IPv4 ? IPv6

- Основы IPv4 и IPv6
- IP-адресация
- IP-пулы
- IP-маршрутизация
- Настройки IP

?????? IPv4 ? IPv6

?????? ?????

Компьютерные сети состоят из множества компонентов и протоколов, работающих совместно. Чтобы понять, как происходит связь между узлами, познакомимся с моделью OSI и моделью TCP/IP. Обе модели помогают визуализировать коммуникацию между узлами.

?????? OSI

Модель открытых систем OSI состоит из 7 уровней и сегодня используется как учебное пособие. Изначально она была задумана как стандартная архитектура для построения сетевых систем, но реальные сети гораздо менее строго структурированы.

- **Уровень 7 (Прикладной)** — протокол, определяющий связь между сервером и клиентом, например, HTTP. Если браузер хочет загрузить изображение, протокол организует и выполняет запрос;
- **Уровень 6 (Представления)** — обеспечивает получение данных в удобном формате. Здесь может происходить шифрование (например, IPSec);
- **Уровень 5 (Сеансовый)** — отвечает за установку, управление и завершение сеансов между клиентом и сервером;
- **Уровень 4 (Транспортный)** — отвечает за сборку и разборку потока данных, делит поток на сегменты с порядковыми номерами и инкапсулирует в протокольный заголовок (TCP, UDP и др.);
- **Уровень 3 (Сетевой)** — отвечает за логическую адресацию устройств, данные инкапсулируются в IP-заголовок и называются "пакетами";
- **Уровень 2 (Канальный)** — данные инкапсулируются в собственный заголовок, например Ethernet 802.3 или беспроводной 802.11, называемый "фреймом", отвечает за управление потоком данных;
- **Уровень 1 (Физический)** — средства передачи данных в виде битов, электрические сигналы и аппаратные интерфейсы;

?????? TCP/IP

Модель TCP/IP выполняет ту же функцию, что и OSI, но лучше подходит для современных сетевых задач. В отличие от 7 уровней OSI, она содержит 4 уровня:

- **Прикладной (4)** — объединяет прикладной, представления и сеансовый уровни OSI, упрощая диагностику;
- **Транспортный (3)** — аналогичен транспортному уровню OSI (протоколы TCP, UDP);
- **Интернет (2)** — аналог сетевого уровня OSI (включая протоколы ARP, IP);
- **Канальный (1)** — также называется уровнем сетевого доступа, включает физический и канальный уровни OSI, отвечает за физическую передачу данных

между узлами;

TCP/IP	Модель OSI	Протоколы
Прикладной уровень	Прикладной уровень	DNS, DHCP, HTTP, SSH и др.
—	Уровень представления	JPEG, MPEG, PICT и др.
—	Сеансовый уровень	PAP, SCP, ZIP и др.
Транспортный уровень	Транспортный уровень	TCP, UDP
Интернет уровень	Сетевой уровень	ICMP, IGMP, IPv4, IPv6, IPSec
Канальный уровень	Канальный уровень	ARP, CDP, MPLS, PPP и др.
Физический уровень	Физический уровень	Bluetooth, Ethernet, Wi-Fi и др.

Ethernet

Самым распространенным протоколом канального уровня (уровень 2 модели OSI) в компьютерных сетях является протокол Ethernet. Каждый узел в сети имеет уникальный адрес, называемый MAC-адресом (Media Access Control), иногда его называют "Ethernet-адресом".

MAC-адрес состоит из 48 бит и обычно задан производителем (изменить нельзя), но в последнее время широко используется настройка пользовательских MAC-адресов. RouterOS позволяет задать кастомный MAC-адрес.

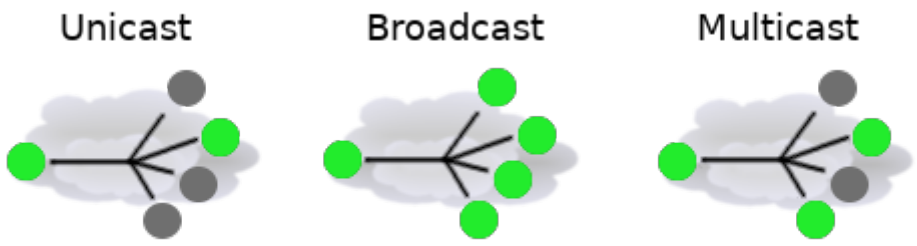
Чаще всего MAC-адрес записывается в виде 6 шестнадцатеричных чисел, разделённых двоеточиями, например `D4:CA:6D:01:22:96`.

В RouterOS MAC-адрес отображается в конфигурации для всех шиноподобных интерфейсов (Ethernet, Wireless, 60G, VPLS и пр.).

```
[admin@rack1_b32_CCR1036] /interface ethernet> print
Flags: X - disabled, R - running, S - slave
# NAME    MTU MAC-ADDRESS          ARP SWITCH
0 R ether1 1500 D4:CA:6D:01:22:96 enabled
1 R ether2 1500 D4:CA:6D:01:22:97 enabled
2 R ether3 1500 D4:CA:6D:01:22:98 enabled
3 ether4 1500 D4:CA:6D:01:22:99 enabled
4 ether5 1500 D4:CA:6D:01:22:9A enabled
5 ether6 1500 D4:CA:6D:01:22:9B enabled
6 ether7 1500 D4:CA:6D:01:22:9C enabled
7 R ether8 1500 D4:CA:6D:01:22:9D enabled
8 sfp-sfpplus1 1500 D4:CA:6D:01:22:94 enabled
9 sfp-sfpplus2 1500 D4:CA:6D:01:22:95 enabled
```

???? MAC-???????

- **Unicast** — адрес, отправляется всем узлам в области коллизии (например, кабелю между двумя узлами или всем приёмникам в беспроводной сети). Только узел с совпадающим MAC принимает фрейм (если не включен режим promiscuous).
- **Broadcast** — адрес `FF:FF:FF:FF:FF:FF`, принят и перенаправлен всеми узлами в сети второго уровня.
- **Multicast** — адрес, принимаемый всеми узлами, настроенными на приём сообщений с данным адресом.



IP-??????? ??????????

Протокол Ethernet подходит для передачи данных между двумя узлами в сети Ethernet, но не используется самостоятельно. Для доступа третьего уровня (сетевого) применяется протокол IP для уникальной адресации хостов.

В большинстве современных сетей используются IPv4-адреса, 32-битные, записанные в десятичном формате с точками, например, `192.168.88.1`.

Сетей может быть несколько логических, чтобы определить принадлежность IP адреса к сети, используется маска сети (netmask). Маска задается числом бит, определяющих подсеть, либо десятичной записью, например, 24-битовая маска — `255.255.255.0`.

Рассмотрим адрес `192.168.3.24/24`:

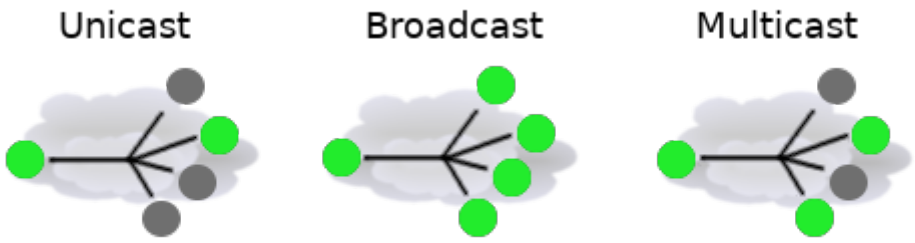
```
11000000 10101000 00000011 00011000  => 192.168.3.24
11111111 11111111 11111111 00000000  => /24 или 255.255.255.0
```

Как видно, старшие 24 бита маскированы, оставляя диапазон адресов от 0 до 255.

Адрес с первым значением в диапазоне используется для идентификации сети (здесь 192.168.3.0), а последний — для широковещательной передачи по сети (здесь 192.168.3.255). Это оставляет диапазон 1..254 для адресации хостов — unicast-адреса.

??????????? IPv4-??????

- **Broadcast** — адрес для рассылки данных сразу всем в сети, например `255.255.255.255` для локального широковещания, или направленное широковещание к адресу сети;
- **Multicast** — адреса из диапазона `224.0.0.0` до `239.255.255.255` для групповых сообщений. Отправитель посылает один пакет на адрес группы, роутеры копируют его для всех подписавшихся участников;



В логической IP-сети unicast, broadcast и multicast визуализируются по-другому.

Существуют зарезервированные адреса, например для частных сетей, которые используются только в локальной сети и обычно не маршрутизируются в Интернет:

- 10.0.0.0/8 — 10.0.0.0 до 10.255.255.255
- 172.16.0.0/12 — 172.16.0.0 до 172.31.255.255
- 192.168.0.0/16 — 192.168.0.0 до 192.168.255.255

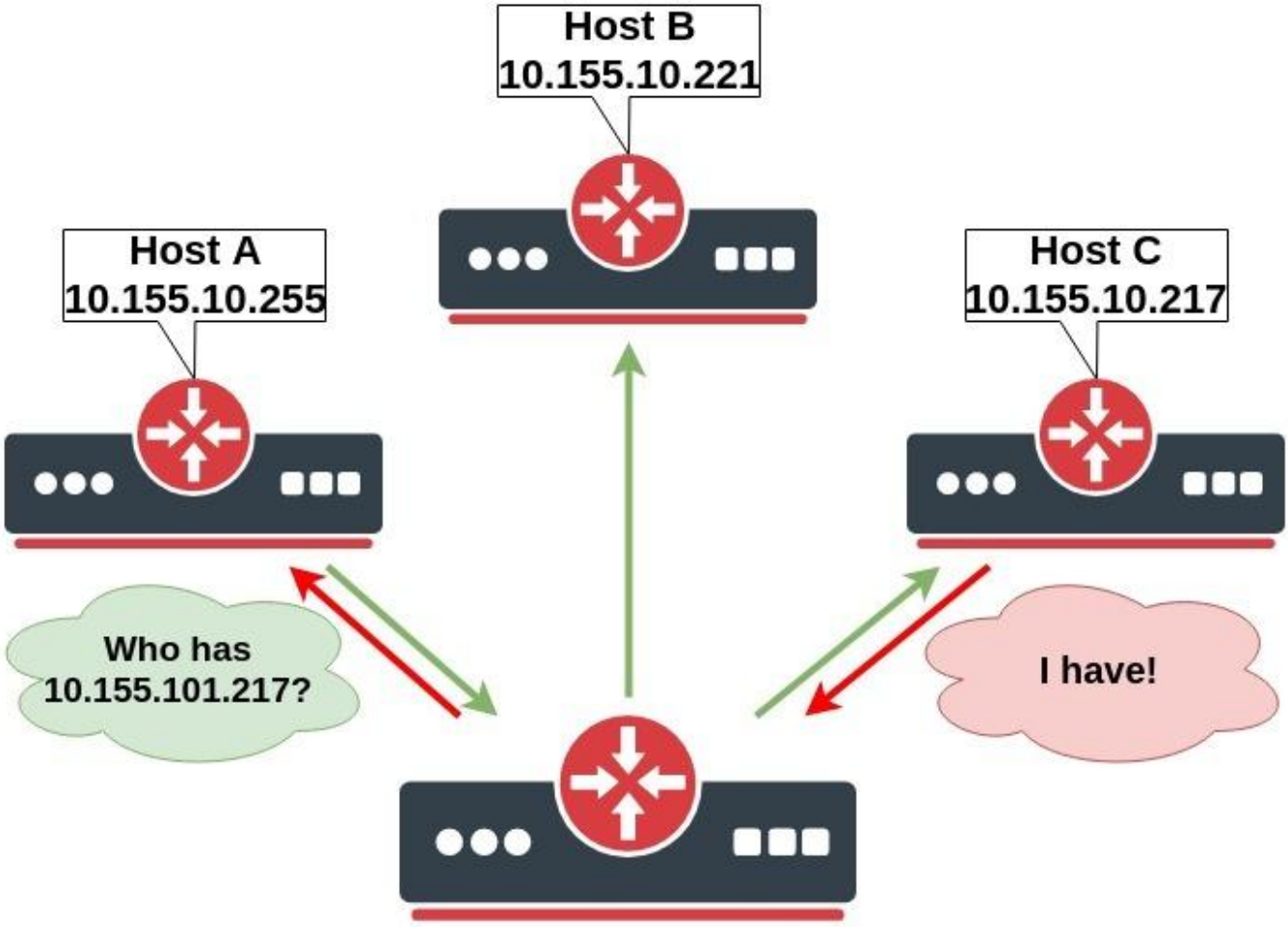
ARP ? ????? ???????

Хотя IP-пакеты адресуются IP-адресами, для физической передачи данных между хостами используются аппаратные адреса (MAC). Для сопоставления IP и MAC применяют протокол ARP (Address Resolution Protocol), описанный в RFC 826.

Каждое устройство хранит таблицу ARP. Обычно она формируется динамически, но может быть частично или полностью статической для повышения безопасности.

IPv6 избавляет от необходимости ARP.

При отправке пакета в локальной сети хост проверяет таблицу ARP для поиска MAC получателя. Если MAC отсутствует, посылается широковещательный ARP-запрос, чтобы получить MAC по IP. Хост с данным IP отвечает своим MAC.



?????? ????????? ? ARP

Добавим IP-адреса на хосты A, B и C:

```
/ip address add address=10.155.101.225 interface=ether1  (Host A)
/ip address add address=10.155.101.221 interface=ether1  (Host B)
/ip address add address=10.155.101.217 interface=ether1  (Host C)
```

Запустим сниффер пакетов и пинг с Host A к Host C:

```
/tool sniffer set file-name=arp.pcap filter-interface=ether1 start
/ping 10.155.101.217 count=1
/stop
```

Скачайте arp.pcap файл и откройте в Wireshark для анализа:

PcsCompu_85:69:b5	Broadcast	ARP	42 Who has 10.155.101.217? Tell 10.155.101.225
PcsCompu_3c:79:3a	PcsCompu_85:69:b5	ARP	42 10.155.101.217 is at 08:00:27:3c:79:3a
10.155.101.225	10.155.101.217	ICMP	70 Echo (ping) request id=0x1c01, seq=0/0, ttl=255 (reply in 428)
10.155.101.217	10.155.101.225	ICMP	70 Echo (ping) reply id=0x1c01, seq=0/0, ttl=64 (request in 427)

- Host A посылает ARP-запрос, кто владеет 10.155.101.217]

- Host C отвечает своим MAC-адресом]
- Оба хоста обновляют ARP таблицы, теперь пинг успешно проходит]

В RouterOS ARP таблицу можно просмотреть командой `/ip arp print`]

```
[admin@host_a] /ip arp> print
Flags: X - disabled, I - invalid, H - DHCP, D - dynamic, P - published, C - complete
# ADDRESS         MAC-ADDRESS          INTERFACE
0 DC 10.155.101.217 08:00:27:3C:79:3A ether1
```

?????? ARP

По умолчанию ARP включён на интерфейсах (Enabled), динамические записи добавляются автоматически.

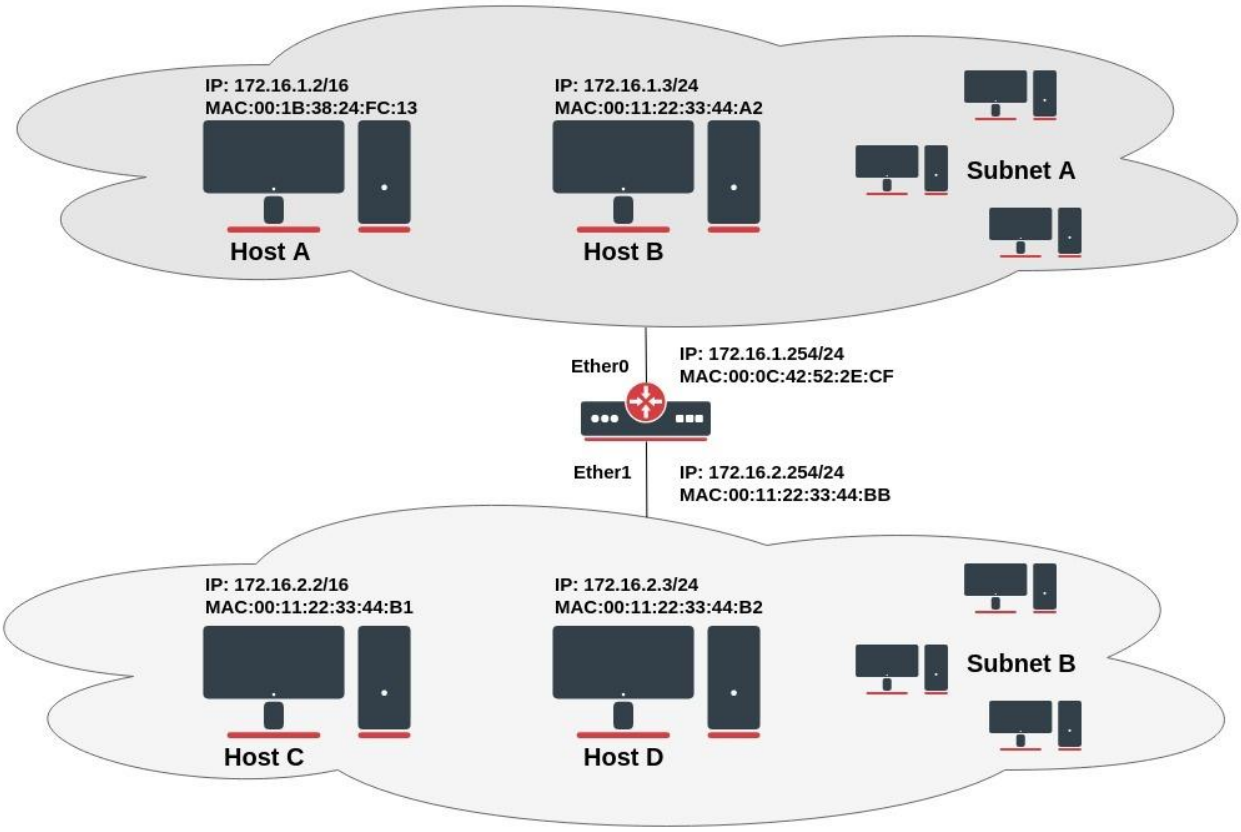
Если ARP отключён (arp=disabled), роутер не отвечает на ARP-запросы клиентов, требуется настройка статической ARP записи на клиентах. Например:

```
/ip arp add mac-address=08:00:27:3C:79:3A address=10.155.101.217 interface=ether1
```

Режим reply-only означает, что роутер отвечает только на ARP-запросы. MAC соседей нужно задавать статически (через /ip arp), но не требуется добавлять MAC адрес роутера в таблицы других хостов как при отключённом ARP.

Proxy ARP

Корректно настроенный proxy ARP позволяет роутеру работать прозрачным ARP прокси между напрямую подключёнными сетями. Это позволяет, например, назначать клиентам dial-in (PPP, PPPoE, PPTP) IP адреса из той же подсети, что и LAN.



Рассмотрим пример настройки на изображении выше. Хост A (172.16.1.2) в подсети A хочет отправить пакеты хосту D (172.16.2.3) в подсети B.
У хоста A маска подсети /16, что означает, что он считает себя напрямую подключённым ко всей сети 172.16.0.0/16 (то есть к одной и той же локальной сети).
Поскольку хост A считает, что адрес назначения находится в той же сети, он отправляет ARP-запрос, чтобы определить MAC-адрес хоста D.
(Если бы хост A понял, что IP-адрес назначения не принадлежит его подсети, он бы отправил пакет на шлюз по умолчанию.)
Хост A рассылает ARP-запрос в подсети A.

Информация из анализатора пакетов:

```
No.     Time       Source             Destination       Protocol  Info
12   5.133205  00:1b:38:24:fc:13  ff:ff:ff:ff:ff:ff  ARP      Who has 173.16.2.3?  Tell
173.16.1.2
```

Детали пакета:

```
Ethernet II, Src: (00:1b:38:24:fc:13), Dst: (ff:ff:ff:ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Source: (00:1b:38:24:fc:13)
```

```
    Type: ARP (0x0806)
Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    [Is gratuitous: False]
    Sender MAC address: 00:1b:38:24:fc:13
    Sender IP address: 173.16.1.2
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 173.16.2.3
```

С этим ARP-запросом хост A (172.16.1.2) запрашивает у хоста D (172.16.2.3) его MAC-адрес. ARP-запрос инкапсулируется в Ethernet-кадр, где MAC-адрес хоста A используется как адрес источника, а адрес назначения — широковещательный (FF:FF:FF:FF:FF:FF). Широковещательная передача на канальном уровне (Layer 2 broadcast) означает, что кадр будет отправлен всем устройствам в том же домене широковещательной рассылки уровня 2, включая интерфейс ether0 маршрутизатора, но не достигнет хоста D, так как маршрутизатор по умолчанию не пересылает широковещательные кадры уровня 2.

Поскольку маршрутизатор знает, что целевой адрес (172.16.2.3) находится в другой подсети, но может достичь хоста D, он отвечает своему хосту A собственным MAC-адресом.

```
No.     Time       Source             Destination        Protocol   Info
13   5.133378  00:0c:42:52:2e:cf   00:1b:38:24:fc:13    ARP        172.16.2.3 is at
00:0c:42:52:2e:cf
```

Детали пакета:

```
Ethernet II, Src: 00:0c:42:52:2e:cf, Dst: 00:1b:38:24:fc:13
   Destination: 00:1b:38:24:fc:13
   Source: 00:0c:42:52:2e:cf
   Type: ARP (0x0806)
Address Resolution Protocol (reply)
   Hardware type: Ethernet (0x0001)
   Protocol type: IP (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: reply (0x0002)
   [Is gratuitous: False]
```

```
   Sender MAC address: 00:0c:42:52:2e:cf
   Sender IP address: 172.16.1.254
   Target MAC address: 00:1b:38:24:fc:13
   Target IP address: 172.16.1.2
```

Это ответ Proxy ARP, который маршрутизатор отправляет хосту A. Маршрутизатор посылает обратно unicast-ответ ARP со своим MAC-адресом в качестве источника и MAC-адресом хоста A в качестве назначения, как бы говоря: «Отправляй пакеты мне, а я доставлю их туда, куда нужно».

Когда хост A получает ARP-ответ, он обновляет свою ARP-таблицу, как показано ниже:

```
C:\Users\And>arp -a
Interface: 173.16.2.1 --- 0x8
  Internet Address      Physical Address      Type
  173.16.1.254          00-0c-42-52-2e-cf    dynamic
  173.16.2.3            00-0c-42-52-2e-cf    dynamic
  173.16.2.2            00-0c-42-52-2e-cf    dynamic
```

После обновления ARP-таблицы хост A пересылает все пакеты, предназначенные для хоста D (172.16.2.3), напрямую на интерфейс маршрутизатора ether0 (00:0c:42:52:2e:cf), а маршрутизатор передаёт их хосту D. Кэш ARP на хостах в подсети A содержит MAC-адрес маршрутизатора для всех хостов подсети B, поэтому все пакеты, адресованные подсети B, отправляются маршрутизатору, который пересылает их соответствующим узлам в подсети B.

При использовании Proxy ARP несколько IP-адресов разных хостов могут быть сопоставлены с одним MAC-адресом — MAC-адресом маршрутизатора.

Proxy ARP можно включить на каждом интерфейсе отдельно с помощью команды:

```
[admin@MikroTik] /interface ethernet> set 1 arp=proxy-arp
[admin@MikroTik] /interface ethernet> print
Flags: X - disabled, R - running
  #    NAME                 MTU   MAC-ADDRESS         ARP
  0  R ether1               1500  00:30:4F:0B:7B:C1 enabled
  1  R ether2               1500  00:30:4F:06:62:12 proxy-arp
[admin@MikroTik] interface ethernet>
```

Local Proxy ARP

Если свойство `arp` на интерфейсе установлено в `local-proxy-arp`, маршрутизатор выполняет

Proxy ARP только для трафика, приходящего и уходящего через этот же интерфейс. В обычной локальной сети два хоста обмениваются данными напрямую, без участия маршрутизатора.

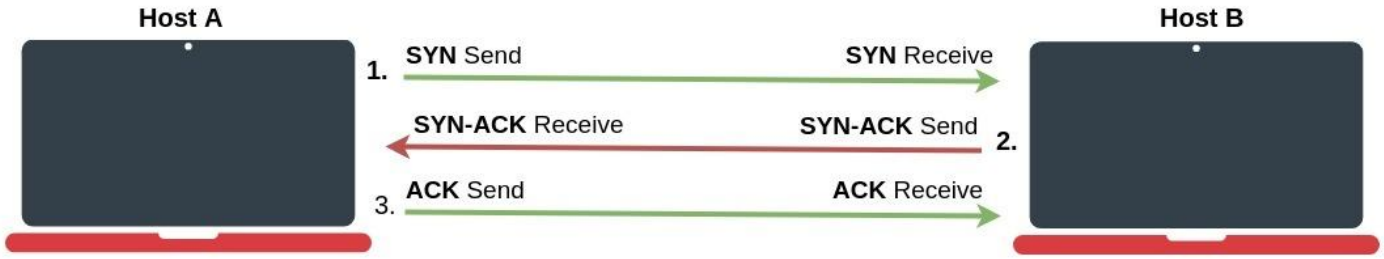
Эта функция используется для поддержки возможностей коммутаторов Ethernet, таких как описанные в RFC 3069, где отдельные порты не могут общаться друг с другом, но могут обмениваться данными с вышестоящим маршрутизатором. Как описано в RFC 3069, можно позволить таким хостам взаимодействовать через маршрутизатор с помощью Proxy ARP. Не обязательно использовать вместе с обычным Proxy ARP.

Эта технология известна под разными названиями:

- В RFC 3069 — **VLAN Aggregation**;
- У Cisco и Allied Telesis — **Private VLAN**;
- У Hewlett-Packard — **Source-Port Filtering** или **Port Isolation**;
- У Ericsson — **MAC-Forced Forwarding** (черновик RFC).

??????????? ? ?????????? TCP ??????

TCP — протокол с установлением соединения, не отправляющий данные до установления соединения. В процессе используется трёхстороннее рукопожатие (three-way handshake), устанавливающее логическое связь с контролем потока и подтверждением доставки.

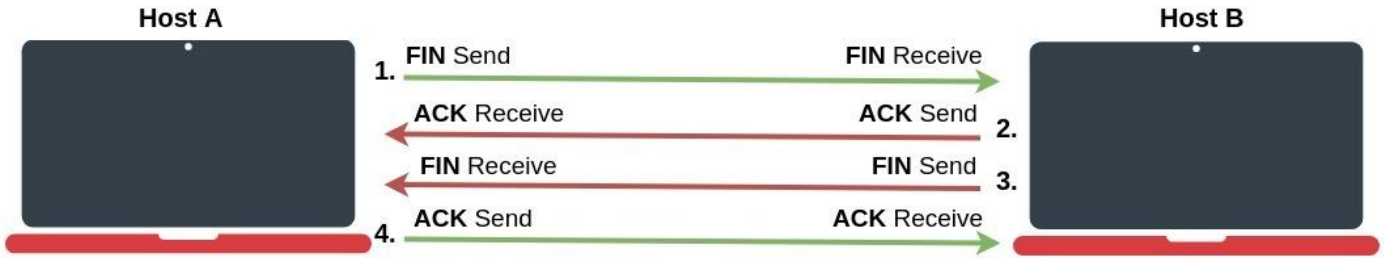


- Host A посылает SYN-пакет с начальным номером последовательности Host B.
- Host B получает SYN и отвечает SYN-ACK.
- Host A получает SYN-ACK и отправляет ACK.
- Host B получает ACK — соединение установлено.

После успешной передачи данных отправитель ждёт подтверждения (ACK). При таймауте пакет пересылается повторно.

?????????? TCP ??????????

Завершение соединения осуществляется четырёхсторонним обменом.



- Host A посылает FIN, сигнализируя о завершении передачи данных.
- Host B входит в состояние CLOSE_WAIT, посылает ACK на FIN. Если у Host B нет данных для передачи, он посылает FIN и переходит в LAST_ACK.
- Host A получает FIN, переходит в TIME_WAIT и посылает ACK.
- Host B получает ACK и соединение завершено.

???????? TCP ????????? (????)

Теперь, когда мы знаем, как устанавливается TCP-соединение, нужно понять, как управляется и поддерживается передача данных. В сетях TCP/IP передача между хостами осуществляется с помощью протокола TCP.

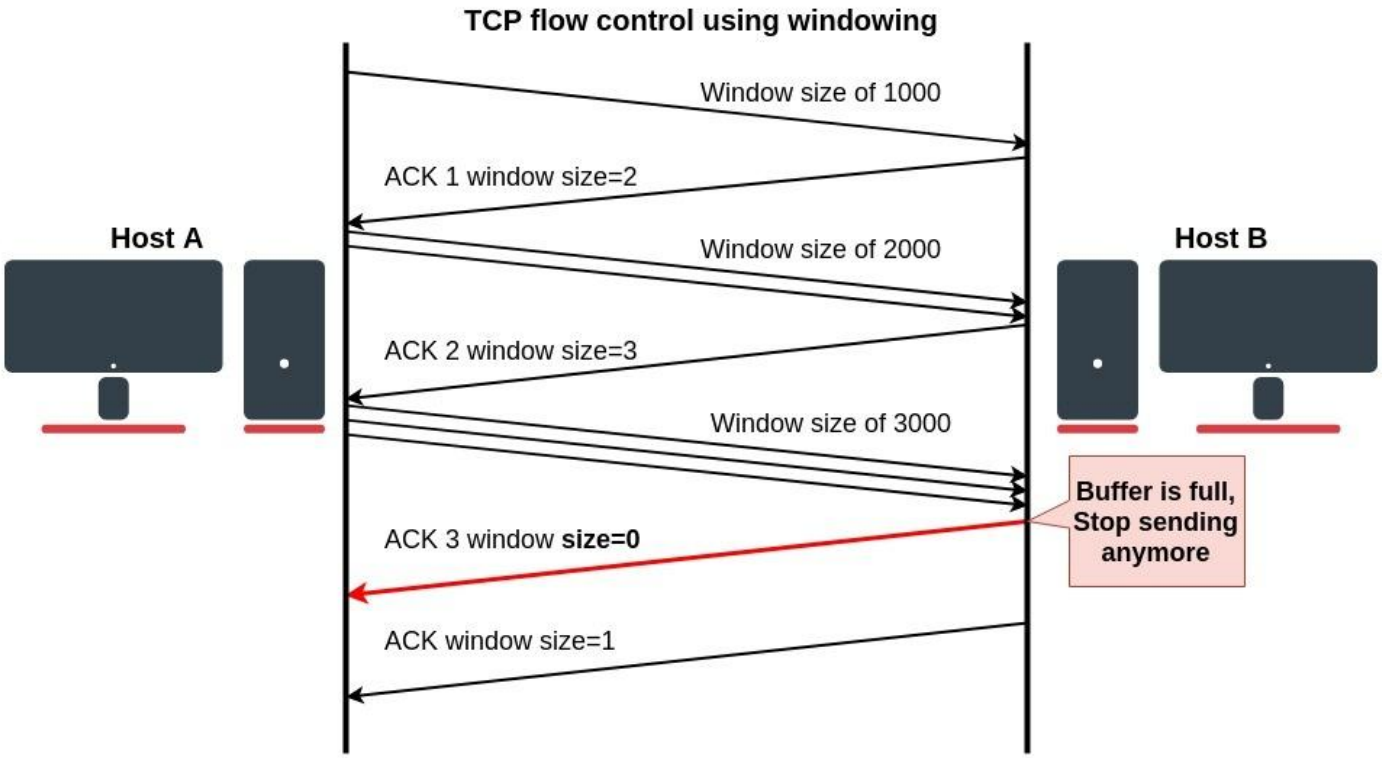
Рассмотрим, что происходит, когда дейтаграммы отправляются быстрее, чем принимающее устройство успевает их обработать. Приёмник сохраняет их в области памяти, называемой буфером. Но поскольку объём буфера ограничен, при его переполнении приёмник начинает отбрасывать кадры. Все отброшенные кадры должны быть переданы повторно, что приводит к снижению производительности передачи данных.

Для решения этой проблемы TCP использует механизм управления потоком (flow control). Для этого применяется окно передачи данных (window mechanism), с помощью которого регулируется объём передаваемой информации.

Когда соединение установлено, приёмник указывает значение поля **window** в каждом TCP-сегменте. Размер окна определяет количество данных, которые приёмник готов временно сохранить в своём буфере. Размер окна (в байтах) передаётся вместе с подтверждениями (ACK) отправителю. Таким образом, размер окна определяет, сколько данных может быть передано от одного хоста к другому без получения подтверждения. Отправитель передаёт только объём данных, соответствующий размеру окна, а затем ждёт подтверждения с обновлённым значением окна.

Если принимающее приложение способно обрабатывать данные так же быстро, как они поступают от отправителя, то приёмник будет отправлять положительные уведомления о размере окна (увеличивая его значение) с каждым подтверждением. Так происходит до тех пор, пока скорость отправителя не превысит скорость обработки приёмника — тогда входящие данные начнут заполнять буфер приёмника, и он отправит подтверждение с **нулевым окном** (zero window). Отправитель, получивший уведомление о нулевом окне, должен приостановить передачу данных, пока не получит новое подтверждение с положительным размером окна.

Рассмотрим иллюстрированный процесс работы оконного механизма:



1. Host A посылает фрейм размером 1000 байт (окно=1000).
2. Host B присылает ACK с размером окна 2000.
3. Host A получает ACK и отправляет два фрейма по 1000 байт.
4. Host B увеличивает окно до 3000, Host A отправляет три фрейма и ждёт подтверждения.
5. Буфер Host B переполняется быстрее, чем данные обрабатываются, окно уменьшается до нуля — нужно ждать.
6. Алгоритмы управления перегрузкой TCP (Reno, Vegas, Tahoe и др.) регулируют размер окна.

IP-?????????

?????

IP-адреса служат для общей идентификации хостов в IP-сетях (RFC 791). Типичный адрес (IPv4) состоит из четырех октетов. Для правильной адресации роутеру также необходимо значение сетевой маски, то есть какие биты полного IP-адреса соответствуют адресу хоста, а какие — адресу сети. Значение сетевого адреса вычисляется бинарной операцией И по сетевой маске и значениям IP-адреса. Также возможно указать IP-адрес, за которым следует слэш "/" и количество бит, формирующих сетевой адрес.

В большинстве случаев достаточно указать адрес, маску и интерфейс. Префикс сети и широковещательный адрес вычисляются автоматически.

Можно добавить несколько IP-адресов к одному интерфейсу или оставить интерфейс без назначенных адресов. В случае объединения интерфейсов (bridge) или подключения PPPoE физический интерфейс может не иметь адрес, но при этом оставаться пригодным к использованию. Назначение IP-адреса физическому интерфейсу, входящему в bridge, означает фактическую настройку адреса на самом bridge-интерфейсе.

Можно использовать `/ip address print detail`, чтобы увидеть, к какому интерфейсу принадлежит адрес.

IPv4 ?????????

IPv4 использует 4-байтовые адреса, которые разделены на четыре 8-битных поля, называемых октетами. Каждый октет преобразуется в десятичный формат и отделяется точкой. Например:

11000000 10101000 00000011 00011000 => 192.168.3.24

Сеть IPv4 состоит из трех адресов:

- **сетевой адрес** — стандартный способ обозначения IPv4-адреса, назначенного сети. Например, сеть 192.168.1.0 или 172.16.0.0 называется «сетевой адрес».
- **широковещательный адрес** — специальный адрес для каждой сети, позволяющий передавать сообщения всем хостам в этой сети. Широковещательный адрес использует самый высокий адрес в диапазоне сети. Например, для сети 192.168.1.0/24 широковещательный адрес будет 192.168.1.255.
- **адрес хоста** — любой другой адрес, не являющийся сетевым или широковещательным, можно использовать как адрес хоста. Например, в диапазоне 192.168.1.0/24 доступны адреса хостов 192.168.1.2 - 254.

Существует несколько типов IP-адресации:

- **unicast** — обычно относится к одному отправителю или одному получателю и может использоваться как для отправки, так и для приёма. Обычно unicast-адрес связан с одним устройством или хостом, но не обязательно в одном к одному.
- **broadcast** — адрес для передачи данных всем возможным получателям («all-hosts broadcast»), что позволяет отправителю отправить данные один раз, а все получатели получат копию. В протоколе IPv4 для локального широковещания используется адрес *255.255.255.255*. Кроме того, можно сделать направленное (ограниченное) широковещание, комбинируя префикс сети с суффиксом хоста, состоящим из всех единиц в двоичном виде. Например, адрес назначения для направленного широковещания в сети 192.0.2.0/24 будет 192.0.2.255.
- **multicast** — адрес, ассоциированный с группой заинтересованных получателей. В IPv4 адреса с 224.0.0.0 по 239.255.255.255 выделены под multicast. Отправитель посылает одиночный датаграмм с unicast-адреса на multicast-адрес группы, а промежуточные маршрутизаторы обеспечивают копирование и доставку всем участникам группы, которые присоединились к ней.

???????? ????????? ???????

Следующие диапазоны IP-адресов зарезервированы (RFC 6890) для приватной адресации. Эти адреса не маршрутизируются в глобальной таблице маршрутизации и должны преобразовываться в глобальные адреса с помощью NAT:

- 10.0.0.0/8 — начало: 10.0.0.0; конец: 10.255.255.255
- 172.16.0.0/12 — начало: 172.16.0.0; конец:172.31.255.255
- 192.168.0.0/16 — начало: 192.168.0.0; конец: 192.168.255.255

?????? ???????????????? ????????? ???????

- 198.18.0.0/15 — для тестирования производительности (benchmarking)
- 192.88.99.0/24 — 6to4 relay anycast
- 192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24 — для документации
- 169.254.0.0/16 — для авто-конфигурации (link-local)

?????????? IP-??????

Рассмотрим настройки, где два роутера напрямую соединены кабелем, и нам не хочется тратить адресное пространство:

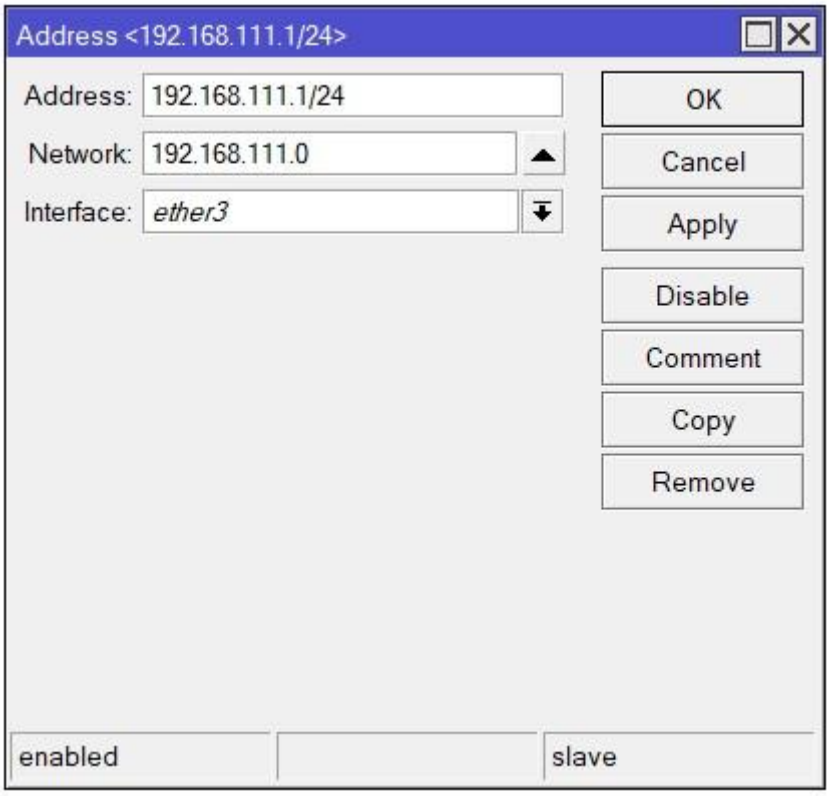
Конфигурация R1:

```
/ip address add address=10.1.1.1/32 interface=ether1 network=172.16.1.1
```

Конфигурация R2:

```
/ip address add address=172.16.1.1/32 interface=ether1 network=10.1.1.1
```

????????



Свойство	Описание
address (IPv4 address [IPv4] / netmask [0..32]; По умолчанию:)	IPv4-адрес с маской.
comment (string; По умолчанию:)	Описание элемента.
disabled (yes | no; По умолчанию: no)	Включён ли адрес или отключен. По умолчанию включён.
interface (interface; По умолчанию:)	Интерфейс, на котором настроен IPv4-адрес. Можно выбрать из доступных интерфейсов роутера.
network (IPv4 address; По умолчанию:)	Сетевой адрес, вычисляемый из параметра address и маски.

?????? ??? ??????

Свойство	Описание
actual-interface (string)	Фактически настроенный интерфейс. Например, если адрес назначен ethernet-интерфейсу, который входит в bridge, то фактический интерфейс — это bridge, а не ethernet.
dynamic (yes | no)	Создан ли адрес динамически.
invalid (yes | no)	Является ли адрес недействительным.

Свойство	Описание
slave (yes | no)	Относится ли адрес к интерфейсу, который является слейвом другого мастера.
VRF (string)	Указывает, с каким VRF связан этот IP.

IPv6 ?????????

Internet Protocol версии 6 (IPv6) — это новая версия IP. Изначально ожидалось, что IPv6 быстро заменит IPv4, но пока эти две версии будут сосуществовать в обозримом будущем. Тем не менее, IPv6 становится всё важнее, по мере исчерпания пула незанятых адресов IPv4.

Два основных преимущества IPv6 по сравнению с IPv4:

- гораздо большее адресное пространство;
- поддержка безсостояной (stateless) и состояной (stateful) автоматической конфигурации адресов;
- встроенная безопасность;
- новый формат заголовка (ускорение обработки).

IPv6 использует 16-байтовые адреса по сравнению с 4-байтовыми в IPv4. Синтаксис и типы IPv6 адресов описаны в RFC 4291.

Существует несколько типов IPv6-адресов, которые можно распознать по префиксу. RouterOS различает следующие группы:

- multicast (префикс ff00::/8)
- link-local (префикс fe80::/10)
- unique local addresses (префикс fc00::/7)
- loopback (адрес ::1/128)
- unspecified (адрес ::/128)
- другие (все остальные, включая устаревшие site-local и RFC 4193 ULA; все считаются глобальными unicast)

Одно из отличий IPv6 от IPv4 — это автоматическая генерация **link-local** IPv6 адреса для каждого активного интерфейса с поддержкой IPv6.

IPv6 адреса представлены иначе, чем IPv4. В IPv6 128-битный адрес делится на восемь 16-битных блоков, и каждый блок переводится в 4-значное шестнадцатеричное число, разделенное двоеточиями. Такое представление называется colon-hexadecimal.

В примере ниже IPv6 адрес в двоичном формате преобразован в colon-hexadecimal:

0010000000000001 0000010001110000 0001111100001001 0000000100110001
0000000000000000 0000000000000000 0000000000000000 0000000000001001

2001:0470:1f09:0131:0000:0000:0000:0009

IPv6 адрес можно упростить, удалив ведущие нули в каждом блоке:

2001:470:1f09:131:0:0:0:9

Как видно, IPv6 адреса могут содержать длинные последовательности нулей. Эти последовательности можно сжать, заменив их на двойное двоеточие **::**:

2001:470:1f09:131::9

Сжатие нулей можно использовать только один раз, иначе невозможно будет однозначно определить число пропущенных нулей.

Префикс IPv6 записывается в формате **address/prefix-length**. В отличие от IPv4, десятичное представление маски сети здесь не применяется. Примеры префиксов:

2001:470:1f09:131::/64 2001:db8:1234::/48 2607:f580::/32 2000::/3

???? ???????

Существуют несколько типов IPv6 адерсов:

- Unicast
- Anycast
- Multicast

В IPv6 отсутствуют широковещательные (Broadcast) адреса, их функционал полностью заменён multicast.

Unicast ??????

Пакеты, адресованные на unicast, доставляются только одному интерфейсу. К ним относятся:

- глобально уникальные адреса и могут использоваться для соединения с адресами с глобальной областью;
- link-local адреса;
- уникальные локальные адреса (ULA RFC4193);
- site-local адреса (FEC0::/48) — устарели;
- специального назначения;
- компатибельные адреса.

Глобальный unicast-адрес может быть автоматически назначен узлу с помощью **Stateless Address auto-configuration**.

Link-local ?????

Link-local адрес обязателен для каждого интерфейса с поддержкой IPv6. Приложения могут рассчитывать на наличие link-local адреса, даже если маршрутирования IPv6 нет, поэтому

этот адрес генерируется автоматически для каждого активного интерфейса с использованием идентификатора интерфейса (вычисленного EUI-64 от MAC-адреса, если он есть). Префикс адреса всегда **FE80::/64**, и IPv6 роутер не пересылает link-local трафик за границы сегмента.

Эти адреса аналогичны автонастраиваемым адресам IPv4 169.254.0.0/16.

Link-local адрес также обязателен для процессов обнаружения соседей (Neighbor Discovery) в IPv6.

Если интерфейс является портом бриджа, интерфейс-специфичный link-local адрес удаляется, оставляя только link-local адрес на bridge.

?????????? ????????? ?????? (Unique Local Address)

ULA зарезервированы для локального использования в домашних и корпоративных сетях. Они не маршрутизируются в публичном адресном пространстве и эквивалентны приватным диапазонам IPv4.

Зарезервированный диапазон — **fc00::/7**.

?????? ???????????? ??????????

Адрес	Описание
Unspecified address (::/128)	Никогда не назначается интерфейсу и не используется в качестве адреса назначения, служит для указания отсутствия адреса. Эквивалент 0.0.0.0 в IPv4.
loopback address (::1/128)	Используется для идентификации интерфейса loopback, позволяя узлу отправлять пакеты самому себе. Эквивалент 127.0.0.1 в IPv4.
2002::/16	Префикс для 6to4 адресации. Использует IPv4 сеть 192.88.99.0/24.
2001:db8::/32	Диапазон адресов, зарезервированных для документации. Не должны встречаться в исходящих или входящих пакетах.
2001:0010::/28	Экспериментальный префикс Orchid. Не должен встречаться в трафике.
2001:0002::/48	Используется для тестирования производительности. Не должен использоваться как источник или назначение.
2001:0000::/32	Teredo адреса.

??????????? ?????? (Compatibility Address)

Адрес	Описание

IPv4 compatible address	Используется узлами с двойным стеком, которые обмениваются IPv6 трафиком поверх IPv4 инфраструктуры. При использовании такого адреса IPv6 трафик инкапсулируется в IPv4 заголовок и отправляется через IPv4 сеть. Адрес записывается в формате `::w.x.y.z`, где w.x.y.z — IPv4 адрес в точечной десятичной форме.
IPv4 mapped address	Используется для представления IPv4-only узла узлу IPv6. Используется только для внутреннего представления. IPv4-mapped адрес никогда не используется в качестве источника или назначения IPv6 пакета. IPv6 протокол не поддерживает использование таких адресов. Записывается в формате `::ffff:w.x.y.z`.

Multicast ??????

Главные особенности multicast:

- трафик отправляется на один адрес, но обрабатывается множеством хостов;
- членство в группе динамично, хосты могут присоединяться к группе и выходить из неё;
- в IPv6 сообщения Multicast Listener Discovery (MLD) используются для определения членства в группе на сетевом сегменте, известном как линк или подсеть;
- хост может отправлять трафик в адрес группы, не будучи участником соответствующей группы.

Один IPv6 multicast адрес идентифицирует каждую группу multicast. Зарезервированный IPv6 адрес каждой группы используется всеми хостами группы, которые слушают и получают сообщения, посылаемые на этот адрес.

Multicast адрес состоит из следующих частей:

- первые 8 бит всегда 1111 1111 (в шестнадцатеричном формате — FF);
- флаг, использующий 9-й по 12-й бит, показывает, задан ли адрес как предопределённый (well-known) или нет. Если предопределён, все биты равны 0;
- ID области (Scope ID) указывает, к какой области принадлежит multicast адрес, например Scope ID=2 — это link-local;
- ID группы указывает multicast группу. Есть предопределённые группы, например Group ID=1 — все узлы. Адрес ff02::1 означает Scope ID=2 и Group ID=1, то есть все узлы в link-local области, аналогично broadcast в IPv4.

Таблица зарезервированных IPv6 multicast адресов:

Адрес	Описание
FF02::1	Адрес для всех узлов на том же линке.
FF02::2	Адрес для всех маршрутизаторов на том же линке.

Адрес	Описание
FF02::5	Адрес для всех OSPF маршрутизаторов на том же линке.
FF02::6	Адрес для всех OSPF designated routers на том же линке.
FF02::1:FFXX:XXXX	Solicited-node адрес, используемый для разрешения IPv6 адреса link-local узла к его канальному адресу. Последние 24 бита совпадают с последними 24 битами IPv6 unicast адреса.

Ниже частичный список multicast адресов IPv6, зарегистрированных IANA. Для полного списка обратитесь к документу IANA.

Multicast адреса можно использовать для обнаружения узлов в сети. Например, обнаружить все узлы:

```
mrz@bumba:/media/aaa/ver$ ping6 ff02::1%eth0
PING ff02::1%eth0(ff02::1) 56 data bytes
64 bytes from fe80::21a:4dff:fe5d:8e56: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from fe80::20c:42ff:fe0d:2c38: icmp_seq=1 ttl=64 time=4.03 ms (DUP!)
64 bytes from fe80::20c:42ff:fe28:7945: icmp_seq=1 ttl=64 time=5.59 ms (DUP!)
64 bytes from fe80::20c:42ff:fe49:fce5: icmp_seq=1 ttl=64 time=5.60 ms (DUP!)
64 bytes from fe80::20c:42ff:fe21:f1ec: icmp_seq=1 ttl=64 time=5.88 ms (DUP!)
64 bytes from fe80::20c:42ff:fe72:a1b0: icmp_seq=1 ttl=64 time=6.70 ms (DUP!)
```

Обнаружить все маршрутизаторы:

```
mrz@bumba:/media/aaa/ver$ ping6 ff02::2%eth0
PING ff02::2%eth0(ff02::2) 56 data bytes
64 bytes from fe80::20c:42ff:fe28:7945: icmp_seq=1 ttl=64 time=0.672 ms
64 bytes from fe80::20c:42ff:fe0d:2c38: icmp_seq=1 ttl=64 time=1.44 ms (DUP!)
```

Anycast ?????

Anycast адрес — новый тип адреса, введённый в IPv6.

Anycast — это новая парадигма сетей, поддерживающая сервисно-ориентированные адреса, где идентичный адрес может быть назначен нескольким узлам, обеспечивающим определённый сервис. Anycast-пакет (с anycast-адресом назначения) доставляется одному из этих узлов с таким же адресом.

Anycast адрес не назначается из отдельного диапазона. Он выделяется из unicast диапазона.

????????????? ??????????

Последние 64 бита IPv6 адреса — это идентификатор интерфейса, уникальный для префикса из 64 бит адреса. Существует несколько способов определения идентификатора интерфейса:

- EUI-64;
- случайно сгенерированный для анонимности;
- ручная настройка.

EUI-64

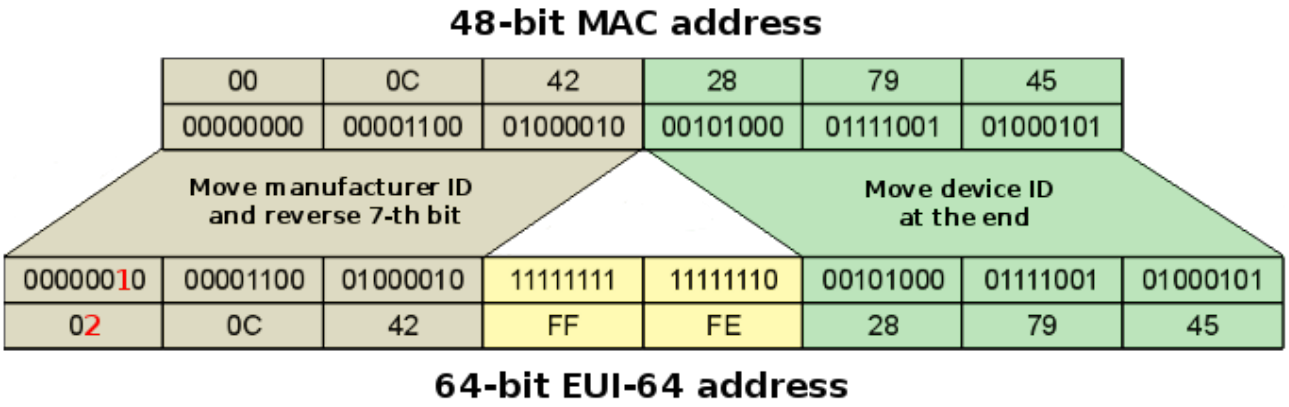
Традиционные идентификаторы интерфейсов для сетевых адаптеров — 48-битные MAC-адреса. Адрес состоит из 24-битного идентификатора производителя и 24-битного идентификатора платы.

IEEE EUI-64 — новый стандарт для сетевых интерфейсных адресов. Идентификатор компании остаётся 24-битным, а идентификатор расширения — 40-битным, что создаёт гораздо большее адресное пространство.

Чтобы создать адрес EUI-64 из MAC-адреса интерфейса:

- В MAC-адрес вставляется 0xFFFE между идентификатором производителя и платы.
- Седьмой бит первого байта инвертируется.

Пример с MAC-адресом 00:0C:42:28:79:45.



Результат после преобразования в colon-hexadecimal будет:

```
20C:42FF:FE28:7945
```

Соответствующий link-local адрес будет:

```
FE80::20C:42FF:FE28:7945/64
```

В RouterOS, если параметр EUI-64 настроен для адреса, последние 64 бита этого адреса автоматически генерируются и обновляются с использованием идентификатора интерфейса. Последние биты должны быть нулевыми для этого случая. Пример:

```
[admin@MikroTik] > ipv6 address add address=fc00:3::/64 interface=ether3 eui-64=yes
[admin@MikroTik] > ipv6 address print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
#   ADDRESS                                  INTERFACE    ADVERTISE
5   G fc00:3::20c:42ff:fe1d:3d4/64           ether3      yes
[admin@MikroTik] > interface ethernet set ether3 mac-address=10:00:00:00:00:01
[admin@MikroTik] > ipv6 address print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
#   ADDRESS                                  INTERFACE    ADVERTISE
5   G fc00:3::1200:ff:fe00:1/64              ether3      yes
```

????????? IPv6 ??????

Пример показывает, как настроить простую адресацию с глобальными IPv6 адресами между двумя роутерами.

Конфигурация R1:

```
/ipv6 address add address=2001:DB8::1/64 interface=ether1 advertise=no
```

Конфигурация R2:

```
/ipv6 address add address=2001:DB8::2/64 interface=ether1 advertise=no
```

Проверка списка адресов:

```
[admin@R1] /ipv6 address> print
Flags: X - disabled, I - invalid, D - dynamic, G - global, L - link-local
 0  G 2001:db8::1/64           ether1    no
 3  DL fe80::219:d1ff:fe39:3535/64  ether1    no
```

Наш добавленный адрес имеет флаг G, обозначающий, что адрес может маршрутизироваться глобально. Также на интерфейсе создан автоматически link-local адрес для каждого IPv6-совместимого интерфейса.

Тест подключения:

```
[admin@R1] /ipv6 address> /ping 2001:DB8::2
HOST             SIZE TTL TIME STATUS
2001:db8::2      56  64  12ms echo reply
2001:db8::2      56  64   0ms echo reply
sent=2 received=2 packet-loss=0% min-rtt=0ms avg-rtt=6ms max-rtt=12ms
```

????? SLAAC IPv6

Если в меню IPv6/Settings включена опция "accept-router-advertisements", и роутер получает пакет Router Advertisement, то SLAAC IPv6 адрес автоматически присваивается интерфейсу, на котором принимаются объявления. Этот адрес будет иметь флаги DG, что означает, что адрес динамический и глобальный. Такие адреса показывают параметры valid и lifetime.

```
[admin@R1] /ipv6/address/print detail where dynamic && global
Flags: X - disabled, I - invalid, D - dynamic; G - global, L - link-local
 0 DG address=2001:db8::::ba69:f4ff:fe84:545/64 from-pool="" interface=ether1 actual-
interface=test_fp eui-64=no advertise=no no-dad=no valid=4w2d preferred=1w
```

Если SLAAC адреса принимаются, то также будет сформирован динамический маршрут в Интернет. В маршруте могут быть указаны ограничения, если они прописаны в пакете объявления, например, hop-limit и MTU. При наличии нескольких адресов на одном интерфейсе будет использовано минимальное значение MTU из всех адресов.

```
[admin@R1] /routing/route/print detail where slaac
Flags: X - disabled, F - filtered, U - unreachable, A - active;
c - connect, s - static, r - rip, b - bgp, o - ospf, d - dhcp, v - vpn, m - modem,
a - ldp-address, l - ldp-mapping, g - slaac, y - bgp-mpls-vpn;
H - hw-offloaded; + - ecmp, B - blackhole
Ag + afi=ip6 contribution=active dst-address=::/0 routing-table=main pref-src=""
gateway=fe80::ba69:f4ff:fe84:7b2%ether1 immediate-gw=fe80::ba69:f4ff:fe84:7b2%ether1
distance=1 scope=30 target-scope=10 belongs-to="slaac" mtu=1400 hoplimit=10 debug.fwp-
ptr=0x201C2C00
```

????????

Свойство	Описание
address (IPv6 address [IPv6] / netmask [0..128]; По умолчанию:)	IPv6 адрес. Адрес также может быть построен из пула, если задан атрибут from-pool. Например, если адрес задан как ::1/64, он будет построен так: <prefix_from_pool>::1/64.
advertise (yes | no; По умолчанию: no)	Включает безсостоянную (stateless) автонастройку адреса. Префикс такого адреса автоматически анонсируется трижды хостам с помощью протокола ICMPv6. Опция по умолчанию включена для адресов с длиной префикса 64. Если адрес удалён или изменён, старый префикс считается устаревшим и анонсируется с временем жизни "0s" трижды.
comment (string; По умолчанию:)	Описание элемента.
disabled (yes | no; По умолчанию: no)	Адрес включён или отключён. По умолчанию не отключён.

Свойство	Описание
eui-64 (yes | no; По умолчанию: no)	Вычислять ли EUI-64 адрес и использовать его в последних 64 битах IPv6 адреса.
from-pool (string; По умолчанию:)	Имя пула, из которого берётся префикс для построения IPv6 адреса, берётся последняя часть из свойства address.
no-dad (yes | no; По умолчанию: no)	Если включено (yes), отключает дублирующее обнаружение адресов (DAD) для IPv6 адресов на интерфейсе.
interface (interface; По умолчанию:)	Интерфейс, для которого настроен IPv6 адрес.
auto-link-local (yes | no; По умолчанию: yes)	Если задан адрес link-local вручную, этот параметр позволяет переопределять автоматически сгенерированный link-local адрес.

?????? ??? ??????

Свойство	Описание
actual-interface (string)	Фактический интерфейс, на котором настроен адрес. Например, если адрес назначен ethernet интерфейсу, входящему в bridge, фактический интерфейс — bridge.
dynamic (yes | no)	Является ли адрес динамическим.
global (yes | no)	Является ли адрес глобальным.
invalid (yes | no)	Является ли адрес недействительным.
link-local (yes | no)	Является ли адрес link-local.
deprecated (yes | no)	Является ли адрес устаревшим.
slave (yes | no)	Принадлежит ли адрес интерфейсу, который является слейвом другого мастера.
VRF (string)	Указывает, с каким VRF связан адрес.

????? ?????????? ???????

Вопрос: Поддерживает ли RouterOS NAT64?

Ответ: Нет, в настоящее время NAT64 в RouterOS не реализован.

IP-????

IP пулами называют диапазоны IP-адресов, которые могут использоваться разными утилитами RouterOS, например, DHCP сервером, Point-to-Point серверами и прочими. Отдельные списки для IPv4 и IPv6 доступны. По возможности каждому клиенту выдается один и тот же IP-адрес (пара OWNER/INFO).

IPv4 ???

Подменю:

```
/ip pool
```

Свойство	Описание
comment (string; По умолчанию: пусто)	Краткое описание пула.
unique identifier	Уникальный идентификатор пула.
next-pool (string; По умолчанию: пусто)	Если при выдаче IP из пула нет свободных адресов и задано свойство next-pool, адрес выдается из следующего пула.
ranges (IP; По умолчанию: пусто)	Список непересекающихся диапазонов IP-адресов в виде: from1-to1,from2-to2,...,fromN-toN. Например, 10.0.0.1-10.0.0.27,10.0.0.32-10.0.0.47.

??????

Определение пула "my-pool" с диапазонами адресов 10.0.0.2-10.0.0.99 и 10.0.0.101-10.0.0.126 (исключая адрес шлюза 10.0.0.1 и сервера 10.0.0.100) и другого пула dhcp-pool с диапазоном 10.0.0.200-10.0.0.250:

```
[admin@MikroTik] ip pool> add name=my-pool ranges=10.0.0.2-10.0.0.99,10.0.0.101-10.0.0.126
[admin@MikroTik] ip pool> add name=dhcp-pool ranges=10.0.0.200-10.0.0.250
[admin@MikroTik] ip pool> print
# NAME       RANGES
0 ip-pool    10.0.0.2-10.0.0.99 10.0.0.101-10.0.0.126
1 dhcp-pool  10.0.0.200-10.0.0.250
```

???????????? ??????

Подменю:

```
/ip pool used
```

Здесь отображаются все занятые IP-адреса из пулов.

Свойство	Описание
address (IP)	IP-адрес, выданный клиенту из пула.
owner (string)	Сервис, использующий данный IP-адрес (для DHCP это MAC-адрес из меню leases, для PPP — имя пользователя типа PPP).
pool (string)	Имя пула.

IPv6 ???

Подменю:

```
/ipv6 pool
```

Свойство	Описание
name (string; По умолчанию: пусто)	Описание пула.
prefix (IPv6/0..128; По умолчанию: пусто)	IPv6 префикс адреса.
prefix-length (integer [1..128]; По умолчанию: пусто)	Размер префикса, который будет выдаваться клиенту.

???????? ?????? ??? ??????

Свойство	Описание
dynamic (yes | no)	Является ли пул динамическим.
expire-time (время)	Время истечения действия для динамических пулов, добавленных DHCPv6 клиентом.

??????

Создание пула с префиксом "2001::/60" для выдачи префиксов длиной /62:

```
[admin@test-host] /ipv6 pool> add name=test prefix=2001::/60 prefix-length=62
[admin@test-host] /ipv6 pool> print
# NAME  PREFIX        PREFIX-LENGTH
0 test  2001::/60    62bits
```

??????????? ??????

Подменю:

```
/ipv6 pool used
```

Свойство	Описание
info (string)	Показывает DUID и другую информацию, полученную от клиента (в шестнадцатеричном виде). Может содержать необработанное время в шестнадцатеричном формате.
owner (string)	Источник резервирования префикса (например, "DHCP").
pool (string)	Имя пула.
prefix (IPv6/0..128)	IPv6 префикс, выданный клиенту из пула.

IP-??????????????

?????

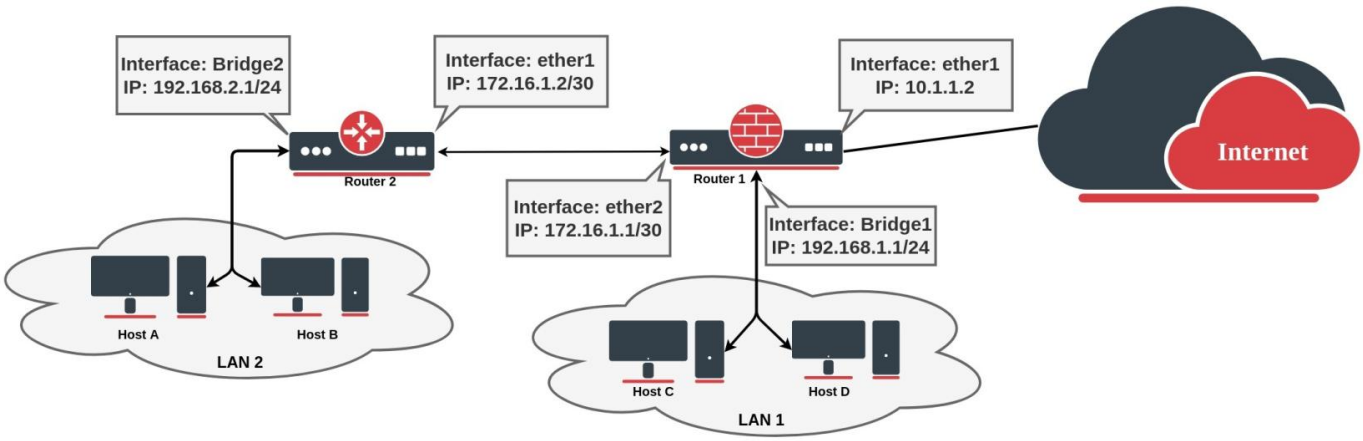
Маршрутизация — это процесс выбора путей по сетям для передачи пакетов от одного хоста к другому.

??? ???????? ?????????????

Рассмотрим базовый пример конфигурации, чтобы показать, как маршрутизация используется для пересылки пакетов между двумя локальными сетями и в Интернет.

В этой схеме у нас есть несколько сетей:

- две клиентские сети (192.168.2.0/24 и 192.168.1.0/24);
- одна сеть для подключения роутеров (172.16.1.0/30), обычно называемая магистралью;
- последняя сеть (10.1.1.0/24) соединяет наш шлюзовый роутер (Router1) с Интернетом.



Роутер 2:

```
/ip address add address=172.16.1.2/30 interface=ether1
/ip address add address=192.168.2.1/24 interface=bridge2
```

Роутер 1 (шлюз), где ether1 подключён к интернету:

```
/ip address add address=10.1.1.2/24 interface=ether1
/ip address add address=172.16.1.1/30 interface=ether2
/ip address add address=192.168.1.1/24 interface=bridge1
```

Если взглянуть, например, на таблицу маршрутизации Router1, можно увидеть, что роутер знает только о *напрямую подключённых* сетях. Если клиент из LAN1 попытается достучаться до клиента из LAN2 (192.168.2.0/24), пакет будет отброшен на роутере, потому что для этого роутера неизвестен маршрут до назначения:

```
[admin@MikroTik] > /ip/route> print
Flags: D - dynamic; X - disabled, I - inactive, A - active; C - connect, S - static,
r - rip, b - bgp, o - ospf, d - dhcp, v - vpn
Columns: DST-ADDRESS, GATEWAY, Distance
DST-ADDRESS      GATEWAY      D
DAC 10.1.1.0/24  ether1       0
DAC 172.16.1.0/30 ether2      0
DAC 192.168.1.0/24 bridge1    0
```

Для исправления нужно добавить маршрут, который укажет роутеру, какое устройство в сети является следующим для достижения назначения. В нашем примере следующий хоп — Router2, поэтому нужно добавить маршрут с gateway, указывающим на связанный адрес Router2. Такой маршрут называется *статическим маршрутом*.

```
[admin@MikroTik] > /ip route add dst-address=192.168.2.0/24 gateway=172.16.1.2
[admin@MikroTik] > /ip/route> print
Flags: D - dynamic; X - disabled, I - inactive, A - active; C - connect, S - static,
r - rip, b - bgp, o - ospf, d - dhcp, v - vpn
Columns: DST-ADDRESS, GATEWAY, Distance
DST-ADDRESS      GATEWAY      D
DAC 10.1.1.0/24  ether1       0
DAC 172.16.1.0/30 ether2      0
DAC 192.168.1.0/24 bridge1    0
  0 AS 192.168.2.0/24 172.16.1.2
```

Теперь пакеты из LAN1 будут успешно пересылаться в LAN2, но на этом дело не кончается. Router2 не знает, как достучаться до LAN1, поэтому любые пакеты из LAN2 будут отброшены на Router2.

Если снова взглянуть на сетевую схему, видно, что у Router2 есть только одна точка выхода. Безопасно предположить, что все неизвестные сети следует достигать через связь с Router1. Самый простой способ — добавить **дефолтный маршрут** с адресом назначения 0.0.0.0/0 или оставить поле пустым:

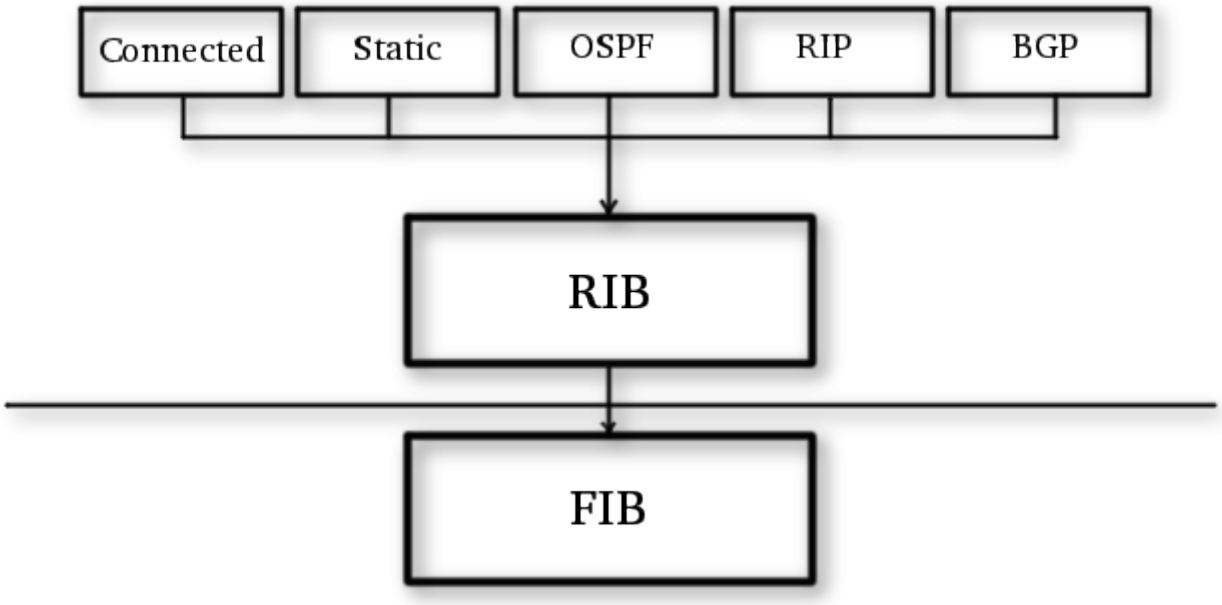
```
/ip route add gateway=172.16.1.1
```

Как показано, маршруты делятся на группы по происхождению и свойствам.

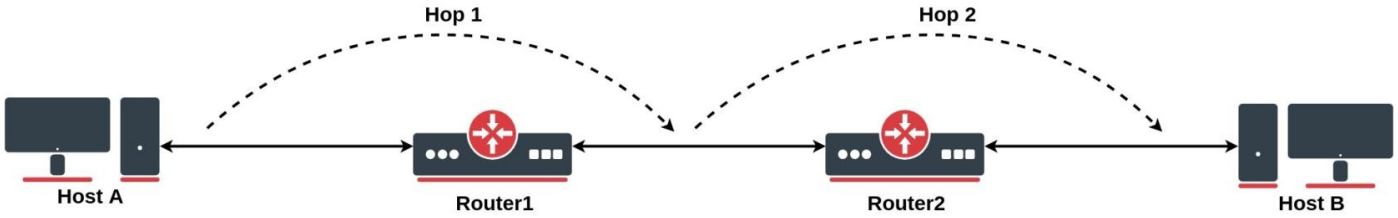
?????????? ? ????????????

Информация о маршрутизации в RouterOS состоит из двух основных частей:

- **FIB** (Forwarding Information Base) — база для принятия решений о пересылке пакетов. Содержит копию необходимой информации о маршрутах.
- **RIB** (Routing Information Base) — содержит все изученные префиксы из протоколов маршрутизации (connected, static, BGP, RIP, OSPF).



Routing Information Base



RIB — база данных, которая содержит записи о конкретных сетевых назначениях и их *gateway* (адрес следующего устройства на пути или просто *next-hop*). Каждая такая запись называется *маршрутом*.

Хоп — это переход пакета от одного сегмента сети к другому.

По умолчанию все маршруты организованы в одну «главную» таблицу маршрутизации. Можно создать несколько таблиц маршрутизации (об этом далее), но для простоты рассмотрим только одну «главную» таблицу.

RIB содержит полную информацию о маршрутах, включая статические маршруты и правила политической маршрутизации, настройку пользователя, информацию, полученную из динамических протоколов (RIP, OSPF, BGP) и информацию о подключённых сетях.

Его задача не только хранить маршруты, но и фильтровать информацию для выбора лучшего маршрута к каждому префиксу назначения, строить и обновлять FIB и распределять маршруты между протоколами маршрутизации.

??????????? ????????

Подключённые маршруты представляют сети, в которых хосты могут быть достигнуты напрямую (прямое подключение к Layer2 широковещательному домену). Эти маршруты создаются автоматически для каждой IP-сети с хотя бы одним включённым интерфейсом (указанном в конфигурации `/ip address` или `/ipv6 address`). RIB отслеживает статус подключённых маршрутов, но не изменяет их.

Для каждого подключённого маршрута существует один IP-адрес, такой что:

- часть **address** в *dst-address* подключённого маршрута равна сети IP-адреса;
- часть **netmask** в *dst-address* равна маске IP-адреса;
- **gateway** подключённого маршрута равен фактическому интерфейсу IP-адреса (за исключением портов bridge) и указывает интерфейс, где можно найти напрямую подключённые хосты сетевого слоя 3.

Preferred source больше не используется для подключённых маршрутов. FIB выбирает исходный адрес на основе исходящего интерфейса. Это позволяет строить конфигурации, которые в ROS v6 и старых считались недопустимыми.

????????? ???????

Дефолтный маршрут используется, если нельзя определить адрес назначения каким-либо другим маршрутом. В RouterOS *dst-address* дефолтного маршрута — 0.0.0.0/0 (IPv4) или ::/0 (IPv6). Если таблица маршрутизации содержит активный дефолтный маршрут, поиск всегда будет успешен.

Обычно в таблице маршрутизации домашнего роутера есть только подключённые сети и один дефолтный маршрут для отправки всего исходящего трафика на шлюз провайдера.

```
[admin@TempTest] /ip/route> print
Flags: D - dynamic; X - disabled, I - inactive, A - active;
C - connect, S - static, r - rip, b - bgp, o - ospf, d - dhcp, v - vpn
Columns: DST-ADDRESS, GATEWAY, Distance
# DST-ADDRESS   GATEWAY       D  DA
0 0.0.0.0/0     10.155.125.1  1
DAC 10.155.125.0/24 ether12     0
DAC 192.168.1.0/24  vlan2       0
```

????????? ???????????? ???????

Устройства с аппаратной разгрузкой:

```
[admin@MikroTik] > /ip/route print where static
Flags: A - ACTIVE; s - STATIC, y - COPY; H - HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
# DST-ADDRESS    GATEWAY        D
0 AsH 0.0.0.0/0   172.16.2.1    1
1 AsH 10.0.0.0/8  10.155.121.254 1
2 AsH 192.168.3.0/24 172.16.2.1 1
```

По умолчанию все маршруты претендуют на аппаратную разгрузку. Чтобы точнее указать, какой трафик разгружать, для каждого статического IP/IPv6 маршрута есть опция включения или отключения `suppress-hw-offload`.

Например, если большая часть трафика идёт к сети серверов, можно включить разгрузку только для этого направления:

```
/ip route set [find where static && dst-address!="192.168.3.0/24"] suppress-hw-offload=yes
```

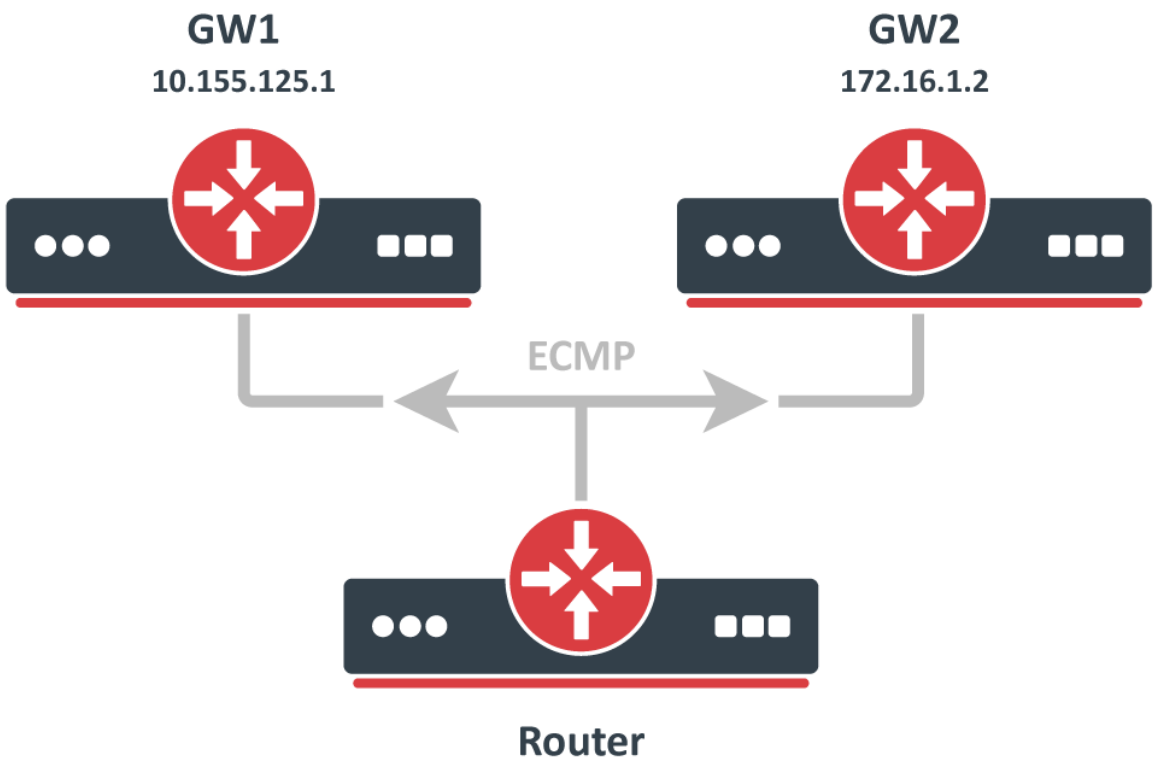
Теперь только маршрут к 192.168.3.0/24 помечен флагом H, что указывает на его пригодность для аппаратной разгрузки:

```
[admin@MikroTik] > /ip/route print where static
Flags: A - ACTIVE; s - STATIC, y - COPY; H - HW-OFFLOADED
Columns: DST-ADDRESS, GATEWAY, DISTANCE
# DST-ADDRESS      GATEWAY      D
0 As 0.0.0.0/0      172.16.2.1  1
1 As 10.0.0.0/8     10.155.121.254 1
2 AsH 192.168.3.0/24 172.16.2.1 1
```

Флаг H указывает лишь на возможность аппаратной разгрузки, а не на её реальное применение.

?????????? (ECMP) ????????

Для реализации балансировки нагрузки может потребоваться использовать несколько путей до назначения.



ECMP (Equal Cost Multi-Path) маршруты имеют несколько шлюзов (next-hop). Все доступные next-hop копируются в FIB и используются для пересылки пакетов.

Такие маршруты можно создавать вручную или они могут быть динамически добавлены протоколами маршрутизации (OSPF, BGP, RIP). Несколько равнозначных маршрутов автоматически группируются с флагом +.

```
[admin@TempTest] /ip/route> print
Flags: D - DYNAMIC; I - INACTIVE, A - ACTIVE; C - CONNECT, S - STATIC, m - MODEM; + - ECMP
Columns: DST-ADDRESS, GATEWAY, DISTANCE
# DST-ADDRESS      GATEWAY      D
0 AS+ 192.168.2.0/24 10.155.125.1 1
1 AS+ 192.168.2.0/24 172.16.1.2  1
```

По умолчанию ECMP использует хеширование уровня 3, которое учитывает исходный и назначенный IP (для IPv4) или исходный IP, назначенный IP, метку потока и протокол IP (для IPv6).

Можно изменить политику хеширования в настройках /ip/setting и /ipv6/settings на:

- Хеширование уровня 4 (Layer4)

- Хеширование внутреннего уровня 3 (inner Layer3)

IPv4	IPv6
L3: srcIPv4, dstIPv4	srcIPv6, dstIPv6, flow label, IP proto
L4: srcIPv4, dstIPv4, srcPort, dstPort, IP proto	srcIPv6, dstIPv6, srcPort, dstPort, IP proto
L3-Inner	

????? ????????

Для одного назначения может приходить несколько маршрутов из разных протоколов или статических настроек, но для пересылки пакетов используется только один лучший маршрут. Для выбора лучшего пути RIB запускает алгоритм выбора, который выбирает лучший маршрут среди кандидатов для каждого назначения.

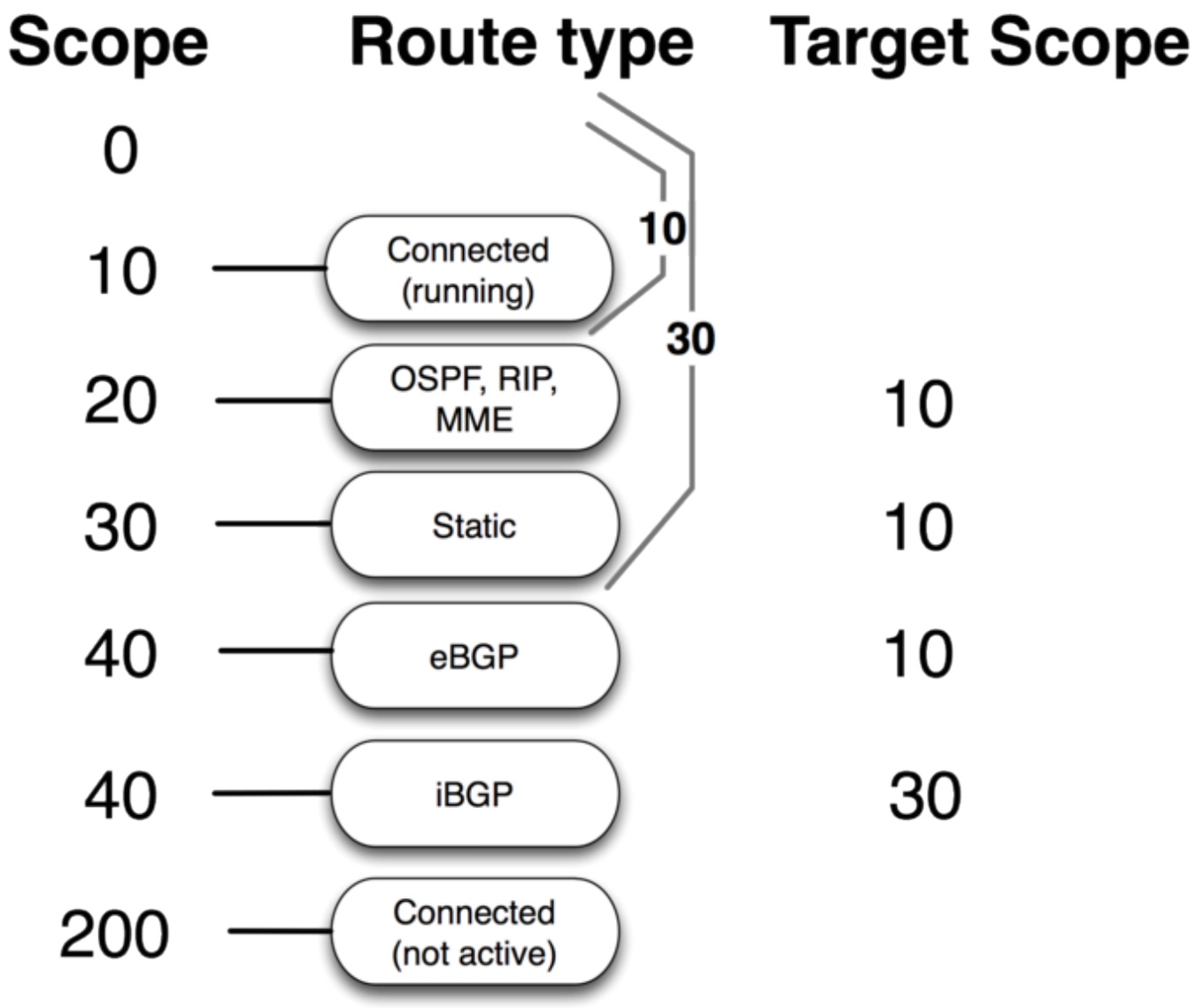
Для участия в выборе маршрут должен:

- не быть отключённым;
- если это unicast маршрут, иметь хотя бы один доступный next-hop (если gateway из связанной сети и связанный маршрут активен, gateway считается доступным);
- не быть синтетическим.

Кандидат с наименьшим расстоянием становится активным маршрутом. При равных расстояниях выбор активного маршрута произвольный.

????? ?????????? ???? (Nexthop Lookup)

Это часть процесса выбора маршрута. Основная задача — найти напрямую доступный адрес шлюза (next-hop). После выбора валидного next-hop роутер знает, какой интерфейс использовать для пересылки пакета.



Поиск next-hop усложняется, если gateway находится в нескольких хопах от роутера (например, iBGP, multihop eBGP). Такие маршруты устанавливаются в FIB после определения напрямую доступного gateway (immediate next-hop).

Нужно ограничить набор маршрутов, используемых для поиска immediate next-hop. Значения next-hop протоколов RIP или OSPF предполагаются напрямую доступными и должны искаться только среди подключённых маршрутов, используя свойства scope и target-scope.

Маршруты со scope больше максимального не участвуют в поиске next-hop. Каждому маршруту присваивается максимальный scope для next-hop в target-scope. По умолчанию разрешен поиск next-hop только через подключённые маршруты, кроме iBGP с расширенным scope, допускающим поиск через IGP и статические маршруты.

В RouterOS v7 произошли изменения в механизме поиска next-hop. Маршруты обрабатываются по возрастанию scope, и изменения в маршрутах с большим scope не влияют на маршруты с меньшим scope.

Пример из v6:

```
/ip route add dst-address=10.0.1.0/24 gateway=10.0.0.1 scope=50 target-scope=30 comment=A
/ip route add dst-address=10.0.2.0/24 gateway=10.0.0.1 scope=30 target-scope=20 comment=B
/ip route add dst-address=10.0.0.0/24 scope=20 gateway=WHATEVER comment=C
```

Gateway 10.0.0.1 рекурсивно разрешается через C с самым маленьким использованием scope (20 из маршрута B), оба маршрута активны. При изменении маршрутов A и B одновременно:

```
/ip route set A target-scope=10
```

Теперь обновление маршрута A делает шлюз маршрута B неактивным, потому что в v6 на один адрес приходится только один объект шлюза.

v7 хранит несколько объектов шлюзов на адрес, по каждой комбинации scope и проверки шлюза. При изменении `target-scope` или проверки шлюза маршрута в ROS v7 эти свойства связаны с самим шлюзом, а не маршрутом, поэтому не влияют на другие маршруты.

Некорректные значения scope автоматически исправляются:

- если scope шлюза = 255, меняется на 254;
- если scope маршрута меньше scope шлюза, меняется на scope шлюза + 1;

Актуальные значения scope и target-scope видны в меню `/routing/nexthop`.

Проверка доступности шлюза расширяется параметром `check-gateway`. Доступность проверяется посылкой ARP-запросов, ICMP-сообщений или проверкой активных BFD-сессий. Каждые 10 секунд роутер шлёт либо ICMP echo request (`ping`), либо ARP-запрос (`arp`). Если 10 секунд ответ не приходит, запрос тайм-аутится. После двух тайм-аутов шлюз считают недоступным. После получения ответа шлюз считается доступным, а счётчик тайм-аутов сбрасывается.

???????? ?????????

Информация о маршрутах хранится с целью минимизации использования памяти. Эти оптимизации имеют худшие случаи, заметные в производительности.

Все маршруты и шлюзы хранятся в единой иерархии по префиксу/адресу.

Каждый "Dst" соответствует уникальному *dst-address* маршрута или адресу шлюза. Для каждого Dst требуется один или более объектов типа T2Node.

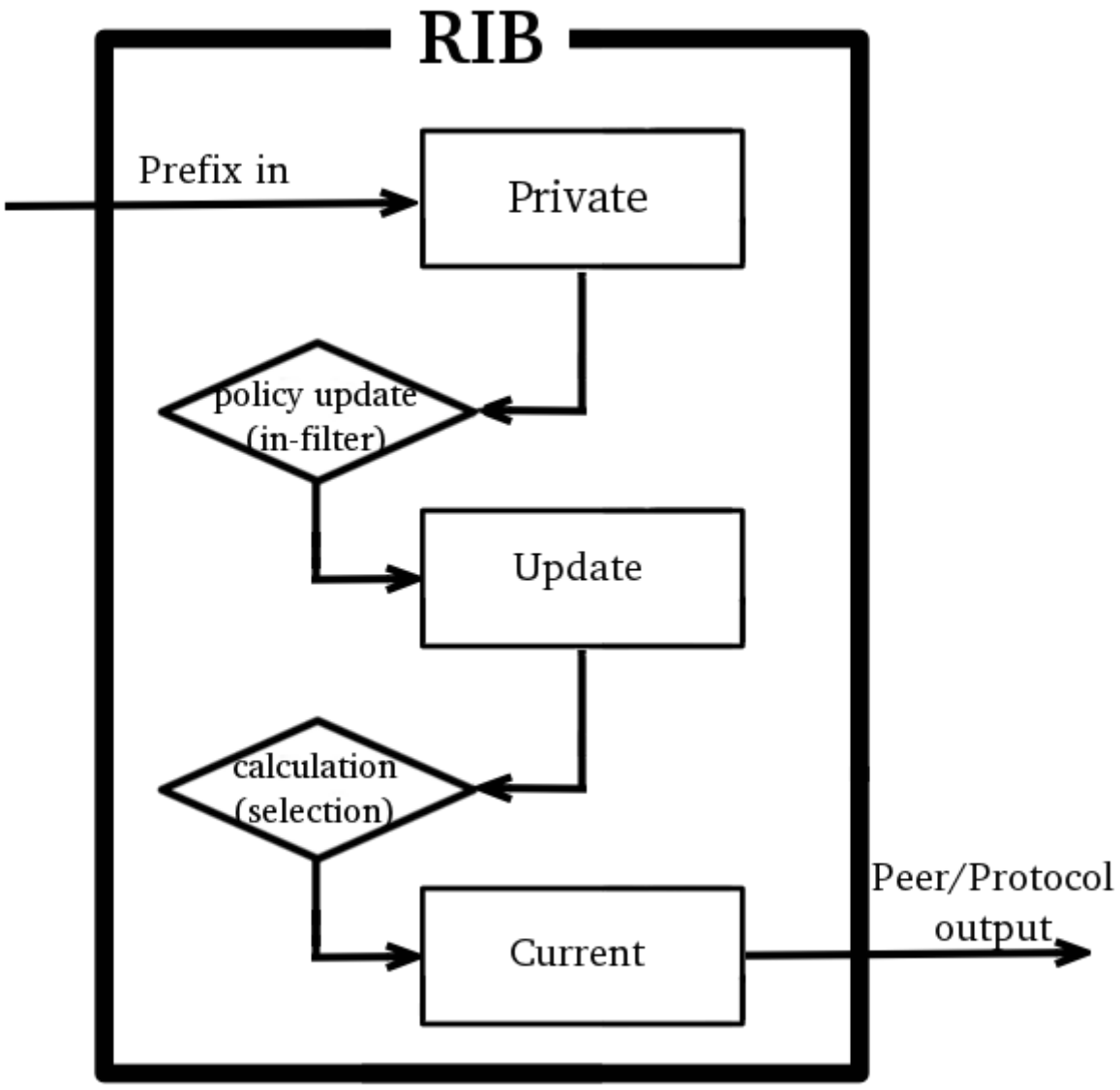
Все маршруты с одним *dst-address* хранятся в списке в Dst, отсортированном по предпочтению.

Примечание: худший случай — много маршрутов с одинаковым dst-address сильно замедляет обновления, даже если они неактивны, из-за накладных расходов сортировки списка.

Порядок маршрутов меняется только при изменении их атрибутов; при смене активности порядок — нет.

Каждый маршрут имеет три копии атрибутов:

- **private** — получено от пира, до применения фильтров;
- **updated** — результат применения фильтров;
- **current** — атрибуты, используемые маршрутом в данный момент.



Атрибуты периодически пересчитываются при поступлении обновлений или изменении фильтров.

Без фильтров private и updated обычно совпадают и равны current.

Атрибуты хранятся в нескольких группах:

- L1 Data — флаги, список доп. свойств, as-path;
- L2 Data — next-hop, метрики RIP, OSPF, BGP, теги маршрутов, инициаторы и т.д.;
- L3 Data — расстояние, scope, тип ядра, MPLS;
- Дополнительные свойства — communities, originator, aggregator-id, cluster-list и др.

Большое разнообразие комбинаций distance и scope увеличивает расход памяти.

Для ускорения фильтрации matching communities или as-path кешируются. Каждый as-path или community имеет кеш для всех regexp с результатами поиска.

Примечание: изменение атрибутов в in-filter увеличивает память (private и updated расходятся), большое количество regexp замедляет работу и расходует память за счёт кешей.

Подробности по памяти маршрутизации доступны в меню `/routing stats memory`.

Forwarding Information Base

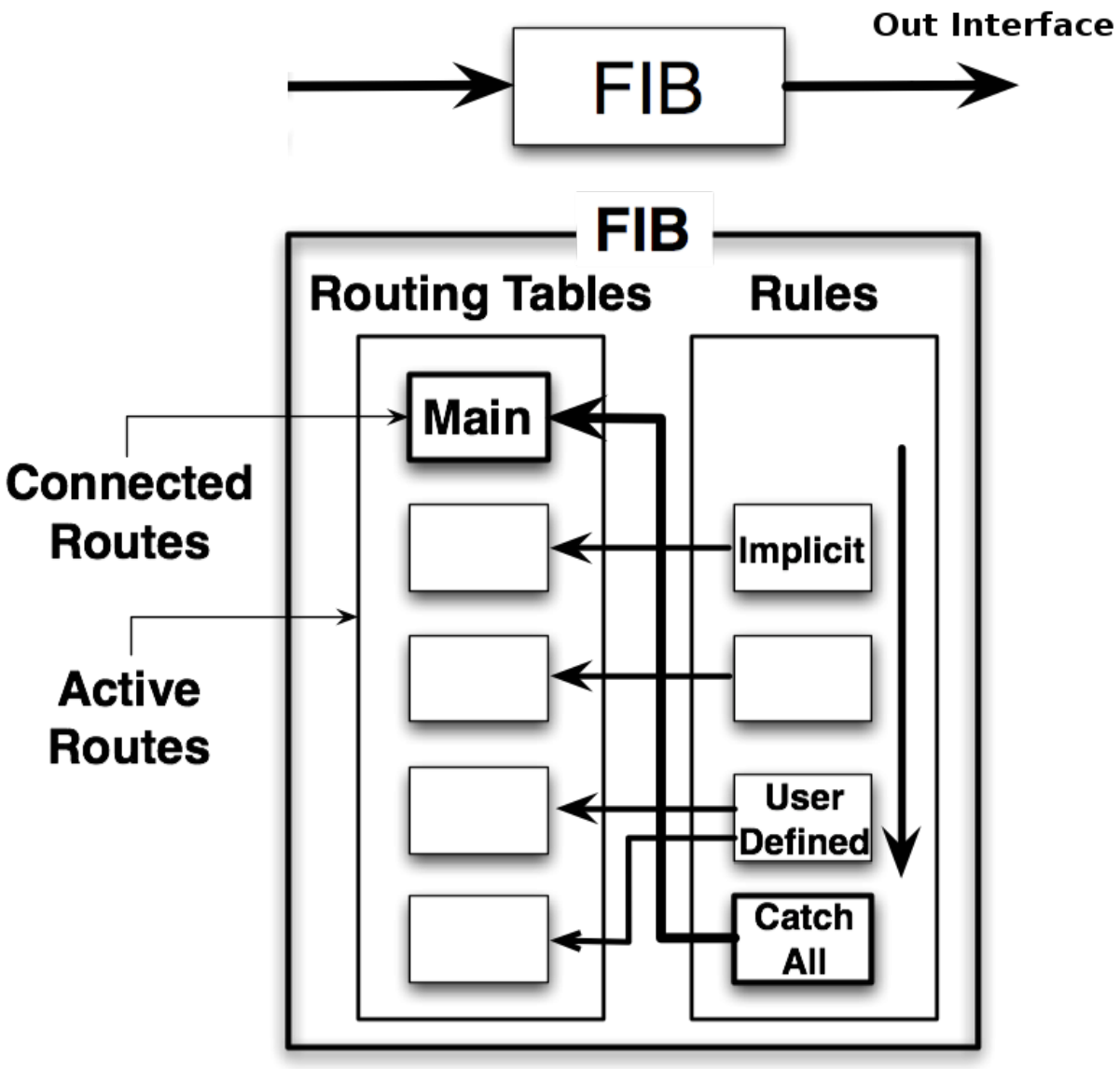
FIB содержит информацию, необходимую для пересылки пакетов:

- все активные маршруты;
- правила политической маршрутизации.

Каждый маршрут имеет свойство **dst-address**, указывающее назначения, для которых он применим. Если несколько маршрутов подходят под конкретный IP, выбирается наиболее специфичный (с самой большой маской). Эта операция называется *поиском в таблице маршрутизации*.

Для пересылки используется только один лучший маршрут. Если несколько маршрутов равнозначны, они объединяются в ECMP маршрут. Лучший маршрут устанавливается в FIB как активный.

Если решение о маршрутизации учитывает дополнительные данные (например, исходный адрес пакета), это называется *политической маршрутизацией*. Она реализована через список правил политики, которые выбирают таблицы маршрутизации по адресу назначения, исходному адресу, исходному интерфейсу и метке маршрутизации (routing mark), которая может изменяться через firewall mangle.



????? ? ??????? ?????????????

FIB использует для определения назначения пакета:

- исходный адрес;
- адрес назначения;
- исходный интерфейс;
- маркировку маршрутизации.

Возможные решения:

- принять пакет локально;
- отбросить пакет (молча или с ICMP сообщением отправителю);
- отправить пакет на конкретный IP на определённом интерфейсе.

Последовательность принятия решения:

1. проверка, должен ли пакет быть доставлен локально (адрес назначения совпадает с адресом роутера);
2. применение неявных правил политики маршрутизации;
3. применение пользовательских правил политики маршрутизации;
4. неявное правило, выполняющее поиск назначения в "главной" таблице маршрутизации;
5. если ничего не найдено — возвращается "сеть недоступна".

Результатом может быть:

- адрес next-hop + интерфейс;
- point-to-point интерфейс;
- локальная доставка;
- отбрасывание;
- ICMP запрещено;
- ICMP хост недоступен;
- ICMP сеть недоступна.

Правила, не совпадающие с пакетом, игнорируются. Если действие:

- **drop** или **unreachable** — возвращается как результат;
- **lookup** — ищет адрес назначения в таблице, указанной в правиле. При неудаче ищется следующее правило;
- **lookup-only** — аналогично lookup, но при отсутствии совпадения поиск завершается неудачей.

В противном случае:

- если маршрут — blackhole, prohibit или unreachable — возвращается соответствующее действие;
- если маршрут подключён или имеет интерфейс в качестве gateway — возвращается интерфейс и адрес назначения;
- если gateway — IP адрес — возвращается адрес и интерфейс;
- если несколько next-hop — выбирается один по кругу.

Установка интерфейса как gateway для статических маршрутов в целом не рекомендуется. Это полезно только в двух случаях:

- point-to-point интерфейсы;
- интерфейсы с напрямую подключённым адресом назначения.

Если интерфейс назначен как gateway, при маршрутизации на широковещательной сети роутер пытается определить destination адрес, отправляя ARP-запросы. Если никому из хостов сети не принадлежит IP, пересылка не состоится. Такие gateway не подходят для маршрутизации пакетов с несколькими хопами.

Gateway, установленный на не point-to-point интерфейсе, не может использоваться для пересылки пакетов с дестинацией, удалённой на несколько хопов.

Параметр `check-gateway` также не применяется для таких gateway по очевидным причинам — нет известного IP назначения.

???????? ?????????

В RouterOS есть три меню, отображающие состояние маршрутов в таблице:

- `/ip route` — список IPv4 маршрутов с основными свойствами;
- `/ipv6 route` — список IPv6 маршрутов с основными свойствами;
- `/routing route` — список всех маршрутов с расширенными свойствами.

Меню `/routing route` только для чтения. Добавлять или удалять маршруты нужно через меню `/ip` или `/ipv6 route`.

?????? ??????

```
[admin@MikroTik] /ip/route> print
Flags: D - dynamic; X - disabled, I - inactive, A - active; C - connect, S - static,
r - rip, b - bgp, o - ospf, d - dhcp, v - vpn
Columns: DST-ADDRESS, GATEWAY, Distance
# DST-ADDRESS      GATEWAY        DI
0 XS 10.155.101.0/24 1.1.1.10       1
1 XS 11.11.11.10      D              d
2 AS 0.0.0.0/0        10.155.101.1  10
3 AS 0.0.0.0/0        10.155.101.1   1
4 AS+ 1.1.1.0/24      10.155.101.1  10
5 AS+ 1.1.1.0/24      10.155.101.2  10
6 AS 8.8.8.8          2.2.2.2       1
DAC 10.155.101.0/24 ether12          0
```

Вывод команды `/routing route` похож на `/ip route`, но показывает маршруты всех семейств адресов в одном меню и отображает отфильтрованные маршруты.

```
[admin@MikroTik] /routing/route> print
Flags: X - disabled, I - inactive, F - filtered, U - unreachable, A - active;
c - connect, s - static, r - rip, b - bgp, o - ospf, d - dhcp, v - vpn,
a - ldp-address, l - ldp-mapping
Columns: DST-ADDRESS, GATEWAY, DISTANCE, SCOPE, TARGET-SCOPE, IMMEDIATE-GW
DST-ADDRESS      GATEWAY        DIS SCO TAR IMMEDIATE-GW
Xs 10.155.101.0/24  Xs d 0.0.0.0/0 10.155.101.1  10 30 10 10.155.101.1%ether12
```

```
As 0.0.0.0/0        10.155.101.1  1  30 10 10.155.101.1%ether12
As 1.1.1.0/24        10.155.101.1  10 30 10 10.155.101.1%ether12
As 8.8.8.8           2.2.2.2       1  254 254 10.155.101.1%ether12
Ac 10.155.101.0/24   ether12       0  10    ether12
Ic 2001:db8:2::/64   ether2        0  10
Io 2001:db8:3::/64   ether12       110 20 10
Ic fe80::%ether2/64  ether2        0  10
Ac fe80::%ether12/64  ether12       0  10    ether12
Ac fe80::%bridge-main/64 bridge-main 0  10    bridge-main
A ether12            0             250
A bridge-main        0             250
```

Команда `/routing route print detail` показывает расширенную информацию, полезную для отладки:

```
[admin@MikroTik] /routing route> print detail
Flags: X - disabled, I - inactive, F - filtered, U - unreachable, A - active;
c - connect, s - static, r - rip, b - bgp, o - ospf, d - dhcp, v - vpn,
a - ldp-address, l - ldp-mapping; + - ecmp
Xs dst-address=10.155.101.0/24
Xs d afi=ip4 contribution=best-candidate
dst-address=0.0.0.0/0 gateway=10.155.101.1 immediate-gw=10.155.101.1%ether12
distance=10 scope=30 target-scope=10 belongs-to="DHCP route"
mpls.in-label=0 .out-label=0 debug.fwp-ptr=0x201C2000
As afi=ip4 contribution=active dst-address=0.0.0.0/0 gateway=10.155.101.1
immediate-gw=10.155.101.1%ether12 distance=1 scope=30 target-scope=10 belongs-to="Static
route"
mpls.in-label=0 .out-label=0 debug.fwp-ptr=0x201C2000
```

????????? IP

????????? IPv4

Подменю:

```
/ip settings
```

Свойство	Описание
accept-redirects (yes | no; По умолчанию: no)	Принимать ли ICMP-сообщения переадресации. Обычно включают на хостах и выключают на роутерах.
accept-source-route (yes | no; По умолчанию: no)	Принимать пакеты с опцией source route (SRR). Обычно включают на роутерах.
allow-fast-path (yes | no; По умолчанию: yes)	Разрешить Fast Path.
arp-timeout (интервал времени; По умолчанию: 30s)	Устанавливает Linux base_reachable_time (base_reachable_time_ms) на всех интерфейсах, использующих ARP. Начальная валидность записи ARP выбирается из интервала [timeout/2 - 3*timeout/2] (по умолчанию от 15 с до 45 с) после нахождения соседа. Можно использовать постфиксы ms, s, m, h, d для миллисекунд, секунд, минут, часов или дней. Если постфикса нет, используется секунда (s). Параметр обозначает, как долго запись ARP считается полной, если в это время не было обмена с конкретным MAC/IP. Параметр не указывает время удаления записи из кеша ARP (см. max-neighbor-entries).
icmp-errors-use-inbound-interface-address (yes | no; По умолчанию: no)	Если включено, ICMP-ответы с ошибками отправляются с адреса основного интерфейса, на котором обнаружена ошибка. Может быть полезно для сложного сетевого отладки.
icmp-rate-limit (целое [0..4294967295]; По умолчанию: 10)	Ограничение максимальной частоты отправки ICMP-пакетов типов, определённых маской icmp-rate-mask, по конкретным целям. 0 — без ограничений, другие значения указывают минимальный интервал между ответами в миллисекундах.
icmp-rate-mask ([0..FFFFFFFF]; По умолчанию: 0x1818)	Маска типов ICMP, для которых применяется ограничение скорости. Подробнее в man-страницах Linux.
ip-forward (yes | no; По умолчанию: yes)	Включить/выключить пересылку пакетов между интерфейсами. Сбрасывает параметры конфигурации в соответствии с RFC1812 для роутеров.
ipv4-multipath-hash-policy (l3 | l4 | l3-inner; По умолчанию: l3)	Политика хеширования IPv4 для маршрутизации ECMP.
rp-filter (loose | no | strict; По умолчанию: no)	Включить или отключить валидацию источника.

Свойство	Описание
secure-redirects (yes | no; По умолчанию: yes)	Принимать ICMP-переадресации только от шлюзов из списка дефолтных шлюзов.
send-redirects (yes | no; По умолчанию: yes)	Отправлять ли ICMP redirect. Рекомендуется включить на роутерах.
tcp-timestamp (параметр)	Включение/отключение TCP timestamp или добавление случайного смещения (поведение по умолчанию). Полное отключение может снизить всплески падений производительности.
tcp-syncookies (yes | no; По умолчанию: no)	Отправлять syncookies при переполнении очереди syn backlog сокета. Защита от атаки SYN flood. Syncookies нарушают стандарты TCP и отключают использование расширений протокола, что может привести к ухудшению работы некоторых служб (например, SMTP relay). Эффект может быть не заметен вам, но виден клиентам и ретрансляторам.
max-neighbor-entries (целое [0..4294967295]; По умолчанию: пусто)	Устанавливает максимальное количество записей ARP. Если записи неполные, они могут оставаться в кэше бесконечно, если их количество менее четверти максимума. Это позволяет избежать частого вызова сборщика мусора при заполненной таблице ARP.
route-cache (yes | no; По умолчанию: yes)	Включить или отключить кэш маршрутов в Linux. Отключение кэша также отключит fast path.

???????? ?????? ??? ?????? IPv4

Свойство	Описание
ipv4-fast-path-active (yes | no)	Показывает, активен ли fast-path.
ipv4-fast-path-bytes (целое)	Количество байт, обработанных fast-path.
ipv4-fast-path-packets (целое)	Количество пакетов, обработанных fast-path.
ipv4-fasttrack-active (yes | no)	Показывает, активен ли fasttrack.
ipv4-fasttrack-bytes (целое)	Количество байт, обработанных fasttrack.
ipv4-fasttrack-packets (целое)	Количество пакетов, обработанных fasttrack.

????????? IPv6

Подменю:

```
/ipv6 settings
```

Изменение настроек /ipv6 не приводит к удалению старых SLAAC конфигураций. Для применения новых настроек требуется перезагрузка.

Свойство	Описание

accept-redirects (no | yes-if-forwarding-disabled; По умолчанию: yes-if-forwarding-disabled)	Принимать ли ICMP redirect сообщения. Обычно включают на хостах и выключают на роутерах.
accept-router-advertisements (no | yes | yes-if-forwarding-disabled; По умолчанию: yes-if-forwarding-disabled)	Принимать ли сообщения Router Advertisement (RA). Если включено, роутер сможет получить адрес через Stateless Address Configuration.
accept-router-advertisements-on (список интерфейсов; По умолчанию: все)	Указывает интерфейсы, на которых принимаются RA сообщения.
disable-ipv6 (yes | no; По умолчанию: no)	Включение/отключение системных настроек IPv6 (предотвращает генерацию link-local адресов).
forward (yes | no; По умолчанию: yes)	Включение/отключение пересылки пакетов между интерфейсами.
max-neighbor-entries (целое [0..4294967295]; По умолчанию: пусто)	Максимальное количество IPv6 соседей. В RouterOS с версии 7.1 значение по умолчанию зависит от объёма установленной памяти. Можно увеличить, но это повышает риск нехватки памяти. Значения по умолчанию зависят от объёма RAM.
multipath-hash-policy (l3 | l4 | l3-inner; По умолчанию: l3)	Политика хеширования IPv6 для ECMP маршрутизации.
disabled-link-local-address (no | yes; По умолчанию: no)	Отключить автоматическое создание link-local адресов для интерфейсов, не являющихся VPN. Полезно при использовании вручную заданных адресов.
stale-neighbor-timeout (время; По умолчанию: 60)	Время, после которого устаревшие записи IPv6 соседей должны быть очищены.
min-neighbor-entries (целое; По умолчанию: 4096)	Минимальное количество записей IPv6 соседей, для которых устройство должно выделять память.
soft-max-neighbor-entries (целое; По умолчанию: 8192)	Ожидаемое максимальное количество записей IPv6 соседей, которое система должна обрабатывать.
16384 max-neighbor-entries (целое; По умолчанию: пусто)	Максимальное число записей в списке IPv6 соседей.
allow-fast-path (yes | no; По умолчанию: yes)	Разрешить Fast Path.

???????? ?????? ??? ?????? IPv6

Свойство	Описание
ipv6-fast-path-active (yes | no)	Показывает, активен ли fast-path.
ipv6-fast-path-bytes (целое)	Количество байт, обработанных fast-path.
ipv6-fast-path-packets (целое)	Количество пакетов, обработанных fast-path.
ipv6-fasttrack-active (yes | no)	Показывает, активен ли fasttrack.
ipv6-fasttrack-bytes (целое)	Количество байт, обработанных fasttrack.
ipv6-fasttrack-packets (целое)	Количество пакетов, обработанных fasttrack.