Network Lab Documentation

0xaryan

May 03, 2019

Contents:

CHAPTER 1

Introduction

1.1 Author

Hello everyone!

This is a simple documentation for Network Lab course of University of Guilan made by Aryan Ebrahimpour, a Computer Engineering BSc student.

> **Warning:** If GNS3 drove you crazy, please calm down, it's totally normal. GNS3 officialy has 2 purpose: 1. Network Simulation 2. Driving people crazy

1.2 Contributions

Contributions are very welcome. You can simply use that **Edit on GitHub** link on top of each page to improve these pages.

You may need Sphinx docs if you are not familiar with reStucturedText.

CHAPTER 2

Basics

These are snippets and codes we use a lot in our projects

2.1 Routers

Codes frequently used for routers

2.1.1 Config Mode

There are multiple modes in routers, including **Normal Mode** and **Config Mode**.

You can switch to config mode with `config terminal` or simply `conf t` command and get back to normal mode with `exit`:

```
R1#
R1# conf t
R1(config)#
R1(config)# exit
R1#
```

Note: Pay attention to what mode you are in.

2.1.2 Ping

You can simply ping a destination with `ping` command in **Normal Mode**. If you are in **Config** mode, use `do ping` command.

Mode	Command	Example
Normal	ping x.x.x.x	`ping 192.168.1.23`
Config	do ping x.x.x.x	`do ping 192.168.1.23`

2.1.3 Save or Show configs

If you are in **Normal** mode, simply type `show running-config` to show the current config, and `copy running-config startup-config` to save the configs for next start.

If you are in **Config** mode, put a `do` prefix before those commands:

```
R1# ping 192.168.1.1
R1# show running-config
R1# copy running-config startup-config

R1(config)# do ping 192.168.1.1
R1(config)# do show running-config
R1(config)# do copy running-config startup-config
```

2.1.4 Config interfaces

You can config interfaces of the router in *config mode*. You may have multiple interfaces on your router such as **FastEthernet** or **GigabitEthernet**, etc. Simply use `int <interface_id>` to config the interface. The *interface_id* parameter should be in any forms of the interface type, for example all of these are accepted: `FastEthernet0/0`, `fa0/0` or `f0/0`.

```
R1#conf t
R1(config)#int fa0/0
R1(config-if)#ip addr 192.198.1.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#
```

Here we first switched to *interface config mode* with `int fa0/0` command and then changed the IP address and subnet mask of the interface. The `shutdown` command disables the interface. Putting a *no* (`no shutdown` or simply `no shut`) before this command (re)enables the interface.

Notice that the `exit` command only changes the mode one level upper and does not directly switch to the *Normal mode*.

Warning: Interfaces of the same router can not be in the same network. For example you can not have two interfaces in a router with IPs `192.168.1.1/24` and `192.168.1.2/24`.

2.1.5 Show interface configs

You can see the IP and status of the interfaces with `show ip interface brief` in Normal mode. The shortened version `sh ip int br` also works.

```
R1#show ip interface brief
```

(continues on next page)

(continued from previous page)

```
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.1     YES manual up                    up
GigabitEthernet1/0         192.168.2.1     YES manual up                    up
```

2.1.6 Add item to route table

A router must know on which interface it should to forward a packet, based on the network address of it. You can manually add items to the routing table of a router using `ip route x.x.x.x y.y.y.y <interface_id>` command where the *x.x.x.x* is the network address and *y.y.y.y* is the subnet mask.

```
R1(config)#ip route 192.168.1.0 255.255.255.0 fa0/0
```

The example above simply forwards every packet with destination of `192.168.1.x` to its `FastEthernet0/0` port.

2.2 VPCs

Codes frequently used for VPCs

2.2.1 Short Codes

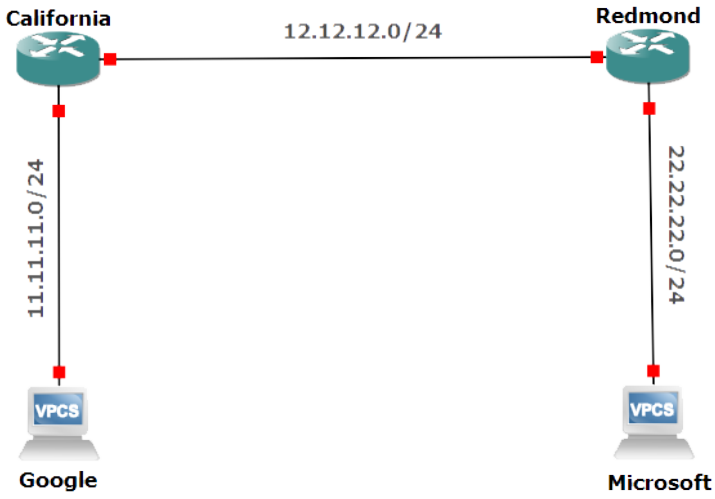
Command	Description	Example
ping x.x.x.x	Pings an IP address	`ping 192.168.1.23`
save	Saves the configs of the VPC	`save`
ip x.x.x.x/y z.z.z.z	Sets IP, Subnet mask and Gateway	`ip 192.168.1.2/24 192.168.1.1`
ip dhcp	Gets the IP from DHCP server	`ip dhcp`

CHAPTER 3

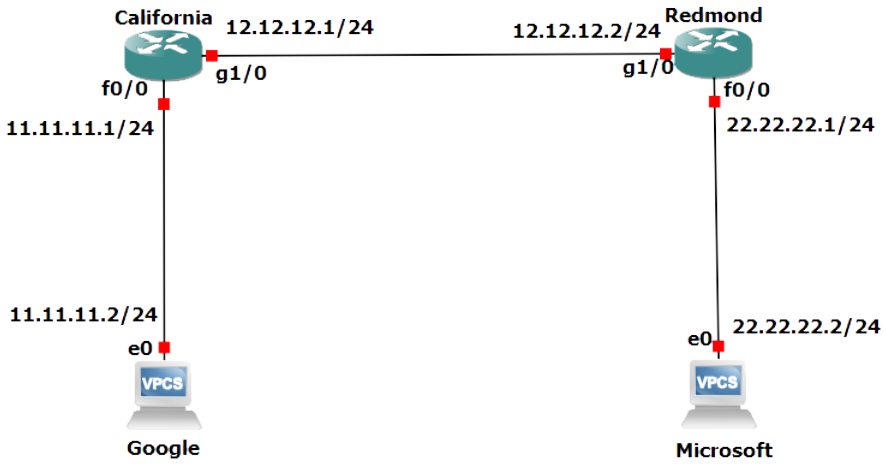
Simple IP Route

3.1 Project description

In this section, we want to ping *Microsoft* from *Google* using two routers named *Redmond* and *California*, and vice versa.



We just need to setup the interfaces with proper IP addresses, and then write the ip routes. Here is a more detailed image of the project with interface identifiers and chosen example IP addresses for each interface.



3.2 Configuration

Microsoft VPC

```
Microsoft> ip 22.22.22.2/24 22.22.22.1
```

Google VPC

```
Google> ip 11.11.11.2/24 11.11.11.1
```

Redmond Router

```
Redmond#conf t
Redmond(config)#int f0/0
Redmond(config-if)#ip addr 22.22.22.1 255.255.255.0
Redmond(config-if)#no shut
Redmond(config-if)#exit
Redmond(config)#int g1/0
Redmond(config-if)#ip addr 12.12.12.2 255.255.255.0
Redmond(config-if)#no shut
Redmond(config-if)#exit
Redmond(config)#ip route 11.11.11.0 255.255.255.0 g1/0
```

California Router

```
California#conf t
California(config)#int f0/0
California(config-if)#ip addr 11.11.11.1 255.255.255.0
California(config-if)#no shut
California(config-if)#exit
California(config)#int g1/0
California(config-if)#ip addr 12.12.12.1 255.255.255.0
California(config-if)#no shut
California(config-if)#exit
California(config)#ip route 22.22.22.0 255.255.255.0 g1/0
```

Now if you ping Google from Microsoft (or Microsoft from Google), this should be the result:

```
Microsoft> ping 11.11.11.2
84 bytes from 11.11.11.2 icmp_seq=1 ttl=62 time=69.002 ms
84 bytes from 11.11.11.2 icmp_seq=2 ttl=62 time=37.000 ms
84 bytes from 11.11.11.2 icmp_seq=3 ttl=62 time=45.998 ms
84 bytes from 11.11.11.2 icmp_seq=4 ttl=62 time=39.001 ms
84 bytes from 11.11.11.2 icmp_seq=5 ttl=62 time=31.000 ms
```

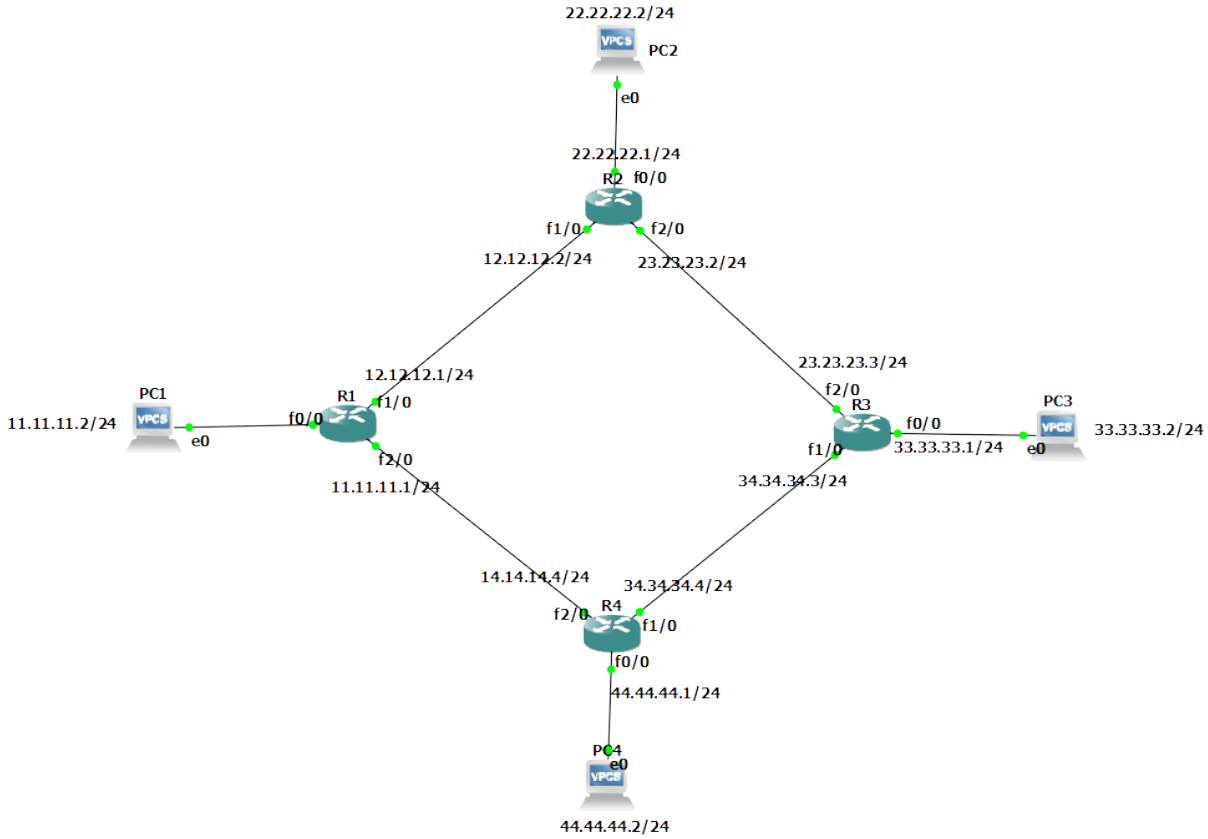
Note: Some of the first pings may timeout on your machine

CHAPTER 4

Diamond IP Route

4.1 Project description

In this section, we have four PCs and four routers in the middle. The objective is to be able to ping any PC from any other.



4.2 Configuration

Note: Becuase configuration of the interfaces and VPCs IPs are similar to the previous project, I simply just write the routing codes.

R1

```
R1(config)#ip route 22.22.22.0 255.255.255.0 f1/0
R1(config)#ip route 33.33.33.0 255.255.255.0 f1/0
R1(config)#ip route 44.44.44.0 255.255.255.0 f2/0
```

R2

```
R2(config)#ip route 11.11.11.0 255.255.255.0 f1/0
R2(config)#ip route 33.33.33.0 255.255.255.0 f2/0
R2(config)#ip route 44.44.44.0 255.255.255.0 f2/0
```

R3

```
R3(config)#ip route 11.11.11.0 255.255.255.0 f1/0
R3(config)#ip route 22.22.22.0 255.255.255.0 f2/0
R3(config)#ip route 44.44.44.0 255.255.255.0 f1/0
```

R4

```
R4(config)#ip route 11.11.11.0 255.255.255.0 f2/0
R4(config)#ip route 22.22.22.0 255.255.255.0 f2/0
R4(config)#ip route 33.33.33.0 255.255.255.0 f1/0
```

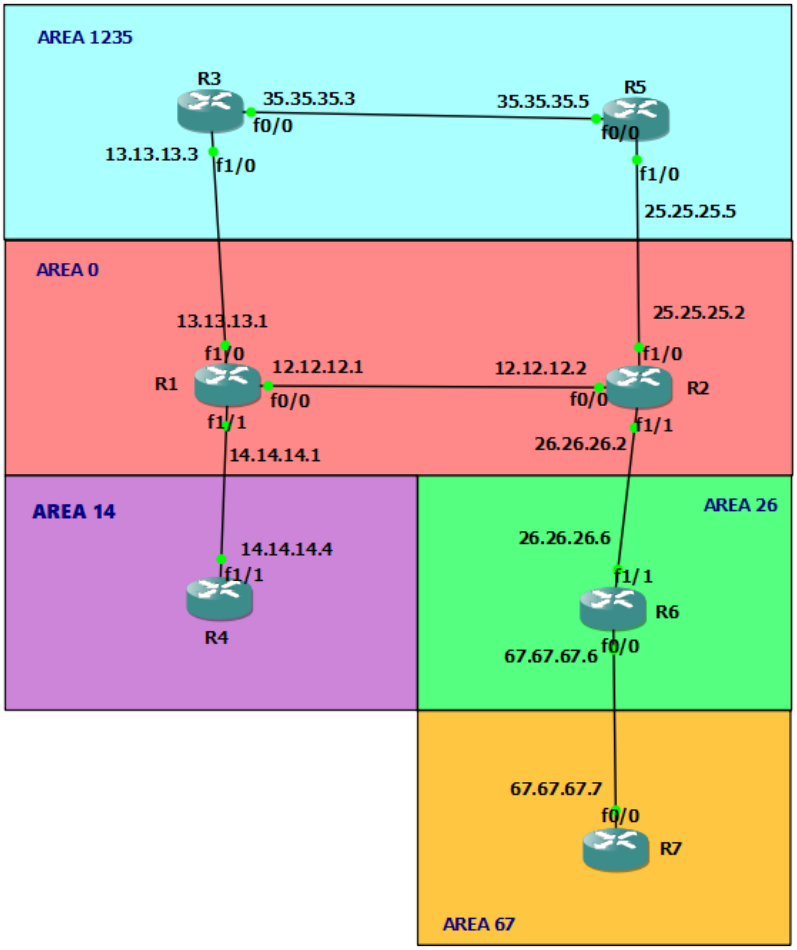
CHAPTER 5

OSPF

5.1 Definition

Open Shortest Path First (OSPF) is a routing protocol in form of a **graph**, operating within a single **autonomous system (AS)** which here we call it an **Area**.

5.2 Project description



In this project, we want to ping routers from each other. There are 7 routers in 5 Areas which has different background colors in the image.

Warning: Because we should set the area of the *edge routers* same as the area of destination router, here in this examples *area 67* and *area 14* are deleted as the area only consists of an edge router.

5.3 Configuration

Note: Becuase configuration of the interfaces and VPCs IPs are similar to the previous projects, I simply just write the routing codes.

Warning: The area number of the backbone area should be lower than others, it won't work otherwise. In this project it is area 0.

5.3.1 Routers config

R1

```
R1(config)#router ospf 1
R1(config-router)#network 12.12.12.0 255.255.255.0 area 0
R1(config-router)#network 13.13.13.0 255.255.255.0 area 0
R1(config-router)#network 14.14.14.0 255.255.255.0 area 0
```

R2

```
R2(config)#router ospf 2
R2(config-router)#network 12.12.12.0 255.255.255.0 area 0
R2(config-router)#network 25.25.25.0 255.255.255.0 area 0
R2(config-router)#network 26.26.26.0 255.255.255.0 area 0
```

R3

```
R3(config)#router ospf 3
R3(config-router)#network 13.13.13.0 255.255.255.0 area 0
R3(config-router)#network 35.35.35.0 255.255.255.0 area 1235
```

R4

```
R4(config)#router ospf 4
R4(config-router)#network 14.14.14.0 255.255.255.0 area 0
```

R5

```
R5(config)#router ospf 5
R5(config-router)#network 25.25.25.0 255.255.255.0 area 0
R5(config-router)#network 35.35.35.0 255.255.255.0 area 1235
```

R6

```
R6(config)#router ospf 6
R6(config-router)#network 26.26.26.0 255.255.255.0 area 0
R6(config-router)#network 67.67.67.0 255.255.255.0 area 26
```

R7

```
R7(config)#router ospf 7
R7(config-router)#network 67.67.67.0 255.255.255.0 area 26
```

5.3.2 View Connection details

Use `sh ip route` and `sh ip protocol` to see the routes and connection details.

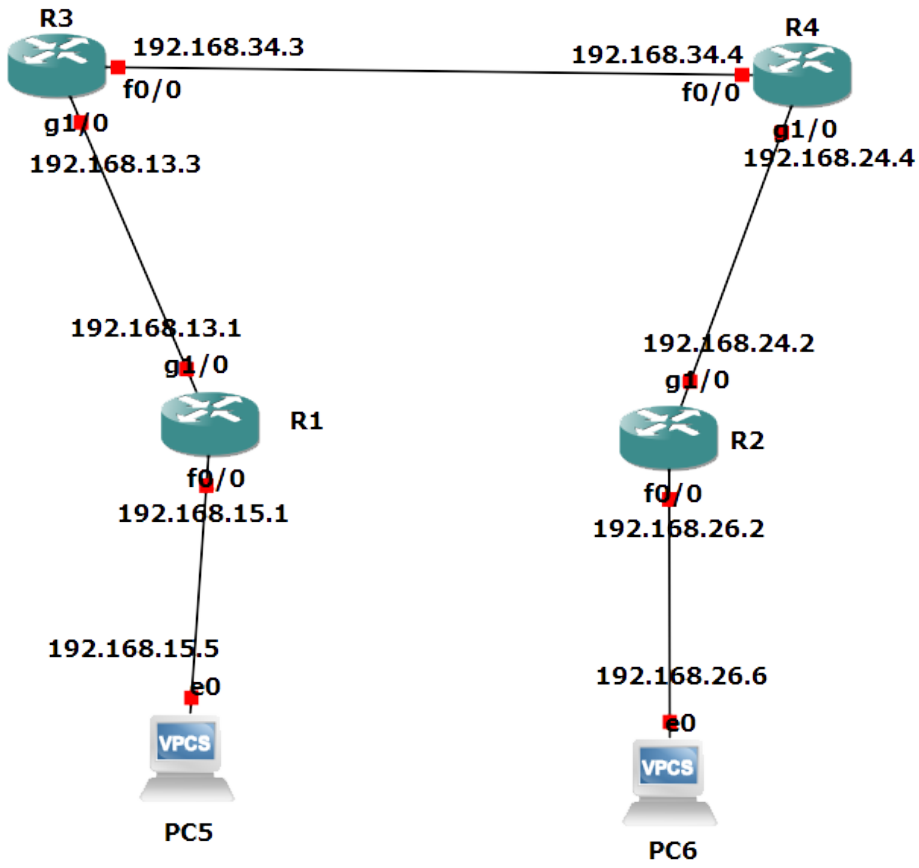
CHAPTER 6

EIGRP

6.1 Definition

Enhanced Interior Gateway Routing Protocol (EIGRP) is an advanced distance-vector routing protocol that is used on a computer network for automating routing decisions and configuration

6.2 Project description



In this project, we want to ping PC5 and PC6 from each other using EIGRP *in the same area*.

6.3 Configuration

Note: Becuase configuration of the interfaces and VPCs IPs are similar to the previous projects, I simply just write the routing codes.

6.3.1 Routers config

Warning: You should use **same area number** for routers, ping won't work otherwise!

R1

```
R1(config)#router eigrp 1
R1(config-router)#network 192.168.13.0 255.255.255.0
R1(config-router)#network 192.168.15.0 255.255.255.0
```

R2

```
R2(config)#router eigrp 1
R2(config-router)#network 192.168.24.0 255.255.255.0
R2(config-router)#network 192.168.26.0 255.255.255.0
```

R3

```
R3(config)#router eigrp 1
R3(config-router)#network 192.168.13.0 255.255.255.0
R3(config-router)#network 192.168.34.0 255.255.255.0
```

R4

```
R4(config)#router eigrp 1
R4(config-router)#network 192.168.24.0 255.255.255.0
R4(config-router)#network 192.168.34.0 255.255.255.0
```

6.3.2 View Connection details

Use `sh ip route` and `sh ip protocol` to see the routes and connection details.

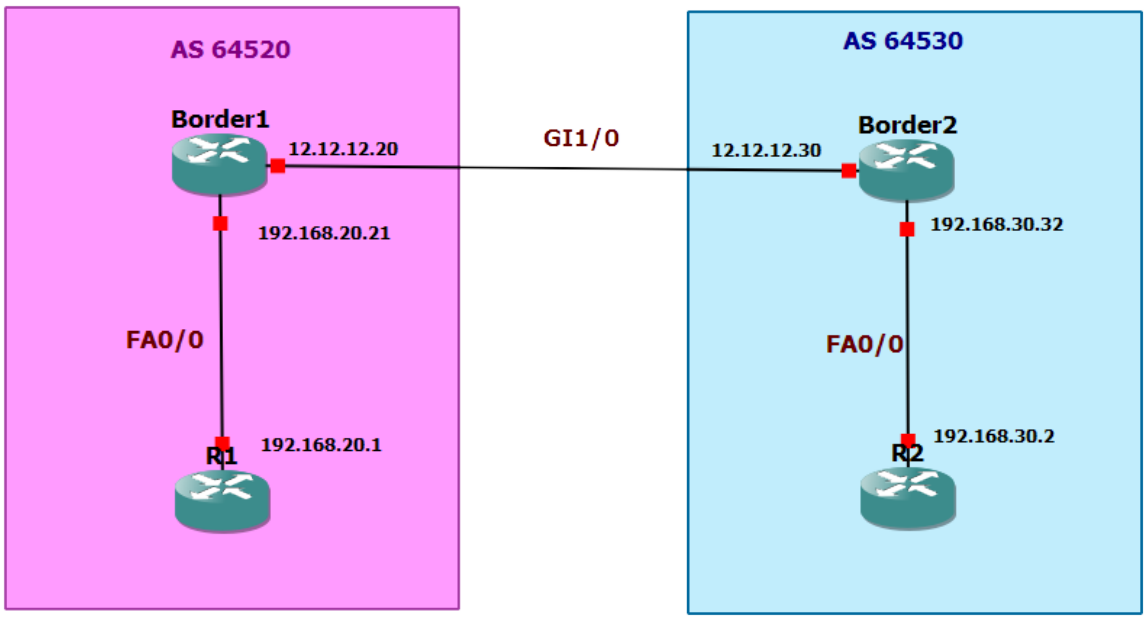
CHAPTER 7

BGP

7.1 Definition

BGP (Border Gateway Protocol) is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers

7.2 Project description



In this project, we want to ping R1 and R2 from each other using BGP.

7.3 Configuration

Note: Becuase configuration of the interfaces and VPCs IPs are similar to the previous projects, I simply just write the routing codes.

7.3.1 Routers config

Border1

```
Border1(config)#router bgp 64520
Border1(config-router)#network 12.12.12.0 mask 255.255.255.0
Border1(config-router)#network 192.168.20.0
Border1(config-router)#neighbor 12.12.12.30 remote-as 64530
Border1(config-router)#neighbor 192.168.20.1 remote-as 64520
Border1(config-router)#neighbor 192.168.20.1 next-hop-self
```

Border2

```
Border2(config)#router bgp 64530
Border2(config-router)#network 12.12.12.0 mask 255.255.255.0
Border2(config-router)#network 192.168.30.0
Border2(config-router)#neighbor 12.12.12.20 remote-as 64520
Border2(config-router)#neighbor 192.168.30.2 remote-as 64530
Border2(config-router)#neighbor 192.168.30.2 next-hop-self
```

R1

```
R1(config)#router bgp 64520
R1(config-router)#network 192.168.20.0
R1(config-router)#neighbor 192.168.20.21 remote-as 64520
```

R2

```
R2(config)#router bgp 64530
R2(config-router)#network 192.168.30.0
R2(config-router)#neighbor 192.168.30.32 remote-as 64530
```

7.3.2 View Connection details

Use `sh ip route` and `sh ip protocol` to see the routes and connection details.

CHAPTER 8

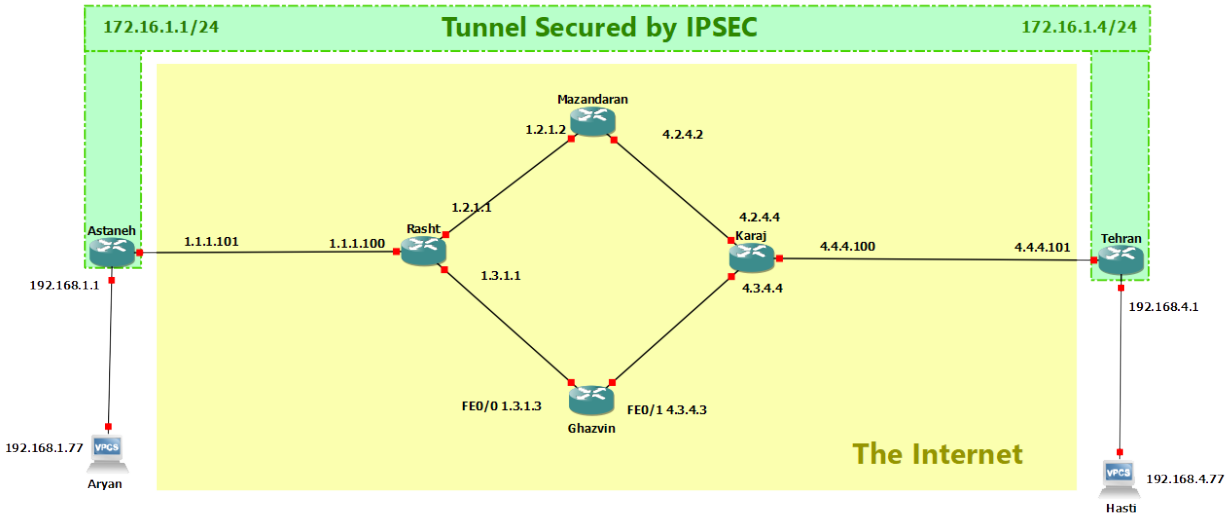
IPSec over GRE

8.1 Definition

Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links or point-to-multipoint links over an Internet Protocol network.

Internet Protocol Security (IPsec) is a secure network protocol suite that authenticates and encrypts the packets of data sent over an internet protocol network. It is used in virtual private networks (VPNs).

8.2 Project description



NETWORK LABRATORY

Students: Aryan Ebrahimpour, Hasti Hassani Moughadam

GRE and IPSEC tunnel

Here in the image, the green tunnel is the GRE tunnel which is secured by IPSEC protocol. We want to secure the packets that 'Aryan' and 'Hasti' PCs are sending to eachother.

8.3 Configuration

Note: You can use any routing algorithm you learnt in previous sections for the four router in the middle (called Internet). In this section, I ignore the four routers and assume that they are preconfigured.

8.3.1 Routers config

Note: You can change the the key part(`hastiaryan`) and the profile name part(`OurProfile`) to your custom names.

Astaneh

```
Astaneh(config)#crypto isakmp policy 10
Astaneh(config-isakmp)#authentication pre-share
Astaneh(config-isakmp)#exit
Astaneh(config)#crypto isakmp key hastiaryan address 4.4.4.101
Astaneh(config)#crypto ipsec transform-set 3des-sha esp-3des esp-sha-hmac
```

(continues on next page)

(continued from previous page)

```
Astaneh(config)#crypto ipsec profile OurProfile
Astaneh(ipsec-profile)#set transform-set 3des-sha
Astaneh(ipsec-profile)#exit

Astaneh(config)#interface Tunnel0
Astaneh(config-if)#ip address 172.16.1.1 255.255.255.0
Astaneh(config-if)#tunnel source FastEthernet0/0
Astaneh(config-if)#tunnel destination 4.4.4.101
Astaneh(config-if)#tunnel protection ipsec profile OurProfile
Astaneh(config-if)#exit

Astaneh(config)#ip route 192.168.4.0 255.255.255.0 Tunnel0
```

Tehran

```
Tehran(config)#crypto isakmp policy 10
Tehran(config-isakmp)#authentication pre-share
Tehran(config-isakmp)#exit
Tehran(config)#crypto isakmp key hastiaryan address 1.1.1.101
Tehran(config)#crypto ipsec transform-set 3des-sha esp-3des esp-sha-hmac

Tehran(config)#crypto ipsec profile OurProfile
Tehran(ipsec-profile)#set transform-set 3des-sha
Tehran(ipsec-profile)#exit

Tehran(config)#interface Tunnel0
Tehran(config-if)#ip address 172.16.1.4 255.255.255.0
Tehran(config-if)#tunnel source FastEthernet0/0
Tehran(config-if)#tunnel destination 1.1.1.101
Tehran(config-if)#tunnel protection ipsec profile OurProfile
Tehran(config-if)#exit

Tehran(config)#ip route 192.168.1.0 255.255.255.0 Tunnel0
```

CHAPTER 9

Central DHCP Server

Warning: Under construction

CHAPTER 10

Indices and tables

- genindex
- modindex
- search