

*Mikro**T**ik.Me*

*Mikro**T**ik.Me*

# MikroTik.Me

- Васильев Кирилл
- Санкт-Петербург
- Курсы MikroTik
- Поддержка

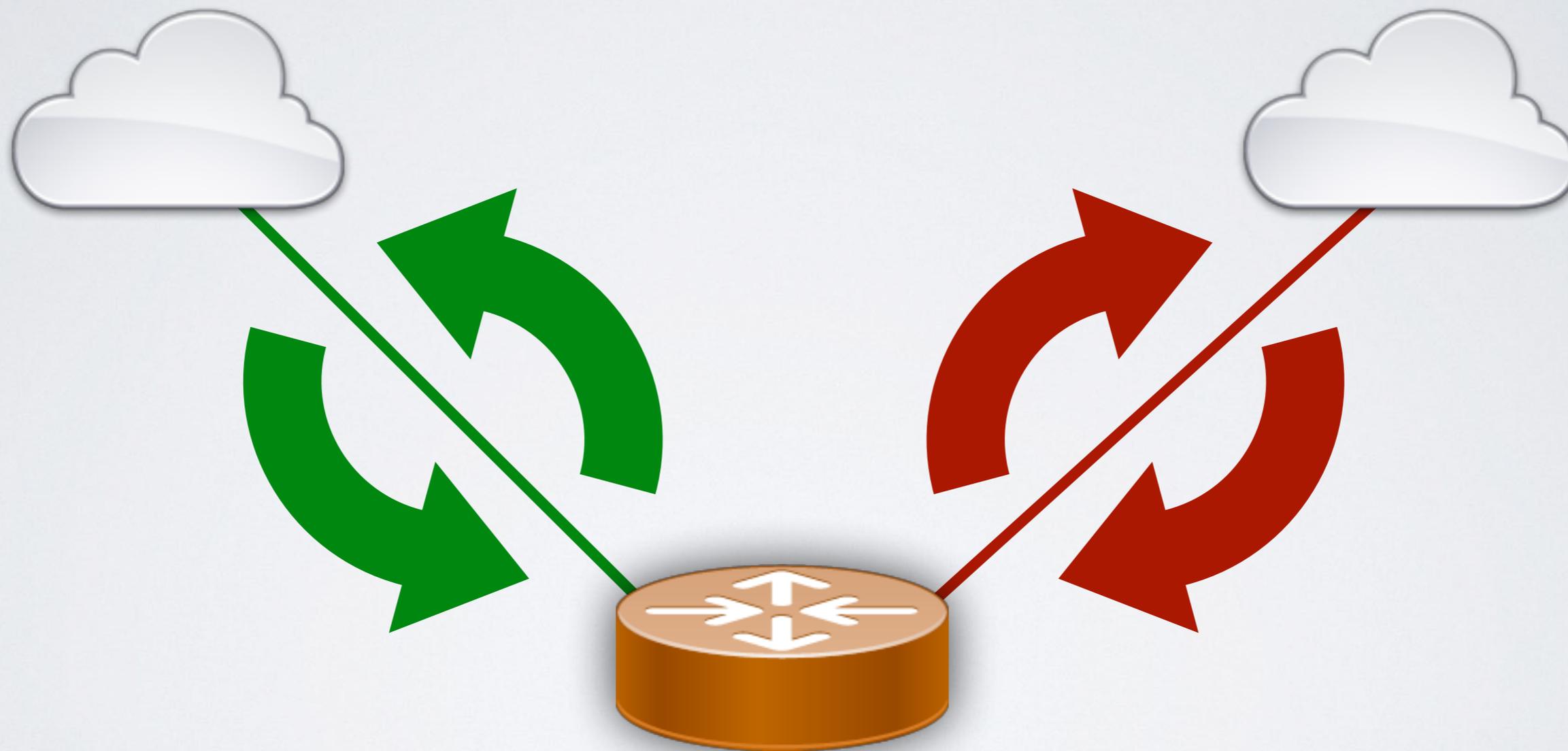


# **Настройка RouterOS для одновременной работы с несколькими провайдерами**

# Multiwan

- Обеспечить доступ к маршрутизатору
- Организовать правильный выход с маршрутизатора
- Одновременный доступ к ресурсам «за» NAT
- Распределение нагрузки по каналам

# Доступ к маршрутизатору



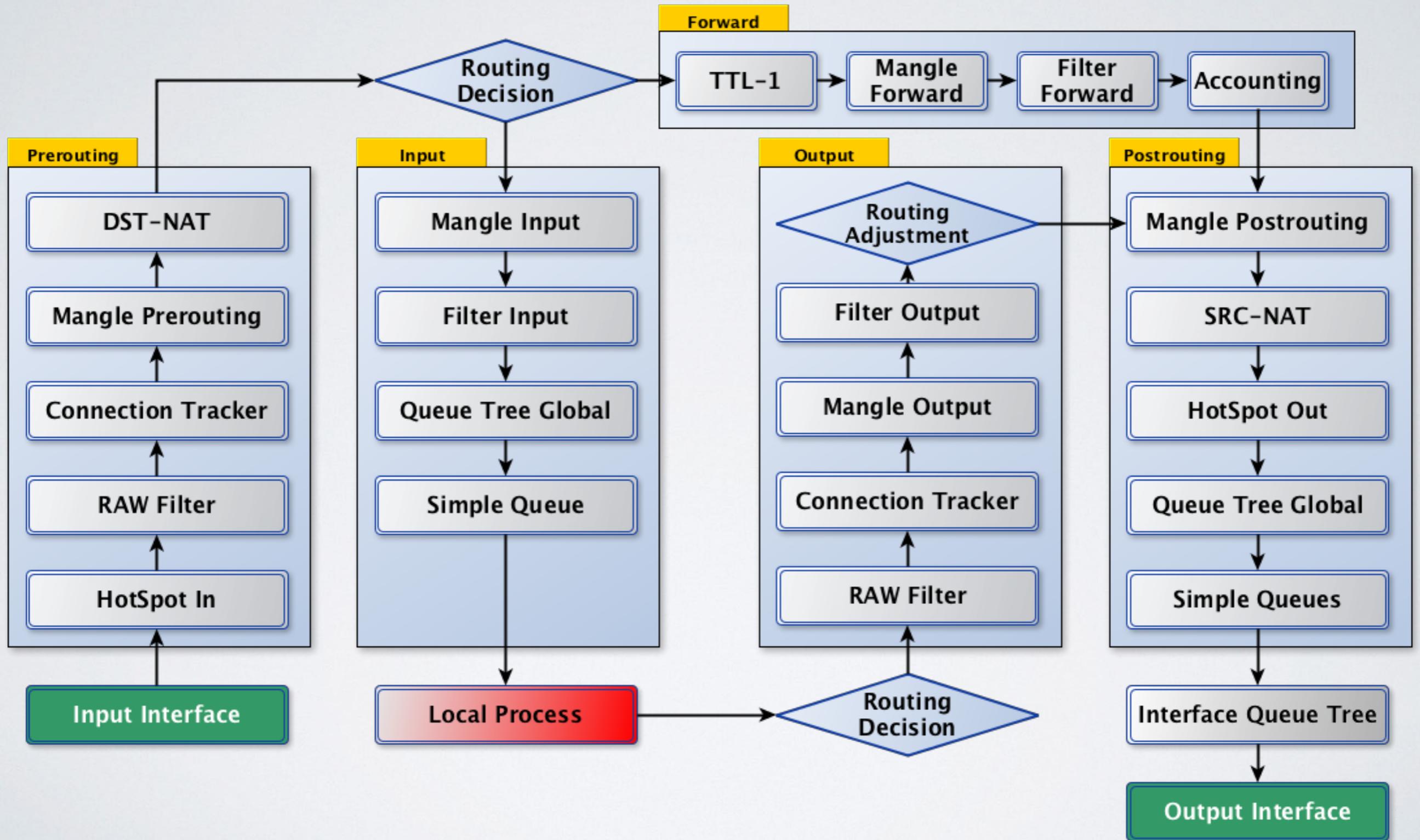
# Доступ к маршрутизатору

- Управление маршрутизатором
  - WinBox, ssh, snmp, icmp etc...
- Нормальная работа VPN
  - Point-to-Point и IP Туннели, а также IPSec
- А также другие сервисы CPU

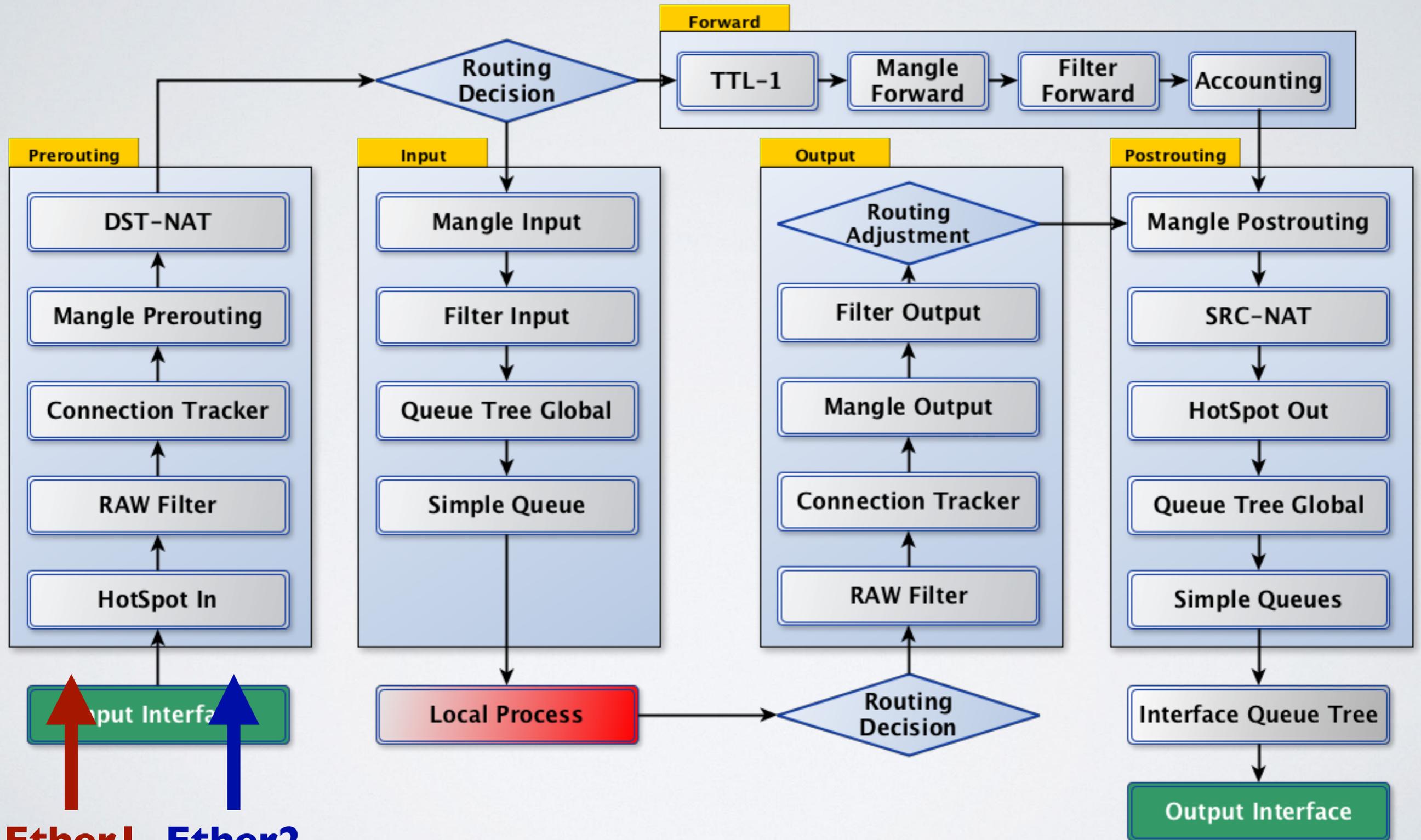
# Доступ к маршрутизатору

- Необходимо обеспечить выход с того же самого интерфейса, с которого пришёл трафик

# Prerouting

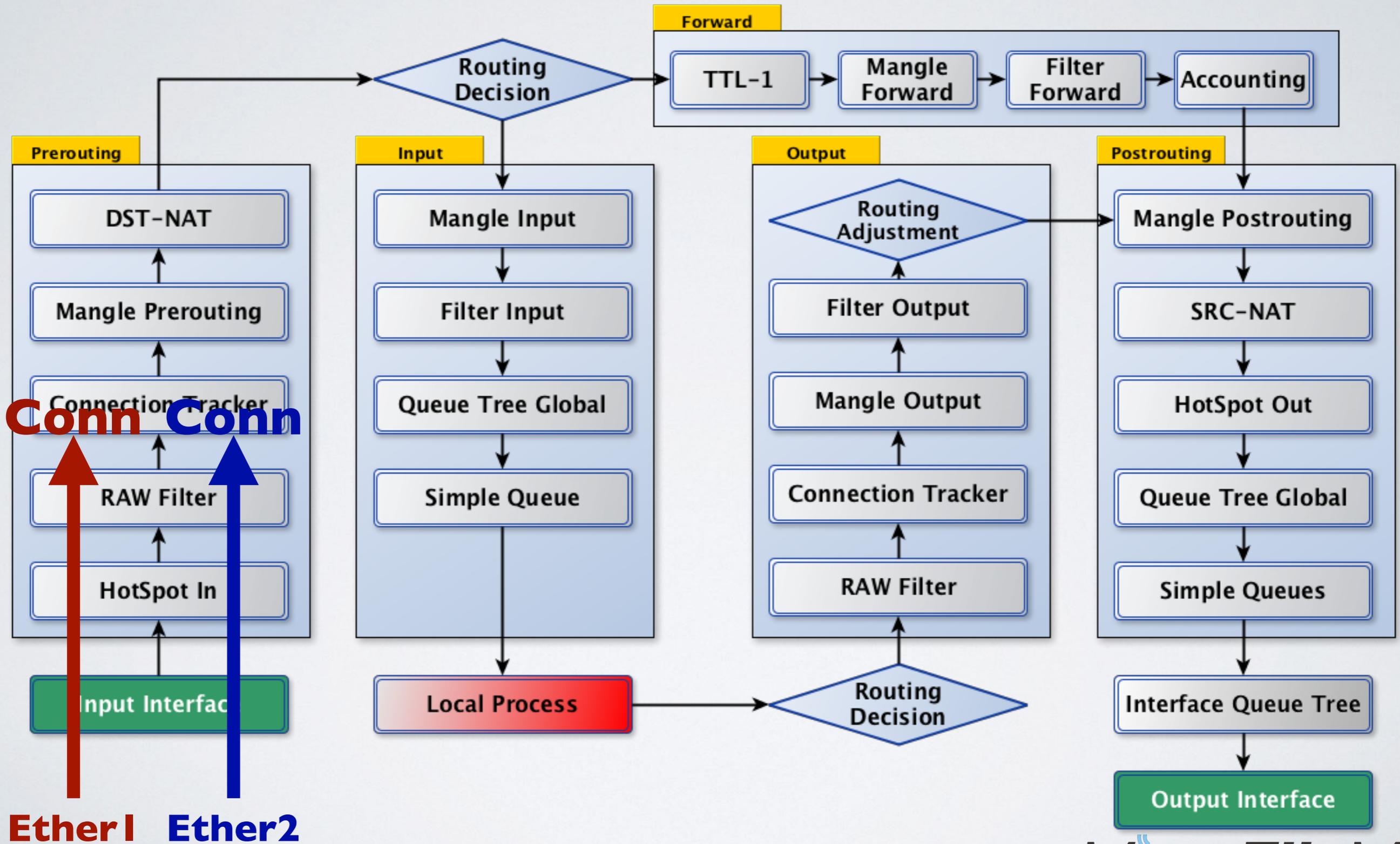


# Prerouting



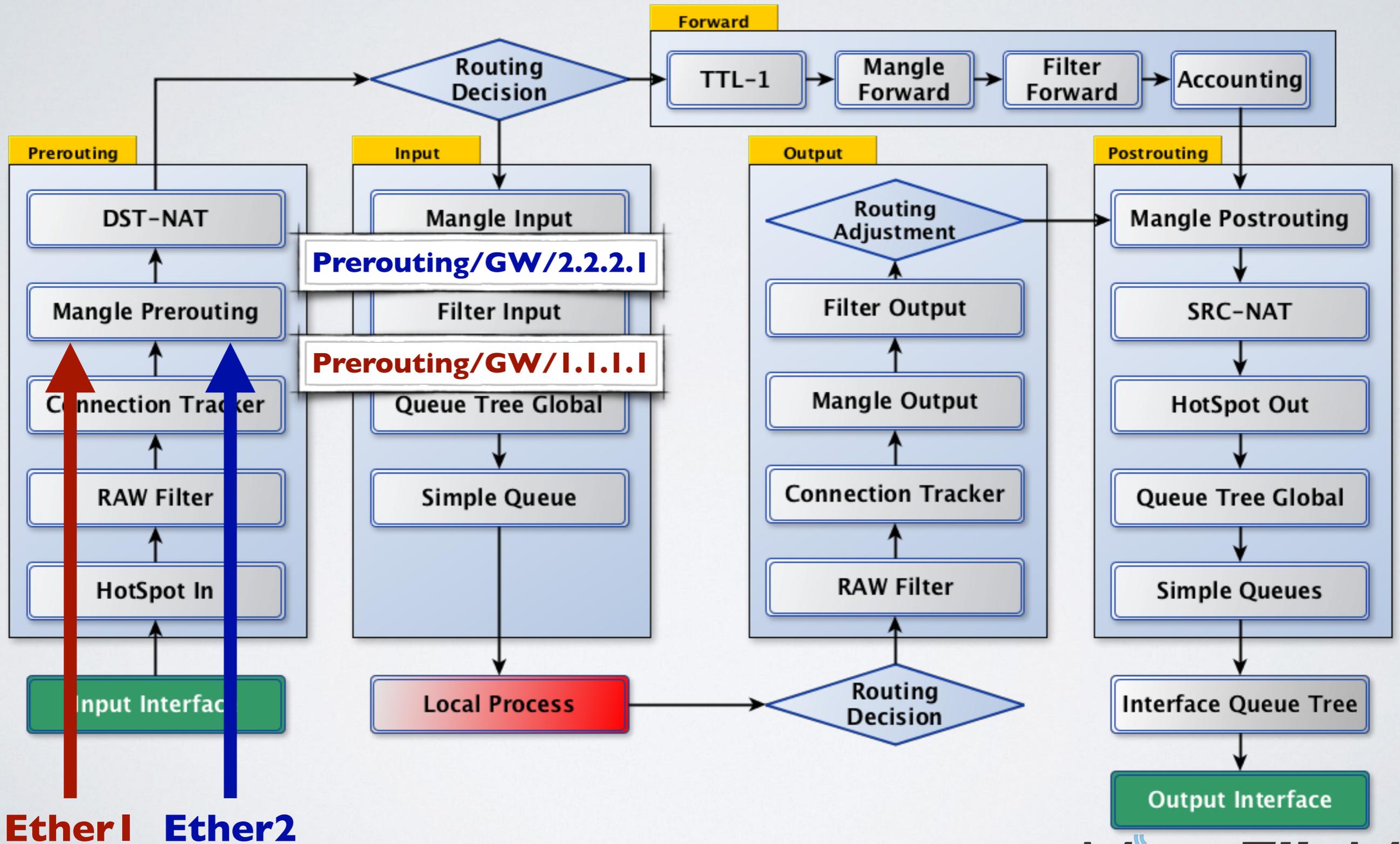
**Ether1** **Ether2**

# Prerouting

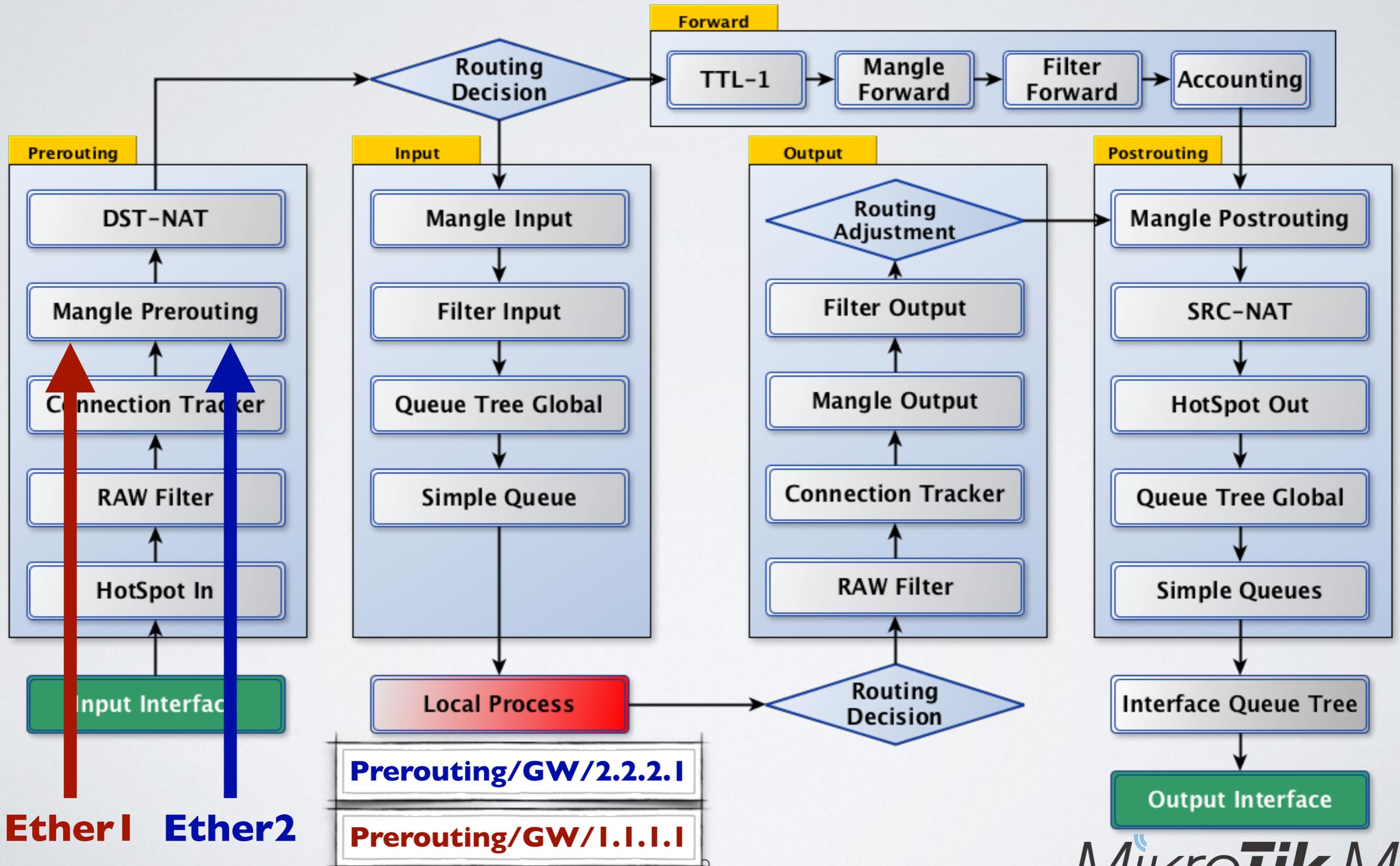


**Ether1 Ether2**

# Prerouting



# Prerouting



# Доступ к маршрутизатору

- Цепочка **Prerouting**
- Маркируйте соединение на входе в маршрутизатор
- Учитывайте в маркировке **каждый шлюз**
- Учитывайте в маркировке интерфейс провайдера
- Только для пакетов **connection-state=new**
- Для удобства именуруйте соединения с учётом IP адреса шлюза
- **«Prerouting/GW/1.1.1.1»**

# Доступ к маршрутизатору

`/ip firewall mangle`

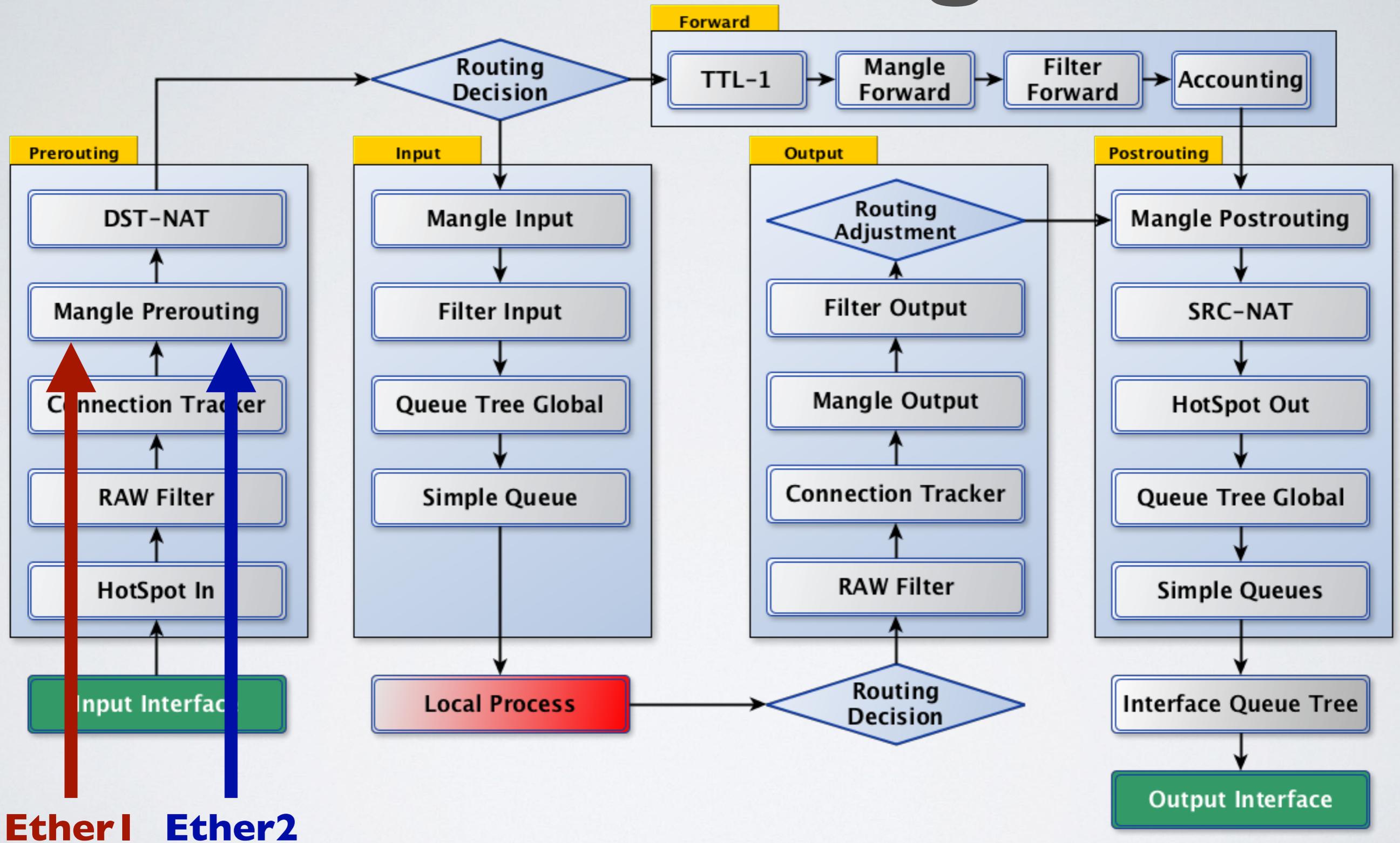
```
add chain=prerouting dst-address=1.1.1.0/29 in-interface=ether1 \  
connection-state=new action=mark-connection \  
new-connection-mark=Prerouting/GW/1.1.1.1 passthrough=no
```

```
add chain=prerouting dst-address=2.2.2.0/29 in-interface=ether2 \  
connection-state=new action=mark-connection \  
new-connection-mark=Prerouting/GW/2.2.2.1 passthrough=no
```

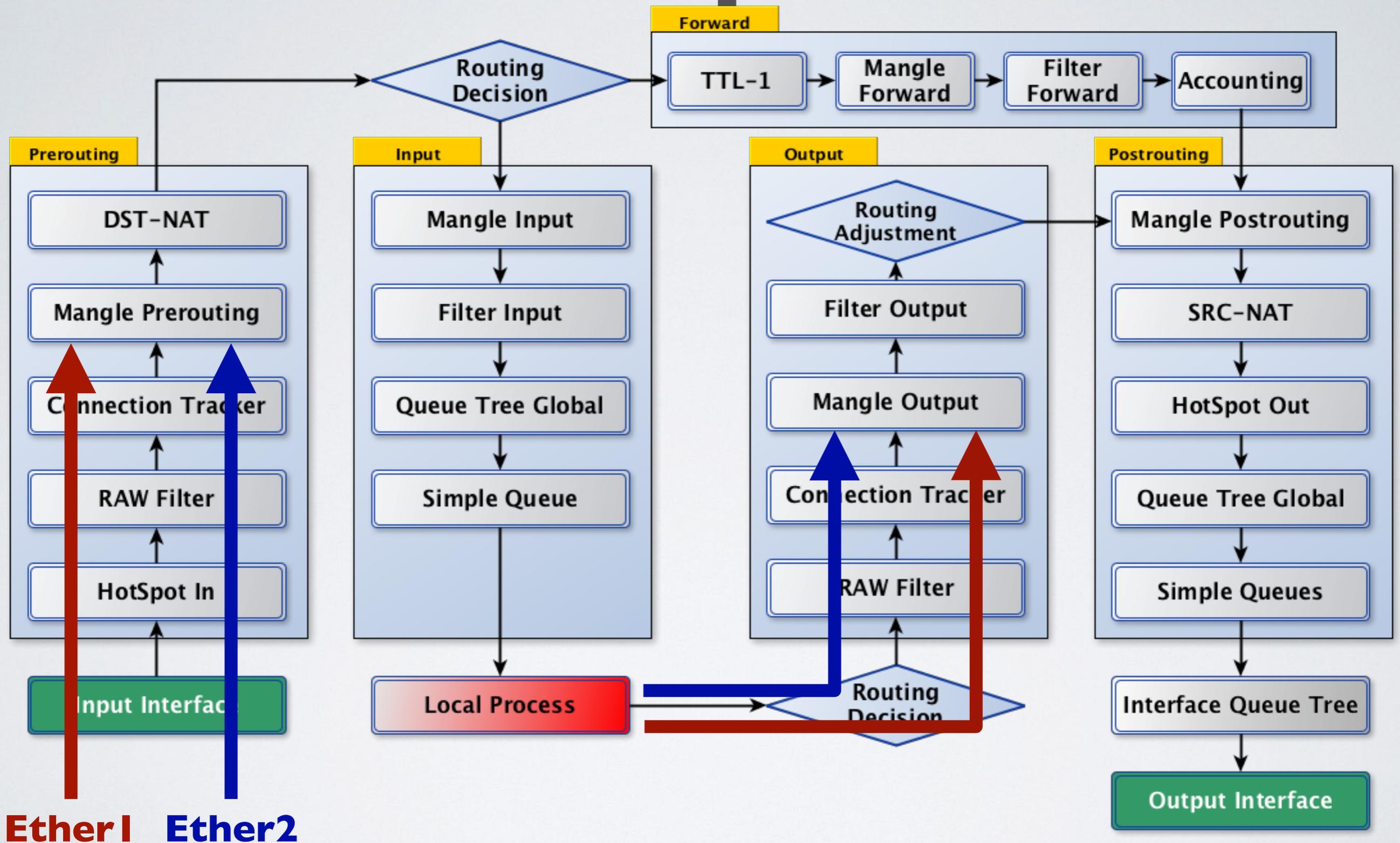
# Доступ к маршрутизатору

- На выходе **из** маршрутизатора, все пакеты в именованном соединении отправляем в отдельную таблицу маршрутизации
- Трафик должен вернуться в тот же интерфейс маршрутизатора, с которого пришёл
- Данный трафик должен уйти с тем же **source** адресом, на который **destination** адрес он пришёл

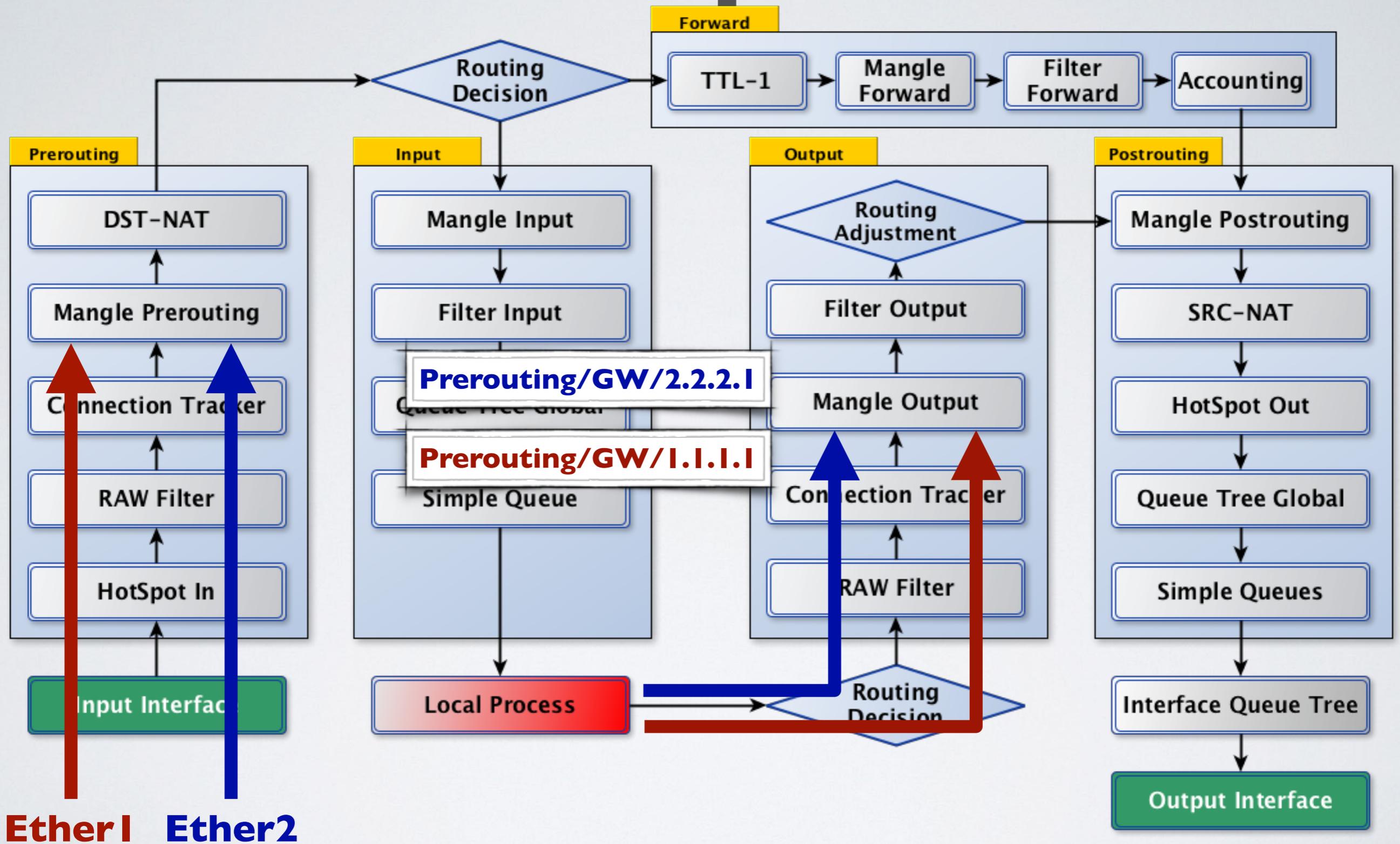
# Prerouting



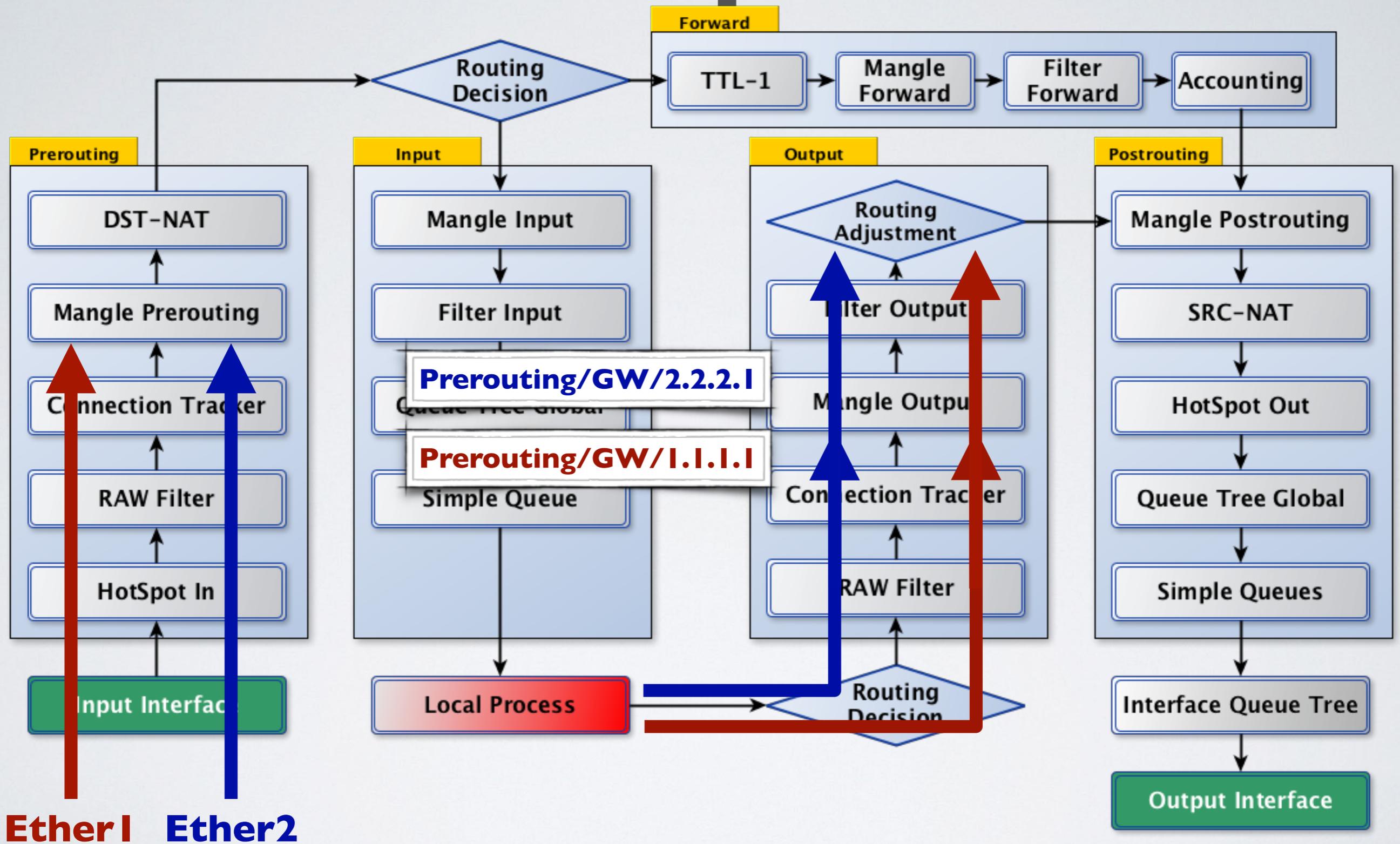
# Output



# Output



# Output



# Доступ к маршрутизатору

```
/ip firewall mangle
```

```
add chain=output connection-mark=Prerouting/GW/1.1.1.1 \  
action=mark-routing new-routing-mark=Next-Hop/1.1.1.1 \  
passthrough=no
```

```
add chain=output connection-mark=Prerouting/GW/2.2.2.1 \  
action=mark-routing new-routing-mark=Next-Hop/2.2.2.1 \  
passthrough=no
```

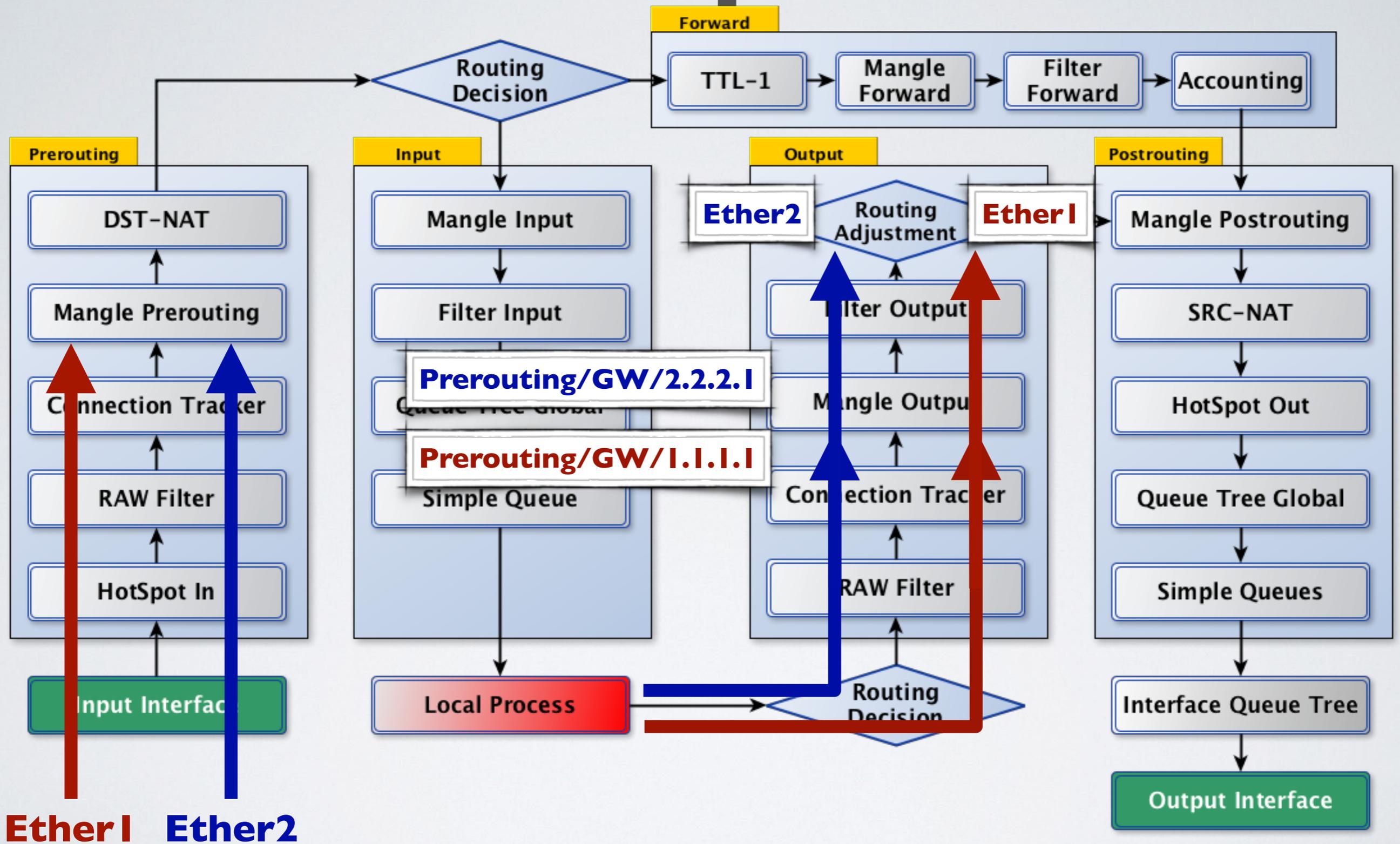
# Доступ к маршрутизатору

- Создайте таблицу маршрутизации, где уникальность маршрута определяется не IP адресом, а адресом nexthop
- Данная таблица должна **«смотреть только в себя»**
- Проверка доступности шлюза **check-gateway** не имеет смысла

# Доступ к маршрутизатору

- Нет необходимости **NAT**ить данный трафик

# Output



# Доступ к маршрутизатору

**/ip route**

**add gateway=1.1.1.1 routing-mark=Next-Hop/1.1.1.1**

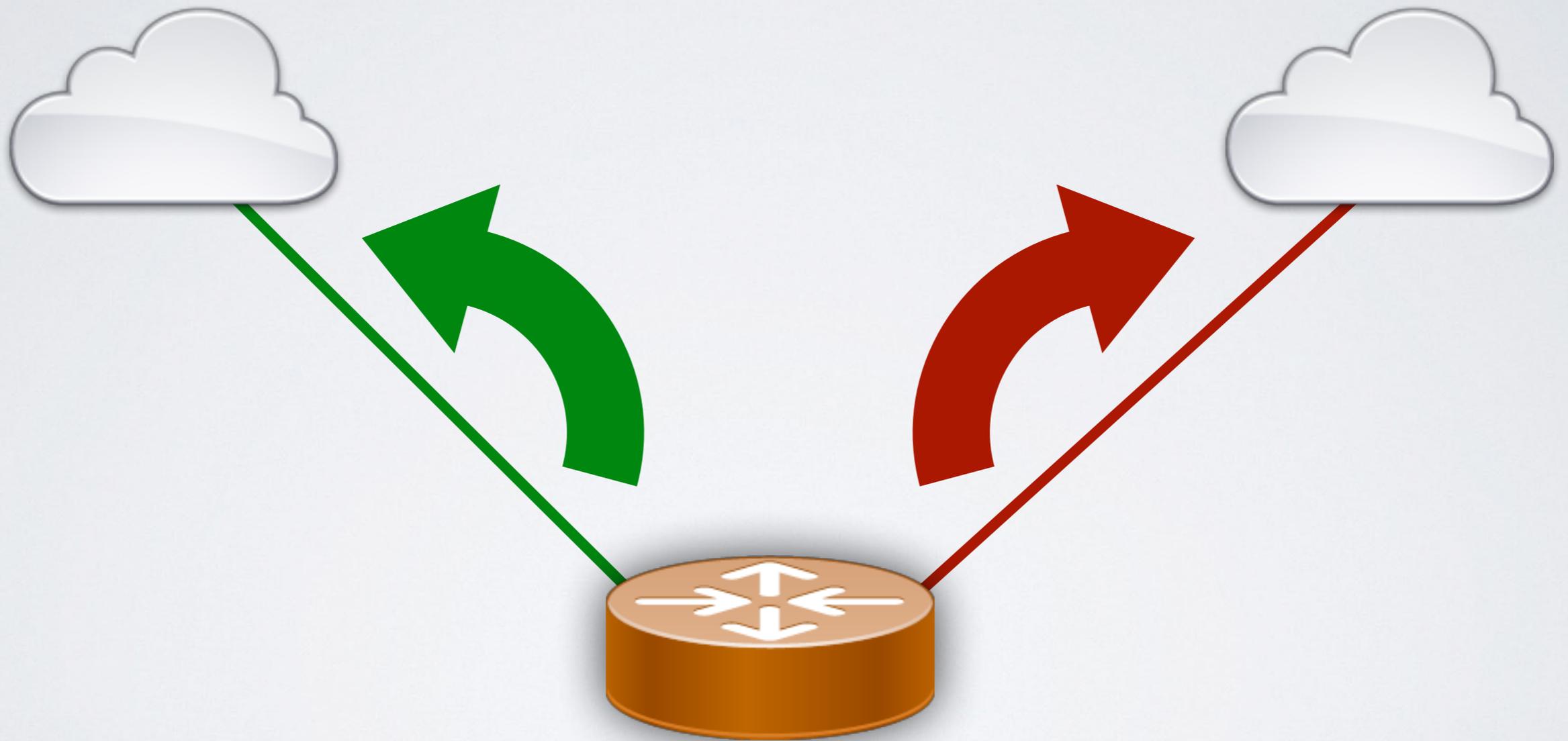
**add gateway=2.2.2.1 routing-mark=Next-Hop/2.2.2.1**

**/ip route rule**

**add action=lookup-only-in-table routing-mark=Next-Hop/1.1.1.1 \  
table=Next-Hop/1.1.1.1**

**add action=lookup-only-in-table routing-mark=Next-Hop/2.2.2.1 \  
table=Next-Hop/2.2.2.1**

# Выход с маршрутизатора



# Выход с маршрутизатора

- Подключение к внешним сервисам с маршрутизатора
- VPN, IP Туннели и IPSec
- Функционал RouterOS, где можно указать source или local адрес

# Выход с маршрутизатора

- Цепочка **Output**
- Нет возможности определить исходящий интерфейс, для определения внутрисетевого трафика
- Используйте префиксы BOGON для **исключения** внутрисетевого трафика
- Учитывайте **каждую** подсеть
- Используйте существующую именную таблицу маршрутизации

# Bogon / RFC5735

0.0.0.0/8	192.0.0.0/24	198.51.100.0/24
10.0.0.0/8	192.0.2.0/24	203.0.113.0/24
127.0.0.0/8	192.88.99.0/24	224.0.0.0/4
169.254.0.0/16	192.168.0.0/16	240.0.0.0/4
172.16.0.0/12	198.18.0.0/15	255.255.255.255/32

# Выход с маршрутизатора

```
/ip firewall mangle
```

```
add chain=output src-address=1.1.1.0/29 dst-address-list=!BOGON \  
    action=mark-routing new-routing-mark=Next-Hop/1.1.1.1 \  
    passthrough=no
```

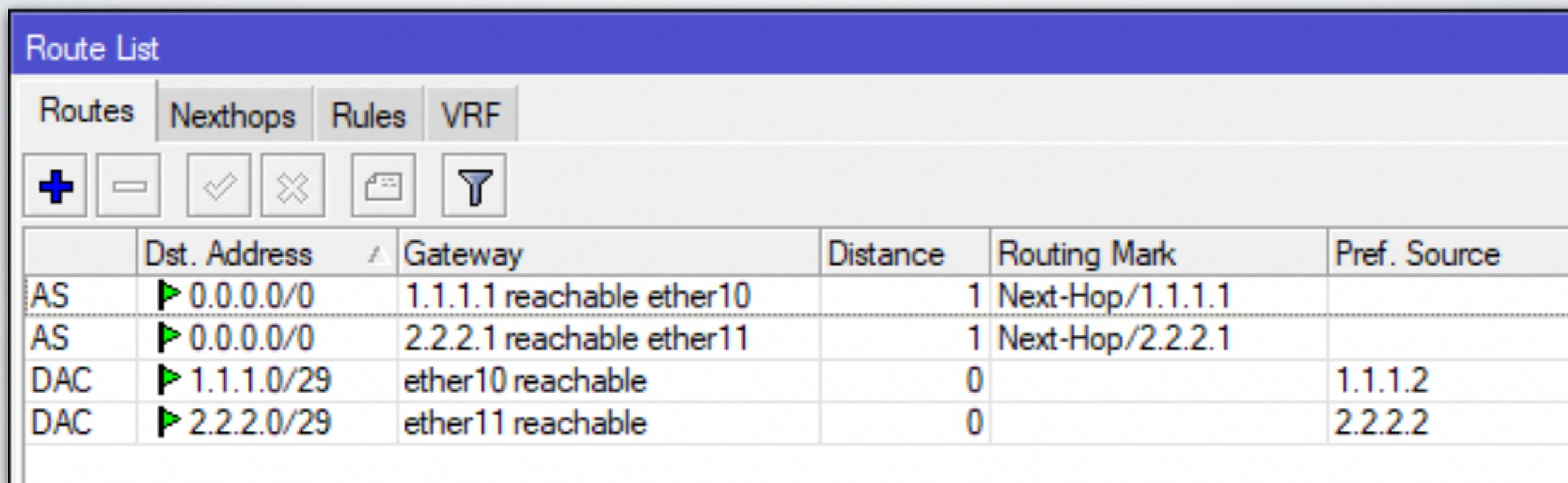
```
add chain=output src-address=2.2.2.0/29 dst-address-list=!BOGON \  
    action=mark-routing new-routing-mark=Next-Hop/2.2.2.1 \  
    passthrough=no
```

# Выход с маршрутизатора

- Нет необходимости **NAT**ить данный трафик

# Выход с маршрутизатора

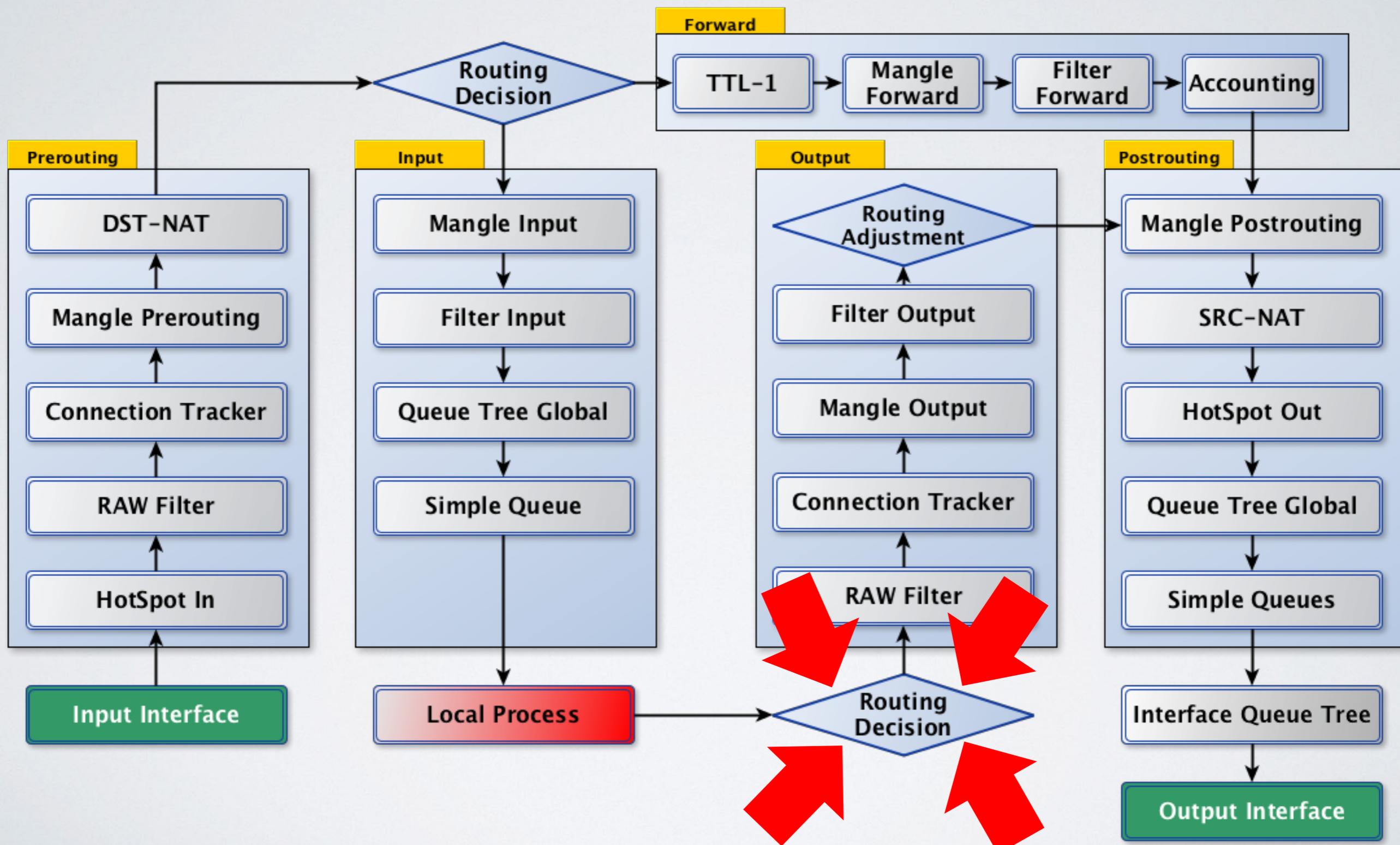
Достаточно ли данных настроек для управления маршрутизатором?



The screenshot shows the 'Route List' window in Mikrotik WinBox. It features a blue header bar with the title 'Route List'. Below the header, there are four tabs: 'Routes' (selected), 'Nexthops', 'Rules', and 'VRF'. A toolbar contains icons for adding (+), subtracting (-), checking (checkmark), deleting (X), refreshing (refresh), and filtering (funnel). The main area is a table with the following columns: 'AS', 'Dst. Address', 'Gateway', 'Distance', 'Routing Mark', and 'Pref. Source'. The table contains four rows of data.

	Dst. Address	Gateway	Distance	Routing Mark	Pref. Source
AS	▶ 0.0.0.0/0	1.1.1.1 reachable ether10	1	Next-Hop/1.1.1.1	
AS	▶ 0.0.0.0/0	2.2.2.1 reachable ether11	1	Next-Hop/2.2.2.1	
DAC	▶ 1.1.1.0/29	ether10 reachable	0		1.1.1.2
DAC	▶ 2.2.2.0/29	ether11 reachable	0		2.2.2.2

# Выход с маршрутизатора



# Выход с маршрутизатора

- Необходим unicast **активный** маршрут в таблице **MAIN**
- Используйте пустой **Bridge**, как **loopback** интерфейс
- Создайте **default route** через **loopback** интерфейс
- Укажите максимальную **distance** для маршрута
- Используйте **pref-src** для определения **src IP** адреса  
остального трафика

# Выход с маршрутизатора

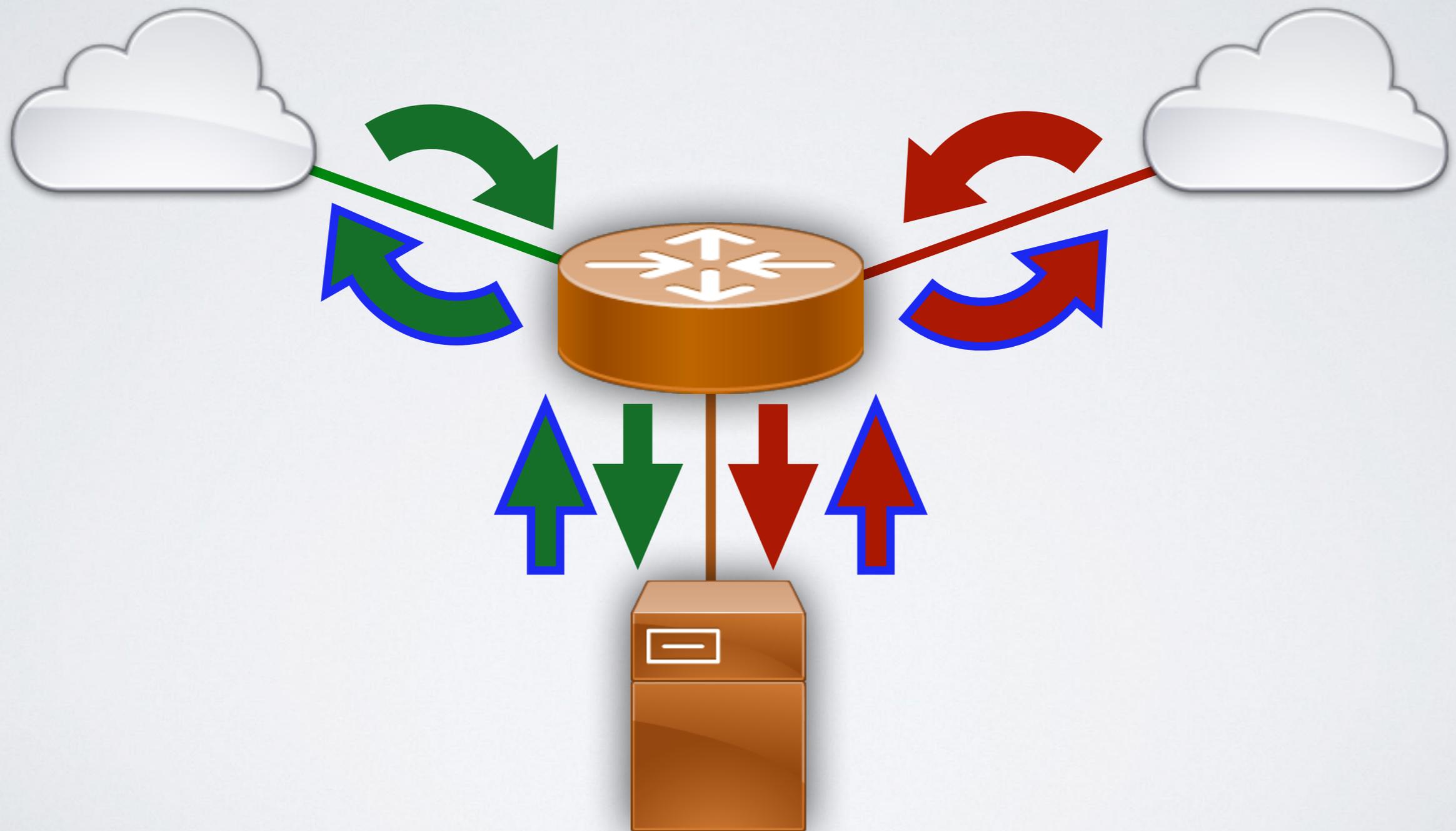
```
/interface bridge
```

```
add name=Br-Loopback
```

```
/ip route
```

```
add gateway=Br-Loopback distance=254 pref-src=1.1.1.2
```

# Доступ «за» NAT



# Доступ «за» NAT

- Одновременный доступ к внутренним сервисам
- CDN, SMTP, WEB, etc...
- **DNS Round Robin**, как вариант распределения внешней нагрузки на каналы, современные браузеры замечательно работают с данной технологией и отрабатывают отказ

# Доступ «за» NAT

Настройте dst NAT так, как обычно это делаете

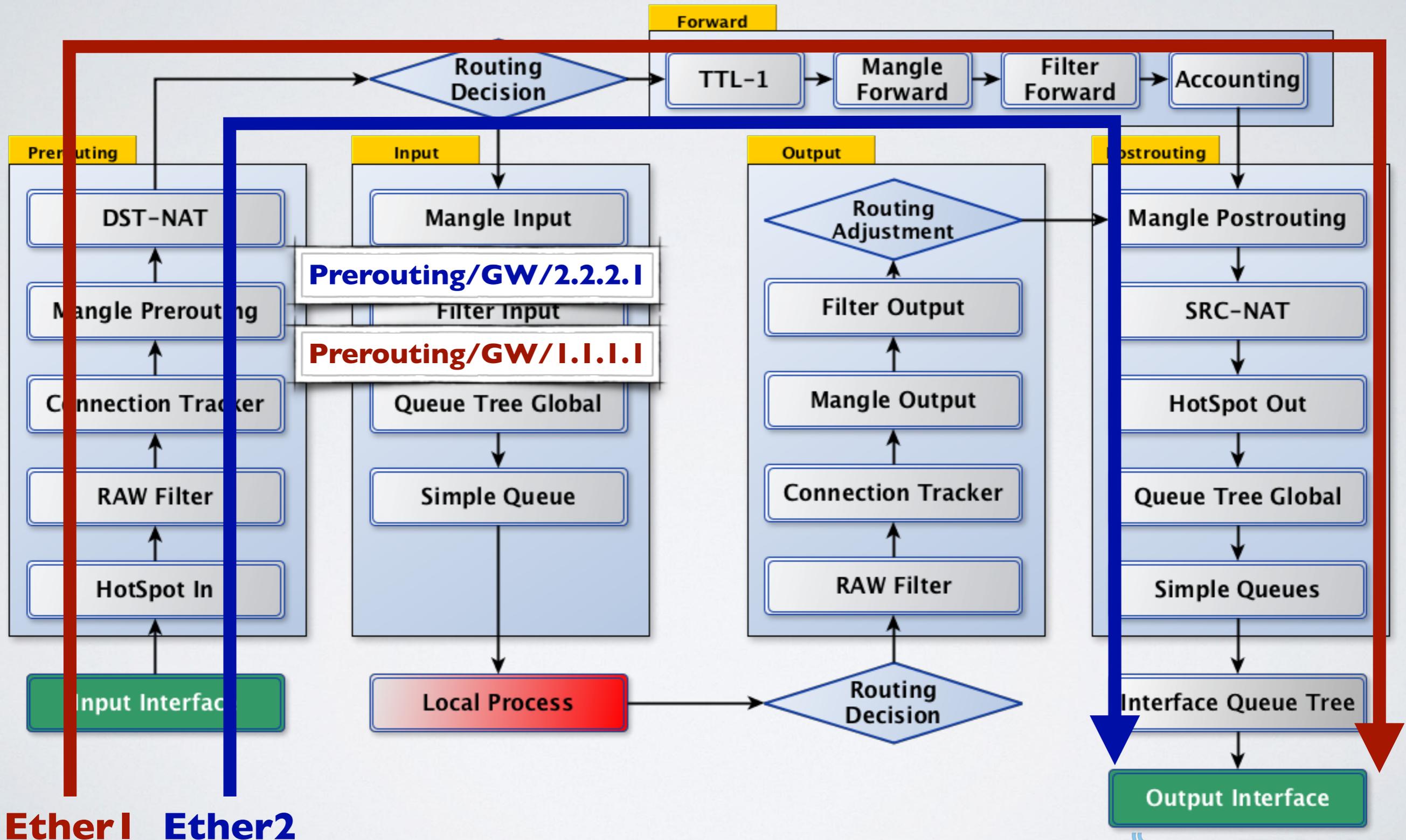
# Доступ «за» NAT

```
/ip firewall nat
```

```
add chain=dstnat dst-address=1.1.1.2 protocol=tcp \  
    dst-port=80,25,443 in-interface=ether1 action=dst-nat \  
    to-addresses=192.168.1.2
```

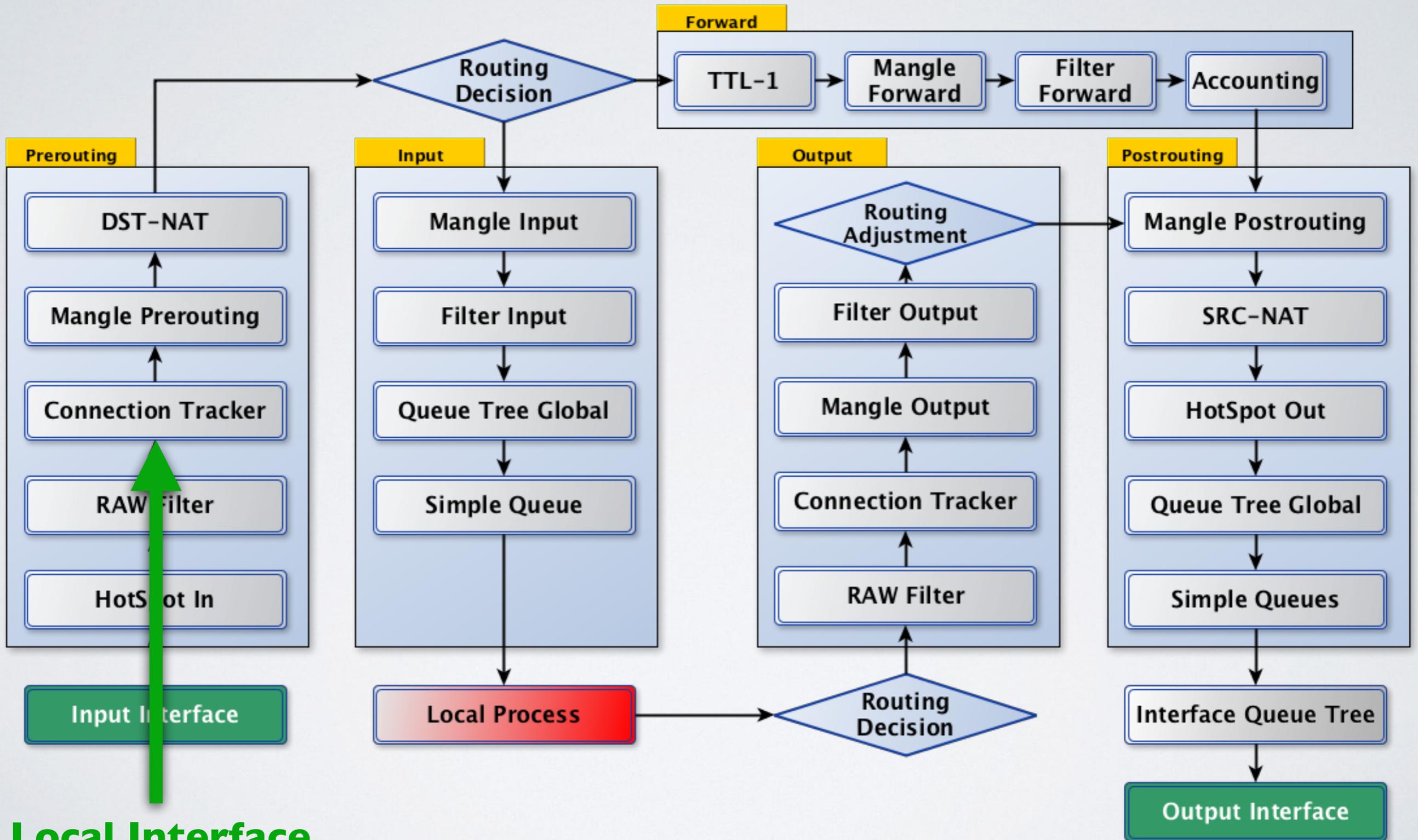
```
add chain=dstnat dst-address=2.2.2.2 protocol=tcp \  
    dst-port=80,25,443 in-interface=ether2 action=dst-nat \  
    to-addresses=192.168.1.2
```

# Forward



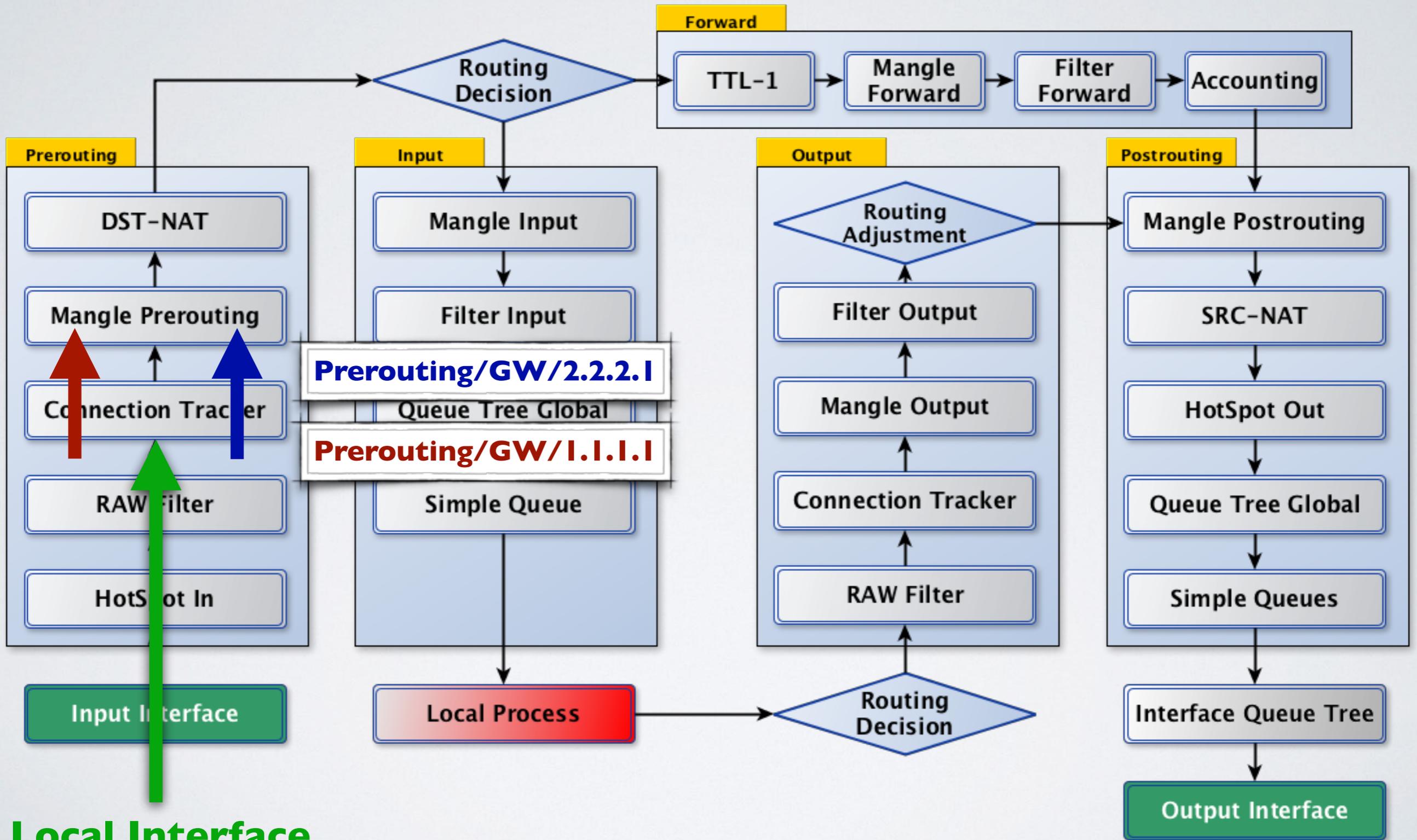
**Ether1** **Ether2**

# Forward



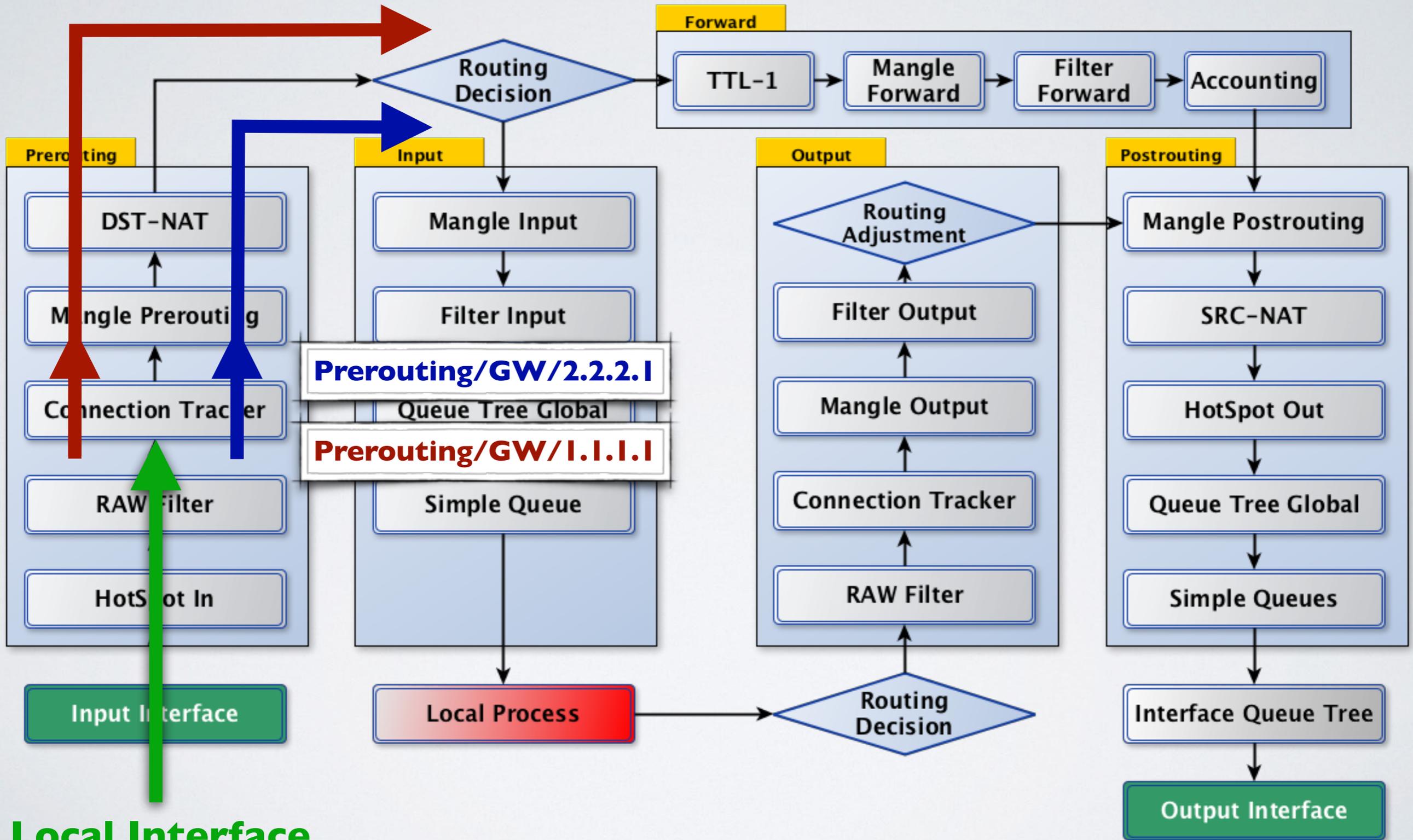
Local Interface

# Forward



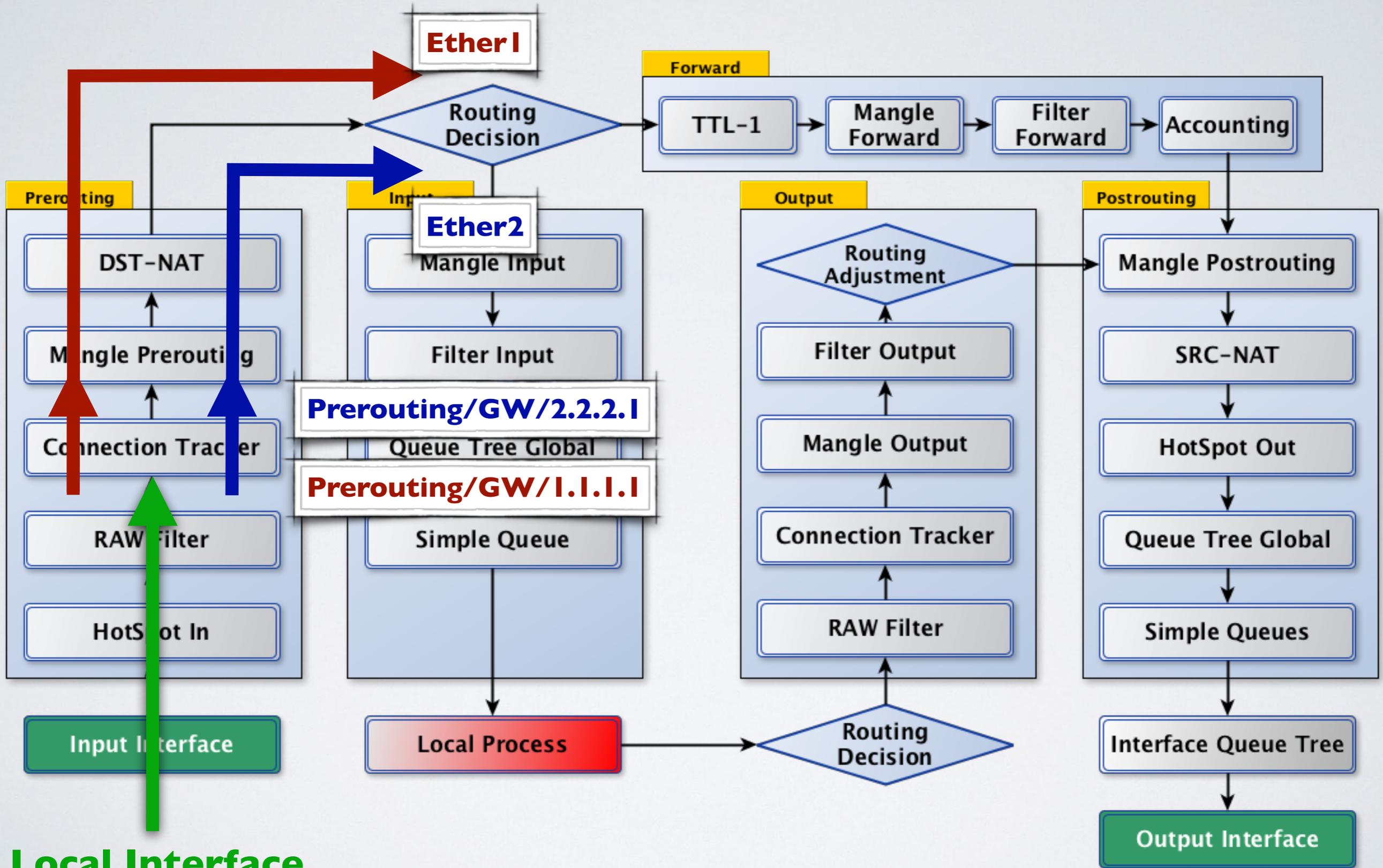
Local Interface

# Forward



Local Interface

# Forward



Local Interface

# Доступ «за» NAT

- Уже существует маркированное соединение
- Цепочка **Prerouting**
- **Исключите** интерфейс провайдера из фильтра
- Отправляется в именную таблицу маршрутизации

# Доступ «за» NAT

```
/ip firewall mangle
```

```
add chain=prerouting in-interface=!ether1 \  
    connection-mark=Prerouting/GW/1.1.1.1 action=mark-routing \  
    new-routing-mark=Next-Hop/1.1.1.1 passthrough=no
```

```
add chain=prerouting in-interface=!ether2 \  
    connection-mark=Prerouting/GW/2.2.2.1 action=mark-routing \  
    new-routing-mark=Next-Hop/2.2.2.1 passthrough=no
```

# Доступ «за» NAT

Нет необходимости **NAT**ить данный трафик

За вас это сделает **connection-tracker**

# Доступ «за» NAT

Connection <5.19.245.3->1.1.1.1:80>

General Statistics

Src. Address: 5.19.245.3:65207

Dst. Address: 1.1.1.1:80

Reply Src. Address: 192.168.0.100:80

Reply Dst. Address: 5.19.245.3:65207

Protocol: 6 (tcp)

Connection Type:

Connection Mark:

Timeout: 23:59:53

TCP State: established

OK

Remove

expected seen reply assured confirmed dying fasttrack srcnat **dstnat**

# Доступ «за» NAT

Connection <5.19.245.3->1.1.1.1:80>

General Statistics

Src. Address: 5.19.245.3:65207

Dst. Address: 1.1.1.1:80

Reply Src. Address: 192.168.0.100:80

Reply Dst. Address: 5.19.245.3:65207

Protocol: 6 (tcp)

Connection Type:

Connection Mark:

Timeout: 23:59:53

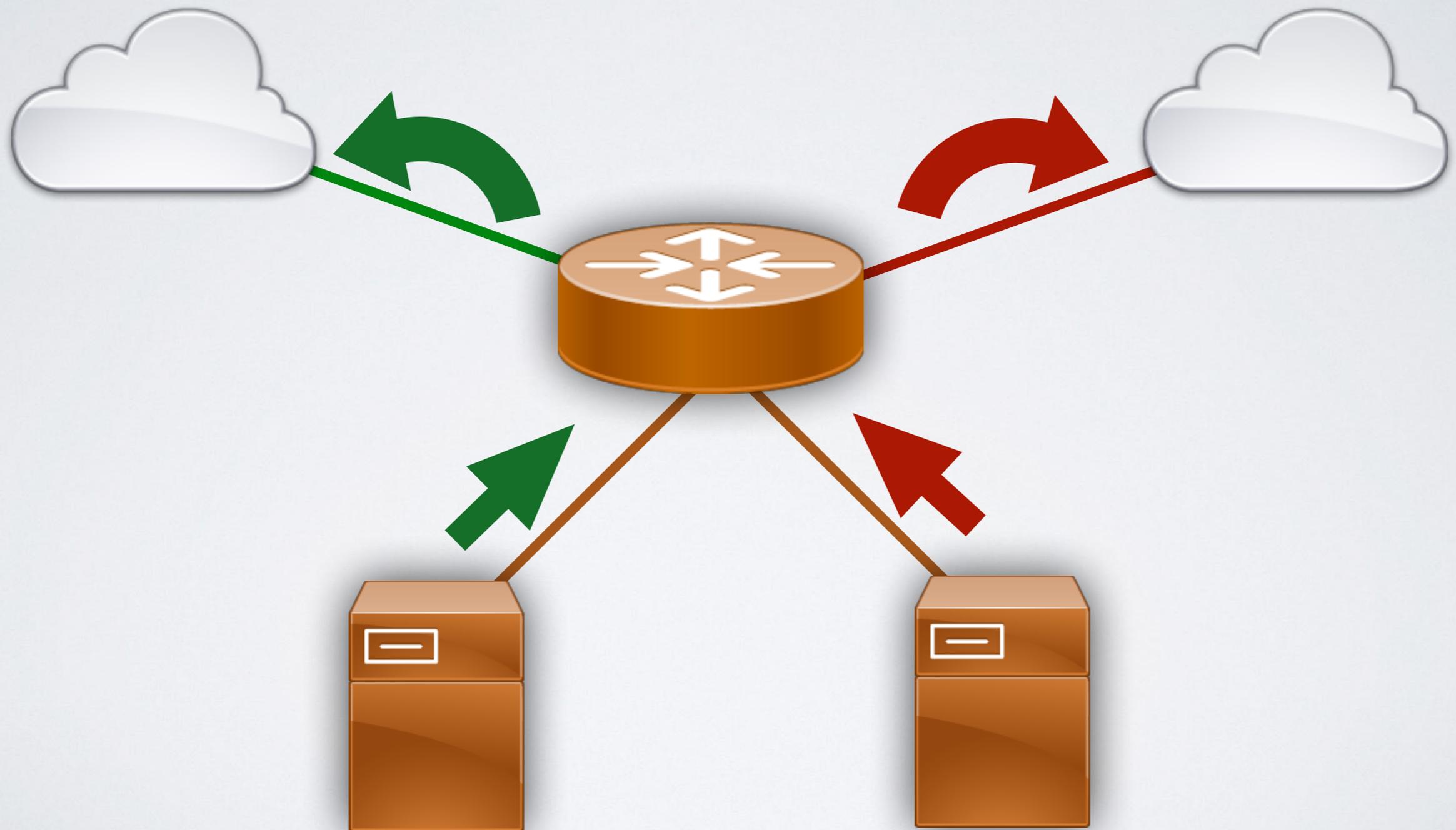
TCP State: established

expected seen reply assured confirmed dying fasttrack srcnat dstnat

OK

Remove

# Предопределить IP Адрес



# Предопределить IP Адрес

- Выход хоста из под определённого IP адреса
- Позволяет организовать работу сложных протоколов или IP зависимых сервисов
- Asterisk, банкинг, IPsec с внутренних хостов etc...

# Предопределить IP Адрес

- Цепочка **Prerouting**
- Используйте **address-list** для определения списка хостов
- Адрес листы именуруйте информативно «**via\1.1.1.2**»
- Количество внутрисетевых интерфейсов, может быть больше чем ОДИН
- Используйте **Bogon** адрес лист для исключения внутрисетевого трафика

# Предопределить IP Адрес

`/ip firewall mangle`

```
add chain=prerouting src-address-list=via/1.1.1.2 \  
dst-address-list=!BOGONS action=mark-routing \  
new-routing-mark=Next-Hop/1.1.1.1 passthrough=no
```

```
add chain=prerouting src-address-list=via/2.2.2.2 \  
dst-address-list=!BOGONS action=mark-routing \  
new-routing-mark=Next-Hop/2.2.2.1 passthrough=no
```

# Предопределить IP Адрес

Необходимо настроить src NAT для данного  
типа трафика

# Предопределить IP Адрес

**/ip firewall nat**

**add chain=srcnat routing-mark=Next-Hop/1.1.1.1 \  
src-address-list=via/1.1.1.2 action=src-nat to-addresses=1.1.1.2**

**add chain=srcnat routing-mark=Next-Hop/2.2.2.1 \  
src-address-list=via/2.2.2.2 action=src-nat to-addresses=2.2.2.2**

# Load Balance

- Распределить нагрузку по каналам
- Обеспечить выход с необходимого IP адреса
- Проверять канал интернета

# Load Balance

- Per-connection-classifier
- Nth
- Random
- ECMP

# Load Balance / ECMP

- Equal-cost multi-path
- Позволяет распределить нагрузку между next-hop
- Очень простой в настройке
- 10 минут хранит в кэше выбор шлюза
- Используйте главную таблицу маршрутизации

# Load Balance / ECMP

**/ip route**

**add gateway=1.1.1.1,2.2.2.1 pref-src=1.1.1.2**

**ИЛИ**

**add gateway=1.1.1.1,2.2.2.1,2.2.2.1 pref-src=1.1.1.2**

# ECMP Cache - 5 Tuple

**src-address**

**dst-address**

**in-interface**

**routing-mark**

**ToS**

# ECMP Cache - 5 Tuple

**src-address**            **192.168.0.100**

**dst-address**

**in-interface**

**routing-mark**

**ToS**

# ECMP Cache - 5 Tuple

**src-address**            **192.168.0.100**

**dst-address**            **5.19.245.3**

**in-interface**

**routing-mark**

**ToS**

# ECMP Cache - 5 Tuple

**src-address**            **192.168.0.100**

**dst-address**            **5.19.245.3**

**in-interface**            **Bridge-local**

**routing-mark**

**ToS**

# ECMP Cache - 5 Tuple

**src-address**      **192.168.0.100**

**dst-address**      **5.19.245.3**

**in-interface**      **Bridge-local**

**routing-mark**      **main**

**ToS**

# ECMP Cache - 5 Tuple

**src-address**      **192.168.0.100**

**dst-address**      **5.19.245.3**

**in-interface**      **Bridge-local**

**routing-mark**      **main**

**ToS**      **8**

# ECMP Cache - 5 Tuple

**src-address**      **192.168.0.100**

**dst-address**      **5.19.245.3**

**in-interface**      **Bridge-local**

**routing-mark**      **main**

**ToS**      **8**

**=hash**

# ECMP Cache - 5 Tuple

hash#1	<b>1.1.1.1 via ether1</b>
hash#2	<b>2.2.2.1 via ether2</b>
hash#3	<b>2.2.2.1 via ether2</b>
hash#4	<b>1.1.1.1 via ether1</b>
hash#5	<b>2.2.2.1 via ether2</b>
hash#6	<b>2.2.2.1 via ether2</b>
hash#7	<b>1.1.1.1 via ether1</b>
hash#8	<b>2.2.2.1 via ether2</b>
hash#9	<b>2.2.2.1 via ether2</b>
hash#10	<b>1.1.1.1 via ether1</b>
hash#11	<b>2.2.2.1 via ether2</b>
hash#12	<b>2.2.2.1 via ether2</b>
hash#13	<b>1.1.1.1 via ether1</b>
hash#14	<b>2.2.2.1 via ether2</b>
hash#15	<b>2.2.2.1 via ether2</b>
hash#16	<b>1.1.1.1 via ether1</b>

**gateway**=1.1.1.1,2.2.2.1,2.2.2.1

**1.1.1.1 = 33%**

**2.2.2.1 = 66%**

# ECMP Cache - 5 Tuple

hash#1	1.1.1.1 via ether1
hash#2	2.2.2.1 via ether2
hash#3	2.2.2.1 via ether2
hash#4	1.1.1.1 via ether1
hash#5	2.2.2.1 via ether2
hash#6	2.2.2.1 via ether2
hash#7	1.1.1.1 via ether1
hash#8	2.2.2.1 via ether2
hash#9	2.2.2.1 via ether2
hash#10	1.1.1.1 via ether1
hash#11	2.2.2.1 via ether2
hash#12	2.2.2.1 via ether2
hash#13	1.1.1.1 via ether1
hash#14	2.2.2.1 via ether2
hash#15	2.2.2.1 via ether2
hash#16	1.1.1.1 via ether1



Каждые 10 Минут



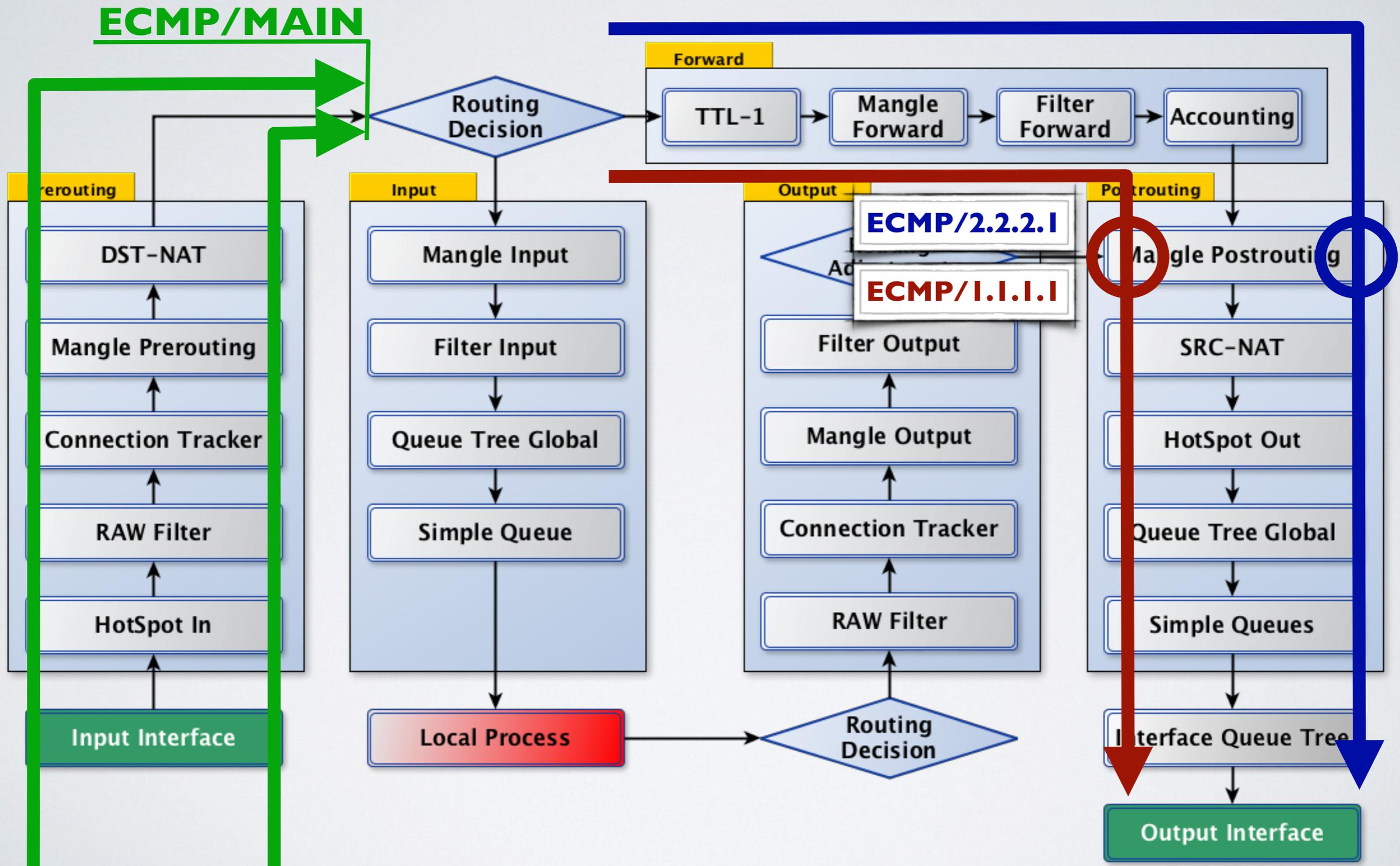
**нельзя просто так взять и настроить  
ЕСМР**

**Через NAT**

# Load Balance / ECMP

Необходимо убрать ограничение 10 минут

# Load Balance / ECMP



# Load Balance / ECMP

- Цепочка **postrouting**
- Только для **connection-state=new**
- Учитывайте таблицу маршрутизации
- Только для соединений без маркировки
- Учитывайте исходящий интерфейс

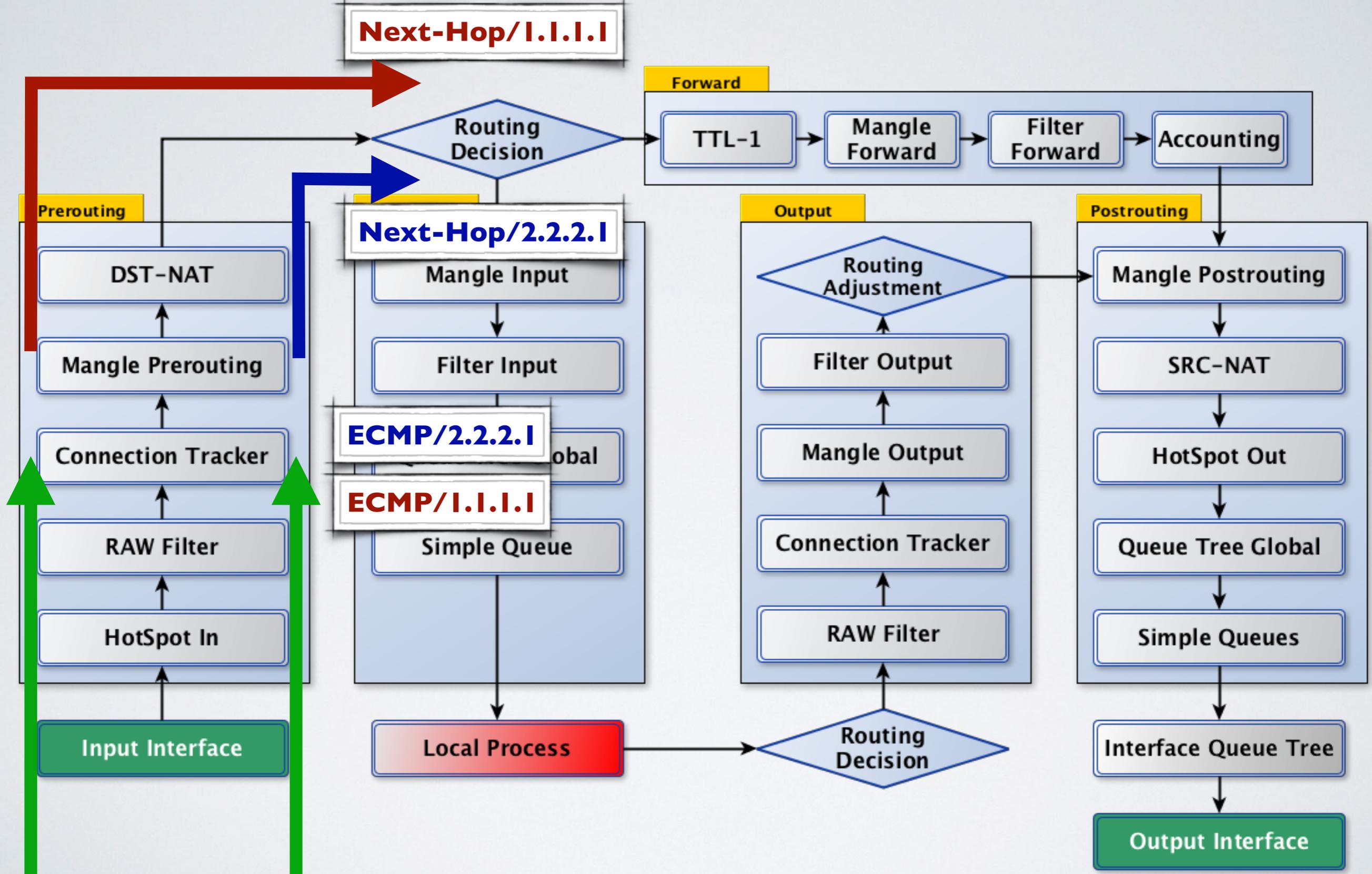
# Load Balance / ECMP

**/ip firewall mangle**

```
add chain=postrouting connection-mark=no-mark \  
connection-state=new out-interface=ether1 routing-mark=main \  
action=mark-connection new-connection-mark=ECMP/1.1.1.2 \  
passthrough=no
```

```
add chain=postrouting connection-mark=no-mark \  
connection-state=new out-interface=ether2 routing-mark=main \  
action=mark-connection new-connection-mark=ECMP/2.2.2.2 \  
passthrough=no
```

# Load Balance / ECMP



# Load Balance / ECMP

- Цепочка **Prerouting**
- Только для маркированных соединений
- Используйте ранее созданную таблицу маршрутизации

# Load Balance / ECMP

**/ip firewall mangle**

```
add chain=prerouting connection-mark=ECMP/1.1.1.2 \  
    action=mark-routing new-routing-mark=Next-Hop/1.1.1.1 \  
    passthrough=no
```

```
add chain=prerouting connection-mark=ECMP/2.2.2.2 \  
    action=mark-routing new-routing-mark=Next-Hop/2.2.2.1 \  
    passthrough=no
```

# Load Balance / ECMP

Необходимо настроить src NAT для данного типа трафика

# Load Balance / ECMP

```
/ip firewall nat
```

```
add action=src-nat chain=srcnat connection-mark=ECMP/1.1.1.2 \  
to-addresses=1.1.1.2
```

```
add action=src-nat chain=srcnat connection-mark=ECMP/2.2.2.2 \  
to-addresses=2.2.2.2
```

# Вопросы?

- MikroTik.me
- VasilevKirill.com
- <https://t.me/mikrotikme>
- <https://vk.com/mikrotikrus>