Facultat Informàtica de Barcelona
FIB

Laboratori de Xarxes de Computadors del Grau en Enginyeria Informàtica (XC-grau)

Llorenç Cerdà Alabern, José M. Barceló Ordinas,
David Carrera i Leandro Navarro

Febrer de 2025

Índex

Entorn del laboratori (imatge “xarxes”)

En aquest capítol introductori hi ha una descripció general de la configuració de l’entorn que es farà servir per fer les pràctiques de laboratori. Al botar el PC s’ha de seleccionar la imatge “xarxes”. Aquesta imatge s’ha confeccionat a partir de la distribució de Linux de mida reduïda anomenada Alpine (http://alpinelinux.org).

1. Informació bàsica

Usuari i password: xc / xc

Superusuari i password: root / root

El funcionament habitual és obrir la sessió com a usuari "xc" i en la consola canviar a root si ho necessiten.

Els icones de les aplicacions que es faran servir habitualment estan a la part de sota de l'escriptori:

Aquestes són, per ordre des de l’esquerra: consola, navegador web, wireshark, calculadora i editor.

Per configurar el PC per DHCP cal executar la següent comanda com a superusuari. Això és necessari per poder accedir al servidor pclabxc per fer els minicontrols.

```
# udhcpc -i e0
```

2. Interfícies dels PCs

Per a fer les pràctiques de xarxes utilitzareu els següents ports de comunicacions dels PCs (vegeu la Figura 1):

- `ttyS0` (COM1 en windows): Aquí hi connectareu la consola per poder configurar els routers i commutadors CISCO.
- `e0`, `e1`, `e2`: son tres targetes ethernet. El sistema operatiu dóna els noms `eth0`, `eth1`, `eth2` a aquestes targetes. Hi ha el problema, però, que la posició física de la targeta amb el mateix nom pot canviar d’un PC a un altre. Perquè la posició de les targetes es correspongui amb la seva posició física en tots els PCs, les imatges fan servir la comanda ifrename/iftab en la fase de boot. Amb aquesta comanda s’anomenen les interfícies `eth0`, `eth1`, `eth2` amb els noms `e0`, `e1`, `e2`, de forma que quedin en les posicions que indica la Figura 1. Tenir en compte, doncs, que tot i que en alguns punts d’aquest manual es fan servir els noms per defecte (`eth0`, `eth1`, `eth2`), cal fer servir els noms `e0`, `e1`, `e2` segons la targeta que es faci servir (que podem identificar per la seva posició física en el PC).

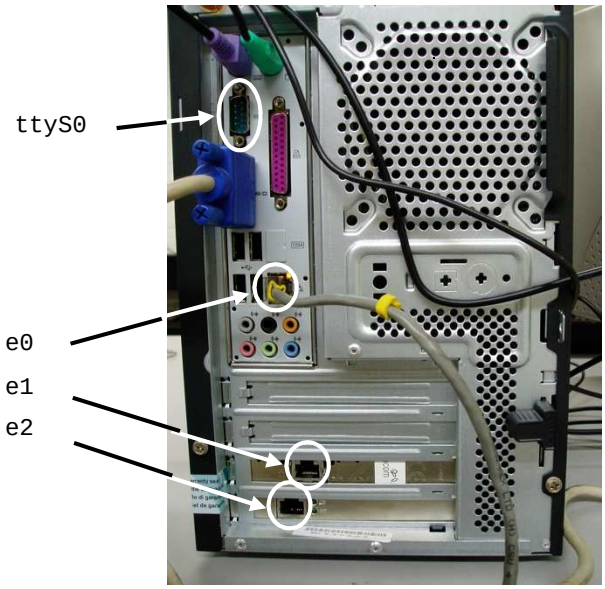


Figura 1: Ports de comunicació dels PCs del laboratori que farem servir en les pràctiques.

2.1. Identificació del nom de les interfícies ethernet

En alguns PCs del laboratori s'ha canviat alguna targeta ethernet que havia deixat de funcionar, i al botar el nom ja no es correspon amb e0, e1 i e2, tal com s'ha descrit anteriorment. A continuació s'explica un mètode senzill per determinar el nom de les interfícies, i quina és la seva ubicació física en el PC.

Primer cal determinar el nom que ha assignat Linux al botar. Per això basta executar "ifconfig –a", tal com es motra a continuació:

```
xc# ifconfig –a
eth3        Link encap:Ethernet  HWaddr 08:00:27:4E:4C:C7
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:10 Base address:0xd020

eth1        Link encap:Ethernet  HWaddr 08:00:27:BE:7D:7F
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:9 Base address:0xd240

eth4        Link encap:Ethernet  HWaddr 08:00:27:5A:83:86
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:11 Base address:0xd260
```

Del bolcat podem veure que, en aquest exemple, el nom de les interfícies és eth3, eth1 i eth4. Ara queda determinar quina és la posició física en el PC. Per fer-ho tindrem en compte que al cable que connecta el PC a la xarxa del laboratori arriben contínuament packets del switch on està connectat. Per tant, només hem de capturar paquets amb tcpdump, si n'arriba algun, vol dir que estem fent la captura en la interfície on hi ha connectat el cable de xarxa. Per exemple, per determinar si el cable de xarxa està connectat en eth3 executariem:

```
xc# ifconfig eth3 up
xc# tcpdump –ni eth3
20:00:14.229940 STP 802.1d, Config, Flags [none], bridge-id
833e.00:11:5c:05:f5:40.8004, length 43
20:00:14.516673 STP 802.1d, Config, Flags [none], bridge-id
8004.00:11:5c:05:f5:40.8004, length 43
^C
```

Del bolcat anterior podem verure que efectivament el cable està en eth3. A continuació desconnectaríem el cable, el connectaríem en una altra targeta, i repetiríem les commandes anteriors amb el nom d'una altra interfície, per exemple eth1. Si arriba tràfic, vol dir que la targeta on està el cable és eth1, i la que queda seria eth4.

Eines per repassar les pràctiques

Totes les pràctiques que es fan en les sessions presencials de laboratori es poden fer també a casa amb les eines que s'expliquen a continuació. És convenient fer-les també a casa si després de la sessió presencial de laboratori queden dubtes o no s'ha tingut temps d'acabar la pràctica.

1. Per a les pàctiques amb els PCs

Són les pràctiques: 1 Configuració LINUX, 6 TCP, 7 DNS

En el següents enllaços podeu trobar 2 màquines virtuals (MV) creades des de VirtualBox (https://www.virtualbox.org) on hi ha instal·lat un Linux amb el mateix programari que la imatge que teniu en el laboratori. Hi ha 2 distribucions. La distribució Slitaz és més compacta i fa servir menys recursos. Es feia servir anteriorment per fer les pràctiques, però s'ha canviat per la distribució Alpine perquè es varen canviar els PCs i Slitaz no detectava les targetes ethernet.

Distribució Alpine: https://studies.ac.upc.edu/FIB/grau/XC/alpine-xarxes.ova

Distribució Slitaz: https://studies.ac.upc.edu/FIB/grau/XC/ slitaz50-xarxes.ova

Per importar-la des de VirtualBox:

Fitxer → Importar màquina virtual

Per a tenir múltiples VMs, clonar la imatge amb virtualbox tantes vegades com faci falta:

Seleccionar la imatge → clone → MARCAR L'OPCIÓ: "reinitialize the mac address of all network cards" → Linked clone

Podeu crear una xarxa de MVs i connectar-les per a repasar la pràctica del laboratori.

Veureu que la MV està configurada amb 4 targetes ethernet. Pequè dues MVs tinguin una targeta en la mateixa xarxa, cal anar a paràmetres → xarxa → nom, i posar el mateix nom en les dues MVs.

2. Per a les pràctiques amb IOS

Són les pràctiques: 2 Configuració IOS, 3 RIP, 4 ACL i NAT, 5 Switches

En el següent enllaç us podeu descarregar el simulador packettracer de CISCO. Només us heu de registrar per poder descarregar-vos el simulador sense cost.

https://www.netacad.com/about-networking-academy/packet-tracer/

El model del routers que hi ha en els racks és 1841, els commutadors són 2950.

Lab 1. Comandes bàsiques per a la configuració del nivell IP amb UNIX

1. La interfície *loopback*

La primera interfície que convé activar al configurar el nivell IP és la interfície *loopback*. Aquesta interfície és una mena de curtcircuit, és a dir, els datagrames que s'envien en aquesta interfície no abandonen mai la màquina, sinó que retornen immediatament al nivell IP que els envia. Així doncs, el *loopback* es fa servir en la comunicació entre processos amb TCP/IP dintre de la mateixa màquina. L'adreça de xarxa assignada al *loopback* és `127.0.0.0`. A la interfície típicament se li assigna l'adreça `127.0.0.1.` El nom que fa servir linux per aquesta interfície és `lo`. A més, amb linux típicament s'assigna el nom `localhost` a l'adreça `127.0.0.1`

El Linux que teniu configura automàticament la interfície de *loopback.*

2. El fitxer `/etc/hosts`

Per no haver d'usar sempre les adreces IP, una màquina UNIX permet assignar noms a les adreces IP amb el fitxer `/etc/hosts`. Per exemple, el contingut d'aquests fitxer podria ser:

```
xc# cat /etc/hosts
127.0.0.1                   localhost
192.168.60.112              linux
192.168.60.101              pc1
```

Exemple 1: Contingut del fitxer `/etc/hosts`.

Quan es dóna un nom en comptes d'una adreça IP a una comanda, aquesta fa una crida al *resolver* del sistema. El *resolver* mira primer el fitxer `/etc/hosts` per fer la resolució. Si el nom no hi és, aleshores mira si en el fitxer`/etc/resolv.conf` hi ha l'adreça d'algun servidor de noms. En cas afirmatiu, farà servir el protocol DNS (RFC1035) per sol·licitar la resolució del nom al servidor.

3. IP *forwarding*

El mecanisme de IP *forwarding* consisteix en la transmissió d'un paquet rebut per una de les interfícies físiques d'un node (un *host* o un *router*) per una altra interfície física (que pot ser la mateixa). El funcionament és el següent: El mòdul IP té una funció que processa els paquets que s'han de transmetre (`ip_output`) i una que processa els paquets que es reben (`ip_input`), tal com mostra la figura . Si la funcionalitat de IP *forwarding* no està activada, la funció `ip_input` descarta tots els paquets que no tinguin com a destinació alguna de les interfícies del node. Per contra, si el node té el IP *forwarding* activat, `ip_input` passa a `ip_output` tots els paquets que es reben i que no tenen com a destinatari el mateix node.

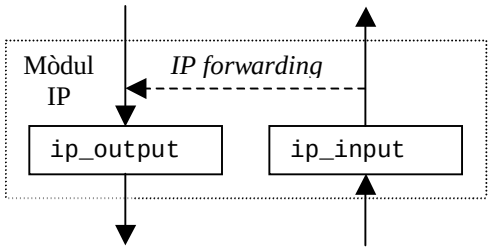


Figura 2: Funcions `ip_output` i `ip_input`.

Un *router* té el IP *forwarding* activat per defecte, donat que la seva funció és la d'encaminar paquets IP. Un *host*, en canvi, normalment no té aquesta funcionalitat activada. En linux el kernel es pot compilar perquè tingui la funcionalitat de IP *forwarding* amb la següent opció:

```
IP forwarding/gatewaying (CONFIG_IP_FORWARD) [n] y
```

Per activar-la, cal a més que algunes variables del kernel tinguin un valor diferent de zero. Podeu veure el valor d'aquestes variables amb la comanda que mostra el següent exemple:

```
xc# sysctl -a | egrep forward
net.ipv6.conf.default.forwarding = 0
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.eth2.forwarding = 1
net.ipv6.conf.eth1.forwarding = 1
net.ipv6.conf.eth0.forwarding = 1
net.ipv6.conf.lo.forwarding = 1
net.ipv4.conf.lo.mc_forwarding = 0
net.ipv4.conf.lo.forwarding = 1
net.ipv4.conf.default.mc_forwarding = 0
net.ipv4.conf.default.forwarding = 1
net.ipv4.conf.all.mc_forwarding = 0
net.ipv4.conf.all.forwarding = 1
net.ipv4.ip_forward = 1
```

Exemple 2: Variables de forwarding del kernel.

La imatge xarxes ja té activat l'IP *forwarding.* Altrament es pot activar executant:

```
root# sysctl –w ip_forward=1
```

4. Comandes bàsiques

En aquesta secció es descriuen les comandes bàsiques per a la configuració d'una màquina UNIX. La descripció que es fa a continuació es correspon amb les comandes que hi ha en la imatge *linux-xarxes* que s'ha preparat per fer les pràctiques. Els paràmetres d'aquestes comandes o el seu comportament pot canviar lleugerament en altres UNIXs, en Windows, o fins i tot en altres distribucions de linux.

4.1. Comanda `ifconfig`

Permet configurar una interfície. Les maneres típiques d'invocar aquesta comanda són:

```
ifconfig interfície adreça_IP [netmask màscara] [broadcast @broadcast]
```

[1]

Comanda 1: Assignació d'una adreça IP i activació d'una interfície.

On [] vol dir paràmetre opcional. Activa una interfície i l'hi assigna una adreça. Si no es dóna la màscara, s'assigna la que correspon segons la classe de l'adreça IP, si no es dóna l'adreça de broadcast el SO calcula la que correspon a la màscara.. Per designar una targeta *ethernet*, Linux fa servir el nom `ethi`, on `i` val `0` per la primera targeta, `1` per la segona etc. Els noms els assigna el kernel automàticament a mesura que carrega amb èxit el driver de cada targeta. Recordar però que les interfícies es reanomenen per ei.

Per exemple, per assignar una adreça IP i activar una targeta *ethernet*:

```
xc# ifconfig e0 10.0.0.1 netmask 255.255.255.0
```

Exemple 3: Configuració de la interfície `e0`.

Si volem desactivar una interfície (p.e. e0) hem d'executar:

```
Ifconfig e0 0.0.0.0
ifconfig e0 down
```

Comanda 2: Desactivació d'una interfície. Al assignar la IP 0.0.0.0 a la interfície s'esborra la IP que tenia assignada prèviament.

I per activar-la de nou:

```
ifconfig e0 up
```

Comanda 3: Activació d'una interfície.

Per mostrar les interfícies actives hem d'executar la comanda sense paràmetres, com mostra el següent exemple:

[1] Farem server el següent conveni: les paraules clau estan en negreta i els paràmetres que dóna l'usuari no.

```
xc# ifconfig
e0      Link encap:Ethernet  HWaddr 00:40:F4:65:E6:BE
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::240:f4ff:fe65:e6be/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75 errors:0 dropped:0 overruns:0 frame:0
          TX packets:213 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:7236 (7.0 Kb)  TX bytes:86584 (84.5 Kb)
          Interrupt:10 Base address:0xbe00
...
```

Exemple 4: Llistat de les interfícies configurades.

Si volem llistar les interfícies conegudes pel kernel (actives o no) hem d'executar:

```
ifconfig –a
```

Comanda 4: Llistat de les interfícies conegudes pel kernel.

4.2. Comanda `route`

Permet afegir/esborrar entrades a la taula d'encaminament i mostrar el seu contingut. Les invocacions típiques són:

```
route add|del -net|-host destinació [netmask màscara] [gw gateway] [dev intf.]
```

Comanda 5: Us de la comanda route.

On | vol dir paràmetres alternatius i [] vol dir paràmetre opcional. Si no es dóna la màscara i el SO assigna la de la classe. Si no es dóna la interfície, el SO mira de deduir-la de les adreces que s'han assignat. El gateway només ha de donar-se si la xarxa destinació no està directament connectada a una de les interfícies.

```
route [-n]
```

Comanda 6: Llistat de la taula d'encaminament.

Mostra el contingut de la taula d'encaminament.

Amb l'opció `-n` mostra les adreces IP en forma numèrica. Per exemple:

```
xc# route –n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.60.0     0.0.0.0         255.255.255.0   U     0      0        0 e0
```

Exemple 5: Llistat de la taula d'encaminament.

La ruta per defecte té l'adreça 0.0.0.0 i màscara 0.0.0.0. Si el gateway de la ruta per defecte és, per exemple, 192.168.1.1, per afegir la ruta per defecte es pot fer d'una de les següents maneres

```
xc# route add -net 0.0.0.0 netmask 0.0.0.0 gw 192.168.1.1
xc# route add [–net] default gw 192.168.1.1
```

Exemple 6: Addició de la ruta per defecte.

Fixeu-vos que la paraula clau default, aquí equival a posar "-net 0.0.0.0 netmask 0.0.0.0".

4.3. Comanda `arp`

La comanda `arp` permet veure i modificar manualment la taula que manté el mòdul ARP (*Address Ressolution Protocol*). En aquesta taula hi ha la correspondència entre les adreces IP i les adreces *hardware*. Les invocacions típiques són:

```
arp
```

Comanda 7: Mostra la taula `ARP`.

```
arp –s adreça_IP adreça_hw
```

Comanda 8: Assigna l'adreça *hardware* `adreça_hw` a l'adreça `IP adreça_IP`.

```
arp –d adreça_IP
```

Comanda 9: Esborra l'entrada `adreça_IP` de la taula.

4.4. Comanda `ping`

La comanda `ping` és una mena de sonar que permet verificar si una certa interfície està a l'abast del nivell de xarxa, i per mesurar el retard d'anada i tornada que hi ha fins el destí. *Ping* envia periòdicament un paquet a l'adreça que es dóna com a paràmetre que provoca la resposta de la destinació. Per parar el *ping* s'ha de fer un `CONTROL-C`. Per exemple, per saber si podem accedir a la màquina `192.168.60.200`:

```
xc# ping 192.168.60.200
PING 192.168.60.200 (192.168.60.200): 56 data bytes
64 bytes from 192.168.60.200: icmp_seq=0 ttl=255 time=0.6 ms
64 bytes from 192.168.60.200: icmp_seq=1 ttl=255 time=0.6 ms
64 bytes from 192.168.60.200: icmp_seq=2 ttl=255 time=0.6 ms
^C
--- 192.168.60.200 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.6/0.6/0.6 ms
```

Exemple 7: Ping a una màquina remota.

També podem fer un *ping* a una interfície de la mateixa màquina, per exemple al *loopback*:

```
xc# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.1 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.0 ms
^C
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

Exemple 8: Ping a la interfície de *loopback*.

A l'Exemple 9 es fa un *ping broadcast* (l'adreça de *broadcast* és la que té el camp de *host* amb tots els bits a 1). Amb aquest ping podrem saber quines altres màquines hi ha connectades a la mateixa xarxa.

```
xc# ping 192.168.60.255
PING 192.168.60.255 (192.168.60.255): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.1 ms
64 bytes from 192.168.60.7: icmp_seq=0 ttl=255 time=0.6 ms (DUP!)
64 bytes from 192.168.60.3: icmp_seq=0 ttl=255 time=0.7 ms (DUP!)
...
^C
--- 192.168.60.255 ping statistics ---
1 packets transmitted, 1 packets received, +12 duplicates, 0% packet loss
round-trip min/avg/max = 0.1/1.9/7.0 ms
```

Exemple 9: Ping *broadcast*.

Els `DUPs` indiquen que s'ha rebut més d'un paquet de resposta a un mateix *ping*.

4.5. Comanda traceroute

Traceroute permet saber els routers que travessa un paquet fins a la destinació. Per a saber-ho traceroute envia seqüències de tres paquets UDP a un port arbitrari (major de 30.000) on es poc probable que hi hagi cap procés escoltant que retorni una resposta. Cada seqüència s'envia amb un TTL que es va incrementant a partir del valor 1 fins que s'arriba a la destinació. D'aquesta manera, els tres primers paquets (que s'envien amb TTL = 1) els descarta el primer router, el qual retorna un missatge ICMP d'error del tipus "TTL = 0 during transit" per cada paquet. Això mateix passarà en els següents routers fins que la seqüència de tres paquets arribi a la destinació. En aquest cas, la destinació descartarà els tres paquets (perquè no hi ha cap procés escoltant el port on van dirigits) i en conseqüència es generaran tres missatges ICMP d'error del tipus "port unreachable".

El següent exemple mostra dos exemples de l'execució de `traceroute`. En el primer cas es fa un traceroute a una màquina de la mateixa xarxa (que es diu beco). En el segon cas es fa a una màquina (rogent) que està en una xarxa diferent, però només ha de passar per 1 router (de nom arenys5). El temps que mostra la sortida de traceroute és el temps que passa des de que s'envia cada un dels tres paquets, fins que es reben les respectives respostes. Si un paquet es perd, traceroute mostra un asterisc.

```
xc# traceroute beco
traceroute to beco.ac.upc.es (147.83.35.81), 30 hops max, 40 byte packets
 1  beco (147.83.35.81)  1.747 ms  0.551 ms  0.531 ms

xc# traceroute rogent
traceroute to rogent.ac.upc.es (147.83.31.7), 30 hops max, 40 byte packets
 1  arenys5 (147.83.35.2)  0.918 ms  0.840 ms  0.762 ms
 2  rogent (147.83.31.7)  0.591 ms *  0.537 ms
```

Exemple 10: `traceroute`.

4.6. Comanda `tcpdump`

La comanda tcpdump permet capturar els paquets que arriben o s'envien des d'una interfície. Per exemple:

```
xc# tcpdump -ni e0
tcpdump: listening on e0
16:14:58.430994 arp who-has 10.0.0.2 tell 10.0.0.1
16:14:58.431080 arp reply 10.0.0.2 is-at 0:40:f4:65:e6:be
16:14:58.431150 10.0.0.1 > 10.0.0.2: icmp: echo request (DF)
16:14:59.430026 10.0.0.1 > 10.0.0.2: icmp: echo request (DF)
16:15:00.430034 10.0.0.1 > 10.0.0.2: icmp: echo request (DF)
^C
```

Exemple 11: `tcpdump`.

En aquest exemple, l'opció `-n` vol dir que no es vol fer la resolució de noms (altrament tcpdump crida al *resolver* del SO i es queda esperant uns segons perquè respongui). L'opció `-i` permet especificar la interfície que volem escoltar. A continuació, tcpdump imprimeix una línia per cada paquet que rep o transmet. Cada línia comença amb l'instant de captura del paquet (en el format: hores:minuts:segons), seguit de l'adreça IP font i destinació (si és un datagrama IP), i altra informació relativa al paquet que ha capturat. En l'exemple anterior es mostren els paquets que es capturen després de fer un ping. En una sessió posterior del laboratori estudiarem amb més detall aquesta comanda.

5. Realització de la pràctica

5.1. Primera part: configuració d'un host

L'objectiu de la practica és la configuració de la xarxa del laboratori tal com es mostra en la Figura 3.

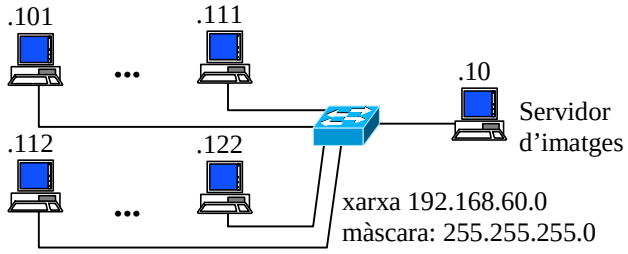


Figura 3: Xarxa del laboratori D6003.

És a dir, hi haurà una xarxa IP `192.168.60.0`. El servidor d'imatges està connectat a la xarxa del laboratori però està situat en una altra sala. Per assignar les adreces IP als PC farem servir el següent conveni: `192.168.60.<número PC>`. Per exemple, si el número que hi ha en l'etiqueta del PC és 3, el PC tindrà com adreça `192.168.60.103`. Per fer aquesta primera part seguiu el següents passos:

1) Llisteu les interfícies (igual que en l'Exemple 4) per comprovar que només hi ha la interfície `lo` configurada. Llisteu la taula d'encaminament (Exemple 5) per comprovar que la taula està buida. Llisteu la taula ARP (Comanda 7) per comprovar que també està buida.
2) Proveu de fer un *ping* al `127.0.0.1`. Comprovareu que el mateix PC contesta.
3) Proveu de fer un *ping* a l'adreça broadcast `192.168.60.255`. Comprovareu que la xarxa és inaccessible.
4) Assigneu l'adreça IP a la targeta *ethernet* `e0` fent servir el conveni explicat anteriorment. Comproveu que la interfície s'ha activat llistant les interfícies. Comproveu que linux ha afegit l'entrada a la taula d'encaminament que permet accedir a la xarxa de la que penja la targeta *ethernet*. Si no fos així, feu servir la comanda `route` per afegir-la.
5) Proveu de fer un *ping* a la targeta *ethernet* per assegurar-vos de que és accessible.
6) Feu un *ping broadcast* per descobrir quines altres màquines hi ha connectades a la xarxa. Llisteu la taula ARP per veure les adreces *hardware* d'aquestes màquines.
7) Afegiu l'entrada "`192.168.60.x pcx`" al fitxer `/etc/hosts`, on x correspon a la IP de una dels PCs que ha

contstat al ping broadcast. Ho podeu fer amb l'editor `vi`, `leafpad` (usuari root) o simplement executant: "`echo 192.168.60.x pcx >> /etc/hosts`".

8) Proveu de fer "`ping pcx`" per comprovar que la màquina és accessible.

5.2. Segona part: configuració d'un router linux

L'objectiu és configurar un PC com a *router* per a poder comunicar PCs situats en xarxes diferents, tal com mostra la Figura 4.

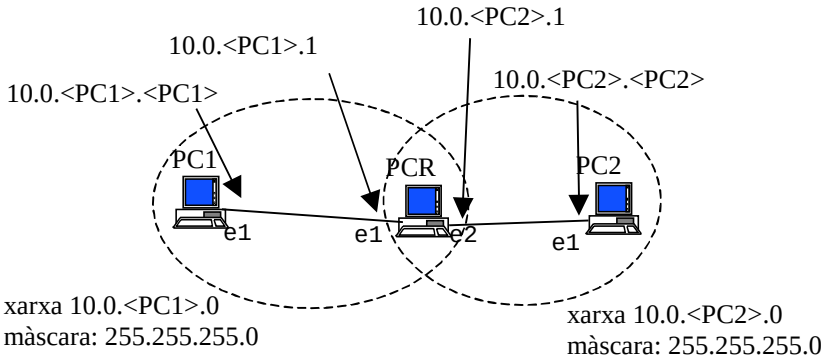


Figura 4: Topologia amb 1 router.

Fixeu-vos que per fer aquesta part necessitareu tres PCs. Així doncs, segons els nombre de PCs lliures que hi hagi podeu ajuntar-vos dos o tres de vosaltres. Apunteu les adreces IP configurades en la següent taula:

PC1/e1	
PCR/e1	
PCR/e2	
PC2/e1	

Per fer la configuració que es demana seguiu els passos següents:

9) Connecteu les interfícies que mostra la figura amb dos cables creuats.
10) Desactiveu la interfície que heu fet servir en l'apartat anterior (Comanda 2) i comproveu amb `ifconfig` que la interfície no està activa. Comproveu amb `route` que l'entrada de la taula d'encaminament que la feia servir s'ha esborrat.
11) Un dels PCs (a partir d'ara l'anomenarem PC1) ha d'estar configurat perquè estigui en la xarxa `10.0.<PC1>.0/24` amb un *hostid* igual a `<PC1>` (on PC1 és el número del PC).
12) Configureu un altra PC (a partir d'ara l'anomenarem PC2) perquè estigui en la xarxa `10.0.<PC2>.0/24` amb un *hostid* igual a `<PC2>`.
13) Configureu el tercer PC perquè faci de *router* entre les dues xarxes (a partir d'ara l'anomenarem PCR) assignant un *hostid* igual a `<PCR>` a les interfícies. La configuració d'un PC com a *router* és exactament la mateixa que la que es faria si no ho fos. L'única diferència és que en el cas del *router* hi haurà més d'una interfície (una per cada xarxa a la que està connectat).
14) Feu un *ping* des del PC1 al PCR per comprovar que és accessible. Feu un *ping* des del PC2 al PCR per comprovar que és accessible. Si no hi ha connectivitat és possible que el cable no estigui connectat en la interfície correcta. Per exemple, si la targeta on heu connectat el cable en PC1 és la que linux identifica amb `e2` i heu configurat `e1`, PCR no rebrà els paquets que envia PC1. En aquest cas, feu servir `tcpdump` i `ping` per descobrir a quina targeta física correspon cada interfície.
15) Si feu un *ping* des del PC1 al PC2 comprovareu que no es poden comunicar. Això és perquè encara s'han de modificar les seves respectives taules d'encaminament perquè facin servir el PCR. Afegir l'entrada en la taula d'encaminament del PC1 perquè tingui el PCR com a *gateway* per accedir a la xarxa `10.0.<PC2>.0/24`. Feu un *ping* al PC2 i comprovareu que encara no es poden comunicar. Això és perquè el PC2 rep el paquet que envia PC1 (a través del PCR) però encara no sap com contestar-li. Podeu mirar amb tcpdump que PC2 afectivament rep el ping de PC1. Afegir l'entrada en la taula d'encaminament del PC2 perquè tingui el PCR com a *gateway* per accedir a la xarxa `10.0.<PC1>.0/24`. Proveu ara de fer un *ping* des de PC1 a PC2 i viceversa per comprovar que ara si que es poden comunicar.
16) Feu servir la comanda `traceroute` per comprovar que el PC1 es comunica amb PC2 a través del PCR.
17) Investiga el tràfic que genera traceroute amb tcpdump.
18) En la configuració de la taula d'encaminament dels *hosts* (PC1 i PC2) hi heu posat una entrada amb la xarxa on està connectat (l'ha afegit linux automàticament quan heu donat l'adreça IP a la interfície) i una altra entrada amb un *gateway* que us permetia arribar a una altra xarxa. En realitat, els *hosts* solen configurar-se amb una

entrada per accedir a la xarxa on estan connectats, i una entrada *per defecte* on envien els *datagrames* destinats a la resta d'Internet. Canvieu la configuració de PC1 i PC2 substituint les rutes a les xarxes que no són la seva, per una ruta per defecte. Comprovar que hi ha connectivitat entre tots els PCs.

5.3. Tercera part: interconnexió de les xarxes de dos grups

19) Ajuntar les xarxes configurades per dos grups per aconseguir l'esquema de la Figura 5. Feu servir la comanda traceroute per a comprovar que la connexió entre PC1 i PC1' travessa els 4 routers. Apunteu les adreces IP configurades en la taula de sota. NOTA: en cas de no haver-hi dos grups disponibles, alternativament es pot configurar la xarxa de la Figura 6.

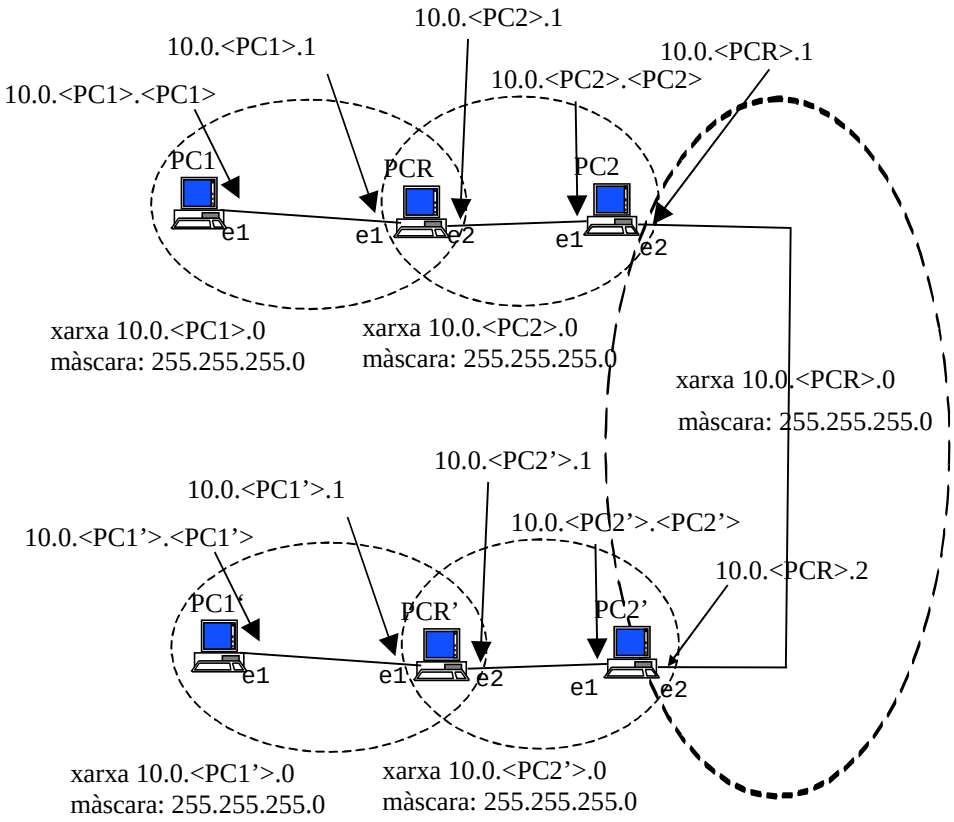


Figura 5: Topologia ajuntant les xarxes de 2 grups.

PC1/e1	
PCR/e1	
PCR/e2	
PC2/e1	
PC2/e2	
PC2'/e2	
PC1'/e1	
PCR'/e1	
PCR'/e2	
PC2'/e1	

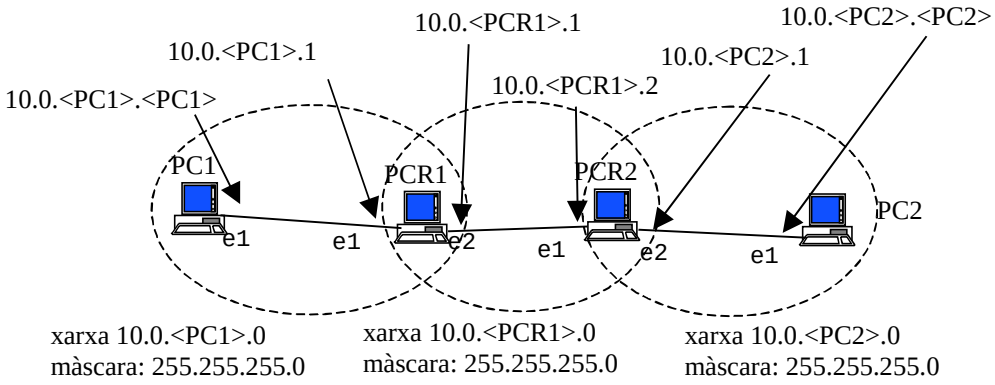
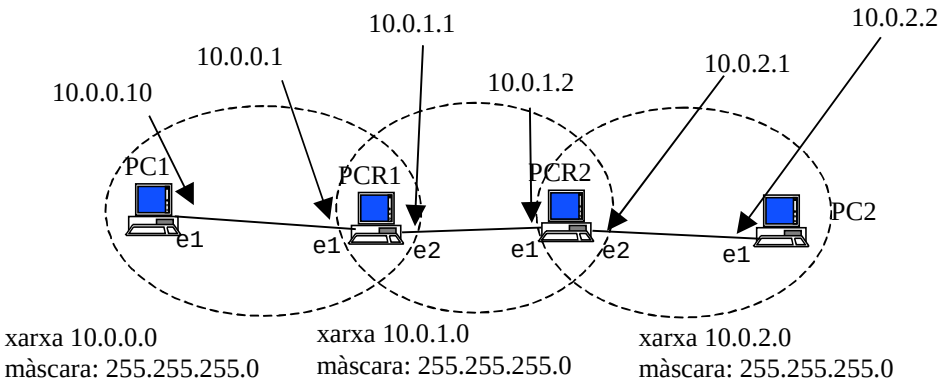


Figura 6: Topologia amb 2 routers.

PC1/e1	
PCR1/e1	
PCR1/e2	
PCR2/e1	
PCR2/e2	
PC2/e1	

6. Informe previ



Respon les següent preguntes per a la xarxa de la figura:

1) Digues quines comandes s'haurien d'executar en PC1 per assignar l'adreça IP a la interfície de xarxa, i posar PCR1 com a router per defecte.
2) Digues quines comandes s'haurien d'executar en PCR1 per assignar les adreces IP i posar PCR2 com a gateway per arribar a la xarxa 10.0.2.0
3) Suposa que, amb la xarxa configurada, en PC1 s'executa la comanda "traceroute 10.0.1.2". Quants missatges UDP enviarà PC1? Quants missatges ICMP enviarà PCR1 i PCR2?
4) Suposa que la taula d'encaminament de PC1 és la que mostra el següent bolcat. Suposant que la resta de la xarxa està correctament configurada, digues quins dels PCs de la figura serien accessibles des de PC1 (respondrien a un ping).

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 e1
10.0.2.0        10.0.0.1        255.255.255.0   U     0      0        0 e1
```

Lab 2. Routers CISCO: IOS

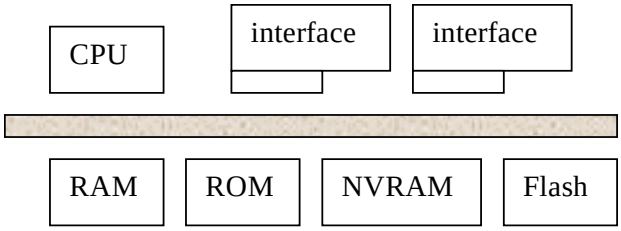
1. Objectiu de la pràctica

L'objectiu de la pràctica és conèixer els conceptes bàsics de la configuració de routers amb sistema operatiu IOS ("Internetworking Operating System") del fabricant de routers Cisco Systems.

2. Estructura d'un router

Un router IP és un ordinador especialitzat a commutar datagrames IP. Depenent de les prestacions que hagi d'oferir, la seva estructura interna és més o menys complexa i especialitzada. Però, per als models de gamma baixa podem pensar en una estructura similar a la d'un PC: CPU, memòria, busos i interfícies de xarxa. Per a l'emmagatzematge de dades és habitual utilitzar memòria ROM, memòria flaix, memòria RAM i RAM no volàtil (NVRAM):

- RAM: codi, taules d'encaminament, buffers, memòria cau ARP, etc.
- NVRAM (no volàtil): fitxer de configuració "startup-config".
- Flash (no volàtil): Imatge de l'IOS
- ROM (no volàtil): part d'imatge IOS, codi bootstrap.



Els sistemes operatius dels routers comercials estan especialment dissenyats per facilitar les tasques de commutació de paquets, l'execució d'algorismes d'encaminament, configuració d'interfícies, etc. Un exemple daquest tipus de sistemes operatius és el IOS. L'IOS té una arquitectura senzilla i normalment ocupa un espai de memòria reduït. Quan encenem un router, s'executa un programa de bootstrap carregat a la ROM que testeja el sistema i carrega a la RAM una imatge de l'IOS, normalment des de la memòria flaix.

Configurarem el router utilitzant una interfície de línia d'ordres (CLI). Normalment es fa a través d'una connexió per la línia sèrie connectada al port CONSOLE del router, usant per exemple l'aplicació Hyperterminal a Windows, MiniCOM a Linux, etc. Els paràmetres necessaris per connectar-se són els següents: Baud Rate 9600 bps, 8 bits/caràcter, 1 bits de Stop, no paritat i no control de flux hardware.

La configuració activa del router es troba en un fitxer anomenat `running-config`. Si apaguem el router, aquesta configuració es perdria i no estaria present en tornar a activar el router. Podem desar aquesta configuració en un fitxer de configuració (`startup-config`) que normalment s'enregistra en una memòria NVRAM. En arrencar el router, la configuració que s'activa és la desada al fitxer `startup-config`.

També podem configurar el router accedint per telnet o utilitzar una interfície web per configurar el router. Així mateix, tant la imatge de l'IOS com el fitxer de configuració es poden obtenir d'un servidor de tftp.

3. Modes de configuració

Els router amb IOS disposen d'un conjunt de modes anomenats de configuració que permeten la visualització i configuració del router. Els modes de configuració són els següents:

- **Mode BOOT o ROM monitor**: s'usa en casos d'emergències (prompt típicament `rmon`) com pot ser la recuperació d'un password, d'un registre de configuració, etc
- **Mode de SETUP**: permet una configuració per menú senzilla i bàsica del router
- **Mode USER EXEC**: és el mode de visualització sense privilegis (prompt `R>`)
- **Mode PRIVILEGED EXEC**: mode de visualització amb privilegis (prompt `R#`)
- **Mode de Configuració Global o CONFIGURE**: permet configurar aspectes senzills del router com poden ser la configuració del nom del router, passwords, etc (prompt `R(config)#`)
- **Mode de configuració específics**: permeten configurar protocols, interfícies o en general aspectes més complexos del router (prompt `R(config-if)#, R(config-route)#`, `R(config-line)#`, etc.)

En arrencar el router podem passar al mode SETUP, que permet donar una primera configuració al router quan aquest no té una configuració preestablerta, o bé passar al mode USER EXEC, quan el router sí que disposa d'una configuració preestablerta.

El primer missatge que emetrà el router quan connectem amb ell serà:

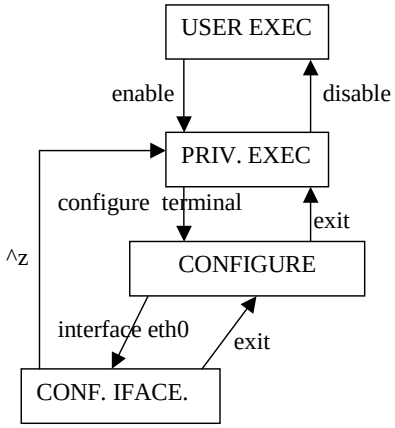
Continue with the configuration dialog [yes/no]: **no**

Al que caldrà contestar **NO.**

En mode USER EXEC podem consultar aspectes bàsics de la configuració del router[2]. Per consultar aspectes més crítics de la configuració del router hem de passar a mode PRIVILEGED EXEC. Per passar del mode USER EXEC al mode PRIVILEGED EXEC és necessari utilitzar un password (que es coneix com "*enable secret password"* que es pot establir des del mode CONFIGURE executant `enable secret <passwd>`)

Des de les maneres USER EXEC i PRIVILEGED EXEC no podem modificar la configuració del router. Per fer-ho hem de passar del mode PRIVILEGED EXEC al mode de configuració general (CONFIGURE). Des d'allà podem configurar aspectes generals del funcionament del router o passar a modes de configuració específics de cada interfície, algoritme d'encaminament, etc.

A la figura següent es mostren els diferents modes de configuració juntament amb les principals ordres necessàries per canviar d'un mode a l'altre.



Quan estem en mode USER EXEC el prompt que ens mostra el router és "**`>''`**. Quan estem a PRIVILEGED EXEC el prompt és "**`#''`** i en el mode de configuració global el prompt és **`(config)#.`**

Per exemple:

```
Router> <ordres en mode USER EXEC>
Router> enable
Router# <ordres en mode PRIV. EXEC>
Router# config terminal
Router(config)# <ordres en mode CONFIGURE>
Router(config)# exit
Router# disable
Router>
```

Com ja hem esmentat, els canvis de configuració que es realitzin en el mode de configuració global o específic es guarden sobre un fitxer de configuració resident a la RAM del router anomenat "`running-config''`. Aquest fitxer es pot visualitzar des del mode de configuració privilegiat amb l'ordre `''show running-config''.` Si el router s'apagués, aquests canvis es perdrien en estar emmagatzemats a RAM. Perquè no es perdin i passin a estar permanentment guardats en una memòria NVRAM cal copiar el fitxer "`running-config''` (RAM) a l'arxiu "`startup-config''` (NVRAM). Això es pot fer des del mode PRIVILEGED EXEC amb l'ordre "`copy running-config startup-config''`.

EN AQUEST CURS NO GUARDAREM LA CONFIGURACIÓ DEL ROUTER ENTRE SESSIONS, DE MANERA QUE NO S'HA DE REALITZAR LA CÒPIA DE CONFIGURACIÓ ESMENTADA

4. Consulta de l'estat (ordres show)

Podem consultar l'estat d'un router mitjançant les ordres `show`. Depenent del tipus d'informació que volem consultar, l'ordre és executable des de mode USER EXEC o bé necessitem els privilegis del mode PRIVILEGED EXEC. Per exemple:

show ip interface brief mostra l'estat de les interfícies, els noms i la configuració.

[2]Amb l'ordre `?` podem obtenir un llistat de les ordres que es poden executar en cada mode.

show running-config mostra el fitxer de configuració que està actiu al router

show startup-config mostra el fitxer de configuració que està gravat a la NVRAM

show ip <parameter > mostra els paràmetres associats a la configuració del protocol IP. Per exemple, la taula d'encaminament IP es consulta amb `show ip route`

show interfaces <nom_interface> mostra els paràmetres associats a la interfície

La taula d'encaminament és una informació que no es considera privilegiada i que es pot consultar des del mode usuari USER EXEC. No obstant això, el contingut dels fitxers de configuració sí que es consideren privilegiats i només poden ser visualitzats des del mode PRIVILEGED EXEC.

5. Configuració bàsica del Router

Configurar el nom i els missatges d'entrada (es mostra en connectar-se al router)

```
R(config)# hostname WORD
```

La comanda telnet pemet accedir al router remotament per a la seva administració un cop està configurat. Per accedir al route amb telnet basta executar des d'un terminal:

- `telnet ip_router`

Per poder accedir al router amb telnet cal assignar un password a enable i als terminals vty:

```
R(config)# enable password cisco →per fer telnet cal configurar password
R(config)# line vty 0 4 →configuració de 5 terminals actius per a telnet
R(config-line)# password cisco
R(config-line)# exit
```

6. Configuració de les interfícies

Des del mode de configuració podem passar a configurar les interfícies. Per exemple, per configurar una interfície ethernet podem fer:

```
Router# configure terminal
Router(config)# interface e0
Router(config-if)# ip address @IP MASK
Router(config-if)# no shutdown
Router(config-if)# exit
```

Recordeu, que amb l'ordre *show ip interface brief* podreu consultar els noms dels interfaces.

L'ordre "`no shutdown''` és necessària per activar la interfície. Per defecte, en arrencar el router tots els interfícies estan desactivats. L'ordre `''shutdown''` per defecte desactivaria administrativament una interfície.

7. Interfícies sèrie

Els nodes d'una xarxa es poden classificar en dos grans grups: equip terminal de dades (DTE) i equip de comunicació de dades (DCE). Els DTE són dispositius de xarxa que generen i son la destinació de les dades: els PC, els routers, les estacions de treball, els servidors de fitxers, els servidors d'impressió; tots són part del grup de les estacions finals. Els DCE són els dispositius de xarxa intermediaris que reben i retransmeten les trames dins de la xarxa; poden ser: commutadors (switch), concentradors (hub), repetidors o interfícies de comunicació.

Les interfícies sèrie estan dissenyades perquè en la situació més normal es connectin a una operadora de telecomunicacions mitjançant un DCE (ex.; un MODEM o una Terminació de Xarxa, TR). El DCE és el que normalment dóna rellotge i per tant fixa la velocitat de modulació i per tant de transmissió.

Tots els cables usats per crear un enllaç DTE-DCE són directes, els cables usats per a DTE-DTE i DCE-DCE són creuats.

Si es connecten dos ports sèrie de router (DTE-DTE) cal fer servir un cable creuat. A més **un dels dos ports ha d'actuar com a DCE donant rellotge.** En principi des del punt de vista de router qualsevol dels dos pot actuar de DCE, així que **l'important és que connector del cable és el que marca quin port és DCE.**

Al laboratori, els cables **de tipus RJ-45** directes seran blancs o grisos, mentre que els cables creuats seran de color vermell.

Les comandes de sota configuren la interfície sèrie. Pero en els routers que tenim al laboratori no cal executar-les perquè ja ho fan automàticament.

```
Router-DCE# configure terminal
Router-DCE(config)# interface <nom interface serie>
Router-DCE(config-if)# ip address <@IP> <MASK>
Router-DCE(config-if)# clockrate 56000
Router-DCE(config-if)# no shutdown
Router-DCE(config-if)# exit
Router-DCE(config)# exit

Router-DTE# configure terminal
Router-DTE(config)# interface <nom interface serie>
Router-DTE(config-if)# ip address @IP MASK
Router-DTE(config-if)# no shutdown
Router-DTE(config-if)# exit
Router-DTE(config)# exit
```

8. Resolució de noms

Al router es poden assignar adreces IP a noms (igual que amb el fitxer /etc/hosts en una màquina UNIX), Figura 7, i també perquè consulteu a un servidor DNS un nom desconegut (igual que amb el fitxer /etc/ resolv.conf en una màquina UNIX), Figura 8.

```
R(config)# no ip domain-lookup    →desactiva el que es busqui servidor de DNS
R(config)# ip host NAME @IP1 @IP2 →assigna noms a adreces IP
R(config)# show hosts             →llista una memòria cau de noms i @IP
                                  (configurar una interfície amb ip host name @IP)
```

Figura 7: DNS estàtic.

```
R(config)# ip domain-lookup
R(config)# ip name-server @IP server1 ….. @IP server6 →màxim 6 servidors DNS
```

Figura 8: DNS dinàmic.

Observació: Per defecte, el router té activat la resolució DNS. Si a la línia d'ordres es tecleja un nom que no es reconeix com una ordre, el router intenta contactar amb el servidor DNS per resoldre el nom, i la consola es queda congelada diversos segons. Si no hi ha servidor de noms configurat i es vol eliminar aquesta espera, es pot desactivar el DNS dinàmic executant "`no ip domain-lookup`".

9. Encaminament estàtic

A continuació veiem un exemple de configuració de l'encaminament estàtic usant l'ordre ''`ip route`''.

```
Router(config)# ip route @IPnet MASK @IPgw
```

La primera adreça és l'adreça de xarxa destinació. A continuació escrivim la màscara associada a aquesta xarxa. La tercera adreça correspon a la de la interfíci del router per on s'estableix la ruta (gateway per defecte).

10. Realització de la pràctica

Per realitzar la pràctica cada grup ha d'agafar una parella de routers connectats pel port sèrie.

10.1. Primera part

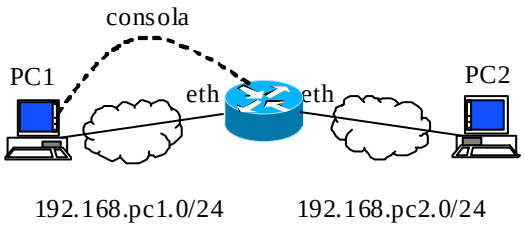


Figura 9: Primera part.

1. Configurar les interfícies dels routers de la xarxa de la Figura 9. NOTA: Les adreces IP més baixes de la subxarxa solen utilitzar-se per a les interfícies dels routers (perquè són més fàcils de recordar), i les més altes per a interfícies dels hosts. Apuntar les adreces IP configurades a la taula següent:

PC1/e1	
R1/e1	
R1/e2	
PC2/e1	

2. Configurar les interfícies dels hosts connectats amb Ethernet i configurar com a router per defecte la interfície del router corresponent. Comprovar que el host té connectivitat amb el router mitjançant l'ordre ping i veure el format de la taula d'encaminament del host. Comprovar que és possible connectar-vos al router amb telnet.
3. Comprovar que els hosts tenen connectivitat entre ells.
4. Veure i interpretar el format de la taula d'encaminament dels hosts (ordre `''route -n''`)i del router (ordre `''show ip route''`).
5. Configurar telnet assignant el password `cisco`. Comprovar que és possible connectar-se al router usant telnet des dels dos PC.

10.2. Segona part

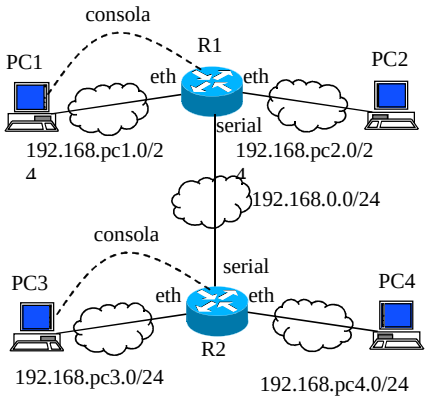


Figura 10: Segona part.

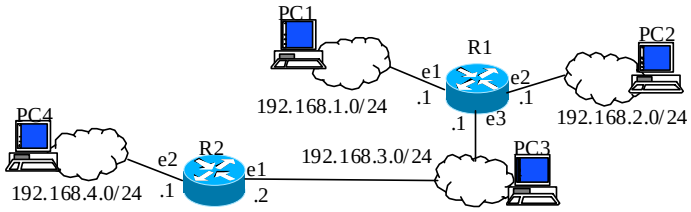
1. Configura la xarxa de la Figura 10 usant encaminament estàtic als routers (ordre "`ip route`"). Apuntar les adreces IP configurades a la taula següent:

PC1/e1	
PC2/e1	
R1/e1	
R1/e2	
R1/s1	
PC3/e1	
PC4/e1	
R2/e1	
R2/e2	
R2/s1	

2. Comprovar amb l'ordre "`show interfaces`" el tipus d'encapsulament de les interfícies sèrie. Assigna adreces IP als extrems de l'enllaç sèrie i comprova que hi ha connectivitat entre els dos routers.
3. Observar que des d'un host només es té connectivitat amb qualsevol interfície del router a la qual està directament connectat, i amb altres hosts situats en xarxes directament connectades al mateix router. Explicar per què.
4. Usar l'ordre `''ip route''` per afegir entrades estàtiques a cada router per tenir connectivitat amb qualsevol subxarxa de la xarxa establerta.
5. Veure el format de la taula d'encaminament del router amb l'ordre `''show ip route''` i comprovar que es té connectivitat amb totes les subxarxes de la xarxa.

6. Usa la ordre `traceroute` per comprovar que el PC1 es comunica amb PC4 a través dels routers.
7. Elimina les entrades no directament connectades de la taula d'encaminament de R1 (ordre “no ip route”) i afegeix una ruta per defecte cap a R2. Comprova que continua havent-hi connectivitat entre tots els PCs.
8. Desconnecta el cable Ethernet entre R1 i PC1 i observa com canvien les taules d’encaminanent en R1 i R2. Estima quan temps tarden en convergir. Torna a connectar el cable i observa de note les taules d’encaminament.

11. Informe previ



Respon les següents preguntes per a la xarxa de la figura:

1. Quines ordres s'haurien d'executar a R2 per assignar l'adreça IP les interfícies de xarxa?
2. Quines ordres s'haurien d'executar a R1 i R2 perquè totes les xarxes siguin accessibles per tots els PCs?
3. Suposa que hi ha algun error a la configuració. Auina ordre del router et permet veure la configuració actual?

Lab 3. Encaminament dinàmic: RIPv1 i RIPv2

1. Introducció a RIP

Les característiques bàsiques de RIP (RFC 2453) són:

- La mètrica és el nombre de salts fins a la destinació: 1 si la destinació és una xarxa directament connectada, 2 si cal passar per un router, etc.
- Els router envien periòdicament (cada 30 segons) un missatge RIP broadcast per cada interfície amb les destinacions i mètriques coneguts. RIP versió 2 inclou també la màscara. Els missatges s'envien a l'adreça multicast: 224.0.0.9 (*all RIPv2 routers)* amb UDP, port font i destinació: 520.
- Si us deixeu de rebre missatges RIP d'un veí (180 segons), s'assumeix que aquest router ha caigut.
- La mètrica infinit val 16.

1.1. Count to infinity

El principal problema de RIP és el temps de convergència: És a dir, el temps que passa des que hi ha un canvi a la topologia de la xarxa fins que les taules d'encaminament s'estabilitzen. Aquest temps pot ser especialment gran quan es produeix el problema anomenat *count to infinity.* Això passa quan hi ha un canvi en la topologia i la seqüència de missatges RIP enviats fan que un router *A* cregui que pot arribar una destinació *D* que ha passat a ser inaccessible, a través d'un altre router *B* que al seu torn depèn de *A* per arribar a *D.*

Per solucionar el problema del *count to infinity* es sol utilitzar el mecanisme *Split horizon.* Aquesta modificació consisteix que en enviar un missatge RIP en una interfície, s'eliminen les entrades de la taula d'encaminament que tinguin un gateway a la mateixa interfície.

Un altre mecanisme en els routers CISCO consisteix en l'anomenat *holddown timer:* Quan es rep un missatge RIP d'un veí indicant que una xarxa que ha quedat inaccessible sí és accessible a través d'aquest router, aleshores marca la ruta i inicia un temporitzador holddown en que ignora aquests missatges. Si quan expira el temporitzador encara s'anuncia la ruta com a accessible a través d'aquest router, aleshores s'actualitza la ruta a través d'aquest router.
Una altra modificació per accelerar la convergència consisteix a no esperar els 30 segons a enviar un missatge RIP quan es produeix un canvi a la taula d'encaminament. Aquesta tècnica es coneix com a *triggered updates.*

2. Configuració de RIP

Per activar l'algoritme d'encaminament RIP, els passos que cal seguir són els següents:

```
Router# configure terminal
Router(config)# ip routing
Router(config)# router rip
Router(config-router)# network @IPnet1
Router(config-router)# network @IPnet2
Router(config-router)# ^Z
```

Figura 11: Configuració de RIP

La ordre "`network`" indica les interfícies que enviaran o processaran missatges de RIP. Cal indicar les adreces de xarxa sense màscara (aquesta ordre assumeix la corresponent a la classe de xarxa). És a dir, la xarxa més gran a què pertany l'adreça IP de la interfície. Per exemple, si la interfície utilitza l'adreça IP 10.5.4.2/24 només cal anunciar la classe A 10/8 de la forma ''`network 10.0.0.0`''. Notar que l'ordre "`network`" no fa servir màscara, només l'adreça de xarxa.

Com que la versió RIPv1 no suporta subnetting, si volem una xarxa subnetejada hem d'usar RIPv2. L'ús de la versió 2 s'indica després de l'ordre "`router rip`", executant "`version 2`''.

Podem capturar els paquets que s'envien i reben amb l'ordre "`debug IP RIP`'' des de mode PRIVILEGED EXEC. Aquesta opció consumeix molts recursos del sistema, per tant, en operació normal hauria d'estar desactivat.

Amb l'ordre `show ip route` podem observar la taula d'encaminament del router. A la informació llistada pel router, apareix indicat si la ruta s'ha fixat de forma estàtica o ha estat apresa amb RIP.

L'ordre ''`show ip protocol`'' permet veure la configuració de RIP. L'ordre mostra la versió de RIP té activada cada interfície tant d'entrada (receive) com de sortida (sent). Noteu que podem enviar RIPv1 i rebre tant de RIPv1 com de RIPv2. El temps de *hold down* és el temps que espera el router a acceptar una nova ruta per a una entrada que ha estat invalidada, per evitar el *counting to infinity.*

```
router# show ip prot
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 8 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
Redistributing: rip
Default version control: send version 1, receive version 1,2
Interfície Send Recv Triggered RIP Key-chain
     Ethernet2 1 1, 2
Ethernet3 1 1, 2
```

Figura 12: Bolcat del comando show ip protocol.

Es pot activar RIPv2 globalment en totes les interfícies amb l'ordre ''versió [1 2]'':

```
Router# configure terminal
Router(config)# router rip
Router(config-router)# version 2
Router(config-if)# exit
Router(config)# exit
```

Figura 13: Activació de RIPv2.

Si un dels routers es manté amb RIPv1 i enviés missatges RIPv1 la interfície els rebutjaria. És millor canviar la versió per interfície amb les ordres: ''ip rip receive versió [1 2]'' i ''ip rip send versió [1 2]''. Per tant, activem enviar només amb versió 2 i rebre tant versió 1 com 2.

```
Router# configure terminal
Router(config)# interface e0/0
Router(config-if)# ip rip receive version 1 2
Router(config-if)# ip rip send version 2
Router(config-if)# exit
Router(config)# exit
```

Figura 14: Configuració per enviar RIPv2, però rebre RIPv1 i RIPv2.

NOTES:

Per defecte el router fa “sumarització de rutes”. La sumarització es fa a la classe, i només quan s'envien els missatges cap a una xarxa amb adreça base diferent. Per exemple, si a la taula hi ha les subxarxes 10.0.1.0/24 i 10.0.2.0/24, en enviar el missatge RIP cap a la xarxa 192.168.0.0/24 anunciarà la xarxa 10.0.0.0/8. Per desactivar la sumarització cal executar la ordre:

```
Router(config-router)# no auto-sum
```

Figura 15: Configuració de RIP perquè no faci sumarització de rutes.

Perquè el router anunciï les entrades estàtiques (això inclou l'entrada per defecte), cal executar l'ordre:

```
Router(config-router)# redistribute static
```

Figura 16: Configuració de RIP perquè afegeixi les entrades estàtiques als missatges d'update.

El router utilitza dues mètriques: la mètrica administrativa i la mètrica de l'algorisme d'encaminament. Si hi ha diverses rutes cap a una mateixa destinació, es tria la ruta amb mètrica administrativa menor. Per exemple, RIP té mètrica administrativa 120 i OSPF 110. Si tots dos afegeixen una entrada a la taula cap a una mateixa destinació, primer es triarà la ruta afegida per OSPF (el router la considera més fiable). En mostrar la taula d'encaminament podem veure les mètriques entre claudàtors:

```
R1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

R 192.168.3.0/24 [120/1] via 192.168.0.2, 00:00:08, Serial0
```

A l'entrada, la R indica que ha estat afegida per RIP. 120 és la mètrica administrativa, i 1 és la mètrica usada pel protocol. La mètrica de RIP mostrada per CISCO és el nombre de routers fins a la destinació. En anunciar la mètrica, el RFC diu que s'han d’anunciar els salts fins a la destinació (és a dir, si hi ha un router fins a la destinació, es faran dos salts). Per aquest motiu, el router CISCO incrementa en 1 les mètriques de RIP que mostra a la taula quan envia els missatges RIP.

3. Subxarxes amb classe i sense classe

Quan es fa subnetting, la primera i última subxarxa queden inutilitzades. Això passa perquè l'adreça de subxarxa de la primera subxarxa coincideix amb l'adreça de subxarxa de la xarxa major (o subnetejada) i l'adreça broadcast de l'última subxarxa coincideix amb l'adreça broadcast de la xarxa major (o subnetejada). Perquè els routers puguin treballar amb la primera subxarxa i amb l'última l'IOS activa per defecte l'ordre ''`ip subnet zero`'' (''`no ip subnet zero`'' per desactivar l'opció).

Una xarxa pot treballar amb classes (A, B o C) o pot fer servir el concepte de sense classe (CIDR). Per poder crear subxarxes independentment de la classe, l'IOS activa per defecte l'ordre "`ip classless`''. De fet, l'ordre funciona de la manera següent: si està actiu, el router envia els paquets a la interfície supernetejada que millor s'ajusti a la taula d'encaminament (o a la ruta per defecte). En el cas que està desactivada ("`no ip classless`'') el router només reenvia el paquet si la ruta és a la taula d'encaminament (o hi ha una ruta per defecte). Si no és a la taula d'encaminament, llavors el router descarta el paquet. Per exemple, si la xarxa 10.0.0.0/8 està subnetejada ia la taula hi ha les xarxes 10.0.1.0/24, 10.0.2.0/24 i una entrada per defecte. En rebre un datagrama dirigit a la xarxa 10.0.3.0/24 amb "`no ip classless`'', el router descarta el datagrama. Amb la configuración per defecte "`ip classless`'', en canvi, el router enviaria el datagrama per la ruta per defecte.

4. Realització de la pràctica

4.1. Xarxa IP amb subnetting. RIPv2 i sumarització

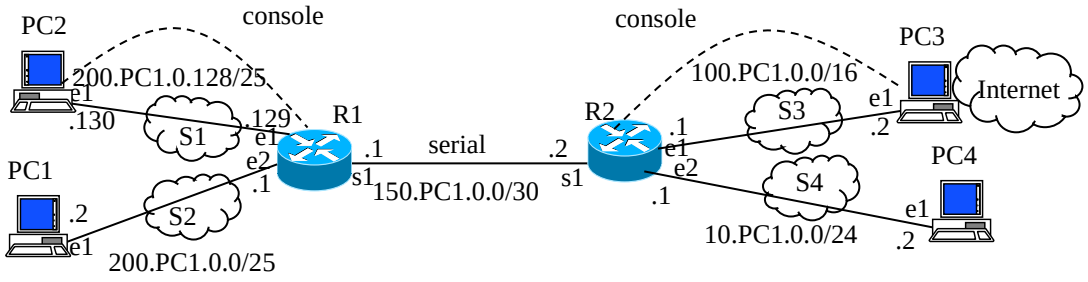


Figura 17

1) Configura la xarxa de la Figura 17. El PC3 representa el router de l'ISP que dóna accés a Internet. Per tant, el router R2 ha de tenir PC3 com a ruta per defecte. A més, PC 3 ha de tenir R2 com a router per arribar a les xarxes 200.PC1.0.0/24 i 150.PC1.0.0/24. Apuntar les adreces IP configurades a la taula següent:

PC1/e1	
PC2/e1	
R1/e1	
R1/e2	
R1/s1	
PC3/e1	
R2/e1	
R2/e2	
R2/s1	
PC4/e1	

2) Configurar les interfícies de cada router per a RIPv2. Configura RIP perquè anunciï la ruta per defecte.
3) Observar l'activació del protocol RIP usant l'ordre "`show ip protocol`" i interpretar la sortida d'aquesta ordre.
4) Observar la taula de routing amb l'ordre ''`show ip route`'' i mirar si hi ha connectivitat entre els PC.
5) Depurar RIPv2 amb l'ordre ''`debug ip rip`'' (''`no debug all`'' per desactivar-la). Interpreta els missatges.
6) Executar l'ordre "no auto-sum" a la configuració de RIP dels routers. Com canvien els missatges RIP i les taules d'encaminament?
7) Observar la convergència del protocol RIP si desconnectem PC1, usant de l'ordre "`debug ip rip`''. Interpreta els missatges. Observa com en desconnectar transcorre un temps fins que les taules

convergeixen i com s'envia immediatament un triggered update amb mètrica infinit (16 salts). Tornar a connectar i observar els canvis.

8) Si desactivem *split-horizon* en una de les interfícies Quines xarxes s'anunciaran en un missatge d'encaminament RIP? Per desactivar split-horizon heu d'executar l'ordre `''no ip split-horizon''` des del submode d'interfície. Per exemple per deshabilitar split-horizon a la interfície e0/0:

```
Router# configure terminal
Router(config)# interface e0/0
Router(config-if)# no ip split-horizon
Router(config-if)# exit
Router(config)# exit
```

4.2. Xarxa IP amb subnetting. RIPv2 entre diversos grups (opcional)

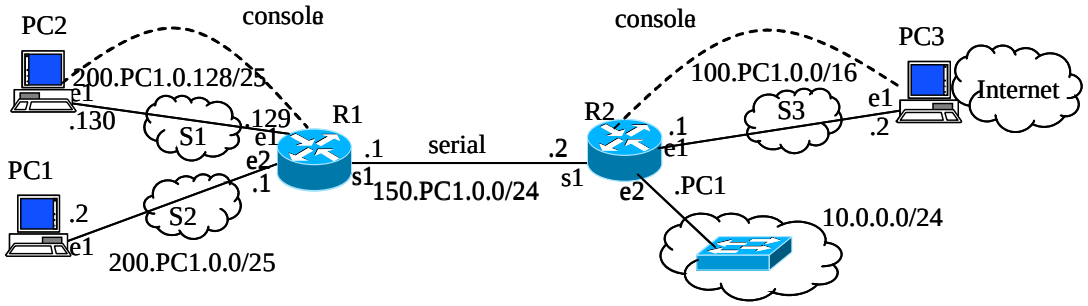
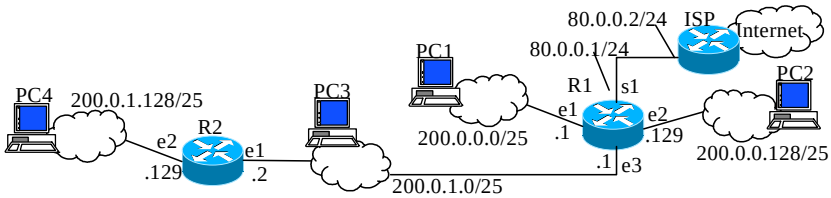


Figura 18

1) Interconnectar les xarxes de diversos grups mitjançant un switch, tal com mostra la figura. Activar RIP a la xarxa 10.0.0.0/24 i comprovar la convergència de les taules d'encaminament.

5. Informe previ



Respondre a les preguntes següents per a la xarxa de la figura. Suposa que les adreces IP de la figura ja s'han assignat a les interfícies.

1) Quina és l'ordre que s'ha d'executar a R1 per tenir una ruta per defecte cap a l'ISP?
2) Quines ordres s'haurien d'executar a R1 i R2 per configurar RIP? És desitja que R1 adverteixi la ruta per defecte.
3) Quina serà la taula d'encaminament de R1 i R2 quan RIP hagi convergit?
4) Suposar que es fa servir split-horizon. Quin és el contingut dels missatges RIP que R1 i R2 enviaran a la xarxa 200.0.1.0/25?
5) Repetir els dos apartats anteriors si executem "no auto-sum" a la configuració de RIP dels dos routers.

Lab 4. Laboratori d'ACLs (Access Lists) i NAT amb IOS

1. Introducció

Les llistes d'accés (ACL) s'usen per al filtratge de paquets en funció de certs paràmetres com ara les adreces de xarxa origen o destinació, els ports origen o destinació, el tipus de protocol (ip, icmp, tcp, udp, etc.)). Una de les aplicacions on es fan servir més les llistes d'accés és a la seguretat de la xarxa. Amb les ACL es pot bloquejar el trànsit no desitjat en una interfície ja sigui de sortida o d'entrada. Les ACLs no només es fan servir per seguretat, sinó que també per identificar paquets en aplicacions com ara NAT (Network Address Translation), en BGP per filtrar rutes en crear polítiques d'encaminament, etc.

Hi ha ACLs per a diferents piles de protocols: TCP/IP, IPX/SPX, Apple, etc. Aquest document se centra en les ACL aplicades a seguretat a la xarxa per a la pila de protocols TCP/IP. Capa protocol té assignat un rang de ACLs. Per exemple les ACLs entre la 1 i la 199 s'usen en TCP/IP.

Quan creem una llista d'accés i l'apliquem a una interfície d'entrada o sortida, estem creant una seqüència d'instruccions que es revisen cada vegada que un paquet entra o surt per aquesta interfície. És important notar diverses característiques de les ACLs.

Primer, una ACL s'aplica a la interfície ja sigui d'entrada o de sortida. Es pot crear una ACL per a la interfície de sortida i una altra de diferent per la interfície d'entrada.

Segon, les ACL són seqüències d'instruccions que són revisades contra el paquet. L'ordre de les instruccions és important, ja que quan una línia de la seqüència dóna cert a la comprobació, es pren una acció i se surt de l'ACL, és a dir no es continua revisant per comprovar que hi hagi una altra línia de la seqüència que també resulta certa. Per tant, és molt important dissenyar l'ACL en la seqüència que ens interessi més.

Per exemple, no és el mateix aquestes dues línies d'una ACL:

- Si el paquet és ICMP rebutjar
- Si el paquet és IP acceptar

que la seqüència:

- Si el paquet és IP acceptar
- Si el paquet és ICMP rebutjar

Suposem que arribés un paquet ICMP. En el primer cas, el paquet es rebutjaria ja que la primera línia es compleix, el paquet és ICMP. En el segon cas el paquet ICMP s'acceptaria ja que la primera línia també es compleix, de manera que ja no es comprovaria la segona.

Un altre aspecte important és que no podem inserir línies a la seqüència. Si ens equivoquem en crear-la o volem inserir una línia, cal esborrar les línies fins al punt d'inserció.

Finalment, també **molt important**, l'última línia d'una llista d'accés **mai** apareix, és a dir, existeix de forma explicita i sempre és **denegar tot**.

Dins les llistes d'accés TCP/IP hi ha dos tipus d'ACLs

- Llistes d'accés IP estàndard (1-99)
- Llistes d'accés IP esteses (100-199)

2. Wildcard mask

La wildcard mask és una màscara de 32 bits que indica quins bits de l'adreça IP s'han de comprovar i quins no. Si els bits de la màscara estan a 0 aleshores es comproven, si estan a 1 aleshores no es comproven.

Per exemple, si volem que un paquet que entra es comprovi si pertany al host amb adreça IP `145.34.5.6`, volem que es comprovin tots els bits de l'adreça IP. Això significa que la wildcard mask seria `0.0.0.0`. En aquest cas se sol substituir la tupla *`@IP WildcardMask`* per *`host @IP`*. Per exemple la tupla *`145.34.5.6 0.0.0.0`* es pot expressar com a *`host 145.34.5.6`*.

Si volguéssim que no es comprovés cap bit, posaríem una wildcard mask de `255.255.255.255`. en aquest cas se sol substituir la tupla `@IP WildcardMask` per `any`. Per exemple la tupla `145.34.5.6 255.255.255.255` es pot expressar com a `any`.

També podem expressar xarxes. Per exemple, per comprovar tots els paquets que vinguin de la xarxa `145.34.5.0/24`. Això vol dir que hem de comprovar tots els paquets els primers 24 bits dels quals coincideixin amb els de l'adreça de xarxa. Després la WildcardMask corresponent hauria de ser `0.0.0.255`.

3. ACL estàndard

Les ACL estàndard només usen les adreces origen per fer la comprovació. Les llistes d'accés estàndard tenen números (`acl#`) compresos entre l'1 i el 99. La ordre té el format següent:

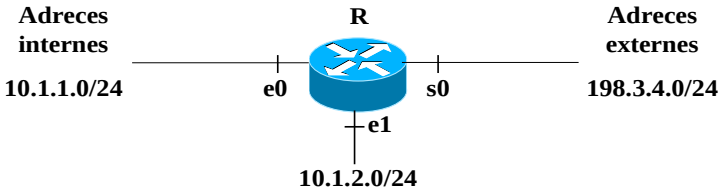
```
access-list acl# {deny|permit} {@IPsource WildcardMask | host @IPsource | any}
ip access-group acl# {in |out}
```

La primera ordre, access-list, crea la llista d'accés amb número `acl#` i amb condició denega o permet sobre l'adreça IP origen especificada amb la corresponent wildcard mask. Recordeu que la darrera línia d'una ACL mai no apareix però sempre és "`access-list acl# deny any`".

La segona ordre, `access-group`, assigna la llista d'accés acl# sobre el protocol IP sobre la interfície d'entrada o de sortida on s'executa aquesta ordre.
Per esborrar una ACL executar l'ordre:

```
no access-list acl#
```

Exemple: Volem denegar a la interfície s0 de sortida qualsevol paquet IP que provingui de la xarxa `10.1.1.0/24.`




```
R# configure terminal
R(config)# access-list 1 deny 10.1.1.0 0.0.0.255
R(config)# access-list 1 permit any
R(config)# interface s0
R(config-if)# ip access-group 1 out
R(config-if)# exit
R# show access-lists
```

Primer creem la llista d'accés amb número igual a 1 i deneguem tot el trànsit que vingui de la xarxa `10.1.1.0/24.` Com que l'última línia seria denegar tota la resta (ex.; la xarxa `10.1.2.0/24`), permetem la resta d'adreces. Apliquem aquesta ACL sobre la interfície de sortida s0 perquè si ho féssim sobre l'e0 d'entrada aleshores bloquejaríem els paquets de la xarxa `10.1.1.0/24` cap a la xarxa `10.1.2.0/24.`

4. ACLs esteses

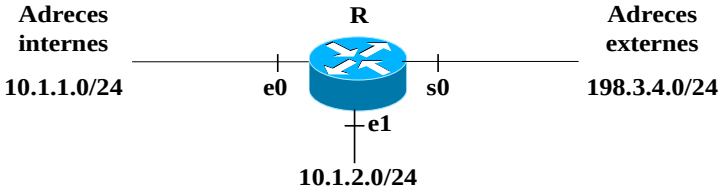
Les ACLs esteses permeten utilitzar tant les adreces origen com a destinació per fer la comprovació. A més, permeten especificar el protocol sobre el qual es vol fer la comprovació i en el cas que sigui TCP o UDP especificar el port. Les llistes d'accés esteses tenen números (`acl#`) compresos entre el 100 i el 199. La ordre té el format següent:

```
access-list acl# {deny|permit} protocol {@IPsource WildcardMask | host @IPsource | any}
[operador portsource] {@IPdest WildcardMask | host @IPdest | any} [operador portdest ]
[established]
ip access-group acl# {in |out}
```

La primera ordre, access-list, crea la llista d'accés estesa amb número acl# i amb condició de negatiu o permet sobre l'adreça IP origen i/o destinació especificades amb les corresponents wildcard masks. *`protocol`* pot ser ip, icmp. tcp, udp, etc. *`Operador`* pot ser *{lt, gt, eq, neq}* (less than, greater than, equal, non equal) i `port` és un port TCP o UDP. `established` només és vàlid amb tcp i, quan es fa servir, captura el trànsit tcp d'una connexió establerta. Per això el router mira els paquets amb el bit ACK o RST activats (el primer paquet de SYN sempre té aquests dos bits desactivats).

Recordeu que l'última línia d'una ACL no apareix però és "`access-list acl# deny ip any any`".

Exemple: Volem denegar a la interfície s0 de sortida qualsevol paquet ICMP que provingui de la xarxa `10.1.1.0/24` i l'accés a qualsevol port telnet (port 23) per part d'un host d'aquesta xarxa.



```
R# configure terminal
R(config)# access-list 101 deny icmp 10.1.1.0 0.0.0.255 any
R(config)# access-list 101 deny tcp 10.1.1.0 0.0.0.255 any eq 23
R(config)# access-list 101 permit ip any any
R(config)# interface s0
R(config-if)# ip access-group 101 out
R(config-if)# exit
R# show access-lists
```

Primer creem la llista d'accés estesa 101, denegant l'accés de paquets ICMP, segon una altra línia denegant l'accés a qualsevol host amb port 23, finalment permetem qualsevol altre tipus de trànsit. A continuació, apliquem la llista d'accés a la interfície de sortida s0.

5. Verificació

`R# show ip interface`	Mostra si hi ha alguna ACL a la interfície.
`R# show access-lists`	Mostra les ACL definides
`R# show running-config`	Per comprovar la configuració.

6. NAT

NAT (Network Address Translation) és el procés que permet la translació d'adreces privades a públiques mitjançant la substitució o alteració de les adreces IP o ports a les capçaleres IP i TCP del paquet transmès. Perquè NAT funcioni hem de disposar d'un router que implementi NAT en alguna o diverses variants: NAT estàtic, NAT dinàmic i NAT per ports (PAT).
No sempre es fa servir NAT per traslladar adreces privades a públiques. Hi ha ocasions en què es traslladen adreces privades a privades o adreces públiques a adreces públiques. Les adreces internes poden ser tant privades com a públiques. El cas més típic és aquell en què la direcció interna és una adreça privada i la direcció externa és una adreça pública. IOS utilitza la següent nomenclatura genèrica a l'hora de fer servir NAT:

- **Adreces locals internes (Inside local addresses):** l'adreça IP interna assignada a un host a la xarxa interna
- **Adreces globals internes (Inside global addresses):** l'adreça IP d'un host a la xarxa interna tal com apareix a una xarxa externa
- **Adreces locals externes (Outside local addresses):** l'adreça IP d'un host extern tal com apareix a la xarxa interna
- **Adreces globals externes (Outside global addresses):** l'adreça IP assignada a un host extern en una xarxa externa

Veure que la diferència entre una adreça local i global interna és que la direcció local interna és la direcció que volem traslladar mentre que la direcció global interna és la direcció ja traslladada.

7. NAT estàtic

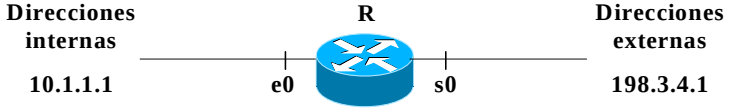
Fem servir NAT estàtic quan les adreces estan emmagatzemades en una taula de consulta del router i s'estableix un mapatge directe entre les adreces internes locals i les adreces internes globals. Això vol dir que per cada adreça interna local existeix una adreça interna global. Aquest mecanisme se sol utilitzar quan es vol canviar un esquema d'adreces d'una xarxa a un altre esquema d'adreces o quan es tenen servidors que han de mantenir una adreça IP fixa de cara a l'exterior com ara DNS o servidors web.

7.1. Configuració de NAT estàtic

Per configurar NAT estàtic seguirem els passos següents:

- Definir el mapeig de les adreces estàtiques:
  ip nat inside source static *local-ip global-ip*
  ip nat inside source static network *local-network global-network mask*
- Especificar la interfície interna
  ip nat inside
- Especificar la interfície externa
  ip nat outside

Exemple:




```
R# configure terminal
R(config)# ip nat inside source static 10.1.1.1 198.3.4.1
R(config)# interface e0
```

```
R(config-if)# ip nat inside
R(config-if)# exit
R(config)# interface s0
R(config-if)# ip nat outside
R(config-if)# exit
R(config)# exit
```

8. NAT dinàmic

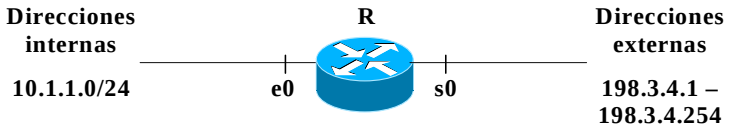
Fem servir NAT dinàmic quan disposem d'un conjunt d'adreces globals internes que s'assignaran de manera dinàmica i temporal a les adreces locals internes. Aquesta assignació s'efectuarà quan es rep trànsit al router i té un temporitzador assignat.

8.1. Configuració de NAT dinàmic

Per configurar NAT dinàmic seguirem els passos següents:

- Crear un conjunt d'adreces globals:
  ip nat pool *name start-ip end-ip {***netmask** *mask* | **prefix-length** *prefix-length}*
- Crear una ACL que identifiqui els hosts per a la translació
  access-list *access -list-number* **permit** *source {source-wildcard}*
- Configurar NAT dinàmic basat en la direcció origen
  ip nat inside source list *access-list-number* **pool** *name*
- Especificar la interfície interna
  ip nat inside
- Especificar la interfície externa
  ip nat outside

Exemple:




```
R# configure terminal
R(config)# ip nat pool fib-xc 198.3.4.1 198.3.4.254 netmask 255.255.255.0
R(config)# access-list 2 permet 10.1.1.0 0.0.0.255
R(config)# ip nat inside source list 2 pool fib-xc
R(config)# interface e0
R(config-if)# ip nat inside
R(config-if)# exit
R(config)# interface s0
R(config-if)# ip nat outside
R(config-if)# exit
R(config)# exit
R# show ip nat translations
```

Les entrades s'assignen per defecte 24 hores. Si es vol modificar el temporitzador, utilitzar la següent ordre:

```
R(config)# ip nat translation timeout seconds
```

On seconds és el temps que s'assignarà al temporitzador.

9. NAT overload o PAT (Port Address Translation)

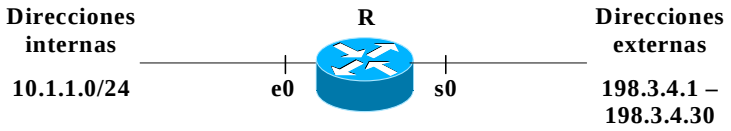
Fem servir PAT (NAT per ports) quan disposem d'una adreça global interna que pot direccionar tot un conjunt gran (centenars) d'adreces locals internes. Aquesta assignació la fa amb el parell direcció global/port. Encara que disposem de 65535 ports (16 bits) en realitat el router PAT només pot fer servir un subconjunt d'aquests ports (depèn del router, però aproximadament unes 4000 ports per adreça global). PAT es pot utilitzar en conjunció amb NAT dinàmic de manera que diverses adreces globals amb múltiples ports direccionen un nombre més gran d'adreces locals internes.

9.1. Configuració de PAT

Per configurar PAT seguirem els passos següents:

- Crear un conjunt d'adreces globals (pot ser una sola adreça):
  ip nat pool *name start-ip end-ip {***netmask** *mask* | **prefix-length** *prefix-length}*
- Crear una ACL que identifiqui els hosts per a la translació
  access-list *access -list-number* **permit** *source {source-wildcard}*
- Configurar PAT basat en la direcció origen
  ip nat inside source list *access-list-number* **pool** *name* **overload**
- Especificar la interfície interna
  ip nat inside
- Especificar la interfície externa
  ip nat outside

Exemple: farem servir fins a 30 adreces internes globals, cadascuna de les quals fa PAT




```
R# configure terminal
R(config)# ip nat pool fib-xc 198.3.4.1 198.3.4.30 netmask 255.255.255.0
R(config)# access-list 2 permet 10.1.1.0 0.0.0.255
R(config)# ip nat inside source list 2 pool fib-xc overload
R(config)# interface e0
R(config-if)# ip nat inside
R(config-if)# exit
R(config)# interface s0
R(config-if)# ip nat outside
R(config-if)# exit
R(config)# exit
R# show ip nat translations
```

En cas que no hi hagi un conjunt d'adreces globals podem fer servir l'adreça assignada a la interfície “s0” de la següent manera:

```
R(config)# ip nat inside source list 2 interfície s0 overload
```

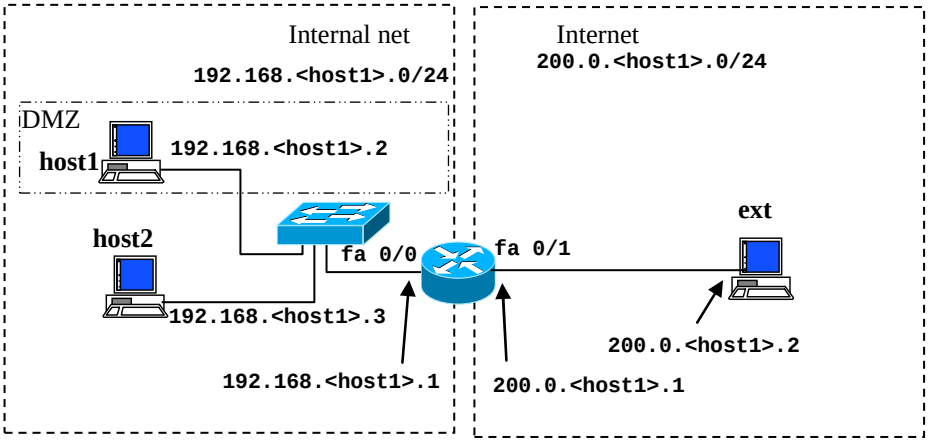
10. Verificació d'una configuració NAT

Fem servir les següents ordres per verificar que la configuració NAT és correcta (des de manera privilegiada):

```
R# show ip nat translations
R# show ip nat translations verbose
R# show ip nat statistics
R# debug ip nat (no debug ip nat)
R# clear ip nat translation *                    → elimina totes les translacions NAT
```

11. Realització de la pràctica

11.1. NAT



1) Configurar la xarxa de la figura. La xarxa de l'esquerra representa una xarxa privada i la de la dreta representa Internet. Fixeu-vos que la xarxa privada té adreces privades (no enrutables a Internet): Recordeu que els rangs d'adreces privades són 10.0.<host1>.0/8, 172.16.<host1>.0/12 i 192.168.<host1>.0/16. Configurar host1 i host2 perquè tinguin una ruta per defecte usant el router. Configurar el host que representa Internet (ext) perquè només sàpiga arribar a les adreces públiques: És a dir, no afegir a la taula d'encaminament d'ext cap ruta per defecte. D'aquesta manera, ext només podrà arribar a la xarxa directament connectada, que representa Internet. Apuntar les adreces IP configurades a la taula següent:

host1/e1	
host2/e1	
R1/fa0/0	
R1/fa0/1	
ext/e1	

2) Comprovar fent *pings* que hi ha connectivitat entre host1-host2-router i entre ext-router. Comprovar que no hi ha connectivitat entre host1/2-ext (ja que ext no pot contestar els datagrames que arriben amb una adreça font privada).
3) Configurar PAT (sense canviar la configuració anterior) perquè tots els hosts de la xarxa interna accedeixin a Internet amb l'adreça pública de la interfície fa0/1 del router (200. 0.< host1>.1):
   - Comprovar que tots dos hosts de la xarxa interna poden accedir a Internet.
   - Comprovar el funcionament de NAT amb `debug ip nat` (executa `no debug ip nat` per desactivar l'ordre).
   - Comprovar la taula NAT (`show ip nat translations`).
   - Comprovar que ext no pot accedir als hosts de la xarxa Interna (host1/2). Raonar perquè no és possible.
4) Configurar un static NAT de 200.0.< host1>.1 cap al host1. Comprovar amb "debug ip nat" al router que ext té connectivitat amb host1 (amb ping des d'ext).

11.2. ACLs

Continuant amb la configuració anterior:

5) Configurar una llista d'accés estàndard perquè només pugui accedir a Internet host2. Tindre en compte que l'ordre en què s'aplica NAT i ACL en una interfície és: primer ACL in, després NAT i finalment ACL out. Què passa amb ext? Es té accés a host1? Per què?
6) Esborrar l'ACL estàndard anterior i fer servir una ACL estesa per crear una configuració que permeti accedir des d'Internet només al servei ssh (port 22) de host1. Per comprovar-ho connectar des d'ext al servidor ssh de host1, i després intentar connectar des d'ext amb telnet a host1. Desitgem que host2 continuï amb accés a Internet, però des de host1 no ha de ser possible iniciar una connexió amb Internet. Comprovar-ho confirmant que és possible connectar amb telnet des de host2 a ext, però no des de host1.

12. Informe previ

1. Digues les comandes que permeten la configuració PAT del punt 3) del guió.
2. Digues les comandes que permeten la configuració NAT estàtic del punt 4) del guió.
3. Digues les comandes que permeten la configuració de les ACL del punt 5) del guió.
4. Digues les comandes que permeten la configuració de les ACL del punt 6) del guió.

Lab 5. Switches

1. Introducció

Un switch Ethernet és un dispositiu de nivell 2 que segmenta els dominis de col·lisions. La configuració d'un switch és totalment dependent del fabricant. En aquest laboratori utilitzarem switches Ethernet de la gamma 2950 de CISCO. Per entrar i configurar el switch seguirem els mateixos passos que en un router CISCO. Igual que els routers, ens connectem pel port consola del switch amb l'adaptador i un cable directe i amb l'aplicació minicom. Un cop connectats entrem en mode setup, o en mode user exec. Del mode user exec hem d'entrar al mode privilegiat amb l'ordre "`enable`". En aquest mode podrem visualitzar taules, fitxers de configuració (running-config), bases de dades del switch, etc. Per configurar qualsevol funcionalitat cal entrar en el mode de configuració global usant l'ordre "`configure terminal`".

2. Taula MAC

Cada port d'un switch és un domini de col·lisions. Per segmentar la xarxa Ethernet, un switch utilitza la taula MAC. El switch inicialment té la taula buida. Cada vegada que una estació envia una trama Ethernet a un altre host, el switch "aprèn" a quin port està connectat una adreça MAC. Per exemple, si una trama Ethernet entra pel port del switch e0 amb direcció origen MAC=A té destinació la MAC=B, el switch aprèn que la MAC=A està connectada al port e0.

A mesura que els hosts envien peticions a altres hosts i aquests responen, la taula MAC es va omplint. Com que els hosts poden canviar de situació (passar a estar connectats a un altre port), no convé que les entrades de la taula MAC siguin estàtiques. Per això les entrades tenen un temps de vida ("age"). Passat el temps de vida, l'entrada de la taula MAC desapareix (aging out). Per això diem que les entrades són *dinàmiques.*

- Verificació:

```
Switch# show mac-address-table
```

Per defecte un switch CISCO de gamma 2950 té assignat un temps de vida d'entrades a la taula MAC de 300 segons (5 minuts), mecanisme d'aprenentatge dinàmic i cap estrada estàtica a la taula.

Per veure la taula MAC d'un switch podem utilitzar l'ordre "`sh mac address-table''`. Per veure el temps de vida es pot fer servir l'ordre "`sh mac address-table aging-time''`. Per eliminar entrades apreses dinàmicament es pot fer servir l'ordre "`clear mac address-table dynamic''` (totes les entrades) o "`clear mac address-table dynamic address @MAC''` (eliminar l'adreça @MAC de la taula) o "`clear mac address-table dynamic interface IFACE''` (per a les MACs d'una interfície) o "`clear mac address-table dynamic vlan VLAN-ID''` (totes les MACs d'una VLAN).

3. VLANs

Definim una VLAN com una xarxa broadcast. Cadascun dels ports d'un router és una xarxa broadcast per definició i per tant una xarxa IP. Per estalviar ports de router es poden crear xarxes broadcast (xarxes IP) en un switch mitjançant per configuració. Això significa que amb un port de router connectat al switch crearem tantes VLANs (xarxes broadcast) com la configuración del switch ens permeti. Un switch CISCO de la gamma 2950 permet crear fins a 1024 VLANs.

És clar que si un port de router ha de suportar N VLANs (N xarxes IP) el port haurà de tenir N adreces IP, una per cada VLAN creada. També és clar que per viatjar des d'una VLAN a una altra cal passar obligatòriament pel router. És a dir, no es pot anar des d'una VLAN a una altra directament a través del switch, de la mateixa manera que el trànsit broadcast de nivell 2 (per exemple les trames ARP) no es propaguen entre VLANs diferents. Per aconseguir aquesta segmentació de nivell 3 s'utilitza un protocol específic anomenat "trunking". Un enllaç en mode trunk pertany a més d'una VLAN, de manera que permet enviar en un sol enllaç tot el trànsit de les VLANs del switch al router (aquesta configuració es coneix amb el nom de router-on-a-stick). Per això, les trames que s'envien al trunk porten una etiqueta (*tag)* amb el número de VLAN a què pertany la trama. Hi ha dos protocols de trunking: el que es va fer servir per primera vegada, propietari de CISCO, conegut com a ISL, i l'estandarditzat per l'IEEE: IEEE802.1Q. Als equips de CISCO podem trobar tots dos protocols (els equips més moderns solen portar només IEEE802.1Q).

3.1. Configuració del switch

Quan encenem un switch CISCO, tots els ports pertanyen a la VLAN nativa. La VLAN nativa per definició és la VLAN-ID=1. Si es defineix un VLAN per a un ús específic és millor utilitzar altres VLAN-ID diferents de l'1. Per definir VLANs en un switch seguirem els passos següents:

```
Sw# configure term
Sw(config)# vlan VLAN-ID
Sw(config-vlan)# name NAME
Sw(config-vlan)# exit
```

on VLAN-ID té rang 0001–1005, **creem** la VLAN amb **nom** i **numero**. Nota: VLAN 1, 1002, 1003, 1004 i 1005 són VLANs per defecte per a diverses tecnologies de nivell 2 (Ethernet, FFDI, TR,...)

```
Sw# show vlan
Sw# show vlan id VLAN-ID
```

llista paràmetres de totes o una VLAN determinada. Per esborrar una VLAN:

```
Sw# configure term
Sw(config)# no vlan VLAN-ID
Sw(config-vlan)# exit
```

Quan la VLAN està creada cal assignar interfícies a la VLAN. Usar l'ordre switchport per assignar de forma estàtica ports a una VLAN:

```
Sw(config)# interface fastethernet0/1
Sw(config-if)# switchport mode access →defineix VLANs en mode estàtic
Sw(config-if)# switchport access vlan VLAN-ID →assignar el port a la vlan creada vlan-id
Sw(config-if)# exit
Sw(config)# exit
Sw# show running-config interface IFACE →verifica el VLAN membership de la interfície tal com està a la memòria física
Sw# show interfaces IFACE switchport →llista el mode administratiu (ex.; accés estàtic), el mode d'accés de la VLAN (ex.; vlan-id), etc
Sw# show vlan →llista informació de les vlans creades
```

Un cop creada la VLAN al switch cal definir l'enllaç entre el switch i el router com un enllaç ("link") de tipus "trunk". Un "link trunk" és un enllaç que pertany a totes les VLANs creades. Ha d'estar assignada a la VLAN nativa (VLAN=1). Només interfícies Fast Ethernet poden ser trunk.

```
Sw(config)# interface fastethernet0/1
Sw(config-if)# switchport mode trunk
Sw(config-if)# exit
Sw(config)# exit
Sw# show interfícies IFACE trunk
```

Ara el switch ja està configurat. Ens falta configurar el router perquè entengui les diferents VLANs creades.

3.2. Configuració del router

L'enllaç del router ha de ser un "link trunk" ia més ha de tenir tantes adreces IP com a VLANs creades. Per això crearem subinterfícies a la interfície Fast Ethernet del router. Cada subinterfície l'assignarem a una VLAN i us donarem una IP. Al següent exemple creem 2 VLANs (VLAN-ID=2 i VLAN-ID=3) al router. Fem servir la interfície Fast Ethernet 0/0 com a interfície de partida on crearem les subinterfícies Fast Ethernet 0/0.1 i Fast Ethernet 0/0.2 i assignem el VLAN-ID a aquesta subinterfície (amb l'ordre encapsulation). Finalment donem una IP a la subinterfície:

```
R(config)# int fastethernet 0/0
R(config-if)# no ip address
R(config-if)# no shutdown
R(config-if)# int fastethernet 0/0.1
R(config-subif)# encapsulation dot1q VLAN-ID2
R(config-subif)# ip address @IP2 MASK2
R(config-subif)# exit
R(config-if)# int fastethernet 0/0.2
R(config-subif)# encapsulation dot1q VLAN-ID 3
R(config-subif)# ip address @IP3 MASK3
R(config-subif)# exit
R(config-if)# exit
R(config)# exit
R# sh ip route
```

Observar que a la taula d'encaminament ha d'aparèixer una entrada amb cada subinterfície i la seva subxarxa IP.

4. Ports segurs

Hi pot haver situacions en què ens interessi fixar adreces MAC a l'entrada de la taula MAC. Per exemple, per motius de seguretat només volem que en un port del switch Ethernet es pugui connectar físicament el host A. Si es connecta un altre host amb diferent adreça MAC a A volem que el port es deshabiliti. Amb això augmentem la seguretat de la nostra xarxa. Aquesta solució s'anomena ports segurs. Per defecte la seguretat per ports està desactivada, per activar-la en una interfície:

```
Switch(config-if)# switchport port-security
```

Per afegir ports segurs:

- El port ha d'estar en mode *access.* Per canviar el mode d'un port:

```
Switch(config-if)# switchport mode {access | dynamic {auto | desirable} | trunk}
```

Descripció:

Access	Set the port to access mode (either static-access or dynamic-access depending on the setting of the switchport access vlan inteface configuration command). The port is set to access unconditionally and operates as a nontrunking, single VLAN interface that transmits and receives nonencapsulated (non-tagged) frames. An access port can be assigned to only one VLAN.
Dynamic auto	Set the interface trunking mode dynamic parameter to auto to specify that the interface convert the link to a trunk link.
Dynamic desirable	Set the interface trunking mode dynamic parameter to desirable to specify that the interface actively attempt to convert the link to a trunk link.
Trunk	Set the port to trunk unconditionally. The port is a trunking VLAN Layer 2 interface. The port transmits and receives encapsulated (tagged) frames that identify the VLAN of origination. A trunk is a point-to-point link between two switches or between a switch and a router.

The default mode is **dynamic desirable.**

La manera d'aconseguir un port segur és especificar el nombre màxim d'adreces MAC que es poden associar a un port Ethernet i fixar les adreces MAC que ens interessen com a segures en aquest port. Però primer hem de buidar la taula MAC esborrant les adreces dinàmiques que hagi pogut afegir el switch amb l'ordre:

```
Switch# clear mac address-table {dynamic [address mac-addr | interfície interfície-
id | vlan vlan-id]}
```

Per limitar el màxim nombre de MAC permeses en una interfície:

```
Switch(config-if)# switchport port-security maximum max_addrs
```

Si volem assignar una MAC segura a una interfície d'una VLAN determinada cal executar:

```
Sw(config-if)# switchport port-security mac-address @MAC
Sw# show mac-address-table static
```

A continuació es defineix l'acció a prendre quan es produeix una violació de ports.

```
Sw(config-if)# switchport port-security violation {protect | restrict | shutdown}
```

on "protect" significa que es descarten trames de les MAC que violen el sistema, "restrict" significa que a més s'envia un trap (avís) al gestor de xarxa (protocol SNMP) i "shutdown" (per defecte) que es desactiva el port.

Verificació:

```
Switch# show port-security [interfície interfície-id | address]
Switch# show mac-address-table
Switch# show running-config
```

NOTA: En violar la seguretat del port, aquest queda bloquejat. Per reactivar-lo, executar:

```
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
```

5. Realització de la pràctica

La configuració del lab serà d'un router connectat a un switch per un enllaç Fast Ethernet (ha de ser Fast Ethernet per suportar *trunking)*. A cada switch connectarem 3 PCs.

5.1. VLANs i trunking

1) Esborrar les VLANs creades per l'usuari (p. ex: configure term; no vlan 2). Què passa si intenteu esborrar la VLAN=1?
2) Configura la topologia de la Figura 19. Crea les estacions T_1 i T_2 com a pertanyents a la VLAN=2 i l'estació T_3 a la VLAN=3. Configura el router perquè accepti VLANs. Apuntar les adreces IP configurades a la taula següent.

T1/e1	
T2/e1	
R1/fe1.1	
R1/fe1.2	
T3/e1	

3) Comprovar que podeu fer ping entre totes les estacions.
4) Comprovar que les entrades que s'han afegit en la taula MAC s'esborren automàticament després d'un temps.
5) Fer ping entre els PCs fins que totes les adreces MAC estigun en la taula MAC. A continuació identificar l'adreça MAC i VLAN de tots els PCs i del router en la taula MAC.
6) Observar la taula de routing del router. Quines entrades i quin format tenen?
7) Executar tcpdump a les estacions per veure el trànsit rebut/transmès.
8) Fer un ping des de T_1 a T_2. Quins dispositius veuen trànsit? Per què?
9) Fer un ping des de T_1 a T_3. Quins dispositius veuen trànsit? Per què?
10) Usar la ordre traceroute entre T1 i T2, i entre T1 i T3. Raona les diferències.
11) Executa tcpdump en un dels PCs. Observa les trames 802.1D (protocol spanning tree) que arriben cada segon del switch.
12) Descoonecta un dels PCs i torna'l a connectar. Observa que, un cop connectat el PC, el led del switch està primer en taronja durant uns 30 segons, i després verd. Comprova fent ping que mentre el led està en taronja el PC no té connexió. Comprova amb tcpdump que mentre el led està taronja arriben trames 802.1D al PC. Durant aquest temps el protocol spanning tree actua per detectar si hi ha bucles. Comprova que passats els 30 segons, quan el led es posa verd, el PC torna a tenir connexió.

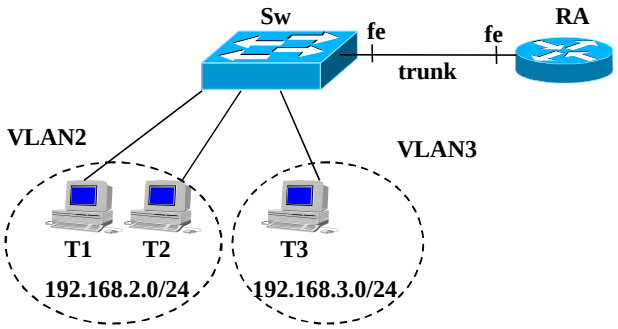
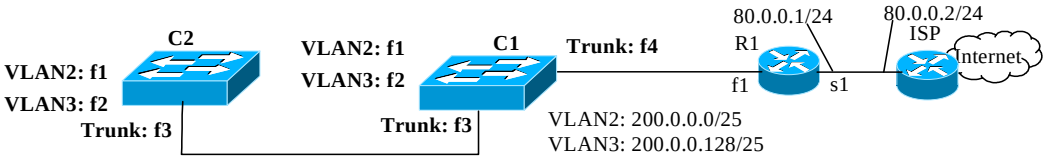


Figura 19

5.2. Ports segurs

13) Configurar un port segur en una de les estacions (ex. T_1). Configura que l'acció per defecte sigui deshabilitar el port si una altra estació es connecta. Desconnecta l'estació T_1 i connecta l'estació T_2. Observa com es desactiva el port i torna a connectar l'estació original. El comportament ha de ser el següent: si l'acció és *shutdown* del port, no acceptarà de nou l'estació original i caldrà habilitar-lo manualment (és a dir entrar a la interfície del switch i executar l'ordre *shutdown* i *no shutdown)*.

6. Informe previ



Respondre a les preguntes següents per a la xarxa de la figura.

1) Posar les ordres per configurar els commutadors i el router R1 de la figura. Suposar que l'hostid del router R1 a cada xarxa és l'adreça numèricament més baixa de la xarxa.
2) Suposar que al commutador C2 hi ha un PC1 connectat a un port de la VLAN2 i PC2 connectat a un port de la VLAN3. Per quins dispositius passaran els paquets si PC1 fa un ping a PC2?

Lab 6. TCP

1. Objectius de la pràctica

Aquesta pràctica té l'objectiu d'estudiar el comportament del protocol TCP i aprendre el funcionament de la comanda `tcpdump`, i especialment saber interpretar el bolcat d'aquesta comanda.

2. Introducció a TCP

TCP és el protocol de nivell de transport que es fa servir en Internet per a la transmissió fiable d'informació. TCP és un protocol extrem a extrem, ARQ (*Automatic Repeat reQuest*), orientat a la connexió, amb els següents objectius: (i) recuperació d'errors, per tenir una transmissió fiable; (ii) control de flux, per adaptar la velocitat entre els dos nodes que es comuniquen; i (iii) control de congestió, per adaptar la velocitat a la xarxa (i evitar així que es col·lapsi).

TCP és un protocol bidireccional, i per a cada direcció es comporta com mostra la Figura 20. Així com l'aplicació escriu la informació que ha d'enviar en el primari, TCP la guarda en un *buffer* de transmissió. Quan el *buffer* està ple, el SO bloqueja l'aplicació fins que torna a haver-hi espai. TCP va agafant aquesta informació i l'envia encapsulada dintre dels segments. A mesura que els segments arriben al secundari, la informació es guarda en un *buffer* de recepció perquè l'aplicació del secundari la vagi llegint. L'objectiu del control de flux és evitar que el *buffer* de recepció s'ompli més aviat del que es llegeix per l'aplicació del secundari (evitant així pèrdues en el receptor). Si la xarxa està congestionada les pèrdues es produiran el *buffer* d'algun dels routers del camí, i s'anomenen pèrdues per congestió.

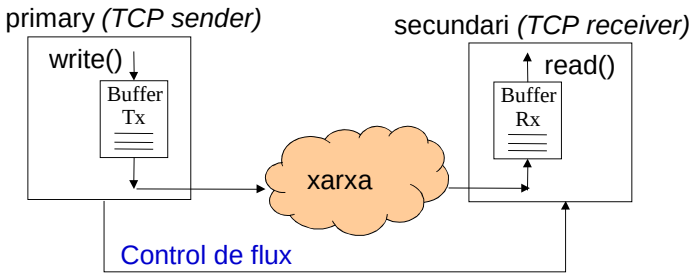


Figura 20: Nivell TCP.

La Figura 21 mostra la capçalera d'un segment TCP. A més del port font i destinació els camps més importants son els següents:

- *sequence number* (número de seqüència).
- *acknowledgement* (o simplement *ack*, confirmació).
- *Length*: mida de la capçalera en words de 32 bits.
- *flags:* U (*urgent*): es fa servir el camp *urgent pointer*. A (*ack*): es fa servir el camp d'ack; P (*push*). passar la informació el més aviat possible a l'aplicació. R (reset): avortar la connexió. S (syn): inici de la connexió. F (fin) terminació de la connexió.
- *Advertized window* (finestra advertida): es fa servir pel control de flux.
- *Options*: Les més importants son: (i) mss (maximum segment size), suggereix la mida del camp d'informació a la màquina remota (típicament la MTU de la xarxa – 40). (ii) timestamp: per a mesurar el retard d'anada i tornada (*round trip time*, RTT) i (iii) sack (*selective acks*): per a donar informació sobre els paquets perduts per a poder fer retransmissió selectiva.

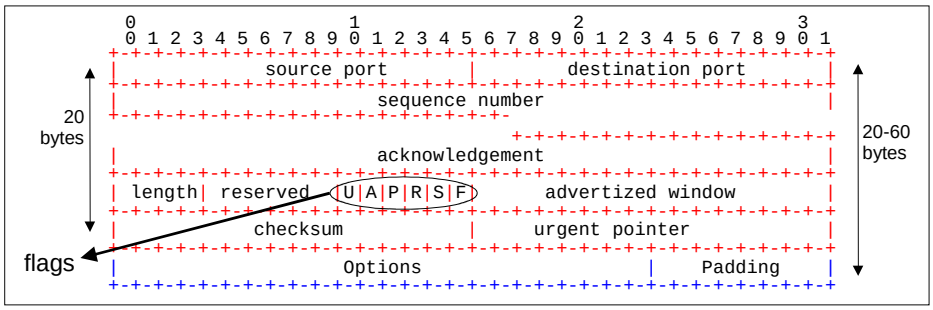


Figura 21: Capçalera TCP.

2.1. Establiment i terminació d'una connexió

La Figura 22 mostra les fases d'establiment (*three way handshake*) i terminació d'una connexió TCP. L'extrem que envia el primer segment és per definició el client. Aquest segment no porta dades, i només té activat el *flag* de syn. El servidor

contesta amb un segment amb el *flag* de syn i ack activats, confirmant l'anterior. Quan el client envïi l'ack la connexió quedarà establerta (established). La terminació es produeix després d'enviar-se segments amb el *flag* de fin activat i els seus respectius acks.

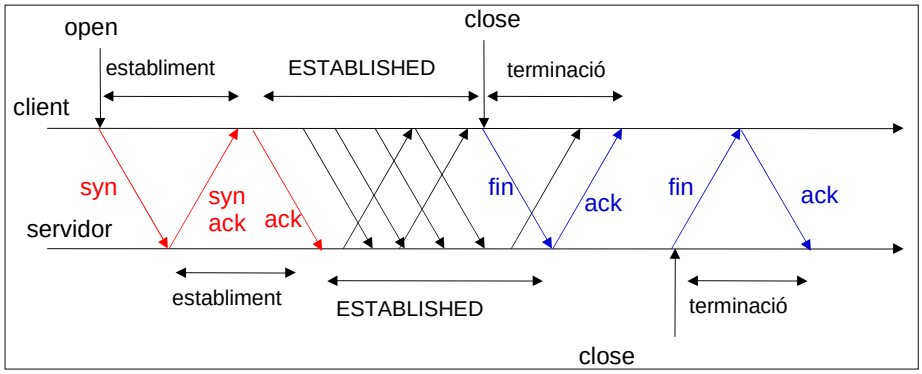


Figura 22: Establiment i terminació d'una connexió TCP.

2.2. Números de seqüència

En TCP el número de seqüència identifica el primer byte de dades que porta el segment. El primer segment porta l'*initial sequence number*, que és un número aleatori de 32 bits. A partir d'aquest valor, el número de seqüència s'incrementa amb el nombre de bytes que porta el segment (veure la Figura 23). La confirmació (ack) identifica el pròxim byte que espera rebre el secundari (i confirma tots els anteriors).

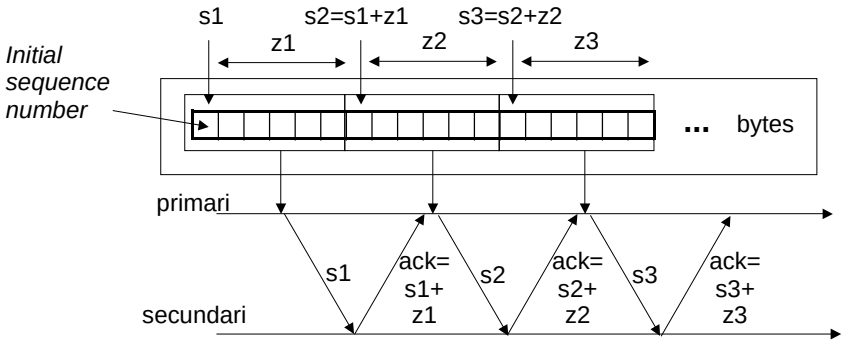
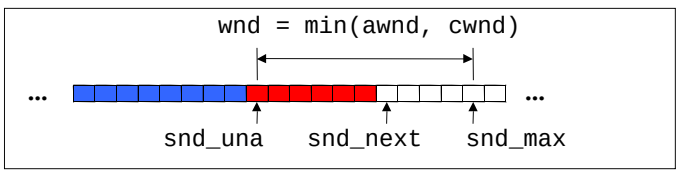


Figura 23: Evolució dels números de seqüència de TCP.

2.3. Mecanisme de finestra

TCP té un mecanisme de finestra variable que ve donada per: wnd = min(awnd, cwnd), on awnd és la finestra advertida pel node remot (control de flux) i cwnd és la finestra de congestió (veure la Figura 24). La finestra advertida s'inicia cada vegada que s'envia un segment al nombre de bytes lliures de la cua de recepció . D'aquesta manera el primari no enviarà mai més bytes dels que pot emmagatzemar el secundari. La finestra de congestió (cwnd) té l'objectiu d'adaptar-se a l'estat de congestió de la xarxa. El seu valor es calcula a partir d'un conjunt d'algorismes. A continuació s'expliquen els més importants.




```
Llegenda:
    segment enviat i confirmat
    segment enviat però no confirmat
    segment encara no enviat
  awnd Finestra advertida
  cwnd Finestra de congestió
snd_una  primer segment no confirmat
snd_next pròxim segment a enviar
snd_max  últim segment que es pot enviar
         (sino es confirmen noves dades)
```

Figura 24: Mecanisme de finestra de TCP.

2.4. Finestra de congestió

Típicament, quan vàries connexions es reparteixen un enllaç d'Internet, el mecanisme de control de congestió de TCP és el responsable d'aconseguir que cada una es quedi amb una part de la velocitat de transmissió de l'enllaç. Si les connexions transmeten "massa", aleshores hi ha pèrdues i les connexions han de transmetre "menys" per adaptar-se a la velocitat efectiva que poden aconseguir de l'enllaç. En aquesta situació, la quantitat d'informació que poden transmetre les

connexions ve donada per la mida de la finestra de congestió (cwnd). Així doncs, transmetre més o menys és equivalent a augmentar/disminuir la mida de cwnd.

TCP fa servir dos algorismes bàsics per a calcular la cwnd: el *slow start* (SS) i el *congestion avoidance* (CA). L'objectiu de l'SS és incrementar cwnd el més aviat possible a un valor on no es produeixin pèrdues per congestió. A partir d'aquest punt, cwnd es calcula amb l'algorisme CA. L'objectiu de CA és incrementar lentament cwnd per poder aprofitar més velocitat de transmissió que pugui quedar disponible. El canvi de SS a CA es produeix quan cwnd assoleix un llindar (*threshold*) mantingut en la variable *slow-start threshold, ssthresh*. La Figura 25 mostra els algorisme SS i CA. Fixeu-vos que mss és la mida d'un segment. Per tant, quan cwnd s'augmenta amb mss (com fa SS), es pot enviar un segment més sense confirmar. Quan cwnd s'incrementa amb mss/cwnd (com fa CA), s'hauran de rebre cwnd segments perquè cwnd s'incrementi amb mss.

Inicialització:

```
cwnd = mss ;
ssthresh = ∞ ;
```

Quan es rep un ack que confirma noves dades:

```
if(cwnd < ssthresh)  /* Slow Start */
      cwnd = cwnd + mss ;
else              /* Congestion Avoidance */
      cwnd = cwnd + mss*mss/cwnd ;
```

Quan s'excedeix el temps màxim d'espera de la confirmació d'un segment (*time-out*):

```
Retransmet el segment snd_una    ;
cwnd = mss ;
ssthresh = max(2, min(awnd, cwnd) / 2) ;
```

Figura 25: Algorismes de *Slow Start* i *Cogestion Avoidance*.

La Figura 26 mostra l'evolució típica de cwnd. Quan s'inicia la connexió, TCP comença amb SS i la cwnd s'incrementa ràpidament fins a la finestra advertida (awnd). Si la transmissió és dintre d'una mateixa LAN típicament no hi ha pèrdues i la finestra de TCP es mantindrà constant i igual a awnd quan cwnd arribi al seu valor. Si hi ha pèrdues (perquè la connexió travessa un enllaç congestionat, aleshores es produiran *time-outs* dels segments que no es confirmen i es retransmetran, reduint cada vegada ssthresh al valor que tenia la finestra en el moment del *time-out*. En aquest cas l'evolució de cwnd segueix una forma de dent de serra com el de la figura.

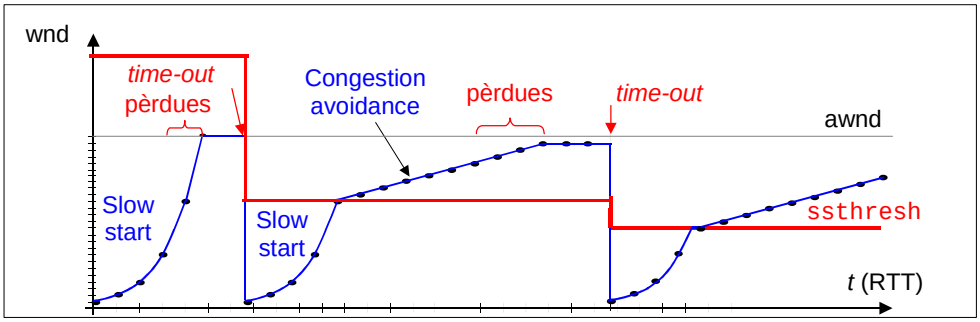


Figura 26: Evolució típica de la finestra de TCP quan hi ha un enllaç congestionat.

3. La comanda tcpdump

La comanda tcpdump permet capturar els paquets que arriben o s'envien des d'una interfície d'acord amb una certa expressió. Els paquets es capturen en el moment en que es passen o es reben pel *driver* de la interfície. Per defecte, tcpdump posa la interfície en mode promiscu, per capturar tots els paquets que hi arriben (vagin dirigits o no a la targeta on es capturen). El format bàsic de la comanda és:

```
tcpdump <opcions> <expressió>
```

Les opcions més comuns son:

- `-i <inteficie>`: captura els paquets de `<interficie>`. Per exemple: `tcpdump –i e0`
- `-n`: Perquè tcpdump no intenti resoldre les adreces als noms.
- `-x`: perquè tcpdump també faci un bolcat en hexadecimal del contingut del paquet.
- `-X`: perquè faci un bolcat en hexadecimal i ASCII del contingut del paquet.
- `-e`: perquè imprimeixi també la capçalera de nivell d'enllaç.
- `-s <n>`: perquè tcpdump capturi fins a <n> bytes de cada paquet (per defecte en captura fins a 64).
- `-c <n>`: captura <n> paquets i acaba

- `-v`: perquè sigui més *verbose* (doni més informació dels paquets capturats). Podem posar `-vv` i `-vvv` perquè doni encara més informació.

Les expressions més comuns son:

- `src|dst host|net|port <i>`: captura els paquets que tenen en el camp font|destinació el host|xarxa|port <i>. Per exemple: `tcpdump src net 10.0.0.0/24`
- `host|net|port <i>`: captura els paquets que tenen en el camp font o destinació el host|xarxa|port <i>. Per exemple: `tcpdump net 10.0.0.0/24`
- `ip|arp|tcp|udp|icmp`: captura paquets d'un d'aquest tipus.

Les expressions admeten els operadors `and`, `or` i `not`. Per exemple:

```
tcpdump –ni e0 icmp and host 10.0.0.1 and not host 10.0.0.2
```

capturarà tots els paquets icmp que tinguin com adreça font o destinació 10.0.0.1 però que no tinguin com a font o destinació l'adreça 10.0.0.2

3.1. Bolcat de tcpdump

Cada vegada que tcpdump captura un paquet, fa un bolcat (*dump*) indicant la informació que tcpdump considera més interessant. La informació que mostra en el bolcat depén del tipus de paquet capturat. La Figura 27 mostra el bolcat d'un segment TCP. Si volem més informació podem demanar a tcpdump que a més del bolcat per defecte ens faci el bolcat del paquet en hexadecimal (opció –x, i asci amb l'opció –X). La Figura 28 n'és un exemple.

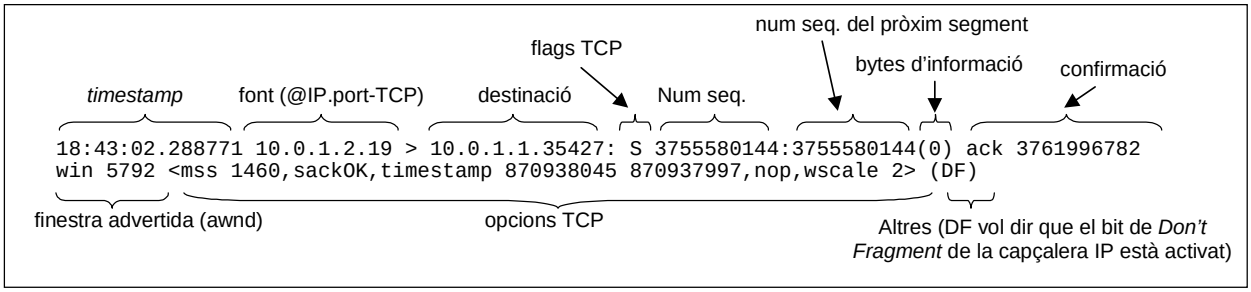


Figura 27: Bolcat d'un segment TCP.

```
xc# tcpdump -Xns 100 -i ppp0
...
18:56:02.628170 10.0.1.1.35434 > 10.0.1.2.21: P 1:17(16) ack 21 win 1460 <nop,nop,timestamp
871718463 871714750> (DF) [tos 0x10]
0x0000   4510 0044 bfe9 4000 3f06 65b8 0a00 0101        E..D..@.?.e.....
0x0010   0a00 0102 8a6a 0015 100a f0d1 105f 6243        .....j....._bC
0x0020   8018 05b4 bf2c 0000 0101 080a 33f5 5e3f        .....,......3.^?
0x0030   33f5 4fbe 5553 4552 2061 6e6f 6e79 6d6f        3.O.USER.anonymo
0x0040   7573 0d0a                                      us..
18:56:02.710769 10.0.1.2.21 > 10.0.1.1.35434: . ack 17 win 1448 <nop,nop,timestamp 871718503
871718463> (DF)
0x0000   4500 0034 2d96 4000 4006 f72b 0a00 0102        E..4-.@.@..+....
0x0010   0a00 0101 0015 8a6a 105f 6243 100a f0e1        .......j._bC....
0x0020   8010 05a8 3874 0000 0101 080a 33f5 5e67        ....8t......3.^g
0x0030   33f5 5e3f                                      3.^?
^C
```

Figura 28: Bolcat de tcpdump en hexadecimal.

Cal destacar el següent:

- El timestamp té el format hora:minuts:segons. Com que els segons es donen amb 6 decimals, tenim una resolució de microsegons (en l'exemple de la Figura 27, el paquet s'ha capturat a les 18:43 i 2 segons, 288 ms, 771 μs). El bolcat no diu si el paquet que s'ha capturat s'ha rebut o transmés. Això ho podem deduir de les adreces. Per exemple, si s'ha capturat en la màquina 10.0.1.2, aleshores el paquet s'ha transmés.

- Per a seguir millor la traça, tcpdump dóna el número de seqüència del paquet i el número de seqüència que portarà el següent paquet. D'aquesta forma, podem veure fàcilment quan es transmeten segments fora d'ordre (que normalment és una indicació de que s'han perdut segments). A més, si tcdpump captura els paquets de syn, normalitza el número de seqüència restant el número de seqüència inicial perquè sigui més fàcil de llegir (tal com mostra la Figura 28). A més, amb aquesta normalització el número de seqüència ens diu directament quants de bytes d'informacío s'han enviat. Si un paquet no porta bytes d'informació, típicament tcpdump no ens mostra els números de seqüència sino només la confirmació (segon paquet de la traça de la Figura 28).

Fixeu-vos que per parar la captura de paquets s'ha de premer CONTROL-C. En el bolcat del primer segment de la Figura 28 podem veure que la capçalera IP (la teniu en la Figura 29) té 20 bytes. Això ho podem deduïr perquè el primer byte del bolcat és 45: 4 és la versió i 5 és la mida de la capçalera en words de 32 bits (és a dir, 5 x 4 = 20 bytes). Si contem 10

grups de 4 xifres hexadecimals (els 20 bytes de la capçalera IP) arribem on comença la capçalera TCP. De la capçalera TCP (Figura 21) deduïm que 8a6a és el port font, 0015 és la destinació, 100af0d1 és el número de seqüència, 105f6243 és la confirmació i 8 és la mida de la capçalera (32 bytes). Així doncs, la capçalera TCP porta 12 bytes d'opcions (l'opció *timestamp* més el *padding*).

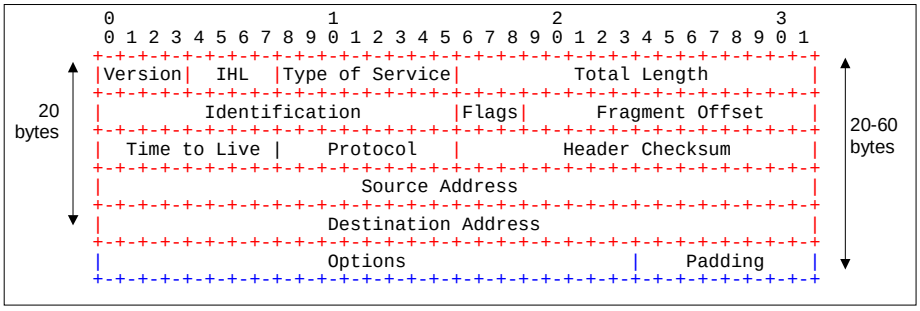
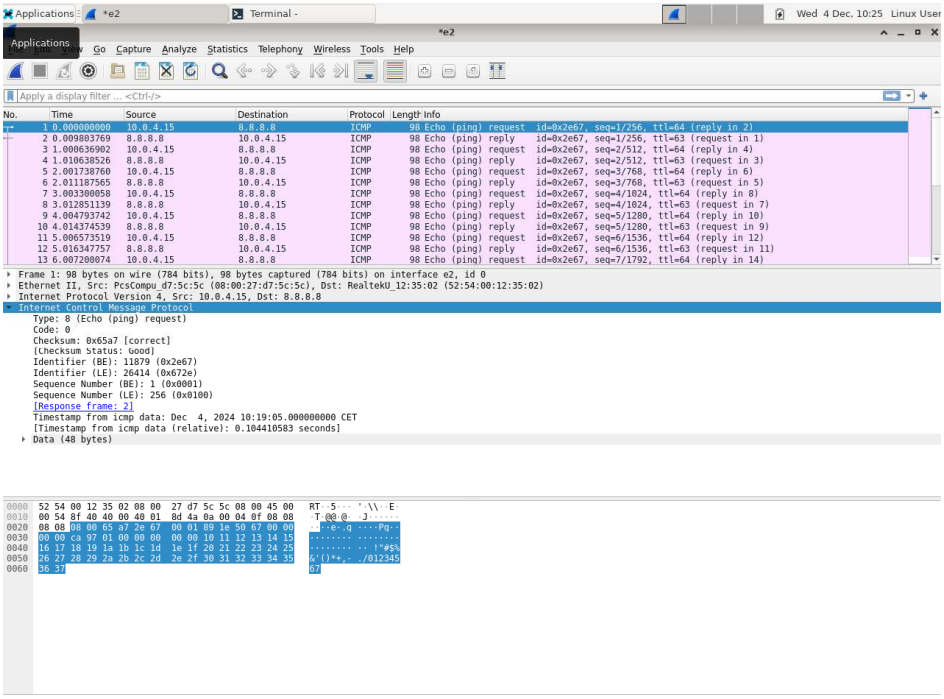


Figura 29: Capçalera IP.

4. Wireshark

Aquesta eina és semblant a tcpdump, però en format gràfic. És útil quan volem inspeccionar les capçaleres o dades d'un paquet. Convé engegar l'eina com usuari root des d'un termianl per not tenir problemes amb els permisos. Quan s'engega hem de triar la interfície on volem capturar els missatges. Wireshark obre 3 finestres, com mostra la figura de sota: en la finestra de dalt hi ha els paquets capturats (un per línia), on mostra la informació més rellavant. En la finestra d'enmig mostra informació del paquet seleccionat en la finestra de dalt. Aquí podem obrir la capçalera que volem inspeccionar. En la finestra de sota hi ha el bolcat en exadecimal. Des del menú es poden posar expressions per filtrar el tràfic, parar/engergar la captura etc.



5. Realització de la pràctica

Per a fer la pràctica capturarem segments TCP d'una connexió que hi ha entre un client i un servidor a través d'un router, tal com mostra la Figura 30. Els enllaços són ethernet.

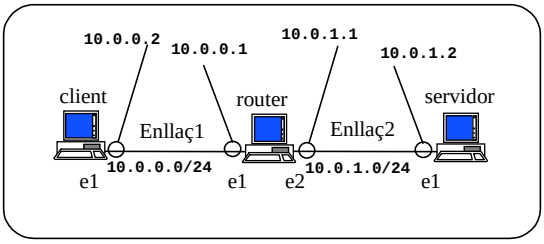


Figura 30: Topologia de la pràctica.

Per poder seguir més fàcilment la captura de les traces de TCP, en la imatge "xarxes" les opcions sack i window-scaling estan desactivades. Altrament es poden desactivar amb les comandes de la Figura 31.

```
servidor# sysctl -w net.ipv4.tcp_sack=0
servidor# sysctl -w net.ipv4.tcp_window_scaling=0
```

Figura 31: Desactivació de l'opció *sack* i *window-scaling* en el servidor.

5.1. Anàlisi dels segments d'una connexió interactiva TCP

Configura la xarxa de la Figura 30. Assegura't fent ping que hi ha connexió entre el client i el servidor.

1. Executa wireshark en el client.
2. Executa telnet 10.0.1.2 en el client, usuari xc, password xc. Observa les comandes capturades amb wireshark al mateix temps que hi ha el diàlec de la connexió telnet. Identifica el *three-way-handshake*. Observa amb els acks que els segments de SYN consumeixen un número de seqüència, i no porten cap byte de dades.
3. Observa en el bolcat de wireshark que la informació no està encriptada. Identifica en el bolcat l'usuari i el password.
4. Executa "netstat –nat" en el client i el servidor. Identifica els sockets que pertanyen a la connexió i l'estat del socket TCP. Identifica quin és el well-kwon port i el port efímer.
5. Executa exit en la connexió telnet i oberva el missatges de FIN en la desconnexió. Comprova amb els acks que els segments de FIN consumeixen un número de seqüència.
6. Després de tancar la connexió telnet observa els estats dels sockets del client i el servidor. Comprova que un dells s'ha tancat, mentre l'altra continua en TIME-WAIT durant uns 2 minuts abans de tancar-se.

5.2. Anàlisi dels segments d'una connexió bulk-TCP en una LAN

```
client# tcpdump -ni e1 port chargen
client# telnet 10.0.1.2 chargen
```

Figura 32: Captura d'una connexió al servidor de chargen.

1. Engega tcpdump, i connecta't al port de chargen executant les comandes en dos terminals diferents (Figura 32). El servidor chargen (port 19) envia una seqüència de caràcters ASCII pseudo-aleatòria a la velocitat màxima que permet l'enllaç. Nota: Per a congelar el scroll de pantalla prémer Ctrl-S, per continuar Ctrl-Q.
2. Estima la velocitat de transmissió eficaç de la connexió. Si necessites calculadora, recorda que en la barra d'aplicacions de l'escriptori en tens una. Fixa't que podem estimar la velocitat eficaç a partir dels números de seqüència del primer i últim segment d'informació de la traça capturada en el client, dividit per l'interval de temps que hi ha entre aquests dos segments.
3. Comprova si hi ha pèrdues. Justifica perquè n'hi ha, o no.
4. Observar que la connexió fa servir l'opció delayed-ack (mentre no hi ha pèrdues el receptor envia un ack cada 2 segments de dades).
5. Observa la relació que hi ha entre els acks rebuts i el número de seqüència dels segments d'informació que s'envien immediatament després de l'ack. Fixa't que en la traça capturada en el servidor la diferència augmenta de cada vegada més, mentre que en la traça capturada en el client la diferència és 0. Perquè és així?
6. Justifica que en la traça capturada en el servidor, la diferència entre el número de seqüència rebut en l'ack i el del segment d'informació que s'envia a continuació, és aproximadament el nombre de bytes d'informació de la connexió que hi ha "en vol". És a dir, bytes enviats pel servidor però que encara no han arribat al client, i per tant, que estan emmagatzemats en el router o que s'han perdut. Fixa't que la mida de la finestra que està fent servir TCP és aquesta diferència més el nombre de bytes d'informació que hi ha en els paquets que envia immediatament després (fins que rep un ack de noves dades). Relaciona l'evolució de la finestra amb el *slow start*.
7. Mira l'evolució de la finestra advertida pel client i el servidor. Perquè penses que la del client varia contínuament i la del servidor no? Mira els últims paquets de la traça capturada en el servidor. Compara la mida de la finestra advertida pel client amb la fa servir el servidor (deduïda de la traça). Quina finestra penses que està limitant la transmissió: l'advertida (awnd) o la de congestió (cwnd)?
8. En aquest experiment es tracta de que el client deixi de llegir el socket perquè s'ompli, i enviï una finestra advertida (awnd) igual a 0. Per parar la connexió amb el servidor i accedir al *prompt* de telnet prémer "Ctrl-AltGr-]" "enter" . Des del *prompt* de telnet es pot sortir amb la comanda quit o continuar al prémer la tecla enter. Connectar el client al servidor de chargen executant "`tcpdump -nli e1 port chargen`" i parar la connexió amb Ctrl-AltGr-]. Observar l'evolució de la finestra advertida que envia el client. Què fa el servidor quan la finestra val 0?
9. Fes servir netstat per veure el nombre de bytes que hi ha en la cua de recepció i transmissió dels sockets en el costat del client i del servidor. Raona les diferències.
10. Amb la connexió de chargen establerta, prova l'execució de tcpdump fent servir expressions (veure la secció 3). Per exemple, les següents comandes capturen els segments que envia o rep el servidor, és a dir, els segments de dades i acks, respectivament.

```
# tcpdump -ni e1 tcp and src 10.0.1.2
# tcpdump -ni e1 tcp and dst 10.0.1.2
```

5.3. Anàlisi dels segments d'una connexió bulk-TCP amb pèrdues

Per introduir pèrdues afegirem una cua de mida i velocitat fixades per nosaltres a la sortida de l'enllaç1 del router, tal com mostra la Figura 33. Linux permet afegir aquesta cua amb la comanda de la Figura 34 (els paràmetres de la cua es poden modificar canviant `add` per `change` en la mateixa comanda, i la cua es pot eliminar canviant `add` per `del`). Els paquets sortiran d'aquesta cua a una velocitat de 100kbps. Si els paquets arriben a una velocitat major, la cua s'omplirà fins un màxim de 10.000 bytes. Els paquets que arribin quan la cua està plena es descartaran. Fixeu-vos que la cua només afecta a un sentit de l'enllaç:

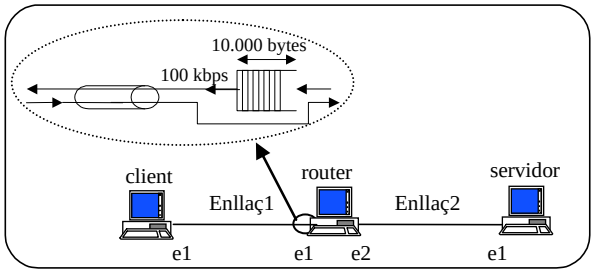


Figura 33: Cua que afegirem a la sortida de l'enllaç e1 del router.

```
router# tc qdisc add dev e1 root tbf burst 5000 rate 100kbit limit 10000
```

Figura 34: Comanda per afegir la cua de la Figura 33.

11. Configura la cua i repeteix les comandes de la Figura 32.

    a) Congela la connexió (Ctrl-S) quan hi ha primers acks duplicats i es produeix la retransmissió del segment perdut. Mira la traça capturada en el client i comprova que efectivament aquest paquet s'havia perdut.

    b) Estima quin és el nombre de bytes d'informació que hi ha "en vol" quan es produeixen pèrdues. Comprova que és major de 10.000 bytes. Comprova que després de la retransmissió la finestra de tcp poc a poc va augmentant fins que torna a haver-hi pèrdues.

12. Estima quina és la velocitat eficaç. Comprova que és aproximadament 100 kbps.

13. Estima que val el RTT en el three-way-handshaking i quan es produeixen pèrdues. Perquè és diferent? Estima quin hauria de ser el RTT degut al retard que introdueix el buffer que hem afegit amb la comanda tc en el router. Comprova que quan es produeixen pèrdues el RTT coincideix aproximadament amb el retard que has calculat.

6. Informe previ

El següent bolcat mostra el timestamp del primer paquet, i els últims 5 paquets d'una captura amb tcpdump. A la vista del bolcat, respon les següents preguntes:

```
1. 18:37:12.234583
2. …
3. 18:38:28.739407 IP 147.83.30.137.22 > 80.102.159.44.1035: P 4672:4801(129) ack 4805119 win 32480
4. 18:38:28.739652 IP 80.102.159.44.1035 > 147.83.30.137.22: P 4805119:4805151(32) ack 4801 win 2092
5. 18:38:28.739729 IP 80.102.159.44.1035 > 147.83.30.137.22: F 4805151:4805151(0) ack 4801 win 2092
6. 18:38:28.851394 IP 147.83.30.137.22 > 80.102.159.44.1035: F 4801:4801(0) ack 4805152 win 32480
7. 18:38:28.851458 IP 80.102.159.44.1035 > 147.83.30.137.22: . ack 4802 win 2092
```

1) Quina és l'adreça IP del client i del servidor?
2) Donar un possible bolcat pel que falta en la primera línia.
3) Quants bytes d'informació (contingut del camp payload) han enviat exactament el client i el servidor?
4) Estimar la velocitat eficaç.
5) Quins son els estats de TCP per els que passa el client i el servidor durant els 5 últims paquets del bolcat?

Lab 7. Domain Name System (DNS)

1. Introducció

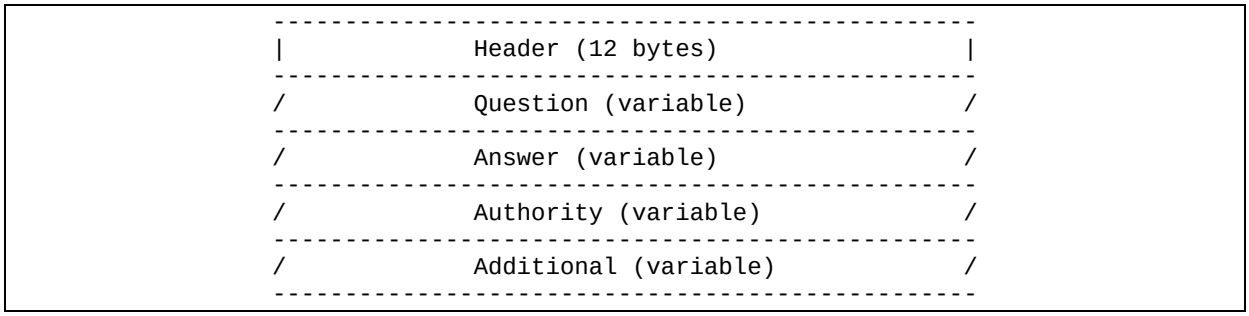
En aquesta pràctica s'aprofundirà en els protocol DNS. Per dur a terme els experiments que es detallen en l'enunciat utilitzarem una de les màquines del laboratori com a servidor DNS.

2. DNS

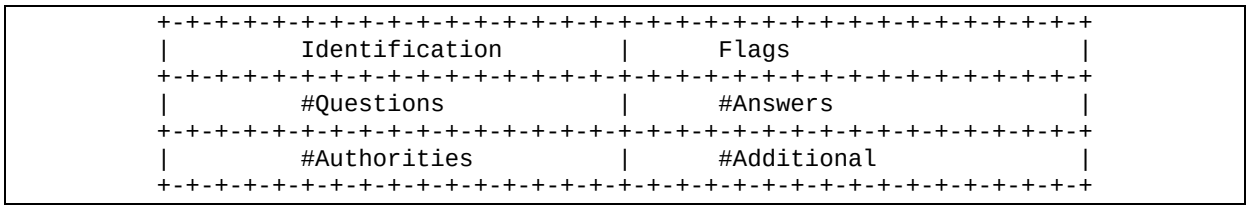
Aplicació client-servidor que es fa servir per la resolució de noms (conversió d'un nom en una adreça IP entre d'altres). Consisteix en una base de dades distribuïda. Les entrades s'anomenen Resource Records (RR) i poden ser de tipus:

- SOA: Start Of Authority.
- NS: NS name.
- MX: the domain mail exchange.
- A: A host address.
- CNAME: Canonical Name Record. E.g. the real hostname of www.foo.org is server.foo.org.

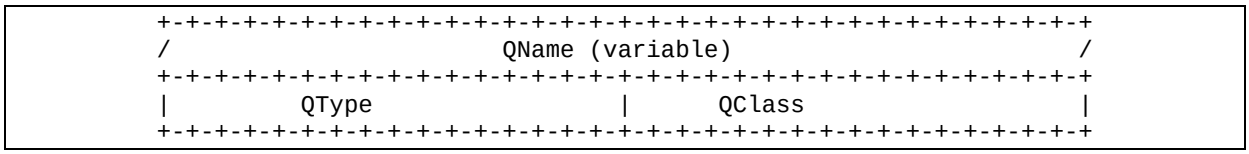
Tots els missatges DNS tenen el format:



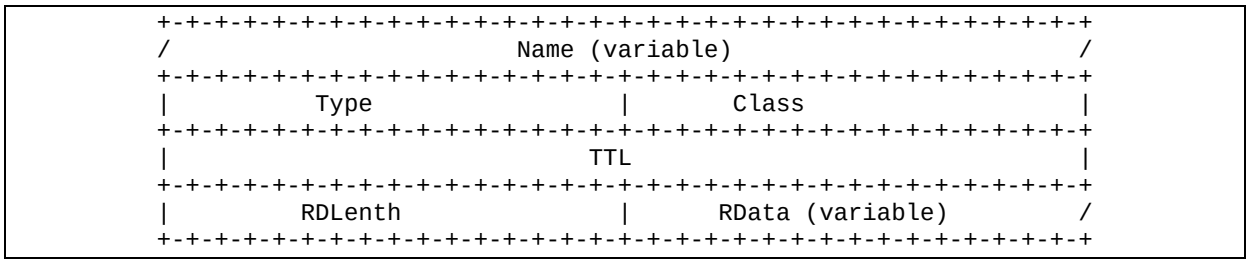
On la capçalera (header) és:



El camp question:



I els camps Answer, Authority i Additional són RRs:



3. Comandes bàsiques

Per a la realització de la pràctica es faran servir les següent comandes:

3.1. Wireshark

En aquest laboratori utilitzarem aquesta eina per a veure com circulen per la xarxa de l'aula els missatges de nivell aplicació que s'intercanviaran durant la realització de la pràctica.

Per a posar en marxa el wireshark cal que executeu des d'un terminal com a usuari root la comanda 'wireshark'. Es necessari ser root perquè l'eina monitoritza tota la informació que circula per la interfície de xarxa independentment del procés i/o usuari que generi les dades, de manera que permet espiar l'activitat de xarxa dels altres usuaris que treballin en l'equip on s'executa el wireshark.

Un cop el programa és en marxa, podem anar al menú 'capture->interfaces' (o directament fer click en la icona que hi ha més a l'esquerra), triarem la interfície (eX) que disposi d'adreça IP, i polsarem 'capture' per a inicial la captura de paquets (veure la Figura 35). Un cop hagi finalitzat l'activitat que volem capturar, caldrà prémer el botó 'stop'.

Figura 35: Captura amb wireshark.

Un cop feta la captura podem filtrar els missatges d'un protocol concret, com mostra la Figura 36. Una característica molt important és la capacitat de poder seleccionar els paquets que volem capturar. Això es fa amb el quadre de text que hi ha al costat de Filter. Aquí hi podem posar una expressió com ara "dns" per capturar missatges que porten informació de nivell d'aplicació relativa al protocol http, o expressions més sofisticades com ara udp.port==53, per a capturar els segments UDP que porten el port font o destinació igual a 53.

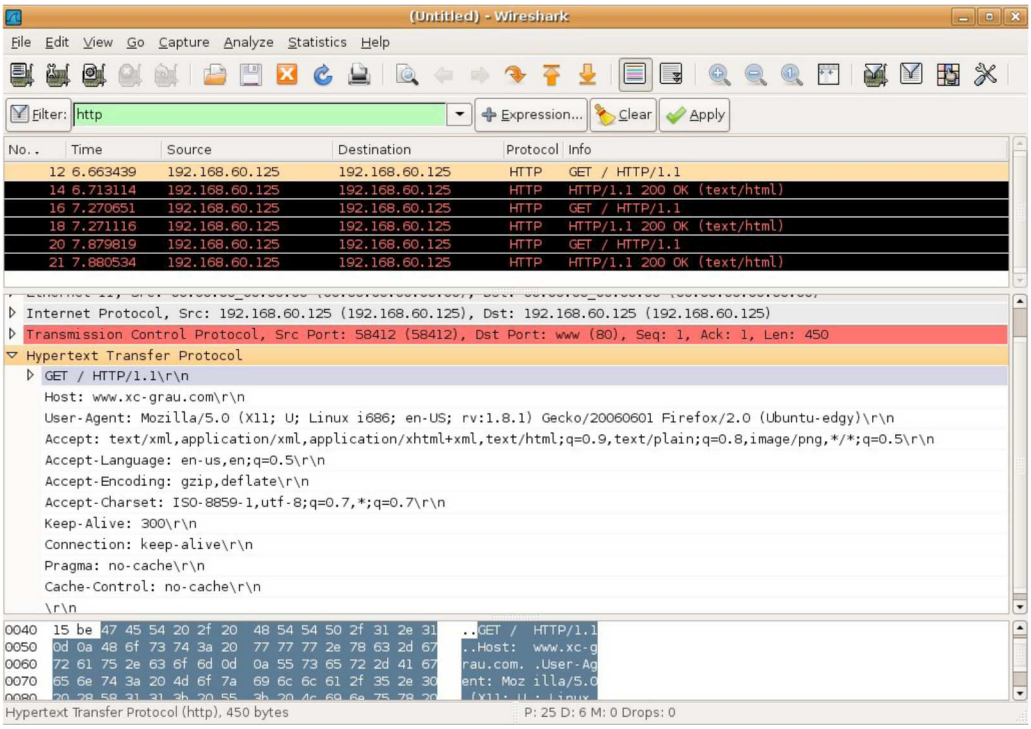


Figura 36: Filtre dels missatges http amb wireshark.

3.2. Comanda `dig`

Comanda per interactuar amb un servidor de noms (NS). S'invoca com:

```
dig [@server] [-p port#] [-t type] [-v] [-x addr] [name] [type] [opt...]
```

A continuació una explicació dels arguments:

server	El nom o l'adreça IP del servidor de noms a consultar. Si no es proporciona cap argument de servidor, dig consulta el servidor de noms a /etc/resolv.conf
name	El nom del registre del recurs que s'ha de cercar. Si un nom no té un punt final, la llista de cerca s'utilitza per qualificar (completar) el nom.

type	Cal el tipus de consulta: ANY, A, MX, PTR, SOA, NS, etc. pot ser qualsevol tipus de consulta vàlida.
opt	Diverses opcions:
+[no]recurse	Commutar la configuració del bit RD (recursió desitjada) a la consulta. Aquest bit està activat per defecte.
+[no]trace	Commutar el seguiment del camí de la delegació des dels servidors de noms arrel per al nom que s'està buscant.
+[no]short	Proporcionar una resposta concisa. El valor predeterminat és imprimir la resposta detallada.

Algunes preguntes útils:

Quina és l'adreça IP d' lloc web?
$ dig www.ac.upc.edu

Com s'identifiquen els servidors de noms associats amb un domini?
$ dig NS upc.edu +short
$ dig NS edu. +short

Quins servidors de correu electrònic són els responsables d'un domini?
$ dig MX ac.upc.edu +short

Quin és el nom de domini associat a l'adreça IP (cerca inversa d'IP)
$ dig -x 1.1.1.1 +short
$ dig -x 147.83.0.1 +short

Quin és el camí de delegació per a qualsevol zona DNS (com funciona el DNS)
$ dig upc.edu +trace

Trobar respostes de resolvers DNS cau específics (p.ex. Cloudflare [1.1.1.1], UPC [147.83.0.1])
$ dig A www.upc.edu @1.1.1.1 +short

Trobar el temps de caducitat de la memòria cau (TTL) per a DNS RR
$ dig A www.upc.edu +nocmd +noall +answer +ttlid

3.3. El resolver

El *resolver* és una biblioteca que proporciona accés al DNS a qualsevol host. El seu fitxer de configuració "/etc/resolv.conf" conté informació per resoldre els noms contactant amb servidors DNS. Per exemple, si l'adreça d'un servidor DNS és 192.168.60.125:

```
root@aula01:/# cat /etc/resolv.conf
search xc.test
nameserver 192.168.60.125
```

on "nameserver" és l'adreça IP del servidor de noms local i "search" és el domini predeterminat per a les consultes. Si una resolució falla i hi ha diversos servidors de noms (diverses línies de "nameserver"), aquests es demanen seqüencialment. El domini predeterminat s'afegeix si el nom a resoldre no està totalment qualificat (complet).

3.4. El servidor Bind

Berkeley Internet Name Domain (BIND) és el programari de servidor DNS més popular. Depèn de dos tipus de fitxers per funcionar correctament: un fitxer de servidors arrel i un fitxer de zona.

El fitxer de servidors arrel, conté una llista de les adreces IP dels servidors DNS arrel a Internet. Aquests servidors proporcionen informació sobre els dominis de primer nivell (TLD) com .com, .net, .org, etc. Quan el servidor DNS BIND rep una sol·licitud d'un nom de domini que no té autoritat, consultarà els servidors arrel per determinar quins dels servidors arrel ha de consultar per obtenir l'adreça IP del següent nivell del nom de domini. Aquest fitxer a "/etc/bind/db.root" no s'ha de modificar. Forma part de la instal·lació de bind.

El fitxer de zona conté informació, registres de recursos (RR), sobre un domini (zona) per al qual és autoritzat un cert servidor DNS BIND. Aquest fitxer inclou registres que assignen noms de domini a adreces IP i altra informació com ara el servidor de correu i la informació del servidor de noms. El servidor DNS BIND utilitza aquesta informació per respondre a les consultes DNS d'aquest domini.

Per exemple, el fitxer db.grupX.xc del laboratori té la informació que es mostra a la figura següent. Tingueu en compte que aquest fitxer és una plantilla d'un fitxer de zona, on la X s'ha de canviar al valor corresponent, tal com s'explica a continuació. El tipus RR SOA (Start Of Authority) té paràmetres de configuración seguit d'altres RR.

```
@       IN      SOA     ns.grupX.xc.test hostmaster.grupX.xc.test (
                                1         ; Serial
                           604800         ; Refresh
                            86400         ; Retry
                          2419200         ; Expire
                           604800 )       ; Negative Cache TTL
        IN      NS      ns.grupX.xc.test.
        IN      MX      10 mail1.grupX.xc.test.
        IN      MX      20 mail2.grupX.xc.test.
;
ns.grupX.xc.test.    A       192.168.60.X ;Adreça IP del NS
mail1.grupX.xc.test. A       192.168.60.X ;Adreça IP del MX
mail2.grupX.xc.test. A       192.168.60.X ;Adreça IP del MX
www.grupX.xc.test.   CNAME   pcserver.grupX.xc.test.
smtp.grupX.xc.test.  CNAME   pcserver.grupX.xc.test.
pop3.grupX.xc.test.  CNAME   pcserver.grupX.xc.test.
pcserver.grupX.xc.test.      CNAME   pcX.grupX.xc.test.
pcX.grupX.xc.test.        A       192.168.60.X ;Adreça IP de PCX
```

4. Realització de la pràctica

L'objectiu de la pràctica és configurar una autoritat del subdomini grupX.xc.test d'un hipotètic domini xc.test, tal com mostra la Figura 37. A partir d'ara X és el nombre del PC que fa de servidor de noms. Per exemple, si aquest és 114, aleshores el subdonini serà grup114.xc.test. En un cas real existiria l'autoritat xc.test, que apuntaria cap a grupX.xc.test i seria accessible a través d'un root-server. Tenim un domini de prova grupX amb el seu servidor, i el servidor xc.test no existeix. Per tant, els noms de la zona no es poden resoldre des de la resta d'Internet.

Per tal de poder realitzar la pràctica oportunament, caldrà que cada grup d'estudiants utilitzi 2 PCs. Un que serà l'autoritat de **grupX**.xc.test, i un que farà de host d'aquest domini (**pcX.grupX**.xc.test) per a fer peticions, tal com mostra la figura.

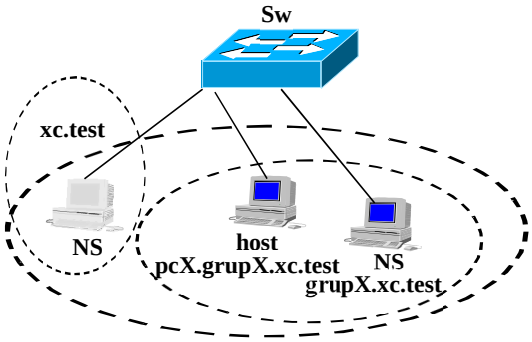


Figura 37. Xarxa a configurar. Només s'han de configurar pcX.grupX.xc.test i grupX.xc.test.

4.1. Configuració de la xarxa

1) Configurar els 2 PCs de la Figura 37 (el host pcX.grupX.xc i el servidor de noms grupX.xc.test) executant el client de dhcp (udhcpc). A continuació en pcX.grupX.xc.test executar "killall udhcpc", per matar el client de dhcp, altrament actualitzarà períodicanent el fitxer resolv.conf, esborrant els canvis que s'hagin fet al fitxer. A continuació modifiqueu el fitxer /etc/resolv.conf de pcX.grupX.xc.test perquè es faci servir el servidor de noms grupX.xc.test:

```
search grupX.xc.test
nameserver 192.168.?.?
```

on 192.168.?.? és l'adreça IP que el servidor de DHCP ha assignat al servidor de noms local grupX.xc.test. IMPORTANT: la xarxa 192.168.0.0/24 que hi ha en aquest apartat i els següents és d'exemple. Heu de fer servir la xarxa/IPs que assigna el servidor de DHCP.

4.2. Configuració del servidor de DNS de la subzona

Els fitxers de configuració es troben en la carpeta /home/xc/dns, que serà la carpeta de treball a partir d'ara.

2) Editar el fitxer 'named.conf' per indicar el domini i on es troba el fitxer de zona. Hi trobareu:

```
zone "grupX.xc.test" IN {
  type master;
  file "/home/xc/dns/db.grupX.xc";
};
```

On s'ha de canviar la X de `grupX.xc.test`. per el nombre del PC que es fa servir com a servidor de noms.

3) Modificar el fitxer ‘db.grupX.xc’.canviant les X que calgui igual que abans (és a dir, amb el nombre del PC que fa de servidor).
4) Un cop fet això, engegar el servidor de noms amb la comanda ‘run_named.sh’ com a root. Cal executar-ho cada cop que es modifiqui named.conf o db.grupX.xc. Comprova que està engegat executant “ps aux | egrep named”. Si named no està corrent, mira si hi ha hagut errors executant: “tail –f /var/log/messages”.

4.3. Observació del comportament del protocol DNS

5) Fent servir l’eina “dig” contesteu les següents preguntes. Quins registres heu consultat en cada ocasió? Assumiu que grupX i pcX són els corresponents al vostre grup:
   a. De quina màquina n'és àlies 'www.grupX.xc.test'?
   b. Quina és l’adreça IP del servidor ‘www. grupX.xc.test’?
   c. Quins són els servidors de correu del domini ‘grupX.xc.test’?
6) Engegueu el wireshark en l’equip que fa de servidor de DNS de la vostra zona per a fer captures del tràfic DNS que genereu, i observeu què passa quan es fan les peticions de l’apartat anterior. Navega a través dels camps dels paquets capturats per respondre el següent:
   a. Quants missatges es generen en cada resolució?
   b. És una resolució recursiva o iteractiva?
   c. Identifica les adreces de tots els servidors de noms que es consulten. S’ha fet servir algun root-server?
   d. Investiga el contingut dels camps (question, answer, authority, additional) de totes les respostes.
7) Captura el tràfic DNS que es genera quan es resol el nom www.microsoft.com. Navega a través dels camps dels paquets capturats ara per respondre el següent:
   a. Quants missatges es generen?
   b. És una resolució recursiva o interactiva?
   c. Identifica les adreces de tots els servidors de noms que es consulten. S’ha fet servir algun root-server?
   d. Investiga el contingut dels camps (question, answer, authority, additional) de totes les respostes enviades per els servidors.
   e. Quantes adreces IP representen el nom que s’ha resolt? Quins son els noms canònics de les adreces?
8) Fes servir la opció debug de dig (opció +trace). Repeteix la resolució de www.microsoft.com. Compara la informació proporcionada per dig amb el contingut dels missatges.
9) Canvieu el mode del vostre client (dig) a no recursiu (+norecurse). Què canvia quan ara repetiu la resolució de www.microsoft.com?
10) Prova de fer la resolució d’un nom configurat per un altre grup del laboratori. Quins missatges es generen? És possible fer la resolució? Perquè?
11) Obrir el navegador web i el wireshark en el host. Connecteu-vos a www.fib.upc.edu i observar les resolucions DNS que es generen.

5. Informe previ

1. Quins fitxers caldrà canviar de l’equip que faci de servidor de DNS de la subzona configurada en la pràctica?
2. Què és el mode recursiu i el mode iteratiu de DNS?
3. Quants missatges i quin contingut és d’esperar que es generin al fer la resolució del nom www.grupX.xc.test?
4. Quants missatges i quin contingut és d’esperar que es generin al fer la resolució del nom www.microsoft.com?