



Instituto Politécnico Nacional
Escuela Superior de Cómputo



ALUMNO:MERCADO ROGEL MARTÍN ISAURO

BOLETA:2014090449

UNIDAD DE APRENDIZAJE: APLICACIONES PARA LA COMUNICACIÓN
EN RED

GRUPO:
CURSO DE RECUPERACIÓN ACADÉMICA

Práctica 1:

**Transmisión de un mensaje a dispositivos en una red
(broadcast)**

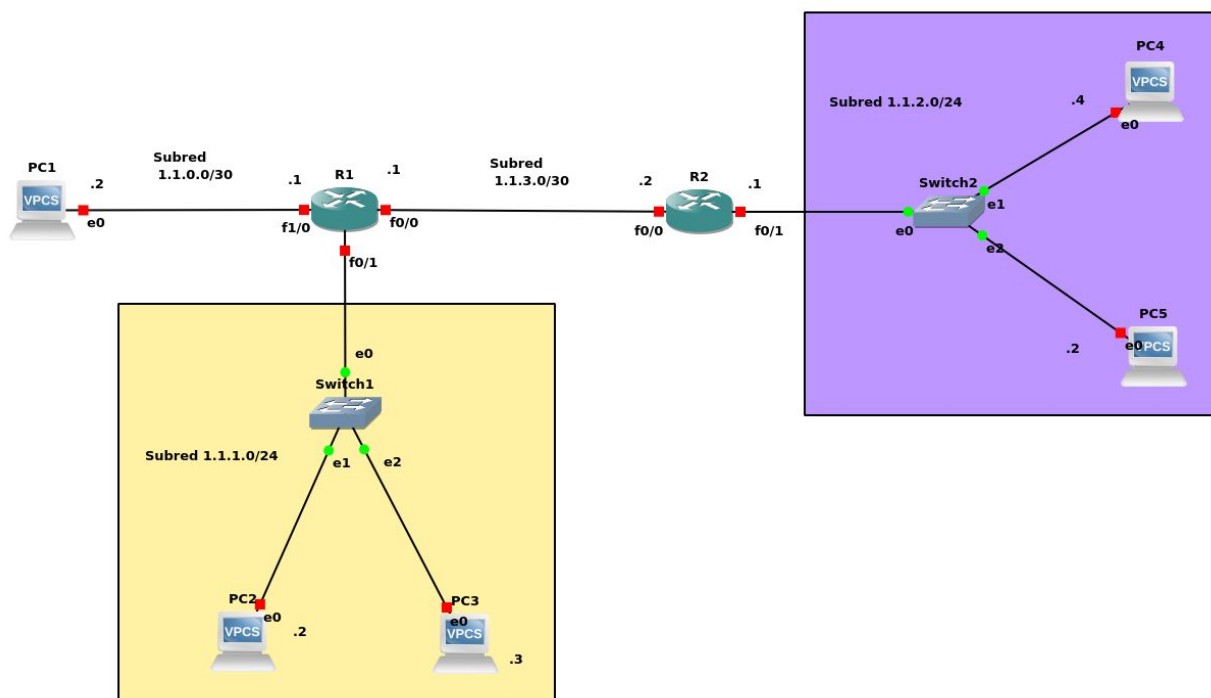
Introducción

En el presente documento se mostrará un ejemplo de cómo se realiza la transmisión de un mensaje dentro de una red para los dispositivos conectados a ella.

Esto es, se llevará a cabo la creación de un programa broadcast, donde el servidor estará transmitiendo un mensaje que el cliente recibe.

Configuración de los routers

Tenemos la siguiente topología:



Primero configuremos las interfaces de los router. Para saber lo que está cargado en el RAM, es decir, la configuración actual utilicemos **show running-configuration**, haciéndolo puedo ver lo siguiente:

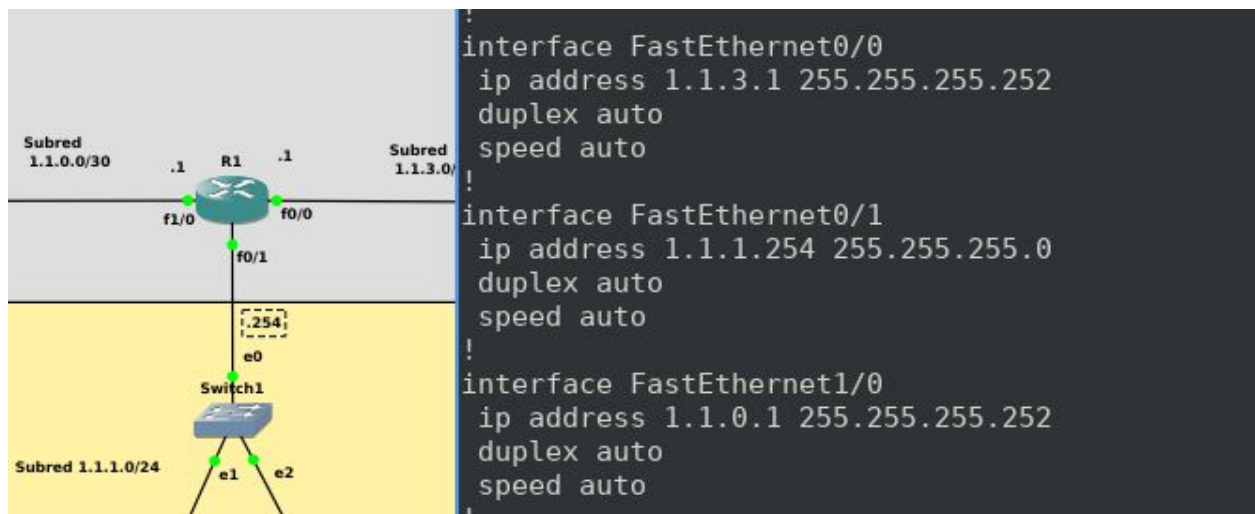


```
R1
File Edit View Search Terminal Help
!
interface FastEthernet0/0
ip address 1.1.0.1 255.255.255.252
duplex auto
speed auto
!
interface FastEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet1/0
no ip address
shutdown
duplex auto
speed auto
!
interface FastEthernet1/1
no ip address
shutdown
duplex auto
speed auto
```

Lo que puedo ver es que sólo una de las 4 interfaces disponibles está configurada (de manera errónea), así que hagamos la configuración.

De manera predeterminada el router inició en modo administrativo (la terminal muestra **Router#**), así que debo cambiar a modo configuración (se debe mostrar **Router(config)#**) utilizando el comando **configure terminal**. Después debo indicarle que quiero configurar una interfaz en específico, esto se hace con **configure interface FastEthernetX/X** también podemos utilizar **int fX/X** lo cual produce **Router(config-if)#** por ejemplo **int f0/0**. Ahora introduzco el comando **ip address dirección máscara**, por ejemplo **ip 1.1.3.1 255.255.255.252**. Ahora activaremos la interface con **no shutdown**. Una vez hecho esto con **exit** regresamos un nivel a la configuración del router y repetimos el proceso para cada interfaz.

El anterior procedimiento debería cambiar la configuración del router como se muestra:

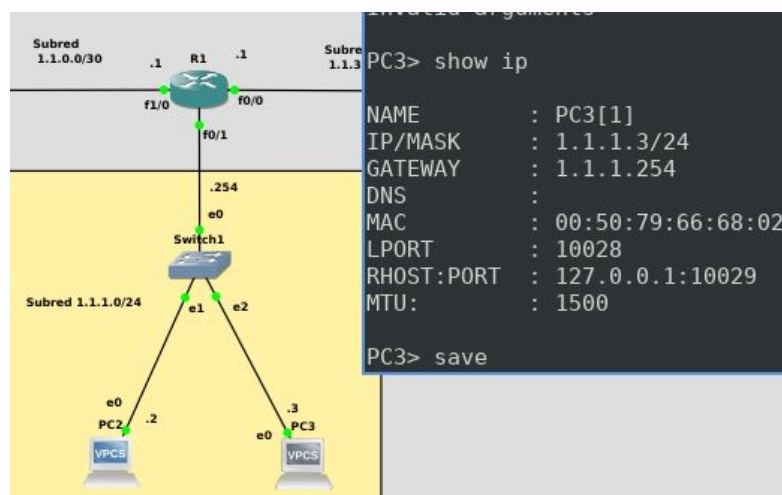


Para guardar la configuración y no se elimine cuando reseteamos el dispositivo podemos usar **write memory**. También se pueden usar los comandos **copy running-config startup-config** y **save**

El procedimiento lo repetimos para los otros dispositivos conectados a el router y para los otros routers.

Configuración de las pc virtuales

Primero configuremos las que tienen por defecto el sistema operativo que proporciona GNS3. Para asignarle la dirección a una de estas utilizemos el comando **ip dirección máscara gateway**. Por ejemplo sea ip 1.1.1.2 255.255.255.0 1.1.1.254 (también se puede ingresar ip 1.1.1.2/24 1.1.1.254).



Como se muestra en la foto anterior para consultar nuestra ip usamos **show ip**. Para que nuestra configuración no se borre al reiniciar la máquina debemos usar **save**.

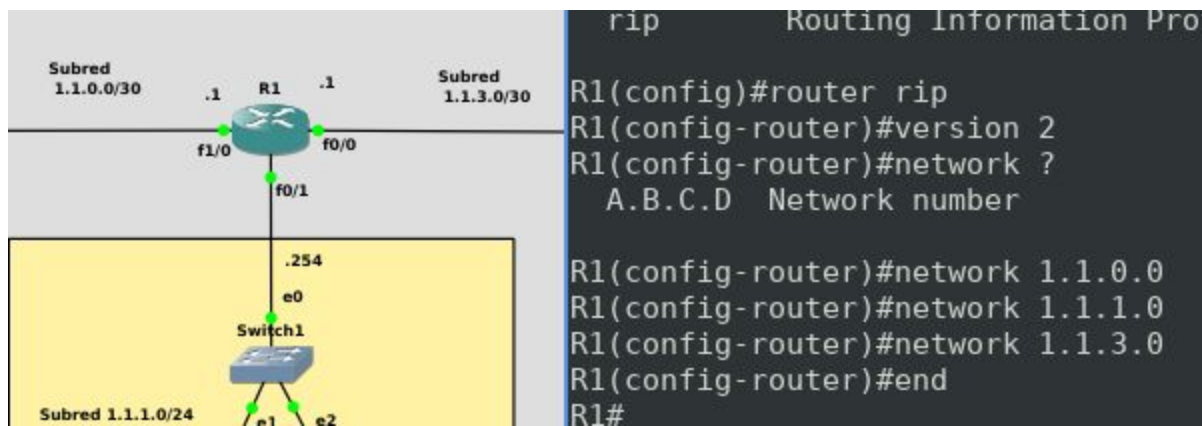
Para probar que hemos hecho la correcta asignación podemos usar **ping dirección**. Por ejemplo de la máquina PC2 a PC3 en la subred 1.1.1.0:

```
PC2> ping 1.1.1.3
84 bytes from 1.1.1.3 icmp_seq=1 ttl=64 time=0.269 ms
84 bytes from 1.1.1.3 icmp_seq=2 ttl=64 time=0.272 ms
84 bytes from 1.1.1.3 icmp_seq=3 ttl=64 time=0.259 ms
84 bytes from 1.1.1.3 icmp_seq=4 ttl=64 time=0.262 ms
84 bytes from 1.1.1.3 icmp_seq=5 ttl=64 time=0.279 ms
^@
PC2>
```

Configuración de la tabla de rutas

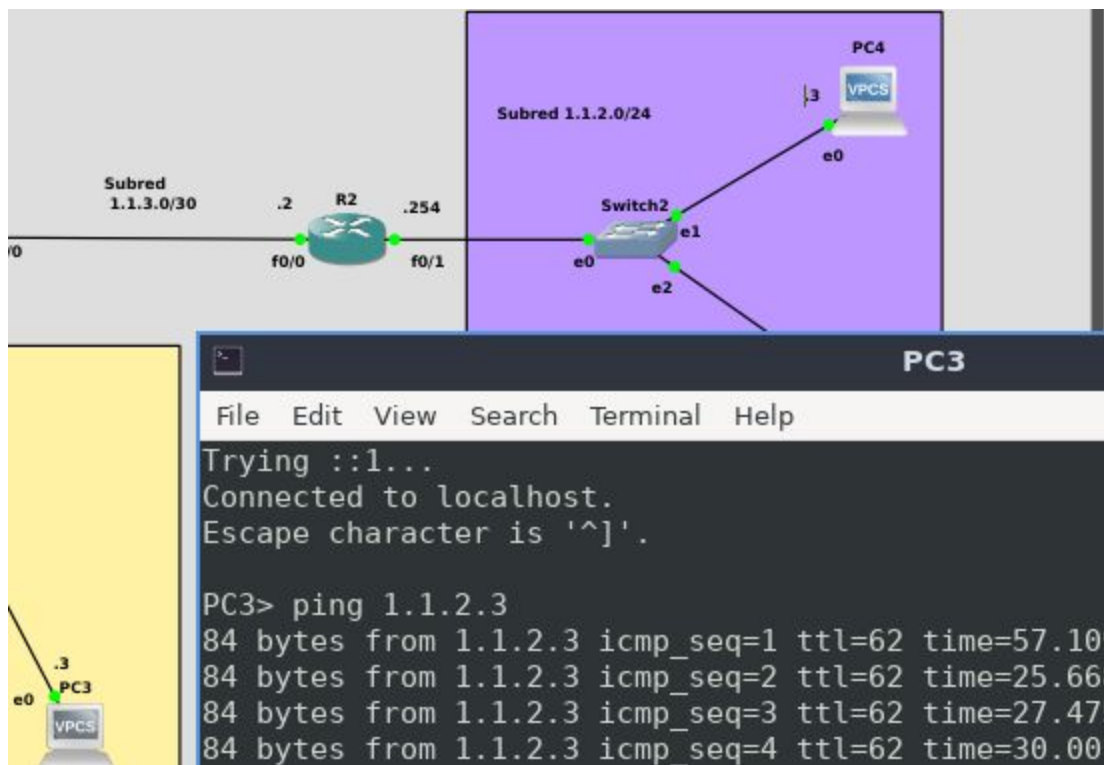
Usemos el protocolo RIP (**Routing Information Protocol**) para que los dispositivos conectados a nuestro router se puedan comunicar con los de otra red.

Para ello entremos en modo configuración y ejecutemos el comando **router rip**. Después le indicamos la versión del protocolo con **version 2** y debemos indicarle las redes que tiene conectadas directamente:

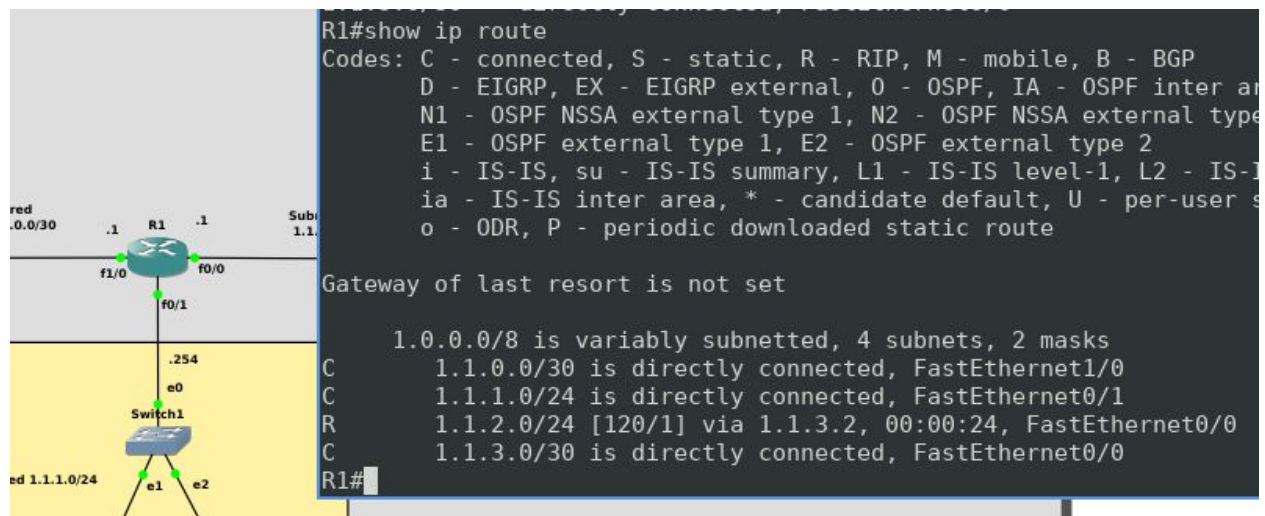


Para verificar que estén guardados podemos usar **show ip rip database**.

Una vez hecho esto podemos hacer un ping de una red a otra y debe ser exitoso:



Revisemos la tabla de ruteo con **show ip route**:



Podemos ver que nuestro router encontró la red 1.1.2.0 mediante la red 1.1.3.0.

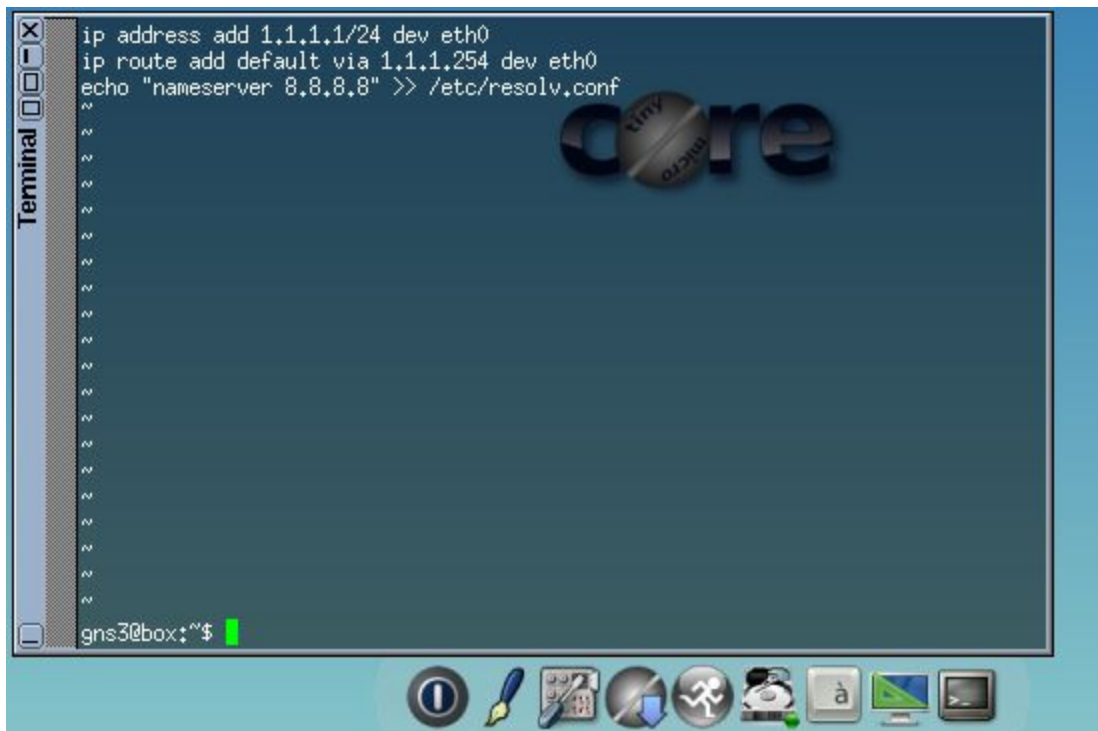
Ahora activemos la configuración para que el router pueda recibir broadcast directo.

Entremos en la configuración del router y luego de la interface que tiene conectada la red a la que queremos hacer broadcast.

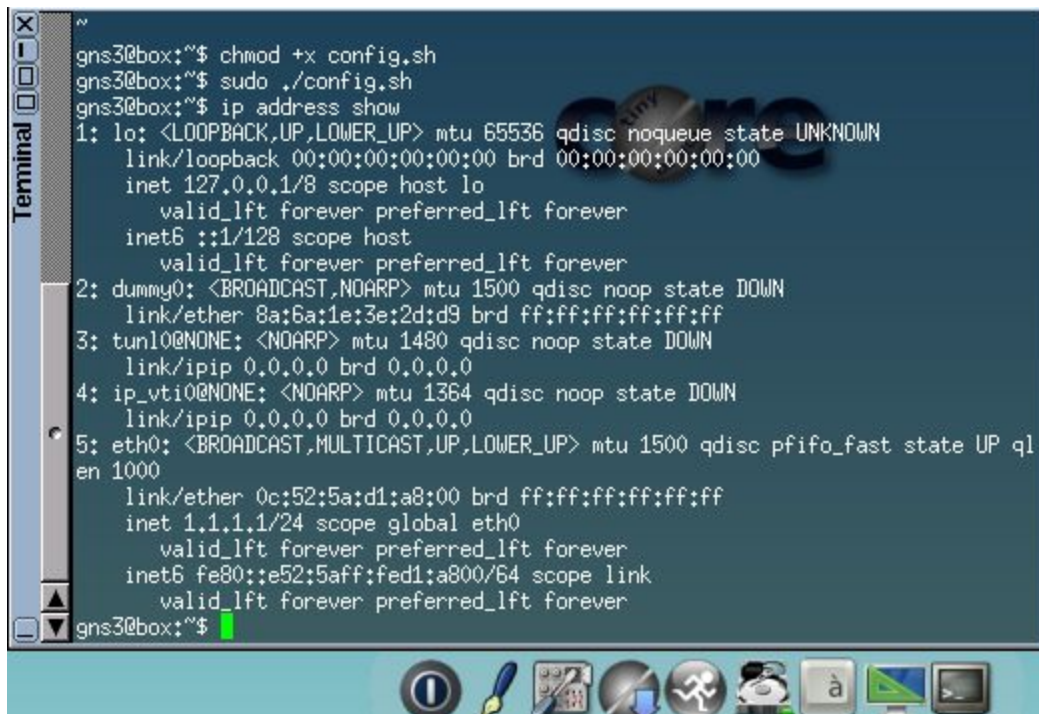
```
show ip interface f0/1 | include broadcast
```

Configuración de las máquinas virtuales

Ya que tengamos instaladas nuestras máquinas virtuales debemos configurarlas para que podamos conectarnos a la red y a internet en mi caso para poder instalar gcc y desarrollar el programa de broadcast. En mi caso estoy usando una imagen de linux muy ligera, llamada **TinyCore** ¡Sólo pesa 16 MB! A continuación muestro lo que debo hacer para conectarme:



Ejecuto un script donde tengo esas órdenes, la primera pone la ip de mi máquina virtual y la asigna a la interfaz eth0. Luego agrego el gateway que será 1.1.1.254 igual a la interfaz eth0. La tercera línea es para agregar un DNS y así me pueda conectar a páginas sin saber su ip. Ejecuto mis comando y puedo ver que he logrado realizar la configuración:



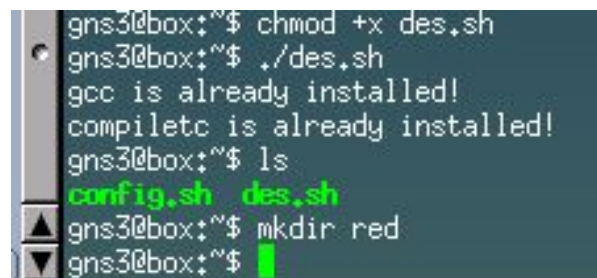
```
gns3@box:~$ chmod +x config.sh
gns3@box:~$ sudo ./config.sh
gns3@box:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: dummy0: <BROADCAST,NOARP> mtu 1500 qdisc noop state DOWN
    link/ether 8a:6a:1e:3e:2d:d9 brd ff:ff:ff:ff:ff:ff
3: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN
    link/ipip 0.0.0.0 brd 0.0.0.0
4: ip_vti0@NONE: <NOARP> mtu 1364 qdisc noop state DOWN
    link/ipip 0.0.0.0 brd 0.0.0.0
5: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
    en 1000
    link/ether 0c:52:5a:d1:a8:00 brd ff:ff:ff:ff:ff:ff
    inet 1.1.1.1/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::e52:5aff:fed1:a800/64 scope link
        valid_lft forever preferred_lft forever
gns3@box:~$
```

Ahora debo descargar gcc ya que por defecto no lo tengo instalado en esta distribución, necesito hacer uso de su administrador de paquetes:



```
tce-load -wi gcc.tcz
tce-load -wi compiletc
```

Al parecer ya lo tenía instalado:



```
gns3@box:~$ chmod +x des.sh
gns3@box:~$ ./des.sh
gcc is already installed!
compiletc is already installed!
gns3@box:~$ ls
config.sh  des.sh
gns3@box:~$ mkdir red
gns3@box:~$
```

Ahora sólo resta configurar las restantes máquinas de la misma forma.

Creación del programa de broadcast

Basándome en lo que hice en mi tarea 2 sólo debo hacer unos pequeños cambios al código fuente. Básicamente sólo se trata de agregar ciclos infinitos, tanto en cliente y servidor:

```
int permiso = 1;
int status = setsockopt(socket_fd, SOL_SOCKET, SO_BROADCAST, (void *)&permiso, sizeof(int));
if(status == -1)
{
    perror("Error al establecer permisos \n");
    return -1;
}

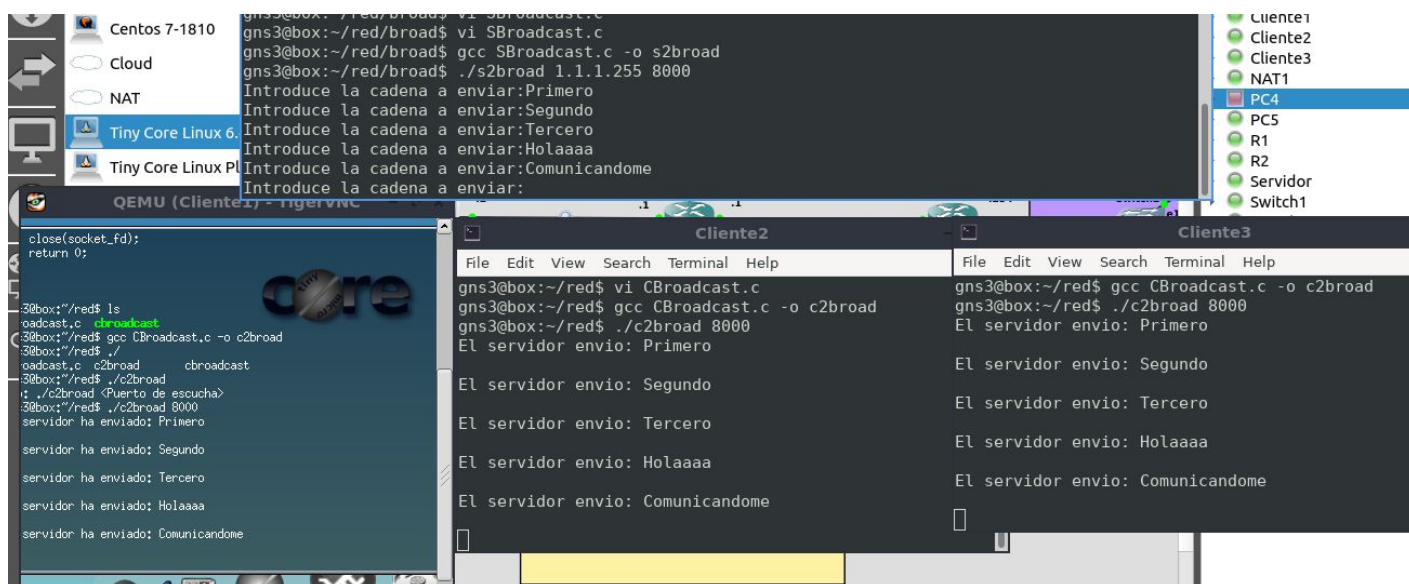
/* Recibir datagramas del servidor */
struct sockaddr_in dir_serv;
socklen_t tam_serv = sizeof(dir_serv);
int tam_recv;
while(1)
{
    int tam_recv = recvfrom(socket_fd, cad_recv, MAX_RECV, 0, (struct sockaddr *)&dir_serv, &tam_serv);
    cad_recv[tam_recv] = '\0';
    printf("El servidor ha enviado: %s\n", cad_recv);
}
close(socket_fd);
return 0;
```

CBroadcast.c 66/67 98%

```
for(;;)
{
    printf("Introduce la cadena a enviar:");
    fgets(buffer, MAX_BUF, stdin);
    buflen = strlen(buffer); /* Tamaño de la cadena a enviar */
    status = sendto(sockfd, buffer, buflen, 0, (struct sockaddr *)&addr_broadcast, sizeof(addr_broadcast));
    if(buflen != status)
    {
        perror("Hubo un error al enviar los datos...\n");
        return -1;
    }
}
close(sockfd);
return 0;
```

SBroadcast.c [Modified] 73/73 100%

Y ahora debo de probar de nuevo los programas. Cabe resaltar que sólo en una consola tengo una conexión con vnc y las demás con telnet:



Conclusiones

La realización de esta práctica fue más rápida teniendo ya la tarea 2, no obstante este documento es largo ya que quería crear una especie de manual sobre como utilizar gns3 y como podríamos comunicarnos con varias máquinas virtuales. Esta práctica fue muy provechosa, ya que yo no tenía nada de conocimiento acerca de gns3 y con el desarrollo, configuración de routers, conexión a internet y creación de máquinas virtuales aprendí mucho. No solamente de gns3 sino también acerca de sistemas operativos basados en GNU/Linux y sobre redes.

Me llevo de esta y las otras prácticas muchos aprendizajes que no tenía antes de su realización.