Enrutamiento dinámico (CISCO-Linux), switches y VLANs

Carlos Daniel de la Cruz García

Abstract-

Index Terms - VLAN, Enlace Troncal, Paquete 802.1Q, Encapsulation dot1q, Máquina Virtual, Loopback.

I. Introducción

En este reporte se enseña la creación de VLANs en los switches y la comunicación entre estas mediante enrutamiento y conexiones troncales, además, se usa una computadora con el sistema operativo Linux, la cual cumple la función de un router.

En el desarrollo de este reporte se muestra todos los procesos usados, los pasos seguidos para la configuración exitosa de todo lo requerido. En una configuración con VLANs dentro de un switch, la comunicación entre VLANs no es posible, al menos de que se agregue un router que pueda crear conexión entre estos y se configure las interfaces de salida como enlaces troncales. Los paquetes recibidos por el router que está conectado con el switch, son paquetes tipo 802.1Q, que incluyen la VLAN remitente, este paquete puede ser leído y decodificado por el router con la encapsulación dot1q, dentro de la subinterfaz para cada VLAN. Gracias a esta configuración en el router, y correctamente configuradas las VLANs, la respuesta será exitosa cuando se haga ping a alguna de las direcciones dentro de la VLAN.

La configuración de esta práctica consiste en 3 routers cisco, una computadora virtual Linux, la cual se configura para que pueda actuar como router, 4 switches, VLANs, computadoras en cada VLAN y en los otros routers, se configura con interfaz loopback.

II. Objetivos

Crear una conexión exitosa entre las VLANs de los switches con el resto de la red, incluido a esto, la creación y modificación correcta de la máquina virtual con sistema operativo linux, para que se comporte como router, que esta pueda hacer el enrutamiento necesario para que toda la red pueda hacer ping.

III. Marco Teórico

VLAN

Una VLAN (Virtual Local Area Network) es cualquier dominio de difusión que está particionado y aislado de una red de computadoras en la capa de enlace datos (capa OSI 2).En este contexto, virtual se refiere a un objeto físico recreado y alterado por lógica adicional, dentro de la red de área local. Las VLAN funcionan aplicando etiquetas a tramas de red y manejando estas etiquetas en sistemas de redes, creando la apariencia y funcionalidad del tráfico de red que está físicamente en una sola red pero actúa como si estuviera dividido entre redes separadas. Esto permite a los

administradores de red agrupar hosts incluso si no están conectados directamente al mismo switch de red (Harmoush, 2022).

Enlace Troncal

Un enlace troncal es una conexión de red que lleva el tráfico de múltiples VLAN a través de una red. Permite a las VLAN comunicarse entre sí a través de diferentes dispositivos de red, como los switches. (Harmoush, 2022).

Paquete 802.1Q

Un paquete 802.1Q es un paquete de red que ha sido etiquetado con información de VLAN adicional según el estándar IEEE 802.1Q.1 Esta etiqueta identifica a qué VLAN pertenece el paquete y permite que el tráfico de VLAN se transmita a través de enlaces troncales (Harmoush, 2022).

Encapsulation dot1q

La encapsulación dot1q es un método para etiquetar tramas de red con información de VLAN según el estándar IEEE 802.1Q. Esta encapsulación permite que las tramas de red sean transmitidas a través de enlaces troncales y separadas en VLAN en diferentes dispositivos de red (Harmoush, 2022).

Máquina Virtual

Una máquina virtual es un software que emula a una computadora y puede ejecutar programas como si fuera una computadora real. Las máquinas virtuales permiten la consolidación de servidores, la creación de entornos de prueba y desarrollo, y la ejecución de múltiples sistemas operativos en un solo hardware físico (VMware, s.f.).

Loopback

La interfaz loopback es una interfaz lógica interna del router. Esta no es asignada a un puerto físico, se crea como un puerto para pruebas, lo que hace que no se pueda conectar algún dispositivo a esta interfaz. Es considerada como interfaz de software que se coloca de manera automática en estado UP (activo), siempre que el router funcione correctamente. (sapalomera, s.f.)

IV. Desarrollo

Diagrama

El diagrama físico consta de 4 routers totales, dos de esos routers tienen una interfaz loopback para pruebas, los otros dos tienen un switch conectado y de ese switch otro conectado a ese. Se incluyen 4 VLANs totales, 1 Vlan x Switch.

La forma de la topología es una topología de línea, (Ver Anexo 2). Donde se puede ver como en los finales de la topología, hasta arriba un router con dos switches y dos VLANs (representadas con colores), como se comenta anteriormente, hasta abajo el router-linux igual con dos switches y sus VLANs, y en los routers centrales dos direcciones loopback.

Equipo

- 3 enrutadores Cisco
- 1 enrutador en Linux (máquina virtual)
- 4 switches

IV.I Desarrollo Práctico

Configuración de enrutador Linux

1. Mientras la máquina se encuentra apagada, se enlazaran los puertos físicos a los adaptadores de la máquina virtual a través del Virtual Network Editor.
2. Encender la máquina virtual
3. Acceder a la terminal e ingresar los siguientes comandos para la utilización de diferentes herramientas:
    a. *sudo apt install frr*
    b. *sudo apt install net-tools*
4. Se cambiará la preferencia de los puertos desde configuración para dejar la conexión NAT y continuar con el enlace personalizado.
5. Reingresar a la terminal, acceder como usuario privilegiado con el comando:
    a. *sudo -i*
6. Inmediatamente después, se activarán los protocolos a usar:
    a. nano /etc/frr/daemons
7. Guardar el buffer de los protocolos y activar el reenvío de paquetes en ipv4 e ipv6:
    a. *nano /etc/sysctl.conf*
    b. *net.ipv4.ip_forwarding=1*
    c. *net.ipv6.conf.all.forwarding=1*
8. Nuevamente guardar el buffer y empezar la configuración de puertos.
9. Identificar los puertos:
    a. *ip link*

Para fines de esta práctica se usarán como referencia los puertos *ens33* y *ens37* ya que son los puertos del dispositivo.

10. Al puerto *ens33* se le asignan direcciones en ipv4 e ipv6
    a. *ip link addr add 4.0.0.10/30 dev ens33*
    b. *ip -6 add 2001:3::2/64 dev ens 33*
    c. *ip link set dev ens 33 up*

El puerto *ens37* solo tendrá las subinterfaces respectiva a las VLANs configuradas.

11. Prender el puerto *ens37*
    a. *ip link set dev ens37 up*
12. Creación de las VLANs y subinterfaces:
    a. *ip link add link ens37 name ens37.10 type vlan id 10*
    b. *ip link add link ens37 name ens37.20 type vlan id 20*
13. Añadir direcciones tanto en ipv4 como en ipv6 a las subinterfaces:
    a. *ip addr add 1.0.0.17/29 dev ens37.10*
    b. *ip addr add 1.0.0.25/29 dev ens37.20*
    c. *ip -6 addr add 2001:F::1/64 dev ens37.10*
    d. *ip -6 addr add 2001:E::2/64 dev ens37.20*

Finaliza la configuración de ip en Linux. Pasar a la configuración de ruteo.

14. Entrar al modo router:
    a. *vtysh*
15. Añadir las configuraciones para puertos:
    a. *ens33*

```
cdelax# conf t
cdelax(config)# interface ens33
cdelax(config-if)# ip addr 4.0.0.10/30
cdelax(config-if)# ipv6 addr 2001:3::2/64
```

b. ens37 (.10, .20)

```
cdelax(config)# interface ens37.10
cdelax(config-if)# ip addr 1.0.0.17/29
cdelax(config-if)# ipv6 addr 2001:F::1/64

cdelax(config)# interface ens37.20
cdelax(config-if)# ip addr 1.0.0.25/29
cdelax(config-if)# ipv6 addr 2001:E::1/64
```

16. Configuración en RIPng

```
cdelax(config)# router ripng
cdelax(config-router)# network 2001:3::/64
cdelax(config-router)# network 2001:F::/64
cdelax(config-router)# network 2001:E::/64
cdelax(config-router)# network ens33
cdelax(config-router)# network ens37
cdelax(config-router)# network ens37.10
cdelax(config-router)# network ens37.20
```

17. Configuración en OSPF

```
cdelax(config)# router ospf
cdelax(config-router)# ospf router-id 0.0.0.0
cdelax(config-router)# network 4.0.0.8/30 area 0.0.0.0
cdelax(config-router)# network 1.0.0.24/29 area 0.0.0.0
cdelax(config-router)# network 1.0.0.16/29 area 0.0.0.0
```

Configuración de switches

1. Se conecta físicamente a la consola del switch con el serial de la PC.
2. Se abre el Putty, con *sudo putty* en la tabla de comandos, para poder acceder a la consola.
3. Configuramos: *hostname* e *interfaces*. Se le da nombre al switch, seguido a esto con el comando *Interface range fa0/1-23* para poder las interfaces al mismo tiempo.
    a. *switchport access vlan "#"*
    b. *switchport mode access*
4. En la interfaz *fa0/24* agregamos el siguiente comando: *switchport trunk native vlan 1*
5. Se configuran las últimas interfaces del Switch, GigabitEthernet, sólo se configura la interfaz que se conecta a otro switch y la que sale al router. Se pone el comando: *switchport mode trunk*. Para poder comunicar las VLANs de los switches.

Configuración de Routers

Los routers de esta práctica tuvieron configuraciones diferentes.

1. Se le dan nombre al router con *hostname*
2. Se activa Ipv6 con *ipv6 unicast-routing*
3. Dependiendo de la interfaz, esta se configura de la siguiente manera:
    a. Se agregan las ips correspondientes con: *Ip address "direccion" "subnet mask"* e *ipv6 address "direccion/mask"*
    b. Se activa ripng con *ipv6 rip "name" enable*
    c. Se prende la interfaz con *no shutdown*
4. Se configura el Ospf
    a. Para este método de enrutamiento se agrega un router-id con el comando *router-id "número de 32 bits"*
    b. Se agregan las redes que este router conoce con *Network "red ej. 1.0.0.1" "wildcard" area "#"*

Esta fue la configuración básica de todos los routers, dos routers tuvieron que agregar una interfaz loopback la cual una vez prendida se le agrega la ip. Con la siguiente configuración.

Interface loopback "#"

Ip address "direccion" "subnet mask"

Ipv6 address "direccion/mask"
Ipv6 rip "name" enable
no shutdown

En el caso de los routers que están conectados físicamente a los switches además de la configuración base, se hace una modificación en la interfaz directamente conectada al switch. Se crean dos interfaces, que corresponden a cada VLAN, seguido a esto se configura. Esta es la configuración de estos routers.

ej.

Interface g0/1.10
encapsulation dot1Q 10
Ip address "direccion" "subnet mask"
Ipv6 address "direccion/mask"
Ipv6 rip "name" enable
no shutdown

exit
Interface g0/1.20
encapsulation dot1Q 20
Ip address "direccion" "subnet mask"
Ipv6 address "direccion/mask"
Ipv6 rip "name" enable
no shutdown

V. Problemas y Soluciones

En el proceso durante esta práctica sucedieron varios problemas que no permitieron proceder con esta. Por ejemplo, el primer obstáculo a presentarse fue con el uso de la máquina virtual como router, y el ping del router-linux hacía cualquier conexión que tuviera, este problema se presentó debido a que la máquina virtual que estaba usando no tomaba correctamente los puertos físicos de la laptop, lo que hacía que cualquier configuración que se hiciera dentro de esta, no sirviera, al igual a la hora de intentar mandar o recibir paquetes. La solución para este problema era configurar correctamente la máquina virtual, en este caso usamos el software VMware Workstation Player, el cual no viene con la opción de poder hacer este arreglo, con la implementación extra de un programa que el VMware Workstation si ofrece, el Player podía cumplir con lo requerido, una correcta asignación de puertos físicos con los virtuales de esta máquina. Otro detalle a considerar era que dentro de la máquina física donde se tenía la virtual, en su sistema operativo, en este caso Windows 11, se tenía que hacer un cambio en los puertos físicos. Dentro del panel de control y conexiones de red, se modifica una propiedad del adaptador. En opciones avanzadas se cambia la prioridad y VLAN que por defecto viene activada a desactivada. Junto a estos cambios hechos se pudo proseguir con la práctica.

VI. Resultados

Los resultados en esta práctica fueron los esperados, aunque hubo claramente varios obstáculos durante el desarrollo de esta, la satisfacción de lograr lo propuesto, nadie la quita. Se logró una exitosa conexión entre todos los dispositivos conectados, las VLANs estaban puestas en los switches y cumpliendo su función, junto a los enlaces troncales. Se logró exitosamente el uso de una máquina virtual para que cumpla las funciones de un router, la configuración correcta de todos los

dispositivos, justo a la conexión física adecuada, se pudo lograr lo que se buscaba desde el principio de la práctica. *(Ver anexos pings)*

VII. Conclusiones

Para concluir, en esta práctica se crearon exitosamente las VLANs en los switches, estableciendo una correcta comunicación entre ellas gracias a los métodos de enrutamiento y las conexiones troncales. La implementación de la máquina virtual Linux, usada como router, cumple perfectamente su propósito y lo configurado dentro de ella, el enrutamiento correcto entre la red exterior junto a las subredes de su conexión con el switch.

Esta práctica enseña la importancia de una correcta planificación, y la necesidad de estar en constante investigación para poder lograr el objetivo. El uso de nuevas tecnologías como la máquina virtual y FRR obligó a estar en constante búsqueda de adaptaciones y solución de errores debido a estas nuevas herramientas. El logro exitoso de esta práctica demuestra el conocimiento que se tiene y la capacidad de resolver errores en este entorno.

Carlos Daniel de la Cruz García

Honestamente lo más difícil de esta práctica fue la investigación y la configuración para hacer funcionar una laptop como enrutador en Linux puesto que nunca había tenido un acercamiento a este tipo al sistema ni a una máquina virtual.

Referencias

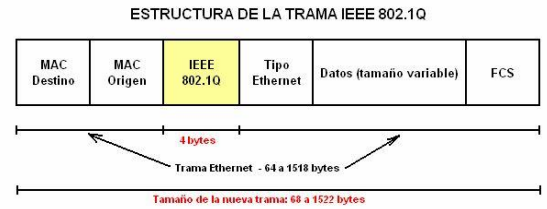
Harmoush, E. (2022, May 17). Virtual Local Area Networks (VLANs). Recuperado el 2 de diciembre de 2023, de https://www.practicalnetworking.net/stand-alone/vlans/

VMware. (s.f.). What is a Virtual Machine? | VMware Glossary. Recuperado el 2 de diciembre de 2023, de https://www.vmware.com/topics/glossary/content/virtual-machine.html

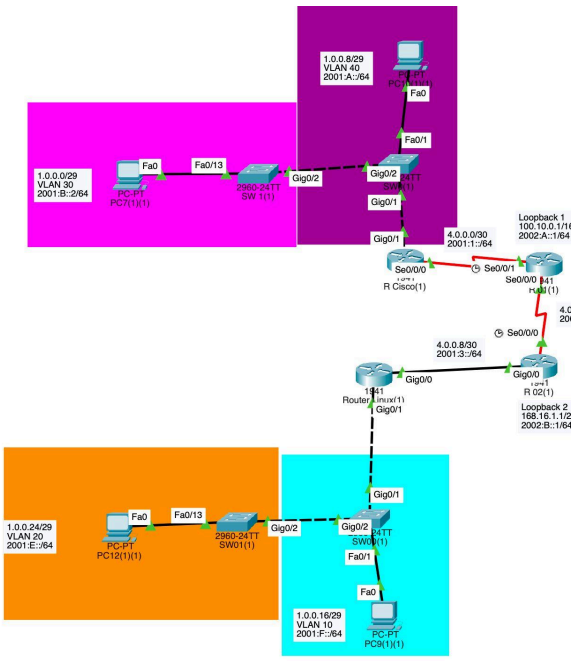
Configuración inicial de un router. (s.f.). https://www.sapalomera.cat/moodlecf/RS/2/course/module4/4.1.3.4/4.1.3.4.html

Anexos

Anexo 1



Anexo 2



Pings

Ipv4

```
64 bytes from 4.0.0.10: icmp_seq=4 ttl=64 time=1.26 ms
^Z
[5]+  Detenido                ping 4.0.0.10
labredes@T20800:~$
labredes@T20800:~$
labredes@T20800:~$ ping 1.0.0.25
PING 1.0.0.25 (1.0.0.25) 56(84) bytes of data.
64 bytes from 1.0.0.25: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 1.0.0.25: icmp_seq=2 ttl=64 time=1.99 ms
64 bytes from 1.0.0.25: icmp_seq=3 ttl=64 time=1.74 ms
64 bytes from 1.0.0.25: icmp_seq=4 ttl=64 time=1.88 ms

64 bytes from 1.0.0.25: icmp_seq=5 ttl=64 time=1.80 ms
64 bytes from 1.0.0.25: icmp_seq=6 ttl=64 time=1.44 ms
Z64 bytes from 1.0.0.25: icmp_seq=7 ttl=64 time=1.71 ms

64 bytes from 1.0.0.25: icmp_seq=8 ttl=64 time=1.65 ms
^Z
[6]+  Detenido                ping 1.0.0.25
labredes@T20800:~$ ping 1.0.0.17
PING 1.0.0.17 (1.0.0.17) 56(84) bytes of data.
64 bytes from 1.0.0.17: icmp_seq=1 ttl=64 time=1.28 ms
64 bytes from 1.0.0.17: icmp_seq=2 ttl=64 time=2.03 ms
64 bytes from 1.0.0.17: icmp_seq=3 ttl=64 time=1.85 ms
64 bytes from 1.0.0.17: icmp_seq=4 ttl=64 time=1.91 ms
64 bytes from 1.0.0.17: icmp_seq=5 ttl=64 time=1.83 ms
^Z
[7]+  Detenido                ping 1.0.0.17
labredes@T20800:~$ ping 1.0.0.27
PING 1.0.0.27 (1.0.0.27) 56(84) bytes of data.
From 1.0.0.30 icmp_seq=1 Destination Host Unreachable
From 1.0.0.30 icmp_seq=2 Destination Host Unreachable
From 1.0.0.30 icmp_seq=3 Destination Host Unreachable
^Z
[8]+  Detenido                ping 1.0.0.27
labredes@T20800:~$ ping 1.0.0.18
PING 1.0.0.18 (1.0.0.18) 56(84) bytes of data.
From 1.0.0.25 icmp_seq=1 Destination Host Unreachable
From 1.0.0.25 icmp_seq=2 Destination Host Unreachable
From 1.0.0.25 icmp_seq=3 Destination Host Unreachable
From 1.0.0.25 icmp_seq=4 Destination Host Unreachable
^Z
[9]+  Detenido                ping 1.0.0.18
labredes@T20800:~$
labredes@T20800:~$
labredes@T20800:~$ ping 1.0.0.13
PING 1.0.0.13 (1.0.0.13) 56(84) bytes of data.
64 bytes from 1.0.0.13: icmp_seq=1 ttl=61 time=5.29 ms
64 bytes from 1.0.0.13: icmp_seq=2 ttl=61 time=4.73 ms
64 bytes from 1.0.0.13: icmp_seq=3 ttl=61 time=4.20 ms
64 bytes from 1.0.0.13: icmp_seq=4 ttl=61 time=4.55 ms
64 bytes from 1.0.0.13: icmp_seq=5 ttl=61 time=4.55 ms
64 bytes from 1.0.0.13: icmp_seq=6 ttl=61 time=3.84 ms
^Z
```

I

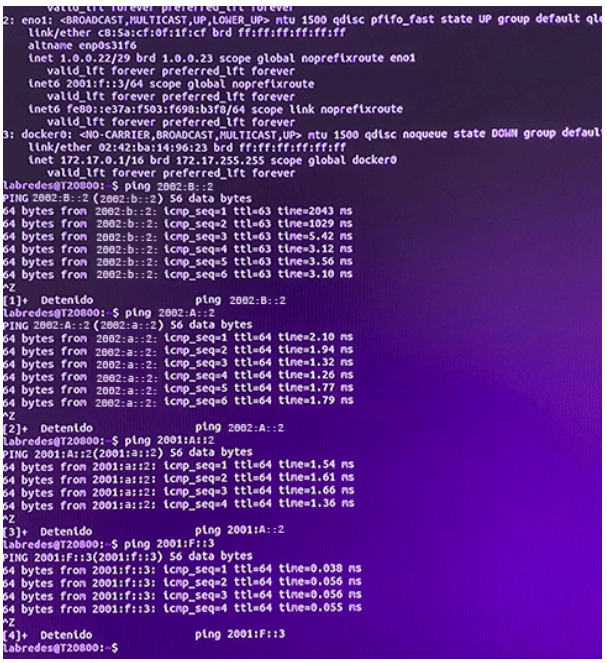
```
root@cdelax:~# ping 1.0.0.30
PING 1.0.0.30 (1.0.0.30) 56(84) bytes of data.
64 bytes from 1.0.0.30: icmp_seq=1 ttl=64 time=1.22 ms
64 bytes from 1.0.0.30: icmp_seq=2 ttl=64 time=1.99 ms
64 bytes from 1.0.0.30: icmp_seq=3 ttl=64 time=1.80 ms
64 bytes from 1.0.0.30: icmp_seq=4 ttl=64 time=1.58 ms
64 bytes from 1.0.0.30: icmp_seq=5 ttl=64 time=1.75 ms
64 bytes from 1.0.0.30: icmp_seq=6 ttl=64 time=1.25 ms
64 bytes from 1.0.0.30: icmp_seq=7 ttl=64 time=1.85 ms
```

```
root@cdelax:~# ping 1.0.0.13
PING 1.0.0.13 (1.0.0.13) 56(84) bytes of data.
64 bytes from 1.0.0.13: icmp_seq=1 ttl=62 time=2.27 ms
64 bytes from 1.0.0.13: icmp_seq=2 ttl=62 time=2.86 ms
64 bytes from 1.0.0.13: icmp_seq=3 ttl=62 time=2.68 ms
64 bytes from 1.0.0.13: icmp_seq=4 ttl=62 time=2.70 ms
64 bytes from 1.0.0.13: icmp_seq=5 ttl=62 time=2.41 ms
64 bytes from 1.0.0.13: icmp_seq=6 ttl=62 time=2.01 ms
```

```
[4]+  Detenido                ping 2001:F::3
labredes@T20800:~$
labredes@T20800:~$ ping 1.0.0.13
PING 1.0.0.13 (1.0.0.13) 56(84) bytes of data.
64 bytes from 1.0.0.13: icmp_seq=1 ttl=61 time=6.73 ms
64 bytes from 1.0.0.13: icmp_seq=2 ttl=61 time=3.99 ms
64 bytes from 1.0.0.13: icmp_seq=3 ttl=61 time=3.48 ms
64 bytes from 1.0.0.13: icmp_seq=4 ttl=61 time=3.77 ms
^Z
[5]+  Detenido                ping 1.0.0.13
labredes@T20800:~$ ping 100.10.0.1
PING 100.10.0.1 (100.10.0.1) 56(84) bytes of data.
64 bytes from 100.10.0.1: icmp_seq=1 ttl=254 time=3.34 ms
64 bytes from 100.10.0.1: icmp_seq=2 ttl=254 time=3.91 ms
64 bytes from 100.10.0.1: icmp_seq=3 ttl=254 time=4.27 ms
^Z
[6]+  Detenido                ping 100.10.0.1
labredes@T20800:~$ ping 1.0.0.30
PING 1.0.0.30 (1.0.0.30) 56(84) bytes of data.
64 bytes from 1.0.0.30: icmp_seq=1 ttl=63 time=3.84 ms
64 bytes from 1.0.0.30: icmp_seq=2 ttl=63 time=3.56 ms
64 bytes from 1.0.0.30: icmp_seq=3 ttl=63 time=3.19 ms
^Z
[7]+  Detenido                ping 1.0.0.30
labredes@T20800:~$ ping 1.0.0.22
PING 1.0.0.22 (1.0.0.22) 56(84) bytes of data.
64 bytes from 1.0.0.22: icmp_seq=1 ttl=64 time=0.061 ms
64 bytes from 1.0.0.22: icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from 1.0.0.22: icmp_seq=3 ttl=64 time=0.066 ms
^Z
[8]+  Detenido                ping 1.0.0.22
labredes@T20800:~$ ping 168.16.1.1
PING 168.16.1.1 (168.16.1.1) 56(84) bytes of data.
64 bytes from 168.16.1.1: icmp_seq=1 ttl=254 time=3.01 ms
64 bytes from 168.16.1.1: icmp_seq=2 ttl=254 time=3.16 ms
64 bytes from 168.16.1.1: icmp_seq=3 ttl=254 time=3.90 ms
64 bytes from 168.16.1.1: icmp_seq=4 ttl=254 time=3.66 ms
^Z
[9]+  Detenido                ping 168.16.1.1
labredes@T20800:~$ ping 168.16.1.1
PING 168.16.1.1 (168.16.1.1) 56(84) bytes of data.
64 bytes from 168.16.1.1: icmp_seq=1 ttl=254 time=3.35 ms
64 bytes from 168.16.1.1: icmp_seq=2 ttl=254 time=3.92 ms
64 bytes from 168.16.1.1: icmp_seq=3 ttl=254 time=3.95 ms
64 bytes from 168.16.1.1: icmp_seq=4 ttl=254 time=4.02 ms
^Z
[10]+  Detenido               ping 168.16.1.1
labredes@T20800:~$ ping 1.0.0.6
PING 1.0.0.6 (1.0.0.6) 56(84) bytes of data.
64 bytes from 1.0.0.6: icmp_seq=1 ttl=61 time=5.39 ms
64 bytes from 1.0.0.6: icmp_seq=2 ttl=61 time=4.32 ms
64 bytes from 1.0.0.6: icmp_seq=3 ttl=61 time=17.3 ms
^Z
[11]+  Detenido               ping 1.0.0.6
labredes@T20800:~$
```

Ipv6

```
labredes@T20800:~$ ping
PING 2001:b::2 (2001:b::2) 56 data bytes
64 bytes from 2001:b::2: icmp_seq=1 ttl=64 time=4.89 ms
64 bytes from 2001:b::2: icmp_seq=2 ttl=64 time=1.70 ms
64 bytes from 2001:b::2: icmp_seq=3 ttl=64 time=1.98 ms
^Z
[1]+  Detenido    2001:b::2   ping
labredes@T20800:~$
labredes@T20800:~$ ping 2001:E::2
PING 2001:E::2(2001:e::2) 56 data bytes
64 bytes from 2001:e::2: icmp_seq=1 ttl=63 time=7.04 ms
64 bytes from 2001:e::2: icmp_seq=2 ttl=63 time=3.25 ms
64 bytes from 2001:e::2: icmp_seq=3 ttl=63 time=3.60 ms
64 bytes from 2001:e::2: icmp_seq=4 ttl=63 time=3.30 ms
^Z
[2]+  Detenido                ping 2001:E::2
labredes@T20800:~$
labredes@T20800:~$
labredes@T20800:~$
```


valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default ql
link/ether c8:5a:cf:0f:1f:cf brd ff:ff:ff:ff:ff:ff
altname enp0s31f6
inet 1.0.0.22/29 brd 1.0.0.23 scope global noprefixroute eno1
valid_lft forever preferred_lft forever
inet6 2001:f::3/64 scope global noprefixroute
valid_lft forever preferred_lft forever
inet6 fe80::e37a:f503:f698:b3f8/64 scope link noprefixroute
valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group defaul
link/ether 02:42:ba:14:96:23 brd ff:ff:ff:ff:ff:ff
inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
valid_lft forever preferred_lft forever
labredes@T20800:~$ ping 2002:B::2
PING 2002:B::2 (2002:b::2) 56 data bytes
64 bytes from 2002:b::2: icmp_seq=1 ttl=63 time=2043 ms
64 bytes from 2002:b::2: icmp_seq=2 ttl=63 time=1029 ms
64 bytes from 2002:b::2: icmp_seq=3 ttl=63 time=5.42 ms
64 bytes from 2002:b::2: icmp_seq=4 ttl=63 time=3.12 ms
64 bytes from 2002:b::2: icmp_seq=5 ttl=63 time=3.56 ms
64 bytes from 2002:b::2: icmp_seq=6 ttl=63 time=3.10 ms
^Z
[1]+ Detenido ping 2002:B::2
labredes@T20800:~$ ping 2002:A::2
PING 2002:A::2 (2002:a::2) 56 data bytes
64 bytes from 2002:a::2: icmp_seq=1 ttl=64 time=2.10 ms
64 bytes from 2002:a::2: icmp_seq=2 ttl=64 time=1.94 ms
64 bytes from 2002:a::2: icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from 2002:a::2: icmp_seq=4 ttl=64 time=1.26 ms
64 bytes from 2002:a::2: icmp_seq=5 ttl=64 time=1.77 ms
64 bytes from 2002:a::2: icmp_seq=6 ttl=64 time=1.79 ms
^Z
[2]+ Detenido ping 2002:A::2
labredes@T20800:~$ ping 2001:A::2
PING 2001:A::2(2001:a::2) 56 data bytes
64 bytes from 2001:a::2: icmp_seq=1 ttl=64 time=1.54 ms
64 bytes from 2001:a::2: icmp_seq=2 ttl=64 time=1.61 ms
64 bytes from 2001:a::2: icmp_seq=3 ttl=64 time=1.66 ms
64 bytes from 2001:a::2: icmp_seq=4 ttl=64 time=1.36 ms
^Z
[3]+ Detenido ping 2001:A::2
labredes@T20800:~$ ping 2001:F::3
PING 2001:F::3(2001:f::3) 56 data bytes
64 bytes from 2001:f::3: icmp_seq=1 ttl=64 time=0.038 ms
64 bytes from 2001:f::3: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 2001:f::3: icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from 2001:f::3: icmp_seq=4 ttl=64 time=0.055 ms
^Z
[4]+ Detenido ping 2001:F::3
labredes@T20800:~$