизах

**name** (строка;)– название интерфейса

**noise-floor-threshold** (default | целое значение [-128...127]; По умолчанию: **default**) – Параметр актуален только для чипсета AR5211.

**nv2-cell-radius** (целое значение [10...200]; По умолчанию: **30**) – параметр задается на базовой станции и определяет тайм-слот для клиентских станций; также используется для оценки дистанции до клиентов, поэтому слишком маленькое значение параметра может

приводить в потере связи. Также не рекомендуется без явных причин указывать слишком большое значение данного параметра, поскольку часть временного интервала, выделенного для обмена данными, будет пропадать впустую.

- Для базовых станций: дистанция в километрах до самого удаленного клиента
- Для клиентских станций: не актуально

**nv2-noise-floor-offset** (default | целое значение [0...20]; По умолчанию: **default**) – смещение для уровня шума

**nv2-preshared-key** (строка) – значение предустановленного ключа в протоколе NV2

**nv2-qos** (default | frame-priority; По умолчанию: **default**) – установка приоритета пакета, в первую очередь отправляются пакеты с наибольшим приоритетом. Используйте даный параметр с осторожностью: пакеты с небольшим приоритетом не будут отправлены до тех пор, пока не закончится рассылка пакетов с макимальными приоритетами. Параметр актуален только для базовых станций.

- **frame-priority** – приоритет выставляется вручную через правила файервола (таблица Mangle)
- **default** – оптимальное значение приоритета для получения пакетов небольших размеров

**nv2-queue-count** (целое значение [2…8]; По умолчанию: **2**) – количество очередей в протоколе NV2

**nv2-security** (disabled | enabled; По умолчанию: **disabled**) – включение шифрования в протоколе NV2

**on-fail-retry-time** (время [100мсек…1сек]; По умолчанию: **100мсек**) – система выжидает в течение указанного интервала времени после троекратной безуспешной попытки отправки данных

**periodic-calibration (**default | disabled | enabled; По умолчанию: **default**) – калибровка радиокарты. Значение параметра default зависит от типа используемой радиокарты. Актуально только для чипсета Atheros.

**periodic-calibration-interval** (целое значение [1...10000]; По умолчанию: **60**) – периодичность ралибровки радиокарты. Актуально только для чипсета Atheros.

**preamble-mode** (both | long | short; По умолчанию: **both**) – тип преамбулы в служебной части пакета при работе по протоколу 802.11b.

- Для базовой стации:
  - **long** – не использовать короткую преамбулу
  - **short** – совмесимость с короткой преамбулой, не соединяется с клиентами, не имеющими данной совместимости
  - **both** – совмесимость с короткой преамбулой
- Для клиентской станции:
  - **long** – не использовать короткую преамбулу

- **short** – не соединяется с базовой станцией, если та не паоддерживает короткую преамбулу
- **both** – используется короткая преамбула, если она поддерживается базовой станцией

**prism-cardtype** (100mW | 200mW | 30mW; По умолчанию: ) – указание типа радиокарты Prism

**proprietary-extension** (post-2.9.25 | pre-2.9.25; По умолчанию: **post-2.9.25**) – способ передачи служебной информации по управлению пакетами

- **pre-2.9.25** – устаревший метод, не совместим с некоторым клиентским оборудованием
- **post-2.9.25** – стандартный метод, поддерживается современным беспроводным оборудованием.

**radio-name** (строка; По умолчанию: **MAC-адрес устройства**) – описательное название устройства, отображаемое в регистрационной таблице удаленного устройства

**rate-selection** (advanced | legacy; По умолчанию: **advanced**) – режим выбора скорости, в прошивке версии 5.9 и выше по умолчанию используется режим **advanced**, поскольку режим **legacy** оказался малоэффективен.

**rate-set** (configured | default; По умолчанию: **default**) – доступны два варианта:

- **default** – используются режимы **default basic** и **supported rat**. Значения режимов **basic-rates** и **supported-rates** не используются.
- **configured** – используются значения режимов **basic-rates**, **supported-rates**, **basic-mcs**, **mcs**.

| **Используемые режимы в зависимости от частотного диапазона** | | | | | | |
|---|---|---|---|---|---|---|
| **Диапазон** | **basic rates** | **basic-HT-mcs** | **basic-VHT-mcs** | **VHT-mcs** | **HT-mcs** | **supported rates** |
| 2.4ghz-b | 1 | - | - | - | - | 1-11 |
| 2.4ghz-onlyg | 6 | - | - | - | - | 1-11,6-54 |
| 2.4ghz-onlyn | 6 | 0-7 | - | - | 0-23 | 1-11,6-54 |
| 2.4ghz-b/g | 1-11 | - | - | - | - | 1-11,6-54 |
| 2.4ghz-b/g/n | 1-11 | none- | - | - | 0-23 | 1-11,6-54 |
| 2.4ghz-g-turbo | 6 | - | - | - | - | 6-54 |
| 5ghz-a | 6 | - | - | - | - | 6-54 |
| 5ghz-a/n | 6 | none | - | - | 0-23 | 6-54 |
| 5ghz-onlyn | 6 | 0-7 | - | - | 0-23 | 6-54 |
| 5ghz-a/n/ac | 6 | none | none | 0-9 | 0-23 | 6-54 |
| 5ghz-onlyac | 6 | none | 0-7 | 0-9 | 0-23 | 6-54 |

| **Используемые параметры, если rate-set=configured** | |
|---|---|
| **Диапазон** | **Параметры** |
| 2.4ghz-b | basic-b, supported-b |
| 2.4ghz-b/g, 2.4ghz-onlyg | basic-b, supported-b, basic-a/g, supported-a/g |
| 2.4ghz-onlyn, 2.4ghz-b/g/n | basic-b, supported-b, basic-a/g, supported-a/g, ht-basic-mcs, ht-supported-mcs |
| 5ghz-a | basic-a/g,supported-a/g |
| 5ghz-a/n, 5ghz-onlyn | basic-a/g,supported-a/g,ht-basic-mcs,ht-supported-mcs |
| 5ghz-a/n/ac, 5ghz-onlyac | basic-a/g,supported-a/g,ht-basic-mcs,ht-supported-mcs,vht-basic-mcs,vht-supported-mcs |

Настройки, не зависящие от **rate-set**:

1. допустимые **mcs** в зависимости от количества используемых каналов:
   - 1 канал: 0-7
   - 2 канала: 0-15
   - 3 канала: 0-23
2. Если ширина спектра сигнала составляет более 20 МГЦ, то протокол 802.11b не используется (если явно не выбран вариант **2.4ghz-b**)

**scan-list** (список частот и частотных диапазонов | default; По умолчанию: **default**) – При выставлении значения **default** могут быть использованы все каналы из выбранного радиодиапазона, поддерживаемого радиокартой, в соответствии со значениями параметров country (см. стр. 120) и frequency-mode (см. стр. 121). В стандартном скан-листе для режима **5ghz** частотные каналы меняются с шагом в 20 Мгц, для режима **5ghz-turbo** – с шагом в 40 МГц, для других режимов – с шагом в 5 МГц. Если **scan-list** настроен вручную, то доступны все указанные в нём каналы. (Пример: **scan-list**=default,5200-5245,2412-2427 – используются все стандартные значения скан-листа для текущего диапазона + каналы из диапазонов 5200-5245 или 2412-2427 МГц.) При использовании Winbox или Webfig в прошивках версии 6.0 и выше, необходимо добавлять каждую частоту или диапазон частот в ***отдельный*** скан-лист. Списки со значениями, разделенными запятой, в этих прошивках более не поддерживаются.

**security-profile** (строка; По умолчанию: **default**) – название профайла из security-profiles

**ssid** (строка (0...32 символа)) – SSID (service set identifier) – название идентификатора беспроводной сети

**station-bridge-clone-mac** (MAC-адрес;) – параметр актуален только в режиме station-pseudobridge-clone (см. стр. 127). Указанный MAC-адрес используется для подключения к базовой станции. Если указан адрес **00:00:00:00:00:00**, клиентская станция будет использовать MAC-адрес беспроводного интерфейса. Если потребуется передать пакет от устройства с другим MAC-адресом, клиент переподключится к базовой станции с новым MAC-адресом.

**supported-rates-a/g** ([12Мбит/с | 18Мбит/с | 24Мбит/с | 36Мбит/с | 48Мбит/с | 54Мбит/с | 6Мбит/с | 9Мбит/с]; По умолчанию: **6Мбит/с; 9Мбит/с; 12Мбит/с; 18Мбит/с; 24Мбит/с;**

**36Мбит/с; 48Мбит/с; 54Мбит/с**) – список поддерживаемых режимов, используемых во всех диапазонах кроме **2ghz-b**.

**supported-rates-b** ([11Мбит/с | 1Мбит/с | 2Мбит/с | 5.5Мбит/с]; По умолчанию: **1Мбит/с; 2Мбит/с; 5.5Мбит/с; 11Мбит/с**) – список поддерживаемых режимов, используемых во всех диапазонах **2ghz-b**, **2ghz-b/g** и **2ghz-b/g/n**. Устройства смогут установить соединение только если они используют аналогичные режимы работы. Данный параметр актуален только если параметр rate-**set** установлен как **configured**.

**tdma-period-size** (целое значение [1…10]; По умолчанию: **2**) – интервал TDMA в милисекундах. Параметр может увеличить надежность связи на больших дистанциях, но также может снизить пропускную способность радиоканала.

**tx-power** (целое значение [-30…30]) –изменение мощности передатчика (для всего канала при работе протокола 802.11ac или для каждого из каналов при работе протоколов 802.11a/b/g/n).

**tx-power-mode** (default, card-rates, all-rated-fixed, manual-table; По умолчанию: **default**) – настройка режима работы усилителя радиокарты

- **default** – использовать значения мощности, прописанные в параметрах радиокарты
- **card-rates** – использовать значения мощности**,** указанные в команде **tx-power**
- **all-rated-fixed** – использовать одинаковую мощность передатчика для всех скоростных режимов. Возможно повреждение радиокарты, если мощность будет превышать стандартное значение для текущего скоростного режима.
- **manual-table** – определять мощность передатчика отдельно для каждого скоростного режима. Возможно повреждение радиокарты, если мощность будет превышать стандартное значение для текущего скоростного режима.

**update-stats-interval** – частота обновления информации о подключенных клиентских станциях в таблице регистрации (см. стр. 138).

**vht-basic-mcs** (none | MCS 0-7 | MCS 0-8 | MCS 0-9; По умолчанию: **MCS 0-7**) – модуляция и схемы кодирования, применяемые в протоколе 802.11n, для установления соединения с базовой станцией должны поддерживаться на стороне клиентской станции

Можно установить интервал MCS для каждого пространственного потока:

- **none** – не использовать выбранный пространственный поток
- **MCS 0-7** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-7
- **MCS 0-8** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-8
- **MCS 0-9** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-9

**vht-supported-mcs** (none | MCS 0-7 | MCS 0-8 | MCS 0-9; По умолчанию: **MCS 0-9**) модуляция и схемы кодирования, применяемые в протоколе 802.11n, поддерживаемые текущим устройством.

Можно установить интервал MCS для каждого пространственного потока

- **none** – не использовать выбранный пространственный поток
- **MCS 0-7** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-7
- **MCS 0-8** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-8
- **MCS 0-9** – клиентское устройство должно поддерживать диапазон от MCS-0 до MCS-9

**wds-cost-range** (start [-end] целое значение [0...4294967295]; По умолчанию: **50-150**) – стоимость порта прозрачного моста в линке WDS, определяется автоматически в зависимости от пропускной способности канала каждые 5 секунд, если значение изменилось более чем на 10%, или если прошло более 20 секунд со времени последнего определения.

Установка данного параметра в 0 отключает автоматическое определение стоимости.

Автоматическое определение стоимости не работает для WDS-линков, настроенных вручную как порт бриджа.

**wds-default-bridge** (строка | none; По умолчанию: **none**) – при установке WDS-соединения (статус WDS *указан* как **running)**, указанный в данном параметре интерфейс будет автоматически добавлен в порт брижда. При разрыве соединения интерфейс автоматически будет удален из бриджа.

**wds-default-cost** (целое значение [0...4294967295]; По умолчанию: **100**) – стоимость порта бриджа по умолчанию

**wds-ignore-ssid** (yes | no; По умолчанию: **no**) – по умолчанию WDS-соединение между двумя базовыми станциями может быть установлено только в том случае, если они вещают на аналогичной частоте и имеют одинаковые идентификаторы беспроводной сети (SSID). При включении данного параметра значение SSID при установке соединения не будет учитываться. Данный параметр не актуален, если одно из утройств работает в режиме **station-wds** или **wds-mode** установлен как **static-mesh** или **dynamic-mesh**.

**wds-mode** (disabled | dynamic | dynamic-mesh | static | static-mesh; По умолчанию: **disabled**) – способ установки WDS-соединения (для базовых станеций и клиентских устройств, работающих в режиме **station-wds**).

- **disabled** – запрет на установление WDS-соединения.
- **static** – установление WDS-соединения для устройств, в которых вручую настроена конфигурация прозрачного моста

- **dynamic** – установление динамического WDS-соединения в том числе и для устройств, у которых не настроена конфигурация прозрачного моста. При разрыве соединения соотвтетствующие WDS-записи будут автоматически удалены.
- **mesh** – в данных режимах используются другие методы для установления связи между базовыми станциями и эти режимы не совместимы с предыдущими. В данных режимах при установлении WDS-соединения между устройствами проверяется connect-list (см. стр. 135) на предмет того, разрешено ли соединение между устройствами. Если устройство работает в режиме **station-wds,** то возможность установления соединения проверяется через access-list (см. стр. 132) и режимы mesh игнорируются.

**wireless-protocol** (802.11 | any | nstreme | nv2 | nv2-nstreme | nv2-nstreme-802.11 | unspecified; По умолчанию: **unspecified**) – протокол передачи даных через беспроводной иетерфейс

- **unspecified** – протокол указывался на старых прошивках до 5 версии
- **any** – на базовой станции включена поддержка протоколов 802.11 и Nstreme; клиентские станции ищут БС с подержкой протокола, прописанного в правилах connect-list.
- **nstreme** – включение протокола Nstreme
- **nv2 –** включение протокола Nv2
- **nv2 nstreme** – на базовой станции всегда используется протокол Nv2; клиентские станции прежде всего ищут БС с подержкой Nv2, если не находят – ищут БС с подержкой Nstreme
- **nv2 nstreme 802.11** – на базовой станции всегда используется протокол Nv2; клиентские станции ищут последовательно БС с подержкой протоколов Nv2, Nstreme, и , в заключение – 802.11.

**Обратите внимание:** протокол Nv2 не поддерживает виртуальные базоыве станции!

**wmm-support** (disabled | enabled | required; По умолчанию: **disabled**) – настройка поддержки QoS

### Подержка RTS/CTS

В беспроводных сетях, работающих по протоколу 802.11, используется специальный механизм RTS/CTS (Request To Send / Clear To Send — запрос на отправку / разрешение отправки), исключающий коллизию кадров и помогающий рещать проблему т.н. «скрытого узла». В данном случае используется два варианта защиты:

- RTS/CTS – отправитель рассылает RTS-кадр и ждет от получателя CTS-кадр. Если ответ получен, то отправитель воздерживается от отправки информации на указанное в RTS/CTS-кадрах время.
- «CTS to self» – устройство сразу отправляет получателю CTS-кадр, и он не начинает передачу в течении указанного времени. Данный способ уменьшает накладные расходы и увеличивает пропускную способность сети, но может

быть использован, если количество скрытых узлов не превышает одного, в противном случае необходимо использовать классический RTS/CTS.

Настройка варианта защиты осуществляется параметром hw-protection-mode (см. стр. 123).

Настройка размера защищаемого пакета осуществляется параметром hw-protection-threshold (см. стр. 123).

Пример: настройка защиты типа "CTS-to-self" для ВСЕХ пакетов базовой станции, вне зависимости от их размера.

```
/interface wireless> set 0 hw-protection-mode=cts-to-self hw-protection-threshold=0
```

Для включения защиты типа RTS/CTS на клиентском устройстве используйте следующую команду:

```
/interface wireless> set 0 hw-protection-mode=rts-cts hw-protection-threshold=0
```

В RouterOS есть возможность включеения протокола Nv2, базирующегося на технологии TDMA (Time Division Multiple Access — множественный доступ с разделением по времени). TDMA позволяет нескольким пользователям использовать один и тот же частотный ресурс, подключаясь к общей базовой станции в различные временнЫе интервалы (т.н. тайм-слоты).

Протокол Nv2 обладает следующими преимуществами:

- Увеличенная пропускная способность канала
- Большое количество одновременных клиентских подключений к одной БС (до 511 клиентов)
- Минимизация накладных расходов
- Отстутствие ограничения на дистанцию связи

**Обратите внимание:** протокол Nv2 не поддерживает виртуальные базовые станции.

Для увеличения надежности связи на больших дистанциях используйте параметр tdma-period-size (см. стр. 128). Чем больше дистанция – тем больше времени необходимо клиентской станции на получение пакета от БС. Увеличение данного параметра повышает вероятность гарантированной доставки пакета, но уменьшает пропускную способность радиоканала.

### Настройка списка доступа (ACL)

Уровень подменю: **/interface wireless access-list**

Список доступа используется базовой станцией для ограничения подключений клиентских устройств к БС и для контроля параметров подключений

Принцип обработки правил в списке доступа:

- Все правила проверяются последовательно
- Отключённые правила игнорируются
- Обрабатывается только **первое** совпадение
- Если совпадений не обнаружено, тот используются стандартные настройки беспроводного интерфейса
- Если удалённое устройство подпадает под указанное правило, но при этом **authentication**=*no,* то соединение установлено не будет

**Обратите внимание:** Если в правилах ACL относительно клиентского устройства, пытающегося установить связь с БС, не обнаружено ни одного совпадения, ACL для данного клиента будет проигнорирован на пряжении всего сеанса связи.

Например, если клиентский сигнал при подключении к БС равен -41db и правило выглядит следующим образом:

```
/interface wireless access-list
add authentication=no forwarding=no interface=wlan2 signal-range=..-55
```

то параметры подключения не соотвтетствуют ни одному параметру в правиле и при уменьшнии уровня сигнала до -70…-80db соединение не будет разорвано. Для того, чтобы правило стало рабочим, изменим его следующим образом:

```
/interface wireless access-list
add interface=wlan2 signal-range=-55
add authentication=no forwarding=no interface=wlan2 signal-range=..-56
```

### Описание параметров

**ap-tx-limit** (целое значение [0...4294967295]; По умолчанию: **0**) – максимальная скорость передачи от БС на указанного клиента (бит/сек.) Для отключения параметра значение необходимо выставить в «0».

**authentication** (yes | no; По умолчанию: **yes**).

- no – ассоциация с клиенторм не будет установлена
- yes – используется процедура аутентификации согласно security-profile интерфейса

**client-tx-limit** (целое значение [0...4294967295]; По умолчанию: **0**) - максимальная скорость передачи от указанного клиента на БС (бит/сек.) Для отключения параметра значение необходимо выставить в «0». Параметр доступен только в RouterOS.

**comment** (строка; По умолчанию: ) – комментарий к текущему правилу

**disabled** (yes | no; По умолчанию: **no**) – выключение текущего правила

**forwarding** (yes | no; По умолчанию: **yes**) .

- no – клиент НЕ может обмениваться трафиком с другими клиентами БС
- yes – клиент может обмениваться трафиком с другими клиентами БС

**interface** (строка | all; По умолчанию: **all**) – название интерфейса; если в качестве названия интерфейса указано «all», то правило будет использовано для всех беспроводных интерфейсов.

**mac-address** (MAC-адрес; По умолчанию: **00:00:00:00:00:00**) – правило применяется к клиенту с указанным MAC-адресом. Если значение равно 00:00:00:00:00:00, то правило применяется ко всем клиентам.

**management-protection-key** (строка; По умолчанию: "")

**private-algo** (104bit-wep | 40bit-wep | aes-ccm | none | tkip; По умолчанию: **none**) – параметр используется только для WEP-шифрования. Данный тип шифрования устарел и не рекомендуется к использованию.

**private-key** (строка; По умолчанию: "") – параметр используется только для WEP-шифрования. Данный тип шифрования устарел и не рекомендуется к использованию.

**private-pre-shared-key** (строка; По умолчанию: "") – значение ключа для режима WPA PSK.

**signal-range** (целое число..целое число; По умолчанию: **-120...120**) Правило отрабатывает, если уровень сигнала клиента находится в указанном диапазоне. Допустимый диапазон: -120...120db. Если уровень сигнала клиента находится ВНЕ указанного диапазона, соединение с клиентом будет принудительно разорвано.

**time** (время начала – время окончания, [sun,mon,tue,wed,thu,fri,sat] – время указывается в секундах, в диапазоне 0..86400 секунд, отсчёт ведется с полуночи; По умолчанию: ) – правило будет отрабатывать только если текущее время попадает в указанный в правиле временной диапазон и если текущий день недели соответствует дням, указанным в правиле

### Юстировка

Уровень подменю: **/interface wireless align**

#### Описание параметров

**active-mode** (yes | no; По умолчанию: **yes**) – отправка станцией пакетов для юстировки

**audio-max** (целое значение [-2147483648..2147483647]; По умолчанию: -20) – максимальный уровень сигнала встроенного бипера

**audio-min** (целое значение [-2147483648..2147483647]; По умолчанию: -100) минимальный уровень сигнала встроенного бипера

**audio-monitor** (MAC; По умолчанию: 00:00:00:00:00:00) – только указаные адреса будуи использоваться при аудио-мониторинге

**filter-mac** (MAC; По умолчанию: 00:00:00:00:00:00) – фильтрация MAC-адресов; только эти адреса будут отображаться в окне мониторинга

**frame-size** (целое значение [200..1500]; По умолчанию: 300) – размер фреймов, используемых при мониторинге

**frames-per-second** (целое значение [1..100]; По умолчанию: 25) – интервал передачи фреймов

**receive-all** (yes | no; По умолчанию: **no**) По умолчанию режим мониторинга будет работать только если оба беспроводных устройства находятся в режиме юстировки

**ssid-all** (yes | no; По умолчанию: **no**) – отображать в окне мониторинга все доступные SSID или только те, что прописаны в настройках беспроводных интерфейсов

#### Описание команд

**monitor** (название интерфейса) – запуск мониторинга юстировки

**test-audio** (целое значение [-2147483648..2147483647]) – тест встроенного бипера

**Список подключений**

Уровень подменю: **/interface wireless connect-list**

Список подключениий (connect-list) используется для назначения приоритета, настройки параметров безопасности при подключении клиентов к базовым станциям. Список представляет собой построчный набор правил, где каждое правило присвоено определённому беспроводному интерфейсу через параметр **interface** (в отличие от access-list, где правила могут применяться сразу ко всем интерфейсам). В правилах могут быть прописаны различные критерии, в том числе MAC-адрес удаленнного маршрутизатора, допустимые уровни сигналов и т.д.

Принцип обработки правил в списке подключений:

- Все правила проверяются построчно, сверху вниз
- Отключённые правила игнорируются
- Обрабатывается только **первое** совпадение
- Если совпадений не обнаружено, тот используются стандартные настройки беспроводного интерфейса
- Если базовая станция подпадает под указанное правило, но при этом параметр **connect**=*no,* то попытки подключения к БС не будет
- Если базовая станция подпадает под указанное правило, но при этом параметр **connect**=*yes,* то клиент будет пытаться подключиться к БС
    - Если несколько БС подпадают под указанные в списке правила и параметр **connect**=*yes,* то клиент (режим «station») будет пытаться подключиться к той БС, которая подпадает под самое верхнее правило в списке
    - Если ни одна из БС не подпадают под правила с включённым параметром **connect** (**connect**=*yes*), то возможность подключения клиента к любой доступной БС будет определяться настройками параметра **default-authentication.** Если **default-authentication**=*yes*, то клиент будет пытаться подключиться к БС с максимальным уровнем сигнала и и поддерживаемым протоколом шифрования
- Если устанавливается WDS-соединение между БС, то список подключений проверяется до установки соединения. Если ни одного соответствия в правилах списка не обнаружено, то возможность установки WDS-соединения будет определяться настройками параметра **default-authentication.**

**Описание параметров**

**area-prefix** (строка; По умолчанию: ) Правило отрабатывает только в том случае, если БС принадлежит определённой группе в беспроводной сети. Параметр доступен только в RouerOS (см. параметр area на стр. **119**).

**comment** (строка; По умолчанию: ) – короткий коментарий к правилу

**connect** (yes | no; По умолчанию: yes)

- yes – подключиться к БС при нахождении соответствий в правиле
- no – НЕ подключаться к БС при нахождении соответствий в правиле

**disabled** (yes | no; По умолчанию: no) – отключение правила

**mac-address** (MAC-адрес; По умолчанию: **00:00:00:00:00:00**) – правило актуально только для БС с указанным MAC-адресом. Адрес «00:00:00:00:00:00» соответвтвует всем базовым станциям.

**security-profile** (строка | none; По умолчанию: **none**) – название security profile, используемого при подключении к соответствующей БС. Если название не указано, то будет использован профайл, указанный в настройках интерфейса. Клиентские устройства будут проверять соответствия в правилах только для тех БС, которые поддерживают указанный **security-profile**. При работе маршрутизатора в режиме базовой станции параметр не учитывается.

**signal-range** (целое число..целое число; По умолчанию: **-120...120**) – правило отрабатывает, если уровень сигнала клиента находится в указанном диапазоне. Допустимый диапазон: -120...120db. Если уровень сигнала клиента находится ВНЕ указанного диапазона, соединение с клиентом будет принудительно разорвано.

**ssid** (строка; По умолчанию: "") – правило отрабатывает только для БС с указаным SSID. Пустое значение SSID соответвтвует всем базовым станциям. Параметр актуален только в том случае, если SSID клиента пуст, или если в параметрах беспроводного интерфейса БС установлено **wds-ignore-ssid=yes.**

**wireless-protocol** (802.11 | any | nstreme | tdma; По умолчанию: any) – правило отрабатывает только для указанных протоколов

**interface** (строка; По умолчанию: ) – название интерфейса. Правило отрабатывает только для беспроводного интерфейса, указанного в данном параметре.

**Примеры использования списка подключений**

- **Подключение клиентских станций только к определённой БС**

Установите значение параметра **default-authentication** равным «no»:

```
/interface wireless> set name=station-wlan default-authentication=no
```

Создайте правило в списке подключений, разрешающее подключение к БС с определёнными MAC-адресами. В правиле разрешите подключение к БС (**connect**=yes) и укажите название беспроводного интерейса:

```
/interface wireless connect-list add interface=station-wlan connect=yes mac-address=00:11:22:33:00:01
/interface wireless connect-list add interface=station-wlan connect=yes mac-address=00:11:22:33:00:02
```

- **Запрет подключения клиентских станций к определённой БС**

Установите значение параметра **default-authentication** равным «yes»:

```
/interface wireless> set name=station-wlan default-authentication=yes
```

Создайте правило в списке подключений, запрещающее подключение к БС с определённым MAC-адресом (**connect**=no) и укажите название беспроводного интерейса:

```
/interface wireless connect-list add interface=station-wlan connect=no mac-address=00:11:22:33:44:55
```

- **Указаниие приоритета подключений к определённым БС**

В списке подключений создайте правила для подключения определёных БС.
В правиле разрешите подключение к БС (**connect**=yes) и укажите название беспроводного интерейса.
Переместите созданные правила в верхнюю часть списка подключений – эти правила будут обрабатываться в первую очередь.

### Ручная настройка мощности сигнала на беспроводном интерфейсе

Уровень подменю: **/interface wireless manual-tx-power-table**

**Описание параметров**

**comment** (строка) – краткий комментарий

**manual-tx-powers** (список значений в формате [скорость:TxPower]; где:

- скорость = 11Mbps | 12Mbps | 18Mbps | 1Mbps | 24Mbps | ...
- TxPower = значение из диапазона [-30..30]

**name** (строка) - название беспроводного интерфейса

## Таблица регистрации клиентских станций

В данной таблице представлена различная информация о подключенных к БС клиентских устройствах. Актуально только для базовых станций.

**Описание параметров (доступны только для чтения)**

**802.1x-port-enabled** (yes | no) – разрешён ли обмен данными с партнером (завершена ли аутентификация 802.1x)

**ack-timeout** (целое значение) – текущее значение ack-timeout

**ap** (yes | no) – является ли текущее устройство базовой станцией

**ap-tx-limit** (целое значение) – лимит передаваемого БС трафика (бит/сек)

**authentication-type** () – метод аутентификации, используемый в текущем подключении

**bridge** (yes | no) – настроено ли текущее устройство прозрачным мостом

**bytes** (целое значение , целое значение) – количество принятых/отправленных байт

**client-tx-limit** (целое значение) – лимит передаваемого клиентом трафика (бит/сек)

**comment** (строка) – комментарий, отображается из соответствующего списка доступа, если он там прописан

**compression** (yes | no) – используется ли компрессия данных в текущем соединении

**distance** (целое значение) – дистанция между БС и клиентской точкой

**encryption** (aes-ccm | tkip) – используемый алгоритм шифрования

**frame-bytes** (целое значение, целое значение) – количество отправленных и полученных байт (без учета размеров заголовков пакетов)

**frames** (целое значение, целое значение) – количество пакетов, которое необходимо отправить через беспроводное соединение. Значение данного параметра обычно сравнивают со значением параметра **hw-frames** для оценки качества беспроводного соединения.

**framing-current-size** (целое значение) – текущий размер объединенного пакета

**framing-limit** (целое значение) – максимальный размер объединенного пакета

**framing-mode** () – метод объединения пакетов

**group-encryption** () – используемый групповой алгоритм шифрования

**hw-frame-bytes** (целое значение, целое значение) – количество отправленных и полученных байт (с учетом размеров заголовков пакетов)

**hw-frames** (целое значение, целое значение) – реальное количество пакетов, переданных через беспроводное соединение, с учетом всех переповторов передачи данных, если пакеты были утеряны. Значение данного параметра обычно сравнивают со значением параметра **frames** для оценки качества беспроводного соединения. Если значение **hw-frames** *значительно* превышает значение **frames**, то стоит подумать об улучшении качества связи.

**interface** (строка) – название интерфейса, с которым установлено беспроводное соединение

**last-activity** (время) – активность интерфейса (прием/передача данных)

**last-ip** (IP-адрес) – IP-адрес, полученный из крайнего пакета, полученного от клиентского устройства

**mac-address** (MAC-адрес) – MAC-адрес клиента

**management-protection** (yes | no) – используется ли режим **management protection**. В RouterOS реализован собственный алгоритм защиты беспроводного канала. Маршрутизатор может проверить источник пакета и подтвердить, что он не представляет опасности. Режим настраивается командой:

```
/interface wireless security-profiles> set default management-protection=
```

Где параметр management-protection может принимать одно из значений:

- **allowed** – режим включается, если он поддерживается и БС, и клиентом
- **disabled** – режим не используется (стандартное значение)
- **required** – связь между БС и клиентом возможна только в том случае, если обоими точками подерживается режим management-protection

**nstreme** (yes | no) – используется ли протокол **nstreme**

**p-throughput** (целое значение) – оценочная пропускная способность текущего соединения. Рассчитывается на основе скорости и качества соединения в течение 5 секунд

**packed-bytes** (целое значение, целое значение) – количество байт в объединённом пакете (framing)

**packed-frames** (целое значение, целое значение) – количество фреймов в объединённом пакете (framing)

**packets** (целое значение, целое значение) – количество отправленных и принятых пакетов

**radio-name** (строка) – название беспроводного соединения

**routeros-version** (строка) – версия RouterOS клиентского устройства

**rx-ccq** () – качество беспроводного соединения (Client Connection Quality) для входящего трафика, отображается в процентах. Рассчитывается из соотношения $T_{min}/T_{real}$, где $T_{min}$ – минимальное время, требуемое на получение пакета (без учета переповторов), и $T_{real}$ – реальное время, потраченное на получение пакета.

**rx-rate** (целое значение) – скорость получения данных

**signal-strength** (целое значение) – среднее значение уровня сигнала, полученного от клиента

**signal-to-noise** () – соотношение сигнал/шум

**strength-at-rates** () – значения уровней сигналов в зависимости от используемых скоростных режимов, а так же время, в течение которого использовался тот или иной режим

**tx-ccq** () – качество беспроводного соединения (Client Connection Quality) для исходящего трафика, отображается в процентах. Рассчитывается из соотношения $T_{min}/T_{real}$, где $T_{min}$ – минимальное время, требуемое на отправку пакета (без учета переповторов), и $T_{real}$ – реальное время, потраченное на отправку пакета.

**uptime** (время) – время, в течение которого клиент ассоциирован с БС

**wds** (yes | no) – используется ли wds при подключении клиентского устройства

**wmm-enabled** (yes | no) – используется ли WMM

### Утилита «Sniffer»

Уровень подменю: **/interface wireless sniffer**

Утилита представляет собой простой сниффер беспроводного трафика.

#### Описание параметров

**channel-time** (По умолчанию: **200мс**) - в течении какого времени перехватывать трафик каждого канала, если параметр **multiple-channels=yes**

**file-limit** (целое значение [10..4294967295]; По умолчанию: **10) –** размер файла в байтах, в который будет сохраняться перехваченный трафик. Актуально, если указано значекние параметра **file-name**

**file-name** (строка) – имя файла, в который будет сохраняться перехваченный трафик

**memory-limit** (целое значение [10..4294967295]; По умолчанию: **10**) – размер буфера в оперативной памяти в байтах, в который будет сохраняться перехваченный трафик

**multiple-channels** (yes | no; По умолчанию: **no**) – перехват трафика одного или нескольких каналов

**only-headers** (yes | no; По умолчанию: **no**) – по умолчанию перехватывается трафик только из заголовков пакетов

**receive-errors** (yes | no; По умолчанию: **no**) – перехватывать ли пакеты с неверной контрольной суммой

**streaming-enabled** (yes | no; По умолчанию: **no**) – перенаправлять ли перехваченный трафик на сервер в формате TZSP (Tazmen Sniffer Protocol)

**streaming-max-rate** (целое значение [0...4294967295]; По умолчанию: **0**) – максимальное количество пакетов в секунду, принимаемых маршрутизатором; 0 – без ограничений

**streaming-server** (IPv4; По умолчанию: **0.0.0.0**) – IP-адрес сервера для перенаправления перехваченного трафика

Посмотр информации о перехваченных пакетах:

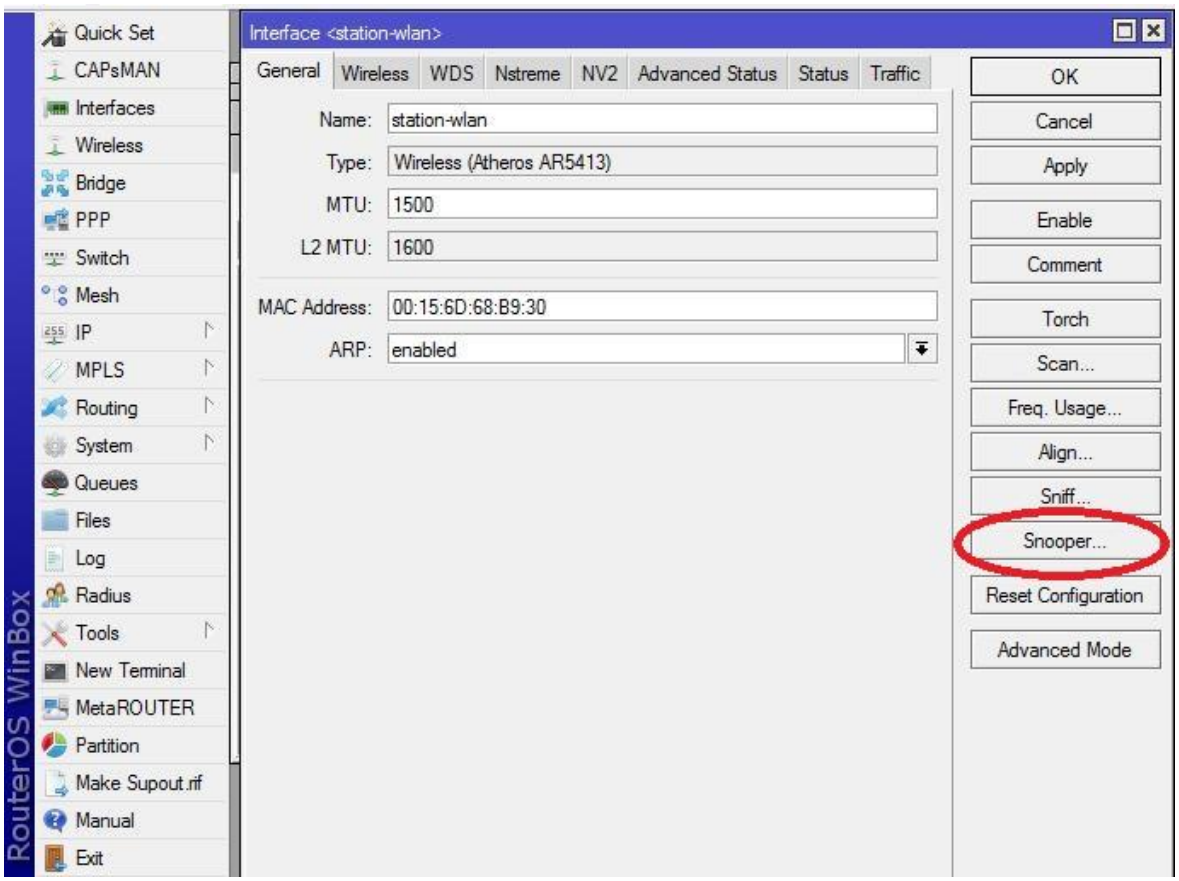
```
/interface wireless sniffer packet print
```

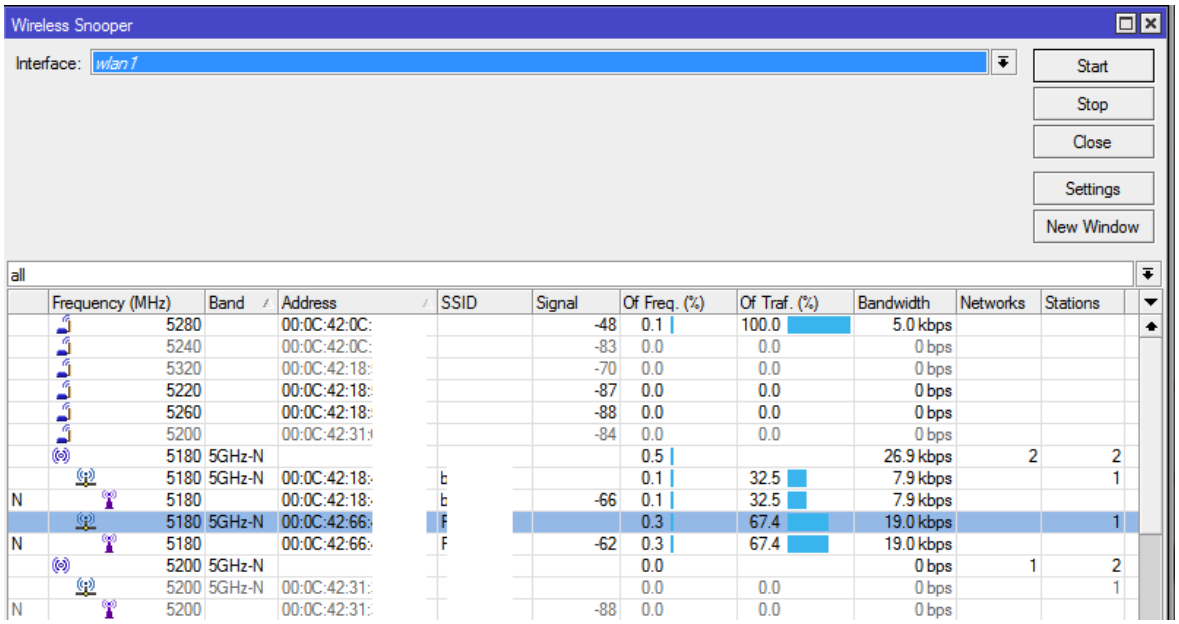
**Утилита «Snooper»**

Уровень подменю: **/interface wireless snooper**

Утилита сканирукет радиоэфир и отображает список используемых радиочастот, MAC-адреса беспроводных устройств, SSID доступных базовых станций и различную дополнительную информацию.

Для отображения списка в реальном времени рекомендуется использовать утилиту Winbox.






# Основные параметры интерфейса Ethernet

Уровень меню: **/ interface ethernet**

### Описание параметров

**arp** (disabled | enabled | proxy-arp | reply-only; По умолчанию: **enabled**) – режим использования протокола ARP:

- disabled – интерфейс не использует ARP
- enabled – интерфейс использует ARP
- proxy-arp – интерфейс использует проксирование запросов ARP
- reply-only – интерфейс будет отвечать только на запросы, которые соответствуют только статической комбинации IP-адрес/MAC-адрес из таблицы arp. Соответственно, для установления связи в таблице должны присутствовать статические значения.

**auto-negotiation** (yes | no; По умолчанию: **yes**) – функция автопереговоров, позволяет настроить максимально эффективное соединение между интерфейсами

- **Обратите внимание:** функция не может быть выключена только на одном из интерфейсов, в противном случае Ethernet-соединение может работать некорректно
- **Обратите внимание:** функция не может быть выключена на Gigabit-Ethernet

**bandwidth** (целое значение/целое значение; По умолчанию: **unlimited/unlimited**) – максимальное значение скорости приёма/передачи интерфейса. Лимит скорости передачи поддерживается всеми радиокартами, работающими на чипсете Atheros. Лимит скорости приема поддерживается только радиокартами, работающими на чипсете AR8327.

**cable-setting** (default | short | standard; По умолчанию: **default**) – указание длины используемого кабеля (актуально только для сетевых карт NS DP83815/6)

**comment** (строка) - краткое описание интерфейса

**disable-running-check** (yes | no; По умолчанию: **yes**) – отключение проверки активности интерфейса. Если параметр отключён, маршрутизатор автоматически определяет, есть ли подключения к данному интерфейсу. По умолчанию проверка отключена для совместимости с устаревшими сетевыми картами.

**tx-flow-control** (yes | no | auto) – приостановка отправки пакетов при заполнении буфера. Значение **auto** работает так же как **on**, если только не установлен параметр **auto-negotiation=yes**. Параметр актуален для чипсетов AR724x, AR9xxx, QCA9xxx, CCR и Atheros.

**rx-flow-control** (yes | no | auto) – приостановка приёма пакетов. Значение **auto** работает так же как **on**, если только не установлен параметр **auto-negotiation=yes**. Параметр актуален для чипсетов AR724x, AR9xxx, QCA9xxx, CCR и Atheros.

**full-duplex** (yes | no; По умолчанию: **yes**) – возможность одновременной работы интерфейса на приём и передачу данных

**l2mtu** (целое значение [0...65536]; По умолчанию: **1600**) – максимальный размер пакета без учета размера MAC-заголовка, который может быть отправлен через указанный интерфейс

**mac-address** (MAC-адрес) – MAC-адрес проводного интерфейса

**master-port** (название; По умолчанию: **none**) – использовать ли данный интерфейс в качестве мастер-порта

**mdix-enable** (yes | no; По умолчанию: **yes**) – поддержка MDI/X, позволяющая работать с любым кабелем, и обычным, и кроссоверным

**mtu** (целое значение [0...65536]; По умолчанию: **1500**) – размер пакета в байтах

**name** (строка) – название интерфейса

**orig-mac-address** (MAC)

**poe-out** (auto-on | forced-on | off; По умолчанию: **off**) – определение возможности использования технологии PoE (Power over Ethernet) на порту

- **auto-on** – устройство автоматически пытается определить возможность использования технологии PoE исходя из значения сопротивления на неиспользуемой паре жил кабеля (синий и коричневый провода). Для использования PoE сопротивление должно быть в диапазоне 3kΩ - 26.5kΩ. Данный параметр используется по умолчанию;
- **forced-on** – автоматическое определение не используется. PoE включено в принудительном порядке;
- **off** - автоматическое определение и PoE отключено на данном порту

**poe-priority** () – установка приоритета порта относительно других портов **poe-out.** Максимальный приоритет – «0». Минимальный приоритет – «99». Если несколько портов имеют одинаковый приоритет, то используется порт с наименьшим номером. Если на порту из-за низкого приоритета была отключена технология PoE, то каждые 6 секунд на нм будет проверяться возможность использования PoE посредством **poe-out**

**sfp-rate-select** (high | low; По умолчанию: **high**)

**speed** (10Mbps | 10Gbps | 100Mbps | 1Gbps) – установка скорости передачи данных через интерфейс. По умолчанию используется максимальное значение.

**Описание параметров (доступны только для чтения)**

**running** (yes | no) – определение активности интерфейса. Некоторые устаревшие интерфейсы не поддерживают данный режим и их статус всегда обозначается как "running"

**rx-1024-1518** (целое значение) – количество полученных пакетов размером 1024 - 1518 байт

**rx-128-255** (целое значение) – количество полученных пакетов размером 128 - 255 байт

**rx-1519-max** (целое значение) – количество полученных пакетов размером более чем 1519 байт

**rx-256-511** (целое значение) – количество полученных пакетов размером 256 - 511 байт

**rx-512-1023** (целое значение) – количество полученных пакетов размером 512 - 1023 байт

**rx-64** (целое значение) – количество полученных пакетов размером 64 байт

**rx-65-127** (целое значение) – количество полученных пакетов размером 65 - 127 байт

**rx-align-error** (целое значение) – количество полученных сообщений об ошибках

**rx-broadcast** (целое значение) – количество полученных широковещательных пакетов

**rx-bytes** (целое значение) – количество полученных байт

**rx-fcs-error** (целое значение) – количество полученных пакетов с неправильной контрольной суммой

**rx-fragment** (целое значение) – количество полученных фрагментированных пакетов

**rx-multicast** (целое значение) – количество полученных мультикастовых пакетов

**rx-overflow** (целое значение) – количество переполнений буфера при получении пакетов

**rx-pause** (целое значение) – количество полученных pause frames

**rx-runt** (целое значение) – количество полученных пакетов размером менее 64 байт, но с корректной контрольной суммой

**rx-too-long** (целое значение) – количество полученных пакетов размером более максимального

**slave** (yes | no) – определяет, является ли интерфейс ведомым

**switch** (целое значение) – ID свича, к которому подключён интерфейс.

**tx-1024-1518** (целое значение) – количество переданных пакетов размером 1024 to 1518 байт

**tx-128-255** (целое значение) – количество переданных пакетов размером 128 to 255 байт

**tx-1519-max** (целое значение) – количество переданных пакетов размером более 1519 bytes

**tx-256-511** (целое значение) – количество переданных пакетов размером 256 to 511 байт

**tx-512-1023** (целое значение) – количество переданных пакетов размером 512 to 1023 байт

**tx-64** (целое значение) – количество переданных пакетов размером 64 байта

**tx-65-127** (целое значение) – количество переданных пакетов размером 65 to 127 байт

**tx-align-error** (целое значение) – количество переданных сообщений об ошибках

**tx-broadcast** (целое значение) – количество переданных широковещательных пакетов

**tx-bytes** (целое значение) – количество переданных байт

**tx-fcs-error** (целое значение) – количество переданных пакетов с неправильной контрольной суммой

**tx-fragment** (целое значение) – количество переданных фрагментированных пакетов

**tx-multicast** (целое значение) – количество переданных мультикастовых пакетов

**tx-overflow** (целое значение) – количество переполнений буфера при передаче пакетов

**tx-pause** (целое значение) – количество переданных кадров паузы

**tx-runt** (целое значение) – количество переданных пакетов размером менее 64 байт, но с корректной контрольной суммой

**tx-too-long** (целое значение) – количество переданных пакетов размером более максимального

### Описание команд

**blink** ([id, name]) – мигание светодиодом Ethernet

**monitor** ([id, name]) – запуск мониторинга интерфейса (см. раздел ниже)

**reset-counters** ([id, name]) – сброс статистики интерфейса (см. стр. 150)

**reset-mac-address** ([id, name]) – сброс MAC-адреса в заводские настройки

**cable-test** (строка) – диагностика кабеля (см. стр.149 )

## Мониторинг проводного интерфейса

Уровень подменю: **/interface ethernet monitor**

Команда отображает текущее состояние проводного интерфейса

### Описание параметров

**auto-negotiation** (done | incomplete) – статус функции автопереговоров:

- **done** – автопереговоры завершены
- **incomplete** – автопереговоры закончились неудачей или еще не закончены

**default-cable-settings** (short | standard) – настройка стандартной длины кабеля (актуально только для сетевых карт NS DP83815/6)

- **short** – поддерживаются короткие кабели
- **standard** – поддерживаются стандартные кабели

**full-duplex** (yes | no) – поддерживается ли двунаправленная передача данных

**rate** (10Mbps | 100Mbps | 1Gbps) – текущая скорость соединения

**status** (link-ok | no-link | unknown) – текущий статус сетевого интерфейса

- **link-ok** – устройство подключено к сети
- **no-link** – устройство не подключено к сети
- **unknown** – статус подключения не распознан (если сетевая карта не сообщает статус соединения)

**tx-flow-control** () – используется ли TX flow control

**rx-flow-control** () – используется ли RX flow control

**sfp-module-present** (yes | no) – установлен ли модуль SFP

**sfp-rx-lose** (yes | no)

**sfp-tx-fault** (yes | no)

**sfp-connector-type** (строка) – тип SFP-коннектора

**sfp-link-length-copper** (строка) – определение длины линка при использовании медного модуля SFP

**sfp-vendor-name** (строка) – производитель модуля SFP

**sfp-vendor-part-number** (строка) – артикул модуля SFP

**sfp-vendor-revision** (строка) – номер версии модуля SFP

**sfp-vendor-serial** (строка) – серийный номер модуля SFP

**sfp-manufacturing-date** (строка) – дата производства модуля SFP

**eeprom** () – EEPROM модуля SFP

**Пример1**:

```
interface ethernet> monitor ether1
        status: link-ok
 auto-negotiation: done
          rate: 1Gbps
     full-duplex: yes
```

**Пример2**:

```
/interface ethernet> monitor sfp1
                  name: sfp1
                status: link-ok
       auto-negotiation: done
                  rate: 1Gbps
             full-duplex: yes
          tx-flow-control: no
          rx-flow-control: no
     sfp-module-present: yes
           sfp-rx-lose: no
          sfp-tx-fault: no
     sfp-connector-type: optical-pigtail
sfp-link-length-copper: 1m
        sfp-vendor-name: OEM
sfp-vendor-part-number: SFP-10G-CU1M
    sfp-vendor-revision: A0
     sfp-vendor-serial: E1309250082
sfp-manufacturing-date: 13-10-10
               eeprom: 0000: 03 04 21 00 00 00 00 00  04 00 00 00 67 00 00 >
                       0010: 00 00 01 00 4f 45 4d 20  20 20 20 20 20 20 20 >
                       0020: 20 20 20 20 00 00 40 20  53 46 50 2d 31 30 47 >
                       0030: 43 55 31 4d 20 20 20 20  41 30 20 20 00 00 00 >
                       0040: 00 00 00 00 45 31 33 30  39 32 35 30 30 38 32 >
                       0050: 20 20 20 20 31 33 31 30  31 30 20 20 00 00 00 >
```

### Диагностика Ethernet-кабеля

В прошивке RouterOS версии 6rc4 и более поздних появилась возможность тестирования проводного соединения и выявления различных проблем, в том числе:

- Определять какие пары проводов повреждены
- Определять дистанцию до места повреждения
- Определять тип повреждения кабеля (короткое замыкание или обрыв)

Если противоположный конец кабеля не подключен к оборудованию, то будет показана суммарная длина кабеля.

Тестирование возможно для следующих типов устройств (а также для других устройств, работающих на аналогичных чипсетах): **SXT-G**, **SXT Lite**, **RB711G**, **RB2011**, **RB750**

**Пример**:

```
interface ethernet cable-test ether2
       name: ether2
     status: no-link
cable-pairs: open:4,open:4,open:4,open:4
```

В данном примере обрыв всех 4 пар находится на расстоянии порядка 4 метров от маршрутизатора.

### Просмотр суммарной статистики проводного интерфейса

Уровень подменю: **/interface ethernet print stats**

Данная команда отображает суммарную статистику проводного интерфейса. Не все сетевые интерфейсы поддерживают указанные здесь параметры, актуально только для RB450G ether2-ether5, RB750 ether2-ether5, RB750G ether1-ether5 и RB1100 ether1-ether10. Полное описание отображаемых в списке параметров доступно выше (см. стр. 145).

**Пример статистики для RB450G:**

```
/interface ethernet> print stats
                 name: ether1-gateway ether2-local ether3-local ether4-local ether5-local
          rx-broadcast:              22          31          3666          11
            rx-pause:               0           0             0           0
          rx-multicast:               4           7          1423           5
          rx-fcs-error:               0           0             2           0
        rx-align-error:               0           0             0           0
             rx-runt:                 0           0             0           0
         rx-fragment:                 0           0             1           0
               rx-64:                 0           0             0           0
           rx-65-127:                 8          14         21598          10
          rx-128-255:                 0           0             0           0
          rx-256-511:                18          24          2245           6
         rx-512-1023:             28926        7649        371938       24476
        rx-1024-1518:                 0           0             0           0
         rx-1519-max:                 0           0             0           0
         rx-too-long:                 0           0             0           0
         rx-overflow:                 0           0             0           0
            rx-bytes:          15337844     4063737     199738064    12975401
          tx-broadcast:              13          13          1496           8
            tx-pause:               0           0             0           0
          tx-multicast:              13          13          1496           8
         tx-underrun:                 0           0             0           0
               tx-64:                 0           0             0           0
           tx-65-127:                26          26          2992          16
          tx-128-255:                 0           0             0           0
          tx-256-511:                 0           0             0           0
         tx-512-1023:                 0           0             0           0
        tx-1024-1518:                 0           0             0           0
         tx-1519-max:                 0           0             0           0
         tx-too-long:                 0           0             0           0
         tx-collision:                0           0             0           0
   tx-excessive-collision:            0           0             0           0
   tx-multiple-collision:             0           0             0           0
    tx-single-collision:              0           0             0           0
   tx-excessive-deferred:             0           0             0           0
          tx-deferred:                0           0             0           0
     tx-late-collision:               0           0             0           0
            tx-bytes:             2561        2561        294712        1576
```

# Описание скриптового языка

### Спецификация

Требуемые пакеты: **system**

Уровень подменю: **/system script**

Стандарты и технологии: Нет

Использование аппаратных средств: Не существенно

Раздел содержит введение во встроенный в RouterOS скриптовый язык.

Скрипты предоставляют возможность автоматизировать некоторые задачи по обслуживанию маршрутизатора путём выполнения определенных сценариев при наступлении указанного события.

Скрипты могут быть записаны в хранилище скриптов, либо могут быть введены непосредственно в консоли. Существует множество событий, используемых для запуска скрипта, включая события встроенного планировщика (см. стр. 169), утилиты Netwatch (см. стр. 171) и утилиты мониторинга трафика (см. стр. 174).

### Синтаксис консольных команд

Консольные команды состоят из следующих частей:

**[prefix] [path] command [uparam] [param=[value]] .. [param=[value]]**

- **[prefix]** – символ ":" или "/», с которых начинаются команды, например, :put или /ping 10.0.0.1. Необязательная часть команды.
- **[path]** – относительный путь к необходимому уровню меню. Необязательная часть команды.
- command – одна из команд, доступных на том или ином уровне меню
- **[uparam]** – неименованный параметр команды
- **[params]** – последовательность именованных параметров, вводимых в формате: параметр=значение

В конце команды может быть поставлен необязательный символ «;», обычно используемый в качестве разделителя, если несколько команд вводятся

последовательно в одной строке. Одиночные команды внутри круглых, квадратных или фигурных скобок также не требуют наличия символа разделителя, например:

```
:if ( true ) do={ :put "halt!" }
```

Команды, вводимые внутри другой команды, должны быть ограничены квадратными скобками, например:

```
:put [/ip route get [find gateway=1.0.0.1]];
```

Обратите внимание, что указанная выше команда содержит 3 команды:

- :put
- /ip route get
- find gateway=1.0.0.1

Длинные команды можно вводить в несколько строк: в этом случае строки должны быть разделены символом обратного слеша, например:

```
:if ($a = true \
    and $b=false) do={ :put “$a $b”; }
```

### Формат строк скрипта

Строки скрипта представляют собой последовательность символов, оканчивающихся символом конца строки (EOL). Поддерживаются стандартные форматы конца строки:

- **unix** – ASCII LF;
- **windows** – ASCII CR LF;
- **mac** – ASCII CR;

Так же поддерживается C-соглашение для обозначения новой строки (символ \n)

### Комментарии

Комментарии начинаются с символа решётки и продолжаются до конца строки. Другие символы перед символом комментария не допускаются. Знак решетки, заключённый в кавычки, не рассматривается как комментарий:

```
:global myStr "lala # this is not a comment":put $myStr
```

### Объединение строк

Как уже писалось выше, длинные команды можно вводить в несколько строк, разделив их символом обратного слеша (\). После символа слеша не может стоять символ комментария (#), и наоборот, например:

```
:if ($a = true \
     and $b=false) do={ :put “$a $b”; }

:if ($a = true \      # bad comment
     and $b=false) do={ :put “$a $b”; }

# comment \
   continued – invalid  (syntax error)
```

### Пробелы между частями команды

Пробелы могут быть использованы для отделения части команд друг от друга, пробел обязателен лишь в том случае, если при его отсутствии части команды могут быть интерпретированы как другая команда, например:

```
{
  :local a true; :local b false;
# whitespace is not required
  :put (a&&b);
# whitespace is required
  :put (a and b);
}
```

Пробелы не допускаются между:

- Параметром и знаком равенства – '<parameter>='
- Ключевым словом и знаком равенства – 'from=' 'to=' 'step=' 'in=' 'do=' 'else='

Например:

```
#incorrect:
:for i from = 1 to = 2 do = { :put $i }
#correct syntax:
:for i from=1 to=2 do={ :put $i }
:for i from= 1 to= 2 do={ :put $i }

#incorrect
/ip route add gateway = 3.3.3.3
#correct
/ip route add gateway=3.3.3.3
```

### Области видимости переменных

#### Глобальная область видимости

Стандартная область видимости скрипта. Создается автоматически и не может быть отключена. Переменная, созданная в глобальной области, доступна как из любой части текущего скрипта, так и из других скриптов.

#### Локальная область видимости

Пользователь может определять собственные локальные области для оганичения видимости переменных. Локальные области разграничиваются фигурными скобками:

```
{
  :local a 3;
  {
    :local b 4;
    :put ($a+$b);
  }
#line below will generate error
  :put ($a+$b);
}
```

Здесь переменная **«b»** объявлена в собственной локальной области и не будет доступна за её пределами.

**Обратите внимание:** каждая строка, введенная в консоли, рассматривается как локальная область.

Например, объявленная локальная переменная myVAR не отобразится командой put:

```
:local myVAR test
:put $myVar
```

**ВНИМАНИЕ:** не объявляйте глобальные переменные внутри локальной области

#### Ключевые слова

Следующие слова являются ключевыми и не могут использоваться как переменные или имена функций:

- and
- or
- not
- in

**Разделители**

Следующие символы рассматриваются в качестве разделителей:

- ()
- []
- {}
- :
- ;
- $
- /

**Типы данных**

В скриптах RouterOS существуют следующие типы данных:

**num** (число) - 64-битное целое число, возможно указание в шестнадцатеричном виде

**bool** (boolean) - логическое значение, может быть true или false

**str** (строка) - последовательность символов

**ip** - IPv4-адрес

**ip6-prefix** - IPv6-адрес

**id** (внутренний ID) - шестнадцатеричное значение, начинающееся с символа '*'. Каждый элемент меню имеет свой уникальный внутренний идентификатор

**time** - значение даты и времени

**array** - последовательность значений, организованная в виде массива

**nil** - тип переменной по умолчанию, пока переменной не присвоено какое-либо значение

**Escape-последовательности**

Следующие сочетания могут быть использованы для ввода специальных символов:

| **Сочетание** | **Символ** |
|---|---|
| **\"** | Двойная кавычка |
| \\ | Обратный слеш |
| **\n** | Новая строка |

| | |
|---|---|
| **\r** | Возврат каретки |
| **\t** | Табуляция (горизонтальная) |
| **\$** | Символ «$» |
| **\?** | Символ «?» |
| **_** | Пробел |
| **\a** | Бипер (0x07) |
| **\b** | Возврат каретки (0x08) |
| **\f** | Пропуск строки (0xFF) |
| **\v** | Табуляция (вертикальная) |
| **\xx** | Символ из hex-таблицы, вводится в верхнем регистре |

Пример:

```
:put "\48\45\4C\4C\4F\r\nThis\r\nis\r\na\r\n\ftest"
HELLO
This
is
a

test
```

**Операторы**

| **Арифметические операторы** | **Описание** | **Пример** |
|---|---|---|
| "+" | сложение | :put (3+4); |
| "-" | вычитание | :put (1-6); |
| "*" | умножение | :put (4*5); |
| "/" | деление | :put (10 / 2); :put ((10)/2) |
| "-" | смена знака | { :local a 1; :put (-a); } |
| **Операторы сравнения** | **Описание** | **Пример** |
| "<" | меньше | :put (3<4); |
| ">" | больше | :put (3>4); |
| "=" | равно | :put (2=2); |
| "<=" | меньше или равно | |
| ">=" | больше или равно | |
| "!=" | не равно | |

| **Логические операторы** | **Описание** | **Пример** |
|---|---|---|
| ***“!”, “not”*** | отрицание | `:put (!true);` |
| ***“&&”, “and”*** | И | `:put (true&&true)` |
| ***“\|\|”, “or”*** | ИЛИ | `:put (true\|\|false);` |
| ***“in”*** | | `:put (1.1.1.1/32 in 1.0.0.0/8);` |
| **Битовые операции** | **Описание** | **Пример** |
| ***“~”*** | НЕ | `:put (~0.0.0.0)` |
| ***“\|”*** | Побитовое ИЛИ | |
| ***“^”*** | Исключающее ИЛИ (XOR) | |
| ***“&”*** | Побитовое И | |
| ***“<<”*** | Сдвиг влево | |
| ***“>>”*** | Сдвиг вправо | |
| **Конкатенация** | **Описание** | **Пример** |
| ***“.”*** | Конкатенация строк | `:put (“concatenate” . “ “ . “строка”);` |
| ***“,”*** | Конкатенация массивов или добавление нового элемента в массив | `:put ({1;2;3} , 5 );` |

Возможна конкатенация и без указанных операторов, например:

```
:global myVar "world";
:put ("Hello " . $myVar);
```

В следующем примере будет тот же результат:

```
:put "Hello $myVar";
```

Используя конструкции $[] и $(), возможно добавление выражений непосредственно в строку, например:

```
:local a 5;
:local b 6;
:put " 5x6 = $($a * $b)";

:put " We have $[ :len [/ip route find] ] routes";
```

| **Другие операторы** | **Описание** | **Пример** |
|---|---|---|
| ***“[]”*** | Подстановка команды. Выражение может занимать только одну командную строку | `:put [ :len "my test строка"; ]` |
| ***“()”*** | Подвыражение или группировка | `:put ( "value is " . (4+5));` |
| ***“$”*** | Подстановка значения переменной | `:global a 5; :put $a;` |

| | | |
|---|---|---|
| “~” | Соответствие значения указанному регулярному выражению, введенному в формате POSIX | Отображаем все маршруты со шлюзом, оканчивающимся на 202<br>`/ip route print where \ gateway~"^[0-9 \\.]*202"` |
| “->” | Получение значения элемента массива по его ключу | `global aaa {a=1;b=2}`<br>`:put ($aaa->"a")`<br>`1`<br>`:put ($aaa->"b")`<br>`2` |

**Переменные**

В скриптах доступно 2 типа переменных:

- Глобальные – доступные из любых скриптов текущего пользователя, при создании переменной указывается ключевое слово «**global».**
- Локальные – доступны только из текущей области видимости, при создании переменной указывается ключевое слово «**local».**

Любая переменная, за исключением переменных, встроенных в RouterOS, должна быть объявлена с соответствующим ключевым словом. Начиная с RouterOS версии 6.2 парсер автоматически попытается определить тип переменной, если она не была объявлена как global или local, например, при настройке DHCP lease-script:

```
/system script
add name=myLeaseScript policy=\
    ftp,reboot,read,write,policy,test,winbox,password,sniff,sensitive,api \
    source=":log info \$leaseActIP\r\
    \n:log info \$leaseActMAC\r\
    \n:log info \$leaseServerName\r\
    \n:log info \$leaseBound"

/ip dhcp-server set  myServer lease-script=myLeaseScript
```

В именах переменных обычно используются буквы и цифры, если имя содержит какие-либо другие символы, то его необходимо заключать в двойные кавычки:

```
#valid variable name
:local myVar;
#invalid variable name
:local my-var;
#valid because double quoted
:global "my-var";
```

Если переменная объявлена, но не инициализирована конкретным значением, то тип переменной устанавливается в nil. Тип инициализированной переменной

определяется автоматически. Если необходимо изменить тип переменной, то это можно сделать соответствующими командами (см. стр. 162), например:

```
#convert строка to array
:local myStr "1,2,3,4,5";
:put [:typeof $myStr];
:local myArr [:toarray $myStr];
:put [:typeof $myArr]
```

**Обратите внимание**: Имена переменных чувствительны к регистру:

```
:local myVar "hello"
# following line will generate error, because variable myVAr is not defined
:put $myVAr
# correct code
:put $myVar
```

Для удаления ранее объявленной переменой из окружения используйте команду **set** (в RouterOS версии 6.2 и выше):

```
:global myVar "myValue"
:put $myVar
myValue
:set myVar
:put $myVar
```

**Команды**

**Глобальные команды**

Глобальные команды должны начинаться с префикса «:», в противном случае они будут трактоваться как переменные.

| Команда | Синтаксис | Описание | Пример |
|---|---|---|---|
| / | | Переход в корневое меню | |
| .. | | Переход на предыдущий уровень меню | |
| ? | | Просмотр перечня команд меню с их кратким описанием | |

| | | | |
|---|---|---|---|
| ***global*** | `:global <переменная> [<значение>]` | Объявление глобальной переменной | **:global myVar "something";**<br>**:put $myVar;** |
| ***local*** | `:local <переменная> [<значение>]` | Объявление локальной переменной | **{ :local myLocalVar "I am local";**<br>**:put $myVar; }** |
| ***beep*** | `:beep <частота> <продолжительность>` | Подача сигнала на встроенный бипер | |
| ***delay*** | `:delay <сек>` | Задержка выполнения скрипта на указанное количество секунд | |
| ***Put*** | `:put <выражение>` | Вывод аргумента в консоль | |
| ***Len*** | `:len <выражение>` | Определение длины строки или количества элементов массива | **:put [:len "length=8"];** |
| ***typeof*** | `:typeof <переменная>` | Определение типа указанной переменной | **:put [:typeof 4];** |
| ***Pick*** | `:pick <переменная> <начало>[<конец>]` | Возвращает диапазон элементов или значение подстроки, если последний арумент не указан, возвращает только указанное начальное значение диапазона | **:put [:pick "abcde" 1 3]** |
| ***Log*** | `:log <уровень> <сообщение>` | Запись сообщения в системный журнал. Уровни детализации: `debug, error, info` и `warning` | **:log info "Hello from script";** |
| ***time*** | `:time <выражение>` | Возвращает временной интервал, | **:put [:time {:for i from=1 to=10 do={ :delay 100ms }}];** |

| | | | |
|---|---|---|---|
| | | необходимый для выполнения команды | |
| ***set*** | `:set <переменная> [<значение>]` | Присваивает переменной указанное значение | **:global a;**<br>**:set a true;** |
| ***find*** | `:find <аргумент> <аргумент> <начало>` | Возвращает позицию подстроки в строке или позицию элемента массива | **:put [:find "abc" "a" -1];** |
| ***environment*** | `:environment print <start>` | Вывод списка проинициализированных переменных | **:global myVar true;**<br>**:environment print;** |
| ***terminal*** | | Использование терминальных команд | |
| ***error*** | `:error <выражение>` | В консоли инициируется исключение с последующим прерыванием выполнения скрипта | |
| ***execute*** | `:execute <выражение>` | Выполнение скрипта в фоновом режиме | **:local j [:execute {/interface print follow where [:log info ~Sname~]}];**<br>**:delay 10s;**<br>**:do { /system script job remove Sj } on-error={}** |
| ***parse*** | `:parse <выражение>` | Разбирает строку на команды и выполняет полученные команды. Может быть использована в качестве функции. | **:global myFunc [:parse ":put hello!"];**<br>**$myFunc;** |
| ***resolve*** | `:resolve <аргумент>` | Возвращает IP-адрес указанного DNS | **:put [:resolve "nporapira.ru"];** |

| **Команды изменения типа переменной** | | | |
|---|---|---|---|
| ***toarray*** | `:toarray <переменная>` | Конвертация переменной в массив | |
| ***tobool*** | `:tobool <переменная>` | Конвертация переменной в логическое значение | |
| ***toid*** | `:toid <переменная>` | Конвертация переменной во внутренний ID | |
| ***toip*** | `:toip <переменная>` | Конвертация переменной в IP-адрес | |
| ***toip6*** | `:toip6 <переменная>` | Конвертация переменной в IPv6-адрес | |
| ***tonum*** | `:tonum <переменная>` | Конвертация переменной в целое значение | |
| ***tostr*** | `:tostr <переменная>` | Конвертация переменной в строку | |
| ***totime*** | `:totime <переменная>` | Конвертация переменной в значение времени | |
| **Основные команды меню**<br>Данные команды доступны на большинстве уровней меню | | | |
| **Команда** | **Синтаксис** | **Описание** | |
| ***add*** | `add <параметр>=<значение> ..<параметр>=<значение>` | Добавление нового элемента | |
| ***remove*** | `remove <id>` | Удаление указанного элемента | |
| ***enable*** | `enable <id>` | Включение указанного элемента | |
| ***disable*** | `disable <id>` | Выключение указанного элемента | |
| ***set*** | `set <id> <параметр>=<значение> ..<параметр>=<значение>` | Изменение параметра элемента, можно указывать сразу несколько параметров. Значение параметра может быть обнулено при помощи префикса «!»<br>**/ip firewall filter add chain=blah action=accept protocol=tcp port=123 nth=4,2** | |

| | | |
|---|---|---|
| | | **/ip firewall filter print**<br>**0 chain=blah action=accept protocol=tcp port=123 nth=4,2**<br><br>**set 0 !port chain=blah2 !nth protocol=udp**<br>**/ip firewall filter print**<br>**0 chain=blah2 action=accept protocol=udp** |
| ***get*** | `get <id> <параметр>=<значение>` | Получение значения параметра указанного элемента |
| ***print*** | `print <параметр><параметр>= [<значение>]` | Отображает информацию, доступную в текущем уровне меню. Вывод информации зависит от указанных параметров. см. «Основные параметры команды print» на стр. 28. |
| ***export*** | `export [file=<значение>]` | Экспорт настроек текущего меню и всех подменю (если таковые имеются) в текущую консоль. Если указан параметр **file**, то экспорт будет перенаправлен в указанный файл с расширением '.rsc'. Обратный импорт осуществляется при помощи команды **import** из корневого меню. |
| ***edit*** | `edit <id> <параметр>` | Редактирование параметров указанных элементов во встроенном редакторе. |
| ***find*** | `find <выражение>` | Возвращение номеров всех элементов, соответствующих указанному выражению<br>**:put [/interface find name~"ether"]** |
| **Циклы** | | |
| **Команда** | **Синтаксис** | **Описание** |
| ***do..while*** | `:do { <команды> } while=( <условия> );`<br>`:while ( <условия> ) do={ <команды> };` | Выполнение команд, пока есть соответствие заданному условию |
| ***for*** | `:for <var> from=<int> to=<int> step=<int> do={ <команды> }` | Выполнение команд указанное количество раз |
| ***foreach*** | `:foreach <var> in=<array> do={ <команды> };` | Выполнение команд над каждым элементом списка |
| **Условия** | | |

**Функции**

До версии 6.2 нельзя было создать функцию напрямую, но можно было воспользоваться командой :**parse** в качестве альтернативного метода.

Начиная с версии 6.2 возможно более простое создание функций с указанием необходимых параметров.

```
#define function and run it
:global myFunc do={:put "hello from function"}
$myFunc

output:
hello from function
```

```
#pass arguments to the function
:global myFunc do={:put "arg a=$a"; :put "arg '1'=$1"}
$myFunc a="this is arg a value"  "this is arg1 value"

output:
arg a=this is arg a value
arg '1'=this is arg1 value
```

Возвращаемое функцией значение задается командой :return, например:

```
:global myFunc do={ :return ($a + $b)}
:put [$myFunc a=6 b=2]

output:
8
```

Возможно сохранение существующего скрипта под указанным именем и дальнейшее использование этого имени как функции, например:

```
#add script
 /system script add name=myScript source=":put \"Hello $myVar !\""

:global myFunc [:parse [/system script get myScript source]]
$myFunc myVar=world

output:
Hello world !
```

**Обратите внимание**: Если функция содержит ранее определенную глобальную переменную, имя которой **совпадает** с именем передаваемого параметра, то такая переменная будет проигнорирована (для обратной совместимости со скриптами, написанных для старых версий прошивок), поэтому **избегайте совпадения имён параметров функции и глобальных переменных**.

Пример того, как не надо делать:

```
:global my2 "123"

:global myFunc do={ :global my2; :put $my2; :set my2 "lala"; :put $my2 }
$myFunc my2=1234
:put "global value $my2"
#Output will be:
1234
lala
global value 123
```

### Использование вложенных функций

Для вызова функции из другой функции она должна быть предварительно объявлена, например:

```
:global funcA do={ :return 5 }
:global funcB do={
  :global funcA;
  :return ([$funcA] + 4)
}
:put [$funcB]


Output:
9
```

### Обработка ошибок времени выполнения

С версии 6.2 появилась возможность обрабатывать ошибки времени выполнения.

Например, в результате работы команды **:resolve** возникнет ошибка и выполнение скрипта будет остановлено:

```
{ :put [:resolve www.my_example.com]; :put "lala";}
failure: dns name does not exist
```

Теперь добавим обработчик, в результате чего работа скрипта не будет прервана:

```
:do {
     :put [:resolve www.my_example.com];
} on-error={ :put "resolver failed"};
:put "lala"

output:

resolver failed
lala
```

### Работа с массивами

**Обратите внимание:** Если у элемента массива есть ключ, содержащий символы в верхнем регистре, то такой ключ должен быть заключён в кавычки:

```
{:local a { "aX"=1 ; ay=2 }; :put ($a->"aX")}
```

### Доступ к ключам и значениям элементов массива

Для доступа к элементам массива используйте команду **:foreach**, например:

```
:foreach k,v in={2; y=2; "aX"=1 ; 5} do={:put ("$k=$v")}

0=2
1=5
aX=1
y=2
```

**Обратите внимание:** Если у элемента масива есть ключ, то эти элементы выводятся в алфавитном порядке, порядок вывода элементов без ключей не меняется (см. пример выше).

### Изменение значения элемента массива

```
:global a {x=1; y=2}
:set ($a->"x") 5
:environment print
a={x=5; y=2}
```

### Хранилище скриптов

Уровень подменю: **/system script**

Хранилище содержит все скрипты, введеные пользователем.
Скрипт может быть выполнен несколькими способами, а именно:

- По событию – автоматическое выполнение скрипта при наступлении некоего события (см. планировщик на стр. 169, утилиту netwatch на стр. 171, протокол VRRP)
- Из другого скрипта
- Вручную, указав имя или ID скрипта в качестве параметра команды **run**

**Описание параметров**

**name** (строка) – название скрипта

**policy** (*строка*) – используемые политики

- **api** – доступ к api
- **ftp** – удалённое подключение через ftp, отправка/получение файлов
- **local** – локальное подключение через консоль
- **password** – смена паролей
- **policy** – управление политиками пользователей, удаление/добавление пользователей
- **read** – просмотр настроек
- **reboot** – перезагрузка маршрутизатора
- **sensitive** – просмотр паролей и другой скрытой информации
- **sniff** – запуск sniffer, torch и т.д.
- **ssh** – удаленное подключение по SSH
- **telnet** – удаленное подключение через telnet
- **test** – запуск тестов: ping, трассировка, пропускная способность
- **web** – удалённое подключение по http
- **winbox** – удаленное подключение через winbox
- **write** – чтение/запись настроек маршрутизатора

**source** (строка) – исходный код скрипта

**Описание параметров (доступны только для чтения)**

**last-started** (дата) – дата и время последнего запуска

**owner** (строка) – пользователь, создавший скрипт

**run-count** (целое значение) – количество запусков скрипта

**Команды**

**run** (id|name) – запуск скрипта, указанного по названию или ID

**Окружение**

Уровень подменю: **/system script environment**
Уровень подменю: **/environment**

Содержит созданные пользователем переменные и их значения, например:

```
:global example;
:set example 123
/environment print
"example"=123
```

**Описание параметров (доступны только для чтения)**

**name** (строка) – имя переменной

**user** (строка) – пользователь, создавший переменную

**value** () – значение переменной

**Задачи**

Уровень подменю: **/job**

Содержит перечень исполняемых в данный момент скриптов.

**Описание параметров (доступны только для чтения)**

**owner** (строка) – пользователь, запустивший скрипт

**policy** (массив) – перечень политик, применённых к данному скрипту

**started** (дата) – локальные дата и время запуска скрипта

# Планировщик

### Спецификация

Уровень подменю: /**system scheduler**

Стандарты и технологии: нет

Аппаратное обеспечение: не существенно

### Описание

Планировщик обеспечивает возможность:

- Исполнять сценарии однократно в назначенное время
- Исполнять сценарии периодически через указанный интервал времени

### Описание параметров

**interval** (время; по умолчанию 0) – временной интервал между двумя исполнениями скрипта, если значение установлено в ноль, то скрипт будет исполнен только в назначенное время, в противном случае выполнение будет периодически повторятся через указанный временной интервал.

**name** (название) – название задачи

**on-event** (название) – название исполняемого скрипта, скрипт уже должен существовать в */system script*

**run-count** (только для чтения; целое значение) – используется для контроля выполнения скрипта, счетчик увеличивается на единицу каждый раз при выполнении скрипта.

**start-date** (дата) – дата первого запуска скрипта

**start-time** (время) – время первого запуска сценария

- **startup** – выполнение скрипта при включении маршрутизатора, параметр **interval** должен быть равне нулю, в противном случае скрипт не будет исполнен при старте системы

### Обратите внимание:

- Перезагрузка маршрутизатора сбрасывает значение счётчика **run-count**.

- Если несколько скриптов должны быть выполнены в одно и то же указанное время, то они будут выполняться именно в том порядке, в каком они прописаны в настройках планировщика. Это может иметь принципиальное значение, если одни скрипт отключает другой. Порядок исполнения скриптов может быть изменен при помощи команды **move**.
- Если необходимо обеспечить более гибкое выполнение скриптов, то в планировщике можно прописать несколько скриптов, включающих/ отключающих другие скрипты.

### Примеры

Добавим в планировщик задачу, запускающую скрипт log-test каждый час

```
system script> add name=log-test source=:log message=test
system script> print
   0 name="log-test" source=":log messgae=test" owner=admin run-count=0
system script> .. scheduler
system scheduler> add name=run-1h interval=1h on-event=log-test
system scheduler> print

Flags: X – disabled
 #   NAME     ON-EVENT  START-DATE  START-TIME INTERVAL     RUN-COUNT
 0   run-1h    log-test               mar/30/2004 06:11:35             1h                        0
```

В следующем примере добавим два скрипта, которые будут изменять параметры пропускной способности очереди "Cust0". Каждый день в 9 часов утра очередь будет установлена в 64 кб/сек, а в 5 часов вечера очередь будет установлена в 128 кб/сек.

```
queue simple> add name=Cust0 interface=ether1  dst-address=192.168.0.0/24 limit-at=64000
queue simple> print
Flags: X – disabled, I – invalid
 0   name="Cust0" target-address=0.0.0.0/0 dst-address=192.168.0.0/24
     interface=ether1 limit-at=64000 queue=default priority=8 bounded=yes

queue simple> /system script
system script> add name=start_limit source={/queue simple set  Cust0 limit-at=64000}
system script> add name=stop_limit source={/queue simple set Cust0 limit-at=128000}
system script> print
 0 name="start_limit" source="/queue simple set Cust0 limit-at=64000"
   owner=admin run-count=0

 1 name="stop_limit" source="/queue simple set Cust0 limit-at=128000"
   owner=admin run-count=0

system script> .. scheduler
system scheduler> add interval=24h name="set-64k"  start-time=9:00:00 on-event=start_limit
system scheduler> add interval=24h name="set-128k" start-time=17:00:00 on-event=stop_limit
system scheduler> print
Flags: X – disabled
 #   NAME     ON-EVENT START-DATE  START-TIME INTERVAL          RUN-COUNT
 0   set-64k   start...              oct/30/2008 09:00:00              1d              0
 1   set-128k  stop_...              oct/30/2008 17:00:00              1d              0
```

В следующем примере планировщик каждую неделю отправляет резервную копию конфигурации маршрутизатора на указанный e-mail.

```
system script> add name=e-backup source={/system backup
{... save name=email; /tool e-mail send to="root@host.com" subject=([/system
{... identity get name] . " Backup") file=email.backup}
system script> print
 0 name="e-backup" source="/system backup save name=ema... owner=admin
   run-count=0

system script> .. scheduler
system scheduler> add interval=7d name="email-backup"  on-event=e-backup
system scheduler> print
Flags: X – disabled
 #   NAME      ON-EVENT START-DATE  START-TIME INTERVAL          RUN-COUNT
 0   email-... e-backup             oct/30/2008              15:19:28       7d              1
```

Не забудьте предварительно настроить параметры электронной почты: в установках e-mail прописать SMTP-сервер и указать адрес отправителя, например:

```
tool e-mail> set server=159.148.147.198 from=SysAdmin@host.com
tool e-mail> print
   server: 159.148.147.198
     from: SysAdmin@host.com
```

Пример ниже каждый час с полуночи до полудня помещает в системный журнал записи об условном скрипте «x».

```
system script> add name=enable-x source={/system scheduler enable x}
system script> add name=disable-x source={/system scheduler disable x}
system script> add name=log-x source={:log message=x}

system script> .. scheduler
system scheduler> add name=x-up start-time=00:00:00 interval=24h on-event=enable-x
system scheduler> add name=x-down start-time=12:00:00 interval=24h on-event=disable-x
system scheduler> add name=x start-time=00:00:00 interval=1h on-event=log-x
system scheduler> print
Flags: X – disabled
 #   NAME      ON-EVENT START-DATE  START-TIME INTERVAL          RUN-COUNT
 0   x-up      enable-x             oct/30/2008  00:00:00             1d                  0
 1   x-down    disable-x            oct/30/2008  12:00:00             1d                  0
 2   x         log-x                oct/30/2008  00:00:00             1h                  0
```

# Мониторинг сети

Требуемые пакеты: **advanced-tools**

Уровень лицензии: Level1

Уровень подменю: **/tool netwatch**

Стандарты и технологии: none

Аппаратные требования: не существенно

Утилита **netwatch** контролирует состояние хостов в сети, отправляя пинги (ICMP-пакеты) указанным в списке хостам. Для каждого элемента в списке наблюдения вы можете определить IP-адрес, интервал между пингами и название скрипта. Основное назначение утилиты – выполнение соответствующего скрипта при изменении состояния хоста.

### Описание параметров

**down-script** (строка) – название скрипта, который будет выполнен однократно, если состояние хоста будет определено как **down**

**host** (IP-адрес; по умолчанию: **0.0.0.0**) – IP-адрес контролируемого хоста

**interval** (время; по умолчанию: **1сек**) – время между пингами. Уменьшение этого интервала увеличивает частоту опроса хоста, но может создать избыточный трафик и увеличить потребление системных ресурсов.

**timeout** (время; по умолчанию: **1сек**) – таймаут для каждого пинга, по превышению которого хост считается недоступным (**down**)

**up-script** (строка) – название скрипта, который будет выполнен однократно, если состояние хоста будет определено как **up**

### Просмотр статуса хоста

Просмотр осуществляется командой **/tool netwatch print**. Команда отображает доступные только для чтения параметры:

**since** (время) – время, когда в последний раз менялось состояние хоста.

**status**(up | down| unknown) – текущий статус хоста.

- **up –** хост доступен в сети
- **down –** хост не доступен в сети
- **unknown** – возможное состояние хоста после изменения всех параметров элемента в списке наблюдения или при включении/отключении элемента

### Примеры

Следующий пример запускает скрипты gw_1 или gw_2, которые меняют адрес шлюза по умолчанию в зависимости от состояния одного из шлюзов:

```
system script> add name=gw_1 source={/ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.1}
system script> add name=gw_2 source={/ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.217}

system script> /tool netwatch
tool netwatch> add host=10.0.0.217 interval=10s timeout=998ms up-script=gw_2 down-script=gw_1

tool netwatch> print
Flags: X – disabled
 #  HOST            TIMEOUT          INTERVAL          STATUS
 0  10.0.0.217           997ms       10s               up

tool netwatch> print detail
Flags: X – disabled
 0   host=10.0.0.217 timeout=997ms interval=10s since=feb/27/2003 14:01:03
     status=up up-script=gw_2 down-script=gw_1
```

Рассмотрим подробнее предыдущий пример: скрипт gw_2 исполняется однократно, если хост становится доступен. Скрипт эквивалентен следующей команде:

```
/ip route set [find dst-address="0.0.0.0/0"] gateway=10.0.0.217
```

Команда **find** возвращает список всех маршрутов, где в качестве адреса получателя указано 0.0.0.0/0 – такой маршрут обычно является маршрутом по умолчанию. Полученное значение подставляется в команду **/ip route set**, которая меняет шлюз маршрута на значение 10.0.0.217
Похожее действие выполняет и скрипт gw_1, с тем лишь различием, что он выполняется в случае, когда хост становится недоступен.

Утилита **netwatch** может использоваться и без скриптов, отображая состояние соединений.

Ещё один пример скрипта, отправляющего уведомление на электронный адрес каждый раз, когда хост 10.0.0.215 становится недоступен:

```
system script> add name=e-down source={/tool e-mail send
{... from="support@mt.lv" server="159.148.147.198" body="Router down"
{... subject="Router at second floor is down" to="user@example.com"}
system script> add name=e-up source={/tool e-mail send
{... from="support@mt.lv" server="159.148.147.198" body="Router up"
{.. subject="Router at second floor is up" to="user@example.com"}

system script> /tool netwatch
tool netwatch> add host=10.0.0.215 timeout=999ms interval=20s up-script=e-up down-script=e-down

tool netwatch> print detail
Flags: X – disabled
 0   host=10.0.0.215 timeout=998ms interval=20s since=feb/27/2003 14:15:36
     status=up up-script=e-up down-script=e-down
```

# Мониторинг трафика

### Спецификация

Требуемые пакеты: **advanced-tools**

Уровень подменю: **/tool traffic monitor**

Стандарты и технологии: нет

Аппаратные требования: не существенно

Утилита используется для автоматического запуска скриптов, когда объем трафика на интерфейсе превысит указанное пороговое значение.
Команда утилиты состоит из названия (по которому удобно обратиться из другого скрипта, если необходимо отключить команду или изменить её параметры), группы параметров, задающих условия запуска скрипта и указатель на скрипт или задачу планировщика, которые должны быть запущены при наступлении заданных условий.

### Описание параметров

**interface** (название) – контролируемый интерфейс

**name** (строка) – имя элемента трафик монитора

**on-event** (строка) – название скрипта

**threshold** (целое значение; по умолчанию: **1000000**) – пороговое значение

**traffic** (переданный | принятый; по умолчанию:**transmitted**) – тип контролируемого трафика

- **transmitted –** переданный трафик
- **received –** полученный трафик

**triger** (above | below | always; По умолчанию: **above**) – условие выполнения скрипта

- **above –** скрипт будет запущен при превышении трафиком указанного порога
- **below –** скрипт будет запущен, если трафик будет меньше указанного порога
- **always –** скрипт будет запущен при выполнении любого из первых двух условий

**Пример:**

В данном примере утилита включает интерфейс ether2, если принятый трафик превышает 15 кб/сек на интерфейсе ether1 и отключает его, если принятый трафик падает ниже 12 кб/сек на интерфейсе ether1:

```
system script> add name=eth-down source="/interface disable ether2"
system script> add name=eth-up source="/interface enable ether2"

/tool traffic-monitor
/tool traffic-monitor> add disabled=no interface=ether1 \
name=turn_on on-event=eth-up threshold=15000 traffic=received trigger=above
/tool traffic-monitor> add disabled=no interface=ether1 \
name=turn_off on-event=eth-down threshold=12000 traffic=received trigger=below

/tool traffic-monitor> print
Flags: X – disabled, I – invalid
 #   NAME       INTERFACE      TRAFFIC    TRIGGER THRESHOLD  ON-EVENT
 0   turn_on    ether1         received   above   15000      eth-up
 1   turn_off   ether1         received   below   12000      eth-down
```