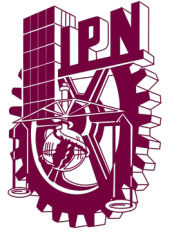




Instituto Politecnico Nacional

Escuela Superior de Computo



Practica 4.1 Enrutamiento multiple

Alumno: Javier Martinez Carranza

Grupo: 4CM11

Materia: Administracion de Servicios en Red

Profesor: Ricardo Martinez Rosales

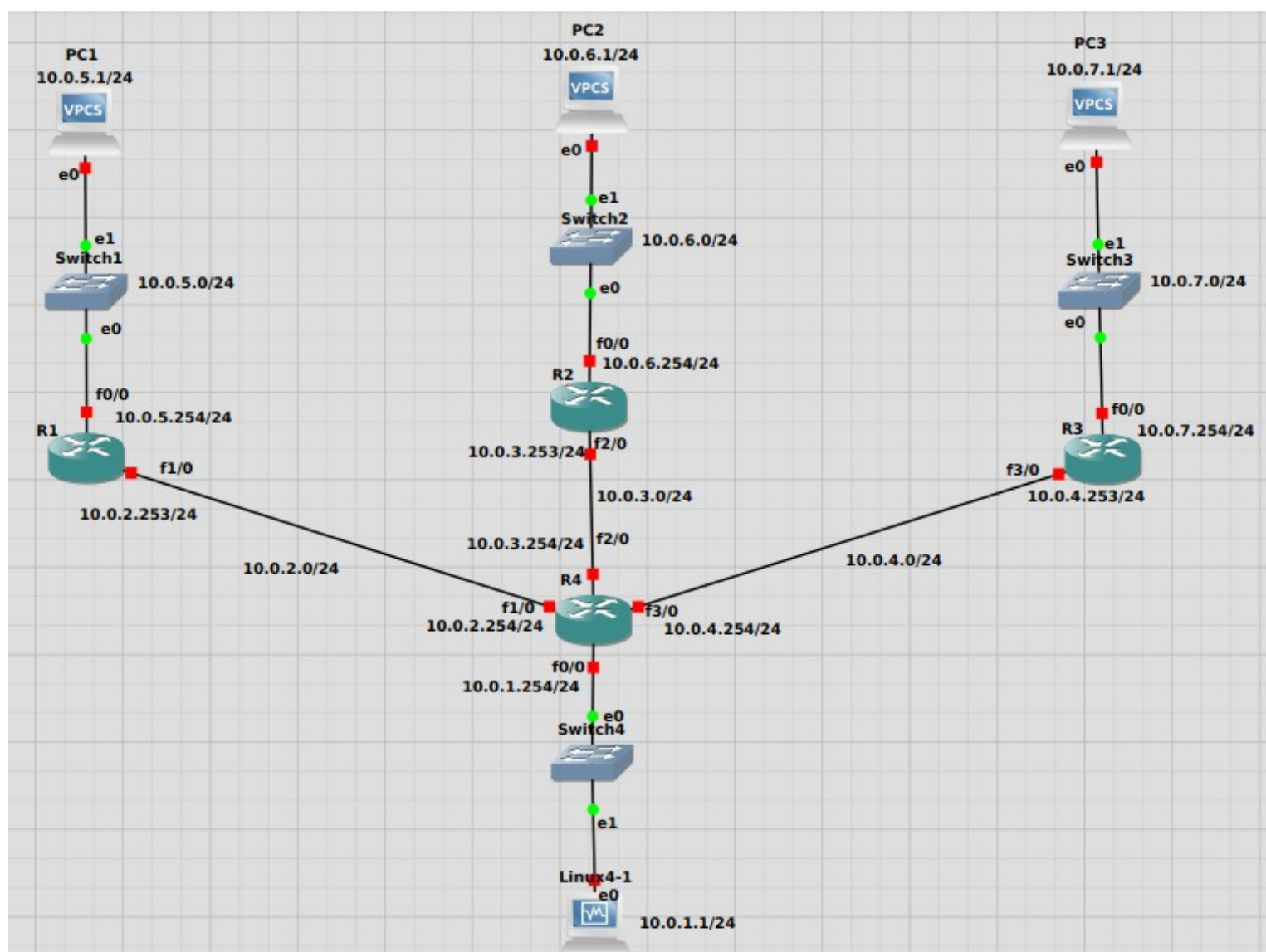
Topologia que sera usada en esta practica

EL router 4 sera configurado simplemente en cada una de sus interfaces, no configuraremos ningun protocolo de enrutamiento dinamico ni estatico, este proceso se realizara despues desde la maquina virtual y Flask.

El router 3 tendra configurado OSPF.

El router 2 tendra configurado RIP.

El router 1 sera configurado con enrutamiento estatico.



Podemos observar como en primera instancia aunque este configurado RIP y OSPF no se muestra en la tabla de enrutamiento porque no tiene conexión con R4 que es el que se configurara despues.

R4

```

E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

4.0.0.0/24 is subnetted, 1 subnets
C 4.4.4.0 is directly connected, Loopback0
10.0.0.0/24 is subnetted, 4 subnets
C 10.0.2.0 is directly connected, FastEthernet1/0
C 10.0.3.0 is directly connected, FastEthernet2/0
C 10.0.1.0 is directly connected, FastEthernet0/0
C 10.0.4.0 is directly connected, FastEthernet3/0
C 192.168.1.0/24 is directly connected, Loopback1
R4#

```

R3

```

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

3.0.0.0/24 is subnetted, 1 subnets
C 3.3.3.0 is directly connected, Loopback0
10.0.0.0/24 is subnetted, 2 subnets
C 10.0.7.0 is directly connected, FastEthernet0/0
C 10.0.4.0 is directly connected, FastEthernet3/0
R3#

```

R2

```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 2 subnets
C 10.0.3.0 is directly connected, FastEthernet2/0
C 10.0.6.0 is directly connected, FastEthernet0/0
R2#

```

R1

```

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 5 subnets
C 10.0.2.0 is directly connected, FastEthernet1/0
S 10.0.3.0 [1/0] via 10.0.2.254
S 10.0.1.0 [1/0] via 10.0.2.254
S 10.0.4.0 [1/0] via 10.0.2.254
C 10.0.5.0 is directly connected, FastEthernet0/0
R1#

```

R2

```

Sending updates every 30 seconds, next due in 20 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Redistributing: rip
Default version control: send version 2, receive version 2
Interface          Send Recv Triggered RIP Key-chain
FastEthernet0/0      2      2
FastEthernet2/0      2      2

Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
 10.0.0.0
Routing Information Sources:
 Gateway      Distance      Last Update
Distance: (default is 120)
R2#

```

R3

```

C 10.0.4.0 is directly connected, FastEthernet3/0
R3#show ip protocols
R3#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 3.3.3.3
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.4.0 0.0.0.255 area 0
    10.0.7.0 0.0.0.255 area 0
  Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway      Distance      Last Update
  Distance: (default is 110)
R3#

```

El código que se usará con Flask es el siguiente:

```
from flask import Flask

import conf_ospf, conf_rip, conf_static, json

app = Flask(__name__)

with open('dispositivos.json', 'r') as f:
    ID_redes = json.load(f)
    host = ID_redes["HOST"]["ip"]
    user = ID_redes["HOST"]["user"]
    passw = ID_redes["HOST"]["password"]
    red_ip = ID_redes["RED"]["ip"]
    red_wild = ID_redes["RED"]["wild"]

@app.route('/')
def index():
    return "Flask"

@app.route('/ospf')
def confOSPF():
    red_ip2 = ID_redes["OSPF"]["ip"]
    red_wild2 = ID_redes["OSPF"]["wild"]
    loop = ID_redes["OSPF"]["id"]
    res = conf_ospf.configura(host, user, passw, red_ip, red_wild, red_ip2, red_wild2, loop)
    return res

@app.route('/rip')
def confRIP():
    red_ip2 = ID_redes["RIP"]["ip"]
    res = conf_rip.configura(host, user, passw, red_ip, red_ip2)
    return res

@app.route('/static')
def confSTATIC():
    red_ip = ID_redes["STATIC"]["ip"]
    netmask = ID_redes["STATIC"]["netmask"]
    inter = ID_redes["STATIC"]["inter"]
    res = conf_static.configura(host, user, passw, red_ip, netmask, inter)
    return res
```

Dentro de nuestro archivo dispositivos.json tenemos ciertos datos que nos ayudarán a hacer una configuración más rápida como id de red o netmask de la red que se configurará en cada caso. Además tenemos una ruta para cada tipo de configuración “/static”, “/ospf” y “/rip” con las cuales mandamos a llamar a ciertas funciones que se encargarán de hacer el trabajo.

Para el enrutamiento OSPF se necesita hacer uso de las ips y wildcards de cada rubred que queramos configurar en este modo de enrutamiento, haremos uso de paramiko y un usuario configurado previamente en el router 4, una vez tengamos acceso, solo hara falta mandar comandos desde Flask al router

```
def configura(host,user,passw,ip1,wildcard1,ip2,wildcard2,id):
    conexion = paramiko.SSHClient()
    conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    conexion.connect(host, username=user, password=passw, look_for_keys=False, allow_agent=False)
    nueva_conexion = conexion.invoke_shell()
    nueva_conexion.send("configure terminal\n")
    time.sleep(0.2)
    nueva_conexion.send("router ospf 1\n")
    time.sleep(0.2)
    nueva_conexion.send("redistribute rip subnets\n")
    time.sleep(0.2)
    #nueva_conexion.send("redistribute static\n")
    #time.sleep(0.2)
    nueva_conexion.send("router-id "+id+"\n")
    time.sleep(0.2)
    nueva_conexion.send("network "+ip1+" "+wildcard1+" area 0\n")
    time.sleep(0.2)
    nueva_conexion.send("network "+ip2+" "+wildcard2+" area 0\n")
    time.sleep(0.2)
    nueva_conexion.send("end\n")
    time.sleep(0.2)
    salida = str(nueva_conexion.recv(3000))
    conexion.close()

    return str(salida)
```

```
def configura(host,user,passw,ip1,ip2):
    conexion = paramiko.SSHClient()
    conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    conexion.connect(host, username=user, password=passw, look_for_keys=False, allow_agent=False)
    nueva_conexion = conexion.invoke_shell()
    nueva_conexion.send("configure terminal\n")
    time.sleep(0.2)
    nueva_conexion.send("router rip\n")
    time.sleep(0.2)
    nueva_conexion.send("version 2\n")
    time.sleep(0.2)
    nueva_conexion.send("redistribute ospf 1 metric 1\n")
    time.sleep(0.2)
    nueva_conexion.send("network "+ip1+" \n")
    time.sleep(0.2)
    nueva_conexion.send("network "+ip2+" \n")
    time.sleep(0.2)
    nueva_conexion.send("end\n")
    time.sleep(0.2)
    salida = str(nueva_conexion.recv(3000))
    conexion.close()

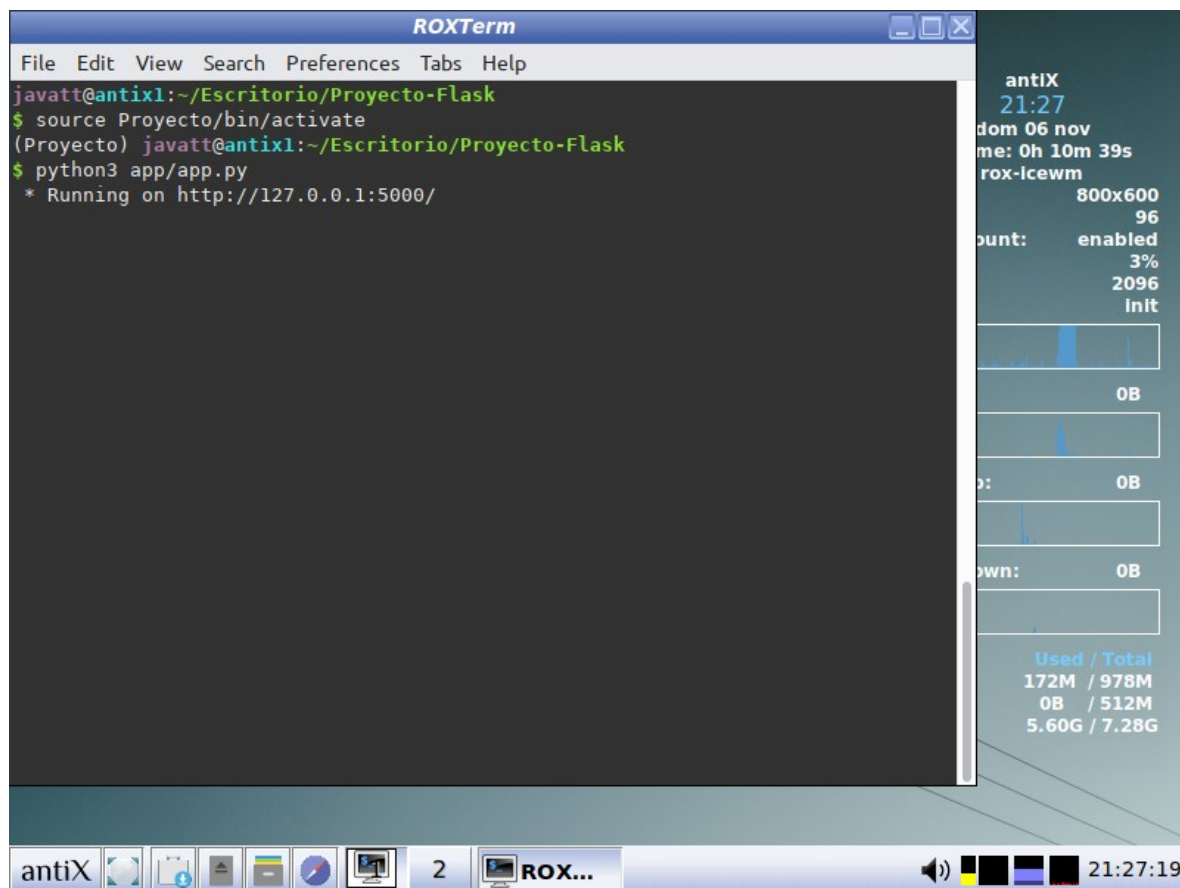
    return str(salida)
```

```
def configura(host,user,passw,ip,netmask,interfaz):
    conexion = paramiko.SSHClient()
    conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    conexion.connect(host, username=user, password=passw, look_for_keys=False, allow_agent=False)
    nueva_conexion = conexion.invoke_shell()
    nueva_conexion.send("configure terminal\n")
    time.sleep(0.2)
    nueva_conexion.send("ip route "+ip+" "+netmask+" "+interfaz+" \n")
    time.sleep(0.2)
    nueva_conexion.send("end\n")
    time.sleep(0.2)
    salida = str(nueva_conexion.recv(3000))
    conexion.close()

    return str(salida)
```

haremos lo mismo con las otras funciones solo cambiando la forma de configuracion.

Al ejecutar el proyecto en Flask



Ejecutaremos primero la configuracion de OSPF

The image shows a network simulation environment. On the left, a terminal window for router R4 displays the following output:

```
R4#  
Gateway of last resort is not set  
Cloud  
  4.0.0.0/24 is subnetted, 1 subnets  
C    4.4.4.0 is directly connected, Loopback0  
C    10.0.0.0/24 is subnetted, 4 subnets  
C    10.0.2.0 is directly connected, FastEthernet1/0  
C    10.0.3.0 is directly connected, FastEthernet2/0  
C    10.0.1.0 is directly connected, FastEthernet0/0  
C    10.0.4.0 is directly connected, FastEthernet3/0  
C    192.168.1.0/24 is directly connected, Loopback1  
R4#  
R4#  
R4#AT  
R4#  
R4#show ip pro  
R4#show ip protocols  
R4#
```

On the right, a Mozilla Firefox browser window is open with the address bar showing `127.0.0.1:5000/ospf`. The page content displays the OSPF configuration commands for R4:

```
b"\nR4#configure terminal\nEnter configuration commands, one per line. End with  
CNTL/Z.\nR4(config)#router ospf 1\nR4(config-router)#redistribute rip subnets  
\nR4(config-router)#router-id 192.168.1.0\nR4(config-router)#network 10.0.1.0 0.0.0.255  
area 0 \nR4(config-router)#network 10.0.4.0 0.0.0.255 area 0 \nR4(config-router)#end  
\nR4#'
```

The image shows a network simulation environment. On the left, a terminal window for router R4 displays the following output:

```
R4#  
*Nov  7 05:28:16.719: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet3/  
0 from LOADING to FULL, Loading Done  
R4#show ip route  
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP  
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
        E1 - OSPF external type 1, E2 - OSPF external type 2  
        I - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
        ia - IS-IS inter area, * - candidate default, U - per-user static route  
        o - ODR, P - periodic downloaded static route  
Gateway of last resort is not set  
VP4  
  4.0.0.0/24 is subnetted, 1 subnets  
C    4.4.4.0 is directly connected, Loopback0  
C    10.0.0.0/24 is subnetted, 5 subnets  
C    10.0.2.0 is directly connected, FastEthernet1/0  
C    10.0.3.0 is directly connected, FastEthernet2/0  
C    10.0.1.0 is directly connected, FastEthernet0/0  
O    10.0.7.0 [110/2] via 10.0.4.253, 00:01:06, FastEthernet3/0  
C    10.0.4.0 is directly connected, FastEthernet3/0  
C    192.168.1.0/24 is directly connected, Loopback1  
R4#
```

On the right, a network diagram is visible, showing a topology with routers R2, R4, and R5, and various interfaces connected. The diagram includes labels for interfaces like `f0/0`, `f1/0`, and `f2/0`, and IP addresses like `10.0.5.254/24`, `10.0.3.253/24`, and `10.0.2.254/24`.

Con RIP y static



b"\r\nR4#configure terminal\r\nEnter configuration commands, one per line. End with CNTL/Z.\r\nR4(config)#router rip\r\nR4(config-router)#version 2\r\nR4(config-router)#redistribute ospf 1 metric 1\r\nR4(config-router)#network 10.0.1.0 \r\nR4(config-router)#network 10.0.3.0 \r\nR4(config-router)#end\r\nR4#'

