

Configuración de /etc/network/interfaces en un sistema GNU/Linux (derivado Debian) y otros ajustes de la red Tests de puertos

Alfredo Abad

04-TCP-IP-Básico_Ext-ConfigLinux.pptx

UA: 27-ago-2019

<https://wiki.debian.org/es/networkConfiguration>

1

Utilizar DHCP para configurar la interfaz automáticamente

- IPv4
 - **auto eth0**
 - **allow-hotplug eth0**
(allow-hotplug permite conexión y desconexión en caliente)
 - **iface eth0 inet dhcp**
- IPv6 (añadir)
 - **iface eth0 inet6 dhcp**
- Alternativamente, se puede autoconfigurar IPv6 utilizando autoconfiguración de dirección sin estado, o SLAAC (*stateless address autoconfiguration*), que se especifica utilizando auto en vez de dhcp en la línea de inet6
 - **iface eth0 inet6 auto**

2

Configurar la interfaz manualmente

- IPv4
 - **auto eth0**
 - **iface eth0 inet static**
 - **address 192.0.2.7**
 - **netmask 255.255.255.0**
 - **gateway 192.0.2.254**
- IPv6
 - **iface eth0 inet6 static**
 - **address 2001:db8::c0ca:1eaf**
 - **[parcial]netmask 255.255.255.0**
 - **gateway 2001:db8::1ead:ed:beef**

3

Configurar la velocidad y bidireccionalidad dúplex (I)

- El que falle repetidamente la autonegociación es frecuentemente un síntoma de cableado defectuoso, así que investigue los problemas físicos antes de asumir que sean incompatibles los algoritmos de autonegociación de las interfaces
- Si apaga la autonegociación y configura manualmente la velocidad y la bidireccionalidad entonces la interfaz compañera del otro cabo del cable asumirá que la ausencia de autonegociación indica una velocidad de 10Mbps y semi-bidireccionalidad o half-duplex
- Para operar sin errores, si usted establece manualmente la velocidad y bidireccionalidad, se debe usted asegurar de que se configuran las mismas condiciones de velocidad y bidireccionalidad en la interfaz compañera

4

Configurar la velocidad y bidireccionalidad dúplex (II)

- Si pone a mano la velocidad y bidireccionalidad de su interfaz, hará falta hacer ciertos ejercicios de prueba y error. Aquí tiene los pasos básicos:
- Instale los paquetes [ethtool](#) y [net-tools](#), para que así tenga los programas ethtool y mii-tool. Uno o sendos podrían funcionar con su interfaz.
- Asegúrese de que tiene algún modo de entrar al sistema en caso de que la interfaz de red se quede fuera de servicio. Una conexión [SSH](#) podría ser interrumpida, así que se necesita una estrategia de último recurso.
- Identifique la interfaz en cuestión (será a menudo la eth0). Ajuste las instrucciones restantes en consecuencia.
- Intente determinar cuáles son los ajustes de velocidad y bidireccionalidad:
 - Como root, intente primero ethtool eth0, y vea si las líneas velocidad "Speed:" y bidireccionalidad "Duplex:" parecen válidas. Si no, puede que la ethtool no esté soportada por su dispositivo.
 - Como root, intente mii-tool -v eth0 y vea si su salida parece correcta. Si no, puede que la mii-tool no esté soportada por su dispositivo.
 - Si ninguna está soportada, puede que necesite poner los parámetros directamente en el módulo del kernel driver. Identifique cuál módulo driver está usted utilizando leyendo la salida de dmesg y lsmod. Puede entonces intentar modinfo NOMBREDEMODULO para ver qué parámetros acepta, si es que lo hace. (Puede utilizar modinfo inclusive en módulos que no están cargados, por comparar.) [ToDo](#): ¿Dónde se ponen los parámetros del módulo de kernel?

5

Configurar velocidad y bidireccionalidad dúplex (y III)

- Siguiente, intente cambiar los ajustes de la interfaz mientras este operando. Usted necesitará ser root, por supuesto. Bien:
 - **ethtool -s eth0 speed 100 duplex full autoneg off** (asumiendo 100 Mbps y bidireccionalidad full duplex)
 - **mii-tool -F 100baseTx-FD eth0** (idéntica asunción)
 - En cada caso, compruebe para ver si los ajustes de la interfaz cambiaron realmente, y entonces intente enviar algunos datos desde y al sistema para ver si la tarjeta de red o NIC está operando correctamente.
- Si una de esas órdenes configura exitosamente su NIC, entonces puede ponerla en /etc/network/interfaces tal que se ejecute cuando usted levante la interfaz (p.e. durante el arranque). Sin embargo, antes de que lo haga, debería entender que algunos drivers y dispositivos se comportan de distinta manera a otros. Cuando se cargue el módulo del driver, la NIC puede comenzar la autonegociación sin que nada la pare (particularmente con drivers que no acepten parámetros). Los ajustes de interfaces se aplican en algún momento tras eso, que puede ser justo en medio de la negociación. Así, que algunas personas encuentran necesario retrasar la orden ethtool o mii-tool por unos segundos. Del siguiente modo:
 - **iface eth0 inet static**
 - **address ...**
 - **netmask ...**
 - **gateway ...**
 - **up sleep 5; ethtool -s eth0 ...**
 - O la orden análoga mii-tool.
- Reinicie la máquina para asegurarse de que se levanta correctamente, y estése preparado para intervenir manualmente (p.e. Ctrl-Alt-Del e iniciar en modo single-user desde GRUB o LILO) si as cosas o funcionasen.

6

Levantar una interfaz sin dirección IP

- Para crear una interfaz de red sin dirección IP utilice el método manual y use las órdenes pre-up y post-down para levantar o tirar la interfaz.
 - **iface eth0 inet manual**
 - **pre-up ifconfig \$IFACE up**
 - **post-down ifconfig \$IFACE down**
- Si la interfaz es una interfaz VLAN, las órdenes levantar/tirar deben de ejecutarse antes/después de los enlaces VLAN. (También usted tiene que instalar el paquete vlan.)
 - **iface eth0.99 inet manual**
 - **post-up ifconfig \$IFACE up**
 - **pre-down ifconfig \$IFACE down**
- Nota:
 - Si usted crea la interfaz VLAN solo para ponerla en un puente o bridge, no hay necesidad de definir manualmente la interfaz VLAN. Justo configure el puente, y la interfaz VLAN se creará automáticamente en creando el puente (véase abajo).

7

Definir los Servidores de nombres (DNS)

- Antes de que un ordenador pueda conectarse a un recurso de una red externa (sea, por ejemplo, un servidor web), debe tener los medios para convertir cualquier nombre alfanumérico (p.e. wiki.debian.org) en direcciones de red numéricas (p.e. 140.211.166.4).
- La C-library y otras bibliotecas de resolución buscan /etc/resolv.conf para tener una lista de servidores de nombres. En el caso más simple, éste es el archivo a editar para poner la lista de servidores de nombres. Pero note que otros programas para configuraciones dinámicas estarán encantados de sobrescribir ésas sus ajustes:
 - El programa resolvconf
 - El demonio network-manager
 - Clientes DHCP
- En la mayoría de las situaciones, el archivo a editar es el de configuración para cada programa.
- En las más complejas situaciones, utilizar resolvconf es verdaderamente el modo de hacer las cosas, a pesar de que en configuraciones más simples sea probablemente abrumador.

8

El archivo de configuración resolv.conf

- El archivo de configuración **resolv.conf** en `/etc/resolv.conf` contiene información que permite a un ordenador conectado a una red resolver nombres a direcciones
 - Nota: No confunda este *archivo de configuración* con el *programa resolvconf*, el cual desafortunadamente tiene casi un nombre idéntico.
- El archivo **resolv.conf** típicamente contiene las direcciones IP de los servidores de nombres (DNS name resolvers) que tratarán de traducir los nombres a direcciones para cualquier nodo disponible de la red. Habrá una o varias líneas que se parezcan a esto:
 - **nameserver 12.34.56.78**
 - **nameserver 12.34.56.79**
- En este ejemplo, el sistema está utilizando servidores de nombres en las direcciones IP 12.34.56.78 y 12.34.56.79
 - Simplemente edite el archivo e introduzca las direcciones IP de los servidores de nombres que usted necesite utilizar tras cada nameserver. Añada más líneas nameserver si tiene más servidores de nombres. **No utilice este método si tiene instalado el programa resolvconf.**
- El archivo de configuración **resolv.conf** tiene muchas otras opciones para definir cómo el resolutor busca los nombres. Véase **man resolv.conf** para más detalles.

9

El programa resolvconf (I)

- El programa **resolvconf** mantiene un seguimiento de la información del sistema sobre los servidores de nombres actualmente disponibles. No debería confundirse con el *archivo de configuración resolv.conf*, el cual desafortunadamente tiene casi un nombre idéntico. El programa **resolvconf** es opcional en un sistema Debian.
- El archivo de configuración **resolv.conf** contiene información sobre los servidores de nombres a utilizar por el sistema. Sin embargo, cuando múltiples programas necesitan modificar dinámicamente el archivo de configuración **resolv.conf** se pueden pisar unos a otros y quedarse desincronizado
 - El programa **resolvconf** soluciona ese problema. Actúa como intermediario entre programas que entreguen la información de servidor de nombres (p.e. clientes dhcp) y programas que utilicen la información de servidor de nombres (p.e. resolver).
- Cuando el **resolvconf** está apropiadamente instalado, el archivo de configuración **resolv.conf** en `/etc/resolv.conf` se reemplaza por un enlace simbólico a `/etc/resolvconf/run/resolv.conf` y el resolutor utiliza en cambio el archivo de configuración que se genera dinámicamente por **resolvconf** en `/etc/resolvconf/run/resolv.conf`.
- El programa **resolvconf** se necesita generalmente solo cuando un sistema tiene múltiples programas que necesitan modificar dinámicamente la información del servidor de nombres. En un sistema simple donde los servidores de nombres no cambian a menudo o son solo cambiados por un programa, el archivo de configuración **resolv.conf** es adecuado.

10

El programa resolvconf (y II)

- Si se instala el programa **resolvconf**, usted no debería editar el archivo de configuración **resolv.conf** manualmente ya que será cambiado dinámicamente por programas del sistema
 - Si necesita definir manualmente los servidores de nombres (como con la interfaz estática), añada una línea como la siguiente al archivo de configuración **interfaces** en **/etc/network/interfaces**.
 - **dns-nameservers 12.34.56.78 12.34.56.79**
- Coloque la línea indentada dentro de la estrofa iface, p.e., justo tras la línea gateway. Introduzca las direcciones IP de los servidores de nombres que usted necesite utilizar detrás de dns-nameservers
 - Ponga todos ellos en una sola línea separados por espacios. No se olvide es "s" al final de dns-nameservers.
- El programa **resolvconf** es un añadido bastante reciente en Debian y muchos programas más viejos necesitan actualizarse o reconfigurarse para trabajar correctamente con ello
 - Si tiene usted problemas, véase /usr/share/doc/resolvconf/README. Tiene montones de información sobre hacer que otros programas funcionen con **resolvconf**.

11

Configuración de DNS para NetworkManager

- [NetworkManager](#) anulará los ajustes dhcp, sobrescribiendo resolv.conf incluso si usted ha configurado DNS en /etc/dhcp/dhclient.conf, p.e. causando que DNS busque primero el dominio local, el cual puede tener que parar antes de que continúe la resolución de DNS provocando largas esperas de resolución DNS. Usted puede hacerse una idea de cómo cree [NetworkManager](#) que deben ser los ajustes ejecutando nm-tool en la línea de comandos.
- Puede configurar estos ajustes utilizando una GUI que lanza nm-connection-editor, actualmente (2015/01/22) está en Aplicaciones → Herramientas de sistema → Preferencias → Conexiones de red. Tras lanzar:
- Escoja una conexión (desde la pestaña cableado o inalámbrico) y pique Editar.
- Haga clic en la pestaña Ajustes de IPv4
- Escoja 'Sólo direcciones automáticas (DHCP)' en vez de justo 'Automático (DHCP)'.
- Introduzca los servidores DNS en el campo "Servidores DNS", separados por espacios (p.e. 208.67.222.222 para OpenDNS).
- Pique "Guardar."
- [NetworkManager](#) guarda estos ajustes en /etc/NetworkManager/system-connections/nombre-de-conexión
 - Ejemplo /etc/NetworkManager/system-connections/Wired connection 1 :

12

Ejemplo de fichero /etc/NetworkManager/system-connections/Wired connection 1

```
[802-3-ethernet]
duplex=full
mac-address=XX:XX:XX:XX:XX:XX

[connection]
id=Wired connection 1
uuid=xxx-xxxxxx-xxxxxx-xxxxxx-xxx
type=802-3-ethernet
timestamp=1385213042

[ipv6]
method=auto

[ipv4]
method=auto
dns=208.67.222.222;
ignore-auto-dns=true
```

Ejecutar de nuevo nm-tool debería mostrar que NetworkManager ahora tiene la idea correcta de cómo el DNS suyo debería resolverse.

13

Puentear o Bridging (I)

- Puentear pone múltiples interfaces en el mismo segmento de red. Esto es muy popular cuando se conecta un servidor a múltiples conmutadores para alta disponibilidad o con virtualización. En este último caso normalmente se utiliza para crear un puente en el host (p.e. dom0) y poner las interfaces virtuales de los invitados (domU) en el puente.
- Ejemplo: Conecte un servidor a dos conmutadores (vía eth0 y eth1) definiendo bridge 0 y dé al servidor una dirección IP en esta subred:

```
auto br0
iface br0 inet static
    address 10.10.0.15
    netmask 255.255.255.0
    gateway 10.10.0.1
    bridge_ports eth0 eth1
    up /usr/sbin/brctl stp br0 on
```

14

Puentear o Bridging (II)

- Si un servidor está conectado a múltiples conmutadores entonces usted necesitará ejecutar el protocolo spanning tree (STP) para evitar bucles. Por tanto STP debe ser encendido vía una orden "up" como se muestra en la diapo anterior
- Ejemplo: Ajuste de Puente sin configuración de dirección IP address (utilícese "manual" en vez de "static") para "reenviar" ("forward") una interfaz a una VM invitada. (La configuración del puente estático contiene sólo una interfaz física. La interfaz virtual se añadirá al puente cuando la VM se inicie.)

```
auto br1  
  
iface br1 inet manual  
    bridge_ports eth4  
  
    up /usr/sbin/brctl setageing br1 0  
  
    up /usr/sbin/brctl stp br0 off
```

15

Puentear o bridging (y III)

- Nota:
 - El puente de Linux sólo soporta STP, no RSTP (Rapid Spanning Tree). Por tanto soporta sólo los viejos costes de STP, no los nuevos costes de RSTP (véase [Spanning Tree Protocol](#))
 - Esto esta bien habitualmente con Conmutadores Cisco, pero p.e. los Conmutadores Juniper utilizan los costes RSTP y por tanto eso puede llevar a diferentes cálculos de spanning tree y problemas de bucles
 - Se puede arreglar ajustado los costes manualmente, bien en el conmutador o en el servidor
 - Se prefiere ajustar el coste en el conmutador ya que Linux cambia a los costes por defecto cada vez que una interfaz se cae y levanta

16

Puentear sin conmutación

- Por defecto el puente de Linux funciona como un conmutador. Esto significa que, recuerda las direcciones MAC detrás del puerto de conmutación y si se conoce la dirección de destino MAC, los paquetes de datos son sólo reenviados a su respectivo puerto - de otro modo los paquetes serán difundidos
- En algunos ajustes eso es malo. Por ejemplo si el puente conecta dos interfaces troncales y se puede ver la misma dirección MAC desde ambas, dependiendo de la VLAN
- Ya que el puente Linux no soporta VLANs (tablas de direcciones MAC dedicadas por cada VLAN), en dichos ajustes tiene usted que desactivar el aprendizaje de direcciones MAC y poner el puente en modo "puente" verdadero con:
 - **up /sbin/brctl setageing br0 0**
 - **up /sbin/brctl stp br0 off**

17

Cómo utilizar VLAN (dot1q, 802.1q, trunk, Etch, Lenny)

```
modprobe 8021q

apt-get install vlan

vconfig add eth0 222    # 222 es el número de VLAN
ifconfig eth0.222 up
ifconfig eth0.222 mtu 1496    #opcional si su NIC no soporta MTU 1504B
ifconfig eth0.222 10.10.10.1 netmask 255.255.255.0
```

18

Configuración de script inicial de red

- Añada la línea a /etc/modules:
 - **8021q**
- En /etc/network/interfaces añade el parámetro a la sección iface:
 - **vlan-raw-device eth0**
- El nombre de la interfaz debería ser el nombre bruto de la interfaz (el mismo que se especifica por vlan-raw-device), luego un punto, después la VLAN ID, por ejemplo eth0.100
 - Puede ser en cambio "vlan" luego la VLAN ID, por ejemplo vlan100. En cualquier caso, la VLAN ID está al otro final, y es en el único lugar que se configura
- Nota:
 - Si usted nombra sus interfaces VLAN como ethX.YYY, entonces no hay necesidad de especificar el vlan-raw-device, ya que el dispositivo bruto puede ser tomado del nombre de la interfaz

19

Ejemplo de configuración VLAN

```
auto eth0.222

iface eth0.222 inet static
    address 10.10.10.1
    netmask 255.255.255.0
    vlan-raw-device eth0
```

20

Puentes y VLANs

- Si usted crea interfaces VLAN sólo para colocarlas en un puente, no se necesita definir las interfaces VLAN manualmente. Justo configure el puente, y la interfaz VLAN se creará automáticamente al crear el puente

```
auto br1
iface br1 inet manual
    bridge_ports eth0.99 eth1.99
    up /usr/sbin/brctl stp br0 on
```

21

Advertencias al utilizar puenteo y VLAN

```
#!/etc/network/interfaces
auto eth0 bri0
iface eth0 inet static
    [parcial]address 10.10.10.1
    netmask 255.255.255.0
iface eth0.110 inet manual
    vlan-raw-device eth0
iface bri0 inet static
    address 192.168.110.1
    netmask 255.255.255.0
    bridge_ports eth0.110
    bridge_stp on
    bridge_maxwait 10
```

22

Creación de uniones tolerantes a fallos con VLAN (Etch)

- Cómo configurar un respaldo activo de servidor uniendo tres redes VLAN {vlan10,vlan20,vlan30} Debian sin SPOF sin VLAN nativas
 - **aptitude install vlan ifenslave-2.6**
- Configuración de Red
 - Configuración de ejemplo de interfaz de Conmutador Cisco
 - **interface GigabitEthernet1/2**
 - **description eth1**
 - **switchport**
 - **switchport trunk encapsulation dot1q**
 - **switchport trunk allowed vlan 10,20,30**
 - **switchport mode trunk**
 - **no ip address**
 - **no cdp enable**
 - **spanning-tree portfast trunk**

23

Unir con Respaldos activos o active backup

- Cree un archivo /etc/modprobe.d/bonding.conf que contenga:
 - **alias bond0 bonding**
 - **options bonding mode=active-backup miimon=100 downdelay=200 updelay=200 primary=eth1**
- Luego habrá que configurar /etc/network/interfaces como aparece en la diapo siguiente

24

25	<pre># La interfaz de red de ciclo cerrado loopback auto lo iface lo inet loopback # La interfaz de red primaria auto bond0 iface bond0 inet manual up ifconfig bond0 0.0.0.0 up slaves eth1 eth0 auto vlan10 iface vlan10 inet static address 10.10.10.12 netmask 255.255.0.0 vlan-raw-device bond0 gateway 10.10.0.1 dns-search hup.hu dns-nameservers 10.10.0.2 auto vlan20 iface vlan20 inet static address 10.20.10.12 netmask 255.255.0.0 vlan-raw-device bond0 auto vlan30 iface vlan30 inet static address 10.30.10.12 netmask 255.255.0.0 vlan-raw-device bond0</pre>	
----	--	--

Múltiples direcciones IP en una Interfaz

- Dar alias de interfaces permite a una interfaz tener múltiples direcciones IP
- Esto es útil cuando se necesita tener visible más de un servidor vía la Internet
 - Note que los hosts virtuales pueden soportar múltiples servidores Apache con una sola dirección IP
 - Apache responde al nombre de dominio entregado por el cliente en la cabecera HTTP
- En muchas otras situaciones, se necesita una IP externa para cada servidor que utilice un puerto

26

Primer método: Modo legado

Un alias de interfaz debería no tener "gateway" o "dns-nameservers"; se admite asignación dinámica de IP

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.42
    netmask 255.255.255.0
    gateway 192.168.1.1

auto eth0:0
allow-hotplug eth0:0
iface eth0:0 inet static
    address 192.168.1.43
    netmask 255.255.255.0

auto eth0:1
allow-hotplug eth0:1
iface eth0:1 inet static
    address 192.168.1.44
    netmask 255.255.255.0
```

27

Segundo método: iproute2

- Actualmente ifupdown utiliza la herramienta ip del paquete iproute2 en vez de ifconfig. La nueva herramienta ip no utiliza el mismo concepto de alias o interfaces virtual
 - Sin embargo, soporta la asignación arbitraria de nombres a las interfaces (se denominan etiquetas). ifupdown utiliza esta característica para soportar las interfaces con alias al usar ip
- También, ifupdown soporta especificar múltiples interfaces repitiendo secciones iface con el mismo nombre de interfaz
 - La diferencia clave respecto del método arriba descrito estriba en que todas esas secciones se tratan por ifupdown como justo una sola interfaz, tal que el usuario no pueda añadir ni eliminarlas individualmente
 - Sin embargo, las órdenes up/down, junto con los scripts, se llaman para cada sección como se solía hacer.

28

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.42
    netmask 255.255.255.0
    gateway 192.168.1.1

iface eth0 inet static
    address 192.168.1.43
    netmask 255.255.255.0

iface eth0 inet static
    address 192.168.1.44
    netmask 255.255.255.0

# añadir direcciones IP desde diferentes subredes es también posible
iface eth0 inet static
    address 10.10.10.14
    netmask 255.255.255.0
```

29

Tercer método: aproximación manual

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.42
    netmask 255.255.255.0
    gateway 192.168.1.1

up    ip addr add 192.168.1.43/24 dev $IFACE label $IFACE:0
down  ip addr del 192.168.1.43/24 dev $IFACE label $IFACE:0

up    ip addr add 192.168.1.44/24 dev $IFACE label $IFACE:1
down  ip addr del 192.168.1.44/24 dev $IFACE label $IFACE:1

up    ip addr add 10.10.10.14/24 dev $IFACE label $IFACE:2
down  ip addr del 10.10.10.14/24 dev $IFACE label $IFACE:2
```

30

<http://www.sysadmit.com/2017/09/linux-deshabilitar-ipv6.html>

En la mayor parte de las distribuciones actuales, IPv6 está habilitado por defecto, pero si o se va a utilizar es mejor deshabilitarlo para evitar ataques a través de la pila IPv6

HABILITAR Y DESHABILITAR IPV6 EN SISTEMAS LINUX

31

Verificar si tenemos IPv6 habilitado con ifconfig

```
[root@LINUX1 ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.19.0.10 netmask 255.255.0.0 broadcast 172.19.255.255
    inet6 fe80::2396:a2f5:8f30:1a95<prefix> prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:1f:8e:f8 txqueuelen 1000 (Ethernet)
    RX packets 15217 bytes 2203719 (2.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 207 bytes 33455 (32.6 KiB)
    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 1 (Local Loopback)
    RX packets 20 bytes 1388 (1.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1388 (1.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

32

Verificar si tenemos IPv6 habilitado con ip

```
[root@LINUX1 ~]# ip -6 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fe80::2396:a2f5:8f30:1a95/64 scope link
        valid_lft forever preferred_lft forever
[root@LINUX1 ~]#
```

33

Verificar si tenemos IPv6 habilitado consultando /proc

La ruta completa es: /proc/net/if_inet6

Ejemplo de IPv6, activado.

```
[root@LINUX1 ~]# cat /proc/net/if_inet6
00000000000000000000000000000001 01 80 10 80      lo
fe8000000000000002396a2f58f301a95 02 40 20 80      ens33
```

Si después de ejecutar el cat, aparece información es que tenemos el protocolo IPv6 activado en el interfaz.

34

Desactivar IPv6 temporalmente hasta el siguiente reinicio de sistema

Para desactivar IPv6 de forma temporal, bastará con ejecutar:

```
sysctl -w net.ipv6.conf.all.disable_ipv6=1  
sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

Después de ejecutar el comando, veremos como la configuración IPv6 de los interfaces, desaparece.

Sin embargo, si reiniciamos el equipo, volvemos a tener dirección IPv6 asignada en el interfaz de red.

35

Desactivar IPv6 permanentemente

Para desactivar IPv6 de forma permanente, deberemos editar el siguiente fichero:

```
/etc/sysctl.conf
```

y al final del mismo, añadir:

```
net.ipv6.conf.all.disable_ipv6=1  
net.ipv6.conf.default.disable_ipv6=1
```

Si reiniciamos el equipo, veremos cómo ya no disponemos de ningún interfaz con IPv6 activado.

- También se puede desactivar ipv6 del lo, añadiendo a /etc/sysctl.conf
 - **net.ipv6.conf.lo.disable_ipv6 = 1**
- Si no se desea reiniciar, los cambios se pueden ejecutar con:
 - **sudo sysctl -p**

36

<https://www.cyberciti.biz/faq/ping-test-a-specific-port-of-machine-ip-address-using-linux-unix/>

TESTS DE PUERTOS DE COMUNICACIONES EN LINUX

37

Mediante el uso de telnet

The syntax is:

```
telnet {host} {port}
telnet www.cyberciti.biz 80
telnet 192.168.2.254 80
```

Sample outputs:

```
Trying 192.168.2.254...
Connected to router.
Escape character is '^]'.
^]
telnet> q
Connection closed.
```

38

To close your session, press **Ctrl** + **]** + **q**.

Mediante el uso del comando nc

The syntax is:

```
nc -vz {host} {port}
nc -vz 192.168.2.254 80
nc -vz www.cyberciti.biz 443
```

Sample outputs:

```
found 0 associations
found 1 connections:
  1: flags=82
    outif utun1
    src 10.8.0.2 port 54997
    dst 104.20.187.5 port 443
    rank info not available
    TCP aux info available
```

39

Connection to www.cyberciti.biz port 443 [tcp/https] succeeded!

Mediante el uso del comando nmap

The syntax is:

```
nmap -PNp {port} {host}
nmap -p {port} {host}
nmap -p 22 www.cyberciti.biz
nmap -p 443 192.168.2.254
```

Sample outputs:

```
Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-24 01:00 IST
Nmap scan report for router (192.168.2.254)
Host is up (0.00034s latency).
PORT      STATE SERVICE
443/tcp   open  https
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

40

Mediante el comando nping

- Nping is an open-source tool for network packet generation, response analysis and response time measurement. Nping allows users to generate network packets of a wide range of protocols, letting them tune virtually any field of the protocol headers. While Nping can be used as a simple ping utility to detect active hosts, it can also be used as a raw packet generator. The syntax is:
- `sudo nping --tcp -p {port} {host}`
- `sudo nping --tcp -p 443 www.cyberciti.biz`
- Sample outputs:

```
(vivek@cyberciti.biz:~)$ sudo nping --tcp -p 443 www.cyberciti.biz
Starting Nping 0.7.40 ( https://nmap.org/nping ) at 2017-05-24 14:08 IST
SENT (0.0051s) TCP 10.8.0.2:35514 > 104.20.186.5:443 S ttl=64 id=21099 iplen=40 seq=3869414838 win=1480
RCVD (0.0556s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>
SENT (1.0076s) TCP 10.8.0.2:35514 > 104.20.186.5:443 S ttl=64 id=21099 iplen=40 seq=3869414838 win=1480
RCVD (1.0459s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>
RCVD (1.0853s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>
SENT (2.0129s) TCP 10.8.0.2:35514 > 104.20.186.5:443 S ttl=64 id=21099 iplen=40 seq=3869414838 win=1480
RCVD (2.0509s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>
SENT (3.0142s) TCP 10.8.0.2:35514 > 104.20.186.5:443 S ttl=64 id=21099 iplen=40 seq=3869414838 win=1480
RCVD (3.0886s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>
SENT (4.0194s) TCP 10.8.0.2:35514 > 104.20.186.5:443 S ttl=64 id=21099 iplen=40 seq=3869414838 win=1480
RCVD (4.0823s) TCP 104.20.186.5:443 > 10.8.0.2:35514 SA ttl=58 id=0 iplen=44 seq=2963972017 win=29200 <msg 1350>

Max rtt: 77.700ms | Min rtt: 37.742ms | Avg rtt: 56.884ms
Raw packets sent: 5 (200B) | Rcvd: 6 (264B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 4.08 seconds
```

41



Estudiar los siguientes documentos:

- 2019_How to Install NTP Server and Client on Ubuntu.pdf

¿CÓMO INSTALAR UN SERVIDOR NTP Y SU CLIENTE EN UBUNTU?

42