configuracion red avanzada

todos los cambios aplicados a través de `ip` son temporales, al reinicar se han perdido

cambiando configuración de red (temporalmente)

asignación de dirección de red:

```
sudo ip address add 10.0.1.1/8 dev ens9
```

eliminación de dirección de red:

```
sudo ip delete add 10.0.1.1/8 dev ens9
```

levantar/parar tarjeta de red:

```
sudo ip link set {up|down} dev ens9
```

asignar puerta de enlace (por defecto solo puede haber una)

```
sudo ip route {add|delete} default via x.x.x.x dev ens9
```

persistiendo configuración de red

debian

```
...
auto ens9
        iface ens9 inet static
        address 192.168.1.2
        netmask 255.255.255.0
        #gateway
        dns-nameservers 8.8.8.8 8.8.4.4
...
```

para que la configuración se aplique, hay que ejecutar `sudo systemctl restart networking`

la instrucción **dns-nameservers** no funcionará directamente, es necesario tener instalado el paquete **resolvconf**

`ping -I <dev> <ip|fqdn>` : forzamos usar una tarjeta de red en concreto para realizar el ping

centos

- **/etc/sysconfig/network-scripts** es la ubicación de todos los scripts de red
- cada tarjeta tiene su fichero de configuración

- ifcfg-ehh0
- ifcfg-lo

```
DEVICE=eth1
BOOTPROTO=none
IPADDR=192.168.1.2
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
NAME=??
```

https://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-networkscripts-interfaces.html

centos (versión moderna)

debian también implementa (que no usa) también este método a través de **systemd**

```
sudo yum install systemd-networkd #en Debian instalado - que no activo, ya que no puede haber dos gestores de networking - por defecto
sudo systemctl start systemd-networkd
sudo md -p /etc/systemd/network
vim eth1.network
```

```
[Match]
Name=eth1

[Network]
DHCP=no
Address=192.168.1.2/24
#Gateway=192.168.1.254
DNS=8.8.8.8
DNS=8.8.4.4
```

```
sudo systemctl restart systemd-networkd
```

comandos

ping

- envía paquetes ICMP
- información de interés:
  - tiempo de respuesta
  - secuencia de los paquetes (salto en la correlación de la numeración)
  - al pulsar Contro+C, aparece: valor mínimo, valor máximo, média, desviación estandard (diferencia entre max y min)
- opciones:
  - `-c` X : número de paquets
  - `-i` X : número de segundos entre peticiones
  - `-I` <dev> : forzar envío a través de una tarjeta en concreto

    - `-f` : (root) follow → envía paquetes a la máxima velocidad posible. Cuando envía escribe un punto, cuando recibe lo borra. Si acumula puntos es síntoma de errores

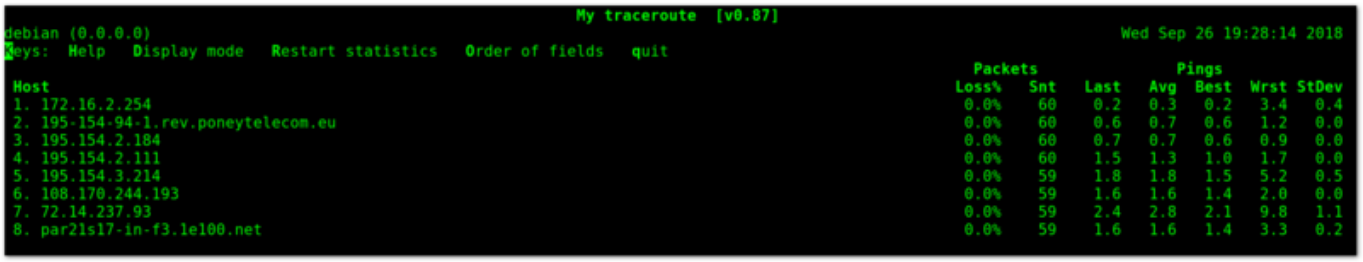
ss

- sustituto de **netstat**
- muestra los sockets abiertos (sin parámetros, todas)
- opciones:
    - `-t` : conexiones tcp
    - `-u` : conexiones udp
    - `-n` : no resuelve el nombre del servicio, muestra el puerto
    - `-p` : proceso asociado al puerto
    - `-l` : puertos que están escuchando

mtr

- o mtr-tiny (versión CLI)
- equivalente a **traceroute**
- nos muestra estadísticas de toda la ruta
- opciones:
    - `-c #` : hace # pasadas y muestra la información
    - `-r`

```
mtr {ip | dominio }
```



ncat

- viene en el paquete **nmap**
- permite realizar conexiones (tcp / udp) entre dos ordenadores
- hay otros «ncat» : **nc** (que ya viene por defecto)
- opciones:
    - `-l` : listen
    - `-p` : port
    - ''-e | –exec '' : vincula un programa, recibe desde el cliente y retorna a través del servidor
- chat con ncat:

```
# ordenador que escucha (servidor)
ncat -l -p <puerto>
```

```
# ordenador que envía (cliente)
ncat <IP> <puerto>
```

- mover un fichero:

```
# ordenador que escucha (servidor)
```

```
ncat -l -p <puerto>
```

```
# otra opción de escucha volcando a fichero
ncat -l -p <puerto> > otrofichero.txt
```

```
cat unfichero.txt | ncat <ip_servidor> <puerto>
```

```
ncat -l -p 5000 -e /bin/bash
```

nmap

- permite ver que puertos tiene abiertos otros ordenadores (escan de puertos)
- permite ver que ordenadores hay conectados en una red (escan de ordenadores/redes)
- opciones:
  - -A : da información extra de los puertos abiertos

scan de red

```
nmap -sn 172.16.3.0/24
```

scan de red

```
nmap -sn 172.16.3.*
```

scan de rango de red

```
nmap -sn 172.16.3.2-56
```

scan puertos 20-3000 máquina

```
nmap -p 20-3000 <ip>
```

extraer información ordenadores según puertos abiertos

```
sudo nmap -A -p 1-20000 <ip>
```

host

- devuelve la dirección IP de un dominio (preguntando a los DNS configurados en **/etc/resolv.conf**
- **nslookup** equivalente (y también disponible en Windows)

```
host www.google.com
# responde: 216.58.215.36
```

whois

- pregunta a quien está registrado un dominio

curl

- realiza peticiones (http por defecto) como si fuese un navegador y devuelve por stdout

wget

- descarga (puede que de manera recursiva) una dirección / fichero
- opciones:
    - `-r` : recursive
    - `-l` : nivel de recursividad

```
wget -r -l=2 https://www.pue.es
```

lynx

- navegador desde terminal

extra: robots.txt

```
User-agent: *
Disallow: /bin/
Disallow: /App_Data/

User-agent: All
Allow:/
```

From:
https://matebcn.eu/ - **miguel angel torres egea**

Permanent link:
https://matebcn.eu/info:cursos:pue:lpic2:6:red-avanzada

Last update: **26/09/2018 20:41**