Administració de xarxa

Remo Suppi Boldrito

PID_00167526

Universitat Oberta de Catalunya
www.uoc.edu

Índex

Introducció

El sistema operatiu Unix (GNU/Linux) es pren com a exemple d'una arquitectura de comunicacions estàndard. Des del mític UUCP (servei de còpia entre sistemes operatius Unix) fins a les xarxes actuals, el Unix sempre ha mostrat la seva versatilitat en aspectes relacionats amb la comunicació i l'intercanvi d'informació. Amb la introducció de xarxes d'ordinadors (àrea local LAN, àrea àmplia WAN o les més actuals, àrea metropolitana MAN) amb enllaços multipunt a diferents velocitats (56 kbps, fins a 1 Gbps), han anat sorgint nous serveis basats en protocols més ràpids, portables entre diferents ordinadors i més ben adaptats, com el TCP/IP (*transport control program / Internet protocol*) [Com01, Mal96, Cis00, Gar98, KD00].

1. Introducció a TCP/IP (paquet TCP/IP)

El protocol TCP/IP sintetitza un exemple d'estandardització i una voluntat de comunicació a escala global.

El protocol TCP/IP és en realitat un conjunt de protocols bàsics que s'han anat agregant al principal per a satisfer les diferents necessitats en la comunicació ordinador-ordinador, com són TCP, UDP, IP, ICMP, ARP.[Mal96]

La utilització més freqüent de TCP/IP per a l'usuari actualment són la connexió remota a altres ordinadors (*telnet,* SSH[1]), la utilització de fitxers remots (NFS[2]) o la transferència (FTP[3], HTTP[4]).

(1)De l'anglès *secure shell*.

(2)De l'anglès *network file system*.

(3)De l'anglès *file transfer protocol*.

(4)De l'anglès *hipertext markup protocol*.

1.1. Serveis sobre TCP/IP

Els serveis TCP/IP tradicionals més importants són [Gar98]:

a) Transferència d'arxius: l'FTP permet a un usuari d'un ordinador obtenir o enviar arxius d'un ordinador a un altre ordinador. Per a això, l'usuari haurà de tenir un compte, en l'ordinador remot, identificar-se per mitjà del seu nom (*login*) i una paraula clau (*contrasenya*) o en ordinadors en què hi ha un repositori d'informació (programari, documentació...), l'usuari es connectarà com a anònim (*anonymous*) per a transferir (llegir) aquests arxius al seu ordinador. Això no és el mateix que els sistemes d'arxius de xarxa més recents, NFS (o protocols Netbios sobre TCP/IP, un "invent" totalment insegur sobre Windows i que és millor reemplaçar per una versió més antiga però més segura del mateix concepte anomenat *netbeui*), que permeten virtualitzar el sistema d'arxius d'una màquina perquè pugui ser accedit de manera interactiva sobre un altre ordinador.

TCP/IP

Utilització típica de TCP/IP *remote login:*

```
telnet localhost
Debian GNU/Linux 4.0
login:
```

b) Connexió (*login*) remota: el protocol de terminal de xarxa (*telnet*) permet a un usuari connectar-se a un ordinador remotament. L'ordinador local s'utilitza com a terminal de l'ordinador remot i tot és executat sobre aquest alhora que l'ordinador local roman invisible des del punt de vista de la sessió. Aquest servei actualment s'ha reemplaçat per l'SHH per raons de seguretat. En una connexió remota mitjançant *telnet*, els missatges circulen tal com estan (text pla); és a dir, si algú "observa" els missatges a la xarxa, equivaldrà a mirar la pantalla de l'usuari. SSH codifica la informació (que significa un cost afegit a la comunicació), que fa que els paquets a la xarxa siguin illegibles per a un node estrany.

c) Correu electrònic: aquest servei permet enviar missatges als usuaris d'altres ordinadors. Aquest mode de comunicació s'ha transformat en un element vital en la vida dels usuaris i permeten que els correus electrònics siguin enviats a un servidor central perquè després puguin ser recuperats per mitjà de programes específics (clients) o ser llegits per mitjà d'una connexió web.

L'avenç de la tecnologia i el baix cost dels ordinadors han permès que determinats serveis s'hi hagin especialitzat i s'ofereixen configurats sobre determinats ordinadors que treballen en un model client-servidor. Un servidor és un sistema que ofereix un servei específic per a la resta de la xarxa. Un client és un altre ordinador que utilitza aquest servei. Tots aquests serveis generalment són oferts dins de TCP/IP:

- **Sistemes d'arxius en xarxa** (NFS): permet a un sistema accedir als arxius sobre un sistema remot d'una manera més integrada que FTP. Els dispositius d'emmagatzematge (o part d'aquests) són exportats cap al sistema que vol accedir i els poden "veure" com si fossin dispositius locals. Aquest protocol permet a qui exporta posar les regles i les formes d'accés, cosa que (ben configurada) fa independent el lloc on es troba la informació físicament del lloc on es "veu" la informació.

- **Impressió remota:** permet accedir a impressores connectades a altres ordinadors.

- **Execució remota:** permet que un usuari executi un programa sobre un altre ordinador. Hi ha diferents maneres de fer aquesta execució: o bé per mitjà d'una instrucció (*rsh, ssh, rexec*) o per mitjà de sistemes amb RPC,[5] que permet a un programa en un ordinador local executar una funció d'un programa sobre un altre ordinador. Els mecanismes RPC han estat objecte d'estudi i hi ha diverses implementacions, però les més comunes són Xerox's Courier i Sun's RPC (aquesta última adoptada per la majoria dels Unix).

[(5)]De l'anglès *remote procedure call*.

- **Servidors de nom** (*name servers*): en grans instal·lacions hi ha un conjunt de dades que necessiten ser centralitzades per a millorar-ne la utilització, com per exemple, nom d'usuaris, paraules clau, adreces de xarxa, etc. Tot això facilita que un usuari disposi d'un compte per a totes les màquines d'una organització. Per exemple, Sun's Yellow Pages (NIS en les versions actuals de Sun) està dissenyat per a manejar tot aquest tipus de dades i està disponible per a la majoria de Unix. El DNS[6] és un altre servei de noms, però que guarda una relació entre el nom de la màquina i la identificació lògica d'aquesta màquina (adreça IP).

[(6)]De l'anglès *domain name system*.

- **Servidors de terminal** (*terminal servers*): connecta terminals a un servidor que executa *telnet* per a connectar-se a l'amfitrió. Aquest tipus

d'instal·lacions permet bàsicament reduir costos i millorar les connexions a l'amfitrió (en determinats casos).

- **Servidors de terminals gràfiques** (*network-oriented window systems*): permeten que un ordinador pugui visualitzar informació gràfica sobre una pantalla que està connectada a un altre ordinador. El més comú d'aquests sistemes és X-Window.

1.2. Què és TCP/IP?

TCP/IP són en realitat dos protocols de comunicació entre ordinadors independents un de l'altre.

D'una banda, TCP[7] defineix les regles de comunicació perquè un ordinador (*amfitrió*) pugui "parlar" amb un altre (si es pren com a referència el model de comunicacions OSI/ISO es descriu la capa 4, vegeu la taula següent). TCP és orientat a connexió, és a dir, equivalent a un telèfon, i la comunicació es tracta com un flux de dades (*stream*).

(7)De l'anglès *Transmission Control Protocol*.

D'altra banda, IP[8], defineix el protocol que permet identificar les xarxes i establir els camins entre els diferents ordinadors. És a dir, encamina les dades entre dos ordinadors per mitjà de les xarxes. Correspon a la capa 3 del model OSI/ISO i és un protocol sense connexió (vegeu la taula següent). [Com01, Rid00, Dra99]

(8)De l'anglès *Internet Protocol*.

Una alternativa al TCP la conforma el protocol UDP[9], el qual tracta les dades com un missatge (*datagrama*) i envia paquets. És un protocol sense connexió[10] i té l'avantatge que exerceix menys sobrecàrrega en la xarxa que les connexions de TCP, però la comunicació no és fiable (els paquets poden no arribar o arribar duplicats).

(9)De l'anglès *User Datagram Protocol*.

(10)L'ordinador de destinació no ha d'estar escoltant necessàriament quan un ordinador estableix comunicació amb ell.

Hi ha un altre protocol alternatiu anomenat ICMP[11]. ICMP s'utilitza per a missatges d'error o control. Per exemple, si algú intenta connectar-se a un equip (*amfitrió*), l'ordinador local pot rebre un missatge ICMP que indiqui *host unreachable*. ICMP també pot ser utilitzat per a extreure informació sobre una xarxa. ICMP és similar a UDP, ja que maneja missatges (datagrames), però és més simple que UDP, ja que no té identificació de ports[12] en l'encapçalament del missatge.

(11)De l'anglès *Internet control message protocol*.

(12)Són bústies on es dipositen els paquets de dades i des d'on les aplicacions servidores llegeixen els paquets esmentats.

El model de comunicacions OSI[13]/ISO[14] és un model teòric adoptat per moltes xarxes. Hi ha set capes de comunicació, i cada una té una interfície per a comunicar-se amb l'anterior i la posterior:

(13)De l'anglès *open systems interconnection reference model*.

Nivell	Nom	Utilització
7	Aplicació	SMTP[15], el servei pròpiament dit
6	Presentació	*telnet*, FTP implementa el protocol del servei
5	Sessió	Generalment no s'utilitza
4	Transport	TCP, UDP, transformació d'acord amb el protocol de comunicació
3	Xarxa	IP permet encaminar el paquet (*routing*)
2	Enllaç	Controladors (*drivers*), transformació d'acord amb el protocol físic
1	Físic	Ethernet, ADSL... envia del paquet físicament

(14)De l'anglès *International Standards Organization*.

(15)De l'anglès *simple mail transfer protocol*.

En resum, TCP/IP és una família de protocols (que inclouen IP, TCP, UDP) que proveeixen un conjunt de funcions a baix nivell utilitzades per la majoria de les aplicacions. [KD00, Dra99]

Alguns dels protocols que utilitzen els serveis esmentats han estat dissenyats per Berkeley, Sun o altres organitzacions, i oficialment no formen part de l'**Internet Protocol Suite** (IPS). Tanmateix, són implementats utilitzant TCP/IP i, per tant, són considerats com a part formal d'IPS. Una descripció dels protocols disponibles a Internet es pot consultar en l'RFC 1011 (vegeu les referències sobre RFC [IET]), que mostra tots els protocols disponibles. Hi ha actualment una nova versió del protocol **IPv6**, també anomenat *Ipng*[16], que reemplaça l'**IPv4.** Aquest protocol millora notablement l'anterior en temes com ara el nombre més elevat de nodes, el control de trànsit, la seguretat o les millores en aspectes d'encaminament.

(16)De l'anglès *IP next generation*.

1.3. Dispositius físics (maquinari) de xarxa

Des del punt de vista físic (capa 1 del model OSI), el maquinari més utilitzat per a LAN és conegut com a Ethernet (o FastEthernet o GigaEthernet). Els seus avantatges són el baix cost, velocitats acceptables (10, 100, o 1.000 megabits per segon) i facilitat en la instal·lació.

Hi ha tres modes de connexió en funció del tipus de cable d'interconnexió: gruixut (*thick*), fi (*thin*) i de parell trenat (*twisted parell*).

Les dues primeres són obsoletes (utilitzen cable coaxial), mentre que l'última es fa per mitjà de cables (parells) trenats i connectors similars als telefònics (es coneixen com a RJ45). La connexió de parell trenat és coneguda com a 10baseT o 100baseT (segons la velocitat) i utilitza repetidors anomenats *concentradors* (o *hubs*) com a punts d'interconnexió. La tecnologia Ethernet utilitza elements intermedis de comunicació (*concentradors, commutadors, encaminadors*)

(17)De l'anglès *Fiber Distributed Data Interface*.

per a configurar múltiples segments de xarxa i dividir el trànsit per a millorar les prestacions de transferència d'informació. Normalment, en les grans institucions aquestes LAN Ethernet estan interconnectades per mitjà de fibra òptica amb tecnologia FDDI[17], que és molt més cara i complexa d'instal·lar, però es poden obtenir velocitats de transmissió equivalents a Ethernet i no tenen la limitació de la distància (FDDI admet distàncies de fins a 200 km). El seu cost es justifica per a enllaços entre edificis o entre segments de xarxa molt congestionats. [Rid00, KD00]

Hi ha a més un altre tipus de maquinari menys comú, però no menys interessant, com és ATM[18]. Aquest maquinari permet muntar LAN amb una qualitat de servei elevada i és una bona opció quan s'han de muntar xarxes d'alta velocitat i baixa latència, com per exemple les que involucren distribució de vídeo en temps real.

(18)De l'anglès *asynchronous transfer mode*.

Hi ha un altre maquinari suportat per GNU/Linux per a la interconnexió d'ordinadors, entre els quals podem esmentar: Frame Relay o X.25, utilitzat en ordinadors que accedeixen o interconnecten WAN i per a servidors amb grans necessitats de transferències de dades; Packet Radio, interconnexió via ràdio amb protocols com AX.25, NetRom o Rose, o dispositius *dialing up*, que utilitzen línies en sèrie, lentes però molt barates, per mitjà d'un mòdem analògic o digital (XDSI, DSL, ADSL, etc.). Aquestes últimes són les que normalment s'utilitzen en pimes o en ús domèstic, i requereixen un altre protocol per a la transmissió de paquets, com ara SLIP o PPP. Per a virtualitzar la diversitat de maquinari sobre una xarxa, TCP/IP defineix una interfície abstracta mitjançant la qual es concentraran tots els paquets que seran enviats per un dispositiu físic (la qual cosa també significa una xarxa o un segment d'aquesta xarxa). Per això, per cada dispositiu de comunicació en la màquina estendrem una interfície corresponent en el nucli del sistema operatiu.

Ethernet

Ethernet en GNU/Linux es crida amb *ethx* (la *x* indica un número d'ordre començant per 0), la interfície a línies en sèrie (mòdem) es crida amb *pppx* (per a PPP) o *slx* (per a SLIP), i per a FDDI és *fddix*. Aquests noms són utilitzats per les ordres per a configurar els seus paràmetres i assignar-los el número d'identificació que posteriorment permetrà comunicarnos amb altres dispositius a la Xarxa.

En GNU/Linux pot implicar haver d'incloure els mòduls adequats per al dispositiu (NIC[19]) adequat (en el nucli o com a mòduls), i això significa compilar el nucli després d'haver escollit, per exemple, amb **make menuconfig**, el NIC adequat, indicant-lo com a intern o com a mòdul (en aquest últim cas s'haurà de compilar el mòdul adequat també).

(19)De l'anglès *network interface card*.

ifconfig -a

Per a veure les interfícies de xarxa disponibles cal aplicar l'ordre *ifconfig -a*. Aquesta ordre mostra totes les interfícies/paràmetres per defecte de cada una.

Els dispositius de xarxa es poden mirar en el directori /dev, que és on hi ha un arxiu (especial, ja sigui de bloc o de caràcters, segons la seva transferència), que representa cada dispositiu maquinari. [KD00, Dra99]

2. Conceptes en TCP/IP

Com s'ha observat, la comunicació significa una sèrie de conceptes que ampliarem a continuació [Mal96, Com01]:

- **Internet/intranets:** el terme *intranet* es refereix a l'aplicació de tecnologies d'Internet (xarxa de xarxes) dins d'una organització, bàsicament per distribuir i tenir disponible informació dins de la companyia. Per exemple, els serveis oferts per GNU/Linux com a serveis Internet i intranet inclouen correu electrònic, WWW, grups de notícies, etc.

- **Node:** es denomina *node* (*amfitrió*) una màquina que es connecta a la xarxa (en un sentit ampli, un node pot ser un ordinador, una impressora, una torre (*rack*) de CD, etc.), és a dir, un element actiu i diferenciable a la xarxa que reclama o deixa algun servei o comparteix informació.

- **Adreça de xarxa Ethernet** (*Ethernet address* o *MAC address*): un número de 48 bits (per exemple 00:88:40:73:AB:FF –en octal–, o 0000 0000 1000 1000 0100 0000 0111 0011 1010 1011 1111 1111 –en binari–) que es troba en el dispositiu físic (maquinari) del controlador (NIC) de xarxa Ethernet i és gravat pel fabricant (aquest número ha de ser únic al món, per la qual cosa cada fabricant de NIC té un rang preassignat).

- **Nom de l'amfitrió:** cada node ha de tenir a més un únic nom a la xarxa. Poden ser només noms o bé utilitzar un esquema de noms jeràrquic basat en dominis (*hierarchical domain naming scheme*). Els noms dels nodes han de ser únics, la qual cosa resulta fàcil en petites xarxes, i més dificultós en xarxes extenses, i impossible a Internet si no es fa algun control. Els noms han de ser d'un màxim de 32 caràcters, han d'usar a-zA-Z0-9.-, no han de contenir espais o # i han de començar per un caràcter alfabètic.

Nota

Nom de la màquina: **more /etc/hostname**

- **Adreça d'Internet** (*IP address*): està composta per quatre nombres en el rang 0-255 separats per punts (per exemple, 192.168.0.1), i s'utilitza universalment per a identificar els ordinadors sobre una xarxa o Internet. La translació de noms en adreces IP la fa un servidor DNS (*domain name system*), que transforma els noms de node (llegibles per humans) en adreces IP (aquest servei el fa una aplicació denominada *named*).

Nota

Adreça IP de la màquina: **more /etc/hosts**

- **Port** (*port*): identificador numèric de la bústia en un node que permet que un missatge (TCP, UDP) pugui ser llegit per una aplicació concreta dins d'aquest node (per exemple, dues màquines que es comuniquin per *telnet* ho faran pel port 23, però aquestes mateixes màquines poden tenir una comunicació FTP pel port 21). Es poden tenir diferents aplicacions comunicant-se entre dos nodes per mitjà de diferents ports simultàniament.

Nota

Ports preassignats en Unix: **more /etc/services.** Aquesta ordre mostra els ports predefinits per ordre, i segons si suporten TCP o UDP.

- **Node encaminador** (passarel·la o *gateway*): és un node que fa encaminaments (transferència de dades *routing*). Un encaminador o *router*, segons les seves característiques, podrà transferir informació entre dues xarxes de protocols similars o diferents, i pot ser, a més, selectiu.

Nota

Visualització de la configuració de l'encaminament: **netstat -r**.

- ***Domain name system*** (DNS): permet assegurar un únic nom i facilitar l'administració de les bases de dades que fan la translació entre nom i adreça d'Internet, i s'estructuren en forma d'arbre. Per a això, s'especifiquen dominis separats per punts, el més alt (de dreta a esquerra) dels quals descriu una categoria, institució o país (com, comercial; edu, educació; gov, governamental; mil, militar (govern); org, sense finalitat de lucre; dues lletres per a un país, o en casos especials tres lletres, com cat, llengua i cultura catalana...). El segon nivell representa l'organització, el tercer i els restants els departaments, seccions o divisions dins d'una organització (per exemple, www.uoc.edu o nteum@pirulo.remix.es). Els dos primers noms (de dreta a esquerra, uoc.edu en el primer cas, remix.es en el segon, han de ser assignats (aprovats) per l'SRI-NIC (òrgan mundial gestor d'Internet) i els restants poden ser configurats o assignats per la institució.

Nota

El nostre domini i servidor de DNS: **more /etc/default domain; more /etc/resolv.conf**.

- **DHCP, bootp:** DHCP i *bootp* són protocols que permeten a un node client obtenir informació de la xarxa (com l'adreça IP del node). Moltes organitzacions amb gran quantitat de màquines utilitzen aquest mecanisme per a facilitar l'administració a grans xarxes o on hi ha una gran quantitat d'usuaris mòbils.

- **ARP, RARP:** en algunes xarxes (com per exemple IEEE 802 LAN, que és l'estàndard per a Ethernet), les adreces IP són descobertes automàticament per mitjà de dos protocols membres d'IPS: ARP[20] i RARP[21]. ARP utilitza missatges de difusió (*broadcast messages*) per a determinar l'adreça Ethernet (especificació MAC de la capa 3 del model OSI) corresponent a una adreça de xarxa particular (IP). RARP utilitza missatges de difusió (missatge que arriba a tots els nodes) per a determinar l'adreça de xarxa associada amb una adreça de maquinari en particular. RARP és especialment important en màquines sense disc, en les quals l'adreça de xarxa generalment no es coneix en el moment de l'inici (*boot*).

[20] De l'anglès *address resolution protocol.*

[21] De l'anglès *reverse address resolution protocol.*

Nota

Taules d'ARP: **ARP a *NomNode.***

- **Biblioteca de sòcols:** a Unix tota la implementació de TCP/IP forma part del nucli del sistema operatiu (o bé a dins o com un mòdul que es carrega

en el moment de l'inici, com el cas de GNU/Linux amb els controladors de dispositius).

La manera d'utilitzar-les per part d'un programador és per mitjà de l'API[22] que implementa aquest sistema operatiu. Per a TCP/IP, l'API més comuna és la Berkeley Socket Library (Windows utilitza una biblioteca equivalent que es diu Winsocks). Aquesta biblioteca permet crear un punt de comunicació (sòcol), associar-lo a una adreça d'un node remot/port (vincle) i oferir el servei de comunicació (per mitjà de *connect, listen, accept, send, sendto, recv, recvfrom*, per exemple). La biblioteca proveeix, a més de la forma més general de comunicació (família AF INET), comunicacions més optimitzades per a casos en els quals els processos es comuniquen a la màquina mateixa (família AF Unix). En GNU/Linux, la biblioteca de sòcols és part de la biblioteca estàndard de C, Libc (Libc6 en les versions actuals), i suporta AF_INET, AF_UNIX, AF_IPX (per a protocols de xarxes Novell), AF_X25 (per al protocol X.25), AF_ATMPVC-AF_ATMSVC (per al protocol ATM) i AF_AX25, F_NETROM, AF_ROSE (per a l'Amateur Radio Protocol).

(22)De l'anglès *application programming interface.*

3. Com s'assigna una adreça d'Internet?

Aquesta adreça és assignada pel SRI-NIC i té dos camps. L'esquerre representa la identificació de la xarxa i el dret la identificació del node. Considerant el que hem dit anteriorment (4 nombres entre 0-255, o sigui, 32 bits o quatre bytes), cada byte representa o bé la xarxa o bé el node. La part de xarxa és assignada pel SRI-NIC i la part del node és assignada per la institució o el proveïdor).

Hi ha algunes restriccions: **0** (per exemple, 0.0.0.0) en el camp de xarxa és reservat per a l'encaminament per defecte i **127** (per exemple, 127.0.0.1) és reservat per a l'autoreferència (*local loopback* o *local host*), **0** en la part de node es refereix a aquesta xarxa (per exemple, 192.168.0.0) i 255 és reservat per a paquets de tramesa a totes les màquines (difusió) (per exemple, 198.162.255.255). En les diferents assignacions es poden tenir diferents tipus de xarxes o adreces:

- **Classe A** (*xarxa.amfitrió.amfitrió.amfitrió*): 1.0.0.1 a 126.254.254.254 (126 xarxes, 16 milions de nodes); defineixen les grans xarxes. El patró binari és: **0** + 7 bits xarxa + 24 bits de nodes.

- **Classe B** (*xarxa.xarxa.amfitrió.amfitrió*): 128.1.0.1 a 191.255.254.254 (16K xarxes, 65K nodes); generalment s'utilitza el primer byte de node per a identificar subxarxes dins d'una institució). El patró binari és **10** + 14 bits de xarxa + 16 bits de nodes.

- **Classe C** (*xarxa.xarxa.xarxa.amfitrió*): 192.1.1.1 a 223.255.255.254 (2 milions de bits de xarxes, 254 de nodes). El patró binari és **110** + 21 bits xarxa + 8 bits de nodes.

- **Classe D i E** (*xarxa.xarxa.xarxa.amfitrió*): 224.1.1.1 a 255.255.255.254, reservat per a multidestinació (des d'un node a un conjunt de nodes que formen part d'un grup) i propòsits experimentals.

Xarxes privades

Màquines que es connecten entre elles sense tenir connexió amb l'exterior.

Alguns rangs d'adreces han estat reservats perquè no corresponguin a xarxes públiques, sinó a xarxes privades, i els missatges no seran encaminats per mitjà d'Internet, cosa que es coneix com a *intranets*. Aquestes són per a la **classe A** des de 10.0.0.0 fins a 10.255.255.255, **classe B des de** 172.16.0.0 fins a 172.31.0.0 i **classe C** des de 192.168.0.0 fins a 192.168.255.0.

L'adreça de difusió és especial, ja que cada node en una xarxa escolta tots els missatges (a més de la seva adreça pròpia). Aquesta adreça permet que datagrames, generalment informació d'encaminament (o *routing*) i missatges d'avís,

puguin ser enviats a una xarxa i tots els nodes del mateix segment de xarxa els puguin llegir. Per exemple, quan ARP busca l'adreça Ethernet corresponent a una IP, utilitza un missatge de difusió (o *broadcast*), el qual és enviat a totes les màquines de la xarxa simultàniament. Cada node a la xarxa llegeix aquest missatge i compara la IP que es busca amb la pròpia i retorna un missatge al node que va fer la pregunta si hi ha coincidència.

Dos conceptes complementaris als descrits anteriorment són el de **subxarxes** i **encaminament** entre elles. *Subxarxes* significa subdividir la part del node en petites xarxes dins de la mateixa xarxa, per exemple, per a millorar el trànsit. Una subxarxa pren la responsabilitat d'enviar el trànsit a certs rangs d'adreces IP i estén el mateix concepte de xarxes A, B, C, però només aplicant aquest readreçament en la part de node de la IP. El nombre de bits que són interpretats com a identificador de la subxarxa és donat per una màscara de xarxa (o *netmask*) que és un nombre de 32 bits (igual que la IP).

Per a obtenir l'identificador de la subxarxa, s'haurà de fer una operació lògica **I** (AND) entre la màscara i la IP, la qual cosa donarà la IP de la subxarxa. Per exemple, tenim una institució que té una xarxa classe B amb número 172.17.0.0; la seva màscara de xarxa és, per tant, 255.255.0.0. Internament, aquesta xarxa està formada per petites xarxes (una planta de l'edifici, per exemple). Així, el rang d'adreces és reassignat en 20 subxarxes (plantes, per a nosaltres): des de 172.17.1.0 fins a 172.17.20.0. El punt que connecta totes aquestes plantes (xarxa troncal) té la seva pròpia adreça, com per exemple 172.17.1.0.

Aquestes subxarxes comparteixen la mateixa IP de xarxa, mentre que la tercera és utilitzada per a identificar cada una de les subxarxes que hi ha a dins (per això s'utilitzarà una màscara de xarxa 255.255.255.0).

El segon concepte, encaminament, representa la manera com els missatges són enviats per mitjà de les subxarxes. Per exemple, tenim tres departaments amb subxarxes Ethernet:

- Compres (subxarxa 172.17.2.0).
- Clients (subxarxa 172.17.4.0).
- Recursos humans, RH, (subxarxa 172.17.6.0).
- Xarxa troncal amb FFDI (subxarxa 172.17.1.0).

Per a encaminar els missatges entre els ordinadors de les tres xarxes es necessitaran tres portes d'intercanvi (passarel·les), que tindran cada una dues interfícies de xarxa per a canviar entre Ethernet i FFDI. Seran les següents:

- CompresGW IP:172.17.2.1 i 172.17.1.1,
- ClientsGW IP:172.17.4.1 i 172.17.1.2
- RHGW IP:172.17.6.1 i 172.17.1.3, és a dir, una IP cap al costat de la subxarxa i una altra cap a la xarxa troncal.

Quan s'envien missatges entre màquines de compres, no és necessari sortir a la passarel·la, ja que el protocol TCP/IP trobarà la màquina directament. El problema és quan la màquina Compres0 vol enviar un missatge a RH3. El missatge ha de circular per les dues passarel·les respectives. Quan Compres0 "veu" que RH3 és en una altra xarxa, envia el paquet per mitjà de la passarel·la CompresGW, que al seu torn l'enviarà a RHGW, que al seu torn l'enviarà a RH3. L'avantatge de les subxarxes és clar, ja que el trànsit entre totes les màquines de compres, per exemple, no afectarà les màquines de clients o de recursos humans (si bé significa un plantejament més complex i car a l'hora de dissenyar i construir la xarxa).

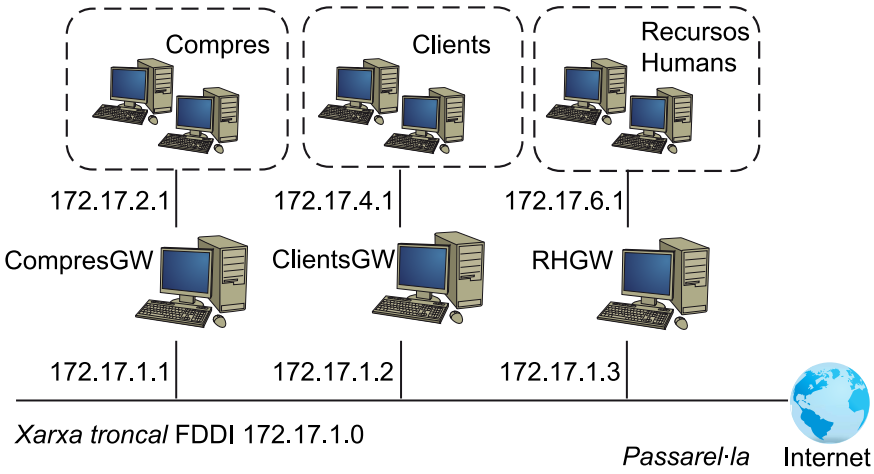

Figura 1. Configuració de segments i passarel·les en una Intranet

IP utilitza una taula per a fer l'encaminament dels paquets entre les diferents xarxes i en la qual hi ha un encaminament per defecte associat a la xarxa 0.0.0.0. Totes les adreces que coincideixen amb aquesta, ja que cap dels 32 bits no són necessaris, són enviades per la passarel·la per defecte (*default gateway*) cap a la xarxa indicada. Sobre CompresGW, per exemple, la taula podria ser:

Adreça	Màscara	Passarel·la	Interfície
172.17.1.0	255.255.255.0	-	fddi0
172.17.4.0	255.255.255.0	172.17.1.2	fddi0
172.17.6.0	255.255.255.0	172.17.1.3	fddi0
0.0.0.0	0.0.0.0	172.17.2.1	fddi0
172.17.2.0	255.255.255.0	-	eth0

El "-" significa que la màquina està directament connectada i no necessita encaminament. El procediment per a identificar si es fa l'encaminament o no es du a terme és per mitjà d'una operació molt simple amb dos AND lògics (subxarxa AND màscara i origen AND màscara) i una comparació entre els dos

resultats. Si són iguals no hi ha encaminament, sinó que s'ha d'enviar la màquina definida com a passarel·la a cada màquina perquè faci l'encaminament del missatge.

Per exemple, un missatge de la 172.17.2.4 cap a la 172.17.2.6 significarà:

```
172.17.2.4 AND 255.255.255.0 = 172.17.2.0
172.17.2.6 AND 255.255.255.0 = 172.17.2.0
```

Com els resultats són iguals, no hi haurà encaminament. En canvi, si fem el mateix amb 172.17.2.4 cap a 172.17.6.6 podem veure que hi haurà un encaminament per mitjà del 172.17.2.1 amb un canvi d'interfície (*eth0* a *ffdi0)* a la 172.17.1.1 i d'aquesta cap a la 172.17.1.2 amb un altre canvi d'interfície (*fddi0* a *eth0*) i després cap a la 172.17.6.6. L'encaminament, per defecte, s'utilitzarà quan cap regla no satisfaci la coincidència. En cas que dues regles coincideixin, s'utilitzarà la que ho faci de manera més precisa, és a dir, la que menys zeros tingui. Per a construir les taules d'encaminament, es pot utilitzar l'ordre **route** durant l'arrencada de la màquina, però si és necessari utilitzar regles més complexes (o encaminament automàtic), es pot utilitzar el RIP[23] o, entre sistemes autònoms, l'EGP[24] o també el BGP[25]. Aquests protocols s'implementen en l'ordre **gated.**

(23)De l'anglès *routing information protocol.*

(24)De l'anglès *external gateway protocol.*

(25)De l'anglès *border gateway protocol.*

Per a instal·lar una màquina sobre una xarxa existent, és necessari, per tant, disposar de la informació següent obtinguda del proveïdor de xarxa o de l'administrador: adreça IP del node, adreça de la xarxa IP, adreça de difusió, adreça de màscara de xarxa, adreça d'encaminador, adreça del DNS.

Si es construeix una xarxa que mai no tindrà connexió a Internet, es poden escollir les adreces que es prefereixin, però és recomanable mantenir un ordre adequat en funció de la mida de xarxa que es vulgui tenir, i per a evitar problemes d'administració dins de la xarxa. A continuació, es veurà com es defineix la xarxa i el node per a una xarxa privada (cal ser acurat, ja que si es té la màquina connectada a la Xarxa, es podria perjudicar un altre usuari que tingués assignada aquesta adreça).

4. Com s'ha de configurar la xarxa?

4.1. Configuració de la interfície (NIC)

Una vegada carregat el nucli de GNU/Linux, aquest executa l'ordre **init**, que, al seu torn, llegeix l'arxiu de configuració /etc/inittab i comença el procés d'inicialització. Generalment, inittab té seqüències com ara *si::sysinit:/etc/init.d/boot*, que representa el nom de l'arxiu d'instruccions (script) que controla les seqüències d'inicialització. Generalment aquest script crida altres scripts, entre els quals hi ha la inicialització de la xarxa.

Exemple

En Debian s'executa etc/init.d/network per a la configuració de la interfície de xarxa i en funció del nivell d'arrencada; per exemple, en el 2 s'executaran tots els fitxers S* del directori /etc/rc2.d (que són enllaços al directori /etc/initd), i en el nivell d'apagat, tots els K* del mateix directori. D'aquesta manera, l'script està només una vegada (/etc/init.d) i d'acord amb els serveis que volem en aquest estat es crea un enllaç en el directori corresponent a la configuració del node estat.

Els dispositius de xarxa es creen automàticament quan s'inicialitza el maquinari corresponent. Per exemple, el controlador d'Ethernet crea les interfícies *eth[0..n]* seqüencialment quan es localitza el maquinari corresponent.

Nota

Consulteu **man ifconfig** per a les diferents opcions de l'ordre.

A partir d'aquest moment, es pot configurar la interfície de xarxa, la qual cosa implica dos passos: assignar l'adreça de xarxa al dispositiu i inicialitzar els paràmetres de la xarxa al sistema. L'ordre utilitzada per a això és **ifconfig** (*interface configure*). Un exemple serà:

```
ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up
```

Això indica configurar el dispositiu *eth0* amb adreça IP 192.168.110.23 i la màscara de xarxa 255.255.255.0. **up** indica que la interfície passarà a l'estat actiu (per a desactivar-la s'hauria d'executar **ifconfig eth0 down**). L'ordre assumeix que si alguns valors no s'indiquen són presos per defecte. En aquest cas, el nucli configurarà aquesta màquina com a tipus C i configurarà la xarxa amb 192.168.110.23 i l'adreça de difusió amb 192.168.110.255. Per exemple:

```
ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up
```

Hi ha ordres com **ifup** i **ifdown**, que permeten configurar i desactivar la xarxa de manera més simple utilitzant l'arxiu /etc/network/interfaces per a obtenir tots els paràmetres necessaris (consulteu *man interfaces* per a la sintaxi).

En Debian, a fi de facilitar la configuració de la xarxa, hi ha una altra manera de configurar la xarxa (considerada d'alt nivell), que utilitza les ordres esmentades anteriorment *ifup, ifdown* i l'arxiu /etc/network/interfaces. Si es decideix utilitzar aquestes ordres, no s'hauria de configurar la xarxa a baix nivell, ja que aquestes instruccions són suficients per a configurar o desactivar la xarxa.

Per a modificar els paràmetres[26] de xarxa de la interfície *eth0*, es pot fer:

```
ifdown eth0
per a tots els serveis de xarxa sobre eth0
vi /etc/network/interfaces
editeu i modifiqueu els que necessiteu
ifup eth0
posa en marxa els serveis de xarxa sobre eth0
```

(26)Consulteu *man interfaces* a la secció 5 del manual per a més informació del format.

Suposem que volem configurar sobre Debian una interfície *eth0* que té una adreça IP fixa 192.168.0.123 i amb 192.168.0.1 com a porta d'enllaç (passarel·la). Cal editar `/etc/network/interfaces` de manera que inclogui una secció com la següent:

```
iface eth0 inet static
      address 192.168.0.123
      netmask 255.255.255.0
      gateway 192.168.0.1
```

Si tenim instal·lat el paquet *resolvconf* podem afegir línies per a especificar la informació relativa al DNS. Per exemple:

```
iface eth0 inet static
      address 192.168.0.123
      netmask 255.255.255.0
      gateway 192.168.0.1
      dns-search remix.org
      dns-nameservers 195.238.2.21 195.238.2.22
```

Després que s'activi la interfície, els arguments de les opcions següents *dns-search* i *dns-nameservers* queden disponibles per a la inclusió a `resolv.conf`. L'argument *remix.org* de l'opció *dns-search* correspon a l'argument de l'opció *search* a `resolv.conf` i els arguments `195.238.2.21` i `195.238.2.22` de l'opció *dns-nameservers* corresponen als arguments de les opcions *nameserver* a `resolv.conf`. També es pot configurar la xarxa a baix nivell per mitjà de l'ordre **ip** (que és equivalent a *ifconfig* i *route*). Si bé

Nota: `resolv.conf`

Podeu consultar el manual per a veure *man resolv.conf*.

aquesta ordre és molt més versàtil i potent (permet establir túnels, encaminaments alternatius, etc.), és més complexa i es recomana utilitzar els procediments anteriors per a configuracions bàsiques de la xarxa.

4.1.1. Configuració de xarxa en distribucions de tipus Fedora

Red Hat i Fedora utilitzen estructures de fitxers diferents per a la configuració de la xarxa: /etc/sysconfig/network. Per exemple, per a la configuració estàtica de la xarxa:

```
NETWORKING=yes
HOSTNAME=my-hostname
      Nom de l'ordinador definit per l'ordre hostname
FORWARD_IPV4=true
      True per a NAT, tallafocs, passarel·les i encaminadors.
      False per a qualsevol altre cas
GATEWAY="XXX.XXX.XXX.YYY"
      Adreça IP de la porta de sortida a Internet.
```

Per a la configuració amb DHCP s'ha de treure la línia de *gateway*, ja que serà assignada pel servidor. I en cas d'incorporar NIS cal agregar una línia amb el servidor de domini: *NISDOMAIN=NISProject1*.

Per a configurar la interfície *eth0* en l'arxiu /etc/sysconfig/network-scripts/ifcfg-eth0 (reemplaçar les X amb els valors adequats):

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=XXX.XXX.XXX.255
IPADDR=XXX.XXX.XXX.XXX
NETMASK=255.255.255.0
NETWORK=XXX.XXX.XXX.0
ONBOOT=yes Activarà la xarxa en l'arrencada
```

També a partir d'FC3 es poden agregar:

```
TYPE=Ethernet
HWADDR=XX:XX:XX:XX:XX:XX
GATEWAY=XXX.XXX.XXX.XXX
IPV6INIT=no
USERCTL=no
PEERDNS=yes
```

O, si no, per a la configuració del DHCP:

```
DEVICE=eth0
ONBOOT=yes
```

```
BOOTPROTO=dhcp
```

Per a deshabilitar DCHP, cal canviar *BOOTPROTO=dhcp* per *BOOTPROTO=none*. Qualsevol canvi en aquests fitxers implicarà reiniciar els serveis amb **service network restart** (o si no **/etc/init.d/network restart**).

Per a canviar el nom de l'amfitrió s'han de seguir aquests tres passos:

1) L'ordre **hostname nom-nou.**

2) Canviar la configuració de la xarxa a `/etc/sysconfig/network` editant: *HOSTNAME=nom-nou*.

3) Restaurant els serveis (o reiniciant):

- **service network restart** (o */etc/init.d/network restart*).
- Reiniciant l'escriptori passant a mode consola amb **init 3** i canviant a mode GUI amb **init 5**.

Verificar si el nom tampoc no està donat d'alta a `/etc/hosts`. El nom de l'ordinador es pot canviar en temps d'execució amb sysctl *-w kernel.hostname="nom-nou"*.

4.1.2. Configuració d'una xarxa Wi-Fi (sense fil)

Per a la configuració d'interfícies Wi-Fi s'utilitzen bàsicament el paquet **wireless-tools** (a més d'*ifconfig* o *ip*). Aquest paquet utilitza l'ordre *iwconfig* per a configurar una interfície sense fil, però també es pot fer per mitjà de l'arxiu */etc/network/interfaces*.

Exemple: Configurar una Wi-Fi en Debian (similar en FC)

Suposem que volem configurar una targeta de xarxa sense fil Intel Pro/Wireless 2200BG (molt comuna en una gran quantitat de portàtils, com per exemple Dell, HP...). Normalment, el programari que controla les targetes es divideix en dues parts: el mòdul programari, que es carregarà en el nucli per mitjà de l'ordre *modprobe*, i el *microprogramari*, que és el codi que es carregarà a la targeta i que ens dóna el fabricant (consulteu la pàgina d'Intel per a aquest model). Com estem parlant de mòduls, és interessant utilitzar el paquet de Debian *module-assistant*, que ens permet crear i instal·lar fàcilment un mòdul (una altra opció seria instal·lar les fonts i crear el mòdul corresponent). El programari (el trobem a la pàgina del fabricant, i el denomina *ipw2200*) el compilarem i instal·larem amb l'ordre **m-a** del paquet *module-assistant*.

```
aptget install module-assistant instal·lació del paquet
m-a -t update
```

```
m-a -t -f get ipw2200
m-a -t -build ipw2200
m-a -t install ipw2200
```

Des de l'adreça indicada pel fabricant (en la seva documentació), es descarrega la versió del microprogramari compatible amb la versió del controlador, en el nostre cas per al controlador versió 1.8 el microprogramari és el 2.0.4, obtingut des de la pàgina http://ipw2200.sourceforge.net/firmware.php

I a continuació es descomprimeix i instal·la el microprogramari:

```
tar xzvf ipw2200fw2.4.tgz C /tmp/fwr/
cp /tmp/fwr/*.fw /usr/lib/hotplug/firmware/
```

Amb això es copiaran tres paquets *(ipw2200-bss.fw, ipw2200*-ibss.fw i *ipw2200-sniffer.fw*). Després es carrega el mòdul amb **modprobe ipw2200**, es reinicia el sistema (*reboot*) i després, des de la consola, podem fer **dmesg | grep ipw;** aquesta ordre ens mostrarà algunes línies similars a les que es mostren a continuació, que indicaran que el mòdul està carregat (es pot verificar amb *lsmod*):

```
ipw2200: Intel(R) PRO/Wireless 2200/2915 Network Driver, git1.0.8
ipw2200: Detected Intel PRO/Wireless 2200BG Network Connection
...
```

Després es descarrega el paquet *wirelesstools*, que conté *iwconfig* (per exemple, amb **aptget install wirelesstools**) i executem **iwconfig;** sortirà una cosa semblant al següent:

```
eth1 IEEE 802.11b ESSID:"Nombre-de-la-Wifi"
Mode:Managed Frequency:2.437 GHz
Access Point:00:0E:38:84:C8:72
Bit Rate=11 Mb/s TxPower=20 dBm
Security mode:open
...
```

A continuació, cal configurar l'arxiu de xarxes; per exemple, **gedit /etc/network/interfaces** i afegir la interfície *wifi eth1*, per exemple:

```
iface eth1 inet dhcp
      pre-up iwconfig eth1 essid "Nom de la Wi-Fi"
      pre-up iwconfig eth1 key open XXXXXXXXXX
```

La línia *preup* executa l'ordre *iwconfig* abans d'activar la interfície. Aquesta configuració és per a quan es vol utilitzar un servei en mode DHCP (assignació automàtica d'IP); s'ha d'utilitzar en comptes de *dhcp* la paraula *static* i a més posar les línies següents, per exemple (com en una targeta de cable):

```
address 192.168.1.132
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.1.1
```

Un mètode alternatiu per a configurar la interfície és:

```
iface eth1 inet dhcp
      wireless-essid "Nom de la Wi-Fi"
      wireless-key 123456789e
```

A continuació es pot posar en marxa la xarxa amb *ifup eth1* i ens donarà informació sobre la connexió i ens indicarà l'estat i la qualitat de recepció. Per a buscar (*scan*) les xarxes Wi-Fi disponibles (punts d'accés) podem utilitzar **iwlist scan**, que ens mostrarà informació de les xarxes disponibles, i si ens volem connectar a una de diferent, es pot utilitzar l'ordre **iwconfig** per a canviar de xarxa o punt d'accés (*access point*).

4.2. Configuració del sistema de resolució de noms

El pas següent és configurar el sistema de resolució de noms (*name resolver*), que converteix noms com ara *pirulo.remix.com* a 192.168.110.23. L'arxiu /etc/resolv.conf és l'utilitzat per a tal finalitat. El format és molt simple (una línia de text per sentència). Hi ha tres paraules clau per a tal finalitat: *domain* (domini local), *search* (llista de dominis alternatius) i *name server* (l'adreça IP del servidor de noms de domini).

4.2.1. Exemple de l'arxiu /etc/resolv.conf

```
domain remix.com
search remix.com piru.com
name server 192.168.110.1
name server 192.168.110.65
```

Aquesta llista de servidors de nom sovint depèn de l'entorn de xarxa, que pot canviar depenent d'on sigui o es connecti la màquina. Els programes de connexió a línies telefòniques (*pppd*) o obtenció d'adreces IP automàticament (*dhclient*) són capaços de modificar `resolv.conf` per a inserir o eliminar servidors, però aquesta característica no sempre funciona adequadament i de vegades pot entrar en conflicte i generar configuracions errònies. El paquet **`resolvconf`** (encara en Unstable) soluciona de manera adequada el problema i permet una configuració simple dels servidors de nom de manera dinàmica. *`resolvconf`* està dissenyat per a funcionar sense que sigui necessària cap configuració manual; no obstant això, el paquet és bastant nou i pot requerir alguna intervenció per a aconseguir que funcioni adequadament.

Nota

Per a més informació sobre el paquet *`resolvconf`*, podeu consultar el web explicatiu:
http://packages.debian.org/unstable/net/resolvconf

Un arxiu important és /etc/host.conf, que permet configurar el comportament del sistema de resolució de noms. La seva importància és que indica on es resol primer l'adreça o el nom d'un node. Aquesta consulta es pot fer al servidor DNS o en taules locals dins de la màquina actual (/etc/hosts).

4.2.2. Exemple de l'arxiu /etc/host.conf

```
order hosts,bind
multi on
```

Aquesta configuració indica que primer es verifica /etc/hosts abans de sol·licitar una petició al DNS i també indica (2a. línia) que retorni totes les adreces vàlides que hi hagi a /etc/hosts. Per això, l'arxiu /etc/hosts és on es col·loquen les adreces locals i també serveix per a accedir a nodes sense haver de consultar el DNS.

La consulta és molt més ràpida, però té el desavantatge que si el node canvia, l'adreça serà incorrecta. En un sistema correctament configurat, només hauran d'aparèixer el node local i una entrada per a la interfície *loopback*.

4.2.3. Exemple de l'arxiu /etc/hosts

```
127.0.0.1 localhost loopback
192.168.1.2 pirulo.remix.com pirulo
```

Per al nom d'una màquina es poden utilitzar àlies, fet que significa que la màquina es pot anomenar de diferents maneres per a la mateixa adreça IP. Amb referència a la interfície *loopback*, aquest és un tipus especial d'interfície que permet fer connexions amb si mateixa (per exemple, per a verificar que el subsistema de xarxa funciona sense accedir a la xarxa). Per defecte, l'adreça IP 127.0.0.1 ha estat assignada específicament al *loopback* (una ordre *telnet 127.0.0.1* connectarà amb la màquina mateixa). La configuració és molt fàcil (la fan generalment els scripts d'inicialització de xarxa).

4.2.4. Exemple del *loopback*

```
ifconfig lo 127.0.0.1
route add host 127.0.0.1 lo
```

En la versió 2 de la biblioteca GNU hi ha una canvi important respecte a la funcionalitat de l'arxiu host.conf. Aquesta millora inclou la centralització d'informació de diferents serveis per a la resolució de noms, la qual cosa presenta grans avantatges per a l'administrador de xarxa. Tota la informació de consulta de noms i serveis ha estat centralitzada en l'arxiu /etc/nsswitch.conf, el qual permet a l'administrador configurar l'ordre i les bases de dades de manera molt simple. En aquest arxiu cada servei apareix en una línia amb un conjunt d'opcions, en què, per exemple, hi ha la resolució de noms de no-

Nota

Exemple de nsswitch.conf:
hosts: dns files
...
networks: files

de. S'hi indica que l'ordre de consulta de les bases de dades per a obtenir la IP del node o el seu nom serà primer el servei de DNS, que utilitzarà l'arxiu /etc/resolv.conf per a determinar la IP del node DNS, i en cas que no el pugui obtenir, utilitzarà el de les bases de dades locals (/etc/hosts). Altres opcions per a això podrien ser *nis, nisplus*, que són altres serveis d'informació que es descriuran en unitats posteriors. També es pot controlar per mitjà d'accions (entre []) el comportament de cada consulta, com per exemple:

```
hosts: xfn nisplus dns [NOTFOUND = return] files
```

Això indica que quan es faci la consulta al DNS, si no hi ha un registre per a aquesta consulta, retorni un zero al programa que la va fer. Es pot utilitzar el "!" per a negar l'acció, com per exemple:

```
hosts dns [!UNAVAIL = return] files
```

4.3. Configuració de l'encaminament

Nota

Consulta de taules d'encaminament:
route -n
netstat -r

Un altre aspecte que cal configurar és l'encaminament. Si bé hi ha el tòpic sobre la seva dificultat, generalment es necessiten uns requisits d'encaminament molt simples. En un node amb múltiples connexions, l'encaminament consisteix a decidir on cal enviar i què es rep. Un node simple (una sola connexió de xarxa) també necessita encaminament, ja que tots els nodes disposen d'un *loopback* i una connexió de xarxa (per exemple, Ethernet, PPP, SLIP...). Com es va explicar anteriorment, hi ha una taula anomenada *routing table*, que conté files amb diversos camps, però tres són summament importants: adreça de destinació, interfície per on sortirà el missatge i adreça IP, que efectuarà el pas següent a la Xarxa (passarel·la).

L'ordre *route* permet modificar aquesta taula per a fer les tasques d'encaminament adequades. Quan arriba un missatge, es mira la seva adreça de destinació, es compara amb les entrades a la taula i s'envia per la interfície en què l'adreça coincideix millor amb la destinació del paquet. Si una passarel·la és especificada, s'envia a la interfície adequada.

Considerem, per exemple, que el nostre node és en una xarxa de classe C amb adreça 192.168.110.0 i té una adreça 192.168.110.23; i l'encaminador amb connexió a Internet és 192.168.110.3. La configuració serà:

- Primer la interfície:

```
ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up
```

- Més endavant, cal indicar que tots els paquets amb adreces 192.168.0. * han de ser enviats al dispositiu de xarxa:

```
route add -net 192.1 ethernetmask 255.255.255.0 eth0
```

El *-net* indica que és una ruta de xarxa, però també es pot utilitzar *-host* 192.168.110.3. Aquesta configuració permetrà connectar-se a tots els nodes dins del segment de xarxa (192.1), però què passarà si ens volem connectar a un altre node fora d'aquest segment? Seria molt difícil tenir totes les entrades adequades per a totes les màquines a les quals ens volem connectar. Per a simplificar aquesta tasca, hi ha el *default route*, que s'utilitza quan l'adreça de destinació no coincideix a la taula amb cap de les entrades. Una possibilitat de configuració seria:

```
route add default gw 192.168.110.3 eth0
```

Nota

El *gw* és la IP o el nom d'una passarel·la o node encaminador.

Una forma alternativa de fer-ho és:

```
ifconfig eth0 inet down deshabilito la interfície
ifconfig
lo Link encap:Local Loopback
... (no mostrarà cap entrada per a eth0)
route
... (no mostrarà cap entrada en la taula de rutes)
```

Després s'habilita la interfície amb una nova IP i una nova ruta:

```
ifconfig eth0 inet up 192.168.0.111 \
      netmask 255.255.0.0 broadcast 192.168.255.255
route add -net 10.0.0.0 netmask 255.0.0.0 \
      gw 192.168.0.1 dev eth0
```

La barra (\) indica que l'ordre continua en la línia següent. El resultat:

```
ifconfig
      eth0 Link encap:Ethernet HWaddr 08:00:46:7A:02:B0
      inet addr:192.168.0.111 Bcast: 192.168.255.255 Mask:255.255.0.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      ...
      lo Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      ...
route
Kernel IP routing table
```

```
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.0.0 * 255.255.0.0 U 0 0 0 eth0
10.0.0.0 192.168.0.1 255.0.0.0 UG 0 0 0 eth0
```

Per a més informació vegeu els manuals de les ordres *ifconfig*(8) i *route*(8).

4.4. Configuració d'*inetd*

El pas següent en la configuració de xarxa és la configuració dels servidors i serveis que permetran a un altre usuari accedir a la màquina local o als seus serveis. Els programes servidors utilitzaran els ports per a escoltar les peticions dels clients, els quals es dirigiran a aquest servei com a *IP:port*. Els servidors poden funcionar de dues maneres diferents: *standalone* (en aquesta mode el servei escolta el port assignat i sempre està actiu) o per mitjà d'*inetd*.

L'*inetd* és un servidor que controla i gestiona les connexions de xarxa dels serveis especificats en l'arxiu /etc/inetd.conf, el qual, davant d'una petició de servei, posa en marxa el servidor adequat i li transfereix la comunicació.

Dos arxius importants necessiten ser configurats de la manera següent: /etc/services i /etc/inetd.conf. En el primer s'associen els serveis, els ports i el protocol, i en el segon, quins programes servidors respondran davant d'una petició a un port determinat. El format de /etc/services és *name port/protocol alias*, en què el primer camp és el nom del servei, el segon, el port on atén aquest servei i el protocol que utilitza, i el següent, un àlies del nom. Per defecte hi ha una sèrie de serveis que ja estan preconfigurats. A continuació es mostra un exemple de l'arxiu /etc/services (# indica que el que hi ha a continuació és un comentari):

```
tcpmux    1/tcp      # TCP port service multiplexer
echo      7/tcp
echo      7/udp
discard   9/tcp      sink null
discard   9/udp      sink null
systat    11/tcp     users
...
ftp       21/tcp
ssh       22/tcp     # SSH Remote Login Protocol
ssh       22/udp     # SSH Remote Login Protocol
telnet    23/tcp
                     # 24 - private
smtp      25/tcp     mail
...
```

L'arxiu /etc/inetd.conf és la configuració per al servei mestre de xarxa (*inetd server daemon*). Cada línia conté set camps separats per espais: *service socket_type proto flags user server_path server_args*, en què *service* és el servei descrit a la primera columna de l'arxiu /etc/services, *socket_type* és el tipus de sòcol (valors possibles: *stream, dgram, raw, rdm*, o *seqpacket*), *proto* és el protocol vàlid per a aquesta entrada (ha de coincidir amb el de l'arxiu /etc/services), *flags* indica l'acció per prendre quan hi ha una nova connexió sobre un servei que està atenent una altra connexió (*wait* diu a *inetd* que no posi en marxa un nou servidor, i *nowait* significa que *inetd* ha de posar en marxa un nou servidor). *user* serà l'usuari amb el qual s'identificarà qui ha posat en marxa el servei, *server_path* és el directori on es troba el servidor i *server_args* són arguments possibles que seran passats al servidor. Un exemple d'algunes línies de l'arxiu /etc/inetd.conf són (cal recordar que després de # hi ha comentaris, per la qual cosa, si un servei té # abans del nom, significa que no està disponible):

```
...
telnet  stream  tcp   nowait   root   /usr/sbin/tcpd   /usr/sbin/in.telnetd
ftp     stream  tcp   nowait   root   /usr/sbin/tcpd   /usr/sbin/in.ftpd
# fsp   dgram   udp   wait     root   /usr/sbin/tcpd   /usr/sbin/in.fspd
shell   stream  tcp   nowait   root   /usr/sbin/tcpd   /usr/sbin/in.rshd
login   stream  tcp   nowait   root   /usr/sbin/tcpd   /usr/sbin/in.rlogind
# exec  stream  tcp   nowait   root   /usr/sbin/tcpd   /usr/sbin/in.rexecd ...
...
```

A partir de Debian Woody 3.0 r1, la funcionalitat d'*inetd* ha estat reemplaçada per *xinetd* (recomanable), el qual necessita l'arxiu de configuració /etc/xinetd.conf (vegeu el final del mòdul). Si es vol posar en marxa el servei d'*inetd*, s'ha d'executar (i crear els enllaços adequats en els directoris /etc/rcX.d) */etc/init.d/inetd.real start* (vegeu un exemple de configuracions en el punt 15 de l'apartat 12, "Configuracions avançades i eines").

A més de la configuració d'*inetd* o *xinetd*, la configuració típica dels serveis de xarxa en un entorn d'escriptori o servidor bàsic podria incloure a més:

Vegeu també

Per a veure més sobre la configuració típica dels serveis de xarxa en un entorn d'escriptori o servidor bàsic vegeu el mòdul de servidors (el mòdul 2) de l'assignatura *Administració avançada de sistemes GNU/Linux*.

- **ssh**: connexió interactiva segura com a reemplaçament de *telnet;* inclou dos arxius de configuració, /etc/ssh/ssh_config (per al client) i /etc/ssh/sshd_config (per al servidor).
- **exim**: agent de transport de correu (MTA), inclou els arxius de configuració: /etc/exim/exim.conf, /etc/mailname, /etc/alias, /etc/email-addresses.
- **fetchmail**: dimoni per a descarregar el correu d'un compte POP3, amb /etc/fetchmailrc.
- **procmail**: programa per a filtrar i distribuir el correu local, ~/.procmailrc.

- **tcpd**: serveis de filtres de màquines i dominis habilitats i deshabilitats per a connectar-se al servidor (*wrappers*); /etc/hosts.allow, /etc/hosts.deny.

- **DHCP**: servei per a la gestió (servidor) o obtenció d'IP (client), /etc/dhcp3/dhclient.conf (client), /etc/default/dhcp3-server (servidor), /etc/dhcp3/dhcpd.conf (servidor).

- **CVS**: sistema de control de versions concurrents; /etc/cvs-cron.conf, /etc/cvs-pserver.conf.

- **NFS**: sistema d'arxius de xarxa; /etc/etc/exports.

- **Samba**: sistema d'arxius de xarxa i compartició d'impressores en xarxes Windows; /etc/samba/smb.conf.

- **lpr**: dimoni per al sistema d'impressió; /etc/printcap (per al sistema *lpr*, no per a CUPS).

- **Apache** i **Apache2**: servidor de web; /etc/apache/* i /etc/apache2/*.

- **squid**: servidor de servidor cau; /etc/squid/*.

4.5. Configuració addicional: protocols i xarxes

Hi ha altres arxius de configuració que en la majoria dels casos no s'utilitzen però que poden ser interessants. /etc/protocols és un arxiu que relaciona identificadors de protocols amb noms de protocols; així, els programadors poden especificar els protocols pels seus noms en els programes.

Exemple de l'arxiu /etc/protocols

```
ip       0   IP       # internet protocol, pseudo protocol number
#hopopt  0   HOPOPT   # IPv6 Hop-by-Hop Option [RFC1883]
icmp     1   ICMP     # internet control message protocol
```

L'arxiu /etc/networks té una funció similar a /etc/hosts, però respecte a les xarxes indica noms de xarxa amb relació a la seva adreça IP (l'ordre *route* mostrarà el nom de la xarxa i no la seva adreça, en aquest cas).

Exemple de l'arxiu /etc/networks

```
loopnet 127.0.0.0
localnet 192.168.0.0
amprnet 44.0.0.0 ...
```

4.6. Aspectes de seguretat

És important tenir en compte els aspectes de seguretat en les connexions a xarxa, ja que una font d'atacs importants es produeix per mitjà de la xarxa. Ja se'n parlarà més sobre aquest tema en la unitat corresponent a seguretat; tanmateix, hi ha unes quantes recomanacions bàsiques que s'han de tenir en compte per a minimitzar els riscos immediatament abans i després de configurar la xarxa del nostre ordinador:

a) No activar serveis a /etc/inetd.conf que no s'utilitzaran; inserir un # abans del nom per a evitar fonts de risc.

b) Modificar l'arxiu /etc/ftpusers per a denegar que certs usuaris puguin tenir connexió via FTP amb la màquina.

c) Modificar l'arxiu /etc/securetty per a indicar des de quins terminals (un nom per línia; per exemple, *tty1 tty2 tty3 tty4*) es permet la connexió del superusuari (*root*). Des dels terminals restants, *root* no es podrà connectar.

d) Utilitzar el programa *tcpd*. Aquest servidor és un *wrapper* que permet acceptar o negar un servei des d'un determinat node, i es col·loca a /etc/inetd.conf com a mediador d'un servei. El **tcpd** verifica unes regles d'accés a dos arxius: /etc/hosts.allow i /etc/host.deny

Si s'accepta la connexió, posa en marxa el servei adequat passat com a argument (per exemple, la línia del servei d'FTP mostrada abans a inetd.conf: *ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd*).

tcpd primer cerca /etc/hosts.allow i després /etc/hosts.deny. L'arxiu hosts.deny conté la informació sobre quins són els nodes que no tenen accés a un servei dins d'aquesta màquina. Una configuració restrictiva és ALL: ALL, ja que només es permetrà l'accés als serveis des dels nodes declarats a /etc/hosts.allow.

L'arxiu /etc/hosts.equiv permet l'accés a aquesta màquina sense haver d'introduir una clau d'accés (contrasenya). Es recomana no usar aquest mecanisme i aconsellar als usuaris no utilitzar l'equivalent des del compte d'usuari per mitjà de l'arxiu .rhosts.

En Debian és important configurar /etc/security/access.conf, l'arxiu que indica les regles de qui i des d'on es pot connectar (*login*) a aquesta màquina. Aquest arxiu té una línia per ordre amb tres camps separats per ":" del tipus *permís:usuaris:origen*. El primer serà un + o – (accés denegat), el segon un nom d'usuari o usuaris, grup o user@host, i el tercer un nom d'un dispositiu, node, domini, adreces de node o de xarxes, o ALL.

Exemple d'access.conf

Aquesta ordre no permet entrades com a *root* sobre *tty1:*

```
ALL EXCEPT root:tty1 ...
```

Permet accedir a *u1, u2, g1* i tots els de domini remix.com:

```
+:u1 u2 g1 .remix.com:ALL
```

4.7. Opcions d'IP

Hi ha una sèrie d'opcions sobre el trànsit IP que és convenient esmentar. La configuració es fa per mitjà de la inicialització de l'arxiu corresponent en el directori /proc/sys/net/ipv4/. El nom de l'arxiu és el mateix que el de l'ordre i per a activar-los s'ha de posar un 1 dins de l'arxiu, i un 0 per a desactivar-lo. Per exemple, si es vol activar *ip_forward*, s'hauria d'executar:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Els més utilitzats són *ip_forward*, utilitzat per a l'encaminament entre interfícies o amb *IP masquerading; ip_default_ttl*, que és el temps de vida per a un paquet IP (64 mil·lisegons per defecte); i *ip_bootp_agent*, variable lògica (booleana) que accepta paquets (o no) amb adreça origen del tipus 0.b.c.d i destinació d'aquest node, difusió o multidestinació.

Ordres per a la solució de problemes amb la xarxa

Si teniu problemes en la configuració de la xarxa, es pot començar verificant la sortida de les ordres següents per a obtenir una primera idea:

```
ifconfig
cat /proc/pci
cat /proc/interrupts
dmesg | more
```

Per a verificar la connexió a la xarxa, es poden usar les ordres següents (cal tenir instal·lats `netkit-ping`, `traceroute`, `dnsutils`, `iptables` i `net-tools`):

```
ping uoc.edu                    # verificar la connexió a Internet
traceroute uoc.edu              # buscar paquets IP
ifconfig                        # verificar la configuració de l'amfitrió
route -n                        # verificar la configuració de la ruta
dig [@dns.uoc.edu] www.uoc.edu  # verificar registres de www.uoc.edu
                                # sobre el servidor dns.uoc.edu
iptables -L -n |less            # verificar filtratge de paquets (nucli >= 2.4)
netstat -a                      # mostra tots els ports oberts
netstat -l --inet               # mostra els ports en escolta
netstat -ln --tcp               # mostrar ports TCP en escolta (numèric)
```

5. Configuració del DHCP

DHCP són les sigles de *Dynamic Host Configuration Protocol*. La configuració és molt simple i serveix perquè, en lloc de configurar cada node d'una xarxa individualment, es pugui fer de manera centralitzada i l'administració sigui més fàcil. La configuració d'un client és molt fàcil, ja que només s'ha d'instal·lar un dels paquets següents: *dhcp3-client* (versió 3, Internet Software Consortium), *dhcpcd* (Yoichi Hariguchi i Sergei Viznyuk), *pump* (Red Hat), i agregar la paraula *dhcp* en l'entrada corresponent a la interfície que es vol que funcioni sota el client DHCP (per exemple, /etc/network/interfaces ha de tenir *iface eth0 inet dhcp*...).

La configuració del servidor requereix una mica més d'atenció, però no presenta complicacions. Primer, perquè el servidor pugui servir tots els clients DHCP (incloent-hi Windows), s'han de resoldre algunes qüestions prèvies relacionades amb les adreces de difusió. Per a això, primer el servidor ha de poder enviar missatges a l'adreça 255.255.255.255, que no és vàlida en GNU/Linux. Per a provar-ho, executeu:

```
route add -host 255.255.255.255 dev eth0
```

Si apareix el missatge *255.255.255.255: Unknown host*, s'ha d'afegir l'entrada següent a /etc/hosts: *255.255.255.255 dhcp*, i intentar-ho novament:

```
route add -host dhcp dev eth0
```

La configuració de *dhcpd* es pot fer amb la interfície gràfica *linuxconf* o bé editar /etc/dhcpd.conf. Un exemple d'aquest arxiu és:

```
# Exemple de /etc/dhcpd.conf:
default-lease-time 1200;
max-lease-time 9200;
option domain-name "remix.com";
deny unknown-clients;
deny bootp;
option broadcast-address 192.168.11.255;
option routers 192.168.11.254;
option domain-name-servers 192.168.11.1, 192.168.168.11.2;
subnet 192.168.11.0 netmask 255.255.255.0
{ not authoritative;
      range 192.168.11.1 192.168.11.254
      host mart {
         hardware ethernet 00:00:95:C7:06:4C;
         fixed address 192.168.11.146;
```

```
            option host-name "mart";
        }
        host saturn {
            hardware ethernet 00:00:95:C7:06:44;
            fixed address 192.168.11.147;
            option host-name "saturn";
        }
    }
```

Això permetrà al servidor assignar el rang d'adreces 192.168.11.1 al 192.168.11.254, tal com es descriu en cada node. Si no hi ha el segment *host* {...} corresponent, s'assignen aleatòriament. Les IP són assignades per un temps mínim de 1.200 segons i màxim de 9.200 (en cas que no hi hagi aquests paràmetres, s'assignen indefinidament).

Abans d'executar el servidor, s'ha de verificar si tenim el fitxer /var/state/dhcp/dhcpd.leases (en cas contrari, caldrà crear-lo amb **touch /var/state/dhcp/dhcpd.leases**). Per a executar el servidor: **/usr/sbin/dhcpd** (o bé posar-lo en els scripts d'inicialització). Amb **/usr/sbin/dhcpd -d -f** es podrà veure l'activitat del servidor sobre la consola del sistema. [Mou01, Rid00, KD00, Dra99]

6. Aliàsing d'IP

Hi ha algunes aplicacions en què és útil configurar múltiples adreces IP a un únic dispositiu de xarxa. Els ISP[27] utilitzen freqüentment aquesta característica per a proveir de característiques personalitzades (per exemple, de World Wide Web i FTP) als seus usuaris. Per a això, el nucli ha d'estar compilat amb les opcions de *network aliasing* i IP (*aliasing support*).

[27] De l'anglès *Internet service providers*

Després d'instal·lar el nou nucli, la configuració és molt fàcil. Els àlies són annexats a dispositius de xarxa virtuals associats al nou dispositiu amb un format com ara:

```
dispositiu: número virtual
```

Per exemple:

```
eth0:0, ppp0:8
```

Considerem que tenim una xarxa Ethernet que suporta dues subxarxes IP diferents simultàniament i que la nostra màquina hi vol tenir accés directe. Un exemple de configuració seria:

```
ifconfig eth0 192.168.110.23 netmask 255.255.255.0 up
route add -net 192.168.110.0 netmask 255.255.255.0 eth0
ifconfig eth0:0 192.168.10.23 netmask 255.255.255.0 up
route add -net 192.168.10.0 netmask 255.255.255.0 eth0:0
```

Això significa que tindrem dues IP, 192.168.110.23 i 192.168.10.23, per a la mateixa NIC. Per a esborrar un àlies, cal agregar un "-" al final del nom (per exemple, *ifconfig eth0:0- 0*). [Mou01, Ran05]

Un cas típic és que es vulgui configurar una única targeta Ethernet perquè sigui la interfície de diferents subxarxes IP. Per exemple, suposem que tenim una màquina que es troba en una xarxa LAN 192.168.0.x/24, i volem connectar la màquina a Internet usant una adreça IP pública proporcionada amb DHCP, usant la targeta Ethernet existent.

Per exemple, es pot fer com en l'exemple anterior o també editar l'arxiu `/etc/network/interfaces`, de manera que inclogui una secció similar a la següent:

```
iface eth0 inet static
      address 192.168.0.1
      netmask 255.255.255.0
```

```
            network 192.168.0.0
            broadcast 192.168.0.255

      iface eth0:0 inet dhcp
```

La interfície *eth0:0* és una interfície virtual i, en activar-se, també ho farà el seu pare *eth0*.

7. IP masquerade

L'*IP masquerade* és un recurs perquè un conjunt de màquines puguin utilitzar una única adreça IP. Això permet que els nodes ocults puguin sortir cap a Internet (són els que utilitzen una IP privada; per exemple, 198.162.10.1); però no poden acceptar trucades o serveis de l'exterior directament, sinó per mitjà de la màquina que té la IP real.

Això significa que alguns serveis no funcionen (per exemple, *talk*) i d'altres han de ser configurats en mode PASV (passiu) perquè funcionin (per exemple, FTP). Tanmateix, WWW, *telnet* o IRC funcionen adequadament. El nucli ha d'estar configurat amb les opcions següents: *network firewalls, TCP/IP networking, IP:forwarding/gatewaying, IP:masquerading*. Normalment, la configuració més comuna és disposar d'una màquina amb una connexió SLIP o PPP i tenir un altre dispositiu de xarxa (per exemple, una targeta Ethernet) amb una adreça de xarxa reservada. Com vam veure, i d'acord amb l'RFC 1918, es poden utilitzar com a IP privades els rangs d'adreces següents (IP/màscara):

- 10.0.0.0/255.0.0.0,
- 172.16.0.0/255.240.0.0,
- 192.168.0.0/255.255.0.0.

Els nodes que han de ser ocultats (*masqueraded*) seran dins d'aquesta segona xarxa. Cada una d'aquestes màquines hauria de tenir l'adreça de la màquina que fa el *masquerade*, com a passarel·la per defecte o encaminador. Sobre la màquina esmentada podem configurar:

- Ruta de xarxa per a Ethernet considerant que la xarxa té un *IP=192.168.1.0/255.255.255.0:*

```
route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

- Ruta per defecte per a la resta d'Internet:

```
route add default ppp0
```

- Tots els nodes sobre la xarxa 192.168.1/24 seran *masqueraded:*

```
ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

- Si s'utilitza *iptables* sobre un nucli (o *kernel*) 2.4 o superior:

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Consulteu les referències i la unitat que tracta sobre la seguretat de la informació d'*ipchains* i *iptables*. [Ran05, KD00]

8. NAT amb el nucli 2.2 o superiors

La IP Network Address Translation (NAT) és el reemplaçament que deixa obsoletes les prestacions d'*IP masquerading* en GNU/Linux, i que aporta noves prestacions al servei. Dins de les millores introduïdes en la pila de TCP/IP del nucli 2.2 de GNU/Linux tenim que el NAT forma part del nucli. Per a utilitzar-lo, és necessari que el nucli es compili amb CONFIG_IP_ADVANCED_ROUTER, CONFIG_IP_MULTIPLE_TABLES i CONFIG_IP_ROUTE_NAT. I si es necessita control exhaustiu de les regles NAT (per exemple, per a activar el tallafocs) ha d'estar activat també CONFIG_IP_FIREWALL i CONFIG_IP_ROUTE_FWMARK. Per a treballar amb aquestes noves característiques, és necessari usar el programa *ip* (inclòs en les distribucions més importants a partir de la versió 2.4 del kernel http://es.wikipedia.org/wiki/Iproute2). Llavors, per a traslladar adreces de paquets d'entrada es pot utilitzar:

```
ip route add nat <extaddr>[/<masklen>] via <intaddr>
```

Això farà que un paquet d'entrada destinat a *ext.-addr* (l'adreça visible des de fora d'Internet) es transcriu la seva adreça destinació a *int-addr* (l'adreça de la seva xarxa interna per mitjà de la passarel·la o tallafocs). El paquet s'encamina d'acord amb la taula local d'encaminament. Es poden traslladar adreces simples o blocs. Per exemple:

```
ip route add nat 240.0.11.34 via 192.109.0.2
ip route add nat 240.0.11.32/27 via 192.109.0.0
```

El primer fa que l'adreça interna 192.109.0.2 sigui accessible com a 240.0.11.34. El segon resitua (*remapping*) el bloc 192.109.0.0-31 a 240.0.11.32-63. En aquest cas s'han utilitzat com a exemple translacions a adreces de la classe DE, com ara 240.0.*.*, a fi de no utilitzar cap adreça pública. L'usuari haurà de reemplaçar aquestes adreces (240.0.11.34 i 240.0.11.32-63) per les adreces públiques corresponents a les quals vulgui fer la translació. [Ran05]

9. Com cal configurar una connexió DialUP i PPP?

Si bé avui dia és poc habitual treballar amb mòdem, ja que es tenen solucions d'ASDL amb preus i amplada de banda millors, es farà una petita introducció sobre la configuració d'una connexió de marcatge sobre PPP en GNU/Linux, que és molt simple.

PPP,[28] que permet fer IP-Links entre dos ordinadors amb un mòdem (heu de considerar que ha de ser un mòdem suportat per GNU/Linux, ja que no tots, especialment els interns, coneguts com a *winmodems*, es poden configurar, ja que molts necessiten programari addicional per a establir la comunicació). [Vas00, Law07, Sec00]

(28) De l'anglès *point to point protocol*.

Com a passos previs s'ha de disposar de la informació següent: l'*init-string* del mòdem (normalment no és necessari, però si es necessita i no en tenim de disponible, es pot utilitzar ATZ, que funciona en la majoria dels mòdems, o es poden consultar llistes especialitzades d'*init-strings*).

A més, necessitarem les dades de l'ISP: identificació de connexió (*login name*), clau (*contrasenya*) i número de telèfon. Seria aconsellable tenir adreces de DNS, però és opcional en les versions actuals de *pppd*. Cal verificar a més que el mòdem estigui correctament connectat. Amb un mòdem extern s'ha d'executar *echo > /dev/ttyS0* i mirar els llums del mòdem per si tenen activitat. En cas contrari, cal intentar-ho amb *ttyS1* per si el mòdem està connectat al segon port sèrie. Amb un mòdem intern cal consultar el manual de maquinari suportat per a veure si aquest mòdem pot ser reconegut per GNU/Linux, i en cas afirmatiu, potser cal reconfigurar el nucli per a utilitzar-lo. També podem utilitzar *cat /proc/pci* per si es troba en el bus PCI. [PPP00]

La manera més fàcil de configurar ara el mòdem és per mitjà del paquet *kppp* (cal instal·lar els paquets *kdenetwork-ppp** i *ppp**). Sobre un terminal, executeu */usr/bin/kppp*. Sobre la finestra, completeu les opcions següents:

- Accounts → New Connection
- Dial → Authentication → 'PAP/CHAP'
- Store Password → yes
- IP → Dynamic IP Address
- Autoconfigure hostname → No
- *Gateway* → Default *Gateway* → Assign the Default Route
- DNS → Configuration Automatic → Disable existing DNS
- Device → ttyS1(com1) o ttyS2 (com2)
- Modem → Query Modem (per a veure els resultats; si no obteniu resultats, canvieu el dispositiu *ttySx*).

Entrarem nom d'entrada (o *login*) i contrasenya, i estarem connectats a Internet (per a verificar la connexió es pot executar *ping www.google.com*, per exemple). Aquí s'ha utilitzat el paquet *kppp*, però igualment es podria utilitzar *linuxconf* o *gnomeppp* indistintament.

Una manera ràpida de configurar *pppd* en Debian consisteix a usar el programa *pppconfig*, que ve amb el paquet del mateix nom. *pppconfig* configura els arxius com els anteriors després de formular preguntes a l'usuari per mitjà d'una interfície de menús. Una altra opció diferent per a usar *pppd* consisteix a executar-lo des de *wvdial*, que ve amb el paquet *wvdial*. En comptes de fer que *pppd* executi *chat* per a marcar i negociar la connexió, *wvdial* fa el marcatge, la negociació inicial i després inicia *pppd* perquè faci la resta. En la majoria dels casos donant només el número telefònic, el nom d'usuari i la contrasenya, *wvdial* aconsegueix establir la connexió.

Una vegada configurat PPP perquè funcioni, per exemple amb *el_meu _isp*, cal editar /etc/network/interfaces, de manera que inclogui una secció com la següent (les ordres *ifup, ifdown* utilitzen les instruccions *pon* i *poff* per a configurar interfícies PPP):

```
iface ppp0 inet ppp
      provider el_meu_isp
   amb aquesta secció, ifup ppp0 fa:
      posa el_meu_isp
```

Actualment no és possible usar ***ifup-down*** per a fer una configuració auxiliar de les interfícies PPP. Com *pon* desapareix abans que *pppd* hagi acabat d'establir la connexió, ***ifup*** executa els scripts *up* abans que la interfície PPP estigui preparada per a utilitzar-la. Fins que se solucioni aquest error continua essent necessari fer una configuració posterior a */etc/ppp/ip-up* o */etc/ppp/ip-up.d/*.

Molts proveïdors de serveis d'Internet (ISP) de banda ampla utilitzen PPP per a negociar les connexions fins i tot quan les màquines dels clients estan connectades mitjançant Ethernet o xarxes ATM. Això s'aconsegueix mitjançant PPP sobre Ethernet (PPPoE), que és una tècnica per a l'encapsulament del flux PPP dins de les trames Ethernet. Suposem que l'ISP es diu *el_meu_isp*. Primer cal configurar PPP i PPPoE per a *el_meu_isp*. La manera més fàcil de fer-ho consisteix a instal·lar el paquet ***pppoeconf*** i executar ***pppoeconf*** des de la consola. A continuació, cal editar */etc/network/interfaces* de manera que inclogui un fragment com el següent:

```
iface eth0 inet ppp
      provider el_meu_isp
```

De vegades sorgeixen problemes amb PPPoE relatius a la unitat de transmissió màxima (MTU[29]) en línies DSL[30]; es pot consultar el DSL-HOWTO per a més detalls. També s'ha de tenir en compte si el mòdem té un encaminador, perquè el mòdem/encaminador maneja per si sol la connexió PPPoE i apareix del costat de la LAN com una simple porta d'enllaç Ethernet a Internet.

(29) De l'anglès *maximum transmit unit*.

(30) De l'anglès *digital subscriber line*.

10. Configuració de la xarxa mitjançant *hotplug*

El paquet *hotplug* permet el suport d'arrencada en calent (s'ha de tenir instal·lat el paquet del mateix nom). El maquinari de xarxa es pot connectar en calent tant durant l'arrencada, després d'haver inserit la targeta en la màquina (una targeta PCMCIA, per exemple), com després que una utilitat com *discover* s'hagi executat i hagin estat carregats els mòduls necessaris. Quan el nucli detecta nou maquinari, inicialitza el controlador per al maquinari i després executa el programa *hotplug* per a configurar-lo. Si més tard s'elimina el maquinari, executa novament *hotplug* amb paràmetres diferents. En Debian, quan es crida *hotplug* aquest executa els scripts dels directoris /etc/hotplug/ i /etc/hotplug.d/ . El maquinari de xarxa recentment connectat és configurat per /etc/hotplug/net.agent. Suposem que la targeta de xarxa PCMCIA ha estat connectada, cosa que implica que la interfície *eth0* està preparada per a utilitzar-la.

/etc/hotplug/net.agent fa el següent:

```
ifup eth0=hotplug
```

Llevat que hàgiu afegit una interfície lògica anomenada *hotplug* a /etc/network/interfaces, aquesta ordre no farà res. Perquè aquesta ordre configuri *eth0*, cal afegir les línies següents a /etc/network/interfaces:

```
mapping hotplug
      script echo
```

Si només voleu que *eth0* s'activi en calent i no altres interfícies, cal utilitzar *grep* en comptes d'*echo*, com es mostra a continuació:

```
mapping hotplug
      script grep
      map eth0
```

`ifplugd` activa o desactiva una interfície segons si el maquinari subjacent està connectat a la xarxa o no ho està. El programa pot detectar un cable connectat a una interfície Ethernet o un punt d'accés associat a una interfície Wi-Fi. Quan ***ifplugd*** veu que l'estat de l'enllaç ha canviat, executa un script que per defecte executa ***ifup*** o ***ifdown*** per a la interfície. ***ifplugd*** funciona en combinació amb ***hotplug***. En inserir una targeta, cosa que significa que la interfície està preparada per a utilitzar-la, */etc/hotplug.d/net/ifplugd.hotplug* inicia una instància d'***ifplugd*** per a aquesta interfície. Quan ***ifplugd*** detecta que la targeta està connectada a una xarxa, executa ***ifup*** per a aquesta interfície.

Per a associar una targeta Wi-Fi amb un punt d'accés, pot ser que necessiteu programar-la amb una clau de xifratge WEP adequada. Si esteu usant ***ifplugd*** per a controlar ***ifup*** com es va explicar anteriorment, llavors, evidentment, no podreu configurar la clau de xifratge usant ***ifup***, ja que aquest només és cridat després que la targeta ha estat associada. La solució més simple és usar ***waproamd***, que configura la clau de xifratge WEP segons els punts d'accés disponibles, que es descobreixen mitjançant la recerca de les xarxes Wi-Fi. Per a més informació consulteu ***man waproamd*** i la informació del paquet.

11. Xarxa privada virtual (VPN)

Una VPN[31] és una xarxa que utilitza Internet com a transport de dades, però impedeix que hi puguin accedir membres externs.

(31)De l'anglès *virtual private network*.

Tenir una xarxa amb VPN significa tenir nodes units per mitjà d'un túnel per on viatja el trànsit i on ningú no hi pot interactuar. S'utilitza quan es tenen usuaris remots que accedeixen a una xarxa corporativa per a mantenir la seguretat i privacitat de les dades. Per a configurar una VPN es poden utilitzar diversos mètodes SSH (SSL), CIPE, IPSec, PPTP, que es poden consultar en les referències (es recomana consultar VPN PPP-SSH HOWTO, de Scott Bronson, i VPN-HOWTO de Matthew D. Wilson). [Bro01, Wil02]

Per a fer les proves de configuració, en aquest apartat s'utilitzarà **OpenVPN**, que és una solució basada en SSL VPN, i es pot usar per a un ampli rang de solucions; per exemple, accés remot, VPN punt a punt, xarxes Wi-Fi segures o xarxes distribuïdes empresarials. OpenVPN implementa les capes OSI 2 o 3 amb protocols SSL/TLS i suporta autenticació basada en certificats, targetes (*smart cards*), i altres mètodes de certificació. OpenVPN no és un servidor intermediari (o *proxi*) d'aplicacions ni opera per mitjà d'un navegador web.

Per a analitzar aquest servei utilitzarem una opció d'OpenVPN anomenada *OpenVPN for Static key configurations*, que ofereix una manera simple de configurar una VPN ideal per a proves o per a connexions punt a punt. Els seus avantatges són simplicitat, i que no és necessari un certificat X509 PKI[32] per a mantenir la VPN. Els desavantatges són que només permet un client i un servidor. En no utilitzar clau pública i clau privada, hi pot haver igualtat de claus amb sessions anteriors, i hi ha d'haver, doncs, una clau en mode text en cada igual (*peer*), i la clau secreta ha de ser intercanviada anteriorment per un canal segur.

(32)De l'anglès *public key infrastructure*.

11.1. Exemple simple

En aquest exemple es configurarà un túnel VPN sobre un servidor amb IP = 10.8.0.1 i un client amb IP = 10.8.0.2. La comunicació serà xifrada entre el client i el servidor sobre el *port* UDP 1194, que és el port per defecte d'OpenVPN. Després d'instal·lar el paquet s'haurà de generar la clau estàtica:

```
openvpn --genkey --secret static.key
```

Després s'ha de copiar l'arxiu *static.key* en l'altre igual sobre un canal segur (per exemple, utilitzant **ssh** o **scp**). L'arxiu de configuració del servidor *openVPN_server*, per exemple:

```
dev tun
ifconfig 10.8.0.1 10.8.0.2
secret static.key
```

L'arxiu de configuració del client, per exemple, *openVPN_client:*

```
remote myremote.mydomain
dev tun
ifconfig 10.8.0.2 10.8.0.1
secret static.key
```

Abans de verificar el funcionament de la VPN, s'ha d'assegurar en el tallafocs que el port 1194 UDP està obert sobre el servidor i que la interfície virtual ***tun0*** usada per OpenVPN no està bloquejada ni sobre el client ni sobre el servidor.

Tingueu en ment que el 90% dels problemes de connexió trobats per usuaris nous d'OpenVPN estan relacionats amb el tallafocs.

Per a verificar l'OpenVPN entre dues màquines, haureu de canviar les IP per les reals i el domini pel que tingueu, i després executar del costat del servidor:

```
openvpn [server config file]
```

Que donarà una sortida com la següent:

```
Sun Feb 6 20:46:38 2005 OpenVPN 2.0_rc12 i686-suse-linux [SSL] [LZO] [EPOLL]
                        built on Feb 5 2005
Sun Feb 6 20:46:38 2005 Diffie-Hellman initialized with 1024 bit key
Sun Feb 6 20:46:38 2005 TLS-Auth MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Sun Feb 6 20:46:38 2005 TUN/TAP device tun1 opened
Sun Feb 6 20:46:38 2005 /sbin/ifconfig tun1 10.8.0.1 pointopoint 10.8.0.2 mtu 1500
Sun Feb 6 20:46:38 2005 /sbin/route add -net 10.8.0.0 netmask 255.255.255.0
                        gw 10.8.0.2
Sun Feb 6 20:46:38 2005 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:23 ET:0
                        EL:0 AF:3/1 ]
Sun Feb 6 20:46:38 2005 UDPv4 link local (bound): [undef]:1194
Sun Feb 6 20:46:38 2005 UDPv4 link remote: [undef]
Sun Feb 6 20:46:38 2005 MULTI: multi_init called, r=256 v=256
Sun Feb 6 20:46:38 2005 IFCONFIG POOL: base=10.8.0.4 size=62
Sun Feb 6 20:46:38 2005 IFCONFIG POOL LIST
Sun Feb 6 20:46:38 2005 Initialization Sequence Completed
```

I del costat client:

```
openvpn [client config file]
```

Per a verificar que funciona, es pot fer *ping 10.8.0.2* des del servidor i *ping 10.8.0.1* des del client.

Nota

Per a més informació podeu consultar el web d'OpenVPN: http://openvpn.net/howto.html

Per a agregar compressió sobre l'enllaç, s'ha d'afegir la línia següent als dos arxius de configuració:

```
comp-lzo
```

Per a protegir la connexió per mitjà d'un encaminador NAT/*firewall alive*, i seguir els canvis d'IP per mitjà d'un DNS, si un dels iguals (*peers*) canvia, cal agregar als dos arxius de configuració:

```
keepalive 10 60
ping-timer-rem
persist-tun
persist-key
```

Per a executar-ho com a dimoni amb els privilegis del grup o usuari **nobody**, **cal** agregar als arxius de configuració:

```
user nobody
group nobody
daemon
```

11.2. Configuració (manual) d'un client Debian per a accedir a un VPN sobre un túnel *pptp*

En primer lloc s'ha d'instal·lar el client PPTP,[33] i no és necessari tenir suport **MPPE** en el nucli.

[33] De l'anglès *point-to-point tunneling protocol*.

Per a això farem:

```
apt-get update
apt-get install pptp-linux
apt-get install resolvconf
```

Cal contestar *sí* si surt el missatge *Append original file to dynamic file?*

Després, cal reiniciar les interfícies amb:

```
/etc/init.d/ifupdown restart
```

O amb:

```
/etc/init.d/networking restart
```

També és necessari tenir instal·lat el paquet *iproute:*

```
apt-get install iproute
```

Per a la configuració del client crearem en el seu directori (per defecte */etc/ppp/*) el fitxer */etc/ppp/options.pptp* (que contindrà opcions comunes a tots els túnels que es creïn en l'equip):

```
lock noauth nobsdcomp nodeflate
```

Agreguem la línia següent a l'arxiu /etc/ppp/chap-secrets:

```
usuari PPTP passwd *
```

On diu *usuari* s'ha de posar el nom de l'usuari del servidor de VPN i en *passwd* la paraula clau d'accés, i cal acabar la línia amb un asterisc.

S'ha de crear l'arxiu /etc/ppp/peers/uoc amb els paràmetres de configuració del túnel:

```
pty "pptp nom.domini --nolaunchpppd"
name usuari
remotename PPTP
usepeerdns
defaultroute replacedefaultroute
file /etc/ppp/options.pptp
ipparam uoc
connect "route add nom.domini gw `ip route | \
grep default | cut -f3 -d' '`"
```

On diu *usuari* s'ha de posar el nom de l'usuari, *nom.domini* és el nom del servidor VPN, per exemple, *vpngw.uoc.es VPN*, i *uoc* és el nom del túnel que crearem i que es referirà a l'arxiu /etc/ppp/peers/uoc.

Abans de finalitzar hem de crear un parell de scripts que encaminaran els paquets pel túnel cap a Internet i el tancaran quan s'hagi acabat.

Creem l'arxiu /etc/ppp/ip-up.d/uoc amb permisos d'execució i amb el contingut següent:

```
#!/bin/sh
cat /etc/ppp/resolv.conf | resolvconf -a tun0
```

Li canviem els atributs d'execució:

```
chmod +x /etc/ppp/ip-up.d/uoc
```

Finalment, creem l'arxiu /etc/ppp/ip-down.d/uoc amb el contingut següent:

```
#!/bin/sh
route del nom.domini
# canviar pel servidor, per exemple, vpngw.uoc.es
resolvconf -d tun0
```

I també amb permisos d'execució:

```
chmod +x /etc/ppp/ip-down.d/uoc
```

Ara només falta posar en marxa el túnel amb **pon uoc.**

Després de posar en marxa el túnel i de fer **ifconfig**, veurem una nova interfície de xarxa amb el nom de **ppp0.** Per a finalitzar la utilització del túnel, només haurem de fer **poff uoc**, amb la qual cosa desapareixerà la interfície **ppp0.**

Hi ha una interfície gràfica anomenada **kvpnc** per a configurar diferents clients de VPN. En aquesta aplicació no és necessari crear els scripts i no hi ha cap problema per a instal·lar un client VPN **pptp** o de qualsevol altre tipus seguint les instruccions. També és possible configurar fàcilment un client des del Gnome per mitjà del Gnome Network Manager per a **pptp:**

```
sudo apt-get install network-manager-pptp pptp-linux
```

Nota

Podeu consultar els detalls de la interfície gràfica *kvpnc* a la seva pàgina web:

http://home.gna.org/kvpnc/en/index.html

Un altre web interessant on trobareu informació sobre VPN és la pàgina del Network Manager:

http://projects.gnome.org/NetworkManager/admins/

12. Configuracions avançades i eines

Hi ha un conjunt de paquets complementaris (o que substitueixen els convencionals) i eines que o bé milloren la seguretat de la màquina (recomanats en ambients hostils), o bé ajuden en la configuració de xarxa (i del sistema en general) de manera més amigable.

Aquests paquets poden ser de gran ajuda a l'administrador de xarxa per a evitar intrusos o usuaris locals que s'excedeixen en les seves atribucions (generalment, no per part de l'usuari local, sinó per mitjà d'una suplantació d'identitat) o bé ajudar l'usuari novell a configurar adequadament els serveis.

En aquest sentit, és necessari preveure:

1) Configuració avançada de TCP/IP: per mitjà de l'ordre *sysctl* és possible modificar els paràmetres del nucli durant l'execució o en l'inici per a ajustar-los a les necessitats del sistema. Els paràmetres susceptibles de modificar són els que es troben en el directori /proc/sys/, i es poden consultar amb **sysctl -a**. La manera més simple de modificar aquests paràmetres és per mitjà de l'arxiu de configuració /etc/syscntl.conf. Després de la modificació, s'ha de tornar a engegar la xarxa:

```
/etc/init.d/networking restart
```

En aquest apartat veurem algunes modificacions per a millorar les prestacions de la xarxa (segons les condicions) o la seguretat del sistema (consulteu les referències per a més detalls) [Mou01]:

```
net.ipv4.icmp_echo_ignore_all = 1
```

2) No respon paquets ICMP, com per exemple l'ordre *ping*, que podria significar un atac DoS (*denial-of-service*).

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

3) Evita congestions de xarxa no responent la difusió (*broadcast*).

```
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.lo.accept_source_route = 0
net.ipv4.conf.eth0.accept_source_route = 0
```

```
net.ipv4.conf.default.accept_source_route = 0
```

4) Inhibeix els paquets d'IP *source routing* que podrien representar un problema de seguretat (en totes les interfícies).

```
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.accept_redirects = 0
```

5) Permet rebutjar un atac DoS de paquets SYNC que consumiria tots els recursos del sistema i forçaria a fer un reinici de la màquina.

```
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
```

6) Útil per a evitar atacs amb CMP *redirect acceptance* (aquests paquets són utilitzats quan l'encaminament no té una ruta adequada) en totes les interfícies.

```
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

7) Envia alertes sobre tots els missatges erronis a la xarxa.

```
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
```

8) Habilita la protecció contra l'IP *spoofing* en totes les interfícies.

```
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.lo.log_martians = 1
net.ipv4.conf.eth0.log_martians = 1
net.ipv4.conf.default.log_martians = 1
```

9) Generarà registres sobre tots els *spoofed packets, source routed packets i redirect packets*.

10) Els paràmetres següents permetran que el sistema pugui atendre millor i més ràpidament les connexions TCP.

```
net.ipv4.tcp_fin_timeout = 40, Per defecte, 60.
net.ipv4.tcp_keepalive_time = 3600, Per defecte, 7.200.
net.ipv4.tcp_window_scaling = 0
net.ipv4.tcp_sack = 0
net.ipv4.tcp_timestamps = 0, Per defecte, tots a 1 (habilitats).
```

11) Iptables: les últimes versions de GNU/Linux (nucli 2.4 o superiors) inclouen nous mecanismes per a construir filtres de paquets anomenats *netfilter* [Mou01]. Aquesta nova funcionalitat és gestionada per una eina denominada *iptables*, que presenta característiques millors que el seu predecessor (*ipchains*). Com es veurà en el mòdul corresponent a seguretat, és summament fàcil construir un tallafocs amb aquesta eina per a detectar els atacs *scans* més comuns, DoS, IPMAC *spoofing*, etc., i fer-los front. Per a activar-lo cal que el nucli sigui 2.4 o superior, que estigui configurat per a donar suport d'*ipfilter* (cosa que significarà que s'haurà de recompilar el nucli per a activar l'opció *network packet filtering* [CONFIG_NETFILTER], i totes les subopcions específiques). Les regles específiques s'han d'activar durant l'arrencada (per exemple, per mitjà de l'script /etc/init.d i l'enllaç adequat al directori rc adequat) i té un format similar (consulteu les referències sobre les capacitats i la sintaxi completa) a:

```
iptables -A Type -i Interface -p protocol -s SourceIP
--source-port Port -d DestinationIP
--destination-port Port -j Action
```

12) GnuPG: aquesta eina permet xifrar dades per a la tramesa posterior (per exemple, correu electrònic) o emmagatzemament, i també per a generar firmes digitals (compleix l'estàndard de l'RFC2440) i no utilitza algoritmes patentats, la qual cosa significa més llibertat en la utilització però pèrdua de compatibilitat amb altres eines (per exemple, PGP 2.0) que utilitzen algoritmes com IDEA i RSA. Per a compilar-lo o instal·lar-lo, cal seguir les indicacions dels autors. En primer lloc, s'ha de crear una parella de claus (pública i privada) executant com a *root* l'ordre *gpg --gen-key* dues vegades i contestant les preguntes que ens fa. Generalment, aquestes claus s'emmagatzemaran a /root. El següent és exportar (per exemple, a una pàgina web) la clau pública perquè altres usuaris la puguin utilitzar per a xifrar els correus o la informació que només podrà veure l'usuari que ha generat la clau pública. Per a això, caldrà utilitzar *gpg --export -ao UID*, la qual cosa generarà un arxiu ASCII de la clau pública de l'usuari UID.

Per a importar una clau pública d'un altre usuari, es pot usar *gpg --import filename*, i per a signar una clau (significa indicar al sistema que s'està d'acord que la clau firmada és de qui diu ser), es pot utilitzar *gpg --sign-key UID*. Per a verificar una clau, es pot utilitzar *gpg --verify file/data* i per a xifrar o desxifrar, *gpg -sear UID file g* i *gpg -d file*, respectivament [Gnu].

13) Logcheck: una de les activitats d'un administrador de xarxa és verificar diàriament (més d'una vegada per dia) els arxius de registre (*log*) per a detectar possibles atacs, intrusions o esdeveniments que puguin donar indicis sobre aquestes qüestions. Aquesta eina selecciona (dels arxius de registre) informació condensada de problemes i riscos potencials i després envia aquesta informació al responsable, per exemple, per mitjà d'un correu. El paquet inclou

utilitats per a executar-se de manera autònoma i recordar l'última entrada verificada per a les execucions subsegüents. Per a més informació sobre la configuració i instal·lació, podeu consultar les referències. [Log]

14) PortSentry i Tripwire: aquestes eines ajuden en les funcions de l'administrador de xarxa pel que fa a seguretat. **PortSentry** permet detectar i respondre a accions de recerca de ports (pas previ a un atac o a un *spamming*) en temps real i prendre diverses decisions respecte a l'acció que s'està duent a terme. **Tripwire** és una eina que ajudarà l'administrador notificant sobre possibles modificacions i canvis en arxius per a evitar possibles danys (més greus). Aquesta eina compara les diferències entre els arxius actuals i una base de dades generada prèviament per a detectar canvis (insercions i esborrament), la qual cosa és molt útil per a detectar possibles modificacions d'arxius vitals, com per exemple, en arxius de configuració. Consulteu les referències sobre la instal·lació i configuració d'aquestes eines. [Tri]

15) Xinetd: aquesta eina millora notablement l'eficiència i les prestacions d'*inetd* i *tcp-wrappers*. Un dels grans avantatges de **xinetd** és que pot fer front a atacs de DoA[34] per mitjà de mecanismes de control per als serveis basats en la identificació d'adreces del client, en temps d'accés i temps de connexió (*logging*). No s'ha de pensar que *xinetd* és el més adequat per a tots els serveis (per exemple, FTP i SSH és millor que s'executin només com a dimonis), ja que molts generen una gran sobrecàrrega al sistema i disposen de mecanismes d'accés segurs que no creen interrupcions en la seguretat del sistema. [Xin]

(34) De l'anglès *denial-of-access*.

La compilació o instal·lació és simple, i només és necessari configurar dos arxius: /etc/xinetd.conf (l'arxiu de configuració de *xinetd*) i /etc/rc.d/init.d/xinetd (l'arxiu d'inicialització de *xinetd*). El primer arxiu conté dues seccions: *defaults*, que és on es troben els paràmetres que s'aplicaran a tots els serveis, i *service*, que seran els serveis que es posaran en marxa per mitjà de *xinetd*.

Un exemple típic de la configuració podria ser:

```
# xinetd.conf
# Les opcions de configuració per defecte que s'apliquen a tots els
# servidors es poden modificar per a cada servei
defaults
{
   instances = 10
   log_type = FILE /var/log/service.log
   log_on_success = HOST PID
   log_on_failure = HOST RECORD
}
# El nom del servei ha de ser a /etc/services per a obtenir
# el port correcte
# Si es tracta d'un servidor/port no estàndard, useu "port = X"
service ftp
```

```
{
   socket_type = stream
   protocol = tcp
   wait = no
   user = root
   server = /usr/sbin/proftpd
}
#service telnet
#{
# socket_type = stream
# protocol = tcp
# wait = no
# user = root
# no_access = 0.0.0.0
# only_from = 127.0.0.1
# banner_fail = /etc/telnet_fail
# server = /usr/sbin/in.telnetd
#}
service ssh
{
   socket_type = stream
   protocol = tcp
   wait = no
   user = root
   port = 22
   server = /usr/sbin/sshd
   server_args = -i
}
service http
{
   socket_type = stream
   protocol = tcp
   wait = no
   user = root
   server = /usr/local/apache/bin/httpd
}
#service finger
#{
# socket_type = stream
# protocol = tcp
# wait = no
# user = root
# no_access = 0.0.0.0
# only_from = 127.0.0.1
# banner_fail = /etc/finger_fail
# server = /usr/sbin/in.fingerd
# server_args = -l
```

```
#}
# Fi de /etc/xinetd.conf
```

Els serveis comentats (#) no estaran disponibles. A la secció *defaults* es poden inserir paràmetres com ara el nombre màxim de peticions simultànies d'un servei, el tipus de registre (*log*) que es vol tenir, des de quins nodes es rebran peticions per defecte, el nombre màxim de peticions per IP que s'atendran, o serveis que s'executaran com a superservidors (*imapd* o *popd*), com per exemple:

```
default {
instances = 20
log_type = SYSLOG
authpriv log_on_success = HOST
log_on_failure = HOST
only_from = 192.168.0.0/16
per_source = 3
enabled = imaps
}
```

La secció *service*, una per cada servei, com per exemple:

```
service imapd {
socket_type = stream
wait = no
user = root
server = /usr/sbin/imapd
only_from = 0.0.0.0/0 #allows every client
no_access = 192.168.0.1
instances = 30
log_on_success += DURATION USERID
log_on_failure += USERID
nice = 2
redirect = 192.168.1.1 993
#Permet redireccionar el trànsit del port 993
#cap al node 192.168.1.1
bind = 192.168.10.4
#Permet indicar a quina interfície està associat el servei per a evitar
# problemes de suplantació de servei.
}
```

L'arxiu /etc/init.d/xinetd permetrà posar en marxa el servidor (amb l'enllaç adequat, segons el nivell d'execució seleccionat; per exemple, 3, 4 i 5). És convenient canviar els atributs de tots dos arxius per a garantir que no són modificats o desactivats amb *chmod 700 /etc/init.d/xinetd; chown 0.0 /etc/init.d/xconfig; chmod 400 /etc/xinetd.conf; chattr +i /etc/xinetd.conf*.

16) Linuxconf: és una eina de configuració i administració d'un sistema GNU/Linux però que ha quedat obsoleta, si bé es pot trobar encara en algunes distribucions.

Nota

Més informació a Solucorp.
http://www.solucorp.qc.ca/linuxconf/

17) Webmin: és una altra eina (paquets *webmin-core, webmin-dhcp, webmin-inetd, webmin-sshd...*) que permet per mitjà d'una interfície web (és necessari tenir per exemple el servidor Apache instal·lat), configurar i afegir aspectes relacionats amb la Xarxa. Si bé encara es desenvolupa, en moltes distribucions no s'inclou per defecte. Per a executar-la, una vegada instal·lada, des d'un navegador cal cridar l'URL https://localhost:10000, que sol·licitarà l'acceptació del certificat SSL i l'usuari (inicialment *root*) i la seva clau (*passwd*).

Nota

Més informació a la pàgina de Webmin.
http://www.webmin.com/

18) System-config-*: en Fedora hi ha una varietat d'eines gràfiques que s'anomenen *system-config-"alguna-cosa"*, en què "alguna-cosa" és la funció per a la qual estan dissenyades. En general, si s'està en un entorn gràfic, es pot arribar a cada una per mitjà d'un menú; tanmateix, cada una d'aquestes eines implica un menú que cal recordar. Una eina que centralitza totes les *system config* és **system-config-control**, en una sola entrada de menú i en una única interfície gràfica, des de la qual es pot seleccionar d'acord amb una organització d'icones. Per a això, cal anar a "Aplicacions" → "Afegeix/Treu programari", i aquí s'arrenca com a *root* el gestor gràfic de programari Pirut (i s'ha de tenir habilitat el repositori Fedora Extras). En la interfície del Pirut, s'usa, per exemple, la recerca de paquets disponibles amb el patró *system-config-*;* trieu *system-config-control** i feu clic a *Apply*. Entre altres opcions, s'hi poden configurar gairebé tots els aspectes de la xarxa i els serveis.

19) Networkmanager: és una eina que permet manejar fàcilment xarxes sense fil i per cable de manera simple i sense grans complicacions, però no és indicat per a servidors (només per a màquines d'escriptori). Instal·lar-lo és molt fàcil: **apt-get install network-manager-xx**, en què *xx* és *gnome* o *kde*, segons l'escriptori instal·lat. Per a configurar-lo s'han de comentar totes les entrades a (Debian) /etc/network/interfaces, excepte la interfície de *loopback interface*, per exemple deixant només:

```
auto lo
iface lo inet loopback
```

Aquest pas no és obligatori però accelera el descobriment de les xarxes o interfícies. Sobre Debian també s'ha d'agregar un pas extra, i és que l'usuari s'ha d'integrar dins del grup *netdev* per una qüestió de permisos. Per a ferho, cal executar (com a *root* o si no amb l'ordre *sudo* per davant) **adduser usuari_actual netdev** i reiniciar (o també reiniciar la xarxa amb /etc/init.d/networking restart i sortir i tornar a entrar perquè l'usuari actual es quedi inclòs en el grup *netdev*).

20) Altres eines:

- **Nmap:** explorar i auditar amb finalitats de seguretat una xarxa.
- **Nessus:** avaluar la seguretat d'una xarxa de manera remota.
- **Wireshark** (antic Ethereal): analitzador de protocols de xarxa.
- **Snort:** sistema de detecció d'intrusos, IDS.
- **Netcat:** utilitat simple i potent per a depurar i explorar una xarxa.
- **TCPDump:** monitoratge de xarxes i adquisició d'informació.
- **Hping2:** genera i envia paquets d'ICMP/UDP/TCP per a analitzar el funcionament d'una xarxa.

Vegeu també

Algunes d'aquestes eines seran tractades en el mòdul d'administració de seguretat en l'assignatura *Administració avançada de sistemes GNU/Linux*, que tracta sobre seguretat.

Activitats

1. Definiu els escenaris de xarxa següents:

a) Màquina aïllada.
b) Petita xarxa local (4 màquines, 1 passarel·la).
c) 2 xarxes locals segmentades (2 conjunts de 2 màquines, un encaminador cada una i una passarel·la general).
d) 2 xarxes locals interconnectades (dos conjunts de 2 màquines + passarel·la cada una).
e) 2 màquines connectades per mitjà d'una xarxa privada virtual. Indiqueu els avantatges i desavantatges de cada configuració, per a quin tipus d'infraestructura són adequades i quins paràmetres rellevants es necessiten.

2. Configureu la xarxa de l'opció *a, b* i *d* del punt 1.

Bibliografia

[Bro01] Bronson, S. (2001). "VPN PPP-SSH". *The Linux Documentation Project*.

[Cis00] Cisco (2000). "TCP/IP White Paper". <http://www.cisco.com>

[Com01] Comer, D. (2001). *TCP/IP Principios básicos, protocolos y arquitectura*. Prentice Hall.

[Dra99] Drake, J. (1999). "Linux Networking". *The Linux Documentation Project*.

[Gar98] Garbee, B. (1998). *TCP/IP Tutorial*. N3EUA Inc.

[Gnu] Gnupg.org. *GnuPG Web Site*. <http://www.gnupg.org/>

[IET] IETF. "Repositori de *Request For Comment* desenvolupats per la Internet Engineering Task Force (IETF) en el Network Information Center (NIC)".

[KD00] Kirch, O.; Dawson, T. (2000). *Linux Network Administrator's Guide*. O'Reilly Associates. [I en línia (gratuït) a Free Software Foundation, Inc.]. <http://www.tldp.org/guides.html>

[Law07] Lawyer, D. (2007). "Linux Modem". *The Linux Documentation Project*.

[Log] LogCheck. <http://logcheck.org/>

[Mal96] Mallett, F. (1996). *TCP/IP Tutorial*. FAME Computer Education.

[Mou01] Mourani, G. (2001). *Securing and Optimizing Linux: The Ultimate Solution*. Open Network Architecture, Inc.

[PPP00] Williams, C.; Drake, J.; Hart, R. (2000). "Linux PPP". *The Linux Documentation Project*.

[Ran05] Ranch, D. (2005). "Linux IP Masquerade" i **Tapsell, J.** (2005). "Masquerading Made Simple". *The Linux Documentation Project*.

[Rid00] López Ridruejo, D. (2000). "The Linux Networking Overview". *The Linux Documentation Project*.

[Sec00] Seco, A. (2000). "Diald". *The Linux Documentation Project*.

[Tri] Tripwire.com. *Tripwire Web Site*. <http://www.tripwire.com/>

[Vas00] Vasudevan, A. (2000). "Modem-Dialup-NT". *The Linux Documentation Project*.

[Wil02] Wilson, M. D. (2002). "VPN". *The Linux Documentation Project*.

[Xin] Xinetd Web Site. <http://www.xinetd.org/>

Annex

Controlar els serveis vinculats a xarxa a FCx

Un aspecte important de tots els serveis és com es posen en marxa. Fcx (des de FC6) inclou una sèrie d'utilitats per a gestionar els serveis –dimonis– (incloent-hi els de xarxa). Com ja s'ha vist en l'apartat d'administració local, el *runlevel* és el mode d'operació que especifica quins dimonis s'executaran. A FC podem trobar: *runlevel* 1 (monousuari), *runlevel* 2 (multiusuari), *runlevel* 3 (multiusuari amb xarxa), *runlevel* 5 (X11 més *runlevel* 3). Típicament s'executa el nivell 5 o 3 si no es necessiten interfícies gràfiques. Per a determinar quin nivell s'està executant, es pot utilitzar */sbin/runlevel* i per a saber quin nivell és el que s'arrenca per defecte, *cat /etc/inittab | grep :initdefault:*, que ens donarà informació com *id:5:initdefault:* (també es pot editar /etc/inittab per a canviar el valor per defecte).

Per a visualitzar els serveis que s'estan executant, podem utilitzar */sbin/chkconfig -list*, i per a gestionar-los podem utilitzar **system-config-services** en mode gràfic o *ntsysv* en la línia d'ordres. Per a habilitar serveis individuals podem utilitzar *chkconfig*. Per exemple, l'ordre següent habilita el servei *crond* per als nivells 3 i 5: */sbin/chkconfig* ***--level 35 crond on.***

Independentment de com s'hagin posat en marxa els serveis, es pot utilitzar */sbin/service* ***-status-all*** o individualment amb */sbin/service crond status* per a saber com està cada servei. I també gestionar-lo (*start, stop, status, reload, restart*); per exemple, ***service crond stop*** per a parar-lo o ***service crond restart*** per a reiniciar-lo.

És important **no deshabilitar els serveis següents** (tret que se sàpiga el que s'està fent): *acpid, haldaemon, messagebus, klogd, network, syslogd*. Els serveis més importants vinculats a la xarxa (encara que no es recullen tots, sí la majoria en aquesta llista no exhaustiva) són:

- **NetworkManager, NetworkManagerDispatcher:** és un dimoni que permet canviar entre xarxes fàcilment (Wi-Fi i Ethernet bàsicament). Si només té una xarxa no és necessari que s'executi.

- **Avahi-daemon, avahi-dnsconfd:** és una implementació de zeroconf i és útil per a detectar dispositius i serveis sobre xarxes locals sense DNS (és el mateix que *mDNS)*.

- **Bluetooth, hcid, hidd, sdpd, dund, pand:** Bluetooth és una xarxa sense fil per a dispositius portàtils (*no és* Wi-Fi, 802.11). Per exemple, teclats, ratolins, telèfons, altaveus, auriculars, etc.

- **Capi, isdn:** xarxa basada en maquinari ISDN[35].

(35) En català correspon a XDSI (Xarxa de Serveis Integrats).

- **Iptables:** és el servei de tallafocs estàndard de Linux. És totalment necessari per seguretat si es té connexió a xarxa (*cable, DSL, T1*).

- **Ip6tables:** és el servei de tallafocs estàndard de Linux per a protocols i xarxes basats en IPv6.

- **Netplugd:** pot monitorar la xarxa i executar instruccions quan en canviï l'estat.

- **Netfs:** s'utilitza per a muntar automàticament sistemes d'arxius per mitjà de la xarxa (NFS, Samba, etc.) durant l'arrencada.

- **Nfs, nfslock:** són els dimonis estàndard per a compartir sistemes d'arxius per mitjà de la xarxa en sistemes operatius d'estil Unix/Linux/BSD.

- **Ntpd:** servidor d'hora i data a través de la xarxa.

- **Portmap:** és un servei complementari per a NFS (*file sharing*) o NIS (*authentication*).

- **Rpcgssd, rpcidmapd, rpcsvcgssd:** s'utilitza per a NFS v4 (nova versió d'NFS).

- **Sendmail:** aquest servei permet gestionar els correus (MTA) o donar suport a serveis com IMAP o POP3.

- **Smb:** aquest dimoni permet compartir fitxers sobre sistemes *Windows*.

- **Sshd:** SSH permet a altres usuaris connectar-se interactivament de manera segura a la màquina local.

- **Yum-updatesd:** servei d'actualitzacions per xarxa d'FC.

- **Xinetd:** servei alternatiu d'**inetd** que presenta un conjunt de característiques i millores, com per exemple llançar múltiples serveis per al mateix port (aquest servei pot no estar instal·lat per defecte).