

UNIVERSIDAD NACIONAL DE CÓRDOBA
Facultad de Ciencias Exactas, Físicas y Naturales



Redes de Computadoras

Práctico 5: Ruteo Internet

Contrera, Ivan
Malano, Leandro

1-2-3) Los primeros 3 pasos se hicieron tal cual como en el trabajo práctico número 4.

4) Al igual que en el trabajo 4, se ejecuta el comando *docker-compose up* sobre el nuevo proyecto. Se comprueba que todo arranque correctamente, donde se muestra una pantalla como la siguiente:

```
b2_1 | 2018-05-09 22:32:35,000 WARN Included extra file "/etc/supervisor/conf.d/supervisord.conf" during parsing
r2_1 | 2018-05-09 22:32:35,000 WARN Included extra file "/etc/supervisor/conf.d/supervisord.conf" during parsing
b1_1 | 2018-05-09 22:32:35,023 CRIT Supervisor running as root (no user in config file)
b1_1 | 2018-05-09 22:32:35,024 WARN Included extra file "/etc/supervisor/conf.d/supervisord.conf" during parsing
r2_1 | 2018-05-09 22:32:35,225 INFO RPC interface 'supervisor' initialized
b2_1 | 2018-05-09 22:32:35,232 INFO RPC interface 'supervisor' initialized
r1_1 | 2018-05-09 22:32:35,233 INFO RPC interface 'supervisor' initialized
r2_1 | 2018-05-09 22:32:35,234 CRIT Server 'unix_http_server' running without any HTTP authentication checking
r1_1 | 2018-05-09 22:32:35,234 CRIT Server 'unix_http_server' running without any HTTP authentication checking
r2_1 | 2018-05-09 22:32:35,235 INFO supervisord started with pid 1
r1_1 | 2018-05-09 22:32:35,236 INFO supervisord started with pid 1
b2_1 | 2018-05-09 22:32:35,239 CRIT Server 'unix_http_server' running without any HTTP authentication checking
b2_1 | 2018-05-09 22:32:35,240 INFO supervisord started with pid 1
b1_1 | 2018-05-09 22:32:35,243 INFO RPC interface 'supervisor' initialized
b1_1 | 2018-05-09 22:32:35,244 CRIT Server 'unix_http_server' running without any HTTP authentication checking
b1_1 | 2018-05-09 22:32:35,247 INFO supervisord started with pid 1
r2_1 | 2018-05-09 22:32:36,280 INFO spawned: 'ospf6d' with pid 7
r1_1 | 2018-05-09 22:32:36,284 INFO spawned: 'ospf6d' with pid 7
b2_1 | 2018-05-09 22:32:36,288 INFO spawned: 'ospf6d' with pid 7
b1_1 | 2018-05-09 22:32:36,292 INFO spawned: 'ospf6d' with pid 7
b1_1 | 2018-05-09 22:32:36,348 INFO spawned: 'bgpd' with pid 8
r2_1 | 2018-05-09 22:32:36,343 INFO spawned: 'zebra' with pid 8
b2_1 | 2018-05-09 22:32:36,351 INFO spawned: 'bgpd' with pid 8
r1_1 | 2018-05-09 22:32:36,364 INFO spawned: 'zebra' with pid 8
b2_1 | 2018-05-09 22:32:36,407 INFO spawned: 'zebra' with pid 9
b1_1 | 2018-05-09 22:32:36,393 INFO spawned: 'zebra' with pid 9
r2_1 | 2018-05-09 22:32:37,395 INFO success: ospf6d entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
r2_1 | 2018-05-09 22:32:37,395 INFO success: zebra entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
r1_1 | 2018-05-09 22:32:37,403 INFO success: ospf6d entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
r1_1 | 2018-05-09 22:32:37,403 INFO success: zebra entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b1_1 | 2018-05-09 22:32:37,432 INFO success: ospf6d entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b1_1 | 2018-05-09 22:32:37,433 INFO success: bgpd entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b1_1 | 2018-05-09 22:32:37,433 INFO success: zebra entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b2_1 | 2018-05-09 22:32:37,441 INFO success: ospf6d entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b2_1 | 2018-05-09 22:32:37,441 INFO success: bgpd entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
b2_1 | 2018-05-09 22:32:37,442 INFO success: zebra entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
```

5) En el archivo “dokcer-compose.yml”, las secciones son similares al analizado en el trabajo anterior, las diferencias están en los routers de borde que, además de correr OSPF, corre BGP. En *volumes*, se agrega la línea */volumes/bgp/b1/bgpd.conf:/etc/quagga/bgpd.conf:ro*, la cual copia las configuraciones establecidas por el operador de la red para dicho contenedor (b1 en este caso). Además se define un numero de puerto a dicho contenedor que se asocia con el puerto de BGP (2605), y de esta forma acceder a las configuraciones de ese protocolo mediante telnet:

```
root@ivanovic-VirtualBox:/home/ivanovic/docker_quagga/bgp# telnet localhost 10104
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.

Hello, this is Quagga (version 0.99.23.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
b1> █
```

6) Para los routers internos, la configuración no varía demasiado respecto a la trabajo anterior, ya que solamente utilizan OSPF para descubrir rutas. La diferencia está en las áreas en la que pertenece cada sistema autónomo, ya que cada uno tiene una distinta. La mayor diferencia está en los routers de borde, donde se agrega un archivo de configuración llamado *bgpd.conf*, donde se especifican algunos parámetros, como la familia de redes de su sistema autónomo (*address-family*), el agregado de vecinos a una lista, y su habilitación.

6.1) El password se ubica en cada uno de los archivos de configuración, en la línea *password admin*.

6.2) Para eliminar el default gateway asignado por docker a cada contenedor, se ejecuta el comando *dokcer exec -it bgp_r1_1 ip -6 route del default* (ejemplo para r1).

7) En esta segunda parte del trabajo, se modifica el archivo *docker-compose.yml* para agregar los routers y hosts faltantes y se crean las carpetas correspondientes en el directorio *volumes*

8.1-2-3) Como se dijo en el punto anterior, se crean los archivos *bgpd.conf*, *ospf6.conf* y *zebra.conf* en carpetas llamadas r2, r3, r4,... como se hizo en el trabajo anterior. La diferencia está en el primer archivo, el cual configura el protocolo BGP. Además, en un script, se eliminan los default gateway y se agregan estaticamente los nuevos, que llevarán todos los paquetes, cuya dirección de destino no pertenezcan al área de origen, al router de borde de su AS.

9) A continuación, se muestran algunas capturas haciendo ping entre distintos puntos de todo el sistema:

ping entre h11 y h12:

```
root@ivan-ubuntu:/home/ivanovic/docker_quagga/bgp# docker exec -it bgp_h11_1 /bin/bash
root@43804bdb0917:/go# ping6 2001:aaaa:bbbb:1::10
PING 2001:aaaa:bbbb:1::10 (2001:aaaa:bbbb:1::10): 56 data bytes
64 bytes from 2001:aaaa:bbbb:1::10: icmp_seq=0 ttl=60 time=1.003 ms
64 bytes from 2001:aaaa:bbbb:1::10: icmp_seq=1 ttl=60 time=0.237 ms
^Z
[1]+  Stopped                  ping6 2001:aaaa:bbbb:1::10
```

ping entre h11 y h14:

```
root@43804bdb0917:/go# ping6 2001:aaaa:bbbb:3::20
PING 2001:aaaa:bbbb:3::20 (2001:aaaa:bbbb:3::20): 56 data bytes
64 bytes from 2001:aaaa:bbbb:3::20: icmp_seq=0 ttl=60 time=0.607 ms
64 bytes from 2001:aaaa:bbbb:3::20: icmp_seq=1 ttl=60 time=0.269 ms
64 bytes from 2001:aaaa:bbbb:3::20: icmp_seq=2 ttl=60 time=0.214 ms
^Z
[2]+  Stopped                  ping6 2001:aaaa:bbbb:3::20
```

ping entre h11 y r2:

```
root@43804bdb0917:/go# ping6 2001:aaaa:cccc:1::3
PING 2001:aaaa:cccc:1::3 (2001:aaaa:cccc:1::3): 56 data bytes
64 bytes from 2001:aaaa:cccc:1::3: icmp_seq=0 ttl=61 time=0.450 ms
64 bytes from 2001:aaaa:cccc:1::3: icmp_seq=1 ttl=61 time=0.254 ms
64 bytes from 2001:aaaa:cccc:1::3: icmp_seq=2 ttl=61 time=0.231 ms
^Z
[3]+  Stopped                  ping6 2001:aaaa:cccc:1::3
```

ping entre r3 y r6:

```
root@ivan-ubuntu:/home/ivanovic/docker_quagga/bgp# docker exec -it bgp_r3_1 /bin/bash
root@5927500aa132:/go# ping6 2001:aaaa:bbbb:2::3
PING 2001:aaaa:bbbb:2::3 (2001:aaaa:bbbb:2::3): 56 data bytes
64 bytes from 2001:aaaa:bbbb:2::3: icmp_seq=0 ttl=62 time=0.255 ms
64 bytes from 2001:aaaa:bbbb:2::3: icmp_seq=1 ttl=62 time=0.170 ms
^Z
```

Los routers de borde tienen una tabla de enrutamiento, donde en parte algunas rutas se aprendieron por ospf y otras por bgp. En las siguientes capturas se muestran las tablas de los tres routers de borde, donde antes de cada ruta se aclara el protocolo que la obtuvo (B = BGP, O = OSPF y C = conectada directamente):

b1:

```
C>* ::1/128 is directly connected, lo
O>* 2001:aaaa:aaaa:1::/64 [110/2] via fe80::42:acff:fe12:2, eth0, 00:37:23
O 2001:aaaa:aaaa:2::/64 [110/1] is directly connected, eth0, 00:37:23
C>* 2001:aaaa:aaaa:3::/64 is directly connected, eth0
O>* 2001:aaaa:aaaa:4::/64 [110/2] via fe80::42:acff:fe13:2, eth1, 00:37:20
O>* 2001:aaaa:aaaa:5::/64 [110/2] via fe80::42:acff:fe12:2, eth0, 00:37:20
O 2001:aaaa:aaaa:6::/64 [110/1] is directly connected, eth1, 00:37:25
C>* 2001:aaaa:aaaa:7::/64 is directly connected, eth1
O 2001:aaaa:aaaa:8::/64 [110/1] is directly connected, eth4, 00:37:28
C>* 2001:aaaa:aaaa:9::/64 is directly connected, eth4
B>* 2001:aaaa:bbbb::/48 [20/0] via fe80::42:acff:fe16:2, eth2, 00:38:14
B>* 2001:aaaa:cccc::/48 [20/0] via fe80::42:acff:fe15:3, eth3, 00:38:09
C>* 2001:bbbb::/64 is directly connected, eth3
C>* 2001:dddd::/64 is directly connected, eth2
```

b2:

```
C>* ::1/128 is directly connected, lo
B>* 2001:aaaa:aaaa::/48 [20/0] via fe80::42:acff:fe16:3, eth2, 00:47:13
O>* 2001:aaaa:bbbb:1::/64 [110/2] via fe80::42:acff:fe19:2, eth0, 00:46:23
O 2001:aaaa:bbbb:2::/64 [110/1] is directly connected, eth0, 00:46:25
C>* 2001:aaaa:bbbb:3::/64 is directly connected, eth0
O>* 2001:aaaa:bbbb:4::/64 [110/2] via fe80::42:acff:fe1a:2, eth1, 00:46:18
O>* 2001:aaaa:bbbb:5::/64 [110/2] via fe80::42:acff:fe19:2, eth0, 00:46:18
O 2001:aaaa:bbbb:6::/64 [110/1] is directly connected, eth1, 00:46:23
C>* 2001:aaaa:bbbb:7::/64 is directly connected, eth1
O 2001:aaaa:bbbb:8::/64 [110/1] is directly connected, eth4, 00:46:28
C>* 2001:aaaa:bbbb:9::/64 is directly connected, eth4
B>* 2001:aaaa:cccc::/48 [20/0] via fe80::42:acff:fe17:2, eth3, 00:47:09
C>* 2001:cccc::/64 is directly connected, eth3
C>* 2001:dddd::/64 is directly connected, eth2
```

b3:

```
C>* ::1/128 is directly connected, lo
B>* 2001:aaaa:aaaa::/48 [20/0] via fe80::42:acff:fe15:2, eth0, 03:08:07
B>* 2001:aaaa:bbbb::/48 [20/0] via fe80::42:acff:fe17:3, eth2, 03:08:08
C>* 2001:aaaa:cccc:1::/64 is directly connected, eth1
C>* 2001:bbbb::/64 is directly connected, eth0
O 2001:cccc::/64 [110/1] is directly connected, eth2, 03:08:08
C>* 2001:cccc::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth2
```

10) Haciendo un analisis de trafico con wireshark, se pudieron capturar 3 tipos de mensajes, de 4 posibles:

- **OPEN:** Se utiliza para establecer una conexion bgp una vez que se haya establecido la conexion TCP. Sucede apenas se inician los contenedores.


```

▶ Frame 69: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface 0
▶ Ethernet II, Src: 02:42:ac:15:00:02 (02:42:ac:15:00:02), Dst: 02:42:ac:15:00:03 (02:42:ac:15:00:03)
▶ Internet Protocol Version 6, Src: 2001:bbbb::3, Dst: 2001:bbbb::2
▶ Transmission Control Protocol, Src Port: 33196, Dst Port: 179, Seq: 1, Ack: 1, Len: 61
▼ Border Gateway Protocol - OPEN Message
  Marker: ffffffffffffffffffffffffffffffffff
  Length: 61
  Type: OPEN Message (1)
  Version: 4
  My AS: 3000
  Hold Time: 180
  BGP Identifier: 192.168.3.10
  Optional Parameters Length: 32
  ▶ Optional Parameters

```

En el encabezado se puede observar que se anuncia un router de borde del AS 3000, cuyo identificador es 192.168.3.10, tal cual se configuró en el archivo bgpd.conf del router.

- **KEEPALIVE:** Una vez establecida la conexión, se envían mensajes periódicos para mantener viva la conexión

```

▶ Frame 74: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0
▶ Ethernet II, Src: 02:42:ac:15:00:03 (02:42:ac:15:00:03), Dst: 02:42:ac:15:00:02 (02:42:ac:15:00:02)
▶ Internet Protocol Version 6, Src: 2001:bbbb::2, Dst: 2001:bbbb::3
▶ Transmission Control Protocol, Src Port: 179, Dst Port: 33196, Seq: 81, Ack: 100, Len: 19
▼ Border Gateway Protocol - KEEPALIVE Message
  Marker: ffffffffffffffffffffffffffffffffff
  Length: 19
  Type: KEEPALIVE Message (4)

```

- **UPDATE:** Este es muy importante, ya que contiene información de nuevos caminos y nuevas redes. Se envían cada vez que haya alguna modificación.

```

▶ Frame 78: 235 bytes on wire (1880 bits), 235 bytes captured (1880 bits) on interface 0
▶ Ethernet II, Src: 02:42:ac:15:00:03 (02:42:ac:15:00:03), Dst: 02:42:ac:15:00:02 (02:42:ac:15:00:02)
▶ Internet Protocol Version 6, Src: 2001:bbbb::2, Dst: 2001:bbbb::3
▶ Transmission Control Protocol, Src Port: 179, Dst Port: 33196, Seq: 100, Ack: 100, Len: 149
▶ Border Gateway Protocol - UPDATE Message
▼ Border Gateway Protocol - UPDATE Message
  Marker: ffffffffffffffffffffffffffffffffff
  Length: 73
  Type: UPDATE Message (2)
  Withdrawn Routes Length: 0
  Total Path Attribute Length: 50
  ▼ Path attributes
    ▶ Path Attribute - MP_REACH_NLRI
    ▶ Path Attribute - ORIGIN: IGP
    ▶ Path Attribute - AS_PATH: 1000 2000

```

Se informa que no existen rutas ya no alcanzables, algunos atributos de un determinado camino, donde aparece, entre otras cosas, el rango de redes de su sistema autónomo.

Mensajes de **NOTIFICATION** no aparecen ya que los routers no se han interrumpido.

11) Para que el AS 1000 prefiera conectarse con el AS 3000 a través del AS 2000 (ya que el 1000 está conectado directamente con ambos y podría evitarse esta política), basta con que el router b3 no acepte como vecino a b1. Antes de aplicar dicha política, se muestra el recorrido que se hace para llegar desde h11 hasta r2:

```

traceroute to 2001:aaaa:cccc:1::3 (2001:aaaa:cccc:1::3), 30 hops max, 80 byte packets
 1  bgp_r5_1.bgp_redA1 (2001:aaaa:aaaa:1::3)  0.149 ms  0.030 ms  0.026 ms
 2  2001:aaaa:aaaa:2::2 (2001:aaaa:aaaa:2::2)  0.122 ms  0.039 ms  0.036 ms
 3  2001:bbbb::3 (2001:bbbb::3)  0.067 ms  0.046 ms  0.046 ms
 4  2001:aaaa:cccc:1::3 (2001:aaaa:cccc:1::3)  0.160 ms  0.051 ms  0.048 ms

```

haciendo el cambio de politica, el nuevo recorrdio desde el mismo origen al mismo destino es:

```
root@ivan-ubuntu:/home/ivanovic/docker_quagga/bgp# docker exec bgp_h11_1 traceroute 2001:aaaa:cccc:1::3
traceroute to 2001:aaaa:cccc:1::3 (2001:aaaa:cccc:1::3), 30 hops max, 80 byte packets
 1 bgp_r5_1.bgp_redA1 (2001:aaaa:aaaa:1::3)  0.067 ms  0.013 ms  0.011 ms
 2 2001:aaaa:aaaa:2::2 (2001:aaaa:aaaa:2::2)  0.064 ms  0.027 ms  0.029 ms
 3 2001:dddd::2 (2001:dddd::2)  0.057 ms  0.041 ms  0.041 ms
 4 2001:cccc::3 (2001:cccc::3)  0.061 ms  0.047 ms  0.045 ms
 5 2001:aaaa:cccc:1::3 (2001:aaaa:cccc:1::3)  0.149 ms  0.046 ms  0.041 ms
```

se observa que ahora que el recorrido se hace a traves de la red 2001:dddd:: que interconecta los sistemas autonomos 1000 y 2000.

ANEXO:

Host (router o PC)	red [nro red - direcc red]	Dirección IP
h11	2001:aaaa:aaaa:1::/64	2001:aaaa:aaaa:1::10
h13	2001:aaaa:aaaa:3::/64	2001:aaaa:aaaa:3::20
r5	2001:aaaa:aaaa:1::/64	2001:aaaa:aaaa:1::3

r5	2001:aaaa:aaaa:4::/64	2001:aaaa:aaaa:4::3
r5	2001:aaaa:aaaa:2::/64	2001:aaaa:aaaa:2::3
r7	2001:aaaa:aaaa:4::/64	2001:aaaa:aaaa:4::2
r7	2001:aaaa:aaaa:3::/64	2001:aaaa:aaaa:3::3
r7	2001:aaaa:aaaa:5::/64	2001:aaaa:aaaa:5::3
r3	2001:aaaa:aaaa:6::/64	2001:aaaa:aaaa:6::2
	loopback	
b1	2001:aaaa:aaaa:6::/64	2001:aaaa:aaaa:6::3
b1	2001:aaaa:aaaa:2::/64	2001:aaaa:aaaa:2::2
b1	2001:aaaa:aaaa:5::/64	2001:aaaa:aaaa:5::2
b1	2001:bbbb::/64	2001:bbbb::2
b1	2001:dddd::/64	2001:bbbb::2
b3	2001:bbbb::/64	2001:bbbb::3
b3	2001:cccc::/64	2001:cccc::3
b3	2001:aaaa:cccc::/64	2001:aaaa:cccc::2
r2	2001:aaaa:cccc::/64	2001:aaaa:cccc::3
	loopback	
b2	2001:cccc::/64	2001:cccc::2
b2	2001:aaaa:bbbb:6::/64	2001:aaaa:bbbb:6::3
b2	2001:aaaa:bbbb:2::/64	2001:aaaa:bbbb:2::2
b2	2001:aaaa:bbbb:5::/64	2001:aaaa:bbbb:5::2
r4	2001:aaaa:bbbb:6::/64	2001:aaaa:bbbb:6::2
	loopkack	
r6	2001:aaaa:bbbb:2::/64	2001:aaaa:bbbb:2::3
r6	2001:aaaa:bbbb:4::/64	2001:aaaa:bbbb:4::3
r6	2001:aaaa:bbbb:1::/64	2001:aaaa:bbbb:1::3
r8	2001:aaaa:bbbb:4::/64	2001:aaaa:bbbb:4::2
r8	2001:aaaa:bbbb:5::/64	2001:aaaa:bbbb:5::3
r8	2001:aaaa:bbbb:3::/64	2001:aaaa:bbbb:3::3
h12	2001:aaaa:bbbb:1::/64	2001:aaaa:bbbb:1::10

h14	2001:aaaa:bbbb:3::/64	2001:aaaa:bbbb:3::20
-----	-----------------------	----------------------

ENLACES:

<https://www.cisco.com/c/en/us/support/docs/ip/ip-version-6-ipv6/113453-redis-ospfv3-bgp-00.html>

https://es.wikipedia.org/wiki/Border_Gateway_Protocol#Tipos_de_mensajes