Facultat Informàtica de Barcelona
FIB
Grau en Enginyeria Informàtica

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

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

Setembre de 2020

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 slitaz (http://www.slitaz.org).

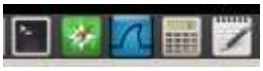
1. Informació bàsica

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).

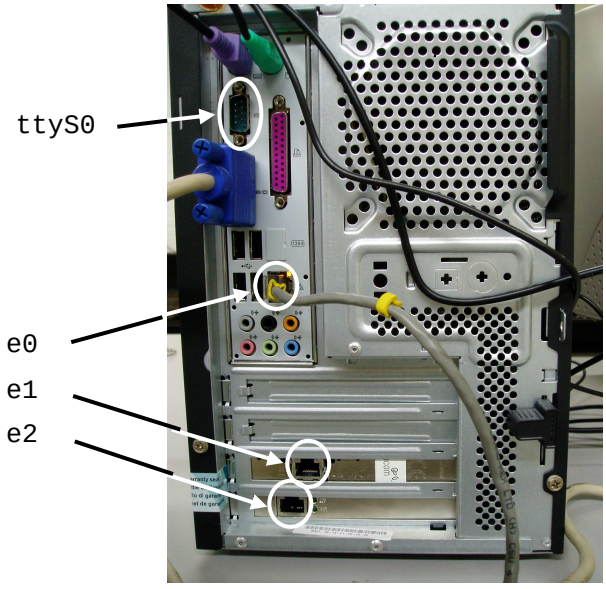


Figura 1: Ports de comunicació dels PCs del laboratori que farem servir en les pràctiques.

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 següent enllaç podeu trobar una màquina virtual (MV) creada des de VirtualBox (https://www.virtualbox.org) on hi ha instal·lada una distribució de Linux com la que hi ha en els PCs del laboratori.

http://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.

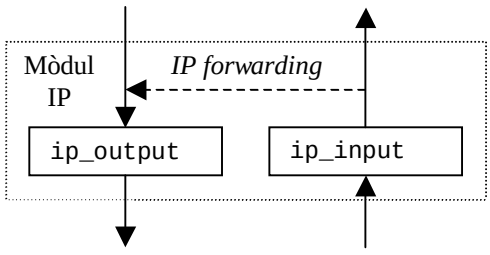


Figura 2: Funcions `ip_output` i `ip_input`.

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 down
```

Comanda 2: Desactivació d'una interfície.

I per activar-la de nou:

```
ifconfig e0 up
```

Comanda 3: Activació d'una interfície.

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

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

```
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.

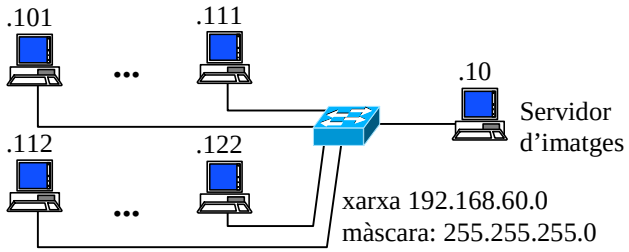


Figura 3: Xarxa del laboratori D6003.

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.

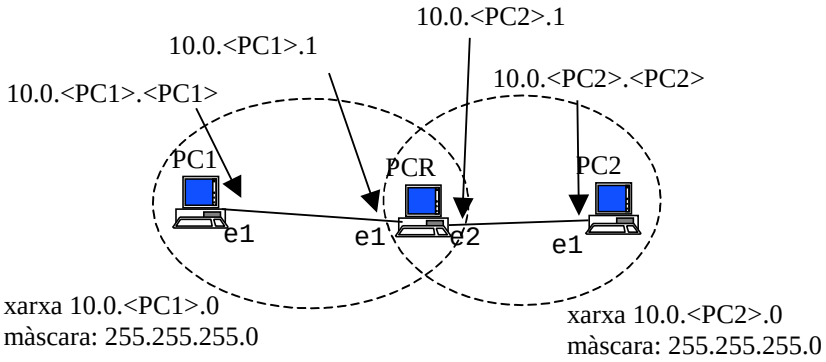


Figura 4: Topologia amb 1 router.

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.

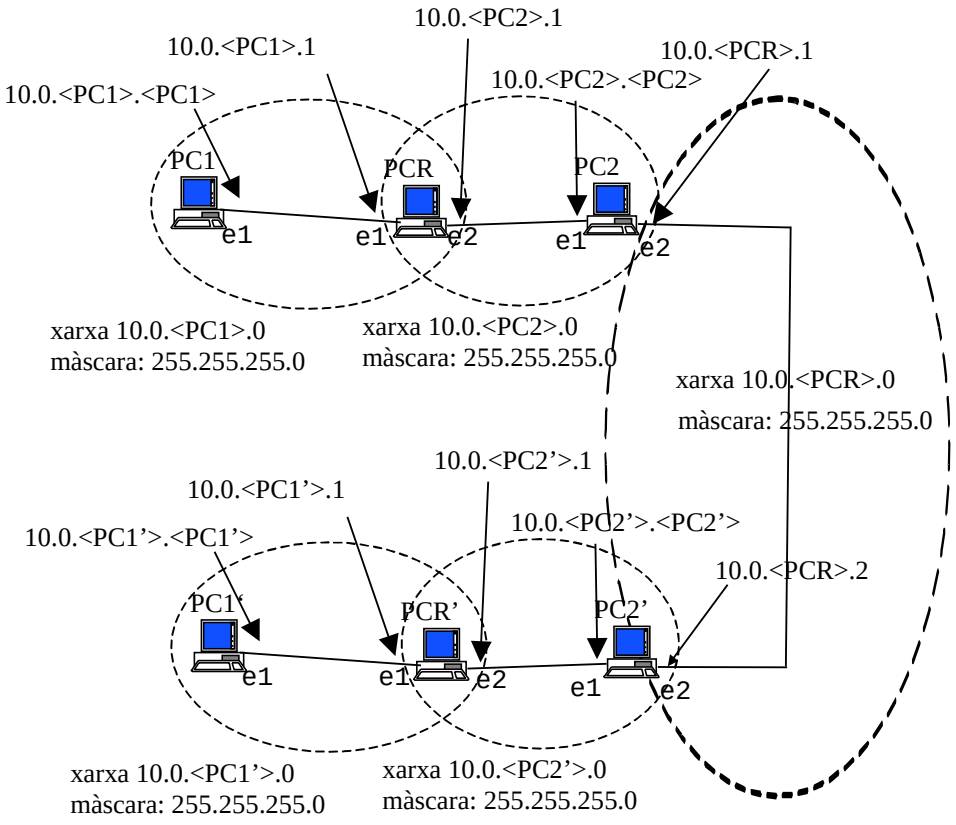


Figura 5: Topologia ajuntant les xarxes de 2 grups.

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

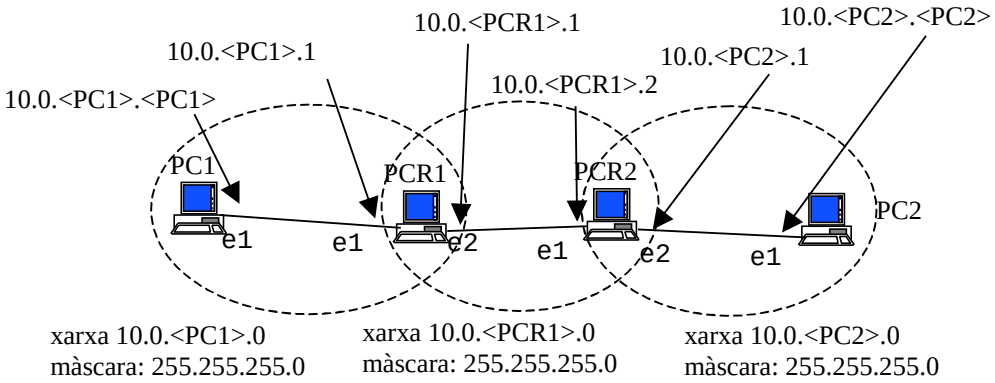
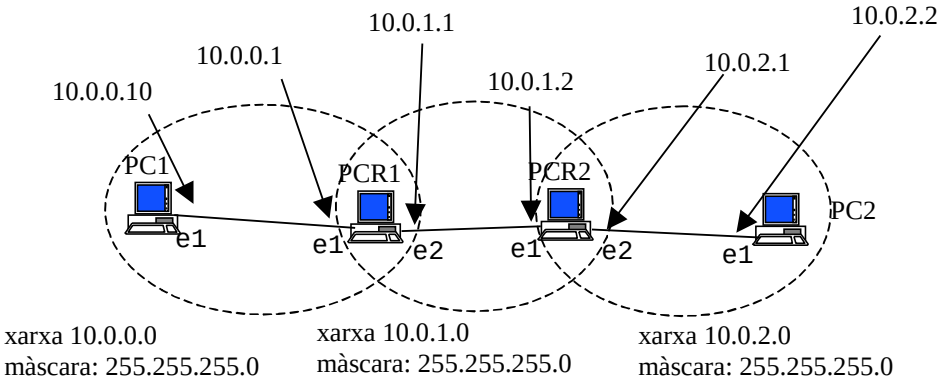


Figura 6: Topologia amb 2 routers.

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

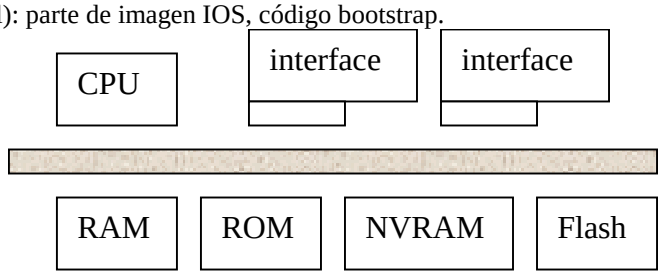
1. Objetivo de la práctica

El objetivo de la práctica es conocer los conceptos básicos de la configuración de routers con sistema operativo IOS ("Internetworking Operating System") del fabricante de routers Cisco Systems.

2. Estructura de un router

Un router IP es un computador especializado en conmutar datagramas IP. Dependiendo de las prestaciones que deba ofrecer, su estructura interna es más o menos compleja y especializada, pero para los modelos de gama baja, podemos pensar en una estructura similar a la de un PC: CPU, memoria, buses e interfaces de red. Para el almacenamiento de datos es habitual utilizar memoria ROM, memoria flash y memoria RAM y RAM no volátil (NVRAM):

- RAM: código, tablas de encaminamiento, buffers, cache ARP, etc.
- NVRAM (no volátil): fichero de configuración "startup-config".
- Flash (no volátil): Imagen del IOS
- ROM (no volátil): parte de imagen IOS, código bootstrap.



Los sistemas operativos de los routers comerciales están especialmente diseñados para facilitar las tareas de conmutación de paquetes, la ejecución de algoritmos de encaminamiento, configuración de interfaces, etc. Un ejemplo de este tipo de sistemas operativos es el IOS. El IOS tiene una arquitectura simple y normalmente ocupa un espacio de memoria reducido. Cuando encendemos un router, se ejecuta un programa de bootstrap cargado en la ROM que testea el sistema y carga en la RAM una imagen del IOS, normalmente desde la memoria flash.

Configuraremos el router utilizando un interface de comandos en línea (CLI). Normalmente se hace a través de una conexión por la línea serie conectada al puerto CONSOLE del router, usando por ejemplo la aplicación HYPERTERMINAL en Windows, MiniCOM en Linux, etc. Los parámetros necesarios para conectarse son los siguientes: Baud Rate 9600 bps, 8 bits/carácter, 1 bits de Stop, No paridad y No control de flujo Hardware.
La configuración activa del router se encuentra en un fichero llamado `running-config`. Si apagamos el router, dicha configuración se perdería y no estaría presente al volver a activar el router. Podemos guardar dicha configuración en un archivo de configuración (`startup-config`) que normalmente se graba en una memoria NVRAM. Al arrancar el router, la configuración que se activa es la guardada en el archivo `startup-config`.

También podemos configurar el router accediendo por telnet o utilizar un interfaz web para configurar el router. Asimismo tanto la imagen del IOS como el archivo de configuración se pueden obtener de un servidor de tftp.

3. Modos de configuración

Los router con IOS disponen de un conjunto de modos llamados de configuración que permiten la visualización y configuración del router. Los modos de configuración son los siguientes:

- **Modo BOOT o ROM monitor**: se usa en casos de emergencias (prompt típicamente `rmon`) como puede ser la recuperación de un password, de un registro de configuración, etc
- **Modo de SETUP**: permite una configuración por menú sencilla y básica del router
- **Modo USER EXEC**: es el modo de visualización sin privilegios (prompt `R>`)
- **Modo PRIVILEGED EXEC**: modo de visualización con privilegios (prompt `R#`)
- **Modo de Configuración Global o CONFIGURE**: permite configurar aspectos sencillos del router como pueden ser la configuración del nombre del router, passwords, etc (prompt `R(config)#`)
- **Modo de configuración específicos**: permiten configurar protocolos, interfaces o en general aspectos más complejos del router (prompt `R(config-if)#`, `R(config-route)#`, `R(config-line)#`, etc)

Al arrancar el router podemos pasar al modo SETUP, que permite dar una primera configuración al router cuando éste carece de una configuración preestablecida, o bien pasar al modo USER EXEC, cuando el router sí dispone de una configuración preestablecida.

El primer mensaje que emitirá el router cuando conectemos con es será:

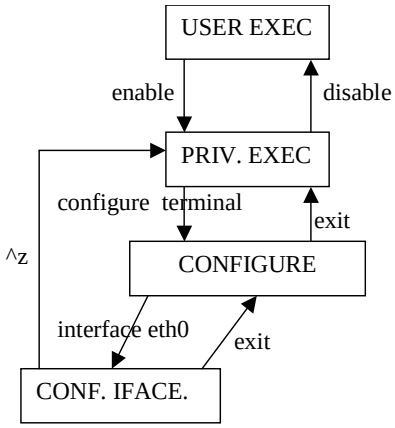
Continue with the configuration dialog [yes/no]: **no**

A lo que habrá que contestar **NO.**

En modo USER EXEC podemos consultar aspectos básicos de la configuración del router[2]. Para consultar aspectos más críticos de la configuración del router debemos pasar a modo PRIVILEGED EXEC. Para pasar de modo USER EXEC a modo PRIVILEGED EXEC es necesario usar un password (que se conoce como "*enable secret password*" que se puede establecer desde el modo CONFIGURE ejecutando `enable secret <passwd>`)

Desde los modos USER EXEC y PRIVILEGED EXEC no podemos modificar la configuración del router. Para hacerlo debemos pasar del modo PRIVILEGED EXEC al modo de configuración general (CONFIGURE). Desde allí podemos configurar aspectos generales del funcionamiento del router o pasar a modos de configuración específicos de cada interfaz, algoritmo de encaminamiento, etc.

En la siguiente figura se muestran los diferentes modos de configuración junto con los principales comandos necesarios para cambiar de un modo a otro.



Cuando estamos en modo USER EXEC el prompt que nos muestra el router es "**>**". Cuando estamos en PRIVILEGED EXEC el prompt es "**#**" y en el modo de configuración global el prompt es **`(config)#`**.

Por ejemplo:

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

Como ya hemos mencionado, los cambios de configuración que se realicen en el modo de configuración global o específico se guardan sobre un archivo de configuración residente en la RAM del router llamado "`running-config`". Este fichero puede ser visualizado desde el modo de configuración privilegiado con el comando `''show running-config''`. Si el router se apagase, estos cambios se perderían al estar almacenados en RAM. Para que no se pierdan y pasen a estar permanentemente guardados en una memoria NVRAM hay que copiar el archivo "`running-config`" (RAM) en el archivo "`startup-config`" (NVRAM). Ello se puede hacer desde el modo PRIVILEGED EXEC con el comando "`copy running-config startup-config`".

EN ESTE CURSO NO VAMOS A GUARDAR LA CONFIGURACIÓN DEL ROUTER ENTRE SESIONES, DE MANERA QUE NO SE DEBE REALIZAR LA COPIA DE CONFIGURACION MENCIONADA

[2] Con el commando `?` podemos obtener un listado de los comandos que se pueden ejecutar en cada modo.

4. Consulta del estado (comandos show)

Podemos consultar el estado de un router mediante los comandos `show`. Dependiendo del tipo de información que queremos consultar, el comando es ejecutable desde modo USER EXEC o bien necesitamos los privilegios del modo PRIVILEGED EXEC. Por ejemplo:

show ip interface brief muestra el estado de los interfaces, sus nombres y su configuración.

show running-config muestra el fichero de configuración que está activo en el router

show startup-config muestra el fichero de configuración que está grabado en la NVRAM

show ip <parameter> muestra los parámetros asociado a la configuración del protocolo IP. Por ejemplo, la tabla de encaminamiento IP se consulta con `show ip route`

show interfaces <nombre_interface> muestra los parámetros asociados al interface

La tabla de encaminamiento es una información que no se considera privilegiada y que puede ser consultada desde el modo usuario USER EXEC. Sin embargo, el contenido de los ficheros de configuración si que se consideran privilegiados y sólo pueden ser visualizados desde el modo PRIVILEGED EXEC.

5. Configuración básica del Router

Configurar el nombre y mensajes de entrada (se muestra al conectarse al router)

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

Para poder acceder al router con telnet hay que asignar un password a los terminales vty:

```
R(config)# enable password cisco → para hacer telnet hace falta configurar password
R(config)# line vty 0 4 → configuración de 5 terminales activos para telnet
R(config-line)# password cisco
R(config-line)# exit
```

6. Configuración de los interfaces

Desde el modo de configuración podemos pasar a configurar los interfaces. Por ejemplo, para configurar un interface ethernet podemos hacer:

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

Recordad, que con el comando *show ip interface brief* podréis consultar los nombres de los interface.

El comando "`no shutdown''` es necesario para activar la interfaz. Por defecto, al arrancar el router todos los interfaces están desactivados. El comando `''shutdown''` en su defecto desactivaría administrativamente una interfaz.

Las interfaces serie están diseñadas para que en la situación más normal se conecten a una operadora de telecomunicaciones a través de un DCE (ej.; un MODEM o una Terminación de Red, TR). El DCE es el que normalmente da reloj y por tanto fija la velocidad de modulación y por consiguiente de transmisión.

Si se conectan dos puertos serie de router (DTE-DTE) hay que usar un cable cruzado. Además uno de los dos puertos tiene que actuar como DCE dando reloj. En principio desde el punto de vista de router cualquiera de los dos puede actuar de DCE, así que lo importante es que conector del cable es el que marca que puerto es DCE.

7. Interfaces serie

Los nodos de una red pueden clasificarse en dos grandes grupos: equipo terminal de datos (DTE) y equipo de comunicación de datos (DCE). Los DTE son dispositivos de red que generan el destino de los datos: los PC, routers, las estaciones de trabajo, los servidores de archivos, los servidores de impresión; todos son parte del grupo de las estaciones finales. Los DCE son los dispositivos de red intermediarios que reciben y retransmiten las tramas dentro de la red; pueden ser: conmutadores (switch), concentradores (hub), repetidores o interfaces de comunicación.

Las interfaces serie están diseñadas para que en la situación más normal se conecten a una operadora de telecomunicaciones a través de un DCE (ej.; un MODEM o una Terminación de Red, TR). El DCE es el que normalmente da reloj y por tanto fija la velocidad de modulación y por consiguiente de transmisión.

Todos los cables usados para crear un enlace DTE-DCE son directos, los cables usados para DTE-DTE y DCE-DCE son cruzados.

Si se conectan dos puertos serie de router (DTE-DTE) hay que usar un cable cruzado. Además **uno de los dos puertos tiene que actuar como DCE dando reloj**. En principio desde el punto de vista de router cualquiera de los dos puede actuar de DCE, así que **lo importante es que conector del cable es el que marca que puerto es DCE**.

En el laboratorio, los cables **de tipo RJ-45** directos serán blancos o grises, mientras que los cables cruzados serán de color rojo.

Una vez que sabemos que puerto es el que actúa de DCE, tiene que dar reloj. Está opción la tenemos que activar vía IOS con el comando ''clockrate Bw'', donde Bw son los bps a los que queremos que trabaje la línea. En el puerto DTE *no debemos* activar este comando. En el siguiente ejemplo el Router-A es el actúa como DCE fijando una velocidad de transmisión de 56 Kbps y el Router-B es el que actúa como DTE.

```
Router-DCE# configure terminal
Router-DCE(config)# interface <nombre 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-DCE#

Router-DTE# configure terminal
Router-DTE(config)# interface <nombre 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
Router-DTE#
```

Las interfaces serie en los routers CISCO usan por defecto encapsulamiento HDLC (High Data-link Level Control) es un protocolo estándar, pero cuidado, CISCO usa una versión propietaria para sus enlaces WAN (sólo compatible con dispositivos CISCO).

HDLC (Standard): flag + address + control + data + FCS + flag

HDLC (Propietary): flag + address + control + Propietary + data + FCS + flag

```
R(config-if)#   encapsulation hdlc          → CISCO HDLC
```

8. Resolución de nombres

En el router se pueden asignar direcciones IP a nombres (igual que con el fichero /etc/hosts en una máquina UNIX), Figura 7, y también para que consulte a un servidor DNS un nombre desconocido (igual que con el fichero /etc/resolv.conf en una máquina UNIX), Figura 8.

```
R(config)# no ip domain-lookup        → desactiva el que se busque servidor de DNS
R(config)# ip host NAME @IP1 @IP2     → asigna nombres a direcciones IP
R(config)# show hosts                 → lista una cache de nombres y @IP
                                      (configurar una intrerficie con ip host name @IP)
```

Figura 7: DNS estático.

```
R(config)# ip domain-lookup
R(config)# ip name-server @IPserver1 ….. @IPserver6 → máximo 6 servidores DNS
```

Figura 8: DNS dinámico.

Observación: Por defecto el router tiene activado la resolución DNS dinámica. Si en la línea de comandos se teclea un nombre que no se reconoce cómo un comando, el router intenta contactar con el servidor DNS para resolver el nombre, y la consola se queda congelada varios segundos. Si no hay servidor de nombres configurado y se desea eliminar esta espera, se puede desactivar el DNS dinámico ejecutando “no ip domain-lookup”.

9. Encaminamiento estático

A continuación vemos un ejemplo de configuración del encaminamiento estático usando el comando ‘‘ip route’’.

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

La primera dirección es la dirección de red destino. A continuación escribimos la máscara asociada a esa red. La tercera dirección corresponde a la del interfaz del router por donde se establece la ruta (gateway por defecto).

10. Realización de la práctica

Para realizar la práctica cada grupo debe coger una pareja de routers conectados por el Puerto serie.

10.1. Primera parte

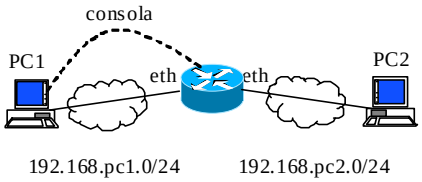


Figura 9: Primera parte.

1. Configurar las interfaces de los routers de la red de la Figura 9. NOTA: Las direcciones IP más bajas de la subred suelen usarse para las interfaces de los routers (porque son más fáciles de recordar), y las más altas para interfaces de los hosts. Apuntar las direcciones IP configuradas en la siguiente tabla:

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

2. Configurar las interfaces de los hosts conectados con Ethernet y configurar como router por defecto la interfaz del router correspondiente. Comprobar que el host tiene conectividad con el router mediante el comando ping y ver el formato de la tabla de encaminamiento del host. Comprobar que es posible conectarse al router con telnet.
3. Comprobar que los hosts tienen conectividad entre ellos.
4. Ver e interpretar el formato de la tabla de encaminamiento de los hosts (comando ‘‘route -n’’) y del router (comando ‘‘show ip route’’).
5. Configurar telnet asignando el password cisco. Comprobar que es posible conectarse al router usando telnet desde los dos PCs.

10.2. Segunda parte

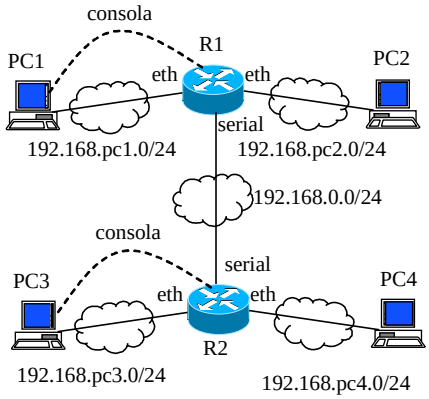


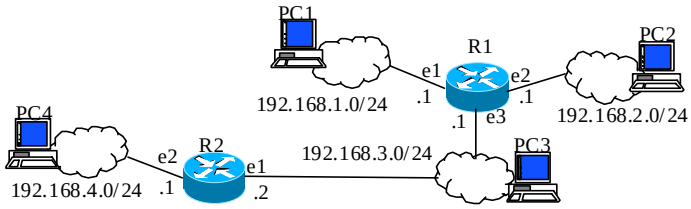
Figura 10: Segunda parte.

6. Configura la red de la Figura 10 usando encaminamiento estàtico en lor routers (comando “ip route”). Apuntar las direcciones IP configuradas en la siguiente tabla:

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

7. Comprobar con el comando “show interfaces” el tipo de encapsulamiento de las interfaces serie. Configura el enlace serie usando encapsulamiento HDLC. Asigna direcciones IP a los extremos del enlace serie y comprueba que hay conectividad entre los dos routers.
8. Observar que desde un host sólo se tiene conectividad con cualquier interfaz del router a la cual está directamente conectado, y con otros hosts situados en redes directamente conectadas al mismo router. Explicar porqué.
9. Usar el comando ‘‘`ip route`’’ para añadir entradas estáticas en cada router para tener conectividad con cualquier subred de la red establecida.
10. Ver el formato de la tabla de encaminamiento del router con el comando ‘‘`show ip route`’’ y comprobar que tienes conectividad con los todas las subredes de la red.
11. Usa el comando `traceroute` para comprobar que el PC1 se comunica con PC4 a través de los routers.
12. Elimina las entradas no directamente conectadas de la tabla de encaminamiento de R1 (comando “no ip route”) y añade una ruta por defecto hacia R2. Comprueba que continua habiendo conectividad entre todos los PCs.

11. Informe previ



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

1. Digues quines comandes s’haurien d’executar en R2 per assignar l’adreça IP les interfície de xarxa
2. Digues quines comandes s’haurien d’executar en R1 i R2 perquè totes les xarxes siguin accessibles per tots els PCs.
3. Suposa que hi ha algun error en la configuració. Digues quina comanda del router et permet veure la configuració actual.

Lab 3. Encaminamiento dinámico: RIPv1 y RIPv2

1. Introducción a RIP (RFC-2453)

Las características básicas son:

- La métrica es el número de saltos hasta el destino: 1 si el destino es una red directamente conectada, 2 si hay que pasar por un router, etc.
- Los router envían periódicamente (cada 30 segundos) un mensaje RIP broadcast por cada interfaz con los destinos y métricas conocidos. Se envía con UDP, puerto fuente y destino: 520.
- Si se dejan de recibir mensajes RIP de un vecino (180 segundos), se asume que ese router ha caído.
- La métrica infinito vale 16.
- RIP versión 2 introduce los cambios: Se añade la máscara a los destinos enviados en los mensajes. Los mensajes se envían a la dirección multicast: 224.0.0.9 (*all RIPv2 routers*)

1.1. Count to infinity

El principal problema de RIP es el tiempo de convergencia: Es decir, el tiempo que pasa desde que hay un cambio en la topología de la red hasta que las tablas de encaminamiento se estabilizan. Este tiempo puede ser especialmente grande cuando se produce el llamando problema de *count do infinity*. Esto ocurre cuando hay un cambio en la topología y la secuencia de mensajes RIP enviados hacen que un router *A* crea que puede llegar un destino *D* que pasado a ser inaccesible, a través de otro router *B* que a su vez depende de *A* para llegar a *D*.

Para solucionar el problema del *count to infinity* suele usarse la modificación *Split horizon*. Esta modificación consiste en que al enviar un mensaje RIP en una interfaz, se eliminan las entradas de la tabla de encaminamiento que tengan un gateway en la misma interfaz.

Otra modificación usada en los routers CISCO consiste en el llamado *holddown timer*: Cuando se recibe un mensaje RIP de un vecino indicando que una red que ha quedado inaccesible es accesible a través de ese router, entonces marca la ruta y inicia un temporizador holddown. Si cuando expira el temporizador todavía se advierte la ruta como accesible a través de ese router, entonces se actualiza la ruta a través de ese router.
Otra modificación para acelerar la convergencia consiste en no esperar los 30 segundos a enviar un mensaje RIP cuando se produce un cambio en la tabla de encaminamiento. Esta técnica se conoce como *triggered updates*.

2. Configuración de RIP

Para activar el algoritmo de encaminamiento RIP, los pasos a seguir son los siguientes:

```
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
Router#
```

Figura 11: Configuración de RIP

El comando "`network`" indica las interfaces que van a enviar o procesar mensajes de RIP. Se debe indicar las direcciones de red sin máscara (este comando asume la correspondiente a la clase). Es decir la red mayor a la que pertenece la dirección IP de la interfaz. Por ejemplo si la interfaz usa la dirección IP 10.5.4.2/24 basta con anunciar la clase A 10/8 de la forma ''`network 10.0.0.0`''. Notar que el comando network no usa máscara, sólo la dirección de red.

La versión de RIPv1 no soporta subnetting. Si queremos una red subneteada debemos usar RIPv2. El uso de la versión 2 se indica después del comando "`router rip`", ejecutando "`version 2`".

Podemos capturar los paquetes que se envían y reciben con el comando "`debug IP RIP`" desde modo PRIVILEGED EXEC. Esta opción consume muchos recursos del sistema, por lo que en operación normal debería estar desactivado.

Con el comando "`show ip route`" podemos observar la tabla de encaminamiento del router. En la información listada por el router, aparece indicado si la ruta se ha fijado de forma estática o ha sido aprendida con RIP.

El comando ''`show ip protocol`'' permite ver la configuración de RIP. El comando muestra la versión de RIP tiene activada cada interfaz tanto de entrada ("receive") como de salida ("sent"). Notar que podemos enviar RIPv1 y recibir tanto de RIPv1 como de RIPv2. El tiempo de *hold down* es el tiempo que espera el router en aceptar una nueva ruta para una entrada que ha sido invalidada, para 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
     Interface         Send  Recv  Triggered RIP  Key-chain
     Ethernet2         1     1, 2
     Ethernet3         1     1, 2
```

Figura 12: Volcado del comando show ip protocol.

Se puede activar RIPv2 globalmente en todas las interfaces con el comando ''`versión [1 2]`'':

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

Figura 13: Activación de RIPv2.

Si uno de los routers se mantiene con RIPv1 y enviase mensajes RIPv1 la interfaz los rechazaría. Es mejor cambiar la versión por interfaz con los comandos: ''`ip rip receive versión [1 2]`'' y ''`ip rip send versión [1 2]`''. De forma que activamos enviar solo con versión 2 y recibir tanto versión 1 como 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: Configuración para enviar RIPv2, pero recibir RIPv1 y RIPv2.

NOTAS:

Por defecto el router hace "sumarización de rutas". La sumarización se hace a la clase, y sólo cuando se envían los mensajes hacia una red con dirección base distinta.. Por ejemplo, si en la tabla hay las subredes 10.0.1.0/24 y 10.0.2.0/24, al enviar el mensaje RIP hacia la red 192.168.0.0/24 advertirá la red 10.0.0.0/8. Para desactivar la sumarización hay que ejecutar el comando:

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

Figura 15: Configuración de RIP para que no haga sumarización de rutas.

Para que el router advierta las entradas estáticas (esto incluye la entrada por defecto), hay que ejecutar el comando:

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

Figura 16: Configuración de RIP para que añada las entradas estáticas en los mensajes de update.

El router usa dos métricas: la métrica administrativa y la métrica del algoritmo de encaminamiento. Si existen varias rutas hacia un mismo destino, se elige la ruta con métrica administrativa menor. Por ejemplo, RIP tiene métrica administrativa 120 y OSPF 110. Si ambos añaden una entrada en la tabla hacia un mismo destino, primero se elegirá la ruta añadida por OSPF (el router la considera más fiable). Al mostrar la tabla de encaminamiento podemos ver las métricas entre corchetes:

```
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
```

En la entrada, la R indica que ha sido añadida por RIP. 120 es la métrica administrativa, y 1 es la métrica usada por el protocolo. La métrica de RIP mostrada por CISCO es el número de routers hasta el destino. Al advertir la métrica, el RFC dice que deben advertirse los saltos hasta el destino (es decir, si hay un router hasta el destino, se darán dos saltos). Por este motivo, el router CISCO incrementa en 1 las métricas de RIP que muestra en la tabla cuando envía los mensajes RIP.

3. Subredes con clase y sin clase

Cuando se hace subnetting, la primera y última subred quedan inutilizadas. Eso ocurre porque la dirección de subred de la primera subred coincide con la dirección de subred de la red mayor (o subneteada) y la dirección broadcast de la última subred coincide con la dirección broadcast de la red mayor (o subneteada). Para que los routers puedan trabajar con la primera subred y con la última el IOS activa por defecto el comando `''ip subnet zero''` (`''no ip subnet zero''` para desactivar la opción).

Una red puede trabajar con clases (A, B o C) o puede usar el concepto de sin clase (CIDR). Para poder crear subredes independientemente de la clase, el IOS activa por defecto el comando "`ip classless`". De hecho el comando funciona de la siguiente manera: si está activo, el router envía los paquetes a la interfaz superneteada que mejor se ajuste en la tabla de encaminamiento (o a la ruta por defecto). En el caso de que está desactivada ("`no ip classless`") el router solo re-envía el paquete si la ruta está en la tabla de encaminamiento (o hay una ruta por defecto). Si no está en la tabla de encaminamiento, entonces el router descarta el paquete. Por ejemplo, si la red 10.0.0.0/8 está subneteada y en la tabla hay las redes 10.0.1.0/24, 10.0.2.0/24 y una entrada por defecto. Al recibir un datagrama dirigido a la red 10.0.3.0/24 con "`no ip classless`", el router descarta el datagrama. Con "`ip classless`", en cambio, el router enviaría el datagrama por la ruta por defecto.

4. Realización de la práctica

4.1. Red IP con subnetting. RIPv2 y sumarización

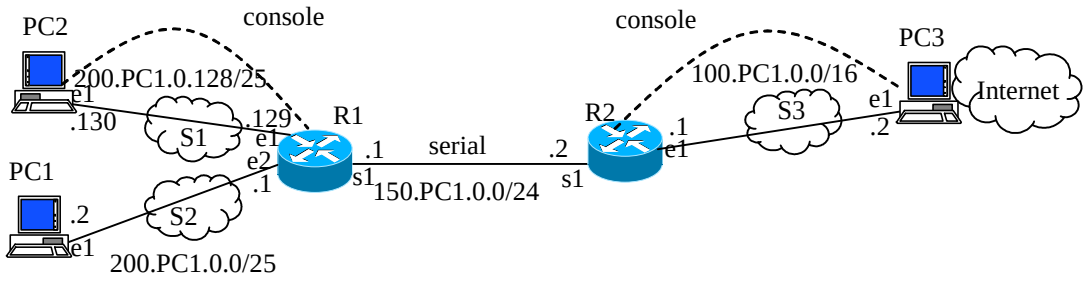


Figura 17

1. Configura la red de la Figura 17. El PC3 representa el router del ISP que da acceso a Internet. Por lo tanto, el router R2 tiene que tener PC3 como ruta por defecto. Además, PC3 tiene que tener R2 como router para llegar a las redes 200.PC1.0.0/24 i 150.PC1.0.0/24. Apuntar las direcciones IP configuradas en la siguiente tabla:

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

2. Configurar las interfaces de cada router para RIPv2. Configura RIP para que advierta la ruta por defecto.
3. Observar la activación del protocolo RIP usando el comando ‘‘`show ip protocol`’’ e interpretar la salida de este comando.
4. Observa la tabla de routing con el comando ‘‘`show ip route`’’ y mira si hay conectividad entre todos los PCs.
5. Debuguea RIPv2 con el comando ‘‘`debug ip rip`’’ (‘‘`no debug ip rip`’’ para desactivarla). Interpreta los mensajes.
6. Ejecutar el comando “no auto-sum” en la configuración de RIP de los routers. ¿Cómo cambian los mensajes RIP y las tablas de encaminamiento?
7. Observar la convergencia del protocolo RIP si desconectamos PC1, usando del comando “`debug ip rip`’’. Interpreta los mensajes. Observa como al desconectar transcurre un tiempo hasta que las tablas convergen y como se envía inmediatamente un triggered update con métrica infinito (16 saltos). Volver a conectar y observar los cambios.
8. Si desactivamos *split-horizon* en una de las interfaces ¿Qué redes se anunciarán en un mensaje de encaminamiento RIP? Para desactivar split-horizon debes ejecutar el comando ‘‘`no ip split-horizon`’’ desde el submodo de interfaz. Por ejemplo para deshabilitar split-horizon en la interfaz 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. Red IP con subnetting. RIPv2 entre varios grupos (opcional)

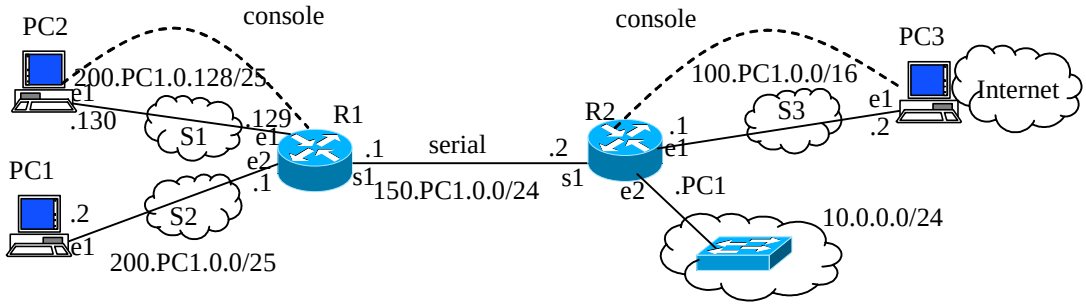
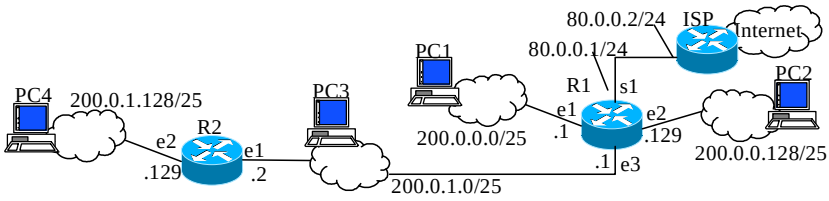


Figura 18

9. Interconectar las redes de varios grupos a través de un switch, tal como muestra la figura. Activar RIP en la red 10.0.0.0/24 y comprovar como convergen las tablas de encaminamiento.

5. Informe previ



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

1. Digues la comanda que s'ha d'executar en R1 per tenir una ruta per defecte cap a l'ISP.
2. Digues quines comandes s'haurien d'executar en R1 i R2 per configurar RIP. Es desitja que R1 adverteixi la ruta per defecte.
3. Digues quina serà la taula d'encaminament de R1 i R2 quan RIP hagi convergit.
4. Suposa que es fa servir split-horizon. Digues el contingut dels missatges RIP que R1 i R2 enviaran en la xarxa 200.0.1.0/25
5. Repeteix els dos apartats anteriors si executem "no auto-sum" en la configuració de RIP dels dos routers.

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

1. Introducción

Las listas de acceso (ACL) se usan para el filtrado de paquetes en función de ciertos parámetros como pueden ser las direcciones de red origen o destino, los puertos origen o destino, el tipo de protocolo (ip, icmp, tcp, udp, etc). Una de las aplicaciones donde se usan más las listas de acceso es en la seguridad de la red. Con las ACLs se puede bloquear el tráfico no deseado en una interfaz ya sea de salida o de entrada. Las ACLs no sólo se usan en temas de seguridad, sino que también para identificar paquetes en aplicaciones como NAT (Network Address Translation), en BGP para filtrar rutas al crear políticas de encaminamiento, etc.

Existen ACLs para distintas pilas de protocolos: TCP/IP, IPX/SPX, Apple, etc. Este documento se centra en las ACLs aplicadas a seguridad en la red para la pila de protocolos TCP/IP. Capa protocolo tiene asignado un rango de ACLs. Por ejemplo las ACLs entre la 1 y la 199 se usan en TCP/IP, mientras que las comprendidas entre la 800 y la 999 se usan para IPX/SPX, otros rangos se usan para DECnet (300-399), XNS (400-599), AppleTalk (600-699), etc.

Cuando creamos una lista de acceso y la aplicamos a una interfaz de entrada o de salida, estamos creando una secuencia de instrucciones que son chequeadas cada vez que un paquete entra o sale por esa interfaz. Es importante notar varias características de las ACLs.

Primero, que una ACL se aplica a la interfaz ya sea de entrada o de salida. Se pueden crear una ACL para la interfaz de salida y otra distinta para esa interfaz de entrada.

Lo segundo, las ACLs son secuencias de instrucciones que son chequeadas contra el paquete. El orden de las instrucciones es importante, ya que cuando una línea de la secuencia da cierta en el chequeo, se toma una acción y se sale de la ACL, es decir no se continua chequeando para comprobar que haya otra línea de la secuencia que también resulta cierta. Por consiguiente es muy importante diseñar la ACL en la secuencia que nos interese más.

Por ejemplo no es lo mismo estas dos líneas de una ACL:

- Si el paquete es icmp recházalo
- Si el paquete es ip acéptalo

que la secuencia:

- Si el paquete es ip acéptalo
- Si el paquete es icmp recházalo

Suponed que llegara un paquete ICMP. En el primer caso el paquete se rechazaría ya que la primera línea se cumple, el paquete es ICMP. En el segundo caso el paquete ICMP se aceptaría ya que la primera línea también se cumple, con lo cual ya no se comprobaría la segunda.

Otro aspecto importante es que no podemos insertar líneas en la secuencia. Si nos equivocamos al crearla o queremos insertar una línea a hay que borrar las líneas hasta el punto de inserción.

Finalmente, también MUY IMPORTANTE, la última línea de una lista de acceso NUNCA aparece, es decir existe de forma explicita y siempre es DENIEGO TODO.

Dentro de las listas de acceso TCP/IP hay dos tipos de ACLs

- Listas de acceso IP estándar (1-99)
- Listas de acceso IP extendidas (100-199)

2. Wildcard mask

La wildcard mask es una mascara de 32 bits que indica que bits de la dirección IP se tienen que comprobar y cuales no. Si los bits de la máscara están a 0 entonces se comprueban, si están a 1 entonces no se comprueban.

Por ejemplo si queremos que un paquete que entra se compruebe si pertenece al host con dirección IP `145.34.5.6`, queremos que se comprueben todos los bits de la dirección IP. Eso significa que la wildcard mask sería `0.0.0.0`. En este caso se suele sustituir la tupla *`@IP WildcardMask`* por *`host @IP`*. Por ejemplo la tupla *`145.34.5.6 0.0.0.0`* se puede expresar como *`host 145.34.5.6`*.

Si quisiéramos que no se comprobase ningún bit, pondríamos una wildcard mask de `255.255.255.255`. en este caso se suele sustituir la tupla `@IP WildcardMask` por `any`. Por ejemplo la tupla `145.34.5.6 255.255.255.255` se puede expresar como `any`.

También podemos expresar redes. Por ejemplo para comprobar todos los paquetes que vengan de la red `145.34.5.0/24`. Eso significa que tenemos que comprobar todos los paquetes cuyos primeros 24 bits coincidan con los de la dirección de red. Luego la wildcard mask debería ser `0.0.0.255`.

3. ACLs estándar

Las ACLs estándar solo usan las direcciones origen para hacer la comprobación. Las listas de acceso estándar tienen números (`acl#`) comprendidos entre el 1 y el 99. El comando tiene el siguiente formato:

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

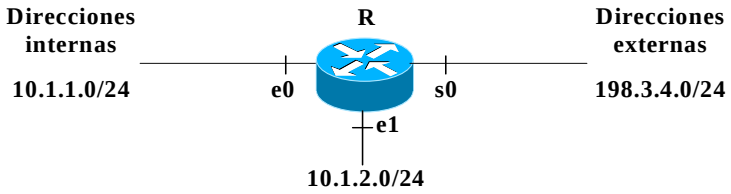
El primer comando, access-list, crea la lista de acceso con número `acl#` y con condición deniego o permito sobre la dirección IP origen especificada con la correspondiente wildcard mask. Recordad que la última línea de una ACL nunca aparece pero siempre es "`access-list acl# deny any`".

El segundo comando, access-group, asigna la lista de acceso acl# sobre el protocolo IP sobre la interfaz de entrada o de salida donde se ejecuta dicho comando.
Para borrar una ACL ejecutar el comando:

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

Ejemplo:

Queremos denegar en la interfaz s0 de salida cualquier paquete IP que provenga de la red `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
```

Primero creamos la lista de acceso con número igual a 1 y denegamos todo el tráfico que venga de la red `10.1.1.0/24`. Como la última línea sería denegar todo lo demás (ej.; la red `10.1.2.0/24`), permitimos el resto de direcciones. Aplicamos esta ACL sobre la interfaz de salida s0 porqué si lo hiciésemos sobre la e0 de entrada entonces bloquearíamos los paquetes de la red `10.1.1.0/24` hacia la red `10.1.2.0/24`.

4. ACLs extendidas

Las ACLs extendidas permiten usar tanto las direcciones origen como destino para hacer la comprobación. Además permiten especificar el protocolo sobre el que se quiere hacer la comprobación y en el caso de que sea TCP o UDP especificar el puerto. Las listas de acceso extendidas tienen números (`acl#`) comprendidos entre el 100 y el 199. El comando tiene el siguiente formato:

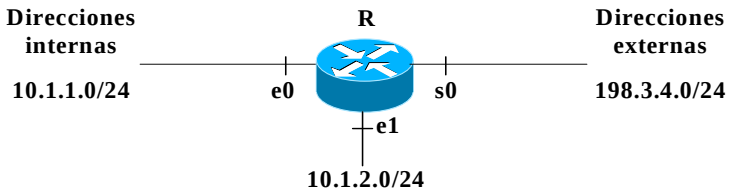
```
access-list acl# {deny|permit} protocol {@IPsource WildcardMask | host @IPsource | any} [operador
portsource] {@IPdest WildcardMask | host @IPdest | any} [operador portdest] [established]
ip access-group acl# {in |out}
```

El primer comando, access-list, crea la lista de acceso extendida con número acl# y con condición deniego o permito sobre la dirección IP origen y/o destino especificadas con las correspondientes wildcard masks. `protocol` puede ser ip, icmp. tcp, udp, etc. *`Operador`* puede ser *{lt,gt,eq,neq}* (less than, greater than, equal, non equal) y `port` es un puerto TCP o UDP. `established` sólo es válido con tcp. Cuando se usa captura el tráfico tcp perteneciente a una conexión establecida. Para ello el router mira los paquetes con el bit ACK o RST activados (el primer paquete de SYN siempre tiene estos dos bits desactivados).

Recordad que la última línea de una ACL nunca aparece pero siempre es "`access-list acl# deny ip any any`".

Ejemplo:

Queremos denegar en la interfaz s0 de salida cualquier paquete ICMP que provenga de la red `10.1.1.0/24` y el acceso a cualquier puerto telnet (puerto 23) por parte de un host de esa red.




```
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
```

Primero creamos la lista de acceso extendida 101, denegando el acceso de paquetes ICMP, segundo otra línea denegando el acceso a cualquier host con puerto 23, finalmente permitimos cualquier otro tipo de tráfico. A continuación aplicamos la lista de acceso a la interfaz de salida s0.

5. Verificación

`R# show ip interface`	`Muestra si hay alguna ACL en la interface.`
`R# show access-lists`	`Muestra las ACLs definidas`
`R# show running-config`	`Para comprobar la configuración.`

6. NAT

NAT (Network Address Translation) es el proceso que permite la traslación de direcciones privadas a públicas mediante la substitución o alteración de las direcciones IP o puertos en las cabeceras IP y TCP del paquete transmitido. Para que NAT funcione debemos disponer de un router que implemente NAT en alguna o varias de sus variantes: NAT estático, NAT dinámico y NAT por puertos (PAT).
No siempre se usa NAT para trasladar direcciones privadas a públicas. Hay ocasiones en que se trasladan direcciones privadas a privadas o direcciones públicas a direcciones públicas. Las direcciones internas pueden ser tanto privadas como públicas. El caso más típico es aquel en que la dirección interna es una dirección privada y la dirección externa es una dirección pública. IOS usa la siguiente nomenclatura genérica a la hora de usar NAT:

- **Direcciones locales internas (Inside local addresses):** la dirección IP interna asignada a un host en al red interna
- **Direcciones globales internas (Inside global addresses):** la dirección IP de un host en la red interna tal como aparece a una red externa
- **Direcciones locales externas (Outside local addresses):** la dirección IP de un host externo tal como aparece a la red interna
- **Direcciones globales externas (Outside global addresses):** la dirección IP asignada a un host externo en una red externa

Ver que la diferencia entre una dirección local y global interna es que la dirección local interna es la dirección que queremos trasladar mientras que la dirección global interna es la dirección ya trasladada.

7. NAT estático

Usamos NAT estático cuando las direcciones están almacenadas en una tabla de consulta del router y se establece un mapeo directo entre las direcciones internas locales y las direcciones internas globales. Eso significa que por cada dirección interna local existe una dirección interna global. Este mecanismo se suele usar cuando se quiere cambiar un esquema de direcciones de una red a otro esquema de direcciones o cuando se tienen servidores que tienen que mantener una dirección IP fija de cara al exterior como DNS o servidores Web.

7.1. Configuración de NAT estático

Para configurar NAT estático seguiremos los siguientes pasos:

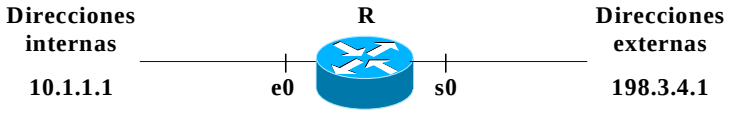
- Definir el mapeo de las direcciones estáticas:

  ip nat inside source static *local-ip global-ip*

  ip nat inside source static network *local-network global-network mask*

- Especificar la interfaz interna
  ip nat inside
- Especificar la interfaz externa
  ip nat outside

Ejemplo:




```
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
R#
```

8. NAT dinámico

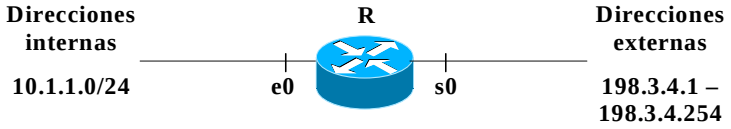
Usamos NAT dinámico cuando disponemos de un conjunto de direcciones globales internas que se asignarán de forma dinámica y temporal a las direcciones locales internas. Esta asignación se efectuará cuando se recibe tráfico en el router y tiene un Temporizador asignado.

8.1. Configuración de NAT dinámico

Para configurar NAT dinámico seguiremos los siguientes pasos:

- Crear un conjunto de direcciones globales:
  ip nat pool *name start-ip end-ip* {**netmask** *mask* | **prefix-length** *prefix-length*}
- Crear una ACL que identifique a los hosts para la traslación
  access-list *access-list-number* **permit** *source {source-wildcard}*
- Configurar NAT dinámico basado en la dirección origen
  ip nat inside source list *access-list-number* **pool** *name*
- Especificar la interfaz interna
  ip nat inside
- Especificar la interfaz externa
  ip nat outside

Ejemplo:




```
R# configure terminal
R(config)# ip nat pool fib-xc 198.3.4.1 198.3.4.254 netmask 255.255.255.0
(config)# access-list 2 permit 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
```

Las entradas se asignan por defecto 24 horas. Si se quiere modificar el valor del temporizador usar el siguiente comando:

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

Donde “seconds” es el tiempo que se asignará al temporizador.

9. NAT overload o PAT (Port Address Translation)

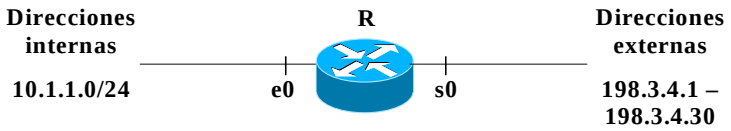
Usamos PAT (NAT por puertos) cuando disponemos de una dirección global interna puede direccional todo un conjunto grande (centenares) de direcciones locales internas. Esta asignación la efectúa cuando el par dirección global/puerto. Aunque disponemos de 65535 puertos (16 bits) en realidad el router PAT solo puede usar un subconjunto de estos puertos (depende del router, pero aproximadamente unas 4000 puertos por dirección global). PAT se puede usar en conjunción con NAT dinámico de forma que varias direcciones globales con múltiples puertos direccionan un mayor número de direcciones locales internas.

9.1. Configuración de PAT

Para configurar PAT seguiremos los siguientes pasos:

- Crear un conjunto de direcciones globales (puede ser una sola dirección):
  ip nat pool *name start-ip end-ip {***netmask** *mask* | **prefix-length** *prefix-length}*
- Crear una ACL que identifique a los hosts para la traslación
  access-list *access-list-number* **permit** *source {source-wildcard}*
- Configurar PAT basado en la dirección origen
  ip nat inside source list *access-list-number* **pool** *name* **overload**
- Especificar la interfaz interna
  ip nat inside
- Especificar la interfaz externa
  ip nat outside

Ejemplo: usaremos hasta 30 direcciones internas globales, cada una de las cuales hace PAT




```
R# configure terminal
R(config)# ip nat pool fib-xc 198.3.4.1 198.3.4.30 netmask 255.255.255.0
(config)# access-list 2 permit 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 el caso de que no haya un conjunto de direcciones globales podemos usar la dirección asignada a la interface "s0" de la siguiente manera:

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

10. Verificación de una configuración NAT

Usamos los siguientes comandos para verificar que la configuración NAT es correcta (desde modo privilegiado):

show ip nat translations

show ip nat translations verbose

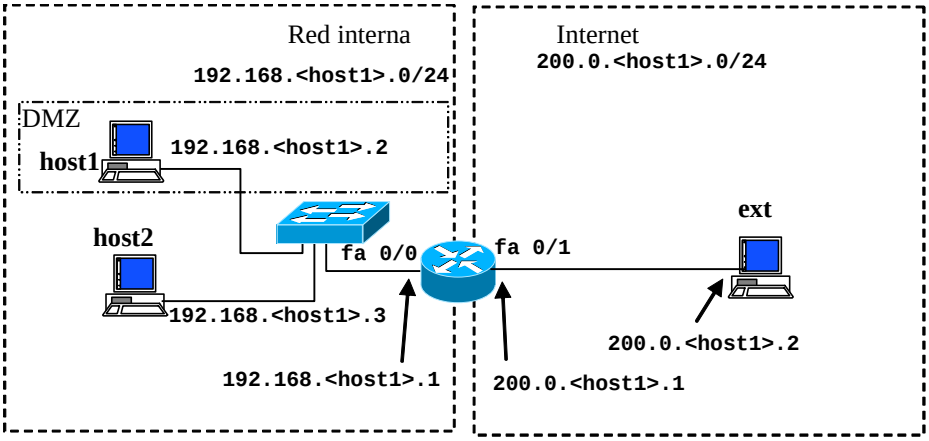
show ip nat statistics

debug ip nat (no debug ip nat)

clear ip nat translation * → elimina todas las traslaciones NAT

11. Realización de la práctica

11.1. NAT



1. Configurar la red de la figura. La red de la izquierda representa una red privada y la de la derecha representa Internet. Fijaros que la red privada tiene direcciones privadas (no enrutables en Internet): Recordar que los rangos de direcciones privadas son 10.0.<host1>.0/8, 172.16.<host1>.0/12 y 192.168.<host1>.0/16. Configurar host1 y host2 para que tengan una ruta por defecto usando el router. Configurar el host que representa Internet (ext) para que sólo sepa llegar a las direcciones públicas: Es decir, no añadir en la tabla de encaminamiento de ext ninguna ruta por defecto. De esta manera, ext sólo podrá llegar a su red directamente conectada, que representa Internet. Apuntar las direcciones IP configuradas en la siguiente tabla:

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

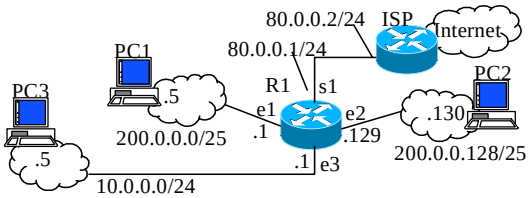
2. Comprueba haciendo *pings* que hay conectividad entre host1-host2-router y entre ext-router. Comprueba que no hay conectividad entre host1/2-ext (puesto que ext no puede contestar a los datagramas que llegan con una dirección fuente privada).
3. Configura PAT (sin cambiar la configuración anterior) para que todos los hosts de la red interna accedan a Internet con la dirección pública de la interface fa0/1 del router (200.0.<host1>.1):
   - Comprueba que ambos hosts de la red interna pueden acceder a Internet.
   - Comprueba el funcionamiento de NAT con `debug ip nat` (ejecuta `no debug ip nat` para desactivar el comando).
   - Comprueba la tabla NAT (`show ip nat translations`).
   - Comprueba que ext no puede acceder a los hosts de la red Interna (host1/2). Razona porqué no es posible.
4. Configura un static NAT de 200.0.<host1>.1 hacia el host1. Comprueba con "debug ip nat" en el router que ext tiene conectividad con host1 (con ping desde ext).

11.2. ACLs

Continuando con la configuración anterior:

5. Configura una lista de acceso estándar para que sólo pueda acceder a Internet host2. Tener en cuenta que el orden en que se aplica NAT i ACLs en una interfaz es: primero ACL in, después NAT i finalmente ACL out. ¿Qué ocurre con ext? ¿Tiene acceso a host1? ¿Por qué?
6. Cambia la configuración de forma que permita acceder desde Internet solamente al servicio ssh (puerto 22) de host1. Para comprobarlo conéctate desde ext al servidor ssh de host1, y después intenta conectarte con telnet a host1. Deseamos que host2 continúe con acceso a Internet, pero desde host1 no tiene que ser posible iniciar una conexión con Internet. Compruébalo confirmando que es posible conectarse con telnet desde host2 a ext, pero no desde host1.

12. Informe previ



1. Digues les comandes que s'haurien de configurar en el R1 perquè els PCs de la xarxa 10.0.0.0/24 puguin accedir a Internet amb PAT amb l'adreça pública que R1 té assignada a la interfície s1.
2. Suposa que en R1 s'executen les comandes:

```
Router(config)# access-list 1 permit 200.0.0.128 0.0.0.7
Router(config)# interface e1
Router(config-if)# ip access-group 1 out
```

   Digues quins dels següents hosts podrien accedir a PC1, explica breument perquè:

   a) PC2

   b) PC3

   c) Un host de la xarxa 200.0.0.128/25 amb adreça 200.0.0.250.

   d) Un host d'internet amb adreça 150.0.0.10

3. Digues quines comanes s'haurien d'afegir a les anteriors per aconseguir el següent:
   a) PC1 pot accedir sense restriccions a Internet, però no a les xarxes 10.0.0.0/24 i 200.0.0.128/24.

   b) Des de les xarxes 10.0.0.0/24 i 200.0.0.128/24 es pot accedir sense restriccions a PC1.

   c) Des d'Internet només es pot accedir al port 80 de PC1.

   d) PC1 respon al ping de qualsevol host.

Lab 5. Switches

1. Introducción

Un switch Ethernet es un dispositivo de nivel 2 que segmenta los dominios de colisiones. La configuración de un switch es totalmente dependiente del fabricante. En este laboratorio vamos a usar switches Ethernet de la gama 2950 de CISCO. Para entrar y configurar el switch seguiremos los mismos pasos que en un router CISCO. Nos conectamos por el puerto consola del switch con un cable rollover y con una aplicación que permita la comunicación asíncrona por el puerto serie del host (ej. hyperterminal o minicom). Una vez conectados entramos en modo setup, o en modo user exec. De modo user exec hemos de entrar a modo privilegiado con el comando "`enable`". En este modo podremos visualizar tablas, ficheros de configuración (running-config), bases de datos del switch, etc. Para configurar cualquier funcionalidad hay que entrar en el modo de configuración global usando el comando "`configure terminal`".

2. Tabla MAC

Cada puerto de un switch es un dominio de colisiones. Para segmentar la red Ethernet, un switch usa la tabla MAC. El switch inicialmente tiene la tabla vacía. Cada vez que una estación envía una trama Ethernet a otro host, el switch "aprende" a que puerto está conectado una dirección MAC. Por ejemplo si una trama Ethernet entra por el puerto del switch e0 con dirección origen MAC=A tiene destino la MAC=B, el switch aprende que la MAC=A está conectada al puerto e0.

A medida que los hosts envían peticiones a otros hosts y estos responden, la tabla MAC se va llenando. Como los hosts pueden cambiar de situación (pasar a estar conectados a otro puerto), no conviene que las entradas de la tabla MAC sean estáticas. Por eso las entradas tienen un tiempo de vida ("age"). Pasado el tiempo de vida, la entrada de la tabla MAC desaparece ("aging out"). Por eso decimos que las entradas son *dinámicas*.

- Verificación:

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

Por defecto un switch CISCO de gama 2950 tiene asignado un tiempo de vida de entradas en la tabla MAC de 300 segundos (5 minutos), mecanismo de aprendizaje dinámico y ninguna estrada estática en la tabla.

Para ver la tabla MAC de un switch podemos usar el comando "`sh mac address-table`". Para ver el tiempo de vida se puede usar el comando "`sh mac address-table aging-time`". Para eliminar entradas aprendidas dinámicamente se puede usar el comando "`clear mac address-table dynamic`" (todas las entradas) o "`clear mac address-table dynamic address @MAC`" (eliminar la dirección @MAC de la tabla) o `clear mac address-table dynamic interface IFACE`" (para las MACs de una interfaz) o `clear mac address-table dynamic vlan VLAN-ID`" (todas las MACs de una VLAN).

3. VLANs

Definimos una VLAN como una red broadcast. Cada uno de los puertos de un router es una red broadcast por definición y por tanto una red IP. Para ahorrar puertos de router se pueden crear redes broadcast (redes IP) en un switch mediante software. Eso significa que con un puerto de router conectado al switch vamos a crear tantas VLANs (redes broadcast) como el software del switch nos permita. Un switch CISCO de la gama 2950 permite crear hasta 1024 VLANs.

Es evidente que si un puerto de router debe soportar N VLANs (N redes IP) el puerto deberá tener N direcciones IP, una por cada VLAN creada. También es evidente que para viajar desde una VLAN a otra hay que pasar obligatoriamente por el router. Es decir, no se puede ir desde una VLAN a otra directamente a través del switch, del mismo modo que el tráfico broadcast de nivel 2 (por ejemplo las tramas ARP) no se propagan entre VLANs distintas. Para conseguir esta segmentación de nivel 3 se utiliza un protocolo específico llamado de "trunking". Un enlace en modo trunk pertenece a más de una VLAN, de modo que permite enviar en un solo enlace todo el tráfico de las VLANs del switch al router (esta configuración se conoce con el nombre de *router-on-a-stick*). Las tramas que se envían en el trunk llevan una etiqueta (*tag*) con el número de VLAN a la que pertenece la trama. Existen dos protocolos de trunking: el que se usó por primera vez, propietario de CISCO, conocido como ISL, y el estandarizado por el IEEE: IEEE802.1Q. En los equipos de CISCO podemos encontrar ambos protocolos (los equipos más modernos suelen llevar sólo IEEE802.1Q).

Cuando encendemos un switch CISCO, todos los puertos pertenecen a la VLAN nativa. La VLAN nativa por definición es la VLAN-ID=1. Si se define un VLAN para un uso específico es mejor usar otras VLAN-ID distintos al 1. Para definir VLANs en un switch seguiremos los siguientes pasos:

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

donde VLAN-ID tiene rango 0001 – 1005, CREAMOS la VLAN con NOMBRE y NUMERO. CUIDADO: VLAN 1, 1002, 1003, 1004 y 1005 son VLANs por defecto para diversos tecnologías de nivel 2 (Ethernet, FFDI, TR, ...)

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

lista parámetros de todas o una VLAN determinada. Para borrar una VLAN:

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

Una vez que la VLAN está creada hay que asignar interfaces a la VLAN. Usar el comando switchport para asignar de forma estática puertos a una VLAN:

```
Sw(config)# interface fastethernet0/1
Sw(config-if)# switchport mode access → define VLANs en modo estático
Sw(config-if)# switchport access vlan VLAN-ID → asignar el puerto 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 interfaz tal como está en la memoria física
Sw# show interfaces IFACE switchport → lista el modo administrativo (ej.; acceso estático), el modo de acceso de la VLAN (ej..; vlan-id), etc
Sw# show vlan → lista información de las vlans creadas
```

Una vez creada la VLAN en el switch hay que definir el enlace entre el switch y el router como un enlace ("link") de tipo "trunk". Un "link trunk" es un enlace que pertenece a todas las VLANs creadas. Tiene que estar asignada a la VLAN nativa (VLAN=1). Solo interfaces Fast Ethernet pueden ser trunk.

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

Ahora el switch ya está configurado. Nos falta configurar el router para que entienda las diferentes VLANs creadas. El enlace del router debe ser un "link trunk" y además debe tener tantas direcciones IP como VLANs creadas. Para ello crearemos subinterfaces en la interfaz Fast Ethernet del router. Cada subinterfaz la asignaremos a una VLAN y le daremos una IP. En el siguiente ejemplo creamos 2 VLANs (VLAN-ID=2 y VLAN-ID=3) en el router. Usamos la interfaz Fast Ethernet 0/0 como interfaz de partida donde crearemos las subinterfaces Fast Ethernet 0/0.1 y Fast Ethernet 0/0.2 y asignamos el VLAN-ID a esa subinterfaz (con el comando *encapsulation*). Finalmente le damos una IP a la subinterfaz:

```
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-ID3
R(config-subif)#  ip address @IP3 MASK3
R(config-subif)#  exit
R(config-if)#  exit
R(config)#  exit
R#  sh ip route
```

Observar que en la tabla de encaminamiento tiene que aparecer una entrada con cada subinterfaz y su subred IP.

4. Puertos seguros

Puede haber situaciones en los que nos interese fijar direcciones MAC en la entrada de la tabla MAC. Por ejemplo, por motivos de seguridad sólo queremos que en un puerto del switch Ethernet se pueda conectar físicamente el host A. Si se conecta otro host con distinta dirección MAC a A queremos que el puerto se deshabilite. Con ello aumentamos la seguridad de nuestra red. A esta solución se le llama puertos seguros. Por defecto la seguridad por puertos está desactivada, para activarla en una interface:

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

Para añadir puertos seguros:

- El puerto debe estar en modo *access*. Para cambiar el modo de un puerto:

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

Descripción:

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 de conseguir un puerto seguro es especificar el número máximo de direcciones MACs que se pueden asociar a un puerto Ethernet y fijar las direcciones MAC que nos interesan como seguras en ese puerto. Pero primero tenemos que vaciar la tabla MAC borrando las direcciones dinámicas que haya podido añadir el switch con el comando:

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

- Para limitar el máximo número de MAC permitidas en una interface:

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

Si queremos asignar una MAC segura en una interfaz de una VLAN determinada hay que ejecutar:

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

- A continuación se define la acción a tomar cuando se produce una violación de puertos.

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

donde "protect" significa que se descartan tramas de las MAC que violan el sistema, "restrict" significa que además se envía un trap (aviso) al gestor de red (protocolo SNMP) y "shutdown" (por defecto) significa que se desactiva el puerto.

- Verificación:

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

NOTA:

Al violar la seguridad del puerto, este queda bloqueado. Para reactivarlo, ejecutar:

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

5. Realización de la práctica

La configuración del lab será de un router conectado a un switch por un enlace Fast Ethernet (tiene que ser Fast Ethernet para soportar *trunking*). A cada switch conectaremos 3 PCs.

5.1. VLANs y trunking

1. Borrar las VLANs creadas por el usuario existentes en el switch. Que pasa si se intenta borrar la VLAN=1?
2. Configura la topología de la Figura 19. Crea las estaciones T_1 y T_2 como pertenecientes a la VLAN=2 y la estación T_3 a la VLAN=3. Configura el router para que acepte VLANs. Apuntar las direcciones IP configuradas en la siguiente tabla:

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

3. Comprueba que puedes hacer un ping a todas las estaciones.
4. Comprueba la tabla MAC del switch. Identifica la dirección MAC y VLAN de todos los PCs en la tabla MAC.
5. Observa la tabla de routing del router. ¿Qué entradas y qué formato tienen?
6. Ejecuta tcpdump en las estaciones para ver el tráfico recibido/transmitido.
7. Haz un ping desde T_1 a T_2. ¿Que dispositivos ven tráfico? ¿Por qué?
8. Haz un ping desde T_1 a T_3. ¿Que dispositivos ven tráfico? ¿Por qué?
9. Usa el comando traceroute entre T1 y T2, y entre T1 y T3. Razona las diferencias.

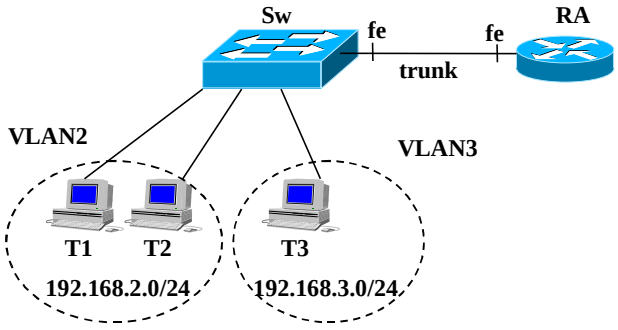
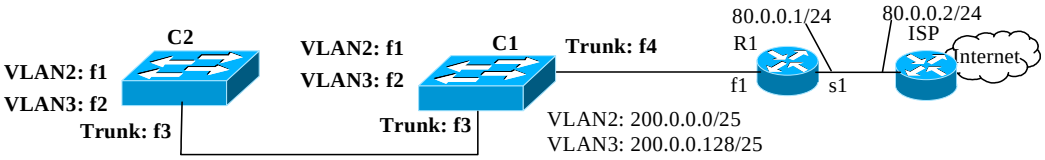


Figura 19

5.2. Puertos seguros

10. Configura un puerto seguro en una de las estaciones (ej. T_1). Configura que la acción por defecto sea deshabilitar el puerto si otra estación se conecta. Desconecta la estación T_1 y conecta la estación T_2. Observa como se deshabilita el puerto y vuelve a conectar la estación original. El comportamiento debe ser el siguiente: si se la acción es *shutdown* del puerto, no aceptará de nuevo la estación original y habrá que habilitarlo manualmente (es decir entrar en la interfaz del switch y ejecutar el comando *shutdown* i *no shutdown*).

6. Informe previ



1. Dóna les comandes per a configurar els commutadors i el router R1 de la figura. Suposa que el hostid del router R1 en cada xarxa és l'adreça numèricament més baixa de la xarxa.
2. Suposa que en el commutador C2 hi ha un PC1 connectat a un port de la VLAN2 i PC2 connectat a un port de la VLAN3. Digues 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.

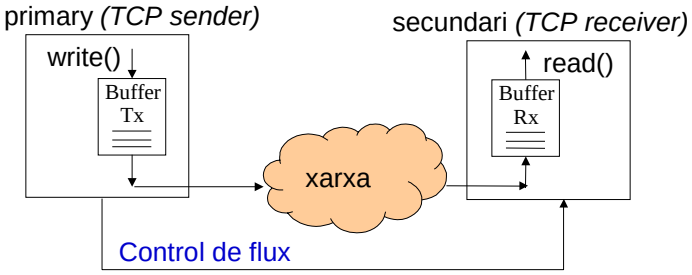


Figura 20: Nivell TCP.

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.

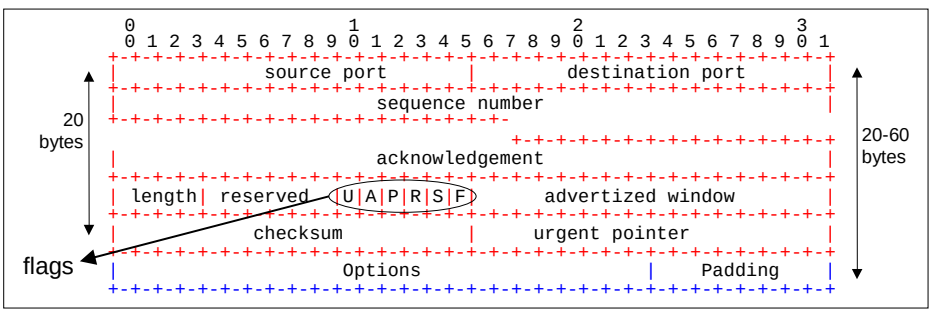


Figura 21: Capçalera TCP.

2.1. Establiment i terminació d'una connexió

La Figura 22 mostra les fases d'establiment (*three way handshaking*) 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.

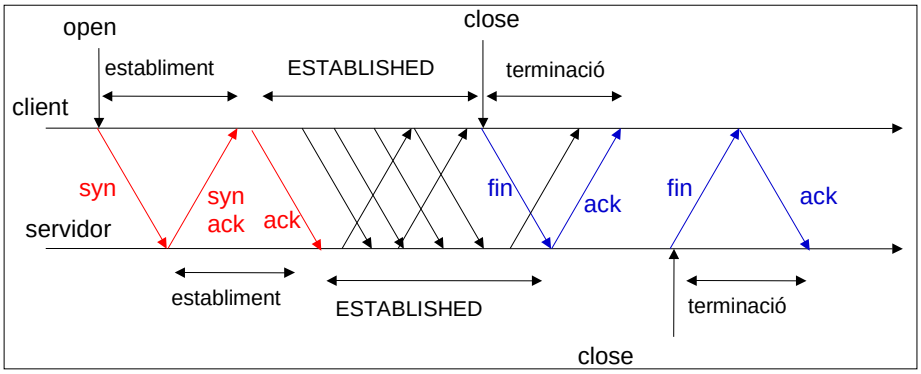


Figura 22: Establiment i terminació d'una connexió TCP.

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).

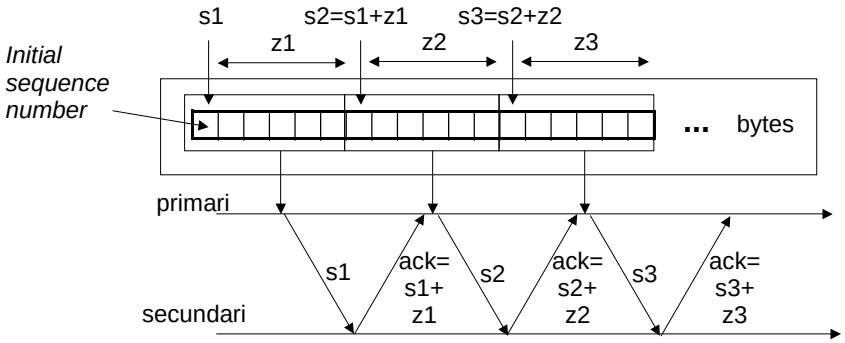
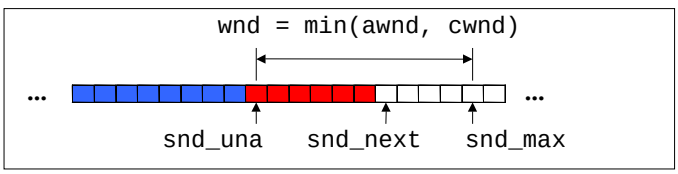


Figura 23: Evolució dels números de seqüència de TCP.

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.

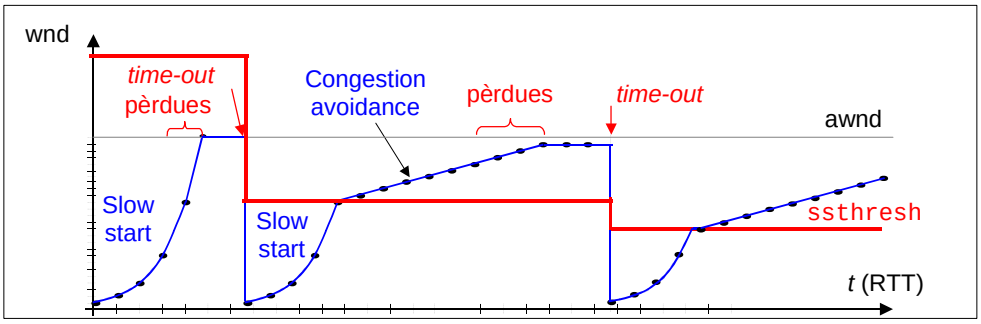


Figura 26: Evolució típica de la finestra de TCP quan hi ha un enllaç congestionat.

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.

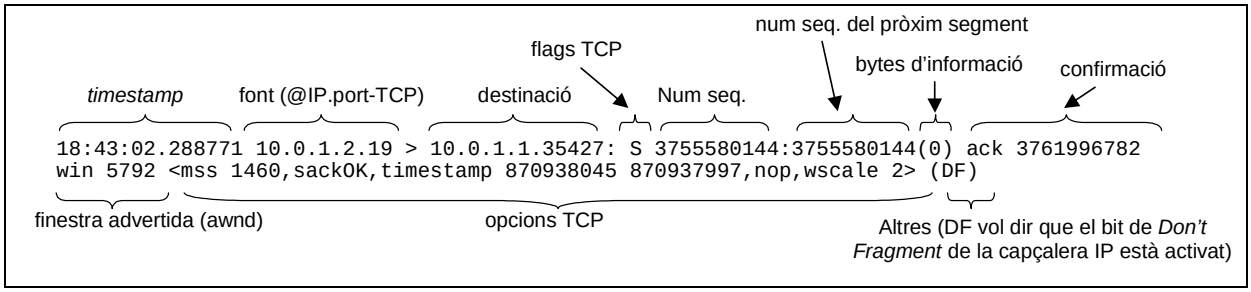


Figura 27: Bolcat d'un segment TCP.

```
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 ressolució 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*).

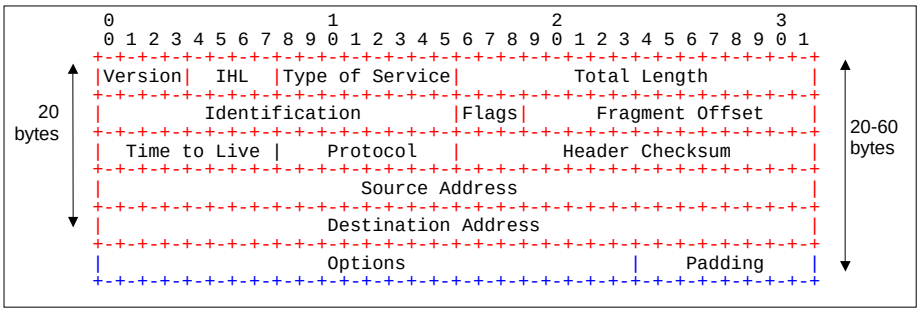


Figura 29: Capçalera IP.

4. 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.

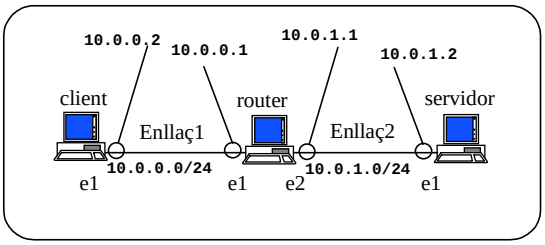


Figura 30: Topologia de la pràctica.

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.

Comentaris importants a tenir en compte en la imatge xarxes:

1) Si el servidor d'ftp rebutja la connexió del client: És perquè no pot fer la resolució inversa de l'adreça IP del client (mesura de seguretat implementada en el servidor). El problema es resol afegint l'adreça IP del client en el fitxer /etc/hosts del servidor (amb un nom arbitrari, per exemple "10.0.0.2 client").

2) Si el client de telnet triga molt en connectar-se al servidor: És perquè el client intenta fer la resolució inversa de l'adreça IP del servidor. Solució: esborrar els servidors de noms que hi ha en el fitxer /etc/resolv.conf del client.

3) Si en el client o el servidor es capturen segments TCP amb més de 1500 bytes, és degut a una optimització que fa Linux per incrementar l'eficiència de la comunicació amb el driver ethernet: Consisteix en passar varis MSS agregats en un sol segment, que el driver divideix en varis segments per ajustar-los la mida de la MTU. Per deshabilitar aquesta optimització cal executar:

```
# ethtool -K e0 tso off
# ethtool -K e0 gso off
```

4) Per a congelar el scroll de pantalla prémer Ctrl-S, per continuar Ctrl-Q.

4.1. Anàlisi dels segments d'una connexió TCP

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

1. Executa tcpdump per capturar els paquets en l'enllaç1 del client amb l'opció –X (Figura 32). En una altra finestra del client, estableix una connexió ftp al servidor, usuari "xc", executa la comanda "dir".
2. Executa "netstat –nat" en el client i el servidor (en una tercera finestra). Identifica els sockets que pertanyen a la connexió i l'estat de TCP. Tanca la connexió (comanda *bye*), i torna a mirar l'estat dels sockets amb netstat –nat.
3. En el bolcat tcpdump, identifica el *three-way-handshaking* i la terminació de la connexió tcp.
4. Mira els missatges que s'envien cada cop que s'executa una comanda en el client d'ftp. Relaciona les comandes que executes en l'aplicació ftp i els segments capturats. Fixa't com el contingut dels segments d'informació porten missatges ASCII sense encriptar. Busca el segment on s'ha enviat el password, i on s'ha executat la comanda "dir".

```
client# tcpdump –s 1500 –lnXi e1 port ftp
```

Figura 32: Captura de la traça ftp.

5. Engega tcpdump perquè capturi 200 paquets en l'enllaç e1 del client i ethernet del servidor, i connecta't al port de chargen (Figura 33). El servidor chargen (port 19) envia una seqüència de caràcters ASCII pseudo-aleatòria a la velocitat màxima que permet l'enllaç. Fixa't que el servidor és el que envia els segments d'informació.

A partir de les dues traces que has capturat:

6. 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.
7. Comprova si hi ha pèrdues. Justifica perquè n'hi ha, o no.
8. Observa la relació que hi ha entre els acks rebuts i el número de seqüència dels segments d'informació que s'envien 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?
9. 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*.
10. Mira l'evolució de la finestra advertida pel client i el servidor. Perquè penses que la del client augmenta 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)?

```
servidor# tcpdump -c 200 -nli e1 port chargen > captura-servidor.dmp
client# tcpdump -c 200 -nli e1 port chargen > captura-client.dmp
client# telnet 10.0.1.2 chargen
```

Figura 33: Captura de la traça d'una connexió al servidor de chargen.

11. 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?
12. 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
```

4.2. Connexió 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 34. Linux permet afegir aquesta cua amb la comanda de la Figura 35 (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:

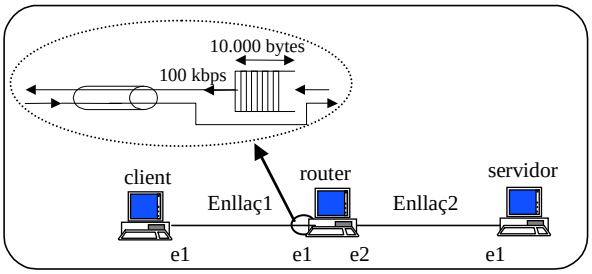


Figura 34: Cua que afegirem a la sortida de l'enllaç e1 del router.

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

Figura 35: Comanda per afegir la cua de la Figura 34.

13. Configura la cua i repeteix les comandes de la Figura 33.

    a) Llista la traça capturada en el servidor amb *less*. Busca quan hi ha primers acks duplicats i comprova que després el servidor retransmet el paquet 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.

14. 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 el RTT coincideix aproximadament amb aquest retard.

5. 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. Dóna 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. Estima la velocitat eficaç.
5. Digues els estats de TCP per els que passa el client i el servidor durant els 5 últims paquets que mostra el bolcat.

Lab 7. Domain Name System (DNS)

1. Introducció

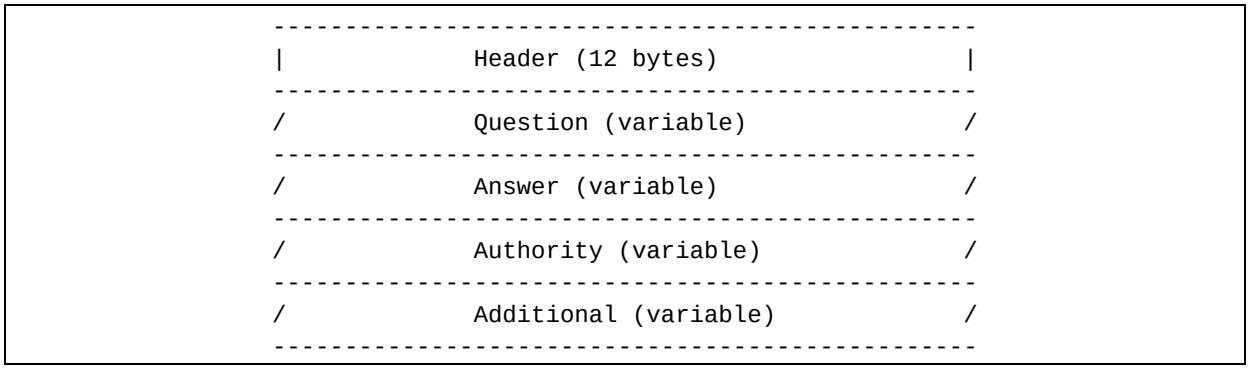
En aquesta pràctica s'aprofundirà en els protocol DNS. Per dur a terme els experiments que es detallen en l'enunciat utilitzarem una de les màquines del laboratori com a servidor DNS.

2. DNS

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

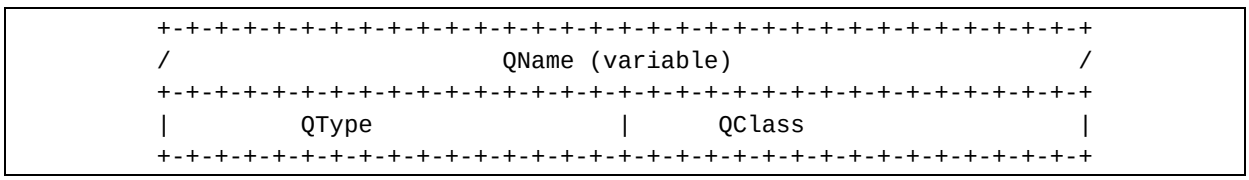
- SOA: Start Of Authority.
- NS: NS name.
- MX: the domain mail exchange.
- A: A host address.
- CNAME: Canonical Name Record. E.g. the real hostname of www.foo.org is server.foo.org.

Tots els missatges DNS tenen el format:



On la capçalera (header) és:

El camp question:



I els camps Answer, Authority i Additional són RRs:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                      Name (variable)                          /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Type                 |       Class                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            TTL                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         RDLenth               |      RData (variable)         /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3. Comandes bàsiques

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

3.1. Wireshark

Wireshark (antigament ethereal) és una eina d'anàlisi de xarxes. Permet monitoritzar amb una interfície gràfica el tràfic que circula per una interfície de xarxa. De fet, és l'equivalent gràfic a la comanda tcpdump. 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 com a usuari root la comanda 'wireshark' (teniu una icona disponible a la barra de tasques de l'escriptori). Es necessari que tingueu privilegis de superusuari 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 36). Un cop hagi finalitzat l'activitat que volem capturar, caldrà prémer el botó 'stop'.

Figura 36: Captura amb wireshark.

Un cop feta la captura podem filtrar els missatges d'un protocol concret, com mostra la Figura 37. 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.

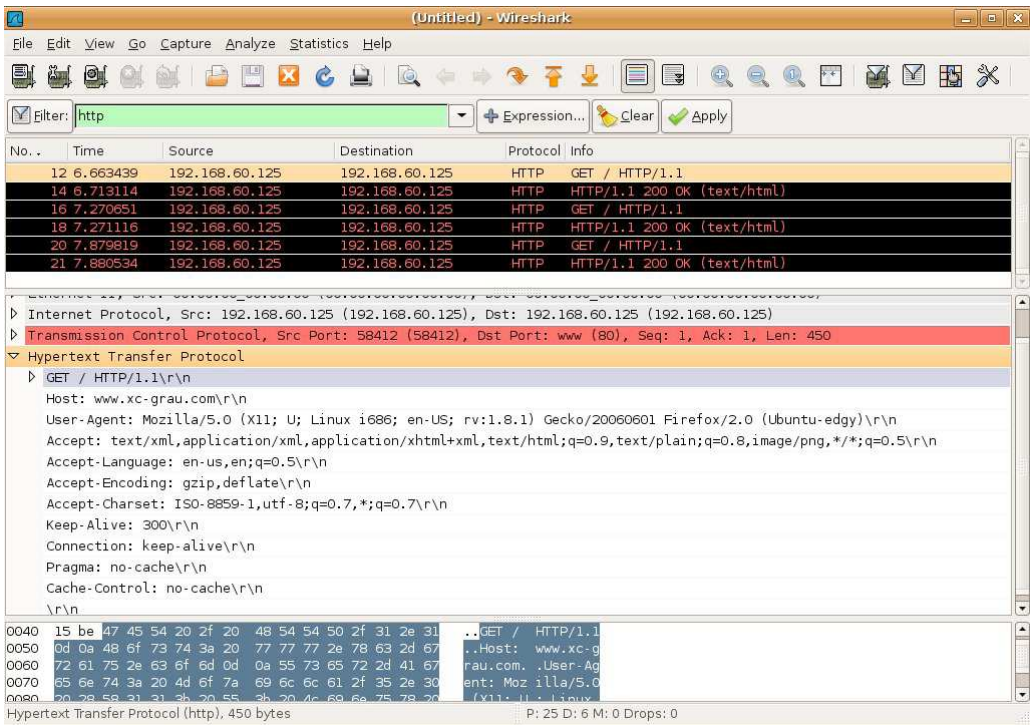


Figura 37: Filtre dels missatges http amb wireshark.

3.2. Comanda `nslookup`

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

```
nslookup
```

A continuació hi ha un extracte del la pàgina man:

INTERACTIVE COMMANDS

host: Look up information for host. If host is an Internet address and the query type is A or PTR, the name of the host is returned. If host is a name and does not have a trailing period, the search list is used to qualify the name. To look up a host not in the current domain, append a period to the name.

exit: Exits the program.

set keyword[=value]: This command is used to change state information that affects the lookups. Valid keywords are:

> **all**: Prints the current values of the frequently used options to set. Information about the current default server and host is also printed.
>
> **domain=name**: Sets the search list to name.
>
> **[no]search**: If the lookup request contains at least one period but doesn't end with a trailing period, append the domain names in the domain search list to the request until an answer is received.
>
> **type=value**: Change the type of the information query. (Default = A; abbreviations = q, ty)
>
> **[no]recurse**: Tell the name server to query other servers if it does not have the information. (Default = recurse; abbreviation = [no]rec)
>
> **[no]debug**: Turn on or off the display of the full response packet and any intermediate response packets when searching. (Default = nodebug; abbreviation = [no]deb).

Per exemple, per demanar l'adreça del servidor de DNS: www.xc-grau.test, suposant que xc-grau.test està en /etc/resolv.conf, executaríem:

```
# nslookup
> www
Server: 192.168.60.125
Address: 192.168.60.125#53
www.xc.test canonical name = pcserver.xc.test.
pcserver.xc.test canonical name = pc125.xc.test.
Name: pc125.xc.test
Address: 192.168.60.125
```

Per demanar un RR de tipus MX:

```
# nslookup
> set type=MX
> upc.edu
Server: 192.168.60.125
Address: 192.168.60.125#53
Non-authoritative answer:
upc.edu mail exchanger = 10 mx1.upc.es.
upc.edu mail exchanger = 10 mx2.upc.es.
```

Per tornar a demanar un RR de tipus A:

```
# nslookup
> set type=A
...
```

3.3. El fitxer resolv.conf

En aquest fitxer hi ha l'adreça IP del servidor de noms local. Per exemple, si l'adreça és 192.168.60.125:

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

Fixeu-vos que "nameserver" és la IP del servidor de noms local i "search" és el domini per defecte. Si hi ha varis nameservers (vàries línies "nameserver"), en cas de fallar la resolució, es demana seqüencialment. El domini per defecte s'afegeix en cas que el nom a resoldre no sigui complert.

3.4. El fitxer /etc/bind/db.root

En el fitxer "/etc/bind/db.root" hi ha els RRs dels root-servers. Aquesta informació es fa servir per accedir a l'arbre DNS. Aquest fitxer no s'ha de modificar, forma part de la instal·lació de bind.

3.5. El fitxer de zona d'un servidor DNS

Té els "resource records" (RR). Per exemple, el fitxer db.grupX.xc de l'aula conté la següent informació. Notar que aquest fitxer és una plantilla d'un fitxer de zona, on s'han de canviar les X per el valor que correspon, tal com s'explica més edavant. El RR tipus SOA (Start Of Authority) és de configuració. A continuació hi ha RRs de tipus NS (servidor de noms), A (adreces), CNAME (alias) i MX (Mail eXchange). També autoritats de subdominis (ns.grupX.xc.test del subdomini grupX.xc.test en l'exemple següent).

```
@       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 38. 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. Evidentment, el domini test és un domini de proves, i xc.test no existeix (no s'ha de configurar).

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 (pxX.grupX.xc.test) per a fer peticions, tal com mostra la figura.

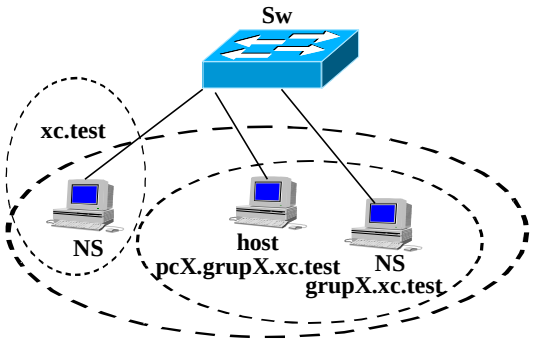


Figura 38. Xarxa a configurar. Només s'han de configurar pcX.grupX.xc.test i grupX.xc.test.

4.1. Configuració de la xarxa

1. Configurar els 2 PCs de la Figura 38 (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, modificant els canvis que s'hagin fet. 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 192.168.60.X
```

   on la X de l'adreça IP i de grupX.xc.test és el nombre del PC que es fa servir com a servidor de noms.

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'aquest moment.

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 'nslookup' contesteu les següents preguntes. Quins registres heu consultat en cada ocasió? (nslookup permet canviar el tipus de petició amb la comanda 'set type=(A,NS,MX,CNAME...)'). 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. Activa el mode debug en nslookup (set debug). Repeteix la resolució de www.microsoft.com. Compara la informació proporcionada per nslookup amb el contingut dels missatges.
9. Canvieu el mode del vostre client (nslookup) a no recursiu (usant la comanda 'set 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 el tràfic que es genera. Tenint en compte que els PCs del laboratori fan servir un Proxy-web per accedir a Intenet, s'observa alguna resolució de noms? Perquè?

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?