

# Xarxes

Adrià Valdueza Roca i Enrique Chueca Negre

Octubre 2024



## 1 Objectius de la pràctica

El propòsit d'aquesta pràctica és obtenir experiència pràctica amb els comandaments més útils per gestionar i resoldre problemes en xarxes d'ordinadors. Es treballaran escenaris típics, com la pèrdua de connexió a Internet mentre es navega dins d'una xarxa local. Durant la pràctica, es posarà èmfasi en la identificació de l'adreça IP del router i la comprovació de la connexió mitjançant comandaments bàsics de xarxa.

A més, es revisaran les diferents classes d'adreces IP (tant públiques com privades) i s'introduirà el protocol NAT (Network Address Translation) per comprendre com es converteixen les adreces privades en adreces públiques a la xarxa. Tot i que la pràctica s'executa en l'entorn de Windows, els comandaments treballats són aplicables a altres sistemes operatius com Linux i macOS.

Per finalitzar, en la pràctica es demana que tinguem la capacitat d'identificar i solucionar problemes bàsics de connectivitat en xarxes locals, així com executar i comprendre comandaments de xarxa en diferents sistemes operatius. També s'utilitzaran eines per visualitzar la configuració de la xarxa, gestionar interfícies i verificar connexions de manera efectiva.

## 2 Resposta a les diferents qüestions que en ella es plantegen als alumnes

### 2.1 Exercici 1

P: Quina és la vostra IP?

Utilitzem la comanda **ipconfig /all**

La nostre direccio ip es: Dirección IPv4. . . . . : 10.111.149.153 o per defecte.

Dirección IPv4. . . . . : 192.168.56.1(Preferido)

### 2.2 Exercici 2

Utilitzem la comanda: **netsh interface ip show config**

Ens surt que DHCP està habilitat amb la qual cosa significa que és dinàmica

### 2.3 Exercici 3

Connexió directa: Realitza un ping a la porta d'enllaç o una direcció externa per a comprovar si hi ha connexió. Després desconnectes la xarxa i verifiques que el ping falli.

Fem la comanda Ping a 127.0.0.1: llavors hem vist que encara que estiguem desconnectats de la xarxa, la comanda utilitzada té èxit, ja que es una prova interna del sistema (loopback) que confirma que els protocols i targeta de la xarxa funcionen.

### 2.4 Exercici 4

Useu la comanda **ping www.google.com**

Ens surt el següent output:

Fent ping a www.google.com [142.250.200.132] amb 32 bytes de dades:

Resposta des de 142.250.200.132: bytes=32 temps=19ms TTL=118

Resposta des de 142.250.200.132: bytes=32 temps=23ms TTL=118

Resposta des de 142.250.200.132: bytes=32 temps=16ms TTL=118

Resposta des de 142.250.200.132: bytes=32 temps=15ms TTL=118

### 2.5 Exercici 5

Els datagrames van passar per les següents direccions públiques:

84.88.18.113 (anella-ub2.cesca.cat)

193.242.98.156 (google.02.catnix.net) 192.178.110.73 142.251.51.141 142.250.200.132 (destí final, Google)

El símbol \* no va aparèixer en aquesta traça, la qual cosa indica que tots els salts van respondre correctament.

Aquest és un bon exemple de com els paquets viatgen a través d'encaminadors locals i de la xarxa pública per a arribar als servidors de Google.

## 2.6 Exercici 6

### 1. Què és una direcció MAC?

La direcció MAC (Mitjana Access Control) és un identificador únic assignat a la targeta de xarxa (NIC) d'un dispositiu. Cada dispositiu de xarxa, com la teva computadora, telèfon mòbil, o encaminador, té una adreça MAC única. Està composta per 48 bits (6 grups de dos caràcters hexadecimals, com 00:1A:2B:3C:4D:5E) i és utilitzada en el nivell d'enllaç de dades (capa 2 del model OSI) per a identificar dispositius en una xarxa local, com una xarxa Ethernet o Wi-Fi.

### 2. Puc tenir una adreça MAC sense una adreça IP?

Sí, pots tenir una adreça MAC sense tenir una adreça IP assignada. L'adreça MAC és un identificador físic i sempre és present en la targeta de xarxa, fins i tot si el dispositiu no està connectat a una xarxa o no té una adreça IP assignada.

Això és perquè l'adreça MAC està associada a la capa 2 del model OSI (capa d'enllaç), mentre que l'adreça IP pertany a la capa 3 (capa de xarxa). L'adreça MAC s'utilitza per a identificar dispositius dins d'una xarxa local (LAN), mentre que l'adreça IP s'usa per a comunicar dispositius entre xarxes diferents (per exemple, en Internet).

### 3. Raó per a la resposta:

L'adreça MAC és necessària perquè els dispositius es comuniquin en una xarxa local (com Ethernet o Wi-Fi) mitjançant el protocol ARP (Address Resolution Protocol), que tradueix adreces IP a adreces MAC dins de la xarxa. No obstant això, si un dispositiu no està connectat a una xarxa o no ha rebut una adreça IP (per exemple, per mitjà de DHCP), encara tindrà la seva adreça MAC, però no tindrà una adreça IP.

## 2.7 Exercici 7

### 1- Comanda ARP inicial:

Inicialment, vaig utilitzar el comando `arp -a` per a visualitzar la taula ARP en el meu sistema. En aquesta taula, es mostraven les associacions entre les adreces IP i les adreces MAC, tant estàtiques com dinàmiques. Per exemple, una de les entrades importants va ser la de la porta d'enllaç predeterminada (encaminador), que tenia l'adreça IP 10.133.255.254 i l'adreça MAC 00-08-e3-ff-fc-50.

### 2- Eliminació de la taula ARP:

Posteriorment, per a eliminar totes les entrades de la taula ARP, vaig executar el comando: `arp -d *`

Això va ser necessari fer-ho des d'una consola amb privilegis d'administrador. Després d'executar aquest comando, totes les entrades en la taula ARP van ser eliminades.

### 3- Reconstrucció de la taula ARP:

Després d'eliminar la taula ARP, vaig realitzar un ping a la porta d'enllaç predeterminada (encaminador) utilitzant el següent comando: `ping 10.133.255.254`

Això va forçar a la meua computadora a enviar una sol·licitud ARP per a obtenir novament l'adreça MAC associada amb l'adreça IP de l'encaminador. Després d'això, vaig tornar a executar el comando `arp -a` per a verificar que la taula ARP s'havia reconstruït.

Com a resultat, l'entrada corresponent a la direcció 10.133.255.254 va ser afegida de nou a la taula ARP, juntament amb la seva adreça MAC 00-08-e3-ff-fc-50.

## 2.8 Exercici 8

La comanda `netstat -r` mostra la taula d'encaminament, que conté les rutes que utilitza el teu sistema per a enviar paquets a diferents destins en la xarxa.

**Mètrica:** La mètrica és el cost o prioritat associat a cada ruta. Quant menor sigui la mètrica, més alta és la prioritat d'aquesta ruta per a ser utilitzada.

## 2.9 Exercici 9

Després d'habilitar l'us del Client Telnet, hem executat la següent comanda:

```
telnet time-A.timefreq.bldrdoc.gov 13
```

El output que ens surt és el següent:

```
60601 24-10-18 15:48:12 17 0 0 51.8 UTC(NIST) * S'ha perdut la connexió amb el host.
```

## 2.10 Exercici 10

Després d'habilitar l'us del Client Telnet, hem executat la següent comanda:

```
telnet time-A.timefreq.bldrdoc.gov 13
```

El output que ens surt es el següent: `60601 24-10-18 15:48:12 17 0 0 51.8 UTC(NIST) * S'ha perdut la connexió amb el host.`

- Ens hem connectat al servidor de temps del NIST en Boulder, Colorado.

## 2.11 Exercici 11

Ens apareix a l'executar la comanda : `ssh -X usuariHost@IPhost` :

Connection closed by 161.116.52.69 port 22

## 2.12 Exercici 12

Primer obtenim l'adreça IP de l'equip remot:

Primer, en l'equip Linux al qual em vull connectar de manera remota, vaig utilitzar el comando `ip addr xou` per a obtenir la seva adreça IP. L'adreça IP assignada a l'equip era 161.116.52.69. També vaig verificar que el servei SSH estava actiu en l'equip remot, assegurant-me que aquest permetés connexions remotes. Connexió des de l'equip principal:

En el meu equip principal, vaig obrir una terminal i vaig executar la següent comanda:

```
ssh -X echuecne7@161.116.52.69
```

Aquí, `echuecne7` és el nom d'usuari en l'equip remot. 161.116.52.69 és l'adreça IP de l'equip remot que vaig obtenir prèviament. Autenticació reeixida:

Després d'executar el comando, el sistema em va demanar la contrasenya del meu compte d'usuari en l'equip remot (`echuecne7`). Vaig ingressar la contrasenya i es va establir la connexió sense problemes. Execució d'una aplicació remota:

Una vegada connectat a l'equip remot, vaig utilitzar la terminal per a executar una aplicació gràfica remota en l'equip Linux. Per exemple, per a obrir Firefox en l'equip remot, vaig executar el següent comando: `firefox`

El navegador Firefox es va executar en l'equip remot, però la finestra del navegador va aparèixer en el meu equip local gràcies a la reexpedició X11 que havia activat amb l'opció -X.

### Finalització de la sessió SSH:

Després de realitzar les proves, vaig tancar la sessió SSH escrivint la comanda: `exit`. La connexió amb l'equip remot es va tancar correctament, finalitzant la sessió.

## 2.13 Exercici 13

Activació del servidor i client de FTP donis de MobaXterm: Hem connectat amb el servidor FTP `ftp.rediris.es` com a usuari anònim, descarregant fitxers amb els comandis `get` o `mget`.

Comanda per veure la configuració de la xarxa: `ip address xou` mostren la configuració de la xarxa. Tots dos permeten veure informació dels interfícies de xarxa. Lynx: Hem instal·lat Lynx i hem utilitzat `lynx http://www.ub.edu` per navegar per la pàgina de la UB en text. La comanda `lynx -dump` permet obtenir ràpidament el contingut de la pàgina, útil en entorns amb baixa connexió o sense interfície gràfica.

## 3 Explicació de la feina realitzada al laboratori

Al laboratori hem fet el codi per la pregunta numero 13 que deixem aquí sota (tenir en consideració que potser no es veu ben indentat per l'ús del latex)

Codi pregunta 13: `import socket import threading`

```
clients = [] aliases = []
```

```
def broadcast(message): for client in clients: client.send(message)
```

```
def handle_client(client): while True: try: message = client.recv(1024) broadcast(message) except: index = clients.index(client) clients.remove(client) client.close() alias = aliases[index] broadcast(f'alias has left the chat.'.encode('utf-8')) aliases.remove(alias) break
```

```
def receive(): server = socket.socket(socket.AF_INET, socket.SOCK_STREAM) server.bind(('192.168.0.19', 55555))
```

```
server.listen()
```

```
print('Server is listening...')
```

```
while True:
```

```
client, address = server.accept() print(f'Connection from {str(address)}')
```

```
client.send('NICK'.encode('utf-8')) alias = client.recv(1024).decode('utf-8') aliases.append(alias) clients.append(client)
```

```
print(f'The alias of this client is {alias}') broadcast(f'alias has joined the chat!'.encode('utf-8')) client.send("You are connected to the chat!".encode('utf-8'))
```

```
thread = threading.Thread(target=handle_client, args=(client,)) thread.start()
```

```
receive()
```

```
import socket import threading
```

```
def receive(): while True: try: message = client.recv(1024).decode('utf-8') if message == 'NICK':
```

```

client.send(alias.encode('utf-8')) else: print(message) except: print("An error occurred!") client.close() break
def write(): while True: message = f'alias: {input("")}' client.send(message.encode('utf-8'))
alias = input("Choose an alias: ")
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM) client.connect(('192.168.0.19', 55555))
alias = input("Choose an alias: ")
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM) client.connect(('192.168.0.19', 55555))
receive_thread = threading.Thread(target=receive) receive_thread.start()
write_thread = threading.Thread(target=write) write_thread.start()

```

## 4 Conclusions

En aquesta pràctica, hem explorat comandis essencials per al diagnòstic de xarxes d'ordinadors, com llaura ip address xou, ping, traceroute i arp. Hem identificat la nostra adreça IP i diferenciat entre adreces públiques i privades, a més d'entendre com funciona el protocol NAT per a la traducció d'adreces.

Hem verificat la connectivitat a Internet amb ping i hem analitzat els salts de dades amb traceroute, observant la latència en cada salt. També hem aprofundit en la importància dels adreces MAC i en com el protocol ARP ressol els adreces IP en adreces MAC.

A més, hem explorat connexions remalnomns utilitzant SSH i Telnet, així com l'ús de FTP i navegadors de text com Lynx, demostrant la seva utilitat en la gestió de xarxes.

En resum, aquesta pràctica ens ha donat una comprensió fonamental sobre la infraestructura de xarxes i els eines necessàries per gestionar-els. Finalment, hem après a establir un xat entre dos ordinadors connectats dins de la mateixa xarxa.