Лабораторная работа 3.

VPN GRE IPSec on Cisco

Схема для разъяснения работы GRE туннелирования показана на рисунке 1:

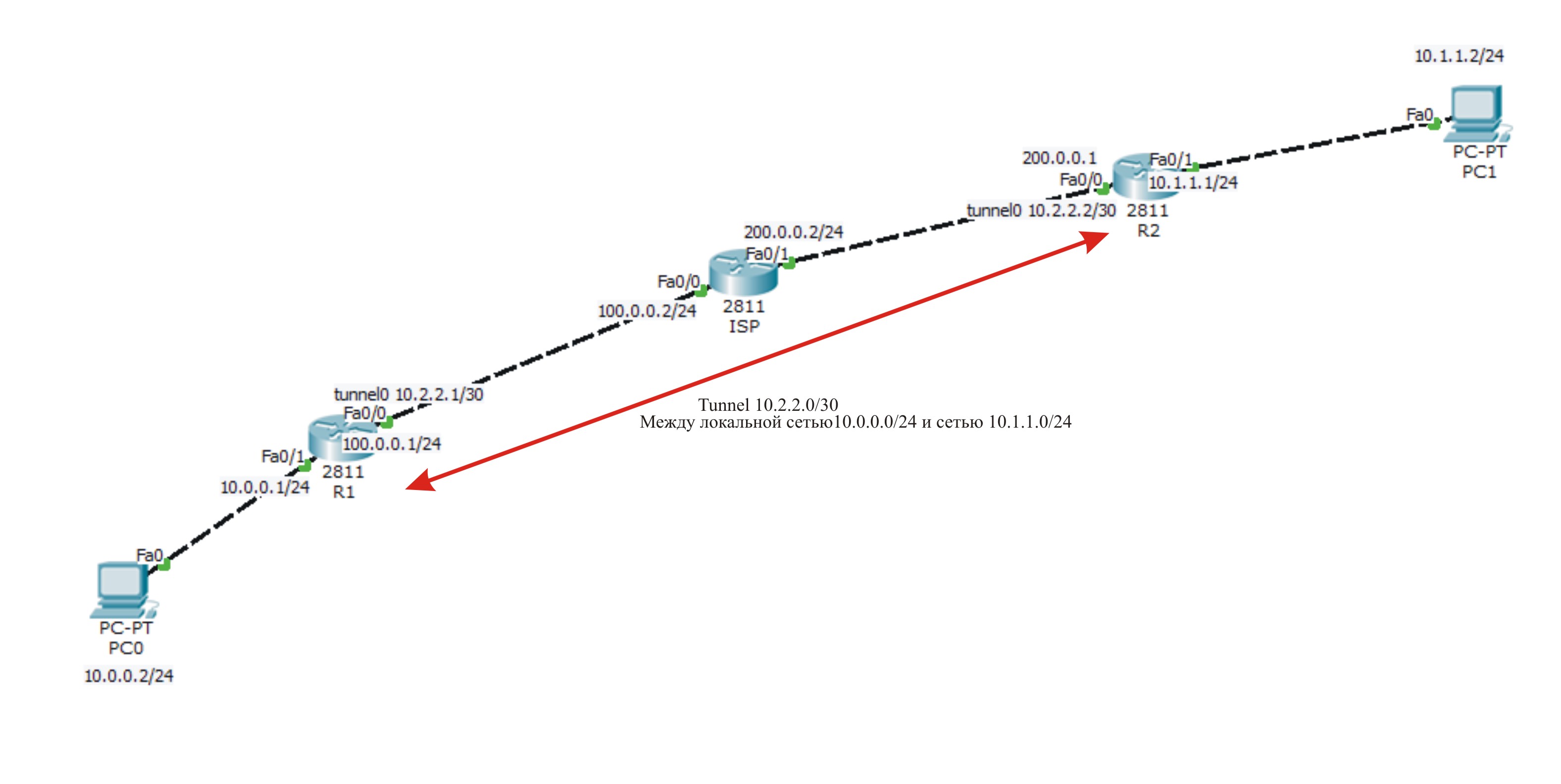


Рисунок 1.

В данной схеме две локальных сети 10.0.0.0/24 и 10.1.1.0/24 подключенные к маршрутизаторам R1 и R2 соответственно могут передавать трафик между собой как буд-то они находятся в локальной сети, несмотря на то что между ними физический канал сети провайдера, роутер ISP.

Маршрутизаторы R1 и R2 являясь граничными роутерами подключены к сети провайдера. R1 через интерфейс Fa0/0 c IP адресом 100.0.0.1/24, роутер R2 через интерфейс Fa0/0 c IP адресом 200.0.0.1/24.

На данных роутерах нам нужно создать виртуальный интерфейс tunnel0, который мы и настроим для нашего GRE канала через сеть провайдера. На самом деле виртуальный интерфейс tunnel 0 имеет возможность настраивать не только GRE каналы, но и множество других типов виртуальных каналов. Возможности PT в этом весьма ограниченны, но вот GNS дает возможность настроить куда больше типов вирт. каналов, так и дополнительных настроек к ним, рисунок 2 и 3:

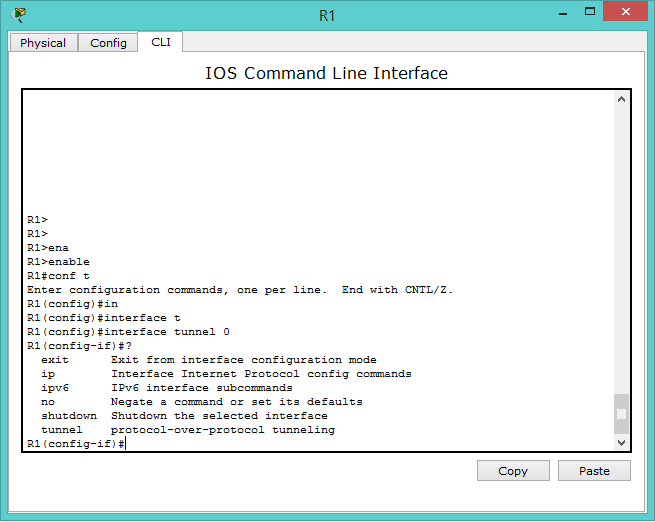


Рисунок 2.

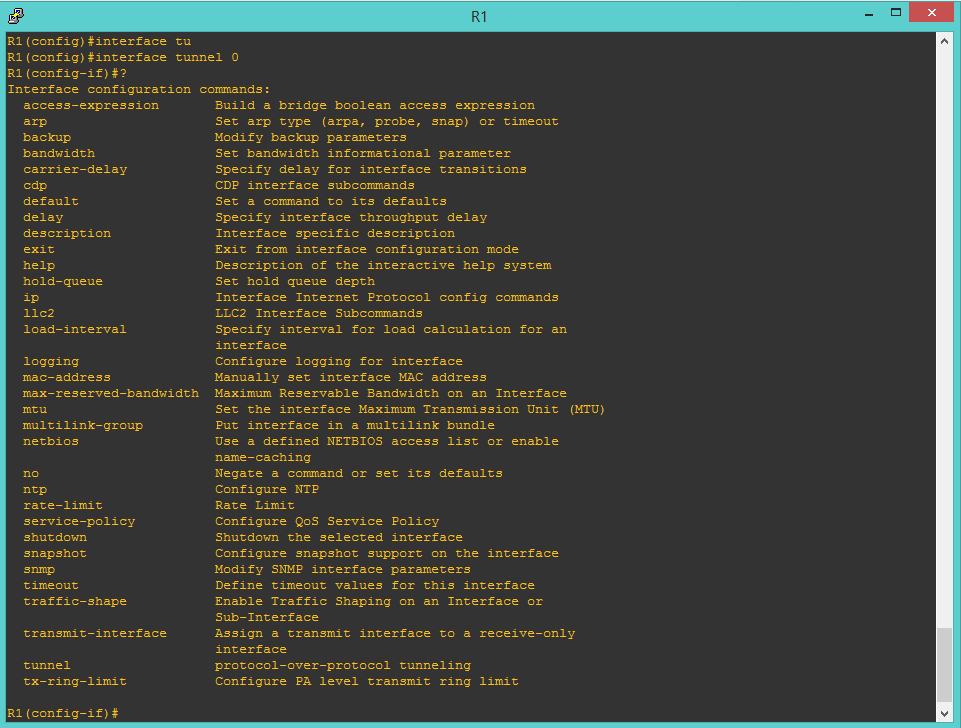


Рисунок 3.

Сети провайдера со стороны R1 - 100.0.0.0/24, со стороны R2 – 200.0.0.0/24. Так как в лабораторной работе мы эмулируем и провайдера, то необходимо настроить и интерфейсы роутера ISP, Fa0/0 100.0.0.2/24 и Fa0/1 200.0.0.2/24.

Настройка GRE туннеля

Осуществляется на маршрутизаторах R1 и R2. Путем настройки на R1 виртуального интерфейса tunnel 0 с IP адресом 10.2.2.1/30 и на R2 виртуального интерфейса tunnel 0 с IP адресом 10.2.2.2/30. То есть Интерфейс входа и выхода из туннеля находяться в одной маленькой подсети 10.2.2.0/30.

Стоит так же обратить внимание, что виртуальный интерфейс tunnel0 сразу после создания переходит в состоянии UP и добавляется в таблицу маршуртизации как Directly Connected, рисунок 4:

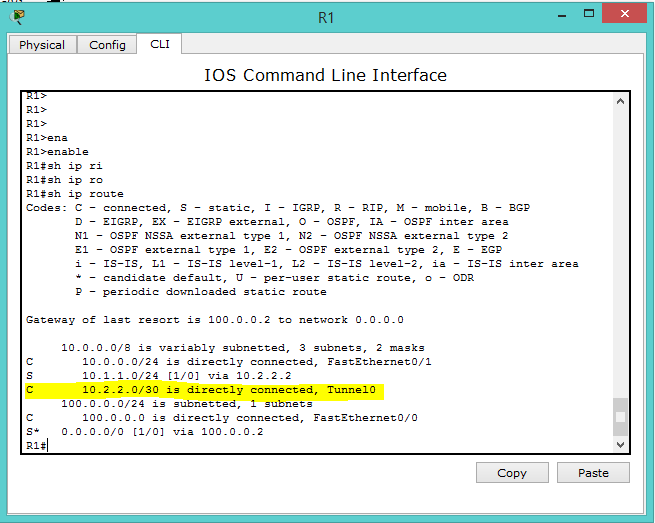


Рисунок 4.

Этим и объясняется тот факт что выйдя из туннеля трафик достигает локальной сети назначения, так как она тоже является C подключением на роутере.

Помимо задание IP адреса для виртуального интерфейса tunnel0 мы должны задать ему точку входа и выходы из туннеля, проще говоря, указать какой интерфейс является входом на нашем локальном маршрутизаторе, в примере ниже на R1, и какой является интерфейсом выхода, он уже указывается в виде IP адреса роутера R2. Думаю понятно, что в нашем случае это будут интерфейсы Fa0/0, которыми R1 и R2 подключаются к ISP для организации физического канала между ними.

Настройка GRE канала со стороны R1, рисунок 5:

**interface Tunnel0**

**ip address 10.2.2.1 255.255.255.252**

**tunnel source FastEthernet0/0**

**tunnel destination 200.0.0.1**

**tunnel mode gre ip**

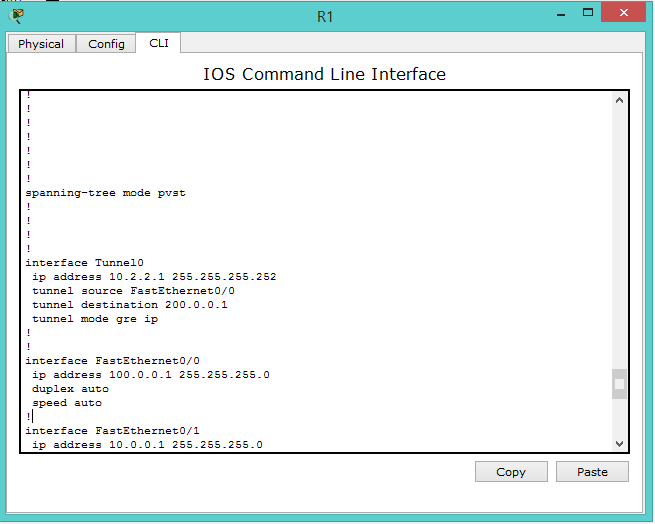
****

Рисунок 5.

Со стороны маршрутизатора R2 (канал в обратную сторону), делаем аналогичные настройки, рисунок 6:

**interface Tunnel0**

**ip address 10.2.2.2 255.255.255.252**

**tunnel source FastEthernet0/0**

**tunnel destination 100.0.0.1**

**tunnel mode gre ip**

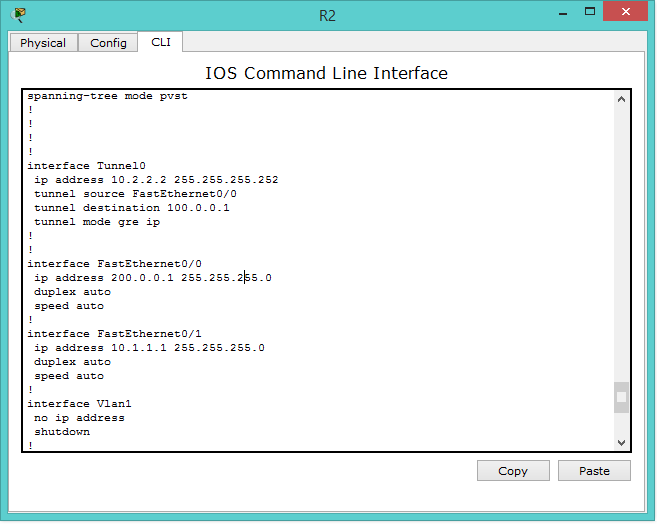


Рисунок 6.

Настройка маршрутизации туннельного трафика до сети провайдера

На данном этапе нужно настроить маршрутизацию трафика в сеть провайдера.

На стороне роутера R1:

**ip route 0.0.0.0 0.0.0.0 100.0.0.2**

**ip route 10.1.1.0 255.255.255.0 10.2.2.2**

Первый маршрут, маршрут по умолчанию, IP следующего перелета – IP адрес маршрутизатора провайдера (ISP). В нашем случае 100.0.0.2 (Fa0/0 ISP).

Второй маршрут определяет, чтобы достигнуть сети 10.1.1.0, к которой подключен PC1, нужно перебросить трафик на интерфейс следующего перелета tunnel0 (интерфейс выхода из нашего GRE туннеля, если смотреть со стороны R1) -10.2.2.2. Трафик выйдя на tunnel 0 на R2 окажется на роутере R2, на котором интерфейс Fa0/1 10.1.1.1 Directly Connected сети 10.1.1.0/24, так что маршрутизатор далее сможет правильно отправить трафик полученный на tunnel0 на Fa0/1 к сети получателя.

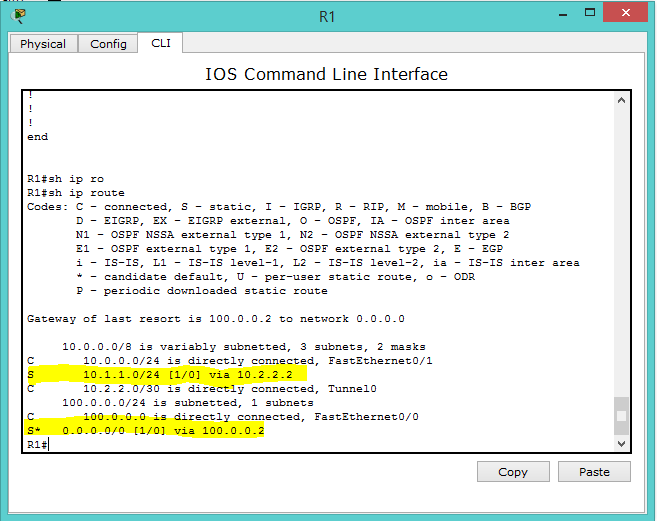


Рисунок 7.

Маршруты со стороны R2, рисунок 8:

**ip route 0.0.0.0 0.0.0.0 200.0.0.2**

**ip route 10.0.0.0 255.255.255.0 10.2.2.1**

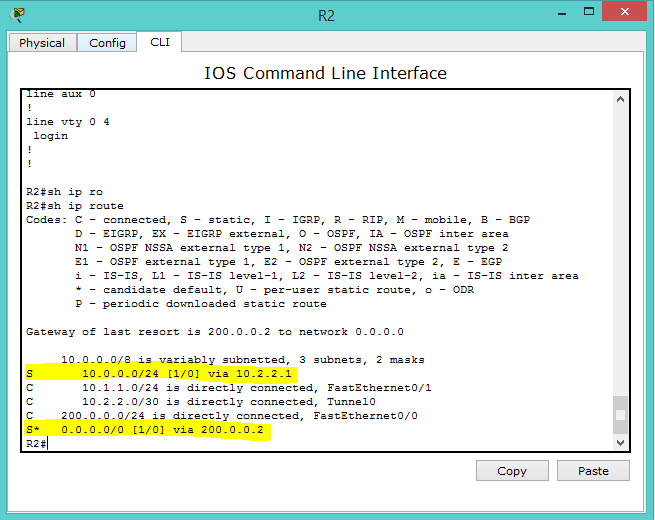


Рисунок 8.

Проверка работоспособности GRE канала

Разумеется самый простой способ проверить работу GRE канала это выполнить с PC0 ping до PC1. Но точно убедится, что фреймы идущие с PC0 на роутере R1 упаковываются в GRE заголовки и действительно перемещаются по сети провайдера в GRE туннеле можно с помощью анализатора трафика встроенного в PT, ну или Wireshark при сборке схемы на GNS.

На рисунках 9-12, показано содержание заголовков IP и канального уровней при передаче пакета ping через роутер R1. Source IP это IP компьютера PC0, Destintation IP, это IP интерфейса Fa0/1 к которому подключен PC0. Данные адреса выделены синим цветом.



Рисунок 9.

По таблице маршрутизации (рис. 7) роутер R1 определяет, что сеть 10.1.1.0/24 доступна через интерфейс следующего перелета 10.2.2.2, сеть которого имеет интерфейс выхода tunnel0!

В свою же очередь у виртуального интерфейса tunnel0 уже заданы точки входа и выхода как IP адреса физических интерфейсов Fa0/0 на R1 и R2.

**R1#sh int tun 0**

**Tunnel0 is up, line protocol is up**

**Hardware is Tunnel**

**Internet address is 10.2.2.1/30**

**MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,**

**reliability 255/255, txload 1/255, rxload 1/255**

**Encapsulation TUNNEL, loopback not set**

**Keepalive not set**

**Tunnel source 100.0.0.1, destination 200.0.0.1**

Рядом с содержимым пакета прибывшим на R1 мы наблюдаем содержимое заголовков пакета, который должен с него отправиться, а именно Source адрес физического интерфейса Fa0/0 на R1 100.0.0.1/24 и Destintation адрес Fa0/0 на R2 200.0.0.1/24. То есть адреса входа и выхода из туннеля. Помечены красным.

То есть к нашему заголовку IP пакета отправленного с PC0 на PC1 добавился заголовок с новыми, физическими точками входа и выхода из виртуального туннеля. Если бы просто была настроена маршрутизация, то разумеется IP адрес исходный и назначения не изменился бы по всему пути следования пакета.



Рисунок 10.



Рисунок 11.

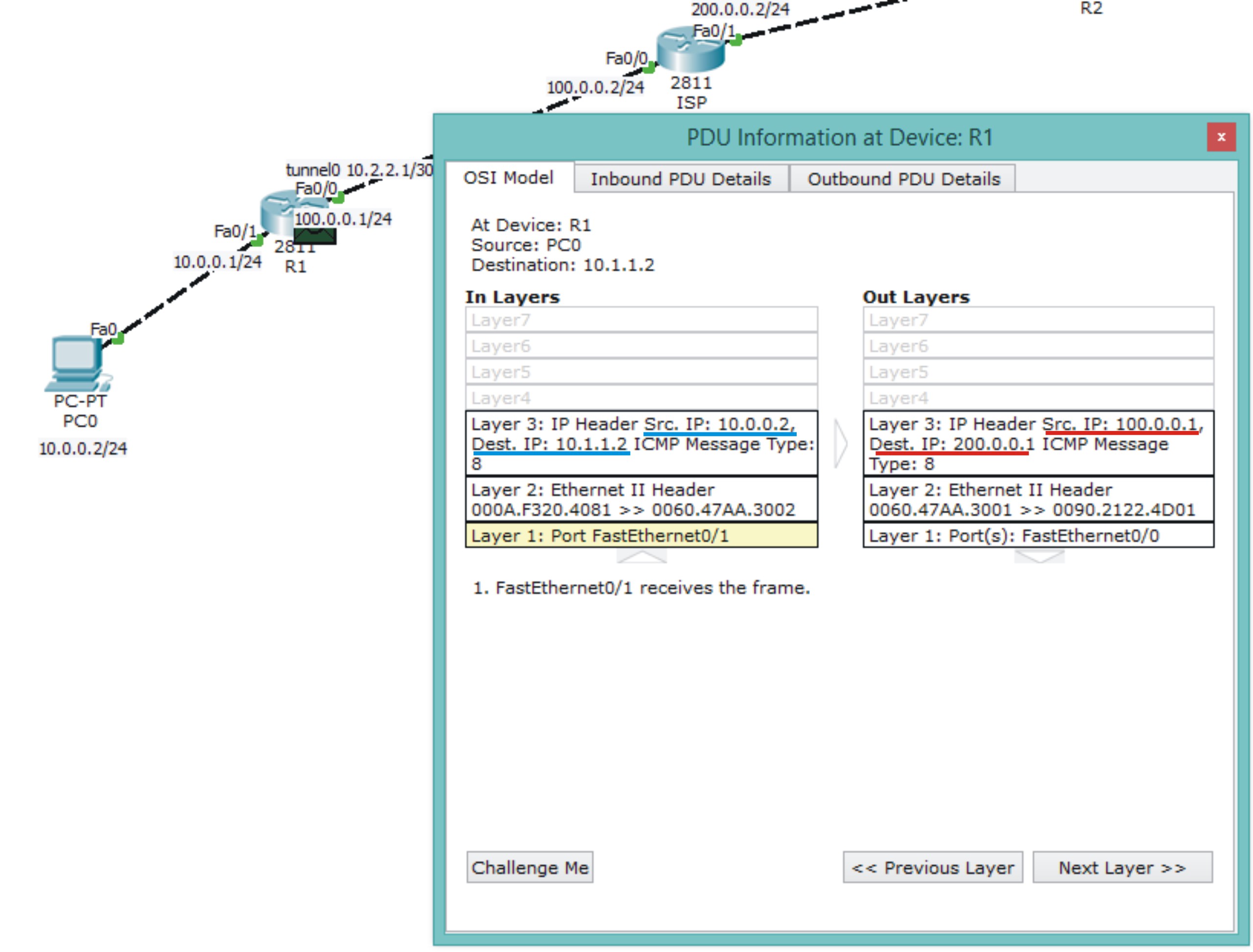


Рисунок 12.

Разумеется когда пакет выйдет из туннуля, то заголовок с Source IP 100.0.0.1 и Destintation IP 200.0.0.1 будет убран и пакет попадет в локальную сеть компьютера PC1 с теми же IP Source и Destintation с которыми он вылетал с PC0.

Если попытаться провести аналогии с осязаемым миром, то представим ситуацию, когда вы едете из деревни, инкапсулированные в автомобиль. Доезжаете до реки, и вам надо перебраться на другой берег и там продолжить своё путешествие в город.

На речном порту ваш автомобиль инкапсулируют в паром и переправляют через бушующие волны на другую сторону, где ваш автомобиль извлекают, и вы продолжаете движение. Так вот этот паром и был GRE-паромом

Минусы использования GRE туннелировния в чистом виде:

- Безопасность. Данные, инкапсулированные в GRE, передаются тем не менее в открытом виде.

- Сложность масштабирования. Если у вас 5-7 филиалов, обслуживание такого количества туннелей ещё кажется возможным, а если их 50? Причём туннелирование трафика зачастую производится на CPU, особенно на младшей и средней линейках, поэтому это лишняя нагрузка на процессор.

- Все филиалы будут взаимодействовать друг с другом через центральный узел, хотя могли бы напрямую.

Практическая часть.

Часть 1.

Настройка GRE канала между основным офисом и филиалом.

Задание 1.

Построение схемы и настройка адресации

В данной части работы мы организовываем GRE туннель для объединения сети основного офиса и филиала. Роутеры основного офиса именуются как mainX. Маршрутизатор филиала имеет название Branch.

Между ними находится эмуляция сети интернет провайдера с роутером ISP. Специально адресация на интерфейсах Fa0/0 роутеров main1 и Branch, а также обоих интерфейсов ISP и интерфейсов нашего туннеля GRE настроена с использованием адресов маршрутизируемых в сетях интернет.

Между маршрутизаторами mainx настроена маршрутизация RIPv2. А вот между main1---Branch статическая. В принципе можно было использовать любой динамический протокол типа RIP/EIGRP/OSPF, но они являются Internal протоколами маршрутизации, а здесь эмулируется сеть провайдера, то есть интернет. В ней надо использовать динамический протокол внешней маршрутизации BGPv4, но его настройку еще не проходили, поэтому ограничимся статикой.

В качестве ВМ Linux в этой работе будут использоваться ВМ на основе Linux Cisco Eagle Server, пароль от пользователя root – cisco. А так же OpenSUSE 13.2 i586 образ в формате OVA, пользователь test, пароль password-1. Переключение в режим root для ВМ на основе Eagle-Server:

**su -**

Так же стоит отметить, что Eagle-Server это LiveCD, вследствие чего, при выключении схемы пропадут все настройки ВМ клиента и сервера на основе Eagle-Server. При следующем перезапуске схемы задавать сетевые параметры на них нужно будет заново.

**Шаг 1.** Собрать схему сети согласно рисунок 13.

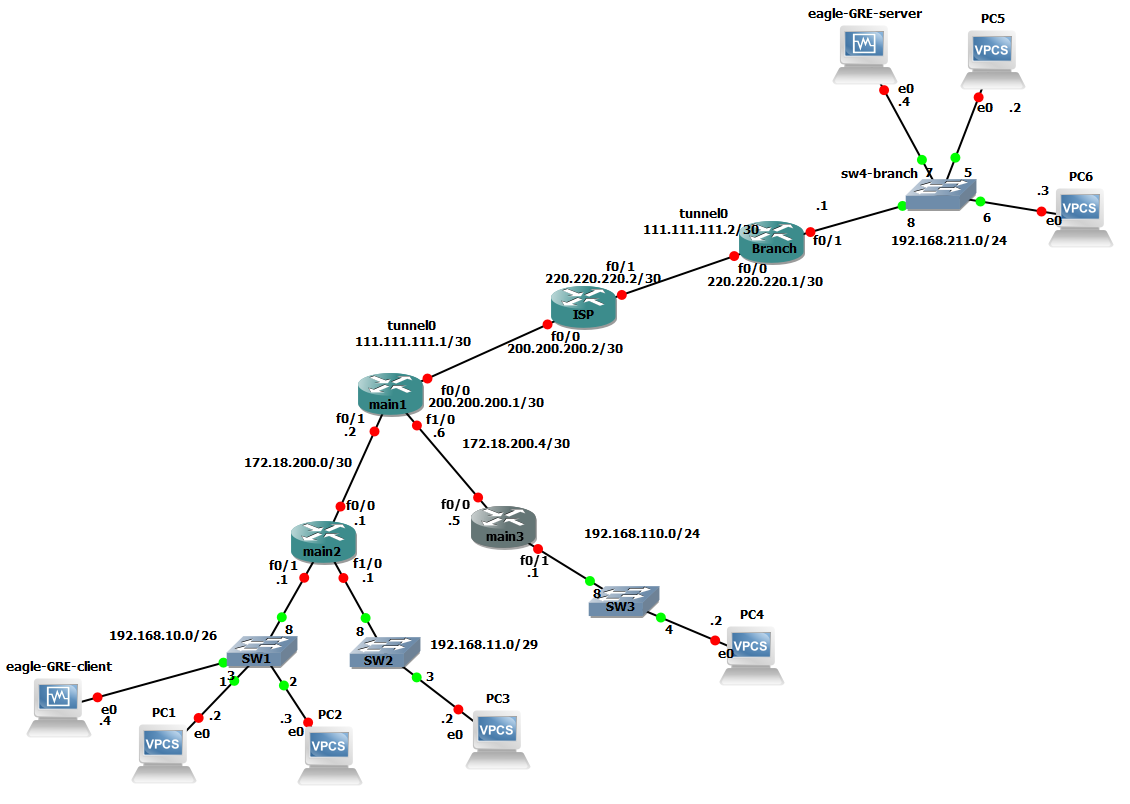


Рисунок 13

**Шаг 2.** Произвести настройку сетевых интерфейсов согласно адресному плану, указанному на схеме сети.

**Шаг 3.** Между маршрутизаторами mainx настроить маршрутизацию RIPv2, не забыв отключить автосуммаризацию маршрутов. Интерфейс Fa 0/0 на маршрутизаторе main1 не включать в процесс маршрутизации, так это внешнее подключение к провайдеру.

**Шаг 4.** Создать две ВМ типа Linux 32-bit с названиями eagle-GRE-client и eagle-GRE-server. HDD для их работы создавать не надо. В качестве носителя подключить ISO образ Linux Cisco Eagle Server, можно один на двоих. Оперативной памяти выделить 256 Мб, тип сетевого интерфейса “Не подключено”.

**Шаг 5.** Настроить на ВМ IP адреса на интерфейсе eth0 согласно адресному плану. Для чего использовать команды:

**ifconfig eth0 192.168.10.4 netmask 255.255.255.192**

**route add default gw 192.168.10.1 eth0**

Данные команды настроят IP адрес и шлюз на ВМ eagle-GRE-client.

По аналогии настроить и ВМ eagle-GRE-server.

Задание 2.

**Настройка виртуального туннеля с инкапсуляцией GRE**

**Шаг 1.** На маршрутизаторе main1 создать виртуальный интерфейс Tunnel0 и произвести его настройку, а именно:

* настроить IP адрес
* указать точку входа в туннель в виде имени интерфейса на роутере main1
* указать точку выхода из туннеля в виде IP адреса физического интерфейса роутера Branch, на котором туннель имеем выход (настроен Tunnel0)
* указать тип инкапсуляции на интерфейсе как gre ip

Пример настройки интерфейса Tunnel0 на роутере main1:

**interface Tunnel0**

**ip address 111.111.111.1 255.255.255.252**

**tunnel source FastEthernet0/0**

**tunnel destination 220.220.220.1**

**tunnel mode gre ip**

**Шаг 2.** По аналогии настроить Tunnel0 интерфейс на роутере Branch.

**Шаг 3.** Настроить статические маршруты на роутерах main1 и Branch для хождения между ними GRE трафика. Для роутера main1 нужно настроить следующие маршруты:

**ip route 0.0.0.0 0.0.0.0 200.200.200.2**

**ip route 192.168.211.0 255.255.255.0 111.111.111.2**

Первый маршрут по умолчанию направляет весь трафик с роутера до неизвестных ему сетей, а именно на интерфейс Fa 0/0 маршрутизатора провайдера ISP, который в свою очередь знает, как достигнуть интерфейса 220.220.220.1/30 который находится на Directly-Connected линке ISP. К данному интерфейсу же привязан выход из GRE туннеля.

Второй маршрут направляет трафик, предназначенный для сети 192.168.211.0/24 на интерфейс выхода из GRE туннеля на стороне роутера Branch.

Для роутера Branch маршруты аналогичны, с той лишь оговоркой, что необходимо обеспечить связь не с одной сетью как было у филиала (192.168.211.0/24), а со всеми сетями основного офиса, то есть (172.18.200.0/30, 172.18.200.4/30, 192.168.10.0/26, 192.168.11.0/29, 192.168.110.0/24):

**ip route 0.0.0.0 0.0.0.0 220.220.220.2**

**ip route 172.18.200.0 255.255.255.252 111.111.111.1**

**ip route 172.18.200.4 255.255.255.252 111.111.111.1**

**ip route 192.168.10.0 255.255.255.192 111.111.111.1**

**ip route 192.168.11.0 255.255.255.248 111.111.111.1**

**ip route 192.168.110.0 255.255.255.0 111.111.111.1**

**Шаг 4.** Чтобы хосты из сетей 172.18.200.0/30, 172.18.200.4/30, 192.168.10.0/26, 192.168.11.0/29, 192.168.110.0/24, могли обращаться к хостам филиала, нужно не забыть настроить распространение дефолтового маршрута с роутера main1 на main2 и main3 в рамках процесса маршрутизации RIPv2:

**default-information originate**

**Шаг 5.** Проверить связь между PC1, PC3, PC4 <---> PC5,PC6. Связь должна быть успешной.

Задание 3.

**Проверка прохождения трафика через GRE туннель с помощью утилиты Wireshark.**

Для тестирования прохождения трафика по GRE туннелю между главным офисом и филиалом, будут использоваться две виртуальные машины eagle-GRE-client и eagle-GRE-server. Настройка данных ВМ описана на шаге 4 и 5 задания 1, первой части работы.

**Шаг 1.** Убедиться в хождении трафика между ВМ eagle-GRE-client и eagle-GRE-server.

**Шаг 2.** Выбрать мышью линк между роутером main1 и ISP и выбрать для захвата и анализа трафика порт Fa0/0 main1 или Fa0/0 ISP.

**Шаг 3.** С помощью утилиты ping проверить связь между ВМ eagle-GRE-client и eagle-GRE-server, в Wireshark будет отображаться весь трафик ICPM инкапсулированный в GRE туннель между main1 и Branch поверх сети интернет провайдера, эмуляцию которого у нас выполняет ISP, рисунок 16:

Чтобы не расходовать RAM после выполнения утилиты ping и съемки нужных данных в Wireshark, желательно выключить захват трафика.

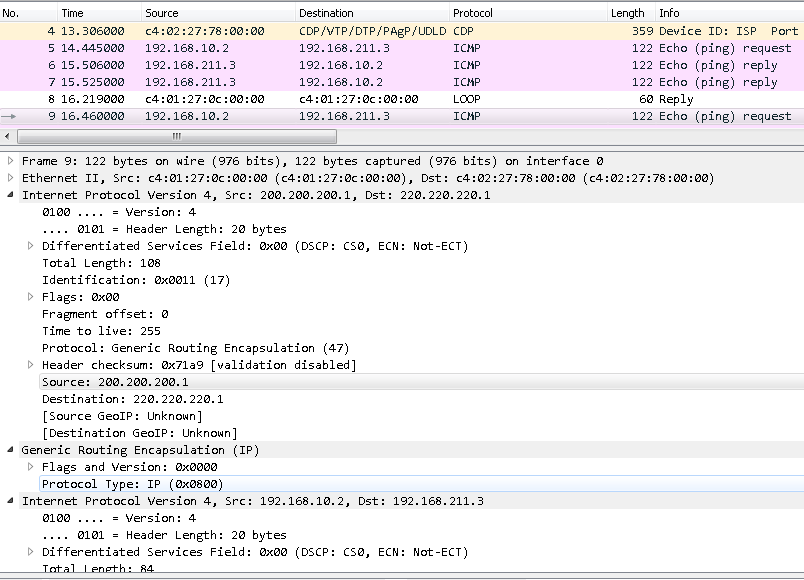


Рисунок 15.

*Примечание: скриншот отображает движение ICPM трафика по GRE туннелю между PC1 и PC 6.*

Как видно из рисунка 16, IP заголовок содержащий Source адрес 192.168.10.2 из главной сети и отдаленный адрес 192.168.211.3 в сети филиала запакован дополнительным заголовком – Generic Routing Encapsulation (IP), то есть GRE. Он уже имеет адрес отправителя – точку входа в GRE туннель со стороны роутера main1 - 200.200.200.1 и отдаленный адрес – точку выхода из GRE туннеля роутера Branch – 220.220.220.1.

**Шаг 5.**

Сделать подобный скриншот, показывающий IP адреса IP заголовка и GRE заголовка для пакетов, идущих от ВМ eagle-GRE-client к eagle-GRE-server и в обратную сторону.

**Шаг 4.** Включитьзахват трафика на линке между роутером main1 и ISP и выбрать для захвата и анализа трафика порт Fa0/0 main1 или Fa0/0 ISP.

На ВМ eagle-GRE-client используя консольную программу ftp подключится к ftp серверу на ВМ eagle-GRE-server и закачать оттуда любой файл, список файлов можно посмотреть с помощью команды **dir**.

Как и говорилось в начале работы, канал GRE не шифрует трафик в отличие от PPTP VPN соединения. Поэтому мы увидим незашифрованный трафик FTP, передающий открытым текстом логин, пароль, все служебные сообщения и сам полезный трафик данных, рисунок 16:

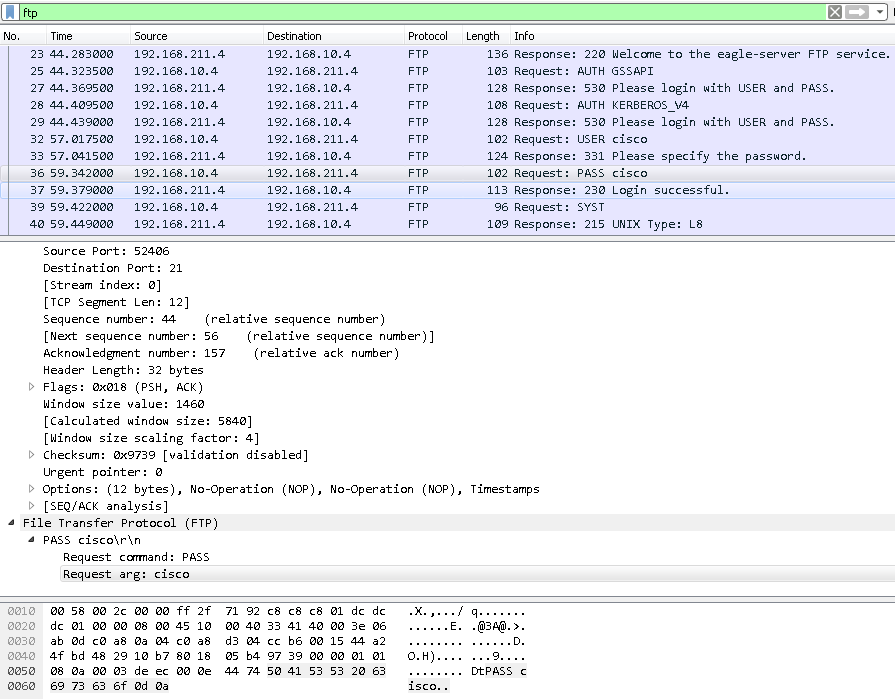


Рисунок 16.

**Шаг 5.** Сделать подобные скриншоты, показывающие IP адреса IP заголовка и GRE заголовка для пакетов, идущих от ВМ eagle-GRE-client к eagle-GRE-server и в обратную сторону, а именно для процесса аутентификации при подключении к FTP серверу клиента.

Часть 2.

Настройка FreeRADIUS для аутентификации маршрутизаторов main1 и main3.

В качестве средства AAA для роутеров main1 и main3 будем использовать открытую реализацию FreeRADIUS на основе Linux OpenSuSE.

ВМ Linux SuSE FreeRADIUS Server подключим к коммутатору SW3, согласно усложненной схемы сети, приведенной на рисунке 14.

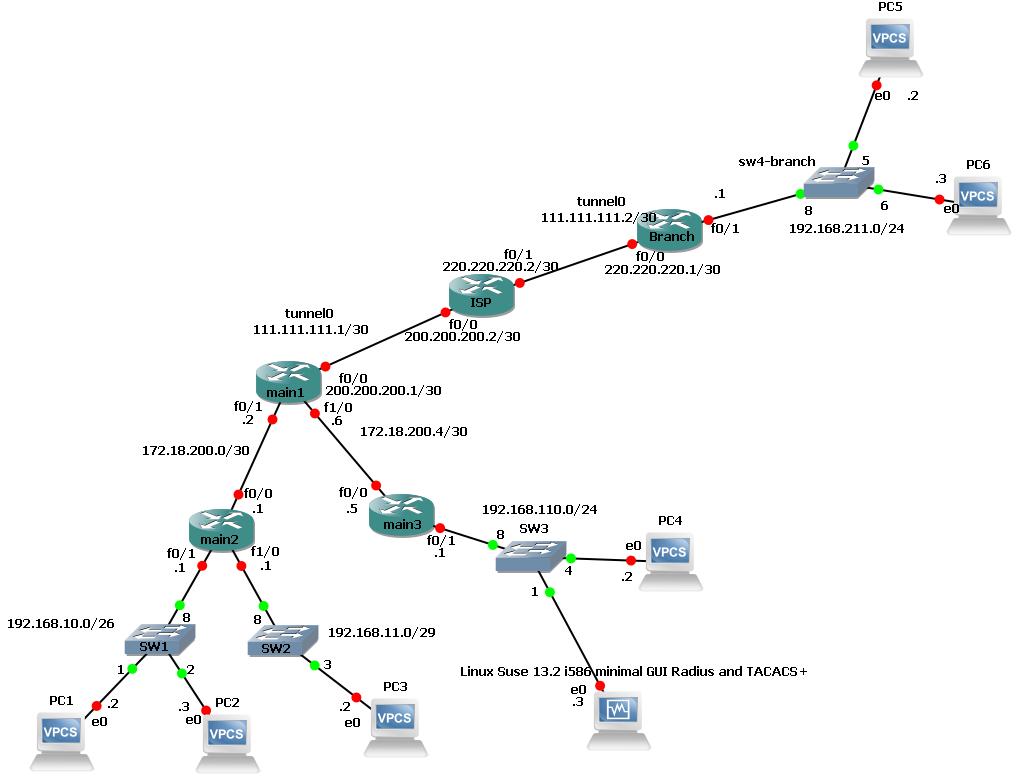


Рисунок 14.

Задание 1.

Запуск и настройка ВМ с FreeRADIUS.

**Шаг 1.** Развернуть ВМ Linux SuSE FreeRADIUS Server. Выделить память 512 Мб. Для экономии RAM рекомендуется выключить ВМ на основе Linux Cisco Eagle Server. В дальнейших настройках они уже более не нужны.

**Шаг 2.** До включения ВМ Linux SuSE FreeRADIUS Server в схему GNS ее следует запустить и установить ПО FreeRADIUS, для чего использовать графическую панель управления системы YAST или консольную утилиту zypper:

**zypper install freeradius-server**

В качестве дополнительно ПО, для удобства работы с командной строкой можно сразу установить двухпанельный файловый менеджер mc:

**zypper install mc**

После установки ПО, необходимо выключить ВМ перевести сетевой интерфейс из типа “ NAT” в тип “Не подключен”.

**Шаг 3.** Запустить схему с подключенной к коммутатору sw3 ВМ Linux SuSE FreeRADIUS Server.

**Шаг 4.**  Настроить IP адрес и шлюз:

Так как ОС Linux используемая для FreeRADIUS современная, то она определяет сетевую карту в VirtualBox не как eth0, а как enp0s3. IP адрес настраивается командой ifconfig, а маршрут по умолчанию добавлением соответствующего маршрута:

**ifconfig eth0 192.168.110.3 netmask 255.255.255.0**

**route add default gw 192.168.110.1 enp0s3**

Ну и разумеется проверить связь с роутером main1, main2 и main3.

**Шаг 5.** Для настройки FreeRADIUS нам по аналогии с RADIUS на основе RRAS в Windows Server нужно настроить клиентов AAA, по заданию это роутеры main1 и main3. А так же настроить пользователей с различными уровнями привелегий. С единственной то разницей, что при использовании Windows NPS они были бы либо системными на сервере выполняющим роль RADIUS или доменные и хранились в БД учетных записей на контроллере домена. Тут же все пользователи службы RADIUS и настройки их привилегий будут храниться в файле users (/etc/raddb/users). То есть, нет необходимости их заводить как системных пользователей Linux.

Все настройки FreeRADIUS в Linux OpenSuSE хранятся в папке /etc/raddb. Рекомендуется сделать бэкап данной папки, с дефолтовыми настройками.

В этой папке нас интересует два файла. Первый это Clients.conf , который содержит настройки клиентов RADIUS. Второй файл users, в него мы добавим пользователей, которые будут обладать различными уровнями привилегий на оборудовании CISCO.

**Шаг 6.** Открываем на редактирование файл clients.conf и дописываем в него данные секции содержащие данные о наших клиентах:

client 172.18.200.6 {

secret = password-1

nas\_type = cisco

shortname = main1

}

client 192.168.110.1 {

secret = password-1

nas\_type = cisco

shortname = main3

}

Первый параметр указывает IP адрес RADIUS клиента, второй общий секретный пароль, в нашем случае это password-1. Тип устройство cisco и его имя.

**Шаг 7.** Далее открываем на редактирование файл users и добавляем в конец, информацию о пользователях. Один пользователь будет иметь административные полномочия – admin-radius. А второй полномочиями оператора – operator-radius

admin-radius Cleartext-Password := admin-radius

Service-Type = Administrative-User,

Cisco-AVPair = "shell:roles=network-admin",

Cisco-AVPair += "shell:priv-lvl=15"

operator-radius Cleartext-Password := operator-radius

Service-Type = NAS-Prompt-User,

Cisco-AVPair = "shell:roles=network-operator",

Cisco-AVPair += "shell:priv-lvl=1"

**Шаг 8.**  Настроим автозапуск сервиса radiusd и включим его:

**chkconfig --level 345 radiusd on**

**service radius start**

Проверим что служба radius запустилась и разрешена к автозагрузке при старте ОС:

**service radius status**

**systemctl list-unit-files**

Задание 2.

Настройка роутеров main1 для работы с аутентификацией с помощью RADIUS сервера на ВМ Linux SuSE FreeRADIUS Server

Чтобы маршрутизатор cisco начал работать с помощью аутентификации, авторизации и учета посредством RADIUS или TACACS+ сервера необходимо включить новую модель аутентификации командой aaa new-model. Но перед этим шагом надо обезопасить себя от того что сервер RADIUS по какой то причине не сможет ответить на запрос аутентификации и будет использоваться локальный процесс аутентификации и авторизации на роутере. То есть для этого надо задать локального пользователя с уровнем привилегий 15.

**Шаг 1.** Шифрование всех паролей **service password-encryption**.

**Шаг 2.** Создание локального пользователя:

username root privilege 15 password password-1

В качестве пароля зададим все тот же password-1.

**Шаг 3.** Включаем новую модель аутентификации, авторизации и учета:

**aaa new-model**

**Шаг 4.** Создаем профиль аутентификации под названием **Login-Radius** и задаем его для группы radius серверов (да у нас он один, но может быть несколько в качестве обеспечения избыточности). Так же в качестве резервного метода указываем local, который будет использовать для аутентификации локальную БД пользователей, у нас это пользователь **root**.

Так же стоит отметить, что до данного пункта процесс выбора методов аутентификации доберется, только если RADIUS сервер не ответит три раза подряд. Разумеется если посылать на Radius сервер неправильные логин-пароль, то это не будет считаться неудачной попыткой аутентификации с RADIUS сервером:

**aaa authentication login Login-Radius group radius local**

Так же не забываем создать профиль авторизации на запуск команд - **Auth-Radius-exec**:

**aaa authorization exec Auth-Radius-exec group radius local**

**Шаг 5.** Теперь надо разрешить доступ по виртуальным консолям vty 0 15 для работы с AAA. Для чего зайти в режим конфигурации линий vty 0 15 и включить профили аутентификации и авторизации настроенные на предыдущем шаге:

**login authentication Login-Radius**

**authorization exec Auth-Radius-exec**

**Шаг 6.** Настроим тайм аут простоя рабочей сессии для всех виртуальных консолей, 15 минут, второе значение позволяет выставлять тайм аут в секундах:

**exec-timeout 15 0**

**Шаг 7.** Настроим тайм аут простоя сессии входа длительностью 180 секунд, если пользователь не введет логин-пароль по прошествии которых сессия отключится:

**timeout login response 180**

**Шаг 8.** Разрешим работу на VTY консолях только протоколов telnet и SSH:

**transport input telnet ssh**

**Шаг 9.** Так же надо указать в настройках роутеров, IP адрес сервера, который мы будем использовать в качестве RADIUS, в нашем случае это 192.168.110.3:

**radius-server host 192.168.110.3**

Если не указать порты которые роутер будет использовать для аутентификации и аккаунтинга, то будут использованы порты автонастройки auth-port 1645 acct-port 1646, то есть в running-config мы будем видеть следующие:

**radius-server host 192.168.110.3 auth-port 1645 acct-port 1646**

Но так как FreeRADIUS работает с портами 1812 и 1813 для аутентификации и авторизации соответственно, то зайти удаленно на роутер с помощью FreeRADIUS будет невозможно!

Периодически на роутере в командной строке будет появляться ошибка:

\*Mar 1 08:50:43.429: %RADIUS-4-RADIUS\_DEAD: RADIUS server 192.168.110.3:1645,1646 is not responding.

\*Mar 1 08:50:43.429: %RADIUS-4-RADIUS\_ALIVE: RADIUS server 192.168.110.3:1645,1646 is being marked alive.

При том, что утилита ping, будет сообщать о том, что FreeRADIUS сервер доступен:

main1#ping 192.168.110.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.110.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/29/48 ms

Поэтому стоит указать правильные порты с самого начала:

**radius-server host 192.168.110.3 auth-port 1812 acct-port 1813**

**Шаг 10.** Укажем максимальное число неправильно введенных паролей, скажем равное двум. Почему двум, ну потому что если бы мы использовали для хранения учетных записей пользователей MS AD, как и следует делать, а не хранить все учётные записи на NPS RADIUS сервере, то скорее всего мы бы столкнулись с блокировкой учетной записи на 30 минут после 3-х неудачных попыток ввода пароля. Это дефолтовая настройка политики безопасностей паролей MS AD, если конечно ее не изменить для всего домена или для групп пользователей cisco вручную.

**radius-server retransmit 2**

**Шаг 11.** Ну и самое главное - укажем общий секрет (общую парольную фразу), которую мы задавали при настройке клиентов RADIUS в задании 2:

**radius-server key password-1**

**Шаг 12.** Проверить возможность входа на роутер main1 с помощью FreeRADIUS с роутера main3 и main2, с учетными записями admin-radius и admin-operator и убедится в получении соответствующих привилегий.

Задание 3.

Настройка роутеров main3 для работы с аутентификацией с помощью RADIUS сервера на ВМ Linux SuSE FreeRADIUS Server

По аналогии с заданием 2 настроить роутер main3 для работы с RADIUS сервером FreeRADIUS.

Проверить возможность входа на роутер main3 с помощью FreeRADIUS с роутера main1 и main2, с учетными записями admin-radius и admin-operator и убедится в получении соответствующих привилегий.

Задание 3.

Настройка эксклюзивных прав доступа privilege level 2-14

Помимо всем известных уровней Cisco IOS и Cisco NX-OS (Nexus девайсы на основе RedHat Linux с консольной оберткой от Cisco) доступа администратора – уровень 15 и оператора – уровень 1. Есть возможность использовать оставшиеся уровни для гибкой настройки доступа администраторам, которым полагаются более низкие права.

Разумеется, данные уровни привилегий можно применять при авторизации пользователей с помощью FreeRADIUS сервера.

Установка привилегий имеет следующий вид:

**privilege <режим CLI> level <номер уровня> <разрешенная команда>**

Большинство режимов работы CLI, приведены на рисунке 15:

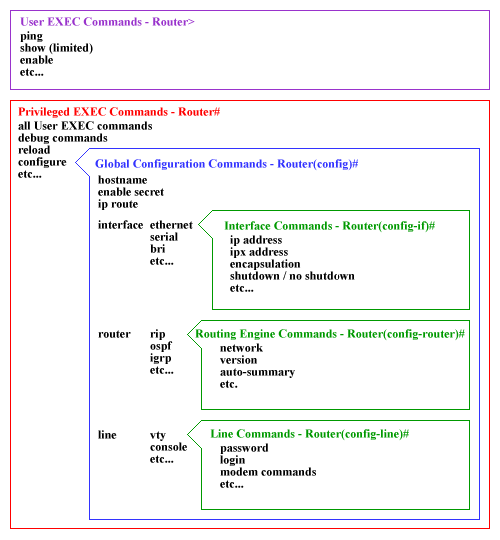


Рисунок 15.

Тут важно понимать, что разрешения для выполнения тех или иных команд надо задавать конкретно в том уровне, где эти команды работают.

Логика простая, допустим, стоит задача сделать пользователя работающего через RADIUS с уровнем привилегий 3:

1. сделать возможным пользователю с privilege level 3 возможность просматривать содержимое running-config
2. заходить в режим конфигурации
3. заходить в режим конфигурации интерфейса
4. задавать IP адреса

Данные настройки будут иметь вид:

**privilege exec level 3 show running-config**

**privilege exec level 3 configure terminal**

**privilege configure level 3 interface**

**privilege interface level 3 ip address**

В реальности, если посмотреть настройки привилегий на роутере main1, мы увидим следующие:

privilege interface level 3 ip address

privilege interface level 3 ip

privilege configure level 3 interface

privilege exec level 3 configure terminal

privilege exec level 3 configure

privilege exec level 3 show running-config

privilege exec level 3 show

То есть мы видим, что для себя CISCO IOS в конфигурации расшифровывает разрешаемые нами команды, состоящие из 2-ух слов на команду из первого слова в составной команде и далее полный вид. Самое интересное, что записывает она это наоборот:

privilege exec level 3 configure terminal

privilege exec level 3 configure

Хотя логичнее было бы сохранять конфиг в таком виде:

privilege exec level 3 configure

privilege exec level 3 configure terminal

Можно разрешить выполнять для какой либо команды нужного нам уровня исполнения в IOS всех возможных опций.

Например, выполнив нижестоящие команды мы добились того, что пользователь с уровнем привилегий 3 может зайти на роутер в режим глобальной конфигурации, далее зайти на нужный интерфейс или создать виртуальный и в нем уже иметь возможность задать ip address.

Мы разрешали на уровне конфигурации интерфейса команду ip address:

**privilege exec level 3 show running-config**

**privilege exec level 3 configure terminal**

**privilege configure level 3 interface**

**privilege interface level 3 ip address**

Пример работы команды **privilege interface level 3 ip address**, на рисунке 16:

****

Рисунок 16.

Но мы можем просто разрешить команду ip со всему возможными ее подкомандами, а не конфигурацией всех возможных команд вручную.

Для чего когда мы конфигурируем команду ip, после задания уровня исполнения IOS надо выставить опцию all, это позволит нам разрешить все подкоманды заданной команды после ключевого слова level №:

**privilege interface all level 3 ip**

На рисунке 17 видно, что для команды ip разрешены уже все опции на выполнение:

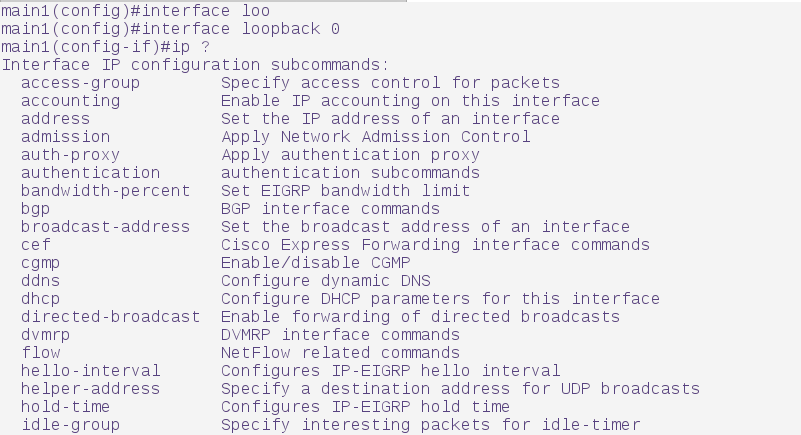


Рисунок 17.

Не стоит забывать, что перед выполнением данной команды надо явно разрешить работу уровня исполнения IOS interface, то есть порядок задания команд:

**privilege configure level 3 interface**

**privilege interface all level 3 ip**

Шаг 1. Создайте уровень привилегий 7, а так же пользователя с таким же правами на FreeRADIUS сервере с именем level7-radius и таким же паролем. Настройки свойства данной учетной записи, аналогичные пользователю operator-radius.

Шаг 2. Настройте на роутере main1 привилегии для пользователей уровня 7. А именно:

* возможность смотреть running-config и startup-config
* разрешить переход в режим глобальной конфигурации
* разрешить настройку имени хоста
* разрешить настройку descriptions на портах

- разрешить настройку IP адреса на интерфейсе

- разрешить настройку синхронного консольного вывода на виртуальных консолях vty 0 15.

Шаг 3. Зайти с ВМ Linux Suse 13.2 i586 minimal GUI Radius на роутер main1 под пользователем level7-radius, проверить возможность выполнения данных административных задач и невозможность настройки других, например конфигурации протокола динамической маршрутизации eigrp.