Архитектура, эволюция и методология конфигурирования подсистемы маршрутизации (Routing) в экосистеме MikroTik RouterOS

19 августа 2026 года

Маршрутизация является фундаментальной функцией любого сетевого устройства класса "маршрутизатор". В экосистеме MikroTik RouterOS подсистема маршрутизации прошла путь от базового механизма пересылки пакетов до высокопроизводительного, унифицированного ядра, способного обеспечивать работу магистральных сетей операторов связи (Carrier-grade) и сложных корпоративных топологий.

Переход на RouterOS v7 ознаменовался радикальным рефакторингом данного модуля. Настоящая статья рассматривает архитектуру унифицированной подсистемы маршрутизации, механизмы реализации Policy-Based Routing (PBR) и VRF, а также анализирует критические улучшения ядра, внедренные в последних релизах.

---

1. Архитектурная трансформация: Унификация в RouterOS v7

Исторически в RouterOS маршруты для IPv4 и IPv6 управлялись раздельно (`/ip route` и `/ipv6 route`). В RouterOS v7 разработчики MikroTik перешли к **унифицированной модели маршрутизации**.

Теперь все маршруты (независимо от семейства протоколов) управляются через единое меню `/routing route`, а таблицы маршрутизации выделены в отдельную сущность `/routing routing-table`.

Преимущества унифицированной архитектуры:
1. **Единая RIB (Routing Information Base):** Все протоколы динамической маршрутизации (OSPF, BGP, RIP, IS-IS) и статические записи помещаются в общее хранилище, что упрощает разрешение конфликтов (route selection) между IPv4 и IPv6.
2. **Разделение RIB и FIB:** Архитектура четко разделяет базу знаний маршрутизатора (RIB) и базу пересылки (FIB — Forwarding Information Base), что оптимизирует процесс обновления аппаратных таблиц коммутационных чипов (Hardware Offloading).
3. **Обратная совместимость:** Меню `/ip route` и `/ipv6 route` сохранены как алиасы (legacy), обеспечивая бесшовную миграцию скриптов и конфигураций.

---

2. Таблицы маршрутизации и VRF (Virtual Routing and Forwarding)

Одним из наиболее значимых архитектурных достижений RouterOS v7 является полноценная поддержка множественных таблиц маршрутизации, что является основой для реализации VRF.

Механизм работы:
По умолчанию все маршруты помещаются в таблицу `main`. Однако администратор может создавать изолированные таблицы маршрутизации:

```
/routing/table/add name=CUSTOMER_A fib
/routing/table/add name=CUSTOMER_B fib
```

Параметр `fib` указывает, что данная таблица должна быть экспортирована в аппаратную базу пересылки (FIB).

Маршруты привязываются к конкретным таблицам с помощью параметра `routing-table`:

```
/routing/route/add dst-address=10.10.10.0/24 gateway=192.168.1.1 routing-table=CUSTOMER_A
```

Это позволяет полностью изолировать трафик различных клиентов или сервисов на одном физическом маршрутизаторе, исключая возможность утечки маршрутов (route leaking) без явного перенаправления.

---

3. Ключевые метрики и атрибуты маршрутов

Для принятия решений о пересылке трафика RouterOS использует строгую иерархию атрибутов маршрута:

1. **Префикс (Dst-address):** Сначала выбирается маршрут с наиболее длинным совпадением префикса (Longest Prefix Match).
2. **Дистанция (Distance):** Если префиксы совпадают, выбирается маршрут с наименьшим значением `distance` (аналог Administrative Distance).
3. **Метрика протокола:** Если дистанции равны (например, два маршрута OSPF), выбор делается на основе внутренней метрики протокола (Cost в OSPF, Local Pref/MED в BGP).

Рекурсивная маршрутизация (Scope и Target-Scope)
Для корректной работы рекурсивных маршрутов (когда шлюз находится не в подключенной сети, а достижим через другой маршрут) используются параметры `scope` и `target-scope`. Они определяют границы поиска следующего прыжка (next-hop), предотвращая бесконечные циклы и обеспечивая корректную работу BGP с удаленными next-hop.

---

4. Эволюция ядра маршрутизации: Критические обновления

В последних релизах RouterOS команда MikroTik сосредоточилась на повышении стабильности ядра маршрутизации, что критически важно для сред с высокой динамикой (flapping) линков. Согласно официальным спискам изменений, были внедрены следующие улучшения:

1. **Поддержка link-local адресов назначения:**
* *Суть:* `route - allow to add route with link-local destination address.`
* *Инженерное значение:* Ранее добавление статических маршрутов на адреса `fe80::/10` было затруднено или невозможно. Теперь это поддерживается нативно, что критически важно для специфических сценариев L2/L3 edge-маршрутизации и протоколов соседства.
2. **Устранение утечек памяти при нестабильности линков:**
* *Суть:* `route - fixed memory leak when flapping addresses or interfaces with routing protocols running.`
* *Инженерное значение:* В сетях с нестабильными каналами (например, беспроводные backhaul-линки или LTE) частое появление и исчезновение интерфейсов приводило к постепенной утечке памяти в процессе пересчета FIB. Устранение данного дефекта гарантирует долгосрочную стабильность (uptime) маршрутизаторов.
3. **Устранение состояний гонки (Race Conditions):**
* *Суть:* `route - fixed potential race condition.`
* *Инженерное значение:* Оптимизация многопоточной обработки обновлений RIB. Это предотвращает редкие, но критические сбои при одновременном поступлении массовых обновлений от нескольких протоколов динамической маршрутизации.
4. **Корректная обработка ECMP для IPv6:**
* *Суть:* `route - respect the "interface" property when pinging IPv6 addresses over ECMP.`
* *Инженерное значение:* При использовании балансировки нагрузки (ECMP) и диагностике пути через `ping` с жестким указанием интерфейса, система теперь корректно выбирает нужный next-hop, исключая асимметричную маршрутизацию диагностических пакетов.

---

5. Policy-Based Routing (PBR) и Routing Marks

Для реализации маршрутизации на основе политик (PBR) в RouterOS используется механизм **Routing Marks**. Он позволяет перенаправлять трафик в специфические таблицы маршрутизации на основе критериев, не связанных с адресом назначения (например, по порту источника, протоколу или пользователю).

Алгоритм настройки PBR:
1. **Маркировка пакетов в Mangle:**

```
/ip/firewall/mangle/add chain=prerouting src-address=192.168.10.0/24
action=mark-routing new-routing-mark=to_ISP2 passthrough=no
```

2. **Создание таблицы маршрутизации и маршрута:**

```
/routing/table/add name=to_ISP2 fib
/routing/route/add dst-address=0.0.0.0/0 gateway=ISP2_GW routing-table=to_ISP2
`
```

Трафик, помеченный как `to_ISP2`, будет использовать таблицу `to_ISP2`, игнорируя глобальную таблицу `main`.

---

6. Инженерные практики и лучшие рекомендации

При проектировании и эксплуатации подсистемы маршрутизации в RouterOS рекомендуется придерживаться следующих стандартов:

1. **Интеграция с BFD (Bidirectional Forwarding Detection):**
Для протоколов OSPF и BGP настройка BFD позволяет сократить время сходимости (convergence time) при обрыве линка с секунд до миллисекунд. Это критически важно для сетей реального времени и VoIP.
2. **Агрегация маршрутов (Route Summarization):**
В крупных сетях следует избегать раздувания таблицы маршрутизации. Используйте агрегацию префиксов на границах зон

(например, в OSPF ABR или BGP Route Reflectors) для снижения нагрузки на CPU и ускорения пересчета FIB.
3. **Защита от инъекции маршрутов:**
При использовании BGP или OSPF необходимо строго настраивать фильтры маршрутов (Routing Filters), чтобы предотвратить прием некорректных или вредоносных префиксов (например, дефолтного маршрута от ненадежного пира), что может привести к перехвату трафика (Blackholing).
4. **Мониторинг состояния FIB:**
Регулярная проверка синхронизации RIB и FIB. Если количество маршрутов в RIB значительно превышает количество записей в FIB, это может указывать на нехватку ресурсов CPU или проблемы с аппаратным ускорением на коммутаторах.

---

7. Заключение

Подсистема маршрутизации в MikroTik RouterOS v7 представляет собой зрелое, высокопроизводительное и архитектурно выверенное ядро. Переход к унифицированной модели, поддержка изолированных таблиц маршрутизации (VRF) и механизмов Policy-Based Routing выводят RouterOS на один уровень с решениями enterprise- и carrier-класса.

Последние итерации, направленные на устранение утечек памяти, оптимизацию многопоточности и расширение поддержки специфических адресов (link-local), подтверждают ориентацию MikroTik на обеспечение максимальной стабильности в средах с высокой динамикой. При условии грамотного проектирования топологии, использования рекурсивной маршрутизации и строгого контроля за анонсируемыми префиксами, маршрутизатор MikroTik способен обеспечить надежную и предсказуемую пересылку трафика в самых сложных сетевых инфраструктурах.