-

UT 10.2
LINUX: NETWORKING

Computer Systems
CFGS DAW

Author: Borja Salom

b.salomsantamaria@edu.gva.es

Reviewed by: Aarón Martín Bermejo

a.martinbermejo@edu.gva.es

2023/2024

Version:240215.1033

Licence

Nomenclature

Throughout this unit different symbols will be used to distinguish important elements within the content. These symbols are:

Importante

Atención

Interesante

ÍNDICE DE CONTENIDO

UD10. LINUX: NETWORKING

1. Linux Networking Commands

Linux networking commands are used extensively to inspect, analyze, maintain, and troubleshoot the network/s connected to the system.

Let us first know the list of the basic networking commands used in Linux followed by a detailed explanation of each.

1.1 Command Ifconfig

Linux ifconfig stands for interface configurator. It is one of the most basic commands used in network inspection.

ifconfig is used to initialize an interface, configure it with an IP address, and enable or disable it. It is also used to display the route and the network interface.

Using `ifconfig`, you can get details of a specific interface. This shown below.

```
ifconfig
```

Using `ifconfig <interface>`, you can get dataisl of specific interface.

```
ifconfig eth0
ifconfig lo
ifconfig wlan0
```

Using `ifconfig <interface> <address> netmask <address>` to assing an IP address and netmask to an interface. Howeverm these details will be reset after the system reboot.

```
ifconfig eth0 172.29.79,70
ifconfig eth0 172.29.79.70 netmask 255.255.0.0
ifconfig eth0 172.29.79.70/16
```

Using `ifconfig <interface> <down | up>` to enable or disable an interface.

```
ifconfig eth0 up
ifconfig eth0 down
```

Using `ifconfig <interface> mtu <number>` to set the size of MTU (Maximum Transmission Unit). By default, MTU has a size of 1500.

```
ifconfig eth0 mtu 1600
```

1.2 Command ip

The ip command is used to assign an address to a network interface and/or configure network interface parameters on Linux operating systems. This command replaces old good and now deprecated ifconfig command on modern Linux distributions.

Using `ip a` or `ip addr` list and show all ip address associated on all network interfaces

```
ip a
ip addr
```

You can select between IPv4 and IPv6 using the following sintax:

```
ip -4 a
ip -6 a
```

It is also possible to specify and list particular interface TCP/IP details:

```
ip a show eth0
ip a list eth0
ip a show dev eth0
```

To assing the IP address to the interface, we can use the command:

```
ip a add <address/mask> dev <interface>
```

```
ip a add 192.168.1.200/255.255.255.0 dev eth0
ip a add 192.168.1.200/24 dev eth0
```

To remove or delete the IP address from the interface we can use the command: `ip a del <address> dev <interface>`

```
ip a del 192.168.1.200/24 dev eth0
```

To change the state of the device to UP or DOWN, we can use this command:

```
ip link set dev <interface> <up | down>
```

```
ip link set dev eth0 up
ip link set dev eth0 down
```

To change the MTU of the device, we can use this command:

```
ip link set mtu <number> dev <interface>
```

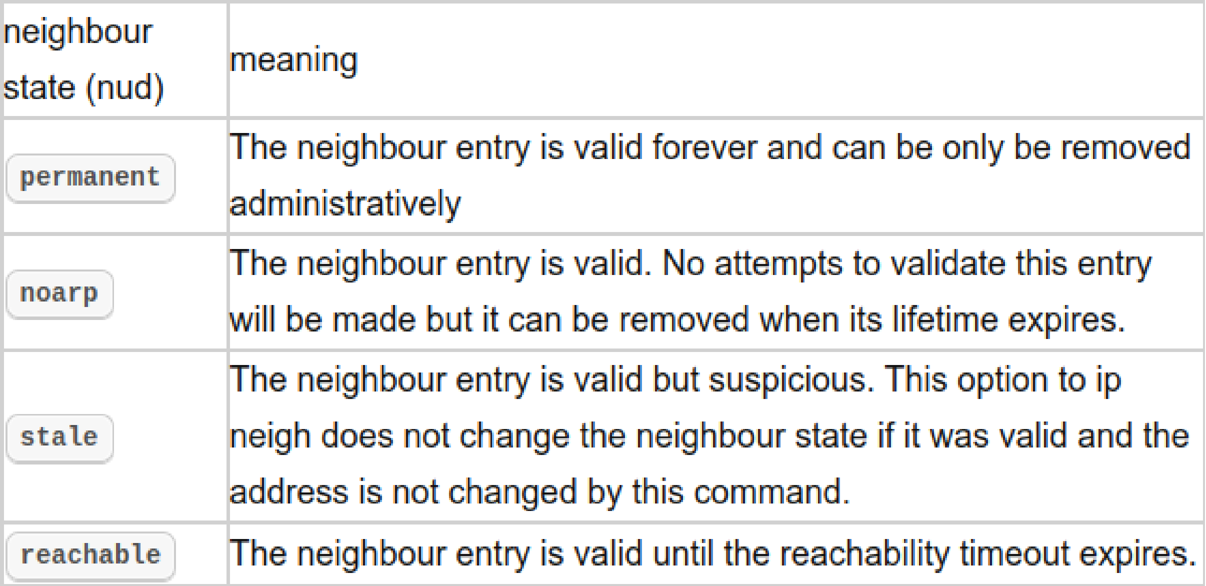
```
ip link set mtu 9000 dev eth0
```

Address Resolution Protocol (ARP) is a procedure for mapping a dynamic IP address to a permanent physical machine address in a local area network. To see the neighbour/arp cache:

```
ip n show
ip neigh show
```

To add a new ARP entry we can use:

```
ip neigh add <IP> lladdr <MAC> dev <interface> nud <state>
ip neigh add 192.168.1.5 lladdr 00:1a:30:38:a8:00 dev eth0 nud perm
```



neighbour state (nud)	meaning
`permanent`	The neighbour entry is valid forever and can be only be removed administratively
`noarp`	The neighbour entry is valid. No attempts to validate this entry will be made but it can be removed when its lifetime expires.
`stale`	The neighbour entry is valid but suspicious. This option to ip neigh does not change the neighbour state if it was valid and the address is not changed by this command.
`reachable`	The neighbour entry is valid until the reachability timeout expires.

To invalidate or delete an ARP:

```
ip neigh del <IP> dev <interface>
```

```
ip neigh del 192.168.1.5 dev eth1
```

Or to change de state

```
ip neigh chg 192.168.1.100 dev eth1 nud reachable
```

To display the contet of the routing tables:

```
ip r
ip r list
ip route list
ip r list [options] ip route
ip r list 192.168.1.0
```

To add a new route;

`ip route add <ip/mask> via <gatewayIP>` or

`ip route add <ip/mask> dev <interface>`

```
ip route add 192.168.1.0/24 via 192.168.1.254
ip route ad 192.168.1.0/24 dev eth0
```

To delete default gateway

```
ip route del default
```

to delete a route:

```
ip route del 192.168.1.0/24 dev eth0
```

1.3 Command ping

Linux ping is one of the most used network troubleshooting commands. It basically checks for the network connectivity between two nodes.

1. ping stands for Packet INternet Groper.
2. The ping command sends the ICMP echo request to check the network connectivity.
3. It keeps executing until it is interrupted.

Use Ctrl+C Key to interrupt the execution.

The syntax is `ping <destinarion>`

```
ping google.com
ping 216.239.38.120
```

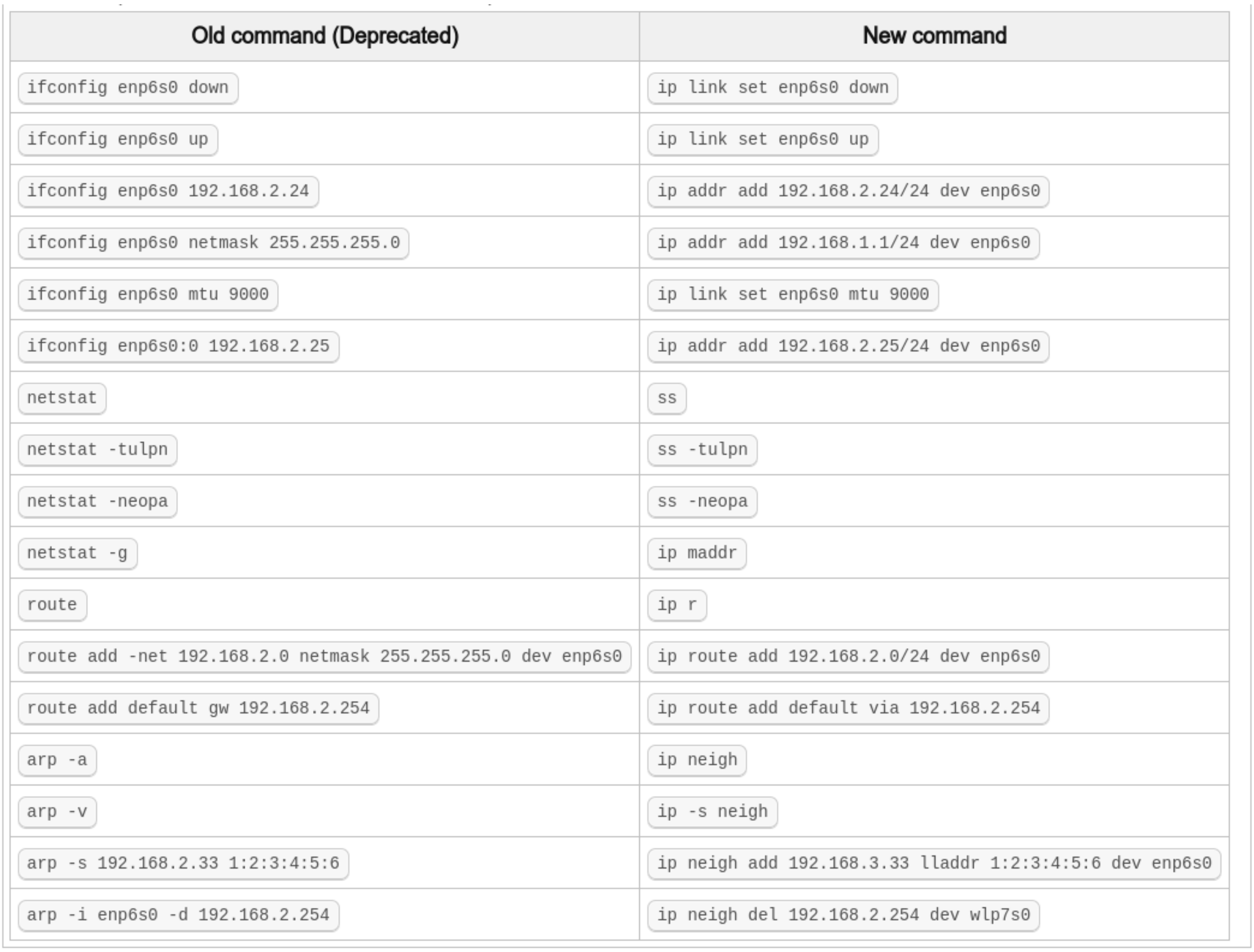
You can limit the number of packets by including "-c" in the ping command.

```
ping -c <number> <destination>
```

The command is used to measure the average response. If there is no response for the ping command, you can assume one of the following issues with the network:

- There is a physical issue causing network loss.
- The destination address might be dysfunctional or incorrect.
- The ping request is blocked due to a target.
- There might be a problem with the routing table.

1.4 Old commands vs new commands



Old command (Deprecated)	New command
`ifconfig enp6s0 down`	`ip link set enp6s0 down`
`ifconfig enp6s0 up`	`ip link set enp6s0 up`
`ifconfig enp6s0 192.168.2.24`	`ip addr add 192.168.2.24/24 dev enp6s0`
`ifconfig enp6s0 netmask 255.255.255.0`	`ip addr add 192.168.1.1/24 dev enp6s0`
`ifconfig enp6s0 mtu 9000`	`ip link set enp6s0 mtu 9000`
`ifconfig enp6s0:0 192.168.2.25`	`ip addr add 192.168.2.25/24 dev enp6s0`
`netstat`	`ss`
`netstat -tulpn`	`ss -tulpn`
`netstat -neopa`	`ss -neopa`
`netstat -g`	`ip maddr`
`route`	`ip r`
`route add -net 192.168.2.0 netmask 255.255.255.0 dev enp6s0`	`ip route add 192.168.2.0/24 dev enp6s0`
`route add default gw 192.168.2.254`	`ip route add default via 192.168.2.254`
`arp -a`	`ip neigh`
`arp -v`	`ip -s neigh`
`arp -s 192.168.2.33 1:2:3:4:5:6`	`ip neigh add 192.168.3.33 lladdr 1:2:3:4:5:6 dev enp6s0`
`arp -i enp6s0 -d 192.168.2.254`	`ip neigh del 192.168.2.254 dev wlp7s0`

2.SSH

SSH, Secure Shell, is a remote administration protocol through which users can both modify and control their remote servers on the Internet. It was created to replace Telnet, a non-encrypted protocol and therefore did not offer any type of security to users.

In return, SSH makes use of the most innovative cryptographic techniques with the clear objective that all communications between users and remote servers are secure. It has a tool that allows to authenticate the remote user to later transfer the entries from the client to the host and, finally, output back to the users.

It is worth noting that users of Linux and MacOS operating systems can implement the SSH protocol on their remote server very easily through the terminal. Of course, Windows users can do it too, although the procedure is different.

2.1 Intall SSH Server

To install an SSH server in Ubuntu it is best to use OpenSSH. One point to keep in mind is that in the vast majority of Linux systems on this server they are already available by default. Therefore, to install it you simply have to give the order to your package manager.

```
sudo apt-get update
sudo apt-get install ssh
```

2.2 SSH Ports

Currently, practically 100% of the servers use Linux as their operating system thanks to the support and stability it offers. That is why cyber attacks against these servers are becoming more frequent. Thus, it is necessary to reinforce security in them in order to avoid any type of unauthorized access.

A great way to improve security on Linux servers is to change the SSH port that the administrator uses to authenticate using the SSH protocol. The truth is that changing the SSH port is a very simple process. We explain it step by step.

1. First you need to edit the ssh_config. To do this you must use the following command: nano /etc/ssh/sshd_config. For what it is necessary that you have installed the command line text editor for Linux.
2. Once the command is executed, look for the line that says “#Port 22”. Thus, what you should do is change 22 to the number of the port that you want to configure. Also, you have to remove the #.
3. Then save all the changes you have made. To do this, press the Control and X keys at the same time.
4. The next step is to restart the SSH service with the following command: /etc/init.d/sshd restart.
5. From that moment you will make all the connections with the port you have chosen.

2.3 SSH Commands

This command offers very secure communication since the data travels encrypted, safe from any type of cyber attack. When you log in to another computer using SSH you must run the following command.

```
ssh <user>@<IP or domain>
```

```
ssh user@192.160.1.1
```

Another command that is worth knowing, because it allows you to move and copy files and files between two computers. It is important to note that it uses SSH to transmit the information, so that it travels encrypted to offer maximum security.

```
scp <filepath> <user>@<IP or Domain>:<destination path>
```

```
scp /tmp/file user@192.160.1.1:/tmp
```

if you want to copy a entire directory you must use the parameter `-r`

```
scp -r /tmp/folder user@192.160.1.1:/tmp
```

3.SFTP

FTP, or “File Transfer Protocol,” was a popular unencrypted method of transferring files between two remote systems.

SFTP, which stands for SSH File Transfer Protocol or Secure File Transfer Protocol, is a separate protocol packaged with SSH that works similarly but over a secure connection. The advantage is the ability to take advantage of a secure connection to transfer files and walk through the file system on both local and remote systems.

In almost all cases, it is preferable to use SFTP, rather than FTP, due to its underlying security features and its ability to take advantage of an SSH connection. FTP is a non-secure protocol that should only be used in limited cases or on trusted networks.

By default, SFTP uses the SSH protocol to authenticate itself and establish a secure connection. Therefore, the same authentication methods are available as in SSH.

If you can connect to the computer using SSH, you have completed all the necessary requirements to use SFTP to manage files.

3.1 Conect to SFTP

We can establish an SFTP session by running the following command:

```
sftp user@192.160.0.1
```

The most useful command to know about first is the help command. This command gives you access to a summary of the help on SFTP. You can invoke it by writing any of these in the statement:

```
help
?
```

We can navigate through the remote system's file hierarchy using various commands that work similarly to their shell counterparts.

3.2 Transfer files with SFTP

If we want to download files from our remote host, we can do it by running the following command:

```
get remoteFile
```

As you can see, by default, the get command downloads a remote file to a file with the same name on the local file system.

We can copy the remote file to a different name by specifying the name after:

```
get remoteFile localFile
```

The get command also takes some option flags. For example, we can copy a directory and all its contents by specifying the recursive option:

```
get -r someDirectory
```

We can tell SFTP to maintain proper permissions and access times by using the -P or -p flag:

```
get -Pr someDirectory
```

Local file transfer to remote system

Transferring files to the remote system is as easy as using the aptly named “put” command:

```
put localFile
```

The same flags that work with `get` apply to `put`. So to copy an entire local directory, you can run `put -r`:

```
put -r localDirectory
```

One familiar tool that is useful when downloading and uploading files is the `df` command, which works similarly to the command line version. Using this, you can check that you have enough space to complete the transfers you are interested in:

```
df -h
```

Any other local command will work as expected. To return to your SFTP session, type:

```
exit
```

3.3 Simple File Manipulations with SFTP

SFTP allows you to perform some kinds of filesystem housekeeping. For instance, you can change the owner of a file on the remote system with:

```
chown userID file
```

Notice how, unlike the system chmod command, the SFTP command does not accept usernames, but instead uses UIDs. Unfortunately, there is no built-in way to know the appropriate UID from within the SFTP interface.

As a workaround, you can read from the /etc/passwd file, which associates usernames with UIDs in most Linux environments:

```
get /etc/passwd
!less passwd
```

The chmod SFTP command works as normal on the remote filesystem:

```
chmod 777 publicFile
```