

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'aquest 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 dona 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.

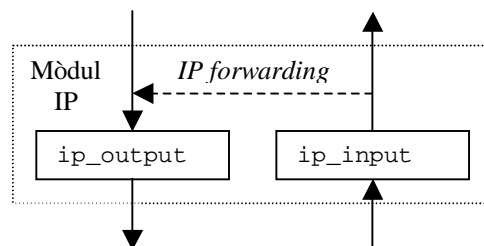


Figura 2: Funcions `ip_output` i `ip_input`.

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

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

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

Exemple 3: Configuració de la interfície e0.

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

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

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

I per activar-la de nou:

```
ifconfig e0 up
```

Comanda 3: Activació d'una interfície.

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

¹ Farem servir el següent conveni: les paraules clau estan en negreta i els paràmetres que dona l'usuari no.

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

Exemple 4: Llistat de les interfícies configurades.

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

```
ifconfig -a
```

Comanda 4: Llistat de les interfícies conegudes pel kernel.

4.2. Comanda route

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

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

Comanda 5: Us de la comanda route.

On | vol dir paràmetres alternatius i [] vol dir paràmetre opcional. Si no es dona la màscara i el SO assigna la de la classe. Si no es dona 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 Resolution 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 dona 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* podem 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.

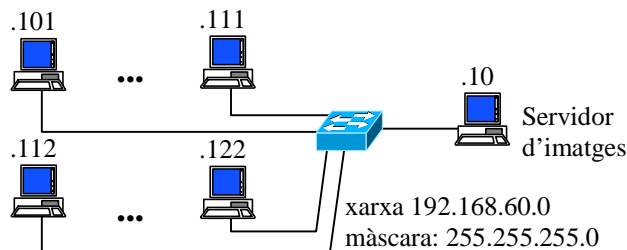


Figura 3: Xarxa del laboratori D6003.

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

constat 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.

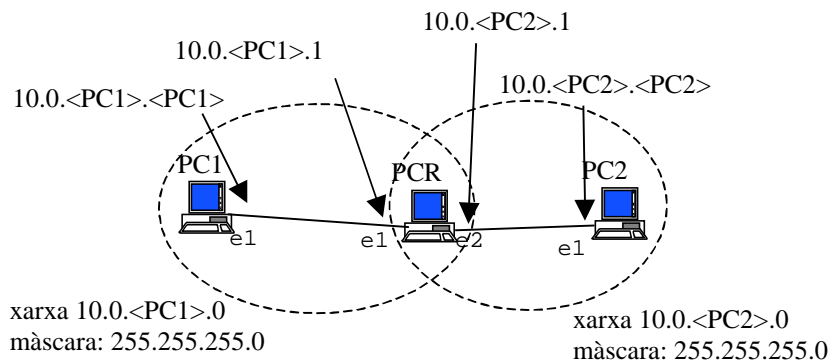


Figura 4: Topologia amb 1 router.

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 PC2 al PCR per comprovar que és accessible. Feu un *ping* des del PC1 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 sí 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) Ajustar 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.

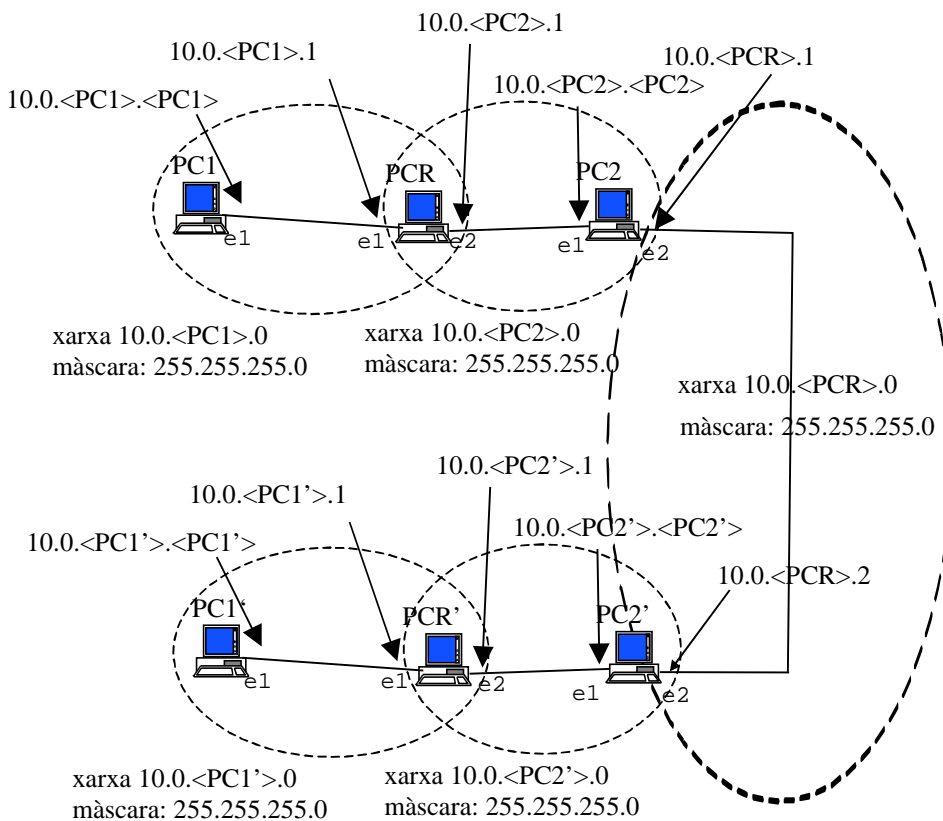


Figura 5: Topologia ajuntant les xarxes de 2 grups.

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

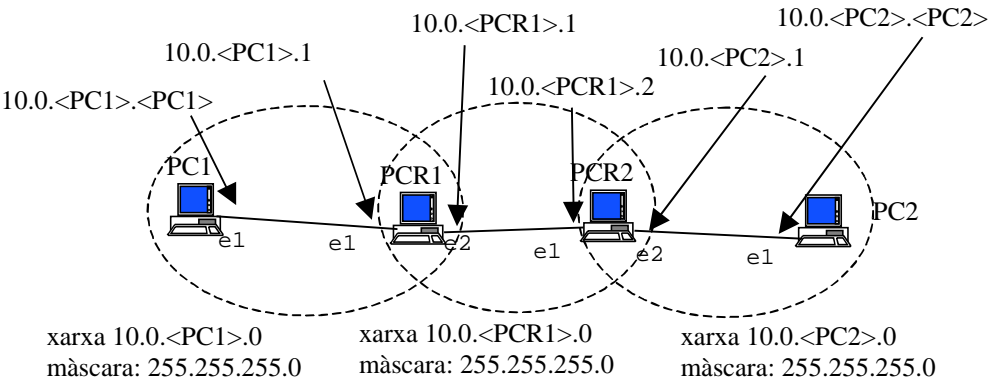
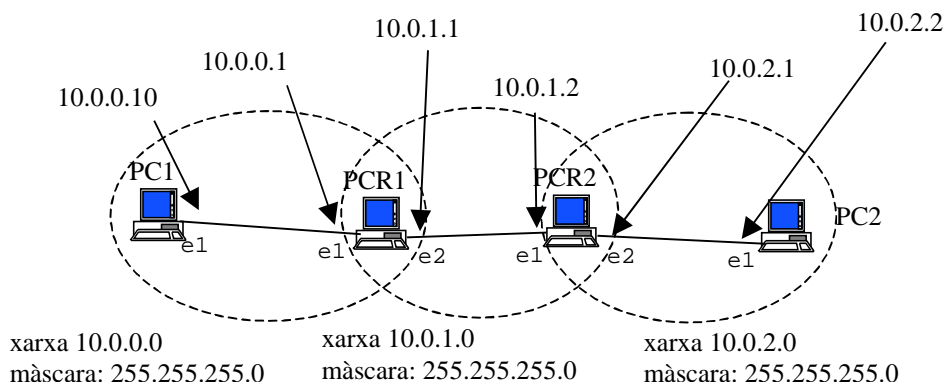


Figura 6: Topologia amb 2 routers.

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