grigoryev.victor@gmail.com vmg.pp.ua labs.mikrotik.com.ua

Григорьев Виктор

Днепропетровский национальный университет
Кафедра ЭВМ
Факультет физики, электроники и компьютерных систем

VPN уровня 2 и 3

grigoryev.victor@gmail.com vmg.pp.ua labs.mikrotik.com.ua

1. Техника работы в виртуальной лаборатории

Демонстрация производится с помощью виртуальной сетевой лаборатории в виде контейнера виртуальных машин GNS3.

GNS3 установлен на компьютере (Intel Core i5-3550 3.30GHz 32Gb) под управлением Ubuntu 10.0.4 64 разряда с именем домена lib.mikrotik.com.ua . Предоставлен Дмитрием Лукиным , Ультратех.

Для доступа к удалённому рабочему столу Ubuntu используем технологию NX фирмы NoMachine, основанную на X-протоколе и протоколе SSH. Для удалённого доступа к рабочему столу Ubuntu на нём поднят freeNX-сервер. Для удалённого доступа к рабочему столу Ubuntu используем NX Client for Windows фирмы NoMachine.

Виртуальные машины в GNS3 работают под управлением RouterOS фирмы Mikrotik. В роли менеджера виртуальных машин выступает Qemu.

Для связи хост-машины с виртуальными машинами в хост-машине созданы tap-интерфейсы, помещённые в мост

openvpn --mktun --dev tap

brctl addbr m

brctl addif m tap

ifconfig tap up

ifconfig m 10.X.X.2 netmask 255.255.255.0 up

Единожды для понимания работы GNS3 запустим qemuwrapper. Запустим GNS3. Создадим первую топологию **first** (**Рис.1.1**)

Рис. 1.1. Первая топология

Добавим в роутеры опции
options = -net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap000
options = -net nic,vlan=6 -net tap,script=no,downscript=no,vlan=6,ifname=tap001

Сохраним проект (топологию)
autostart = False
[Qemu 127.0.0.1:10525]

workingdir = working **– рабочая папка - обязательная строка**
udp = 30700
[[QemuDevice]]
image = /home/user0/LIC/mikrotik-5.21.img **-измените при переносе с другой машины**
netcard = e1000
kvm = True **- эта строка есть, если только работаете на чистом железе с установленным kvm**
[[QEMU R0]] **– начало конфигурации для R0**
console=2700
e0 = R1 e1 **-связь**
[[QEMU R1]] **– начало конфигурации для R1**
console=3701e1 = R0 e0 **-связь**
[GNS3-DATA]
workdir = working **- рабочая папка-обязательная строка**

Запустим топологию. Для топологии first смотрим в окно консоли предварительно запущенного Qemuwrapper. Для каждого маршрутизатора в

топологии GNS делает для mikrotik-5.2.img разностные образы с именем FLASH. SWAP – это образ диска для свопинга:

Formatting '/home/user0/projects/First/working/R0/FLASH', fmt=qcow2 size=103809024 backing_file='/home/user0/LIC/mikrotik-5.21.img' encryption=off cluster_size=0
Formatting '/home/user0/projects/First/working/R0/SWAP', fmt=qcow2 size=1073741824 encryption=off cluster_size=0

Берём содержимое консоли в карман и отформатируем согласно правилам записи командной строки Qemu (см. http://wiki.йemu.org/Manual). MAC-адреса мы не указываем и убираем несоединённые интерфейсы. Получим

qemu -name R0 -m 256 /home/ user0**/projects/first/working/R0/FLASH -hdb /home/user0/projects/first/working/R0/SWAP -net nic,vlan=0 –net udp,vlan=0,sport=27002,dport=27003,daddr=127.0.0.1 -serial telnet:127.0.0.1:3700,server,nowait -net nic,vlan=6 -net nic -net tap,script=no,downscript=no,vlan=6,ifname=tap000**
qemu -name R1 -m 256 /home/user0**/projects/first/working/R1/FLASH -hdb /home/** user0**/projects/first/working/R1/SWAP -net nic,vlan=1 -net**

udp,vlan=1,sport=27003,dport=27002,daddr=127.0.0.1 -serial telnet:127.0.0.1:3701,server,nowait -net nic,vlan=6 -net nic -net tap,script=no,downscript=no,vlan=6,ifname=tap001

Если ввести эти команды в консоли ubuntu, то получим тот же эффект, что и в GNS3. Делать это не обязательно. Если решитесь, то лучше для этого создайте командный файл.

Строки **-serial telnet:127.0.0.1:3700,server,nowait и -serial telnet: 127.0.0.1:3701, server,nowait** отвечают за связь с консолью.

Для R0 строка

-net nic,vlan=0 -net udp,vlan=0,sport=27002,dport=27003,daddr=127.0.0.1

говорит, что все пакеты из интерфейса e0 (ether1) поступают на UDP порт 27002 хост-машины ubuntu и интерфейс e0 (ether1) принимает все пакеты из UDP порта 27003 хост-машины.

Для R1 наоборот. Строка

-net nic,vlan=1 -net udp,vlan=1,sport=27003,dport=27002,daddr=127.0.0.1

говорит, что все пакеты из интерфейса e0 (ether0) поступают на UDP порт 27003

хост-машины ubuntu и интерфейс e0 (ether0) принимает все пакеты из UDP порта 27002 хост-машины.

Таким образом сетевой Ethernet-кабель моделируется двунаправленным UDP-каналом с использованием двух портов **27002** и **27003**. Ethernet-адаптеру поставлен в соответствие двусторонний UDP-канал

И, наконец, строки

-net nic,vlan=6 -net nic –net tap,script=no,downscript=no,vlan=6,ifname=tap000

-net nic,vlan=6 -net nic -net tap,script=no,downscript=no,vlan=6,ifname=tap001

приводят к созданию внутри маршрутизатора сетевой карточки ether7 которая связывается с сетевым tap-интерфейсом с именем tap000 (tap001).

Тап-устройства имеют двоякую природу. С одной стороны это сетевые интерфейсы, а с другой стороны это устройство ввода вывода /dev/net/tun. Сетевые пакеты, пришедшие на Тап-интерфейс можно прочитать как данные из устройства /dev/net/tun. Данные записанные в устройство /dev/net/tun исходят из Тап-интерфейса в виде сетевых пакетов.

Заметим о соотношении названий интерфейсов в GNS и RouterOS

GNS	**RouterOs**

e0	ether1
e1	ether2
...	

В принципе можно обойтись и без GNS. Однако для сложных топологий легко запутаться. Остановите все маршрутизаторы (красный квадрат). Сохраните топологию.

R0

ip address add address=10.0.0.1/24 interface=ether7

ip route add dst-address=10.0.0.0/16 gateway=10.0.0.2
ip address add address=1.1.1.1/24 interface=ether1

R1

ip address add address=10.0.1.1/24 interface=ether7

ip route add dst-address=10.0.0.0/16 gateway=10.0.1.2
ip address add address=1.1.1.2/24 interface=ether1

Проверим связь по ICMP и SSH. Запустим winbox.

Сделаем резервную копию makeGetExport 0 1

```
#!/bin/bash
for ((i=$1;i<=$2;i++)) ;do
ssh admin@10.0.$i.1 "export compact file=E$i"
sftp admin@10.0.$i.1:/E*
done
```

Готовим полученные файлы E0.rsc и E1.rsc для импорта

E0.rsc

```
# nov/12/2012 19:12:15 by RouterOS 5.21
# software id = 18P9-S49L
/ip address
#add address=10.0.0.1/24 interface=ether7
add address=1.1.1.1/24 interface=ether1
/ip route
add distance=1 dst-address=10.0.0.0/16 gateway=10.0.0.2
/system identity
set name=R0
```

E1.rsc

```
# nov/12/2012 19:12:16 by RouterOS 5.21
# software id = 18P9-S49L
#
/ip address
#add address=10.0.1.1/24 interface=ether7
add address=1.1.1.2/24 interface=ether1
/ip route
add distance=1 dst-address=10.0.0.0/16 gateway=10.0.1.2
/system identity
set name=R1
```

Помещаем в архив файлы topology.net, E0.rsc и E1.rsc. На другом компьютере распаковываем архив в папку firstCopy и помещаем туда папку working. Возможно, изменяем в файле topology.net путь к образу RouterOs и имена тап-интерфейсов. Запускаем топологию. Появятся в папке working папки R0 и R1. Останавливаем топологию.

Должен присутствовать шаблон с достаточным количеством роутеров и с назначенными на тап-интерфейсы адресами. В папке working шаблона

подпапка QEMU0 содержит роутер у которго на тап-интерфейс назначен адрес 10.0.0.1/24,

подпапка QEMU1 содержит роутер у которго на тап-интерфейс назначен адрес 10.0.1.1/24,

подпапка QEMU2 содержит роутер у которго на тап-интерфейс назначен адрес 10.0.2.1/24,

…

Заменяем содержимое папки R0 содержимым папки QEMU0 и содержимое папки R1 содержимиым папки QEMU1. Стартуем топологию firstCopy.

Восстанавливаем конфигурацию

Import 0 1

```
#!/bin/bash
for ((i=$1;i<=$2;i++)) ;do
scp E$i.rsc admin@10.0.$i.1:
ssh admin@10.0.$i.1 "import E$i"
done
```

Без пароля консоль R0 можно вызвать так

ssh admin@10.0.0.1

Удобно сделать закладки для одновременной работы с несколькими роутерами

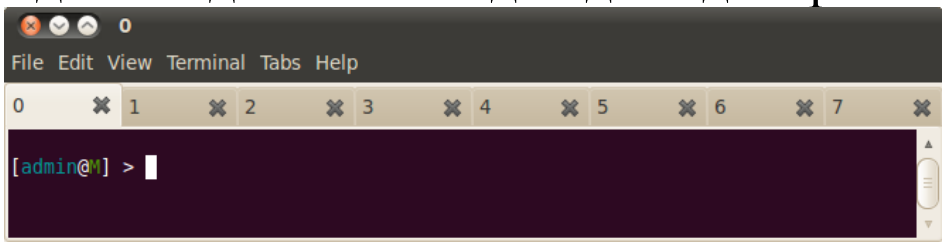

Рис. 1.2. Закладки

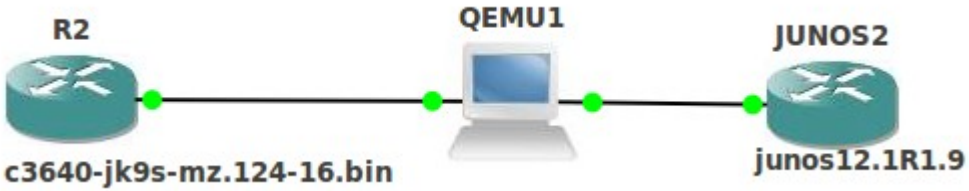

Рис. 1.3. Cisco + RouterOS + Juniper

2. EoIP - Ethernet через IP. VPN уровня 2

Мосты

Mikrotik RouterOs поддерживает объединение Ethernet-портов в мост. Объединяя несколько Ethernet-портов в мост, мы получим на этих портах программный коммутатор второго уровня или свич. Tap-интерфейсы на стороне Ubuntu лежат в мостах. Здесь и дальше свичи – это роутеры Mikrotik, которым мы предадим вид свича (пункт symbol в контекстном меню).

2.1 EoIP

Обычно Ethernet-пакеты ходят по проводам, радиоканалам или оптическому волокну. Ничего не мешает им ходить внутри IP-пакетов. Туннелирование Ethernet через IP (Ethernet over IP - EoIP) - это протокол MikroTik RouterOS, который создаёт Ethernet-туннель между двумя маршрутизаторами поверх IP-соединения. EoIP-тунель может работать через любое соединение, способное передавать IP-пакеты: Ethernet, IPIP-туннель, PPP и т.д.

Если в двух маршрутизаторах настроена поддержка EoIP, то Ethernet-трафик (все Ethernet протоколы) пойдут через интерфейс EoIP так же, как если бы существовали физические Ethernet -интерфейсы и был проложен кабель между маршрутизаторами.

EoIP позволяет с помощью моста объединить через Интернет локальные сети. Протокол EoIP подобно протоколу PPTP инкапсулирует Ethernet-фреймы в GRE

пакеты (протокол IP № 47) и пересылает их на удалённую сторону EoIP-туннеля. Значение mtu=1500 не следует менять для избегания перефрагментации пакета внутри туннеля (wiki.mikrotik.com). Это позволяет так прозрачно объединить Ethernet-сети с помощью моста, что станет возможным транспорт полноразмерных Ethernet-фреймов через туннель.

При использовании мостов для EoIP-тунелей для корректной работы алгоритмов мостов строго рекомендуется следить за уникальностью MAC-адрес для каждого туннеля. Иначе вы должны быть уверены в уникальности хостов, подсоединённых к мосту.

2.2. EoIP VPN уровня 2

Соберём топологию, указанную на рисунке Рис.2.1. (tap-сеть 0)

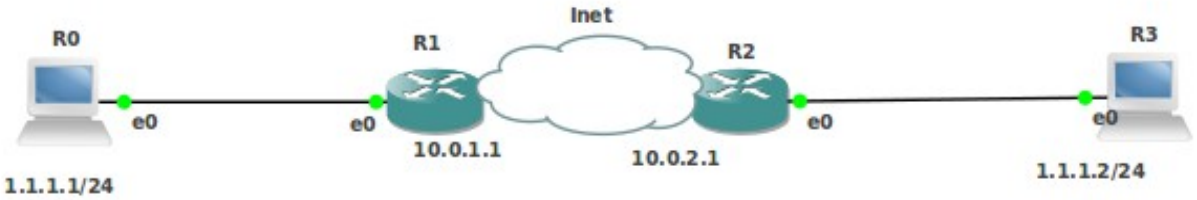

Рис.2.1. Топология Eoip. Два сайта подключены к Интернет через маршрутизаторы R0 и R3. Маршрутизаторы R0 і R3 представляют локальные сети сайтов.

Облако Inet – это просто картинка, символизирующая нашу модель Интернета. Модель заключается в соединении роутеров через tap-интерфейсы хост-машины Ubuntu.

Назначим имена маршрутизаторам. Проверим соседей. R1 не будет видеть R2. Это нормально. Проложим маршрут между tap-сетями 10.0.1.0/24 и 10.0.2.0/24 роутеров R1 и R2 через нашу модель Интернета.

[admin@R1]>**ip route add dst-address=10.0.2.0/24 gateway=10.0.1.2**

[admin@R2]>**ip route add dst-address=10.0.1.0/24 gateway=10.0.2.2**

Здесь 10.0.1.2 и 10.0.2.2 - адреса tap-интерфейсов на стороне хост-машины Ubuntu. Проверим

[admin@R2] > **ping 10.0.1.1**

10.0.1.1 56 63 11ms

Пинги пошли. R1 и R2 соединены через нашу модель Интернета. Настроим EoIP-туннель

[admin@R1] > **int eoip add remote-address=10.0.2.1 disabled=no**

[admin@R2] > **int eoip add remote-address=10.0.1.1 disabled=no**

Создадим на R1 и R2 мосты и добавим в них физический Ethernet-интерфейс e0 (ether1) и интерфейс EoIP

[admin@R1] > **int bridge add**

[admin@R1]> **int bridge port add bridge=bridge1 interface=eoip-tunnel1**

[admin@R1]> **int bridge port add bridge=bridge1 interface=ether1**
[admin@R2] > **int bridge add**
[admin@R2]> **int bridge port add bridge=bridge1 interface=eoip-tunnel1**
[admin@R2]>**int bridge port add bridge=bridge1 interface=ether1**
Посмотрим MAC-адреса
[admin@R0] > **interface Ethernet print**
0 R ether1 1500 00:AA:00:3C:37:00 enabled
[admin@R3] **> interface Ethernet print**
0 R ether1 1500 **00:AA:00:C3:F2:00** enabled
Выведем таблицы, показывающие мостам на какой интерфейс направлять пакет с определённым MAC-адресом
[admin@R1] > **int bridge host pr**

```
bridge1        00:AA:00:C3:F2:00 eoip-tunnel1    11s    ( MAC-адреса  R3)
bridge1        00:AA:00:3C:37:00 ether1          57s    ( MAC-адреса  R0 )
```

…
[admin@R2] > **int bridge host pr**

```
bridge1        00:AA:00:C3:F2:00 ether1          20s    ( MAC-адреса  R3)
bridge1        00:AA:00:3C:37:00 eoip-tunnel1    21s    ( MAC-адреса  R0)
```

Смотрим соседей
[admin@R0] > **ip neighbor print**

R0 видит все роутеры R1, R2 и R3.
[admin@R3] > **ip neighbor print**
R3 видит все роутеры R0, R1 и R2.

Есть связь на втором сетевом уровне. Окончательно убедимся в этом. Например, на R0 с помощью специальной утилиты mac-telnet соединимся по Ethernet с R3, введя MAC-адрес его Ethernet-интерфейса ether1
[admin@R0] **tool mac-telnet 00:AA:00:C3:F2:00**
Попадаем в R3. Возврат CtrlD
На R3 соединимся по Ethernet с R0
[admin@R3] > **tool mac-telnet 00:AA:00:3C:37:00**
Попадаем в R0. Возврат CtrlD
Мы построили виртуальную частную сеть второго уровня над существующей сетью в виде модели Интернета для Ubuntu.
Для полноты картины назначьте IP-адреса для R0 и R1 согласно рисунку и пропингуйте крайние роутеры друг из друга.

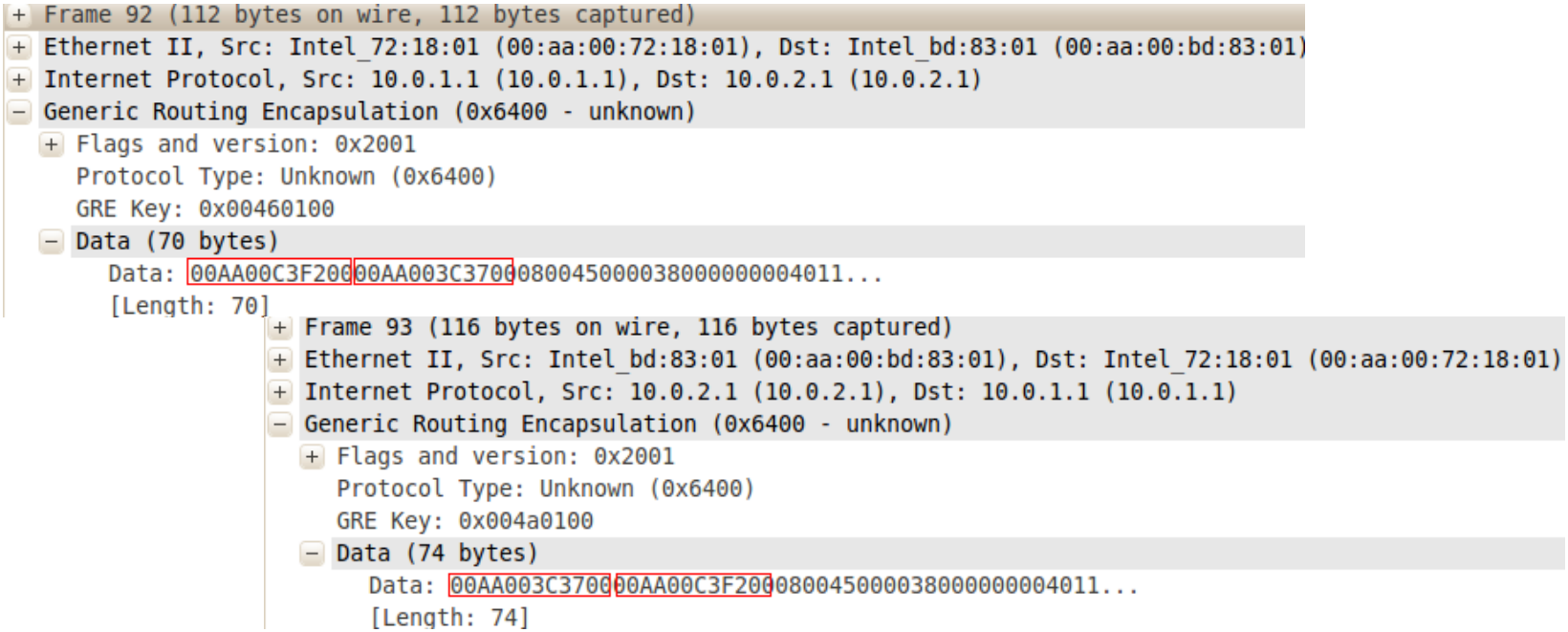

Рис. 2.2 Структура EoIP-пакету.

Выполним команду

[admin@R0] > **ping 00:AA:00:C3:F2:00**

и с помощью анализатора сетевых пакетов Wireshark увидим (**рис. 2.2),** что для инкапсуляции Ethernet-пакета в IP-пакет используется протокол GRE (Generic Routing Encapsulation).

Для топологии EoIP3, изображённой на Рис. 2.3, организetv VPN уровня 2 с помощью EoIP. Используем разные ***tunnel-id.***

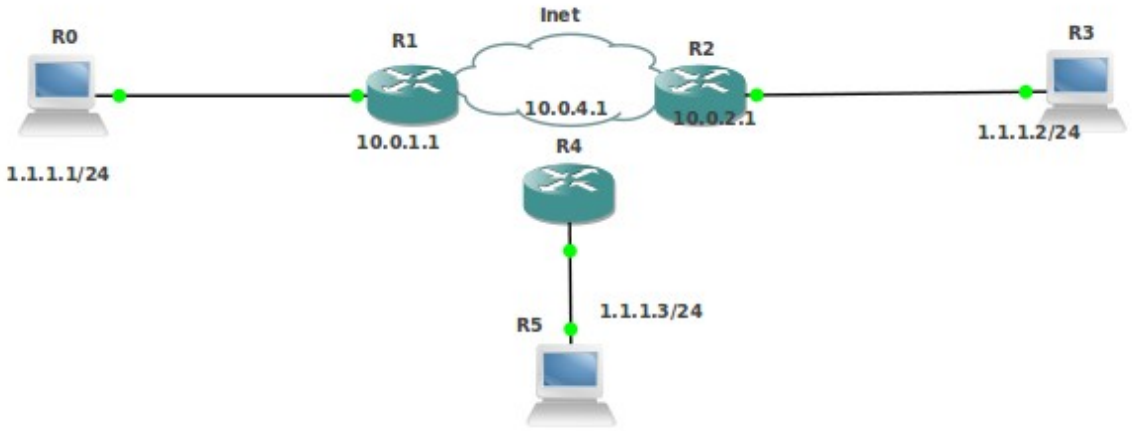

Рис.2.3. Топологии EoIP3

С помощью команд **int bridge host pr** посмотрите таблицы, показывающие мостам на какой интерфейс направлять пакет с определённым MAC-адресом. Убедитесь командой **ip neighbour print**, что все маршрутизаторы видит друг друга как соседа. С помощью команды **tool mac-telnet** окончательно убедитесь, что есть связь на втором сетевом уровне. Назначаем IP-адреса на R0, R3 и R5 согласно рисунку. Пингуем маршрутизаторы. Заметим, что R0, R3 и R5 будут находится в одном домене широковещания. Это позволяет успешно функционировать широковещательным ARP-запросам: от одного ко всем (через EoIP туннели через Интернет)

2.3. EoIP VPN уровня 2 через NAT

Рассмотрим случай, когда филиалы корпорации не имеют прямого выхода в Интернет. Рассмотрим топологию EoIPNAT, изображённую на Рис. 2.4. Здесь R4 и

R5 – маршрутизаторы Интернет-провайдера. Он их конфигурирует по запросу корпорации. Соберите топологию и проверьте соседей.

В начале, согласно рисунку назначьте для R1 и R4 адреса из сети 192.168.1.0/24. Для R2 назначьте шлюз 192.168.1.1. Назначьте для R2 и R5 адреса из сети 192.168.2.0/24. Для R2 назначьте шлюз 192.168.2.1. Назначьте на интерфейсы ethert7 роутеров R4 и R5 дополнительные адреса 10.0.4.22.24 и 10.0.5.22/24 соответственно. Настройте NAT для исходящих и приходящих адресов.

Должны получить нечто подобное

[admin@R4] > **ip firewall nat pr**
Flags: X - disabled, I - invalid, D - dynamic
 0 chain=srcnat action=masquerade out-interface=ether7
 1 chain=dstnat action=dst-nat to-addresses=192.168.1.2 dst-address=10.0.4.22
[admin@R5] > **ip firewall nat print**
Flags: X - disabled, I - invalid, D - dynamic
 0 chain=dstnat action=dst-nat to-addresses=192.168.2.2 dst-address=10.0.5.22
 1 chain=srcnat action=masquerade out-interface=ether7

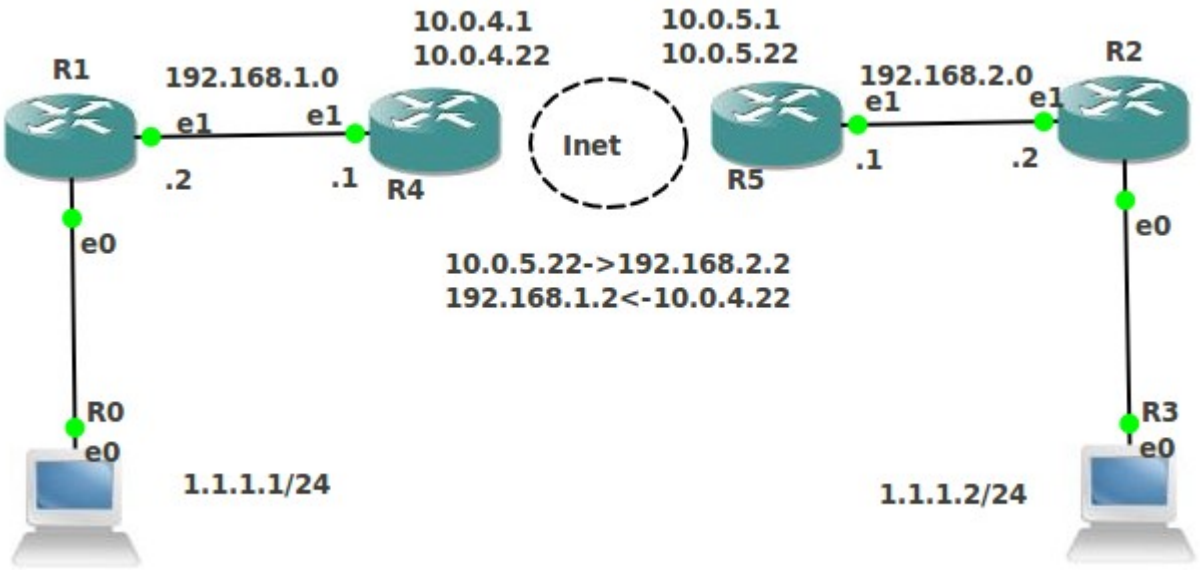

Рис. 2.4. Топология EoIP3

Проверьте работу NAT: роутеры R2 и R6 должны видеть (пинговать) друг друга. Помним, что для внешнего мира они имеют адреса 10.0.3.22 и 10.0.4.22.

Настроим EoIP-туннель между R1 и R2 особым образом, учитывая NAT

[admin@R1] > **int eoip add remote-address=10.0.5.22 disabled=no**

[admin@R2] > **int eoip add remote-address=10.0.4.22 disabled=no**

Создадим на R1 и R2 мосты, добавим в них интерфейс EoIP и физический Ethernet интерфейс e0 (ether1), идущий в сторону R0 (R3).

[admin@R1] > **int bridge add**
[admin@R1]>**int bridge port add bridge=bridge1 interface=eoip-tunnel1**
[admin@R1]>**int bridg port add bridge=bridge1 interface=ether1**
[admin@R2]>**int bridge add**
[admin@R2]>**int bridge port add bridge=bridge1 interface=eoip-tunnel1**
[admin@R2]>**int bridg port add bridge=bridge1 interface=ether1**

Самостоятельно посмотрите таблицы, показывающие мостам на какой интерфейс направлять пакет с определённым MAC-адресом

[admin@R1] > **int bridge host pr**
[admin@R2] > **int bridge host pr**

Убедитесь командой **ip neighbour print**, что крайний маршрутизатор R0 (R3) видит R3 (R0) как соседа. С помощью команды **/tool mac-telnet o**кончательно убедитесь, что есть связь на втором сетевом уровне между R0 и R5. Назначаем IP-адреса на R0 и R3 согласно рисунку. Пингуем из R0 роутер R5 или наоборот.

3. PPP L2 VPN

3.1. Распределённый мост

Возьмём протокол PPTP. Для остальных протоколов конфигурация аналогична. Соберём топологию, изображённую на рис. 3.1. Назначим адреса согласно этому рисунку. Во всех маршрутизаторах R0, R1, R4 и R5 добавим мосты и в них Ethernet-интерфейс, идущий к подсоединённому компьютеру R2, R3, R6 и R7, соответственно.

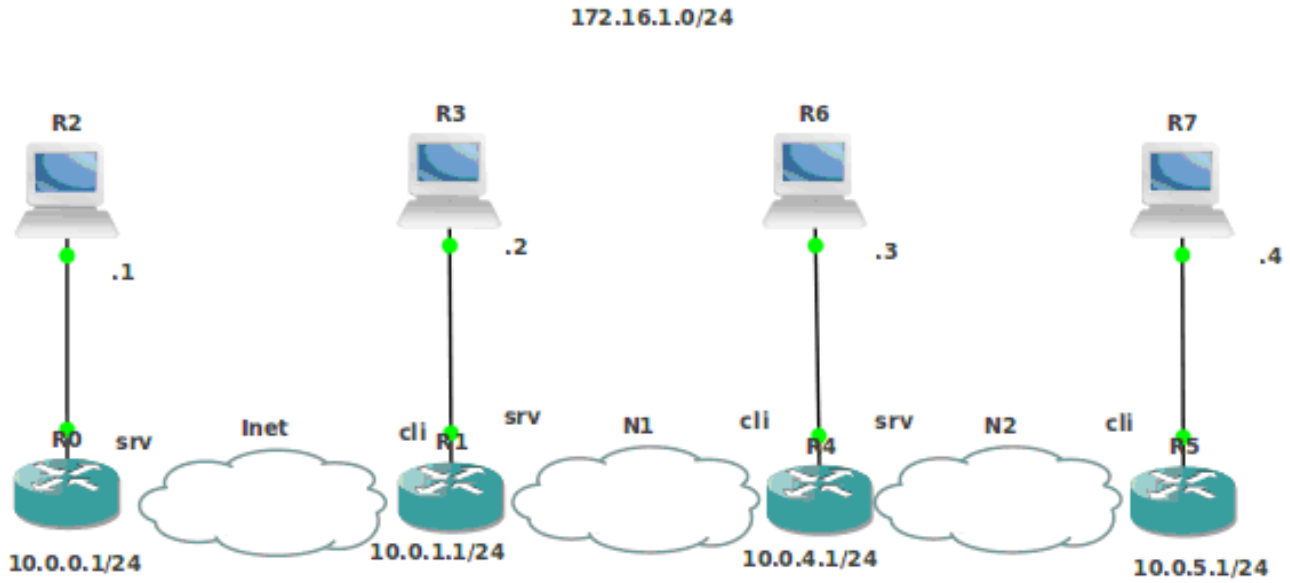

Рис. 3.1. Топология l2vpn1

interface bridge add
interface bridge port add bridge=bridge1 interface=ether1

Маршрутизаторы R0, R1 и R4 будут PPTP-серверами, а маршрутизаторы R1, R4 и R5 – PPTP-клиентами. То есть R1 и R4 являются одновременно и серверами и клиентами.

Определим в серверах пользователя q с паролем q.

ppp secret add name=q password=q

По умолчанию сервера и клиенты имеют профиль default encription. Пользователь по умолчанию имеет профиль default. Профиль пользователя подавляет профиль сервера. В каком профиле определить мост? И на серверах и на клиентах определим мост в профиле default.

ppp profile set 0 bridge=bridge1

Здесь 0 - номер профиля default, который можно увидеть из команды **ppp profile print**

В самих клиентах заменим профиль default encription на профиль default, зададим пользователя q с паролем, зададим адреса серверов и активируем их

[admin@R1]**>interface pptp-client add profile=default user=q password=q connect-to=10.0.0.1 disabled=no**

[admin@R4]**>interface pptp-client add profile=default user=q password=q connect-to=10.0.1.1 disabled=no**

[admin@R5]**>interface pptp-client add profile=default user=q password=q connect-to=10.0.4.1 disabled=no**

Активируем сервера.

interface pptp-server server set enabled=yes

В консолях маршрутизаторов R0, R1, R4, R5 введём команду **interface bridge port print.** Видим, что в мостах появятся новые динамические PPTP интерфейсы. Причем в маршрутизаторах R1, R4 их будет два: для клиента и сервера.

Получился распределённый мост (или свич): все компьютеры R2, R3, R6, R7 лежат в одной Ethernet-сети.

R2 и R7 видят друг друга по протоколам Ethernet и IP. Проверьте это.

Повторите всё для протоколов L2TP, SSTP и OpenVPN.

3.2. Использование профилей пользователя.

Соберём топологию l2vpn2, изображённую на рис. 3.2. В ней адреса компьютеров R2 и R3 лежат в сети 172.16.1.0/24. Адреса компьютеров R6 и R7 лежат в той же сети 172.16.1.0/24. Назначим адреса и имена согласно рисунку.

Маршрутизатор R0 будет PPTP-сервером, а маршрутизаторы R1, R4- PPTP клиентами. Во всех клиентах R1 и R4 добавьте мосты и в них Ethernet-интерфейс, идущий к подсоединённому компьютеру. На клиентах в профиле default определим мост

ppp profile set 0 bridge=bridge1

Здесь 0 - номер профиля default, который можно увидеть из команды **ppp profile print**

На сервере добавим два моста и в каждый из них добавим по одному Ethernet интерфейсу, идущему к разным компьютерам. Пусть ether1 идёт к R2, а ether2 идёт к R7.

[admin@R0]>**interface bridge add**
[admin@R0]>**interface bridge add** (2 раза)
[admin@R0]>**int bridge port add bridge=bridge1 int= ether1**
[admin@R0]>**int bridge port add bridge=bridge2 int= ether2**

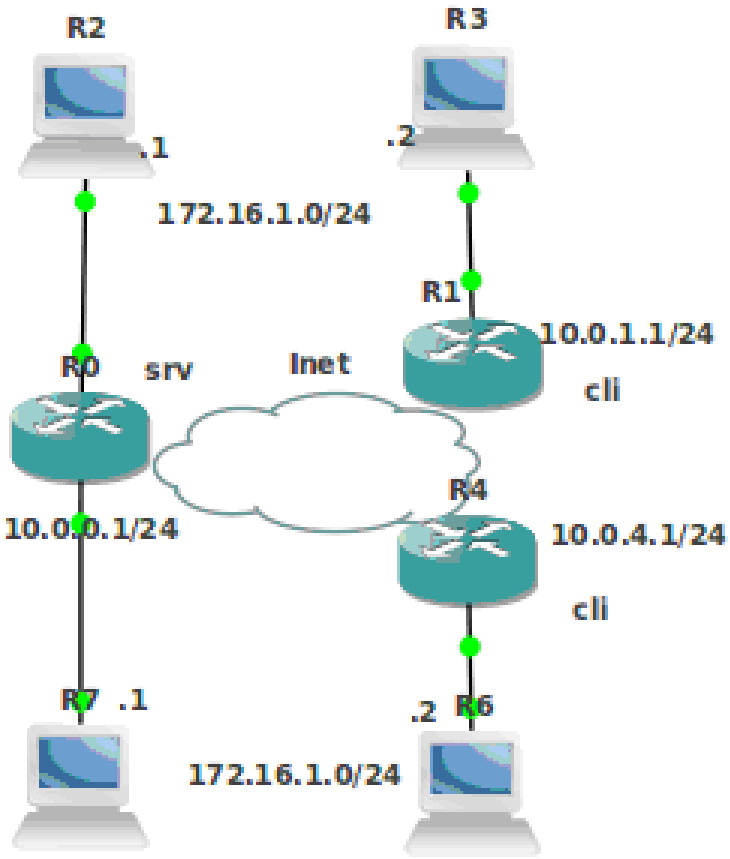

Рис. 3.2. Топология l2vpn2

Создадим для моста bridge1 профиль 1, а для моста bridge2 профиль 2

[admin@R0]**>ppp profile add name=1 bridge=bridge1**
[admin@R0]**>ppp profile add name=2 bridge=bridge2**

Создадим пользователей 1 и 2 с профилями 1 и 2, соответственно

[admin@R0]**>ppp secret add name=1 password=1 profile=1**
[admin@R0]**>ppp secret add name=2 password=1 profile=2**

Активируем сервер
interface pptp-server server set enabled=yes

В самих клиентах заменим профиль default encription на профиль default, зададим разных пользователей с паролем, зададим адрес сервера и активируем их

[admin@R1]**>interface pptp-client add profile=default user=1 password=1 connect-to=10.0.0.1 disabled=no**
[admin@R4]**>interface pptp-client add profile=default user=2 password=2 connect-to=10.0.0.1 disabled=no**

В консолях маршрутизаторов R0, R1 и R4 введём команду **interface bridge port print.** Видим, что в мостах появятся новые динамические PPTP-интерфейсы. Причем в маршрутизаторе R0 они располагаются в разных мостах.

Мы получили два независимых распределённых виртуальных моста (свича). Компьютеры R2 и R3 лежат в одной Ethernet-сети, а компьютеры R6 и R7 лежат в другой Ethernet-сети. Эти Ethernet-сети никак не связаны, и в них даже можно назначать одинаковые адреса, например из сети 172.16.1.0.24.

Проверим
[admin@R3]**>system telnet 172.16.1.1**
Попадём в R2. Выход ctrl-d.
[admin@R6]**>system telnet 172.16.1.1**
Попадём в R7. Выход ctrl-d.
Повторите всё для протоколов L2TP, SSTP и OpenVPN.

4. PPP L3 VPN

Соберём топологию, изображённую рис. 4.1. Назначьте имена и адреса согласно рисунку. Пропишем на компьютерах R2, R3 и R6 маршрут по умолчанию

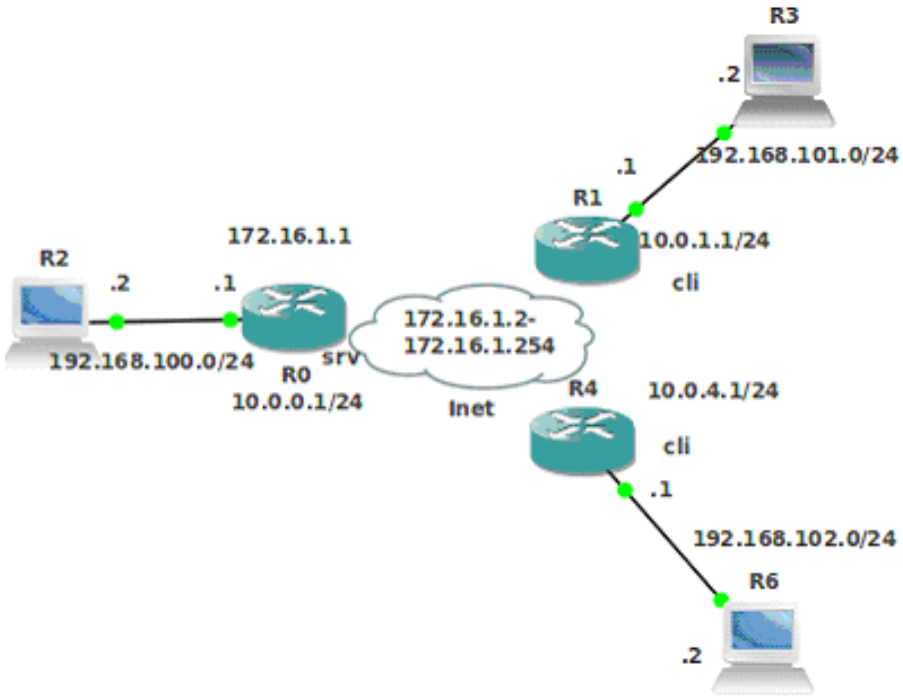

Рис. 4.1. Топология l3vpnrip

Объединим компьютеры R2, R3 и R6 в VPN третьего уровня.

R0 сделаем PPPT-сервером, а R3 и R6 - PPPT-клиентами. Поставим задачу так организовать маршрутизацию, чтобы компьютеры R2, R3 и R6 видели бы друг друга по протоколу IP.

Если мы не используем мосты, то надо определится с адресами, назначаемыми на PPPTP-интерфейсы после установки PPPT-соединения.

Добавим в PPPT-сервере R0 пул адресов, назначаемых подсоединившимся PPPTP-клиентам

[admin@R0]**>ip pool add ranges=172.16.1.2-172.16.1.254 name=pool**

Пропишем это в профиле default. Адрес 172.16.1.1 мы будем назначать на интерфейс PPPT-сервера

[admin@R0]**>ppp profile set 0 local-address=172.16.1.1 remote-address=pool**
[admin@R0]**>ppp secret add name=q password=q**

Активируем сервер

interface pptp-server server set enabled=yes

В самих клиентах R1 R4 зададим пользователя q с паролем и зададим адрес сервера

interface pptp-client add profile=default user=q password=q connect-to=10.0.0.1 disabled=no

На сервере появилось два одинаковых адреса, но в разных сетях
[admin@R0]**>ip ad pr**
2 D 172.16.1.1/32 172.16.1.253 <pptp-q-1>
3 D 172.16.1.1/32 172.16.1.254 <pptp-q>

На клиенте R1 появился адрес 172.16.1.254
[admin@ R1] **> ip ad pr**
2 D 172.16.1.254/32 172.16.1.1 pptp-out1

На клиенте R4 появился адрес 172.16.1.253
[admin@ R4] **> ip ad pr**
2 D 172.16.1.253/32 172.16.1.1 pptp-out2

Обратите внимание на маску назначенных адресов и сети. В нашей топологии фигурирует 3 сети с маской /24 192.168.100.0/24 192.168.101.0/24 192.168.102.0/24 и в общем случае переменное число сетей с маской /32. Это число зависит от количества клиентов. В нашем случае имеем 3 сети 172.16.1.1 172.16.1.253 172.16.1.254

Можно прописать маршрутизацию статически (сделайте это).

4.1. Маршрутизация RIP

Воспользуемся протоколом RIP. Так как нельзя предсказать, какие адреса будут назначены клиентам, будем оперировать сетью 172.16.1.0/24.

[admin@R0]**>routing rip network add network=172.16.1.0/24**
[admin@R0]**>routing rip network add network=192.168.100.0/24**
[admin@R1]**>routing rip network add network=172.16.1.0/24**
[admin@R1]**>routing rip network add network=192.168.101.0/24**
[admin@R4]**>routing rip network add network=172.16.1.0/24**
[admin@R4]**>routing rip network add network=192.168.102.0/24**

Посмотрим созданные RIP-маршруты

```
[admin@R4]> ip ro pr
 3 ADr  172.16.1.254/32                    172.16.1.1        120
 4 ADr  192.168.100.0/24                   172.16.1.1        120
 5 ADr  192.168.101.0/24                   172.16.1.1        120
 6 ADC192.168.102.0/24 192.168.102.1   ether1            0
```

Есть маршруты на сети 192.168.100.0/24 и192.168.101.0/24 и компьютеров R2 и R3, соответственно.

```
[admin@R1]> ip ro pr
3 ADr     172.16.1.253/32                 172.16.1.1        120
 4 ADr    192.168.100.0/24                172.16.1.1        120
 5 ADC    192.168.101.0/24  192.168.101.1  ether1            0
 6 ADr    192.168.102.0/24                172.16.1.1        120
```

Есть маршруты на сети 192.168.100.0/24 и192.168.102.0/24 и компьютеров R2 и R6, соответственно.

Аналогично на R2 есть маршруты на сети 192.168.101.0/24 и192.168.102.0/24 и компьютеров R3 и R6, соответственно.

R2, R3 и R6 увидят друг друга по IP. VPN третьего уровня создана.

Такой трюк с сетью 172.16.1.0/24 для OSPF не проходит

4.2. Маршрутизация OSPF

Теперь переделаем конфигурацию для маршрутизации путём назначения каждому клиенту определённого адреса. Этого добьемся путем назначения каждому клиенту отдельного имени со своим профилем. Сделайте копию l3vpnospf топологии l3vpnrip. На сервере R0 создадим 2 профиля

[admin@R0]>**ppp profile add name=1 local-address=172.16.1.1 remote-address=172.16.1.2**
[admin@R0]>**ppp profile add name=2 local-address=172.16.1.1 remote-address=172.16.1.3**

Создадим пользователей 1 и 2 с профилями 1 и 2, соответственно

[admin@R0]>**ppp secret add name=1 password=1 profile=1**
[admin@R0]>**ppp secret add name=2 password=2 profile=2**
Активируем сервер
[admin@R0]>**interface pptp-server server set enabled=yes**

Добавим клиентов на R1 R4

[admin@R1]**>interface pptp-client add user=1 password=1 connect-to=10.0.0.1 disabled=no**

[admin@R4]**>interface pptp-client add user=2 password=2 connect-to=10.0.0.1 disabled=no**

На сервере появилось два одинаковых адреса, но в разных сетях

[admin@ R0] > **ip ad pr**

2 D 172.16.1.1/32 172.16.1.3 <pptp-2>

3 D 172.16.1.1/32 172.16.1.2 <pptp-1>

На клиентах появились адреса

[admin@ R1] > **ip ad pr**

2 D 172.16.1.2/32 172.16.1.1 pptp-out1

[admin@ R4] > **ip ad pr**

2 D 172.16.1.3/32 172.16.1.1 pptp-out1

Воспользуемся протоколом OSPF

[admin@R0]>**routing ospf network add network=172.16.1.2 area=backbone**
[admin@R0]>**routing ospf network add network=172.16.1.3 area=backbone**
[admin@R0]>**routing ospf network add network=192.168.100.0/24 area=backbone**
[admin@R1]>**routing ospf network add network=172.16.1.1 area=backbone**
[admin@R1]>**routing ospf network add network=192.168.101.0/24 area=backbone**
[admin@ R4]>**routing ospf network add network=172.16.1.1 area=backbone**
[admin@R4]>**routing ospf network add network=192.168.102.0/24 area=backbone**

Обратите внимание, что в настройках сетей для OSPF (как и в RIP) экспортируется сеть, а не адрес. Посмотрим созданные OSPF маршруты

[admin@ R4] > **ip ro pr**

3	ADo	172.16.1.2/32	172.16.1.1	110
4	ADo	172.16.1.3/32	172.16.1.1	110
5	ADo	192.168.100.0/24	172.16.1.1	110
6	ADo	192.168.101.0/24	172.16.1.1	110

Есть маршруты на сети 192.168.100.0/24 и192.168.101.0/24 и компьютеров R2 и R3, соответственно

```
[admin@ R1] > ip ro pr
3 ADo    172.16.1.2/32          172.16.1.1    110
 4 ADo   172.16.1.3/32          172.16.1.1    110
 5 ADo   192.168.100.0/24       172.16.1.1    110
7 ADo    192.168.102.0/24       172.16.1.1    110
```

Есть маршруты на сети 192.168.100.0/24 и192.168.102.0/24 и компьютеров R2 и R6, соответственно.

Аналогично на R2 есть маршруты на сети 192.168.101.0/24 и192.168.102.0/24 и компьютеров R3 и R6, соответственно.

R2, R3 и R6 увидят друг друга по IP. R6 и наоборот. VPN третьего уровня создана

4.3. VPN уровня 3 через NAT

Соберём топологию, изображённую рис. 4.2. Проверьте соседей. Здесь R3 и R4 – маршрутизаторы Интернет-провайдера. Назначьте для R2 и R3 адреса из сети 192.168.1.0/24 согласно рисунку. Для R2 назначьте шлюз 192.168.1.1. Назначьте для R4 и R6 адреса из сети 192.168.2.0/24. Для R2 назначьте шлюз 192.168.2.1. Обеспечьте маршрутизацию между тап-интерфейсами.

Назначьте на тап-интерфейс R4 второй адрес

[admin@R4]>**ip ad ad address=10.0.4.22/24 interface=ether7**

Настройте NAT для исходящих

[admin@R3]>**ip firewall nat add chain=srcnat action=masquerade out-interface=ether7**

и приходящих адресов

[admin@R4]>**ip firewall nat add chain=dstnat action=dst-nat to-addresses=192.168.2.2 dst-address=10.0.4.22**

Набрав на R2

[admin@R2] > **sys telnet 10.0.4.22**

должны попасть в 192.168.2.2 (R6). Выход CtrlD

Настроим PPTP, полагая, что профиль default имеет номер 0

[admin@R6] > **ppp profile set 0 local-address=172.16.1.1 remote-address=172.16.1.2**

[admin@R6] > **ppp secret add name="q" service=pptp password="q" profile=default**

Запускаем PPTP сервер на R6

[admin@R6] >**interface pptp-server server set enabled=yes**

Настроим PPTP-клиент на R2. Соединяемся к PPTP-серверу R6 через NAT т.е. через адрес 10.0.4.22, а не через адрес 192.168.2.2 PPTP-сервера R6 .

[admin@R2] > **interface pptp-client add connect-to=10.0.4.22 user="q" password="q"** disabled=no

Проверим доступность R6 из R2 по новому адресу

[admin@R2] **ping 172.16.1.1**

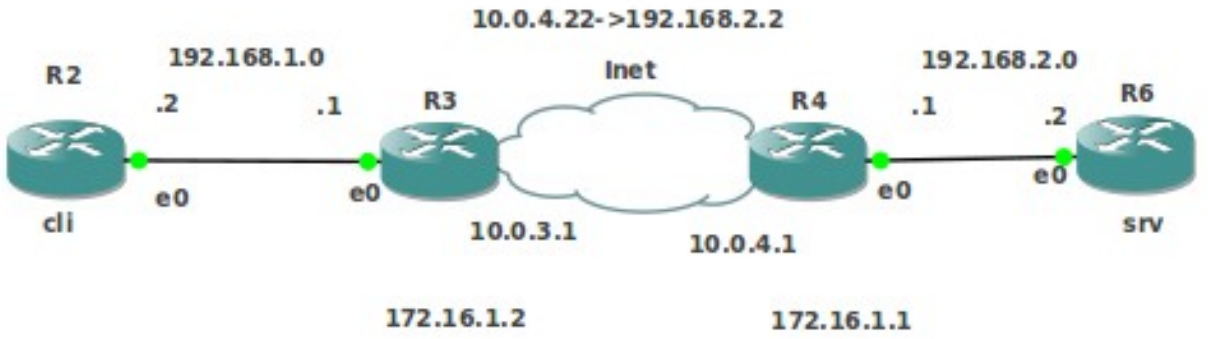

Рис. 4.2 VPN 3 уровня через NAT