

Práctica #4

Introducción

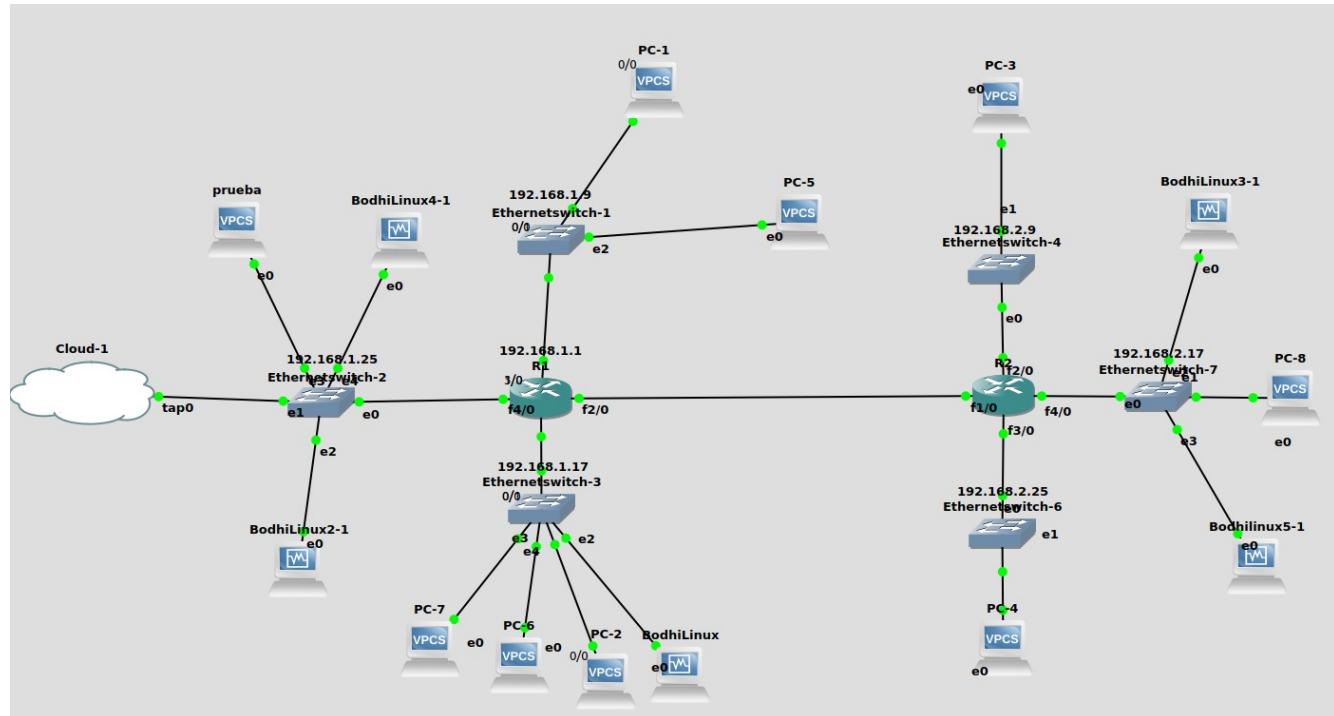
En la siguiente práctica se hace uso del software GNS3, que es un simulador gráfico de red el cual permite diseñar topologías de red complejas y poner en marcha simulaciones sobre ellas. [1]. Este software nos permite simular redes sin tener que gastar mucho dinero en hardware costoso para probarlo.

A diferencia de Cisco Packet Tracer, GNS3 es más un emulador que un simulador, esto quiere decir que para hacer uso de un router, cisco por ejemplo, se debe descargar la imagen del mismo y así contar con todas las opciones que brinda ese router.

GNS3 permite simular particionamiento VLSM, envíos broadcast y multicast, tal y como si se tratara de un red verdadera.

Objetivos

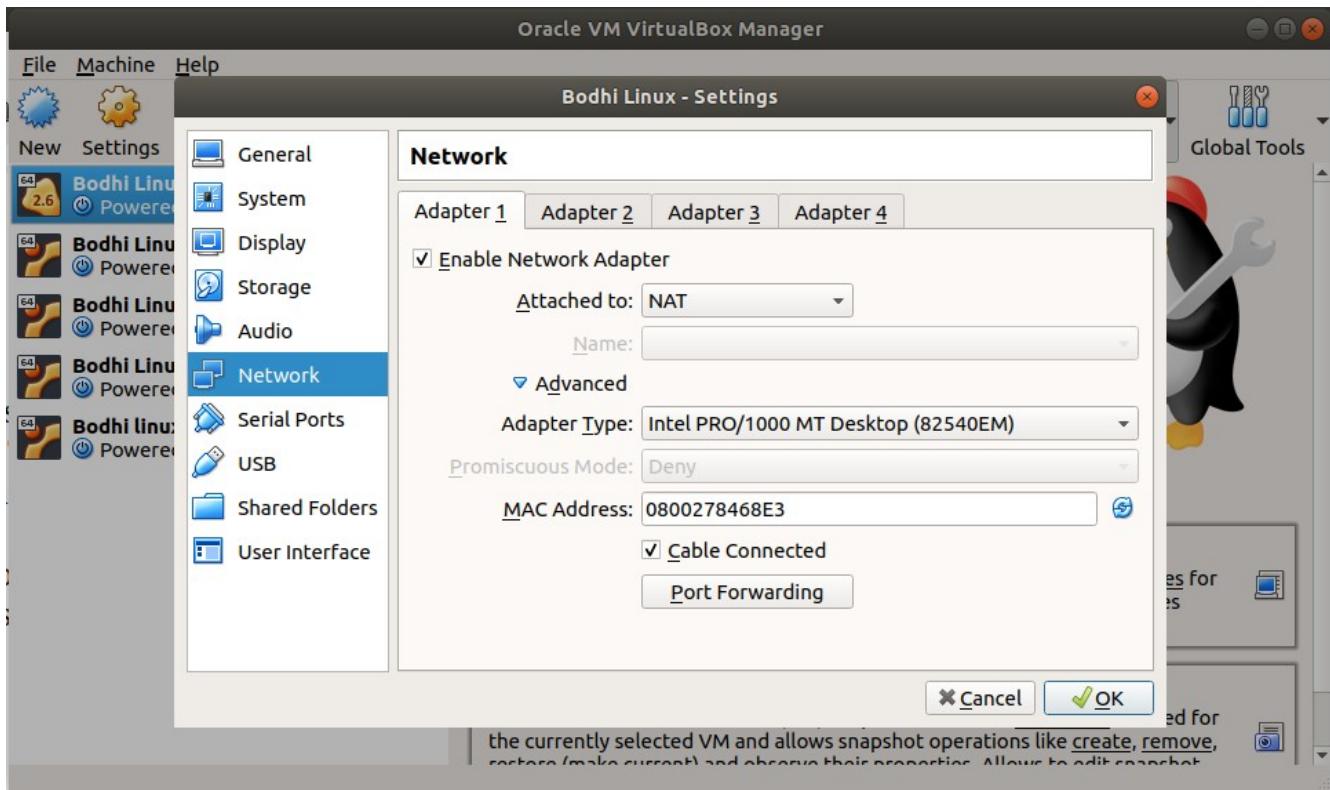
1. Elaborar la siguiente topología en GNS3, donde se conecten de 4 a 7 máquinas virtuales.



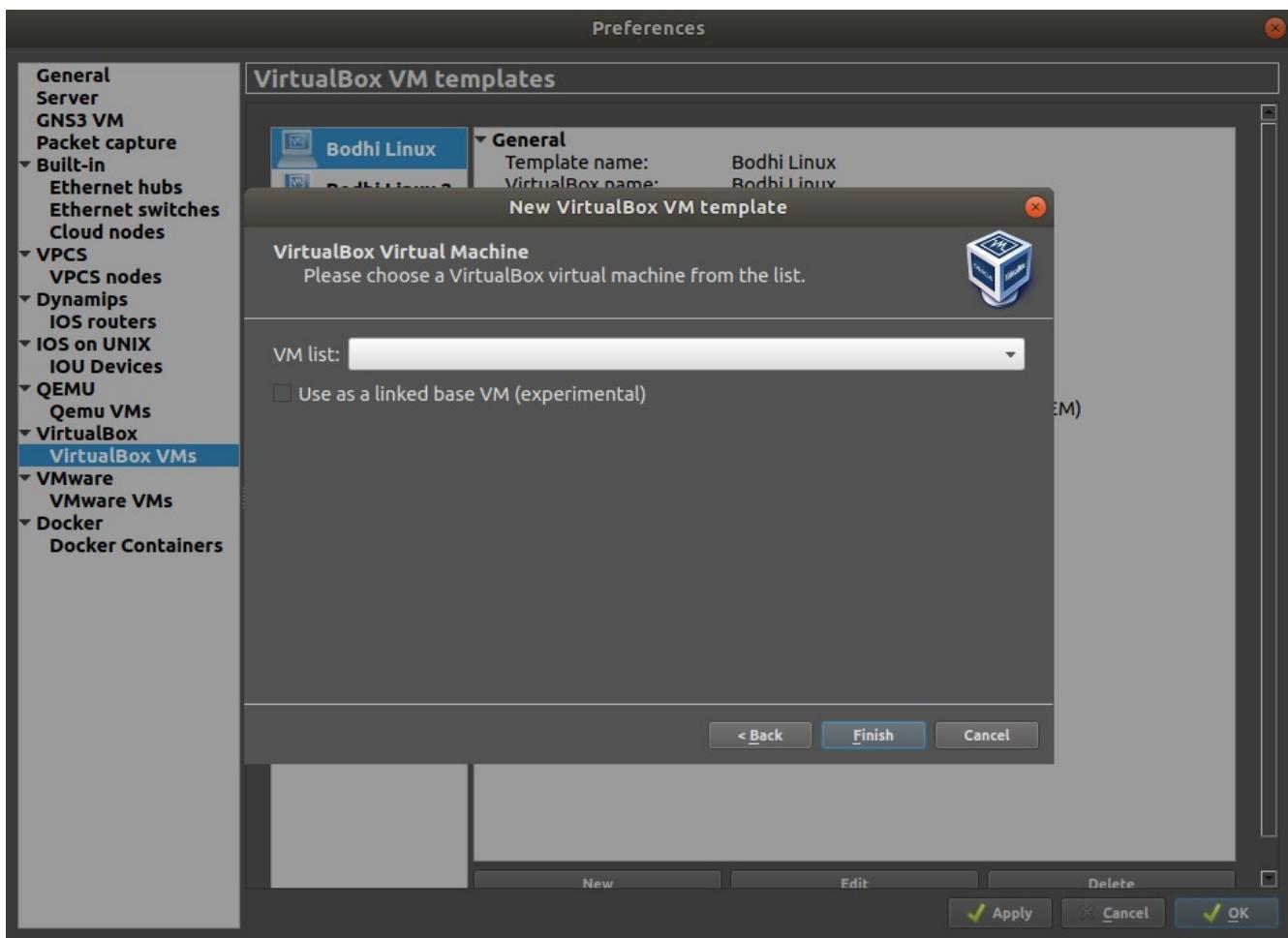
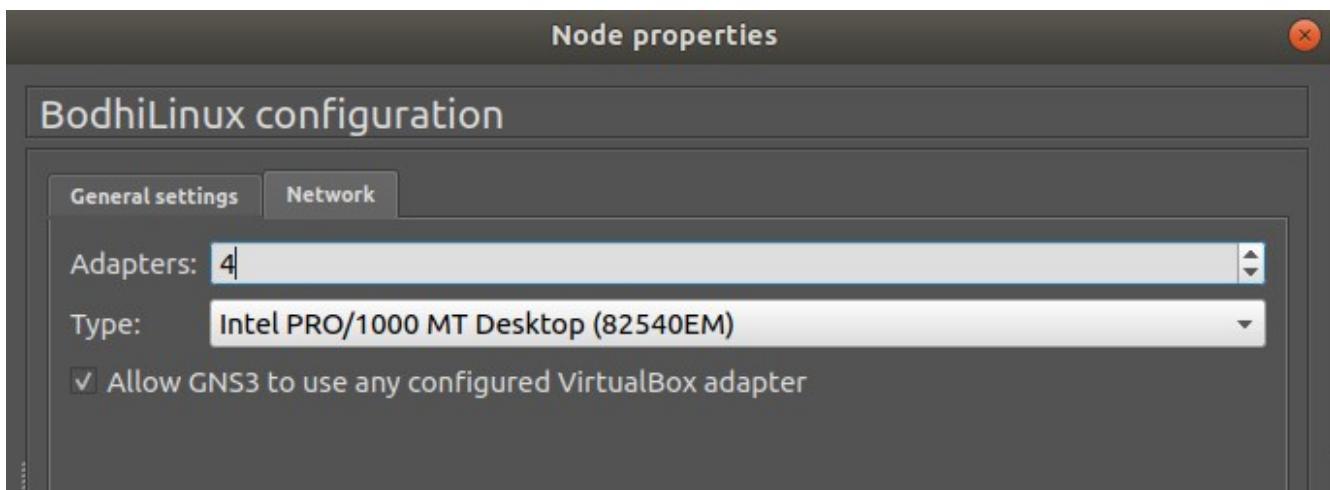
2. Conectar al anfitrión a través de una interfaz de ethernet virtual (la nube).
3. Enviar un mensaje desde un servidor, de manera broadcast y de manera multicast.

Desarrollo

1. Se crea la topología antes mencionada. Para ello se importa la imagen del router que se va a utilizar, en este caso se usó Cisco 3745. Se conectan de manera directa los “Ethernet Switch”, y las VPCs que son como pequeña maquinas virtuales con consumo mínimo de recursos y comandos muy limitados.
2. Se descargó el ISO de Bodhi Linux, el cual es una versión ligera de linux basado en ubuntu, con el fin de instalar las máquinas virtuales.
3. Una vez instalada la máquina, se configura su conexión a red desde virtual box para permitirle tener varias conexiones, y se especifica en las opciones avanzadas que al menos una de ellas va a ser conectada por cable.



4. Se agrega la máquina virtual al GNS3, en el menú de virtual box, se le da a la opción de “VirtualBox VMs”, para agregar la máquina virtual a la topología.
5. Se le da a la opción de “configure” en el ícono de la máquina virtual en el GNS3, y se cambia el número de adaptadores que permite esa máquina virtual de 1 a los que se prefieran. Se especifica el tipo, y se le da permiso a GNS3 de utilizar cualquier adaptador.



6. Abrimos la línea de comandos y creamos a la interfaz de ethernet virtual llamada "tap0", la levantamos, le asignamos una dirección ip y comprabamos que ya exista con los siguientes comandos.

```
sudo tunctl -t "nombre -u "usuario"
```

```
sudo ifconfig tap0 up
```

```
sudo ifconfig tap0 192.168.1.2
```

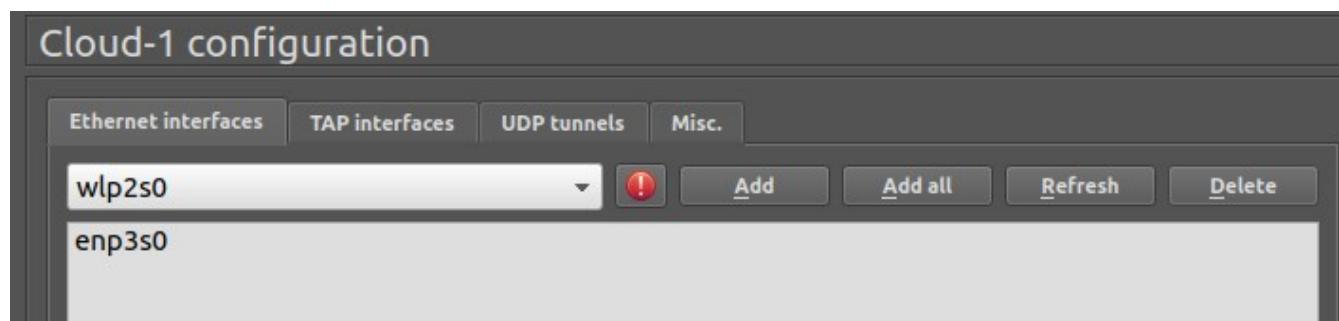
```
ifconfig
```

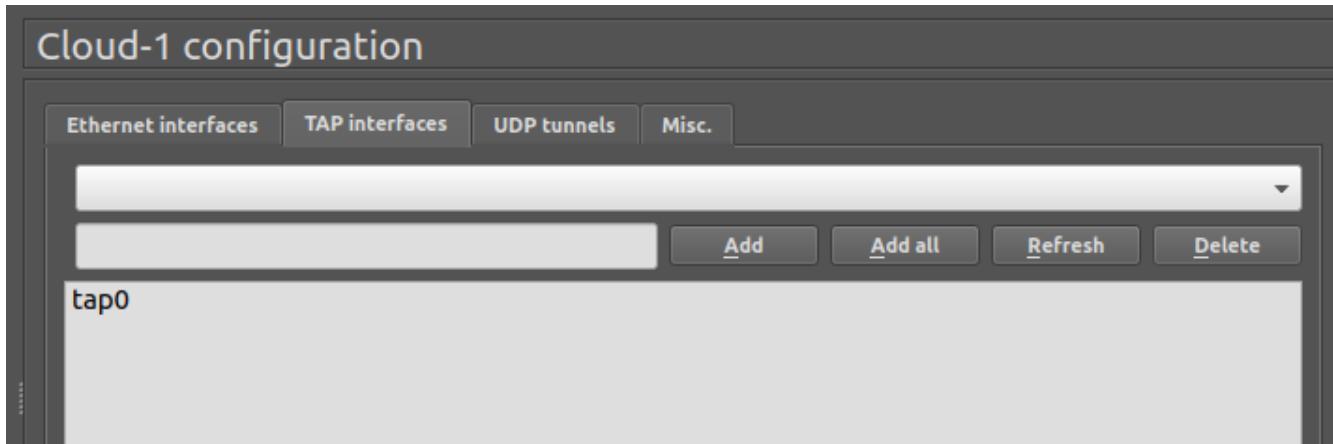
```
ana@ana-Inspiron-7460:~$ sudo ifconfig tap0 192.168.1.26
ana@ana-Inspiron-7460:~$ ifconfig
enp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether a4:4c:c8:0a:f2:e0 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      LibreOfficeWriter s 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
          RX packets 38461 bytes 16959162 (16.9 MB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 38461 bytes 16959162 (16.9 MB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.26 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::5062:27ff:fe0:6fd7 prefixlen 64 scopeid 0x20<link>
          ether 52:62:27:f0:6f:d7 txqueuelen 1000 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 28 bytes 3896 (3.8 KB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

7. Una vez creada la “tap0”, se configura la nube en el GNS3.





8. Se entra en la consola del router R1 y se configura la red y las particiones.

```
R1#enable
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int F
R1(config)#int FastEthernet 1/0
R1(config-if)#ip address 192.168.1.9 255.255.255.248
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#int F
*Mar 1 00:16:02.927: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Mar 1 00:16:03.927: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
R1(config)#int F
R1(config)#int FastEthernet 3/0
R1(config-if)#ip address 192.168.1.17 255.255.255.248
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#int F
*Mar 1 00:16:55.707: %LINK-3-UPDOWN: Interface FastEthernet3/0, changed state to up
*Mar 1 00:16:56.707: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/0, changed state to up
R1(config)#int F
R1(config)#int FastEthernet 4/0
R1(config-if)#ip address 192.168.1.25 255.255.255.248
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Mar 1 00:17:24.579: %LINK-3-UPDOWN: Interface FastEthernet4/0, changed state to up
*Mar 1 00:17:25.579: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet4/0, changed state to up
R1(config)#int F
R1(config)#int FastEthernet 2/0
R1(config-if)#ip address 10.0.0.1 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#
*Mar 1 00:17:58.563: %LINK-3-UPDOWN: Interface FastEthernet2/0, changed state to up
*Mar 1 00:17:59.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/0, changed state to up
```

9. Se configura la tabla de ruteo del router R1.

```
R1#enable
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#router rip
R1(config-router)#network 192.168.1.8
R1(config-router)#network 192.168.1.16
R1(config-router)#network 192.168.1.24
R1(config-router)#network 10.0.0.0
R1(config-router)#no auto-summary
R1(config-router)#exit
```

10. Se le asignan direcciones ip a las VPCS que concuerden con la subred donde se encuetran, más el gateway de esa subred.

PC-5 y PC-1 se encuentran dentro de la subred 192.168.1.8 y ya se pueden hacer ping entre ellas.

```
PC-5> ip 192.168.1.11 /29 192.168.1.9
Checking for duplicate address...
PC1 : 192.168.1.11 255.255.255.248 gateway 192.168.1.9
```

```
PC-1> ip 192.168.1.10 /29 192.168.1.9
Checking for duplicate address...
PC1 : 192.168.1.10 255.255.255.248 gateway 192.168.1.9
```

```
PC-1> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=0.763 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=0.596 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=0.372 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=0.374 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=0.670 ms
```

PC-7 se encuentra dentro de la subred 192.168.1.16, se le asigna la ip 192.168.1.20 con gateway 192.168.1.17.

```
PC-7> ip 192.168.1.20 /29 192.168.1.17
Checking for duplicate address...
PC1 : 192.168.1.20 255.255.255.248 gateway 192.168.1.17
```

Le hace ping al gateway, a la dirección 192.168.1.18 que es la PC-2 y está dentro de su subred y, le hace ping a otras PC en otras subredes.

```
PC-7> ping 192.168.1.10
192.168.1.10 icmp_seq=1 timeout
84 bytes from 192.168.1.10 icmp_seq=2 ttl=63 time=19.406 ms
84 bytes from 192.168.1.10 icmp_seq=3 ttl=63 time=19.477 ms
84 bytes from 192.168.1.10 icmp_seq=4 ttl=63 time=21.362 ms
84 bytes from 192.168.1.10 icmp_seq=5 ttl=63 time=20.514 ms
```

```
PC-7> ping 192.168.1.17
84 bytes from 192.168.1.17 icmp_seq=1 ttl=255 time=3.203 ms
84 bytes from 192.168.1.17 icmp_seq=2 ttl=255 time=8.743 ms
84 bytes from 192.168.1.17 icmp_seq=3 ttl=255 time=10.284 ms
84 bytes from 192.168.1.17 icmp_seq=4 ttl=255 time=8.021 ms
84 bytes from 192.168.1.17 icmp_seq=5 ttl=255 time=9.864 ms

PC-7> ping 192.168.1.18
84 bytes from 192.168.1.18 icmp_seq=1 ttl=64 time=0.949 ms
84 bytes from 192.168.1.18 icmp_seq=2 ttl=64 time=0.689 ms
84 bytes from 192.168.1.18 icmp_seq=3 ttl=64 time=0.792 ms
84 bytes from 192.168.1.18 icmp_seq=4 ttl=64 time=1.197 ms
84 bytes from 192.168.1.18 icmp_seq=5 ttl=64 time=0.815 ms
```

11. Se le asigna una dirección ip al anfitrión. Previamente la nube sabía que se iba a conectar a enp3s0

```
ana@ana-Inspiron-7460:~$ sudo ifconfig enp3s0 192.168.1.27
```

Se le hace ping al gateway, a la VPCS de prueba dentro de la misma subred, y a PC-1 que está en otra subred.

```
ana@ana-Inspiron-7460:~$ ping 192.168.1.25
PING 192.168.1.25 (192.168.1.25) 56(84) bytes of data.
64 bytes from 192.168.1.25: icmp_seq=1 ttl=255 time=25.0 ms
64 bytes from 192.168.1.25: icmp_seq=2 ttl=255 time=6.10 ms
64 bytes from 192.168.1.25: icmp_seq=3 ttl=255 time=4.61 ms
```

```
ana@ana-Inspiron-7460:~$ ping 192.168.1.30
PING 192.168.1.30 (192.168.1.30) 56(84) bytes of data.
64 bytes from 192.168.1.30: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 192.168.1.30: icmp_seq=2 ttl=64 time=0.666 ms
```

```
ana@ana-Inspiron-7460:~$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=63 time=3033 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=63 time=2011 ms
```

12. El Servidor va a estar en la máquina virtual Bodhi Linux 2. Se le da a start a la red y arrancan las máquinas virtuales desde el GNS3.

13. Se le asigna una ip a la máquina Bodhi Linux 2.

```
bodhi@bodhi-VirtualBox:~$ sudo ifconfig enp0s3 192.168.1.28 netmask 255.255.255.248 broadcast 192.168.1.31
[sudo] password for bodhi:
bodhi@bodhi-VirtualBox:~$ sudo route add default gw 192.168.1.25
```

Se le hace ping al gateway y a otras máquinas dentro del a subred. También se le hace ping a otras máquinas en otras subredes.

```
bodhi@bodhi-VirtualBox:~$ ping 192.168.1.25
PING 192.168.1.25 (192.168.1.25) 56(84) bytes of data.
64 bytes from 192.168.1.25: icmp_seq=1 ttl=255 time=23.1 ms
64 bytes from 192.168.1.25: icmp_seq=2 ttl=255 time=10.9 ms
64 bytes from 192.168.1.25: icmp_seq=3 ttl=255 time=9.66 ms
64 bytes from 192.168.1.25: icmp_seq=4 ttl=255 time=2.68 ms
64 bytes from 192.168.1.25: icmp_seq=5 ttl=255 time=13.9 ms
^C
--- 192.168.1.25 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4020ms
rtt min/avg/max/mdev = 2.680/12.085/23.128/6.652 ms
bodhi@bodhi-VirtualBox:~$ ping 192.168.1.26
PING 192.168.1.26 (192.168.1.26) 56(84) bytes of data.
64 bytes from 192.168.1.26: icmp_seq=1 ttl=64 time=1.79 ms
64 bytes from 192.168.1.26: icmp_seq=2 ttl=64 time=0.776 ms
64 bytes from 192.168.1.26: icmp_seq=3 ttl=64 time=1.46 ms
64 bytes from 192.168.1.26: icmp_seq=4 ttl=64 time=0.239 ms
```

14. Se le asigna una ip a la máquina Bodhi Linux 4 quien será uno de los clientes, y se le hace pines a otras máquinas para ver si funciona.

15. Ahora, se hacen los programas de cliente y servidor respectivamente.

```
from socket import *
s=socket(AF_INET, SOCK_DGRAM)
s.bind(('',12345))
while True:
    m=s.recvfrom(1024)
    print (m[0])
```

```
from socket import *
from time import *
s = socket(AF_INET, SOCK_DGRAM)
s.setsockopt(SOL_SOCKET, SO_BROADCAST, 1)
while True:
    sleep(1)
    s.sendto('HOLI', ('192.168.1.31', 12345))
    #s.sendto('HOLI', ('192.168.2.23', 12345))
    #print 'Broadcast sent!'
    #s.sendto('HOLI', ('192.168.2.19', 12345))
    #s.sendto('HOLI', ('192.168.2.20', 12345))
    #s.sendto('HOLI', ('192.168.2.255', 12345))
```

16. Bodhi Linux 2 va a correr el servidor que le va a enviar mensajes de manera broadcast a los clientes dentro de su subred, mientras que Bodhi Linux 4 y el anfitrión van a recibir los mensajes

```
bodhi@bodhi-VirtualBox:~$ python ping.py
cosa enviada
cosa enviada
cosa enviada
cosa enviada
cosa enviada
cosa enviada
```

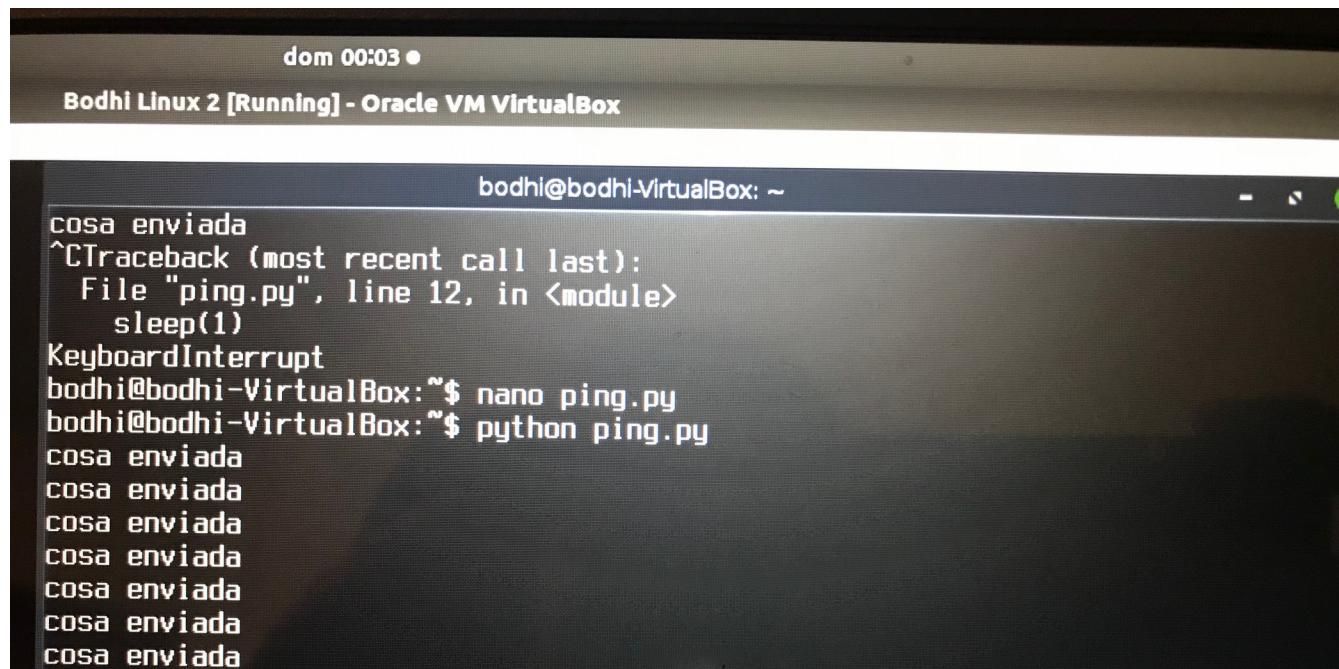
```
bodhi@bodhi-VirtualBox:~$ python2 client.py
HOLI
```

17. Se repiten los pasos del router R1 pero ahora en la red del router R2 (192.168.2.0)

```
10.0.0.0/30 is subnetted, 1 subnets
    10.0.0.0 is directly connected, FastEthernet1/0
192.168.1.0/24 [120/1] via 10.0.0.1, 00:00:25, FastEthernet1/0
192.168.2.0/29 is subnetted, 3 subnets
    192.168.2.8 is directly connected, FastEthernet2/0
    192.168.2.24 is directly connected, FastEthernet3/0
    192.168.2.16 is directly connected, FastEthernet4/0
```

18. Se le asignan direcciones ip a las VPCS y a las máquinas virtuales de la red del router R2.

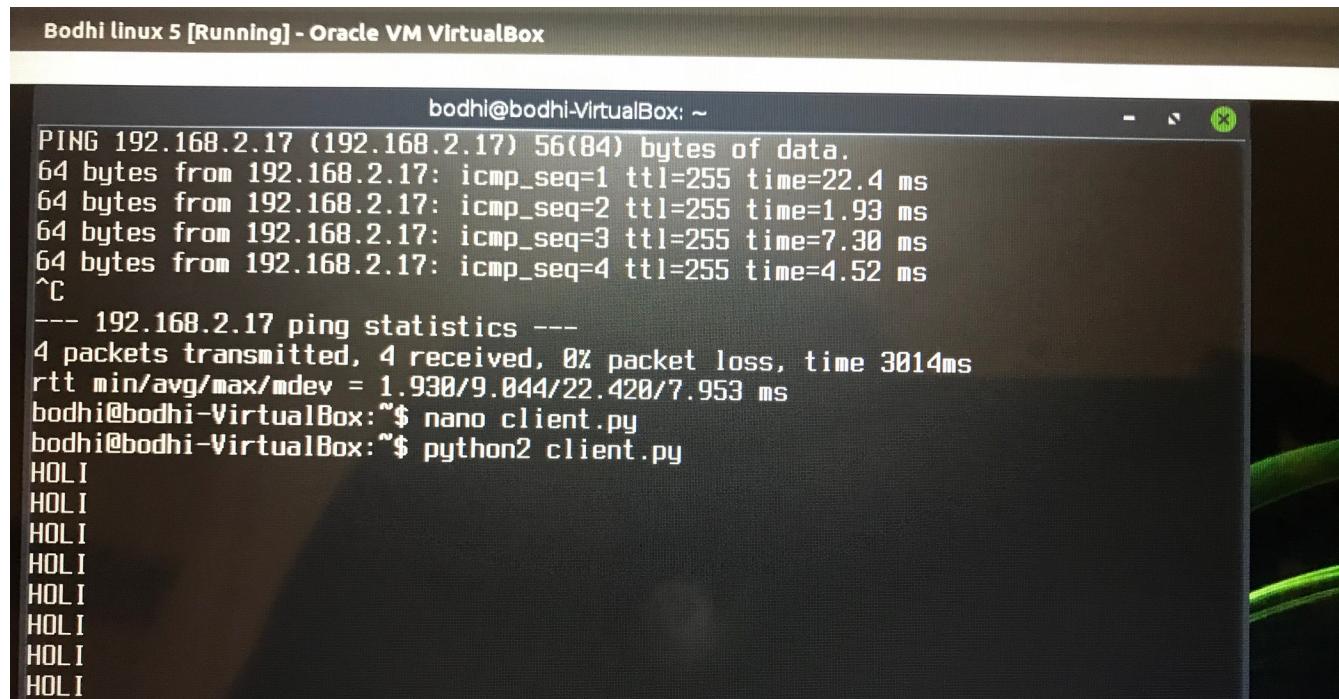
19. Se modifica el programa del servidor para que ahora le envíe el mensaje a “Bodhi Linux 3” y “Bodhi Linux 5”.



```
dom 00:03 •
Bodhi Linux 2 [Running] - Oracle VM VirtualBox: ~

cosa enviada
^CTraceback (most recent call last):
  File "ping.py", line 12, in <module>
    sleep(1)
KeyboardInterrupt
bodhi@bodhi-VirtualBox:~$ nano ping.py
bodhi@bodhi-VirtualBox:~$ python ping.py
cosa enviada
```

“Bodhi Linux 2” de dirección 192.168.1.28, le envía el mensaje a las direcciones 192.168.2.18 y 192.168.2.20.



```
Bodhi linux 5 [Running] - Oracle VM VirtualBox: ~

bodhi@bodhi-VirtualBox: ~

PING 192.168.2.17 (192.168.2.17) 56(84) bytes of data.
64 bytes from 192.168.2.17: icmp_seq=1 ttl=255 time=22.4 ms
64 bytes from 192.168.2.17: icmp_seq=2 ttl=255 time=1.93 ms
64 bytes from 192.168.2.17: icmp_seq=3 ttl=255 time=7.30 ms
64 bytes from 192.168.2.17: icmp_seq=4 ttl=255 time=4.52 ms
^C
--- 192.168.2.17 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 1.930/9.044/22.420/7.953 ms
bodhi@bodhi-VirtualBox:~$ nano client.py
bodhi@bodhi-VirtualBox:~$ python2 client.py
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
```

Bodhi Linux 3 [Running] - Oracle VM VirtualBox

```
bodhi@bodhi-VirtualBox: ~
SIOCADDRT: Network is unreachable
bodhi@bodhi-VirtualBox:~$ sudo ifconfig enp0s3 192.168.2.18 netmask
255.255.255.0 broadcast 192.168.2.23
bodhi@bodhi-VirtualBox:~$ sudo route add default gw 192.168.2.17
bodhi@bodhi-VirtualBox:~$ python2 client.py
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
HOLI
```

Conclusión

El VLSM (variable length subnet mask) o Máscaras de Subred de Longitud Variable, es una solución de Subnetting que favorece el aprovechamiento de direcciones ip al máximo. Sólo puede utilizarse en protocolos de enrutamiento como en el caso de la práctica que se utilizó RIP.

Referencias

- [1] <https://es.wikipedia.org/wiki/GNS3>

sudo netplan apply

if something goes wrong

④.4

sudo netplan --debug apply

Ana Paola Naya Vivas

Broadband Multicost

~~Pendiente~~ pendiente de
revision