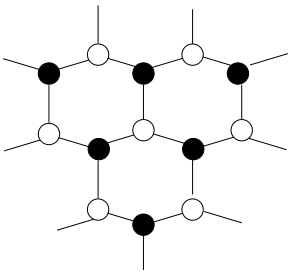


E52. Xarxes Informàtiques

Pràctica 6. El nivell de xarxa en Internet: Protocols IP, ARP i ICMP

Descripció de l'equip i del programari

1. Ordinadors de tipus PC amb S.O. GNU/Linux.
2. Adaptadors *Ethernet* dobles amb connectors RJ-45 (estàndards 10BaseT/100BaseT).
3. Commutador i concentrador 10BaseT.
4. Eines bàsiques de configuració de xarxa sota Linux.
5. Monitor de xarxa i analitzador de protocols *Ethereal*.

1 Introducció

El conjunt de protocols TCP/IP, desenvolupats en els anys 70 per la *Defense Advanced Research Projects Agency* dels E.U.A. i popularitzats a rel de la seua inclusió en el UNIX de Berkeley, consta d'una sèrie de normes de distints nivells que tenen com a característica comuna l'ús d'IP (*Internet Protocol*) per l'intercanvi de blocs d'informació, anomenats *datagrames*, entre estacions (o *hosts*) connectades a xarxes de qualsevol naturalesa. IP defineix un servei *sense connexió* i *no confirmat*, açò és, els datagrames s'encaminen entre l'estació d'origen i l'estació de destí (identificades per sengles adreces IP) sense necessitat de definir prèviament una ruta i, a més a més, el protocol no estableix mecanismes de comprovació per determinar amb certesa si un bloc d'informació ha arribat o no al seu destí. IP és un protocol del nivell de xarxa i, com a tal, la seua missió principal és la de l'*encaminament*: determinar la ruta millor per viatjar entre dues estacions. El camí seguit per un datagrama transmés sobre una xarxa d'àrea extensa implicarà normalment el pas per una sèrie de nodes intermedis (encaminadors o *routers*) que són capaços d'encaminar el datagrama de forma adequada en funció d'una adreça de destí. El format del datagrama IP, juntament amb una descripció completa dels seus camps, pot trobar-se en l'Annex II.

Les unitats de dades dels protocols del nivell de xarxa necessiten encapsular-se en unitats del nivell d'enllaç. En el cas concret de l'intercanvi d'un datagrama IP entre dues estacions d'una xarxa d'àrea local com la del laboratori, la trama d'enllaç que el va a transportar ha de contenir l'adreça física (MAC) de l'estació on va adreçat. Per esbrinar-la, el mòdul d'adaptació d'IP a MAC fa ús del protocol ARP (*Address Resolution Protocol*). Podeu trobar el format del paquet ARP i una descripció dels seus camps en l'Annex I.

Per portar a terme amb eficàcia la funció d'encaminament, el protocol IP fa servir al seu torn ICMP (*Internet Control Message Protocol*), que s'encarrega d'enviar missatges informatius a l'estació que ha originat un datagrama quan aquest, per alguna raó, no pot arribar al seu destí o es troba amb qualsevol problema en el seu camí cap a l'estació destinatària. El format del paquet ICMP està detallat en l'Annex III.

2 Desenvolupament de la pràctica

2.1 Abans de començar...

Per realitzar aquesta pràctica caldrà que vos agrupeu de la següent forma: formeu dos grups de 2-3 persones i prengueu 2 ordinadors per cada grup (a ser possible contigus). Designeu un dels dos ordinadors com a *node local* i l'altre com a *node passarel·la*. Més endavant veurem la utilitat d'aquesta designació. Els dos grups treballaran conjuntament en l'últim apartat de la pràctica, però poden fer-ho per separat en la resta.

En aquesta pràctica anem a usar especialment els programes `ifconfig`, `route` i `ethereal`, així com `arp`, `ping` i `mtr`. Podeu consultar l'Annex IV i les pàgines corresponents del manual per aprendre com usar-los. Sempre que pugueu, eviteu l'accés al servei de resolució de noms (DNS), usant adreces IP en lloc de noms d'estació i indicant als programes que no proven de trobar aquests noms (a sovint amb l'opció `-n`).

Teniu en compte també que per modificar la configuració de la xarxa d'una màquina o per capturar trames caldrà que actueu amb permís de superusuari (*root*). Assegureu-vos d'escollir cada volta la interfície adequada on capturar. Finalment, recordar-vos que és molt aconsellable que aneu apuntant les ordres exactes que executeu. Vos seran molt útils per refer la vostra configuració en cada nova sessió.

2.2 Recollida d'informació dels ordinadors

Useu els programes `ifconfig` i `route` per emplenar les taules següents. Useu també el programa `host` per trobar l'adreça IP corresponent a Anubis.

	Node passarel·la	Node local
Adreça MAC de la interfície pública (`eth0`)		
Adreça IPv4 de la interfície pública		
Adreça MAC de la interfície privada (`eth1`)		
Adreça IPv4 de la interfície privada		

Màscara de subxarxa pública	
Màscara de subxarxa privada	
Adreça IPv4 de l'encaminador per defecte	
Adreça IPv4 d'`anubis.uji.es`	

2.3 Estudi de la xarxa del laboratori

Anem a estudiar l'estructura de la xarxa de la UJI i del laboratori de Sistemes Operatius i Xarxes (TD1108DL). Amb la informació anterior en la mà, contesteu les preguntes següents:

- A quina classe (A, B, C, D, E) pertanyen les adreces de la UJI? Segons això, quina és l'adreça i màscara de la xarxa de la UJI? Quantes estacions seria possible connectar a una xarxa d'aquest tipus?
- Quina és la màscara de subxarxa que s'usa en la xarxa de la UJI? En quantes subxarxes queda dividida? Quantes estacions es poden connectar a cadascuna d'elles?
- Es troben els dos nodes del vostre grup dins de la mateixa subxarxa, si teniu en compte les seues adreces IP privades (192.168.X.Y)? Com ho deduïu?

- Es troben els dos nodes del vostre grup dins de la mateixa subxarxa, si teniu en compte les seues adreces IP públiques (150.128.X.Y)? Com ho deduïu?
- Escolliu un dels nodes. Es troba dins de la mateixa subxarxa que Anubis, si teniu en compte les seues adreces IP públiques (150.128.X.Y)? Com ho deduïu?

2.4 Xarxes locals i ARP

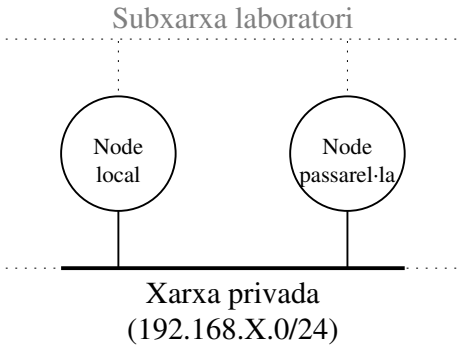


Figura 1: Configuració inicial de la xarxa.

Per cada grup, anem a reconfigurar els nodes local i passarel·la de forma que es troben només ells dos dins d'una nova xarxa IP privada com la de la figura 1, diferent de l'anterior que cobria tot el laboratori. La nova xarxa privada del grup serà 192.168.X.0, on X serà el valor de l'últim octet de l'adreça IP privada que tenia el vostre node passarel·la. Així, si el node passarel·la era 192.168.49.182, la xarxa serà 192.168.182.0.

Primer, escolliu una nova adreça privada dins de la nova xarxa per cada node i empleneu aquestes taules. En acabant, seguiu els passos inferiors en ambdós nodes.

	Node passarel·la	Node local
Nova adreça IP de la interfície privada (`eth1`)		

Nova adreça de xarxa privada	
Nova màscara de subxarxa privada	255.255.255.0

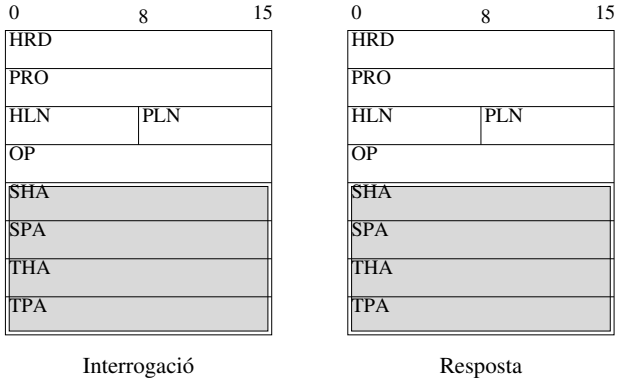
1. Executeu `killall dhcpcd` per detenir l'autoconfiguració de les interfícies. Això les inhabilitarà i deixarà incomunicades les vostres màquines, cosa que podeu comprovar usant `route` i `ping`.

2. Useu `ifconfig` per assignar a la interfície privada del node la seua nova adreça. Comproveu com heu fet abans si la configuració ha funcionat.

   Si feu servir `route` veureu que s'ha afegit automàticament una ruta cap a la nova xarxa. Quina seria l'ordre exacta per crear aquesta ruta manualment?

Anem a comprovar el funcionament del protocol ARP. Per fer-ho ens convé, abans de començar, invalidar totes les entrades de la taula ARP del sistema operatiu usant el programa `arp` (vegeu l'Annex IV). Això forçarà la realització de les consultes ARP que estudiarem a continuació. Escolliu un sol dels nodes del grup i feu-hi el següent:

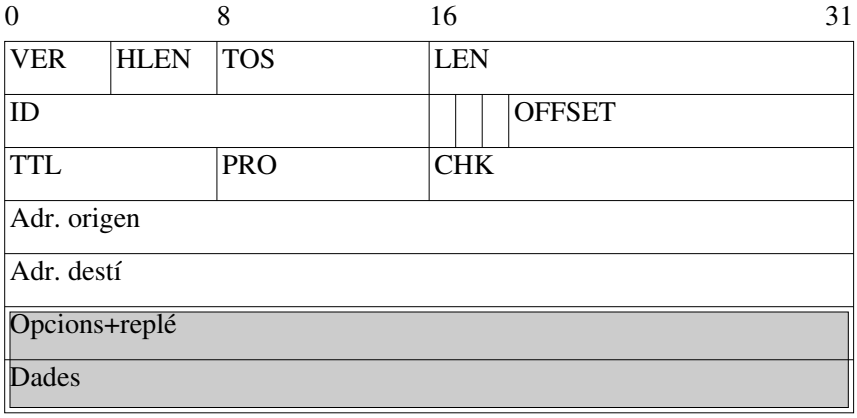
- Llanceu Ethereal com a *root* en el node i en *Capture/Start* establiu un filtre per capturar únicament els paquets ARP originats en, o destinats a l'adreça MAC privada d'aquest node. Inicieu la captura.
- Llanceu un sol paquet de *ping* cap a la IP privada de l'altre node i pareu la captura.
- Quina és l'adreça MAC de destí del primer paquet originat per aquest node? Per què? Per què les trames rebudes de l'altre node apareixen com d'una mida superior a la necessària?
- Elegiu un paquet ARP d'interrogació i un altre de resposta i empleneu, de forma intel·ligible, els camps corresponents a les dues capçaleres ARP següents:



2.5 IP i ICMP

Anem a analitzar el funcionament bàsic d'IP en una xarxa local. Caldrà que:

- Escolliu un node i hi establiu amb Ethereal un filtre per capturar únicament els datagrames IP originats en, o destinats a l'adreça IP privada d'aquest. Inicieu la captura.
- Llanceu un sol paquet de *ping* cap a la IP privada de l'altre node.
- Llanceu un sol paquet de *ping* amb una mida de 2040 octets de dades cap a la IP privada de l'altre node (cal que ho feu com a *root*). Pareu la captura.
- Analitzeu el datagrama de la primera petició de *ping*. Estudieu els camps de la capçalera IP i empleneu, de forma intel·ligible, l'esquema següent:



- Estudieu les capçaleres IP de les dues peticions de *ping*. S'ha fragmentat el primer *ping*? I el segon? Com ho sabeu?
- Respongueu: com estan relacionats els datagrames pertanyents al *ping* fragmentat? Com es sap que el datagrama de l'altre *ping* no té a veure amb aquests fragments?

Ara estudiarem per damunt alguns detalls dels paquets ICMP usats per `ping`: les peticions d'eco i les respostes d'eco.

- Escolliu un node i establiu-hi amb Ethereal un filtre per capturar únicament els paquets ICMP originats en, o destinats a l'adreça IP privada d'aquest. Inicieu la captura.
- Llanceu uns pocs paquets de *ping* cap a la IP privada de l'altre node. En acabant, torneu a llançar la mateixa ordre. Finalment, pareu la captura.
- Estudieu els camps de les capçaleres ICMP dels paquets capturats. En què coincideixen les peticions d'eco d'una mateixa execució de `ping`? En què es diferencien? I en què es diferencien les peticions d'eco de dues execucions diferents?

2.6 Encaminament IP

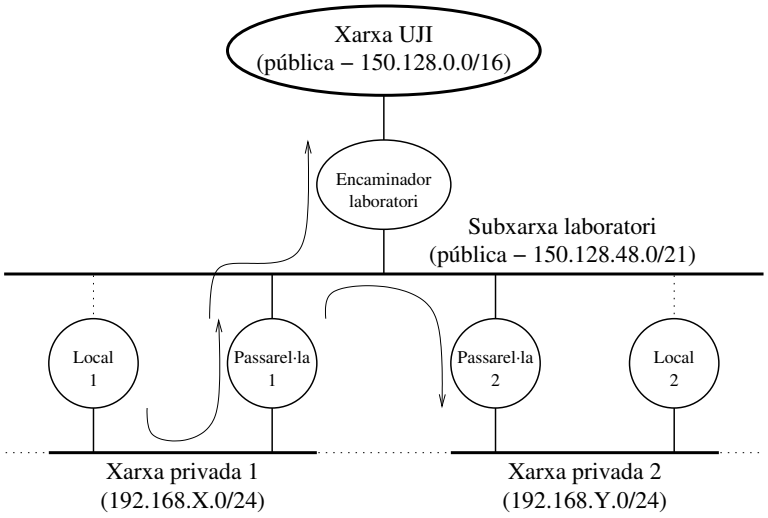


Figura 2: Configuració final de la xarxa.

Per realitzar aquest apartat caldrà que treballeu *els dos grups de forma coordinada*. Anem a provar de reproduir la configuració de la figura 2 (les fletxes mostrades també són aplicables —en sentit invers— a la xarxa privada 2). Per aconseguir-ho seguirem els passos següents:

1. Per *tenir accés a la subxarxa pública del laboratori*, activarem amb `ifconfig` la interfície pública de cada node passarel·la, assignant-li la seua adreça i màscara originals. Feu *ping* entre les dues màquines (fent servir la IP pública) per comprovar la configuració.
2. Per *poder accedir a l'exterior*, caldrà configurar un encaminador d'eixida en els dos nodes passarel·la. Useu la sintaxi específica de `route` per establir-hi com a *encaminador per defecte* el que tenien en un principi. Comproveu que l'accés funciona fent un *ping* a Anubis.

3. Per *intercomunicar les dues xarxes privades* usarem la subxarxa pública del laboratori. Afegiu amb `route` en cada node passarel·la una ruta cap a la xarxa privada de l'altre grup a través del node passarel·la d'aquest últim. Proveu la ruta fent *ping* a la IP privada de l'altre node passarel·la.

4. Per *donar eixida als nodes locals*, establiu com a encaminador per defecte en cadascun d'ells el node passarel·la del mateix grup. Si podeu fer *ping* entre els nodes locals, ho heu fet bé.

Una volta ho tingueu tot llest, cal que cada grup realitze les experiències següents i conteste les qüestions que s'hi indiquen:

- Useu `mtr` en el node local per traçar per quins nodes passen els datagrames IP destinats a les següents adreces:
  - La privada del vostre node passarel·la.
  - La pública del node passarel·la de l'altre grup.
  - La privada del node passarel·la de l'altre grup.
  - La privada del node local de l'altre grup.

  Concorden els resultats dels traçats amb els vostres coneixements d'encaminament IP?

- Realitzeu un traçat fins a Anubis des del node local i des del passarel·la. Quin és el primer node per on passa cada traçat? Arriben tots dos traçats al seu destí? Per què suposeu que ocorre açò?

- Realitzeu en un dels nodes passarel·la la captura d'alguns paquets de *ping* originats en el propi node, adreçats a Anubis. Obteniu l'adreça MAC de destí d'aquests paquets. A qui penseu que pertany aquesta adreça? Per què? Podeu comprovar si la vostra resposta és correcta usant el programa `arp`.

3 Conclusions

- Quina estratègia es segueix en la UJI per tenir diverses subxarxes partint de l'assignació de la xarxa IP única 150.128.0.0?

- A la vista dels paquets de dades que heu analitzat, diríeu que els protocols ARP i ICMP són del mateix nivell? Diríeu que ARP i IP són del mateix nivell, basant-vos en la seua situació en la pila de protocols? I basant-vos en la seua funcionalitat?

- Seríeu capaços d'esbossar una breu descripció algorísmica de les decisions i accions que realitza el nivell de xarxa IP d'una màquina a l'hora d'enviar un datagrama IP a una altra màquina qualsevol?

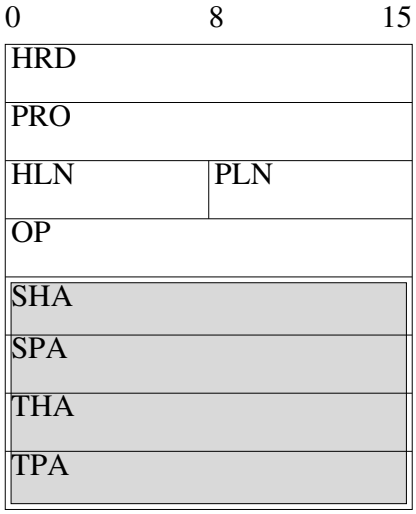


Figura 3: Estructura d'un paquet ARP.

Annex I. Format del paquet ARP

L'estructura del paquet ARP es mostra en la figura 3. Els camps ombrejats tenen longitud variable, que depén dels valors dels camps HLN i PLN. El camps del paquet tenen el significat següent:

HRD Tipus d'adreça física o de maquinari (e.g. Ethernet).

PRO Tipus d'adreça del nivell de xarxa o de protocol (e.g. IP).

HLN Longitud de l'adreça física, en octets.

PLN Longitud de l'adreça del nivell de xarxa, en octets.

OP Codi d'operació (`REQUEST` o `REPLY`).

SHA Adreça física de l'emissor.

SPA Adreça del nivell de xarxa de l'emissor.

THA Adreça física del destinatari (a 0 si `OP=REQUEST`).

TPA Adreça de nivell de xarxa del destinatari.

Annex II. Format del datagrama IP

L'estructura del datagrama IP es mostra en la figura 4. Cada fila representa una paraula de 32 bits. Els camps ombrejats en la figura tenen longitud variable. Els camps del datagrama tenen el significat següent:

VER Versió del protocol IP.

IHL (*IP Header Length*) Indica la longitud, en paraules de 32 bits, de la capçalera del datagrama. El seu valor mínim és 5 (20 octets).

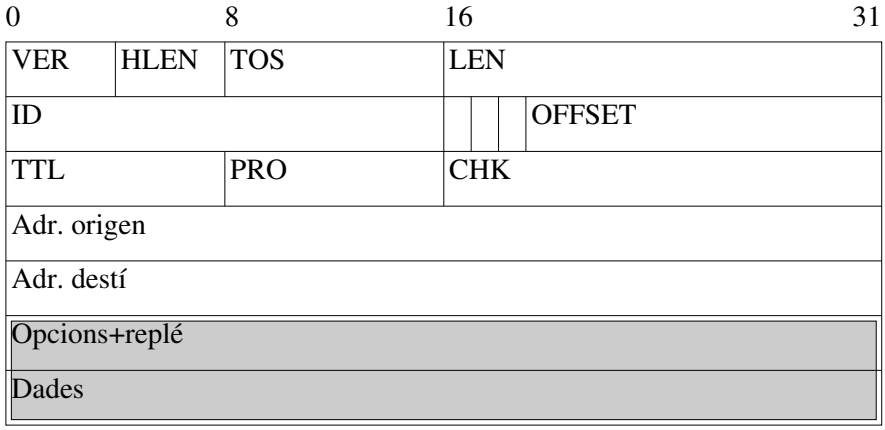


Figura 4: Estructura d'un datagrama IP.

TOS (*Type Of Service*) Indica el tipus de servei que l'estació espera de la subxarxa. Sol ser un compromís entre fiabilitat i velocitat; la forma de sol·licitar un servei determinat per un datagrama és especificant-lo segons l'esquema següent:

P1	P2	P3	D	T	R		

El tres primers bits codifiquen la prioritat del datagrama (000 és la mínima, 111 la màxima); el bit D posat a 1 indica que cal minimitzar el retard en la seua transmissió; el bit T, que s'ha de tractar de triar les rutes de major ample de banda; i l'R, que cal cercar la màxima fiabilitat. Els dos últims bits estan reservats. Aquestes opcions estan contemplades pel protocol IP, tot i que rara volta en fan ús els protocols usuaris.

LEN És la mesura, en octets, de la mida total (capçalera més dades) del datagrama. La mida màxima del datagrama és, per tant, de 65536 octets.

ID Enter que utilitza l'emissor per identificar cadascun dels seus datagrames.

FLAGS Aquest camp i el següent serveixen per controlar la fragmentació que es pot produir en en tràfic entre xarxes. El camp de marcadors (*flags*) està format per tres bits, dels quals el primer no s'usa i es transmet a 0; el següent és el bit DF (*Don't Fragment*), que es posa a 1 per indicar que aquest datagrama o fragment IP no pot tornar a ser fragmentat; el tercer és el bit MF (*More Fragments*), que indica, quan val 0, que el fragment és l'últim d'un datagrama original.

OFFSET Posició relativa d'un fragment dins del datagrama original. Es medeix en unitats de 8 octets. Si un datagrama no pot ser fragmentat, DF=1, MF=0 i OFFSET=0.

TTL (*Time-To-Live*) Manté un comptador que és decrementat per cada entitat que processa la capçalera, de forma que quan una entitat IP detecta un datagrama amb TTL=0, el descarta.

PRO Identifica el protocol del nivell superior que ha de rebre el datagrama en l'estació de destí. Els valors més comuns són 6 per TCP, 17 per UDP i 1 per ICMP.

CHK S'usa per detectar errors en la transmissió de la capçalera. No es realitza cap comprovació sobre el camp de dades. El valor que s'emmagatzema és el complement a u de la suma (usant aritmètica de complement a u i prenent el camp CHK com a 0) de totes les paraules de 16 bits de la capçalera.

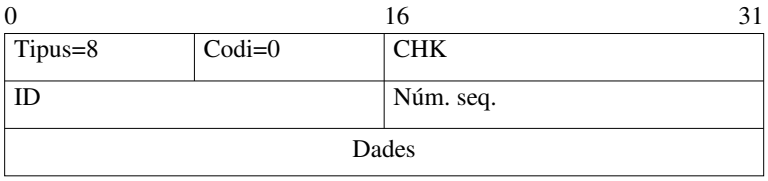
Adreces Els quatre octets de les adreces IP d'origen i destí del datagrama.

Opcions És un camp de longitud variable i opcional en els datagrames, tot i que qualsevol entitat IP ha d'implementar la funcionalitat requerida. Un datagrama pot incloure més d'una opció. Cadascuna d'aquestes vindrà identificada per un camp de codi d'opció (d'un octet) que, en alguns casos, anirà seguit d'un camp de longitud (d'un octet) i aquest, al seu torn, determinarà la mida d'altre camp que contindrà les dades específiques de l'opció.

Annex III. Format del paquet ICMP

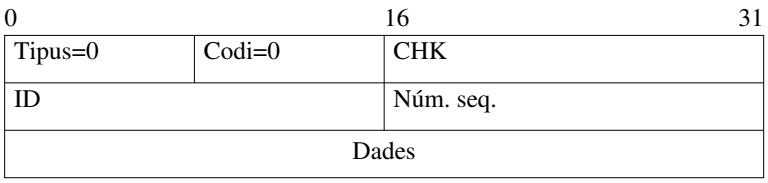
El format del paquet ICMP depén de cadascun dels missatges que s'envie per comunicar situacions especials o anòmales que s'hagen pogut produir al llarg de la ruta que segueix un datagrama. Aquest protocol fa ús d'IP per la transmissió dels seus blocs d'informació. El primer octet del paquet indica el tipus de missatge, el segon un codi d'informació addicional, els octets tercer i quart contenen la suma de comprovació (*checksum*) d'ICMP, que es calcula de forma idèntica a la de la capçalera del datagrama IP. Els tipus de missatge més comuns i els seus formats corresponents es descriuen a continuació.

Echo request (Sol·licitud d'eco)



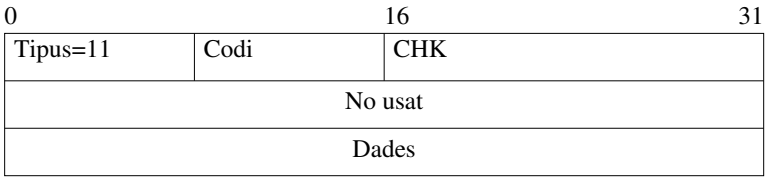
És una petició d'eco d'una entitat ICMP a una altra entitat remota del mateix protocol. Una petició d'eco pot constar de diversos paquets; en aquest cas el camp *Identificador* (ID) indica la petició, mentre que el camp *Número de seqüència* identifica els paquets individuals. La resposta serà un paquet del tipus 0 (resposta d'eco). El camp de dades es replena amb caràcters arbitraris.

Echo reply (Resposta d'eco)



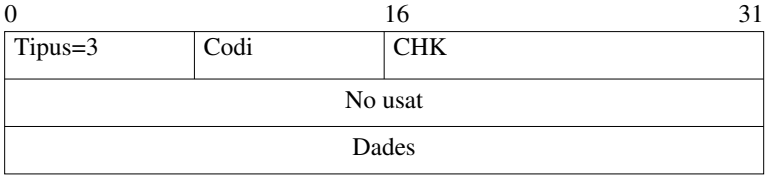
És una resposta a una petició d'eco. Una entitat ICMP envia a una altra una sol·licitud, i aquesta última li respon amb un paquet d'aquestes característiques. Els camps *Identificador* i *Número de seqüència* són els descrits en el paquet de sol·licitud. En el camp de dades es retorna la mateixa informació que portara el paquet de sol·licitud.

Time Exceeded (Temps de vida excedit)



Quan un encaminador IP detecta un datagrama amb el camp TTL igual a 0, ha de descartar-lo. Aquest missatge serveix per informar l'entitat IP emissora d'aquest fet. En el cas que un datagrama haja sigut fragmentat, aquest missatge pot ser generat pel fet que el desinatari no haja rebut tots els fragments i no puga, per tant, reconstruir el datagrama original en un temps prefixat. El primer problema s'indica amb un valor 0 en el camp de *Codi*, mentre que el segon s'hi indica amb un valor 1. El camp de *Dades* conté la capçalera IP bàsica i els primers 64 bits de dades del datagrama que ha originat l'error.

Destination Unreachable (Destí inabastable)



S'envia a l'emissor d'un datagrama quan un encaminador no pot fer-lo arribar al seu destí per alguna raó. El camp de *Codi* indica el tipus de problema; per exemple, 0 si la xarxa no és abastable, 1 si l'estació no és abastable, 2 si el protocol no és abastable, etc. El camp de *Dades* conté la capçalera IP bàsica i els primers 64 bits de dades del datagrama que ha originat l'error.

Annex IV. Els programes `arp`, `ifconfig` i `route`

Manipulació de la taula ARP: `arp`

El programa `arp(8)` és capaç de mostrar i modificar la taula ARP del nucli, que conté les correspondències entre adreces IP, adreces MAC i interfícies. Aquesta és la sintaxi necessària per aquesta pràctica:

Consulta de la taula ARP

Sintaxi: `arp [-n]`

En executar `arp` sense arguments se'ns mostrarà l'estat acutal de la taula ARP. Cada línia representa una entrada d'aquesta, està encapçalada per l'adreça IP d'una estació (si s'ha indicat l'opció `-n`; el seu nom en cas contrari), i conté l'adreça MAC associada i el seu tipus, així com la interfície per on és accessible, entre altres valors. Les entrades invalidades apareixen amb el valor `(incomplete)` en l'adreça MAC.

Invalidació d'entrades de la taula ARP

Sintaxi: `arp -d ESTACIÓ`

Les entrades de la taula ARP només són eliminades quan caduquen o quan se'n desactiva la interfície associada. Malgrat això, una entrada pot ser invalidada en qualsevol moment, de forma que es force la petició ARP en intentar trobar l'adreça MAC de la IP associada. Les entrades invalidades també caduquen.

Configuració de les interfícies: `ifconfig`

El programa `ifconfig(8)` s'utilitza per manipular la configuració de les interfícies de xarxa del nucli Linux, activant-les, desactivant-les i assignant-los adreces de diversos protocols de xarxa. Vegem la sintaxi bàsica que usarem en aquesta pràctica (orientada a IP; podeu obtenir més informació en la pàgina corresponent del manual):

Consulta de la configuració de les interfícies

Sintaxi: `ifconfig`
`ifconfig INTERFÍCIE`

La primera forma mostra la configuració de totes les interfícies actives del nucli, la segona la d'una interfície concreta. Aquesta informació inclou (entre altres): el tipus de dispositiu o *Link encap*, l'adreça física o *HWaddr*, les adreces assignades dels diferents protocols de xarxa (*inet* en el cas d'IP, inclou l'adreça de difusió o *Bcast*, i la màscara de subxarxa o *Mask*, associades), els marcadors de funcionament, la unitat màxima de transferència o *MTU* i la mètrica (o *distància* fins la xarxa). Per exemple, `ifconfig eth0` mostra la configuració del primer dispositiu Ethernet.

Assignació d'adreces a una interfície

Sintaxi: `ifconfig INTERFÍCIE ADREÇA [netmask MÀSCARA_SUBXARXA]`

Aquesta ordre associa l'adreça IP especificada a la interfície indicada, de forma que (i) els paquets IP que s'envien per la interfície porten com a adreça d'origen aquesta adreça IP (ii) s'estableix el mecanisme ARP sobre la interfície per respondre amb la seua adreça física quan es detecten en el seu enllaç peticions ARP de l'adreça IP configurada.

L'argument opcional `netmask` s'utilitza per fixar una màscara de subxarxa diferent a la que correspon per defecte a la classe de l'adreça. Amb l'adreça i la màscara de subxarxa, el sistema és capaç d'afegir automàticament una ruta a la xarxa corresponent[1]. Així, per assignar l'adreça 10.0.1.2 a la interfície `eth1`, amb màscara 255.255.255.0 (diferent a la màscara per defecte, 255.0.0.0) executariem `ifconfig eth1 10.0.1.2 netmask 255.255.255.0`, que afegiria de passada una ruta a la xarxa 10.0.1.0/255.255.255.0 a través de la interfície `eth1`.

Desactivació d'una interfície

Sintaxi: `ifconfig INTERFÍCIE down`

Aquesta és la forma d'inhabilitar una interfície determinada. En fer açò totes les adreces associades a ella deixen d'estar actives. A més a més, les rutes que usen aquesta interfície

[1] La majoria de les voltes aquesta ruta serà la desitjada, però hi haurà casos on no, així que no està de més comprovar si aquesta és correcta amb `route`.

són eliminades, així com tots els parells d'adreces de xarxa i enllaç resolts via ARP sobre la interfície.

Manipulació de les rutes: `route`

El programa `route(8)` s'usa per configurar les rutes estàtiques del nucli cap a les xarxes IP, ja siga directament a través d'interfícies (xarxes locals) o indirectament a través d'encaminadors (xarxes no locals). Aquesta és la sintaxi de les ordres que usarem en aquesta pràctica:

Consulta de les rutes

Sintaxi: `route [-n]`

Si executem `route` sense arguments, ens mostrarà una taula amb les rutes IP actuals del nucli. Per cada línia (ruta) veurem, entre altres, a quina xarxa porta (*Destination*), la màscara de subxarxa (*Genmask*), i a través de quin encaminador (*Gateway*) o interfície (*Iface*) s'hi arriba. La xarxa especial 0.0.0.0 o `default` indica una ruta per defecte, i l'encaminador 0.0.0.0 o `*` que l'accés a la xarxa en qüestió és directe (xarxa local). L'opció `-n` evita la resolució dels noms de les xarxes i encaminadors.

Addició d'una ruta directa a xarxa

Sintaxi:
```
route add -net XARXA netmask MÀSCARA_SUBXARXA
  dev INTERFÍCIE
```

Aquesta ordre afig una ruta directa cap a la xarxa especificada amb la màscara de subxarxa corresponent. La xarxa és local i es troba accessible directament via la interfície esmentada. Es poden afegir diverses rutes a una mateixa xarxa, sempre que no usen la mateixa eixida (interfície o encaminador). Per exemple, l'ordre `route add -net 10.0.1.0 netmask 255.255.255.0 dev eth1` afegiria la ruta automàtica esmentada en l'exemple d'assignació d'adreces amb `ifconfig`.

Addició d'una ruta indirecta a xarxa

Sintaxi:
```
route add -net XARXA netmask MÀSCARA_SUBXARXA
  gw ENCAMINADOR
route add default gw ENCAMINADOR
```

Aquesta ordre afig una ruta indirecta cap a la xarxa especificada a través de l'encaminador indicat, que ha de ser abastable en alguna xarxa local. La primera és la sintaxi genèrica per establir una ruta cap a una xarxa determinada amb la corresponent màscara de subxarxa. La segona és una sintaxi específica per establir una *ruta per defecte*, és a dir, una ruta per on enviar els paquets pels quals no existesca cap altra ruta millor. En tots dos casos es troba la interfície adequada cap a l'encaminador automàticament fent servir la resta de les rutes.

Per exemple, `route add -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.1` establiria 10.0.1.1 com a encaminador per arribar a la xarxa 10.0.2.0/255.255.255.0.

Eliminació d'una ruta a xarxa

Per eliminar una ruta només cal executar la mateixa ordre que hauríem invocat per afegir-la, però substituint-hi l'argument `add` per `del`. Així doncs, per eliminar la ruta de l'exemple anterior escriuriem `route del -net 10.0.2.0 netmask 255.255.255.0 gw 10.0.1.1`.