Serveis de xarxa i Internet

CFGS.ASX.M08/0.12

Administració de sistemes informàtics en xarxa

Generalitat de Catalunya
Departament d'Educació

ioc
institut obert
de catalunya

Aquesta col·lecció ha estat dissenyada i coordinada des de l'Institut Obert de Catalunya.

Coordinació de continguts
Joan Carles Pérez Vázquez

Redacció de continguts
Eduard Canet i Ricard
Josep Ciberta Tirado
Oriol Torres Carrió

Primera edició: setembre 2013

Material realitzat per Eureca Media, SL
Dipòsit legal: B. 15213-2013

Introducció

Avui en dia, quan es pensa en un sistema informàtic rarament es pensa en un equip aïllat; ja sembla inherent a qualsevol sistema informàtic que estigui format per un conjunt d'equips connectats en xarxa. Qualsevol d'aquests equips necessitarà recursos als quals podrà accedir per mitjà de la xarxa i probablement compartirà el resultat de la seva activitat amb altres equips.

Una de les conseqüències d'aquest treball en xarxa és la necessitat de mantenir i administrar adequadament aquests sistemes i els seus serveis. Aquest mòdul aporta la base de coneixements que necessitarà qualsevol tècnic en administració de sistemes i xarxes per desenvolupar amb èxit la seva tasca.

En els cicles de grau superior d'Informàtica, l'estudi dels serveis de xarxes complementa els coneixements bàsics de xarxes i és una eina important per administrar integralment un sistema informàtic.

En la unitat "Serveis de noms i configuració automàtica" es mostra com l'administrador pot establir serveis que permeten configurar automàticament les dades de connexió dels equips i el domini al qual pertanyen. D'aquesta manera permet que els equips de la xarxa es puguin adreçar els uns als altres i a Internet. També s'explica com l'administrador pot posar en funcionament i mantenir actualitzats servidors que proporcionin diferents serveis (com el servei de transferència de fitxers, continguts web, correu, impressió...), i com es configura un equip per actuar de client i fer ús d'aquests serveis.

En la unitat "Serveis web i de transferència de fitxers" es mostra, d'una banda, com l'administrador pot posar en funcionament i mantenir actualitzats servidors web per publicar continguts a Internet, i de l'altra, com es gestionen els mètodes d'accés remot i com s'instal·len els serveis corresponents, i se'n descriuen les característiques.

En la unitat "Correu electrònic i missatgeria" es mostra com l'administrador pot posar en funcionament i mantenir actualitzats servidors que proporcionin diferents serveis (com els serveis de correu i missatgeria) i com es configura un equip per actuar de client i fer ús d'aquests serveis.

En la unitat "Serveis d'àudio i vídeo" es mostra com l'administrador pot posar en funcionament i mantenir actualitzats equips servidors que proporcionin serveis de distribució d'àudio i de vídeo. També s'explica com configurar els equips clients perquè puguin utilitzar les eines de reproducció, tant d'àudio com de vídeo i videoconferència.

Els coneixements que adquirireu en aquestes pàgines són coneixements genèrics. Per això cal fer totes les activitats que us anirà proposant al llarg del curs el professor, amb les quals adquirireu les estratègies que us permetran enfrontar-vos a les situacions que trobareu en el món real. Els coneixements teòrics són importants per poder comprendre les pràctiques que fareu: no vulgueu anar directament a les pràctiques, seguiu l'ordre que us marqui el professor.

Resultats d'aprenentatge

En finalitzar aquest mòdul l'alumne/a:

Serveis de noms i configuració automàtica

1. Administra serveis de resolució de noms, analitzant-los i garantint la seguretat del servei.
2. Administra serveis de configuració automàtica, identificant-los i verificant la correcta assignació dels paràmetres.

Serveis web i de transferència de fitxers

1. Administra servidors web aplicant criteris de configuració i assegurant el funcionament del servei.
2. Administra serveis de transferència de fitxers assegurant i limitant l'accés a la informació.

Correu electrònic i missatgeria

1. Administra servidors de correu electrònic, aplicant criteris de configuració i garantint la seguretat del servei.
2. Administra serveis de missatgeria instantània, notícies i llistes de distribució, verificant i assegurant l'accés dels usuaris.

Serveis d'àudio i vídeo

1. Administra serveis d'àudio identificant les necessitats de distribució i adaptant els formats.
2. Administra serveis de vídeo identificant les necessitats de distribució i adaptant-ne els formats.

Continguts

Serveis de noms i configuració automàtica

Unitat 1

Serveis de noms i configuració automàtica

1. Instal·lació i administració de serveis de noms de domini
2. Instal·lació i administració de serveis de configuració automàtica de xarxa

Serveis web i de transferència de fitxers

Unitat 2

Serveis web i de transferència de fitxers

1. Instal·lació i administració de servidors web
2. Instal·lació i administració de serveis de transferència de fitxers

Correu electrònic i missatgeria

Unitat 3

Correu electrònic i missatgeria

1. Instal·lació i administració del servei de correu electrònic
2. Instal·lació i administració de serveis de missatgeria instantània, notícies i llistes de distribució

Serveis d'àudio i vídeo

Unitat 4

Serveis d'àudio i vídeo

1. Instal·lació i administració del servei d'àudio
2. Instal·lació i administració del servei de vídeo

Serveis de noms i de configuració automàtica

Eduard Canet i Ricart

Índex

Introducció

En el mòdul *Serveis de xarxa i Internet* estudiarem i practicarem la instal·lació i configuració de diversos serveis de xarxa. Molts d'aquests serveis són molt coneguts i són destinats a proporcionar serveis d'Internet a l'usuari final (per això són populars). Parlem per exemple del servei web (HTTP), el de transferència de fitxers (FTP), el de correu, el d'àudio, de vídeo... Tanmateix, hi ha altres serveis que tot i ser imprescindibles a Internet són menys coneguts per als usuaris. Es tracta de serveis com DHCP, DNS, SMTP..., que, tot i ser omnipresents, no són tan coneguts perquè no van destinats a l'usuari final, sinó a la configuració de les xarxes, a fer que les xarxes funcionin correctament.

En la unitat formativa **"Instal·lació i administració de serveis de noms de domini"** s'explica com mantenir i administrar adequadament els equips en xarxa de manera automatitzada. Es mostra com l'administrador pot establir serveis que permeten configurar automàticament les dades de connexió dels equips i el domini al qual pertanyen i com es configuren els equips per actuar de clients i fer ús d'aquests serveis.

En la unitat **"Instal·lació i administració de serveis de configuració automàtica de xarxa"** s'explica el protocol DNS (Domain Name System o sistema de noms de domini), que permet la resolució de noms de domini a adreces IP i a la inversa. La "màgia" amb la qual un usuari indica un nom de domini i obté l'adreça corresponent a aquest domini és obra del DNS.

Primerament caldrà que ens familiaritzem amb el sistema de noms de domini veient-ne l'evolució des dels primers fitxers de noms plans fins a l'actual sistema jeràrquic i distribuït. Aprendreu a reconèixer i a identificar el funcionament dels noms de domini a Internet i com són gestionats per mitjà de servidors encarregats de controlar una zona concreta.

Es presenta la documentació de l'estàndard del protocol DNS i les seves diverses extensions, que permeten actualitzacions dinàmiques, multitud d'opcions de configuració, configuracions condicionals, expressions i tractament de la seguretat en les comunicacions.

El sistema de noms de domini permet identificar un domini a qualsevol lloc del món a Internet. Es mostra com es realitza aquest mecanisme de resolució, que "per art de màgia" sap identificar quina adreça IP correspon a cada domini. Aquesta tasca la realitzen els servidors de noms. Cal, doncs, instal·lar-los i posar-los en funcionament. Veureu, doncs, tot el procés necessari per posar en marxa un servidor DNS i els mecanismes de reconeixement que cal utilitzar per comprovar-ne el funcionament correcte.

Finalment cal saber definir noves zones amb informació dels equips propis de la zona. S'explicarà com definir els equips usant els registres de recurs,

com compartir aquesta informació entre diversos servidors primaris i secundaris mitjançant transferències, com delegar zones entre institucions diferents i, fins i tot, com posar en marxa servidors només cau (que no administren res, simplement agiliten les respostes).

En l'apartat "Administra serveis de configuració automàtica, identificant-los i verificant la correcta assignació dels paràmetres" s'explica el funcionament del protocol DHCP. El servei DHCP (Dynamic Host Configuration Protocol o protocol de configuració dinàmica d'equips) permet la configuració d'adreces IP, màscares, passarel·les per defecte i moltes altres opcions de configuració de manera totalment dinàmica. A cada equip, se li ha de proporcionar un identificador i la informació necessària per poder treballar en xarxa i poder accedir a altres equips i altres xarxes.

Primerament analitzem quina és la informació que ha de tenir un equip per poder treballar en xarxa i disposar d'accés a Internet. Aquesta configuració es pot establir manualment o de manera dinàmica; s'analitzen els pros i contres de cada cas. El creixement que han sofert les xarxes a escala mundial (tant en nombre de xarxes com d'equips i de complexitat de gestió) va propiciar el sorgiment del protocol DHCP. Se n'estudia l'origen, basat en el protocol BOOTP, l'evolució i el funcionament. Així, doncs, descriurem el clàssic diàleg DHCP d'intercanvi de quatre missatges.

Un cop es coneix la finalitat del protocol DHCP i el seu funcionament, cal implementar-ne un servidor. Estudiarem tots els passos necessaris per fer-ho i tots els mecanismes de reconeixement i monitoratge per comprovar que la instal·lació i posada en servei s'han fet correctament.

Finalment cal estudiar les opcions de configuració generals del servei i fer un repàs a les opcions de xarxa més usuals per als clients.

També veurem la configuració d'un client DHCP i les opcions que es poden definir directament en el client.

Els dos temes tractats en aquesta unitat, tot i que relacionats, són absolutament independents l'un de l'altre. Us recomanem fer una primera lectura global del servei DNS i en una segona lectura anar practicant in situ en un servidor els passos que es van descrivint. Aquest procés pràctic es pot ampliar al mateix temps seguint els apunts i les activitats contingudes en el material web. El mateix procediment es pot aplicar per aprendre el funcionament del servei DHCP i per practicar la configuració d'un servidor.

Resultats d'aprenentatge

En acabar aquesta unitat, l'alumne:

1. Administra serveis de resolució de noms, analitzant-los i garantint la seguretat del servei.

- Identifica i descriu escenaris en els quals sorgeix la necessitat d'un servei de resolució de noms.
- Classifica els principals mecanismes de resolució de noms.
- Descriu l'estructura, la nomenclatura i la funcionalitat dels sistemes de noms jeràrquics.
- Instal·la i configura serveis jeràrquics de resolució de noms.
- Prepara el servei per reexpedir consultes de recursos externs a un altre servidor de noms.
- Prepara el servei per emmagatzemar i distribuir les respostes procedents d'altres servidors.
- Afegeix registres de noms corresponents a una zona nova, amb opcions relatives a servidors de correu i àlies.
- Implementa solucions de servidors de noms en adreces IP dinàmiques.
- Realitza transferències de zona entre dos o més servidors.
- Documenta els procediments d'instal·lació i configuració.

2. Administra serveis de configuració automàtica, identificant-los i verificant la correcta assignació dels paràmetres.

- Reconeix els mecanismes automatitzats de configuració dels paràmetres de xarxa i els avantatges que proporcionen.
- Il·lustra els procediments i les pautes que intervenen en una sol·licitud de configuració dels paràmetres de xarxa.
- Instal·la servidors de configuració dels paràmetres de xarxa.
- Prepara el servei per assignar la configuració bàsica als equips d'una xarxa local.
- Configura assignacions estàtiques i dinàmiques.
- Integra en el servei opcions addicionals de configuració.
- Documenta els procediments realitzats.

1. Instal·lació i administració de serveis de noms de domini

El **sistema de noms de domini** o **DNS** (Domain Name System) proporciona un mecanisme eficaç per fer la resolució de noms de domini a adreces IP. Com a usuaris (humans) ens és més fàcil adreçar-nos a un nom de domini (de *host*, de web, de servidor de correu...) utilitzant un text identificatiu com per exemple www.ioc.cat que no pas l'adreça IP 213.73.40.230. El servei DNS no solament permet fer la resolució de noms de domini a adreces IP, sinó també la resolució inversa. És a dir, a partir d'una adreça IP esbrinar el nom de domini del *host*.

El servei DNS proporciona independència del nom de domini respecte a l'adreça IP. Així un domini pot canviar d'adreça IP de manera transparent per als usuaris del domini. Fins i tot és usual que un domini s'identifiqui amb més d'una adreça IP com a mesura de redundància contra la caiguda del sistema o com a balanceig de càrregues. Altres serveis proporcionats pel DNS són la identificació dels servidors de correu d'un domini, de cada un dels *hosts* que pertanyen a la xarxa, servidors d'impressió...

1.1 El servei de resolució de noms

El problema d'identificar els equips es produeix des de bon principi de l'existència de les xarxes d'ordinadors i no és específic de les xarxes TCP/IP. Cal un mecanisme en "llenguatge humà" per identificar els equips de la xarxa. En especial els que proporcionen serveis als altres equips i usuaris. En la xarxa inicial Arpanet, els equips ja rebien un nom. Aquests noms es feien públics per mitjà d'un fitxer centralitzat que contenia els noms de tots els equips de la xarxa i la seva identificació. Aquest fitxer era hosts.txt, conegut en sistemes GNU/Linux com a /etc/hosts.

> Un **sistema de noms pla** es basa en la utilització d'un **fitxer de text** que descriu cada *host* amb la seva corresponent adreça IP. Es pot usar per definir àlies per equips locals en xarxes petites, però no és escalable a xarxes grans, i molt menys a Internet.

En una xarxa petita es pot generar un fitxer amb el nom i l'adreça IP de tots els *hosts* centralitzat en un servidor, i encarregar-se de distribuir còpies d'aquest fitxer a tots els equips de la xarxa. Però aquest model de coneixement no és escalable. Si la xarxa creix és impossible de mantenir. Utilitzar aquest model significaria que hi ha un equip que centralitza els noms de tots els *hosts* d'Internet en un sol fitxer! D'altra banda, també significaria que aquest fitxer s'ha de repartir entre tots els equips d'Internet perquè sàpiguen com es diuen els altres equips cada cop que hi ha una actualització. Evidentment cal una altra solució.

El 1983 sorgeix el Domain Name System (DNS) per aportar una solució escalable i pràctica. El DNS es fonamenta en una base de dades de noms de domini jeràrquica i distribuïda. És **jeràrquica** perquè s'organitza en una estructura de **dominis** que es poden compondre de subdominis que també es poden dividir en subdominis i així fins a 127 nivells (originàriament). Aquests dominis són gestionats per servidors DNS responsables de cada **zona**. I és una base de dades **distribuïda** perquè la informació no està tota junta en un sol repositori central, sinó que es troba repartida per parts en els servidors DNS d'Internet. Cada servidor DNS **autoritari** conté la base de dades de la seva zona.

1.1.1 Classificació dels mecanismes de resolució

Els administradors de xarxes tenen la tasca d'establir el mecanisme d'identificació de *hosts* que volen usar. Determinar quins noms usaran i com es farà per identificar cada nom amb l'adreça IP corresponent. Els mecanismes per anomenar els *hosts* pot ser local a cada *host*, local i intern a una organització i global a Internet.

Un cop posats els noms cal saber-los resoldre, trobar-ne l'adreça IP apropiada. La resolució pot ser local en un *host* usant un fitxer d'associacions o implementada usant el servei de noms DNS. Aquest servei es basa en l'estructura client/servidor, de manera que caldrà aprendre a configurar tant l'un com l'altre. Primerament es descriurà com configurar el client o *resolver* i el servidor serà tractat al llarg dels apartats posteriors. Aquests dos mecanismes, local i DNS, es poden combinar i determinar-ne la precedència.

El servei DNS proporciona múltiples maneres de treballar. Segons quina sigui la seva configuració actuarà de manera diferent en fer la resolució. Caldrà estudiar què és un servidor només cau, què fa quan es permet la utilització de la memòria cau, la utilització d'un forwarder i quina diferència hi ha entre usar recursió o no utilitzar-la. La gestió de dominis i zones pròpies, la creació de subdominis i la delegació són aspectes clau del funcionament d'un servidor de noms de domini que també es tractaran més endavant.

1.1.2 Noms de 'host' locals, dominis locals i dominis d'Internet

Sabem que els *hosts* s'identifiquen en una xarxa i a Internet per la seva adreça IP, però en general els usuaris desconeixen quina és aquesta adreça. Els usuaris estan habituats a connectar-se per exemple a un altre dels ordinadors de casa seva posant un *nom local* que s'han inventat (per exemple, pcJocs, portatilMarta...).

A la feina, els mateixos usuaris accedeixen a diversos equips que tenen noms que els hi han posat els administradors del sistema. Així, per exemple, els informes estan en un ordinador anomenat *watergate*, la gestió de la comptabilitat en un

servidor que es diu *blackhole* i les nòmines es gestionen des del servidor *minix*. Tots aquests ordinadors pertanyen a la xarxa de l'empresa, que s'identifica amb el nom de domini local empresa.cat.

A més a més, resulta que tant des de casa com des de la feina aquests usuaris treballen habitualment consultant serveis a *hosts* com gmail.com, youtube.com, ara.cat... És a dir, consulten serveis i *hosts* de dominis d'Internet.

Aquests tres casos exposats permeten observar tres tipus diferents de resolució de noms:

- Resolució de noms de *host* locals
- Resolució de noms d'un domini local (no integrat a Internet)
- Resolució de noms global (domini integrat a Internet)

Les tres resolucions no són excloents, sinó que s'implementen totes a la vegada combinant la resolució local i la resolució via DNS. Existeix també un mecanisme per indicar la precedència de la resolució, és a dir, indicar si es prefereix primer la resolució local i després la de DNS o a l'inrevés.

1.1.3 El client DNS: 'resolver'

Un equip client que vol resoldre un nom de *host* té diferents maneres de fer-ho. Es pot fer localment mitjançant un fitxer de *hosts* (típicament /etc/hosts) o de manera distribuïda usant DNS (el *resolver*). De fet, es poden aplicar tots dos mètodes conjuntament indicant-ne la precedència en algun fitxer de configuració del sistema (en sistemes GNU/Linux, el fitxer /etc/ nsswitch.conf).

El *resolver* és la part client del sistema de noms de dominis DNS, que està organitzat en una estructura client/servidor. Cada *resolver* implementa les seves opcions, però n'hi ha que són suficientment genèriques per descriure-les aquí. En la majoria de sistemes GNU/Linux, aquestes opcions es defineixen en el fitxer /etc/resolv.conf.

Criteri de resolució

Per defecte, quan cal resoldre un nom de *host* (i no s'ha especificat la directiva *search*), el *resolver* fa el següent: si el nom de *host* inclou un punt (pc30.inf) mira de resoldre'l tal qual, i si no pot hi aplica el nom de domini (pc30.inf.inf.fpoberta.net.). Si el nom de *host* no conté cap punt (pc30), primer li afegeix el domini i el mira de resoldre (pc30.inf.fpoberta.net.), i si no el troba el mira de resoldre tal qual (pc30).

Les següents són les directives del fitxer /etc/resolv.conf:

- **domain** (*local domain name* o nom de domini local) indica el nom de domini del *host* al qual pertany el *resolver*. Serveix per completar els noms de domini no qualificats (FQDN).

- **search** permet modificar el comportament per defecte indicant explícitament la llista de dominis a aplicar. El primer d'aquests és aplicat com el nom del domini local (*local domain name*) i és per això que la directiva *search* és excloent de la directiva *domain*.

- **nameserver** permet especificar el servidor de noms a utilitzar. Se'n poden indicar fins a tres per si no hi ha accés al servidor. El *resolver* intenta connectar amb el primer servidor i si ho aconsegueix realitza les consultes a aquest servidor.

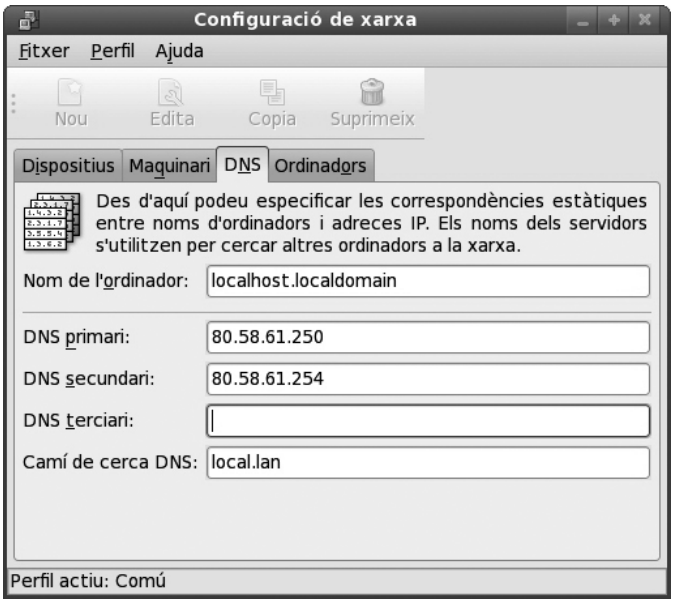
Servidor de noms d'una altra organització

Es poden configurar els *hosts* perquè utilitzin el servidor de noms d'una altra organització (perquè és més ràpida, per estalviar-se feina...), però no és una bona pràctica.

```
# Exemple de fitxer de configuració client /etc/resolv.conf
[root@host ~]$ cat /etc/resolv.conf
search ioc.cat
nameserver 80.58.61.250
nameserver 80.58.61.254
```

La figura 1.1 mostra una interfície gràfica que permet configurar els servidors DNS. En aquest exemple s'observa que s'han definit dos servidors de noms, un de primari i un de secundari. La directiva *search* correspon al valor local.lan. Tal com és habitual en sistemes GNU/Linux aquesta miniaplicació gràfica (*applet*) simplement modifica el fitxer de configuració.

FIGURA 1.1. Miniaplicació gràfica de configuració del resolver



1.2 Funcionalitat del sistema de noms jeràrquics

Sabem que tenim la necessitat de poder-nos adreçar als diversos *hosts*, servidors i dominis que hi ha a les xarxes, i en especial a Internet. Hem d'establir algun mètode per relacionar les adreces IP amb el seu nom corresponent. Un mecanisme és posar noms plans als *hosts* i als dominis, noms independents els uns dels altres sense cap mena d'estructura ni de jerarquia. Amb tota seguretat aquest mecanisme provocaria noms duplicats i un caos organitzatiu important.

El sistema de noms de domini permet identificar qualsevol equip en la xarxa i assegurar-se que no hi ha col·lisions, és a dir, noms duplicats. Es basa en una estructura jeràrquica de noms en forma d'arbre on l'arrel és el node o domini arrel del qual deriven tots els altres nodes. Aquest es divideix en altres dominis com, per exemple, .com, .edu, .org, .cat... Al seu torn, cada domini es pot dividir en

subdominis i així successivament. Les rutes s'indiquen començant pel subdomini més intern i anant cap al node arrel, com per exemple mail.ioc.cat.

1.2.1 El sistema de noms jeràrquic

En un sistema de noms jeràrquic cada node de l'arbre s'identifica per un text (el nom de domini) que no es pot repetir en el mateix nivell, però sí en altres llocs de l'arbre de l'espai de noms. El mateix passa amb els fitxers: no hi pot haver dos fitxers amb el mateix nom en el mateix directori, però sí en ubicacions diferents. Un **domini** està format pel propi node i la resta de l'arbre que penja d'aquest node. Penseu en l'exemple d'un directori: si es vol copiar un directori s'entén que està format pel mateix directori i tots els subdirectoris que conté.

Caràcters en els noms de domini

L'estàndard DNS indica que els noms de domini han de ser de 64 caràcters com a màxim, i només poden incloure caràcters llatins, dígits del 0 al 9 i el guió. Les majúscules són indiferents.

Hi ha mecanismes com l'IDNA (Internationalized Domain Name o noms de domini internacionalitzats) que permeten utilitzar altres alfabets en els noms de domini.

Anomenem **espai de noms** al conjunt de tots els dominis que formen l'arbre de noms DNS. Continuant l'analogia amb el sistema de fitxers diríem que l'espai de noms és equivalent a tot el sistema de fitxers i directoris.

El sistema de noms de domini d'Internet DNS està format pels elements següents:

- **Espai de noms**: el conjunt de tots els noms de domini d'Internet (tot l'arbre).
- **Domini**: text identificatiu d'un domini (un node i tots els seus descendents).
- **FQDN**: nom de domini absolut començant pel node i acabant en l'arrel.
- **Domini absolut**: és un sinònim d'FQDN. Es la ruta sencera que va del node a l'arrel. Els dominis absoluts acaben en punt.
- **Domini relatiu**: nom de domini sense qualificar (no acaba en punt). S'entén que un nom de domini és relatiu al domini al que pertany.
- **Domini arrel**: domini del qual deriven tots els altres. S'indica amb un punt o amb la cadena buida.

Exemples de noms de domini:

- ioc.cat.: nom de domini absolut que, per exemple, inclou mail.ioc.cat i inf.ioc.cat.
- pc01.inf.ioc.cat.: nom de *host* absolut o FQDN.
- pc01: nom de *host* relatiu al domini on pertanyi.
- pc02.inf: nom de domini relatiu al domini on pertany.
- com: nom de domini relatiu. Usualment ens hi referim en lloc de com., que és el FQDN apropiat.
- .: domini arrel.

El punt final en els noms de domini

La majoria de vegades escrivim els dominis com si fossin absoluts, però són relatius al node arrel perquè no posem el punt final. Un altre cop es pot fer l'analogia amb les rutes relatives i les rutes absolutes del sistema de fitxers.

L'estructura d'arbre (jeràrquica) de l'espai de noms proporciona un mecanisme d'identificació únic d'un domini. No pot existir cap domini que tingui exactament el mateix nom absolut o **FQDN** (Fully Qualified Domain Name o nom de domini complet). Els dominis es llegeixen des del node a l'arrel. Així, un domini que correspongui al departament d'administració de l'organització IOC dins del domini cat s'identifica, per exemple, com a admin.ioc.cat. Si ens fixem en el domini anterior, veurem que acaba en punt: és una manera d'indicar el domini arrel. El **domini arrel** es defineix com un domini sense etiqueta o, millor dit, amb la cadena buida com a etiqueta. Això provoca que els dominis que s'indiquin de manera absoluta acabin amb el caràcter punt.

Un **domini absolut** o FQDN és el que inclou tots els nodes des del domini fins a l'arrel (inclosa en forma de punt final). Un **domini relatiu** no inclou l'arrel i pot ser relatiu al domini actual. Per exemple, dins del domini de l'IOC el domini inf (del departament d'informàtica) és un nom relatiu que fa referència al nom absolut inf.ioc.cat.

1.2.2 Els noms de domini d'Internet

Origen dels noms de domini

Si ens fixem en els primers dominis d'alt nivell, estaven basats en una visió estatunidenca del món (de fet la xarxa Arpanet, base de l'actual Internet, va ser desenvolupada pel Departament de Defensa del govern dels EUA). En estendre's Internet globalment i aparèixer dominis d'alt nivell geogràfics, moltes organitzacions es van registrar en més d'un domini (per exemple, empresa.com i empresa.cat).

A Internet els noms de dominis segueixen una estructura basada en els seus inicis però que ha anat evolucionant. El node arrel es va dividir en un conjunt de subdominis anomenats **TLD** (Top Level Domains o dominis d'alt nivell). Aquests dominis eren com, edu, gov, mil, org, net i int. Posteriorment se'n van afegir d'altres com cat, name, biz, info, pro, aero, coop i museum. Es volien organitzar els dominis per funcionalitat posant les empreses en els .com, les organitzacions en els .org...

Es va veure, però, la necessitat de poder agrupar els dominis de manera geogràfica i van sorgit els famosos identificadors de país. Per a cada país es va generar un TLD de dos caràcters utilitzant el preexistent estàndard internacional ISO 3166 (els famosos .es, .fr, .us...).

Degut a aquesta doble nomenclatura, els primers es coneixen com **gTLD** (domini de primer nivell genèric, en anglès *generic top-level domain*) i els segons corresponents a països com a **ccTLD** (domini de primer nivell territorial, en anglès *country-code top-level domain*).

Els servidors arrel són crucials per al funcionament del DNS, ja que coneixen tots els dominis de primer nivell. Han d'admetre un gran volum de consultes i per això n'hi ha tretze repartits per tot el món. A més a més, d'aquests tretze, alguns tenen rèpliques en diversos continents utilitzant un sistema anomenat *anycast*.

Dominis d'alt nivell

Els següents són exemples de dominis d'alt nivell:

- cat: Catalunya
- ad: Andorra

- aq: Antàrtida
- gb: Gran Bretanya
- im: Illa de Man
- ms: Montserrat
- pf: Polinèsia Francesa
- ps: autoritat palestina
- uk: Regne Unit

Així doncs, hi ha un node arrel del qual deriven múltiples nodes de primer nivell, com per exemple com., cat., es., org. Aquests dominis són gestionats per institucions o empreses amb forts lligams amb la indústria informàtica i Internet. Després hi ha els dominis de segon nivell, com per exemple ioc.cat., gmail.com., rediris.es. o escoladeltreball.org., que corresponen a empreses o institucions que han demanat disposar d'un domini propi de segon nivell. Això es fa demanant donar-se d'alta al gestor apropiat del domini de primer nivell del qual es vol formar part. Aquests serveis són de pagament en la gran majoria dels casos.

El model es va repetint de manera que cada gestor d'un domini pot crear (o vendre) subdominis del seu domini. Així, per exemple, si els gestors del domini imaginari *jocs.org.* permeten fer subdominis, es podria crear el subdomini *parxis.jocs.org.* des d'on divulgar la nostra afició al parxís.

A vegades els dominis es classifiquen per nivells, indicant el seu grau de profunditat:

- **Arrel**: el domini pare de tots els dominis, el punt.
- **TLD** o **primer nivell**: domini fill de l'arrel o de primer nivell. En són exemples cat., es., com., org. ...
- **Segon nivell**: format pels dominis fills dels dominis de primer nivell. Per exemple, ioc.cat., gencat.cat., gmail.com., python.org. ...
- **Altres**: a partir d'aquí cada domini de segon nivell genera els subdominis que creu apropiats. Alguns són gestionats per ells mateixos i d'altres són delegats a altres entitats. Per exemple, correu.ioc.cat., ensenyament.gencat.cat.

Els dominis inclouen *hosts*, impressores, servidors de correu... És a dir, noms d'una màquina. Tot nom de *host* és també un nom que es podrà identificar i resoldre usant el DNS. Els noms de *host* pertanyen a un domini, però no són dominis. Sovint els usuaris desconeixen la diferència entre un nom de *host* (un element) i un nom de domini (una àrea que abasta subdominis i que conté *hosts*). Veurem més endavant la relació entre els dominis i les zones, que són les bases de dades que descriuen els *hosts* que formen part del domini.

Per exemple, gmail.com. és un domini, però www.gmail.com. és un *host* (o més d'un en cas de ser *multihomed*). El mateix passa amb inf.ioc.cat., que fa referència al domini format per tots els *hosts* del departament d'informàtica de l'IOC i els possibles subdominis que tingui. En canvi, pc01.inf.ioc.cat., printer.inf.ioc.cat. o ftp.inf.ioc.cat. són noms de *hosts* que pertanyen a aquest domini.

Molt sovint a Internet es confon entre un **nom de *host*** com ftp.rediris.es., que identifica la màquina de RedIRIS que proporciona el servei FTP, i el **nom de domini**, que identifica una àrea de l'espai de noms de domini que inclou els seus subdominis.
Els **dominis** contenen descripcions dels *hosts* i la seva organització.

Alguns exemples són:

- www.gmail.com. és un nom de *host*.
- gmail.com. és un nom de domini.
- www.ioc.cat. és un nom de *host*.
- ioc.cat. és un nom de domini.

```
# Llistat de l'adreça IP d'un host concret ("eines") del domini ioc.cat
root@server:~# host eines.ioc.cat
eines.ioc.cat has address 85.192.111.246

# Llistat d'informació global del domini ioc.cat
root@server:~# host ioc.cat
ioc.cat has address 85.192.111.254
ioc.cat mail is handled by 10 aspmx.l.google.com.
ioc.cat mail is handled by 20 alt2.aspmx.l.google.com.
ioc.cat mail is handled by 30 aspmx4.googlemail.com.
ioc.cat mail is handled by 30 aspmx5.googlemail.com.
ioc.cat mail is handled by 20 alt1.aspmx.l.google.com.
ioc.cat mail is handled by 30 aspmx3.googlemail.com.
ioc.cat mail is handled by 30 aspmx2.googlemail.com.
```

1.2.3 Dominis, subdominis i zones

Exemple d'administració de subdomini

El domini .cat és administrat per una entitat que gestiona la zona .cat. Aquest domini conté el subdomini ioc.cat, però ha **delegat** l'administració d'aquest subdomini a l'IOC. Els administradors de l'IOC disposen d'un servidor que gestiona el seu domini com una zona. El domini .cat és l'arbre que inclou tots els dominis que en deriven, inclòs ioc.cat. Però la zona .cat i la zona ioc.cat no són la mateixa zona. Són administrades per entitats diferents.

Sabem que el sistema de noms de domini està basat en una arquitectura client/servidor en què els clients fan preguntes del tipus "Quina IP té aquest domini?" i els servidors miren de contestar-les. Els servidors de noms DNS són els programes que emmagatzemen i gestionen la informació de la base de dades d'una part de l'espai de noms anomenada *zona*.

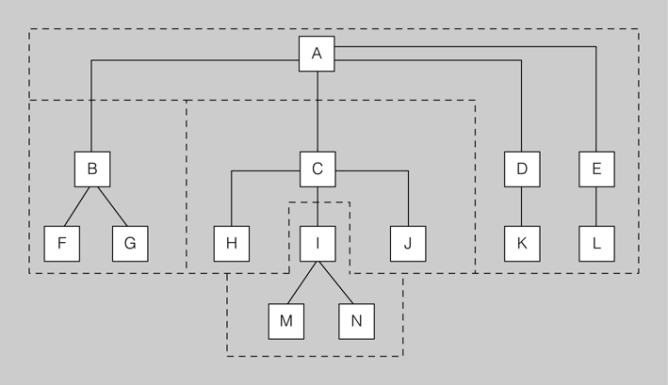
Primerament hem de descriure què és una zona. Una **zona** és la part de l'espai de noms de domini gestionada per un o més servidors DNS. Els servidors que gestionen la zona tenen informació completa sobre ella i es diu que hi tenen **autoritat**. Podríem pensar que un servidor DNS gestiona un domini i que una zona és el mateix que un domini, però això no és necessàriament així. Un domini

es divideix en subdominis per facilitar-ne l'administració. Cada part administrada per un o més servidors DNS és una zona. El domini és l'arbre de l'espai de noms (ell i els seus descendents) i la zona és la part de l'arbre administrada per un servidor de noms de domini concret.

En la figura 1.2 es pot veure un espai de noms amb quatre zones i catorze dominis. Cada lletra és un domini. El nom de domini corresponent a cada zona (s'anomenen segons el seu node superior) és A, B.A, C.A i I.C.A respectivament. Cada una d'aquestes quatre zones tindrà un o més servidors DNS per gestionar-la.

En general, podem dir que una zona conté la informació completa dels equips que formen el domini corresponent a la zona i dels equips dels subdominis que no s'han delegat. Aquesta informació s'emmagatzema en la **base de dades de zona**.

Figura 1.2. Exemple de zones i dominis



Convé tenir clar en tot moment que domini i zona no són equivalents (tot i que poden coincidir).

- El **domini** és l'arbre de l'espai de noms que inclou el node i els seus descendents.
- La **zona** és la part de l'arbre administrada per un servidor DNS concret.
- La **base de dades de zona** la formen els fitxers que emmagatzemen la descripció dels equips que pertanyen a la zona.
- La **delegació** consisteix a passar l'autoritat de la gestió d'un subdomini a una altra entitat. Aquesta serà qui s'encarregarà de gestionar-lo.

Delegar l'administració d'un subdomini no és més que traspassar l'autoritat sobre aquest subdomini a una altra entitat (a uns altres servidors DNS). Aquesta entitat és la responsable de l'administració de la zona delegada. Té tota l'autoritat per fer i desfer al seu criteri. La zona pare perd el control administratiu de la zona delegada i simplement apunta als servidors de noms de la zona delegada per obtenir informació quan la requereix.

L'estàndard que defineix el DNS estableix que cal configurar dos o més servidors autoritaris per a cada zona, anomenats *servidor primari* i *servidor secundari*. El motiu és proporcionar un mecanisme de redundància, robustesa, rendiment i còpia de seguretat. Si el servidor de noms falla i és únic, possiblement la xarxa caurà, serà inoperativa.

Els servidors **primari** (o *master*) i **secundari/s** (o *slave/s*) són autoritat. Només el primari té els fitxers de zona amb les dades in situ. Els servidors secundaris obtenen una còpia de les dades per transferència.

1.2.4 El protocol DNS

El servei de noms de domini utilitza el protocol DNS per fer les consultes i les respostes. Es tracta d'un protocol de capa d'aplicació que pot utilitzar tant UDP (*User Datagram Protocol*) com TCP (*Transmission Control Protocol*) en la capa de transport. Usualment, tant les consultes del client com les respostes del servidor es poden encabir en un datagrama (512 *bytes*) i s'utilitza UDP (de fet, generalment es diu que el DNS usa UDP). Però si la informació a transmetre és àmplia (per exemple, una resposta amb una llista amb molta informació), la comunicació es passa automàticament a TCP. Un altre cas en què la informació usa TCP és quan es realitza la transferència d'informació d'una zona entre servidors primaris i secundaris. El servidor DNS utilitza el port ben conegut (*well known*) 53.

El **protocol DNS** usu habitualment UDP, però pot usar **TCP i UDP**. Es tracta d'un protocol de capa d'aplicació i utilitza el **port 53**.

Els datagrames DNS es componen de diversos apartats, tal com es pot veure en la consulta *host* següent:

```
root@server:~# host -a uoc.edu
Trying "uoc.edu"
Trying "uoc.edu"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39287
;; flags: qr rd ra; QUERY: 1, ANSWER: 16, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;uoc.edu.     IN  ANY

;; ANSWER SECTION:
uoc.edu.    3600  IN  TXT "google-site-verification=
    GrOoO_KKCaNNqkn4y6c3P_AnaoSczfz_mKBeSo4DKZ8"
uoc.edu.    3600  IN  TXT "google-site-verification=
    GxX8kwaNhmRhDUaZrmgdOALPvdor5iFnIorJBGYcEcw"
uoc.edu.    3600  IN  TXT "v=spf1 ip4:213.73.40.0/24 ip4:13.111.22.169 ip4
    :83.169.91.142 ip4:83.169.91.143 ip4:83.169.91.144 ip4:83.169.91.145 ip4
    :83.169.91.146 ip4:83.169.91.147 include:_spf.google.com include:eevid.com
     ~all"
uoc.edu.    3600  IN  TXT "adobe-idp-site-verification=f6f2f671-9e0b
    -4464-9293-5c71951acbc1"
uoc.edu.    86400 IN  NS  nepal.uoc.es.
uoc.edu.    86400 IN  NS  tibet.uoc.es.
```

```
17 uoc.edu.    86400 IN  SOA tibet.uoc.es. root.tibet.uoc.es. 2020063001 286400
       7200 2592000 172800
18 uoc.edu.    600 IN  A 213.73.40.242
19 uoc.edu.    300 IN  NAPTR 0 30 "S" "SIP+D2U" "" _sip\._udp\.uoc\.edu.
20 uoc.edu.    300 IN  MX  5 alt2.aspmx.l.google.com.
21 uoc.edu.    300 IN  MX  10 aspmx2.googlemail.com.
22 uoc.edu.    300 IN  MX  10 aspmx3.googlemail.com.
23 uoc.edu.    300 IN  MX  10 aspmx4.googlemail.com.
24 uoc.edu.    300 IN  MX  10 aspmx5.googlemail.com.
25 uoc.edu.    300 IN  MX  1 aspmx.l.google.com.
26 uoc.edu.    300 IN  MX  5 alt1.aspmx.l.google.com.
27
28 ;; ADDITIONAL SECTION:
29 tibet.uoc.es.   86291 IN  A 213.73.40.45
30
31 Received 819 bytes from 192.168.1.1#53 in 6438 ms
```

La comunicació DNS és un mecanisme de consulta/resposta entre el client i el servidor. Els datagrames, doncs, seran de *query* (consulta) o *answer* (resposta).

Els apartats que componen un missatge DNS són:

- **HEADER**. Capçalera del missatge que indica si és una consulta o una resposta. Conté l'ID (identificador) del missatge, *flags* i un resum de quines seccions del missatge porten informació i quanta.
- **QUESTION**. Aquesta secció conté la consulta que s'ha efectuat. És a dir, quina dada s'ha demanat al servidor. Pot ser una resolució d'adreça IP a un domini, demanar la llista de servidors de correu...
- **ANSWER**. Secció que conté la resposta obtinguda del servidor. S'entén que aquesta secció conté la resposta no autoritativa. A vegades en les utilitats de consulta aquesta secció es mostra com a *non-authority answer*.
- **AUTHORITY**. Aquesta secció conté les respostes que són autoritatives per a la consulta efectuada. Evidentment pot estar buida.
- **ADDITIONAL**. Conté informació addicional per completar la resposta. En l'exemple s'observa que completa la resolució dels noms de màquina que hi ha a la secció *answer* tot indicant la seva adreça IP corresponent.

1.3 Instal·lació i configuració del servei DNS

El servei de xarxa DNS està estructurat en forma de servei client/servidor; per tant, caldrà disposar del programari apropiat per adoptar cada un d'aquests rols. El programari que fa la funció de client usualment ja està integrat en el sistema operatiu (la part que gestiona la xarxa) o en les mateixes aplicacions (per exemple, Firefox). És a dir, per disposar de la part client del servei DNS normalment no cal instal·lar res, tot i que sí que cal configurar-la correctament.

Així, doncs, quan parlem d'instal·lar un servei DNS fem referència al procés d'instal·lació i configuració del programari del servidor.

La instal·lació del programari que proporciona el servei DNS es fa de manera molt similar (per no dir idèntica) a la d'altres serveis de xarxa com el DHCP, l'HTTP o l'FTP. Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada.

Per fer tot això cal fer les reflexions i passos següents:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei DNS.
- Observar l'estat de la xarxa actual. Està el servei ja en funcionament? Existeix ja una configuració DNS activa?
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i activar els clients perquè la utilitzin.
- Comprovar que el servei funciona correctament.

1.3.1 Aplicacions servidor DNS

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha en el mercat que ofereixen aquest servei. És feina seva estudiar les característiques de les diverses aplicacions, com per exemple avaluar-ne l'eficiència, el cost, el que en diuen els altres... La manera més fàcil de fer això és navegar per Internet, consultar les revistes especialitzades o demanar consell a un dels gurus informàtics coneguts.

Cerca de DNS a Internet

Usualment l'administrador s'informa mitjançant el seu cercador preferit, per exemple Google, i de webs com la Viquipèdia. Proveu a buscar "DNS" o "DNS server" al Google i a la Wikipedia (en anglès).

Usualment, però, l'administrador acaba utilitzant l'aplicació servidor DNS que li proporciona el mateix sistema operatiu. Si utilitzeu Windows, l'empresa Microsoft disposa d'una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor DNS o bé n'existeix algun de clàssic provinent de l'Unix. De totes maneres, també en podeu obtenir d'altres a Internet. En la figura 1.3 es pot veure quin servidor utilitzen els nodes arrels. La llista ha estat extreta de la Wikipedia. S'hi poden observar els 13 servidors o nodes arrel del servei DNS, l'entitat que els gestiona, el tipus de difusió que fan i el programari que utilitzen.

FIGURA 1.3. Llista de servidors arrel DNS i programari que utilitzen

Letter	IPv4 address	IPv6 address	Old name	Operator	Location	Software
A	198.41.0.4	2001:503:BA3E::2:30	ns.internic.net	VeriSign	distributed using anycast	BIND
B	192.228.79.201	2001:478:65::53	ns1.isi.edu	USC-ISI	Marina Del Rey, California, U.S.	BIND
C	192.33.4.12		c.psi.net	Cogent Communications	distributed using anycast	BIND
D	128.8.10.90		terp.umd.edu	University of Maryland	College Park, Maryland, U.S.	BIND
E	192.203.230.10		ns.nasa.gov	NASA	Mountain View, California, U.S.	BIND
F	192.5.5.241	2001:500:2f::f	ns.isc.org	Internet Systems Consortium	distributed using anycast	BIND 9[3]
G	192.112.36.4		ns.nic.ddn.mil	Defense Information Systems Agency	distributed using anycast	BIND
H	128.63.2.53	2001:500:1::803f:235	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, U.S.	NSD
I	192.36.148.17	2001:7fe::53 (testing)	nic.nordu.net	Autonomica	distributed using anycast	BIND
J	192.58.128.30	2001:503:C27::2:30		VeriSign	distributed using anycast	BIND
K	193.0.14.129	2001:7fd::1		RIPE NCC	distributed using anycast	NSD[4]
L	199.7.83.42 (since November 2007; originally was 198.32.64.12)[5]	2001:500:3::42		ICANN	distributed using anycast	NSD[6]
M	202.12.27.33	2001:dc3::35		WIDE Project	distributed using anycast	BIND

1.3.2 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per Internet paquets del client i del servidor DHCP usant eines com *apt-get* o *yum* i els repositoris de paquets apropiats segons quina sigui la distribució que utilitzin. A més, sempre es pot utilitzar algun cercador d'Internet per ajudar a localitzar tot allò que faci falta.

Un cop instal·lat el programari caldrà identificar què s'ha instal·lat. Quins paquets i què contenien. A vegades no s'instal·laran paquets sinó fitxers .tar, dels quals també caldrà saber-ne examinar el contingut. És important saber identificar quins dels components instal·lats corresponen a fitxers executables, quins a fitxers de configuració i quins a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant, caldrà saber gestionar l'estat del servei (engegat, aturat...) i definir l'estat que ha de tenir en els diferents *runlevels* del sistema.

En definitiva, el procediment d'instal·lar usualment inclourà:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo utilitzant l'eina apropiada segons quina sigui la distribució.
- Examinar el sistema per identificar quin programari i quins paquets relacionats amb el servei hi ha instal·lats.
- Identificar els components del servei: quins són els fitxers executables, de configuració i de documentació.

- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

1.3.3 Configuració per defecte del servei instal·lat

El servei DNS disposa usualment d'una configuració bàsica per defecte en instal·lar-se que acostuma a ser l'apropiada per a un servidor de noms només cau. A vegades simplement ve amb els fitxers de configuració buits, de manera que cal editar-los apropiadament abans de posar el servei en funcionament. En la configuració d'aquest servei s'usa el programari BIND (*Berkeley Internet Name Domain*) perquè és un dels més usats a nivell global. Cal identificar cadascun dels conceptes que descrits a continuació:

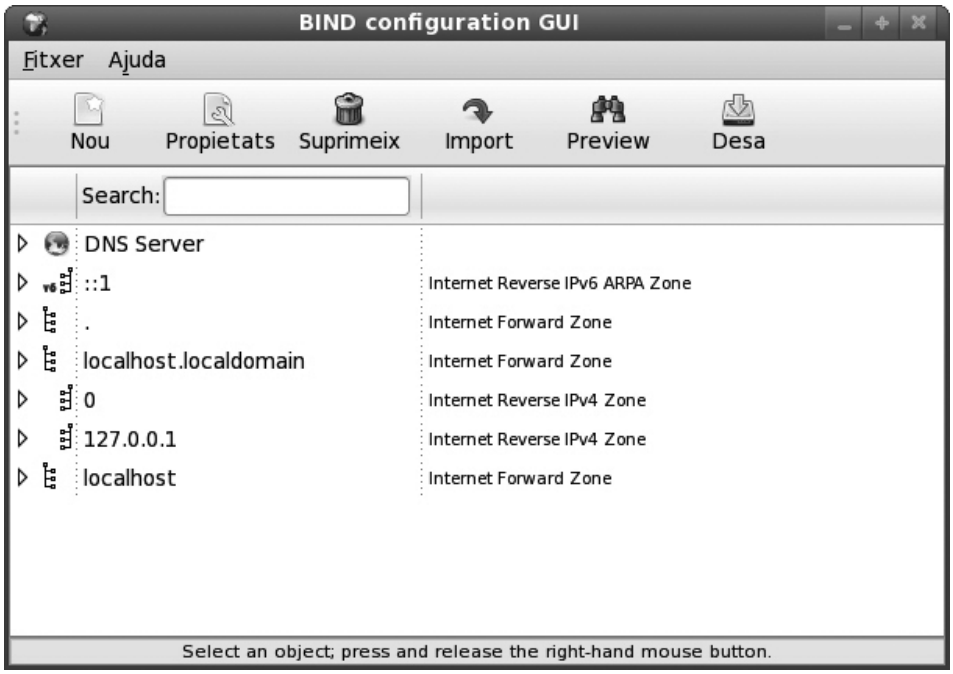
- Nom del servei: bind9 (localitzat a /etc/init.d/bind9).
- Fitxer de configuració: /etc/bind/named.conf.
- Directori dels fitxers de zona: /etc/bind.
- Llistat dels fitxers de zona predefinits que conté el directori anterior.
- Ubicació de fitxers d'exemple i pàgines de manual d'on poder obtenir una configuració inicial bàsica: /usr/share/doc/bind9 i /usr/share/man.

En la configuració per defecte es poden analitzar els diversos elements que es configuren:

- **options**: s'hi defineixen les opcions genèriques del servidor DNS.
- **logging**: es defineix com serà el procés d'enregistrament dels *logs* del servei.
- **localhost_resolver**: permet definir el servidor DNS com un servidor només *cau*. És a dir, no és autoritari de cap domini, no gestiona cap domini, cap zona, no té fitxers de zona; l'única funció que fa és de servidor DNS *cau*.
- **internal**: permet definir les zones i zones delegades que es volen gestionar amb el servidor. Es donarà servei a les xarxes locals internes que es defineixen en aquesta secció.
- **external**: defineix el servei a oferir a clients externs al domini. És per oferir serveis DNS a clients exteriors.

La figura 1.4 mostra una eina gràfica que permet la configuració del servei de noms. En sistemes GNU/Linux és usual que els servidors es gestionin editant directament els fitxers de configuració. Es disposa, però, de miniaplicacins (*applets*) que permeten fer aquesta mateixa tasca des d'un entorn gràfic. Usualment no fan res de nou, és a dir, simplement serveixen per modificar els fitxers de text de configuració mitjançant un entorn gràfic més amable.

FIGURA 1.4. Miniaplicació gràfica de gestió del servidor BIND



Podem observar les zones que s'instal·len per defecte llistant el directori de treball per defecte del servei. Amb tota seguretat hi trobarem els fitxers corresponents a les zones arrel, localhost i loopback. A més, el paquet del servei segurament disposa de fitxers d'exemple de zona igual que disposava d'un exemple de fitxer de configuració del servei.

```
[root@host ~]# ls -l /etc/bind/
total 56
-rw-r--r-- 1 root root 2761 de juny  21  2019 bind.keys
-rw-r--r-- 1 root root  237 de juny  21  2019 db.0
-rw-r--r-- 1 root root  271 de juny  21  2019 db.127
-rw-r--r-- 1 root root  237 de juny  21  2019 db.255
-rw-r--r-- 1 root root  353 de juny  21  2019 db.empty
-rw-r--r-- 1 root root  270 de juny  21  2019 db.local
-rw-r--r-- 1 root bind  463 de juny  21  2019 named.conf
-rw-r--r-- 1 root bind  498 de juny  21  2019 named.conf.default-zones
-rw-r--r-- 1 root bind  156 de maig  27 20:09 named.conf.local
-rw-r--r-- 1 root bind  888 de maig  28 08:19 named.conf.options
-rw-r----- 1 bind bind   77 de maig  27 11:40 rndc.key
-rw-r--r-- 1 root root 1317 de juny  21  2019 zones.rfc1918
```

1.3.4 Exemple de configuració bàsica

La configuració bàsica i l'estructura de fitxers un cop s'ha instal·lat el servidor BIND permet familiaritzar-nos amb els seus components. L'estructura dels fitxers de configuració es basa en un fitxer principal, /etc/bind/named.conf, que està organitzat en 3 fitxers més:

```
root@server:~# cat /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
```

```
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
root@server:~#
```

El contingut de cada un dels tres fitxers és el següent:

- /etc/bind/named.conf.options: conté les opcions del servei DNS.
- /etc/bind/named.conf.local: contindrà les zones noves que es volen afegir.
- /etc/bind/named.conf.default-zones: conté la zona arrel i les zones per defecte especificades al RFC 1912.

Directives

El llistat de totes les directives es pot consultar amb l'ordre:

man 5 vsftpdconf.

El contingut per defecte del **fitxer d'opcions** és el següent:

```
root@server:~# cat /etc/bind/named.conf.options
options {
  directory "/var/cache/bind";

  // If there is a firewall between you and nameservers you want
  // to talk to, you may need to fix the firewall to allow multiple
  // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

  // If your ISP provided one or more IP addresses for stable
  // nameservers, you probably want to use them as forwarders.
  // Uncomment the following block, and insert the addresses replacing
  // the all-0's placeholder.

  // forwarders {
  //  0.0.0.0;
  // };
  //========================================================================
  // If BIND logs error messages about the root key being expired,
  // you will need to update your keys.  See https://www.isc.org/bind-keys
  //========================================================================
  dnssec-validation auto;

  listen-on-v6 { any; };
};
```

Les opcions que apareixen en aquest fitxer són:

- Especificació el directori de treball del servei i els diferents fitxers de monitorització a utilitzar. Per exemple, fitxers per a volcats (*dump*), estadístiques (*statistics*) i estadístiques de memòria (*memstatistics*).
- Autenticació en les respostes DNS, si és el cas.
- Ports d'escolta del servei. En aquest cas també si es vol emetre respostes per a IPv6.

El contingut per defecte del **fitxer de configuració local** és el següent:

```
root@server:~# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Aquest fitxer ve completament buit i és on s'han d'especificar les zones noves que es volen afegir al servidor DNS. Es recomana també afegir l'espai d'adreces privat (RFC 1918) si no s'especifiquen a les noves zones.

Espai d'adreces privat

L'espai d'adreces privat especificat al RFC1918 és el següent:

TAULA 1.1.

10.0.0.0	10.255.255.255	(10/8 prefix)
172.16.0.0	172.31.255.255	(172.16/12 prefix)
192.168.0.0	192.168.255.255	(192.168/16 prefix)

El contingut per defecte del **fitxer de configuració de zones per defecte** és el següent:

```
root@server:~# cat /etc/bind/named.conf.default-zones
// prime the server with knowledge of the root servers
zone "." {
  type hint;
  file "/usr/share/dns/root.hints";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
  type master;
  file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
  type master;
  file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
  type master;
  file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
  type master;
  file "/etc/bind/db.255";
};
```

Aquest fitxer ve emplenat amb les zones per defecte. La més important és la primera, que és la **zona arrel** (fitxer /usr/share/dns/root.hints) i conté les 13 servidor arrels que hi ha a Internet. I les altres zones corresponen als espais d'adreces reservats especificats al RFC1912 i que el mateix document

recomana que apareixin (*certain zones should always be present in nameserver configurations*).

El contingut de la zona arrel és el següent:

```
root@server:~# cat /usr/share/dns/root.hints
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.cache
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:     March 13, 2019
;       related version of root zone:     2019031302
;
; FORMERLY NS.INTERNIC.NET
;
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     199.9.14.201
B.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:200::b
...
```

Espais d'adreces reservats

Els espais d'adreces reservats especificats al RFC1912 són els següents:

TAULA 1.2.

primary	localhost	localhost
primary	0.0.127.in-addr.arpa	127.0
primary	255.in-addr.arpa	255
primary	0.in-addr.arpa	0

1.4 Resolució, 'forwarding' i memòria cau

Tot sovint, en les aplicacions d'usuari i de sistema s'accedeix a recursos pel seu nom de domini. Per exemple, un client web requereix una determinada pàgina web, un navegador de fitxers vol accedir a unes carpetes d'una màquina remota que s'identifiquen pel nom de domini, el sistema ha de validar l'usuari en un servidor LDAP remot... En cada un d'aquests casos caldrà respondre una pregunta del tipus "A quina adreça IP correspon aquest domini?", "Quins són els servidors de noms del domini tal?", "Quins són els servidors de correus?". Aquestes preguntes no les responen les aplicacions individualment (el navegador web, el client d'autenticació...), sinó que utilitzen el *resolver* per fer-ho.

El *resolver* és la part client de l'arquitectura client/servidor del DNS. Ha d'atendre les necessitats de les aplicacions, confeccionar una consulta o *query*, enviar-la a un servidor DNS, obtenir la resposta i passar-la a l'aplicació pertinent. El *resolver* no és usualment una aplicació sinó un conjunt de biblioteques de funcions. Les aplicacions client es compilen i enllacen conjuntament amb aquestes biblioteques.

Els servidors que reben l'encàrrec de fer la resolució d'una consulta es poden comportar de maneres diferents en funció de com s'han configurat. Poden obtenir la resposta de la seva memòria cau, sol·licitar a un altre servidor de noms que sigui ell qui faci tota la feina de resolució (*forwarding*) o encarregar-se de fer la resolució pas a pas pel seu compte (*recursion*), consultant tans servidors de noms externs com faci falta.

En general podem catalogar el funcionament de la resolució en:

- memòria cau sí/no
- recursiu sí/no (recursiu/iteratiu)
- *forward* sí/no, combinat amb *forward only*

Fem una ullada al funcionament del mecanisme de resolució.

1.4.1 La resolució de noms

El mecanisme de resolució de noms DNS consta d'un client o *resolver* que realitzarà consultes (o *queries*) a uns servidors DNS.

Si el servidor disposa de la informació perquè forma part de la base de dades de la seva zona, emetrà una resposta **autoritativa**. Si disposa de la resposta perquè la té emmagatzemada temporalment (en un procés anomenat *cau*) també emetrà la resposta, però aquest cop de manera **no autoritativa**. Si no té informació del domini buscat, el servidor pot fer la consulta a altres servidors en un procés que pot ser **recursiu** o **iteratiu**. Sempre existeix un camí per trobar el domini buscat, que és preguntar als **nodes arrel** (*root servers*) de l'espai de noms de domini. Partint dels nodes arrel i recorrent l'arbre cap avall, es pot arribar al domini buscat, si és que existeix.

Sempre hi ha un camí a un domini existent partint del node arrel. Quan un servidor és consultat sobre un domini que desconeix (no és de la seva zona ni té la resposta en la memòria cau) pot escalar la pregunta a un servidor l'arrel (*root name server*). Això significa que els servidors arrel són crucials per al funcionament del DNS.

Exemple de resolució de noms DNS

Quina adreça IP té el *host* ns1.ioc.cat.? Si un estudiant australià intenta esbrinar això des del seu servidor de noms de Sydney, probablement acabarà preguntant per aquest domini a un dels nodes arrel. El node arrel desconeix el *host* ns1 del domini de l'IOC, però sí que

coneix tots els dominis de primer nivell (TLD). Per tant, l'arrel proporcionarà una llista amb els servidors de noms del domini cat. A continuació, el servidor de Sydney preguntarà a algun dels servidors de noms de la llista (del domini cat.) i obtindrà la llista de servidors DNS del domini ioc.cat. Preguntant als servidors d'aquest domini obtindrà l'adreça IP del *host* ns1 per al qual el domini ioc.cat. és autoritari (forma part de la seva zona).

Recursió i iteració

Quan el client o *resolver* emet una consulta al servidor DNS local (el servidor de noms que té configurat), aquest la pot tractar de manera **recursiva** o **iterativa**. De fet, el client *resolver* ja farà la consulta indicant si exigeix una resposta recursiva o iterativa. La diferència entre un mode i l'altre és com ha d'actuar el servidor DNS per obtenir la resposta quan no la té en la seva base de dades d'informació.

En el mode **iteratiu**, el servidor retorna la millor resposta possible basada en la seva informació local, sense preguntar a ningú més. En el mode **recursiu**, el servidor intenta trobar la resposta preguntant a tants altres servidors com calgui per obtenir-la.

Un servidor pot emetre les respostes següents:

1. Respon enviant la dada que li han sol·licitat (un nom de *host*, una adreça IP, la llista de servidors de noms, de servidors d'autenticació...).
2. Ha localitzat el domini buscat, però no disposa de la dada sol·licitada. Cal tenir en compte que es poden sol·licitar altres dades a part de l'adreça IP, per exemple, els servidors de correu que té el domini.
3. Finalment, pot ser que el domini sol·licitat no existeixi.

Recursió

Quan un servidor rep una consulta del client, mira la base de dades local de la seva zona. Si existeix la informació sol·licitada, la retorna. Si la dada no forma part de la seva zona, però la té emmagatzemada en la memòria cau (perquè ja ha realitzat amb anterioritat un consulta similar i ha emmagatzemat temporalment la resposta), també la retorna. Si la dada no forma part del seu espai de noms ni es troba en la memòria cau, el mode recursiu mana al servidor anar preguntant recursivament a altres servidors, apropant-se més a cada pas al domini sol·licitat. Si el servidor no coneix cap servidor més proper al domini buscat a qui preguntar, acaba preguntant als servidors de l'arrel.

Exemple de recursió

Si es consulta el domini www.inf.ioc.cat. i el servidor desconeix aquest domini, intentarà contactar amb un servidor de noms del domini inf.ioc.cat. Si tampoc sap com adreçar-se a aquest domini, intentarà contactar amb un servidor de noms de domini ioc.cat. Si també el desconeix, provarà de localitzar un servidor per al domini .cat. Si tampoc és el cas, es posarà en contacte amb un servidor de noms de l'arrel, . Un cop en l'arrel, sempre és possible accedir al domini buscat descendint per l'arbre de dominis.

Si tots els processos recursius acabessin preguntant als nodes arrel, aquests se saturarien. El servidor que ha rebut la consulta del *resolver* pregunta al node més proper al domini buscat. Si coneix algun servidor de noms més proper, li ho pregunta i evita d'anar a l'arrel.

Una altra manera d'evitar la sobrecàrrega dels nodes arrel és l'ús de la informació emmagatzemada de consultes anteriors, que es desa localment en la memòria cau del servidor.

Consulta d'informació delegada

Si es consulta l'adreça www.inf.ioc.cat. i el servidor que rep la consulta és el servidor de noms del domini ioc.cat., aquest no pregunta cap amunt (a cat. o a l'arrel .), sinó que obté de la seva pròpia base de dades la llista dels servidors de noms autoritaris de la zona delegada inf.ioc.cat., als quals preguntarà per obtenir una resposta.

Imagineu un alumne de Sydney, estudiant de l'IOC, que genera una consulta al servidor de noms del seu ISP australià per identificar el domini www.int.ioc.cat.. Probablement el servidor desconeix aquest domini i els propers, inf.ioc.cat. i ioc.cat. Però segurament en la memòria cau (per consultes prèvies) té la llista de servidors de noms autoritaris del domini cat. Serà a un d'aquests servidors (i no a un node arrel) a qui farà la consulta que li permetrà accedir de manera descendent al domini buscat.

Per tant, en el procés recursiu, el servidor de noms que rep la consulta del *resolver* ha de tornar una resposta que pot procedir de la seva base de dades de zona, de la memòria cau o de les respostes finals que ha obtingut preguntant recursivament a altres servidors propers al domini a consultar.

Fixeu-vos que un servidor que rep una consulta recursiva del *resolver* té la feina d'esbrinar per si mateix la resposta. Podria repetir la mateixa consulta al servidor més proper fent-la recursiva en lloc d'iterativa. Això exigiria a l'altre servidor fer tota la feina. Aquest plantejament, tot i que possible, es considera abusiu.

Usualment, el **client** consulta el seu DNS de manera **recursiva** i els **servidors** es consulten entre ells de manera **iterativa**.

Iteració

En el mode iteratiu, un servidor dóna la millor resposta possible basant-se en la pròpia informació (base de dades de zones locals i memòria cau). En cap cas no consulta cap altre servidor. Si no disposa de la resposta, lliura una llista amb els servidors més propers al domini que es busca. La llista pot ser d'un o més servidors i és tasca del servidor que ha fet la consulta decidir a quin d'ells tornar a preguntar (en el cas recursiu).

Els nodes arrel no accepten consultes recursives per evitar l'abús i la saturació.

Les consultes iteratives són usualment de servidor a servidor, però no del *resolver* al servidor. Si el *resolver* fes una consulta iterativa a un servidor, significaria que quan la resposta fos una referència a un altre servidor, el *resolver* hauria de fer una altra consulta. Generalment els *resolver* no tenen aquesta capacitat, simplement fan una consulta recursiva al servidor que tenen configurat i és aquest el que ha de fer tota la feina per obtenir la resposta.

El client *resolver* fa una consulta **recursiva** al seu servidor DNS local. Si el servidor DNS disposa de la resposta, la torna. Pot ser de la seva zona i serà una resposta **autoritativa** o pot tenir-la en la memòria cau i serà **no autoritativa**.
Si no disposa de la resposta, consulta **iterativament** altres servidors apropant-se al domini buscat. Cada servidor que consulta iterativament li pot proporcionar la resposta (autoritativa o no), si la coneix, o una llista de servidors DNS autoritatius per al domini indicat.
Finalment, s'obtindrà una resposta que pot ser aquella dada que es buscava o un error si el domini buscat no existeix.

Resolució inversa

Un *host* amb l'adreça IP 192.168.1.24 correspon al domini 24.1.168.192.IN-ADDR.ARPA.

El DNS proporciona un mecanisme per obtenir el nom de domini que correspon a una adreça IP determinada. Aquest mecanisme, anomenat **resolució inversa**, es basa en un domini especial anomenat IN-ADDR.ARPA. Hi ha protocols de xarxa que requereixen una resolució inversa correcta per funcionar bé i sovint s'utilitza com a mesura de seguretat per verificar l'existència de l'adreça IP en un domini.

La xarxa 172.16.32.0/24 correspon al domini ARPA següent: 32.16.172.IN-ADDR.ARPA.

S'ha ideat un domini anomenat IN-ADDR.ARPA que permet representar en forma de nom de domini totes les adreces IP possibles. El format són etiquetes numèriques del 0 al 255 que representen cada octet d'una adreça IP. Les etiquetes dels octets es concatenen en ordre invers i se'ls afegeix el sufix IN-ADDR.ARPA. Un nom de domini amb quatre etiquetes d'octets correspon a un *host*, un nom de domini amb menys etiquetes correspon a una xarxa.

En l'exemple següent es mostren els servidors de noms del domini ioc.cat i es fa una consulta de resolució inversa a cada un:

```
root@server:~# host -vt NS ioc.cat
Trying "ioc.cat"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65177
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 2

;; QUESTION SECTION:
;ioc.cat.     IN  NS

;; ANSWER SECTION:
ioc.cat.    900 IN  NS  dns2.nominalia.com.
ioc.cat.    900 IN  NS  dns1.nominalia.com.

;; ADDITIONAL SECTION:
dns2.nominalia.com. 4800  IN  A 81.88.63.48
dns1.nominalia.com. 254 IN  A 81.88.57.102

Received 108 bytes from 192.168.1.1#53 in 128 ms
root@server:~# host 81.88.57.102
102.57.88.81.in-addr.arpa domain name pointer dns1.nominalia.com.
root@server:~# host 81.88.63.48
48.63.88.81.in-addr.arpa domain name pointer dns2.nominalia.com.
root@server:~#
```

Exemple de configuració amb recursió

Vegeu les opcions que permeten configurar un servidor de noms en mode recursiu o en mode iteratiu. De fet, observant el fitxer de configuració podeu veure que és ben senzill:

```
# Fitxer /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

        allow-query-cache { any; };
        allow-query { any; };
        recursion yes;

        dnssec-validation auto;

        listen-on-v6 { any; };
};
```

Podeu observar que es tracta simplement d'establir l'opció apropiada:

- ***recursion yes;*** estableix el mode de funcionamet com a recursiu.
- ***recursion no;*** estableix el mode de funcionament com a iteratiu.

Com es pot saber si el servidor funciona recursivament o iterativament? És ben senzill: un servidor iteratiu simplement pot resoldre consultes de la zeva zona o consultes prèvies que té emmagatzemades en la memòria cau. Però no pot resoldre consultes externes a la seva zona, ja que la recursió s'ha desactivat.

Per tant, podeu configurar el servidor iterativament (*recursion no;*) i comprovar que no es resolen consultes externes com gmail.com.. Tot seguit podem activar la recursió, *recursion yes;*, i fer la mateixa consulta externa. En aquesta ocasió s'ha d'obtenir el resultat buscat.

1.4.2 Ús de servidor 'forwarder'

Existeix un cas particular de funcionament dels servidors de noms que anomenem *forwarders*. Són servidors que podem anomenar informalment *l'encarregat*, o podem dir, encara més informalment, que "li passem el mort a tal" on *tal* és el ***forwarder***.

Imaginem, per exemple, una xarxa local corresponent al departament d'administració d'una empresa i que forma part d'una xarxa més gran, corresponent a l'empresa. En la xarxa d'administració s'ha instal·lat un servidor de noms només cau que no administra cap domini local. Es vol que aquest servidor tampoc generi consultes externes per motius que els administradors de xarxes creuen oportuns: per exemple, per temes de tallafocs, seguretat... Es vol que el servidor faci totes les consultes al servidor de noms de l'empresa, que serà qui farà la tasca de *forwarder*.

És a dir, quan es vol que un servidor de noms no realitzi consultes externes via recursió, sinó que totes les consultes les realitzi a un mateix servidor, és diu que es traspassen les consultes a aquest servidor encarregat. Aquest servidor és un *forwarder*.

> Es diu que un servidor fa ***forward*** quan passa totes les consultes externes a un altre servidor, que serà l'encarregat de resoldre-les.
> El servidor que rep l'encàrrec de fer aquesta feina de resolució s'anomena ***forwarder***.

Per configurar una estructura de resolució de noms amb un *forwarder*:

- En el servidor de noms que traspassarà les consultes a l'altre cal indicar quin servidor de noms serà el *forwarder*.
- En el que farà la funció de *forwarder* no cal indicar res. Simplement es trobarà que rep moltes consultes de resolució d'un *host* determinat, però no s'ha d'indicar res.

```
1 allow_query_cache { any;};
2 allow_query { any; };
3 recursion yes;
4 forward only;
5 forwarders { 192.168.122.1; 192.168.122.2};
```

En el fragment de codi anterior, extret del fitxer de configuració /etc/named.conf, podem observar les dues directives que indiquen al servidor que ha de traspassar totes les consultes a un servidor "encarregat". Anem a veure el significat de cada opció:

- ***forwarders***: indica l'adreça o adreces IP dels servidors als quals es farà *forward* de la consulta, és a dir, als quals s'encomanarà la tasca de resoldre-la. Com que l'especificació del protocol DNS recomana dos servidors de noms per domini, és usual trobar en aquesta opció almenys dues adreces IP. Aquestes corresponen als dos servidors d'un domini, els típicament anomenats *primari* i *secundari*.
- ***forward only***: si s'activa aquesta opció, el servidor funcionarà exclusivament en mode *forward*, de manera que qualsevol consulta serà tramesa al *forwarder*, idependentment de la memòria cau i de les zones pròpies.

1.4.3 Respostes de memòria cau

Es pot configurar un servidor de noms de domini perquè faci la funció de només cau. No gestiona cap zona, simplement atén les peticions dels *resolvers* client i les passa a altres servidors de noms. La seva funció és emmagatzemar en la memòria cau les respostes que obté abans de passar-les als clients. Això li permetrà que les

futures consultes que siguin repetides les pugui contestar directament en lloc de demanar-les a un altre servidor. Evidentment, aquest tipus de servidor no emet respostes amb autoritat.

La funció de memòria cau es pot activar i desactivar segons convingui. A més, es pot combinar amb les altres opcions de resolució per seleccionar el mode de funcionament del servidor. Les combinacions més usuals són:

- Si únicament es configura la funció de memòria cau es tractarà d'un servidor només cau.
- Si es combina la memòria cau i la recursió es farà resolució externa i es disposarà d'un cau local per poder contestar immediatament les consultes fetes amb anterioritat.
- El mateix passa si es combina la memòria cau amb el *forward*. El *forward* traspassarà la responsabilitat de resoldre les consultes externes a un altre servidor de noms. Disposar de la memòria cau activada permetrà respondre immediatament les consultes repetides.
- Desactivar la memòria cau significa que no s'emmagatzemen localment les respostes que es reben. Per tant, si es tornen a fer consultes que ja s'havien realitzat anteriorment caldrà tornar-les a resoldre pel mètode apropiat.

Si torneu a examinar el bloc d'opcions de configuració del mode de funcionament del servidor, extretes del fitxer /etc/bind/named.conf.options, observareu:

```
allow–query–cache { any;};
allow–query { any; };
recursion yes;
```

L'opció *allow-query-cache* permet indicar si s'activa o no la memòria cau i per a quines consultes. Les opcions que pot prendre són:

- *any;* indica que s'emmagatzemen en la memòria cau totes les respostes rebudes.
- *none;* no permet el funcionament de la memòria cau.
- *adreces-ip;* indica que les consultes que provinguin d'aquestes adreces-ip s'han d'emmagatzemar en la memòria cau. Les adreces-ip poden ser, de fet, combinacions dels tres conceptes següents: adreces IP, de xarxa o una llista-de-noms. Per exemple, 192.168.4.0/24;.

Autoritari, no autoritari i informació de memòria cau

Cada zona de l'espai dels noms de domini és gestionada per un o més servidors autoritaris per a la zona. Això significa que són els servidors que manen, que tenen l'última paraula respecte de la zona. De fet, significa que són els servidors que administren la zona. Aquests servidors es configurem com a **autoritaris** de

la zona. Les respostes que emeten a consultes referents a la seva zona porten un segell indicant que són autoritàries, que provenen de la base de dades distribuïda de l'espai de noms de domini.

Els servidors de noms guarden en una memòria cau les respostes que reben d'altres servidors. Aquesta informació també és utilitzada per elaborar respostes a consultes de dominis de fora de la zona de la qual són autoritaris. És a dir, quan un servidor rep d'un altre la llista de servidors autoritaris (en una consulta iterativa) per a una zona, aquesta llista s'emmagatzema a la memòria cau. Quan un servidor rep una resposta a una consulta també es guarda. Quan rep una resposta negativa, indicant que el domini de la consulta o el tipus de dada sol·licitat no existeix, també ho guarda. En aquest cas s'anomena *negative caching*.

Un servidor respon utilitzant la informació disponible en la base de dades de la seva zona (resposta autoritativa) o de la informació emmagatzemada en la seva memòria cau (provinent de consultes a zones externes efectuades anteriorment).

Quan rep una consulta a una zona externa que ja s'havia efectuat abans, s'utilitza la resposta de la memòria cau. També s'agafa de la memòria cau la llista de servidors autoritaris per a una zona més propera al domini que es busca, per tal de no preguntar als servidors arrel. Quan un servidor de noms respon utilitzant la informació de la memòria cau, la resposta és **no autoritativa**. No prové del servidor autoritari de la zona, sinó que s'utilitza una informació prèviament obtinguda (i que pot estar desfasada).

Per exemple, s'ha fet una consulta per al domini adm.ioc.cat i en el procés de resolució recursiu el servidor ha obtingut la llista de servidors autoritaris per als dominis cat. i ioc.cat. Si ara es demana al mateix servidor pel domini inf.ioc.cat. primerament provarà de resoldre aquest domini, però com que és desconegut, el següent domini més pròxim és ioc.cat. Aquest sí que el coneix, perquè el té emmagatzemat a la memòria cau de la resposta anterior. S'utilitzarà la llista de servidors autoritaris del domini ioc.cat. per obtenir una resposta o per continuar descendint per l'arbre de l'espai de noms.

Emmagatzemar informació de les respostes d'altres servidors en la memòria cau ofereix dos grans avantatges:

1. Incrementa la velocitat de resposta. Ja no cal anar a trobar la resposta a la font de dades de la zona, sinó que s'utilitza la informació d'una resposta anterior.
2. Evita la sobrecàrrega dels servidors arrel.

No cal anar al node arrel per cada consulta un cop que es disposa en la memòria cau d'informació més propera al domini a cercar. La utilització de la memòria cau té, però, un inconvenient que cal considerar: les dades no necessàriament són actualitzades. El DNS es fonamenta en una base de dades jeràrquica i distribuïda en la qual la informació es troba en els fitxers gestionats per cada servidor de zona. Una dada emmagatzemada en la memòria cau pot no reflectir la dada real, que ha estat modificada en el servidor autoritari de zona però que encara no s'ha propagat perquè es manté en la memòria cau fins que caduca el seu TTL (temps de vida).

Són servidors **autoritaris** els que administren una zona (tant el servidor primari o mestre com els secundaris o esclaus). Tenen accés a la informació **original** de la base de dades de zona.
Són respostes **no autoritàries** les que provenen de la informació desada en la memòria cau.

Fixeu-vos que si la informació de la memòria cau es desés indefinidament, els canvis que es fessin en els servidors autoritaris no es propagarien als altres servidors (perquè segurament ja disposarien d'una resposta en la seva memòria cau). Cal un mecanisme perquè la informació de la memòria cau caduqui transcorregut un cert interval de temps. Anomenem **TTL**, *Time To Live* o **temps de vida**, l'interval de temps que les dades han de perdurar en la memòria cau. Un cop transcorregut aquest temps, les dades s'eliminen. Si fa falta una dada, per obtenir-la caldrà fer una nova consulta.

Servidor només cau

La configuració completa d'un servidor DNS no és difícil, però és entretinguda. Cal crear cada fitxer de zona amb les entrades pertinents de cada *host* de la zona. Hi ha organitzacions petites que no necessiten configurar el servei DNS completament, en tindrien prou a tenir un servidor DNS local que permetés accelerar les consultes DNS que es fan a l'exterior.

Sabem que tota resolució de nom de domini comporta una consulta a un servidor DNS, usualment el del proveïdor de servei d'Internet (ISP). Es pot posar en marxa un servidor només cau en una xarxa local per proporcionar més eficiència a les consultes. El servidor només cau no administra cap zona, no té registres de recurs, simplement rep les consultes dels clients, les tramet al servidor DNS extern, rep les seves respostes, les desa en la memòria cau i les retorna al client.

El benefici d'aquest esquema és que el servidor només cau acumula en la memòria les respostes que va obtenint. En les consultes següents, si es demana pels mateixos dominis, ja no li cal passar la consulta a l'exterior, sinó simplement respondre des de la memòria cau. Evidentment, les seves respostes són sempre no autoritàries.

Un servidor **només cau** només emmagatzema les respostes d'altres servidors externs en la memòria, però no gestiona cap zona. No és autoritari, simplement augmenta l'eficiència quan rep consultes de les quals ja en sap la resposta (la té a la memòria cau).

Vegeu un exemple de configuració d'un servidor només cau:

```
root@server:~# cat /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

        allow-query-cache { any; };
        allow-query { any; };
```

```
        recursion yes;

        dnssec-validation auto;

        listen-on-v6 { any; };
};
root@server:~# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
root@server:~#
```

En l'exemple anterior es pot observar que no es defineix cap fitxer de zona per ser administrada, excepte el fitxer de resolució directa dels servidors de noms de la zona arrel. Per tant, aquest servidor únicament atén peticions DNS com a intermediari i les desa a la memòria cau. El directori on emmagatzema temporalment la informació és /var/cache/bind.

1.5 Creació de zones

El propòsit principal d'un servei DNS és administrar una zona com per exemple una xarxa local amb tots els equips d'una organització, o un conjunt de zones d'una organització més complexa. Per fer això caldrà definir els fitxers de configuració del servei DNS i definir cada una de les zones de què es compongui la xarxa. També caldrà crear els fitxers corresponents a la resolució inversa de cada xarxa i del *loopback*.

Per crear una **zona pròpia** caldrà:

- Definir les zones en el fitxer de configuració del servei.
- Crear el fitxer de zona en què es defineix la resolució directa per a cada *host* de la zona i les característiques de la zona.
- Crear el fitxer de resolució inversa de la zona.

Les zones descriuen els equips que en formen part. És a dir, cada fitxer de zona és una base de dades que descriu els *hosts* que hi ha en la zona i la mateixa zona. Vegeu dos exemples de fitxers de zona abans d'explicar com descriure cadascun dels elements que hi pertanyen:

- Zona ioc.cat., corresponent a la resolució directa.
- Zona 2.0.10.in-addr.arpa., corresponent a la resolució inversa de la zona. Aquesta zona correspon a la xarxa 10.0.2.0/24.

```
;
; Fitxer de configuració del domini ioc.cat
;
$TTL 1D
ioc.cat.  IN SOA    server.ioc.cat. admin.ioc.cat. (1 3M 1M 1W 1D)
ioc.cat.  IN NS     server.ioc.cat.
server    IN A      10.0.2.10
www       IN A      10.0.2.11
ftp       IN CNAME  www
pc1       IN A      10.0.2.101
pc2       IN A      10.0.2.102
```

```
;
; Fitxer de configuració de la zona inversa ioc.cat
;
$TTL 1D
@    IN SOA server.ioc.cat. admin.ioc.cat. (1 3M 1M 1W 1D)
@    IN NS  server.ioc.cat.
10   IN PTR server.ioc.cat.
11   IN PTR www.ioc.cat.
11   IN PTR ftp.ioc.cat.
101  IN PTR pc1.ioc.cat.
102  IN PTR pc2.ioc.cat.
```

1.5.1 Tipus de registres

Aprofitar fitxers de zona

Els fitxers de zona de descripció del *loopback* i dels nodes arrel són pràcticament iguals per a totes les zones, de manera que usualment es copien d'una zona ja existent en lloc d'escriure'ls de nou.

El sistema de noms de domini és una base de dades jeràrquica i distribuïda en què cada servidor de noms gestiona la informació corresponent a la zona de la qual és autoritari. Cada zona conté informació dels *hosts* que la formen. La informació de zona s'emmagatzema en forma de **registre de recurs** o *resource record* (RR).

Aquest registre conté la informació que permet identificar cada nom de domini amb l'adreça IP corresponent, anomenat *forward mapping* o **resolució directa**. També conté la informació per identificar cada adreça IP amb el nom de domini corresponent, anomenat *reverse mapping* o **resolució inversa**. La informació de zones conté altres dades que permeten identificar els servidors DNS autoritaris per la zona, els servidors de correu...

La configuració d'una zona s'emmagatzema en un conjunt de fitxers anomenat *fitxers de zona*. L'especificació del DNS diu com han de ser aquests fitxers de zona i com s'hi han de descriure els registres de recurs (descripció de cada element que pertany a la zona).

> El conjunt dels registres de recurs de totes les zones de l'espai de noms formen la **base de dades** distribuïda jeràrquica del sistema DNS.

En qualsevol zona hi haurà almenys els **fitxers de zona** següents:

- Un fitxer amb les associacions dels noms de domini a adreces IP. Aquest fitxer defineix la resolució directa.

- Un fitxer per a cada subxarxa amb l'associació de cada adreça IP al seu nom de domini canònic. Defineix la resolució inversa.
- Un fitxer amb la definició de la resolució inversa del *loopback*.
- Un fitxer amb la descripció dels nodes arrel d'Internet.

Un cop que els fitxers de zona contenen tots els registres de recurs necessaris, cal configurar el servidor de noms perquè utilitzi aquests fitxers. Si bé la configuració dels fitxers de zona és estàndard (definida per l'especificació DNS), la configuració del servidor depèn del programa que s'utilitzi.

1.5.2 Registres de recurs

Cada fitxer de zona conté un conjunt d'entrades cadascuna de les quals defineix un **registre de recurs (RR)**. Els registres més usuals són SOA, NS, A, CNAME, PTR i MX. L'ordre en què apareixen és indiferent, però usualment és el mateix dels exemples. Cada línia té el format:

```
1 domini classe [ttl] tipus rdata:
```

- **domini** és el nom de domini que s'està definint.
- **classe** només pren actualment el valor "IN", per Internet.
- **ttl** és un camp opcional que descriu el temps de vida durant el qual cal emmagatzemar aquest registre en la memòria cau.
- **tipus** és el tipus d'RR que s'està definint.
- **rdata** és el valor que s'associa al nom de domini que es defineix.

Tot i que es pot definir un TTL en cada registre de recurs, el més usual és definir un TTL genèric per a totes les entrades del fitxer de zona. El servidor BIND 9 utilitza la directiva *$ttl* (per exemple: $ttl1 1h) per indicar el temps que els altres servidors de noms han de guardar en la seva memòria cau les respostes d'aquest servidor (una hora en l'exemple).

En la secció "Annexos" del web d'aquest mòdul trobareu altres tipus d'RR. També es poden trobar en l'especificació DNS.

Registre SOA

El registre de recurs **SOA** (*start of authority* o **inici de definició de zona amb autoritat**) diu que el fitxer de zona on es troba és la millor font de dades per a la zona, que el servidor de noms és autoritari per a la zona. Acostuma a ser el primer RR que hi ha en el fitxer de zona, tot i que no és obligatori. Per cada fitxer de zona hi ha d'haver només un registre SOA.

Punt final en el nom de domini

Posar o no el punt al final d'un nom de domini és important. Si acaba amb punt és un nom de domini absolut. Si no du punt és un nom relatiu i s'hi afegirà el domini per defecte al final.

Un registre SOA té el format:

```
nomDomini. IN SOA nsPrimari. admin.nsPrimari. (opcions–slaves)
```

Un exemple seria el següent:

```
inf.ioc.cat. IN SOA ns1.inf.ioc.cat. admin.ns1.inf.ioc.cat. ( 1 3h 1h 1w 1h )
```

La descripció de cada camp és la següent:

- **nomDomini.** indica el nom del domini que s'està definint i pel qual el servidor de noms és autoritari. Fixeu-vos en el punt final: és important posar-lo.
- **IN** indica que la classe és Internet.
- **SOA** informa que és un registre de recurs tipus SOA.
- **nsPrimari.** és el nom del *host* servidor de noms primari per a aquesta zona. Un altre cop pareu atenció al punt final.
- **admin.nsPrimari.** és l'adreça de correu electrònic de l'administrador del servidor de noms de domini, amb el format *usuari.servidor*. El primer punt que separa el nom d'usuari i el nom del servidor cal interpretar-lo com una arrova (*usuari@servidor*).
- **opcions-slaves** són paràmetres que s'indiquen entre parèntesis i que serveixen per definir com ha de ser la comunicació entre el servidor primari (o *master*) i els servidors secundaris (o *slaves*). A grans trets s'indiquen els conceptes següents:

  - *Serial*: el número de sèrie de la versió de les dades. A cada canvi de les dades de la zona, el número s'incrementa.
  - *Refresh*: temps a transcórrer entre cada refresc de dades del servidor secundari.
  - *Retry*: temps d'espera per tornar a intentar un refresc quan el servidor secundari ha fallat en l'intent d'actualitzar les seves dades des del servidor primari.
  - *Expire*: temps a partir del qual les dades del servidor secundari es consideren sense autoritat si no s'han refrescat abans.
  - *Minimum*: valor del TTL dels camps per defecte. Recordeu que a cada camp s'hi pot assignar un TTL específic. Segons la versió del servidor indicarà el TTL de les respostes negatives (*negative caching*), ja que el temps TTL es defineix per la directiva *$ttl*.

Formats de temps del BIND

BIND admet diferents formats per indicar el temps:

- #s: *seconds* (per defecte)
- #m: *minutes*
- #h: *hours*
- #d: *days*
- #w: *weeks*

Admet també combinacions, com per exemple: 3w12h, 2h20m...

Registre NS

El registre de recurs **NS** (*name server* o **servidor de nom**) defineix un servidor de noms autoritatiu per a la zona. Hi haurà tantes entrades NS com servidors de noms autoritatius hi hagi en la zona. L'estàndard DNS en recomana almenys dos (un de primari o *master* i un de seguretat, secundari o *slave*).

Un registre NS consta dels camps:

```
nomDomini. IN NS nameServer.
```

Un exemple seria aquest:

```
inf.ioc.cat. IN NS ns1.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **nomDomini.** indica el nom del domini que s'està definint.
- **IN** indica que la classe és Internet.
- **NS** descriu que es tracta d'un tipus de registre de recurs en què es defineix un servidor de noms.
- **nameServer.** és el nom del servidor de noms. Fixeu-vos un altre cop que tant *nomDomini.* com *nameServer.* acaben en punt per indicar que són noms de domini absolut o FQDN.

En la llista següent es pot veure part d'una resposta a una consulta *nslookup* per observar quins són els servidors de noms de Yahoo:

```
Authoritative answers can be found from:
yahoo.com       nameserver = ns2.yahoo.com.
...
ns8.yahoo.com   internet address = 202.165.104.22
```

Registre A

Un registre de recurs **A** (*address* o **adreça**) associa un nom de *host* a una adreça IP (resolució directa). Per cada nom de *host* de la xarxa caldrà disposar d'una entrada que associï el nom del *host* a la seva adreça IP.

Un registre A consta dels camps:

```
nomHost. IN A IP
```

Un exemple seria aquest:

```
mahatma.inf.ioc.cat. IN A 192.168.0.2
siddhartha           IN A 192.168.0.3
```

La descripció de cada camp és la següent:

- **nomHost.** indica el nom del *host* que s'està definint. Pot ser relatiu (sense punt final) o absolut (afegint el domini complert al final).
- **IN** indica que la classe és Internet.
- **A** informa que es tracta d'un tipus de registre de recurs de definició d'adreça IP.
- **IP** és l'adreça IP assignada al *host*.

Fixem-nos un altre cop que *nomHost.* acaba en punt per indicar que és un FQDN. Si no acabés en punt s'interpretaria com un nom relatiu al SOA que s'està definint actualment. Un *host* pot tenir més d'una IP assignada al mateix nom de *host*. Quan això passa s'anomena *multi-homed*. Simplement caldrà que hi hagi un registre A per a cada adreça IP. Constarà del mateix nom de *host* a l'esquerra de la definició i de la corresponent adreça IP a la dreta. Per exemple:

```
superserver.dom.com. IN A 10.0.0.1
superserver.dom.com. IN A 10.0.0.2
multihomed.ioc.cat.  IN A 172.12.0.1
                     IN A 172.12.0.2
```

Els noms definits en els registres de tipus A són noms **canònics**. Un *host* es pot identificar per més d'un nom, però només un és el nom canònic (original), la resta són **àlies**. Els noms canònics es defineixen amb el tipus de registre A. Els àlies es defineixen amb el tipus de registre CNAME.

A i CNAME

Compte! No s'han de confondre els registres de recurs A i CNAME: el registre A defineix *hosts*, mentre que el registre CNAME defineix àlies.

Registre CNAME

Els registres de recurs **CNAME** (*canonical name* o **nom canònic**) associen un àlies a un nom canònic.

Un registre CNAME consta dels camps:

```
nomHost. IN CNAME hostCanonicalName. | IP
```

Un exemple seria aquest:

```
ftp.inf.ioc.cat. IN CNAME mahatma.inf.ioc.cat.
tftp.inf.ioc.cat IN CNAME 192.168.0.2
```

La descripció de cada camp és la següent:

- **nomHost.** indica el nom de l'àlies que s'està definint.
- **IN** indica que la classe és Internet.
- **CNAME** informa que es tracta d'un registre de recurs de definició d'un àlies.

Exemple de host multi-homed

Es vol posar l'àlies *super1* i *super2* a cada una de les IP del host superserver.com (un *host* que té dues adreces IP assignades a aquest nom). Les entrades CNAME serien les següents:

- super1.dom.com. IN CNAME 10.0.0.1
- super2.dom.com. IN CNAME 10.0.0.2

- **hostCanonicalName | IP** és el nom de *host* canònic al qual s'assigna l'àlies. Fixeu-vos un altre cop que és un FQDN i que acaba en punt. Generalment, els registres CNAME tenen a la part dreta de la definició un nom canònic, però de vegades caldrà indicar-hi una adreça IP. Penseu en un *host multi-homed* amb múltiples adreces IP que a més té àlies. Si la definició fos pel nom canònic del *host*, no se sabria quina de les adreces IP correspon a l'àlies. En aquests casos, el CNAME apunta a una adreça IP.

La resolució dels àlies s'obté buscant l'entrada de l'àlies en el fitxer de zona. Amb l'entrada CNAME s'obté el nom canònic corresponent a l'àlies. Un altre cop es torna a buscar en el fitxer de zona, ara el nom canònic. Una entrada de tipus A proporcionarà l'adreça IP corresponent (àlies –> CNAME –> nom canònic –> A –> adreça IP).

Registre PTR

Un registre de recurs **PTR** (*pointer* o **punter**) associa una adreça IP al nom de *host* corresponent (resolució inversa). Cal una entrada PTR per a cada interfície de xarxa de la zona, per a cada adreça IP.

Un registre PTR consta dels camps:

```
1 ipInversa.in–addr.arpa. IN PTR hostName.
```

Un exemple seria aquest:

```
1 2.20.168.192.in–addr.arpa. IN PTR mahatma.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **ipInversa.in-addr.arpa.** indica l'adreça IP escrita en forma de domini in-addr.arpa per poder fer la resolució inversa. Les adreces IP s'escriuen al revés quan formen part del domini in-addr.arpa. Així, una IP 192.168.20.2 s'escriu 2.20.168.192.in-addr.arpa.
- **IN** indica que la classe és Internet.
- **PTR** informa que es tracta d'un registre de recurs de definició de la resolució inversa d'una adreça IP.
- **hostName.** és el nom de *host* FQDN assignat a l'adreça IP.

Registre MX

Un registre **MX** (*mail exchanger* o **servidor de correu electrònic**) defineix un servidor de correu. Es pot posar una entrada MX per a cada servidor de correu, però no és obligatori que n'hi hagi cap.

Un registre MX consta dels camps:

```
nomDomini. IN MX num mailServer.
```

Un exemple seria aquest:

```
inf.ioc.cat. IN MX 10 mailhost.inf.ioc.cat.
```

La descripció de cada camp és la següent:

- **nomDomini.** indica el nom del domini que s'està definint.
- **IN** indica que la classe és Internet.
- **MX** informa que es tracta d'un registre de recurs que defineix un servidor de correu per a aquest domini.
- **num** és un valor numèric que expressa el grau de preferència d'aquest servidor de correu respecte a altres servidors de correu del domini. El valor més baix és el que es prefereix. Són valors arbitraris que defineix l'administrador de xarxes.
- **mailServer.** correspon al nom FQDN del servidor de correu que s'està definint.

Observeu la llista de servidors de correu de Google a partir de:

```
root@server:~# host google.com
google.com has address 216.58.201.174
google.com has IPv6 address 2a00:1450:4003:80b::200e
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 40 alt3.aspmx.l.google.com.
google.com mail is handled by 20 alt1.aspmx.l.google.com.
root@server:~#
```

Exemples de registres de recurs (RR)

Les dues llistes següents mostren exemples dels fitxers de configuració per a la resolució directa i la resolució inversa de la zona ioc.cat. En el primer es defineixen dos servidors de nom, un encaminador, una impressora i dos *hosts*. El primer dels servidors de noms també fa les funcions de servidor de correu, web i FTP.

```
;Exemple de fitxer de zona ioc.cat
$TTL 3D
ioc.cat. IN SOA ns1.ioc.cat. admin.ioc.cat. { 23; 8H; 2H; 4W; 1D; };
      NS ns1.ioc.cat.
      NS ns2.ioc.cat.
      MX 10 correu.ioc.cat.
ns1.ioc.cat. A     192.168.0.5; servidor amb 2 ip
             A     172.16.20.5
ns2.ioc.cat. A     192.168.0.7; servidor dns slave
router       A     192.168.0.1; router. Nom relatiu
correu       CNAME ns1 ; alias correu
www          CNAME ns1 ; alias web
ftp          CNAME ns1 ; alias ftp
```

```
hp-7200c     A     192.168.0.2; impressora
pc01         A     192.168.0.50
pc02         A     192.168.0.51
```

En la llista següent es pot veure com es defineix una entrada PTR per a cada nom canònic definit en la resolució directa per a una subxarxa concreta. La subxarxa 192.168.0.0/24 utilitza el fitxer 0.168.192.in-addr.arpa per a la resolució inversa:

```
; Zona 0.168.192.in-addr.arpa.
; Exemple de fitxer de zona inversa ioc.cat
; correspon a la xarxa 192.168.0.0/24
$TTL 3D
ioc.cat. IN SOA ns1.ioc.cat. admin.ioc.cat. { 23; 8H; 2H; 4W; 1D;};
   NS ns1.ioc.cat.

5  PTR ns1.ioc.cat.
7  PTR ns2.ioc.cat.
1  PTR router.ioc.cat.
2  PTR hp-7200c.ioc.cat.
50 PTR pc01.ioc.cat.
51 PTR pc02.ioc.cat.
```

1.5.3 Configuració dels fitxers de zona

Exemple de configuració del servidor de noms

Per a una zona que es compon d'una única xarxa 192.168.20.0/24 amb el nom de domini inf.ioc.cat caldran els següents fitxers de zona:

- El fitxer de resolució directa inf.ioc.cat.zona.db.
- El fitxer de resolució inversa 192.168.20.zona.db o inf.ioc.cat.rev.zona.db.
- El fitxer de resolució inversa del *loopback*.
- El fitxer amb la informació dels nodes arrel del DNS, named.ca.

Els fitxers de zona contenen els registres de recurs que formen la base de dades de la zona. Cal configurar el servidor de noms per indicar-li quins són i on són aquests fitxers. Cada administrador anomena els fitxers com li plau o seguint l'estil marcat per l'aplicació servidor DNS que utilitza. Un exemple és anomenar els fitxers de zona amb el format *db.nomDomini* per al fitxer de resolució directa i *db.ipSubXarxa* per a la resolució inversa per a cada xarxa de la zona. En els exemples d'aquesta documentació s'utilitza la sintaxi *nomDomini.zona.db* per a la resolució directa, *nomDomini.rev.zona.db* per als de la resolució inversa i *ip.rev.zona.db* quan cal definir la resolució inversa d'altres xarxes.

Independentment de l'aplicació que s'utilitzi com a servidor de noms, caldrà configurar-la per dir-li on són aquests fitxers, com es diuen, si fan la funció de servidor autoritari per a la zona o no, si fan la funció de primari o secundari i altres opcions possibles.

Per cada fitxer de zona caldrà definir una entrada al fitxer de configuració global de BIND indicant el nom de la zona, el tipus i el nom del fitxer.

La sintaxi és:

```
zone nom_zona in { type master|slave|hint; file "nom_fitxer_zona"; };
```

Un exemple de configuració de zona podria ser:

```
zone inf.ioc.cat in { type master; file "inf.ioc.cat.zona.db;" }.
```

La descripció de cada camp és la següent:

- **zone nom_zona**: com es pot veure en l'exemple es defineix una zona corresponent al domini inf.ioc.cat.
- **type master | slave | hint**: el servidor serà primari (*master*) per a aquesta zona. Serà el que tindrà els fitxers amb les dades de la zona. El camp tipus pot prendre els valors *master*, *slave* i *hint*, que signifiquen el següent:
- **file nom_fitxer_zona**: indica el fitxer amb els registres de recurs de la zona. En l'exemple és el fitxer inf.ioc.cat.zona.db.

- *master*: el servidor té autoritat per a aquesta zona i gestiona els fitxers de zona.
- *slave*: el servidor és autoritari per a la zona, però obté les dades de la zona del servidor primari o *master*.
- *hint*: indica que es tracta de la informació corresponent als servidors de noms de la zona arrel. Aquesta informació té un tractament especial diferent del de les altres zones.

Cal parar especial atenció en la definició dels fitxers de zona per a la resolució inversa, utilitzant per exemple la xarxa 192.168.20.0/24. La zona s'anomena 20.168.192.in-addr.arpa, i el fitxer de zona, per exemple, db.192.168.20. El nom del fitxer conté l'adreça de xarxa en l'ordre natural, però el domini té els octets invertits perquè forma part del domini in-addr.arpa.

Vegeu l'exemple del fitxer de configuració del servei que inclou la definició de la zona directa ioc.cat. i la definició de la corresponent resolució inversa 2.0.10.in-addr.arpa. Aquestes definicions apunten als fitxers corresponents que descriuen els registres de recurs de cada una d'aquestes dues zones.

```
zone "ioc.cat" {
    type master;
    file "/etc/bind/ioc.cat.zona.db";
};
zone "2.0.10.in–addr.arpa" {
    type master;
    file "/etc/bind/ioc.cat.rev.zona.db";
};
```

1.5.4 Delegació de zona

El sistema de noms de domini DNS és una estructura jeràrquica i distribuïda. Hem vist com crear dominis i administrar-los en una zona que pot contenir altres subdominis que no s'hagin delegat. Per fer que l'estructura jeràrquica funcioni apropiadament cal poder delegar subdominis a altres entitats.

Delegar un subdomini implica transferir tota l'administració del domini delegat a una altra entitat. Un cop delegat, aquesta entitat és la responsable total de la seva

gestió. Caldrà establir el lligam entre el domini pare i el domini fill. Segurament caldrà realitzar els passos següents:

En el servidor de noms del domini fill:

- Generar els fitxers de zona directa i inversa i configurar apropiadament el servei.
- Verificar-ne el funcionament.

En el servidor de noms del domini pare:

- Realitzar la delegació.
- Verificar la delegació consultant registres del domini fill.

Optimitzar la consulta de la zona pare en el domini fill:

- Observar que des del servidor de noms fills no es disposa d'accés a la informació del domini pare. No hi ha lligam. Bé, sí, de fet, hi ha el mateix lligam que amb qualsevol altre domini. S'hi podrà accedir o no igual que a qualsevol altre domini.
- Fer del servidor de noms fill un servidor esclau de la zona pare.

Com a exemple pràctic creem una zona delegada inf.ioc.cat. corresponent a la xarxa 172.19.0.0/16. Aquests són els fitxers de zona:

```
; Fitxer de configuració dels hosts de la zona:
; inf.ioc.cat  (172.19.0.0/16)
$TTL 3D
@ IN SOA ns.inf.ioc.cat. admin.inf.ioc.cat. 1 3H 15M 1W 1D
          NS    ns
          MX    10 mailhost
ns        A     172.19.1.1
mailhost  A     172.19.1.2
printer   A     172.19.2.5
server    A     172.19.2.10
router    CNAME server
```

```
; zona 19.172.in–addr.arpa.
; inf.ioc.org  (reverse)  172.19.0.0/16
$TTL 3D
19.172.in–addr.arpa.  IN SOA ns.inf.ioc.cat. admin.inf.ioc.cat. 1 3M 1M 1W 1D
     NS  ns.inf.ioc.cat.
1.1  PTR ns.inf.ioc.cat.
1.2  PTR mailhost.inf.ioc.cat.
2.5  PTR printer.inf.ioc.cat.
2.10 PTR server.inf.ioc.cat.
```

El servidor de noms del domini inf.ioc.cat. es configura de manera anàloga a com s'han configurat els servidors en els exemples anteriors. Evidentment cal indicar quines són les zones que administra:

```
zone "inf.ioc.cat" {
  notify no;
  type master;
  file "inf.ioc.cat.zona.db";
};
zone "19.172.in–addr.arpa" {
  notify no;
  type master;
  file "inf.ioc.cat.rev.zona.db";
};
```

De fet, en el procés de delegació de zona no cal fer res d'especial en el servidor de la zona delegada. Només cal fer la configuració apropiada, posar-lo en funcionament i verificar-lo igual que es fa per a qualsevol altre servidor. On és, doncs, la màgia de la delegació? En el lligam que es fa del domini pare a la zona delegada.

Delegar una zona consisteix a indicar al fitxer de la zona pare quin és el servidor de la zona delegada.

- Això es fa amb un registre de recurs NS, indicant l'FQDN del servidor de noms de la zona delegada.
- Usualment caldrà a més un registre de recurs de tipus A, que indiqui quina és l'adreça IP del servidor de noms descrit en el punt anterior. Aquest és el registre de recurs que fa el **lligam** (*glue record*) que possibilita la "màgia" de la delegació.

Finalment, cal veure com es fa en la zona ioc.cat. per delegar. Cal afegir al fitxer de zona els dos registres de recurs que possibiliten la delegació:

```
; Fitxer de configuració dels hosts de la zona:
; ioc.org  (172.17.1.0/24)
$TTL 3D
@ IN SOA ns.ioc.cat. admin.ioc.cat. 1 3H 15M 1W 1D
          NS  ns
          MX  10 mailhost
ns        A   172.17.1.1
mailhost  A   172.17.1.2
server    A   172.17.1.10
...
; delegació de zona inf.ioc.cat.
inf.ioc.cat.    NS ns.inf.ioc.cat.
ns.inf.ioc.cat  A  172.19.1.1
```

Si analitzeu la primera de les línies de delegació, veureu que es defineix que el domini inf.ioc.cat. està gestionat per un servidor de noms anomenat ns.inf.ioc.cat. En la segona línia es fa el lligam físic que possibilita saber on és aquest servidor de noms del domini delegat. Aquesta segona línia indica que el servidor de noms és el corresponent a l'adreça IP 172.19.1.1.

Quan es pregunta al servidor de noms de la zona ioc.cat. pel *host* printer.inf.ioc.cat., el servidor busca en el seu fitxer de zona i detecta que les consultes que acaben en *inf.ioc.cat.* han de ser gestionades per un *host* anomenat ns.inf.ioc.cat. A continuació, intenta resoldre aquest nom consultant un altre cop el fitxer de zona i troba que el servidor de noms anomenat ns.inf.ioc.cat. correspon

a l'adreça IP 172.19.1.1. Quan ja sap on localitzar-lo continua la resolució (està fent una consulta recursiva) apropant-se al domini de destinació. Pregunta per *printer* al servidor ns.inf.ioc.cat. i d'aquest n'obté la resposta autoritativa.

Comprovació de delegació

Per comprovar la delegació de zona cal tenir present que primer cal verificar la zona delegada per si mateixa per tal que tot sigui correcte. Un cop és segur que funciona bé, cal verificar el lligam de la zona pare amb la zona fill. Això es pot realitzar fent consultes locals des del servidor pare de registres de la zona delegada.

Servidor delegat actuant com a esclau de la zona pare

Des del punt de vista de la zona delegada no hi ha cap lligam a la zona pare. És a dir, des del punt de vista d'un client de dins del domini inf.ioc.cat., demanar pel *host* smtp.gmail.com. o demanar pel *host* server.ioc.cat. implica el mateix procediment de resolució. Implica fer una consulta externa aplicant els mecanismes generals de resolució que s'han explicat en apartats anteriors. Sembla mentida, però el subdomini inf té tant (des)coneixement del seu domini pare com de qualsevol altre domini d'Internet.

Per tal d'implementar el traspàs de coneixement entre la zona pare i la zona delegada, cal fer del servidor de noms delegat un servidor **secundari** de la zona pare. Això és realment útil, ja que hi haurà sovint consultes des del subdomini inf referents a *hosts* del domini ioc.cat degut a que tots formen part d'una mateixa estructura organitzativa.

> Quan es tracta de subdominis dins d'una mateixa estructura organitzativa (una mateixa empresa o institució), és usual que els servidors de noms dels dominis delegats facin també de servidors **esclaus** del domini pare.
> D'aquesta manera poden respondre per si mateixos les consultes referents a tot el domini.

1.6 Transferències de zona

Els dominis d'Internet s'administren en zones, cada una gestionada per dos o més servidors de noms. Segons l'estàndard, cada zona té un servidor primari i almenys un servidor secundari. Ambdós són autoritaris per a la zona que administren. Caldrà, doncs, que aquests servidors disposin d'informació tan coherent com sigui possible, que comparteixin la mateixa informació. Això es fa mitjançant les transferències de zona.

Sovint, els servidors de noms actuen també com a memòria cau, emmagatzemant les respostes d'altres servidors per tal d'incrementar la seva eficiència. Quan emeten aquestes respostes actuen de forma no autoritària.

Sovint es confonen els conceptes relatius a la transferència de zones (servidor primaris i secundaris) amb els conceptes relacionats amb l'autoritat o no de les respostes. El següent destacat intenta aclarir cada un d'aquests termes.

Primari/secundari(s), també anomenats **master/slave(s)**: una zona és gestionada per un servidor primari (*master*) i un o més servidors secundaris (*slave*) o de seguretat.
Autoritari/no autoritari: els servidors d'una zona (el primari i els secundaris) són autoritat per a aquella zona que administren. Les respostes que emeten basant-se en la informació emmagatzemada en la memòria cau (informació procedent d'altres servidors de noms en lloc de procedir de la pròpia base de dades de zona) són no autoritàries.
Transferència de zona: la informació de la base de dades de la zona ha de ser coherent entre els servidors primari i secundaris. La transferència de zona és el mecanisme que s'estableix per fer que comparteixin la mateixa informació.

El servidor **primari** manté la base de dades de la zona i l'actualitza. Aquesta informació es copia als servidors **secundaris** utilitzant un procés de **transferència**.

Establir com i quan s'ha de fer aquesta transferència és important per proporcionar un bon servei DNS. Cal buscar un **compromís** entre **actualitzar** constantment la informació però consumir recursos excessius i no fer-ho però disposar d'informació **caducada**.

De fet, un servidor pot ser secundari per a una zona (o més d'una) i primari per a altres zones. Imaginem que el servidor ioc.cat. és primari per a la seva zona i ha delegat el domini alumnes.ioc.cat. a l'associació d'alumnes. El servidor de noms dels alumnes és primari per a la seva zona. Per agilitar les consultes al domini ioc.cat., l'IOC i els alumnes han acordat permetre que el servidor de noms de la zona alumnes.ioc.cat. faci també de secundari de la zona ioc.cat.

L'actualització de la informació entre el servidor primari i els secundaris és molt important. Establir correctament aquests **paràmetres** és un art. Un servidor amb informació antiga causa un mal funcionament a la xarxa, mentre que un servidor actualitzat constantment consumeix recursos excessius.

1.7 Extensions del protocol DNS

El protocol DNS és un protocol bastant antic, que s'ha anat adaptant als canvis que ha sofert Internet. Aquest apartat tracta els més importants, el servei amb adreces IP dinàmiques i la seguretat. El primer és degut a la manca d'adreces IPv4 i a la dificultat d'obtenir adreces IP públiques estàtiques (en la majoria dels casos són de pagament, almenys per a l'usuari final). El segon és la seguretat, on inicialment el protocol havia estat dissenyat sense tenir-la en compte.

1.7.1 Servei amb adreces IP dinàmiques

El servei de noms permet associar a una adreça IP un nom de domini. Així, per exemple, a l'adreça IP del servidor de l'usuari pere se li assigna el nom de host pere.ioc.cat en el domini de l'IOC. Ara bé, què passa si el *host* rep una IP dinàmica (de manera que la seva adreça IP pot variar)? Evidentment, en els fitxers de zona que lliguen les adreces IP amb els noms de domini cal saber prèviament l'adreça a usar i això no permetria assignar noms a adreces IP dinàmiques. Però també existeix un mecanisme anomenat DDNS o **DNS dinàmic** que permet fer actualitzacions dinàmiques en la base de dades de les zones.

> El procotocol o extensió **DDNS** (Dynamic DNS) permet realitzar **actualitzacions** en una base de dades DNS de manera dinàmica. De fet, es permeten afegir i eliminar registres de recurs dels fitxers de zona.
> Aquest protocol està descrit a l'RFC 2136 de l'IETF.

Utilitzant DDNS es poden afegir i eliminar registres a la base de dades d'una zona de manera que a mesura que assigna adreces dinàmiques als seus clients un servidor DHCP també pot anar realitzant peticions d'actualitzacions al servidor DNS per tal de que aquests clients disposin també de noms de domini.

1.7.2 Seguretat

Un dels principals problemes del protocol DNS és la seguretat, ja que és una cosa que no es va tenir en compte en el disseny inicial, on prevalia més que el disseny fos distribuït i escalable. Això va que aquest protocol es basi en la confiança que qui dona la informació és fiable i que aquesta informació és certa.

Si algú munta un servidor DNS en una xarxa privada juntament amb un servidor DHCP (que dona la configuració de xarxa als equips) al marge dels oficials que hi pugui haver en aquesta organització, i aquest servidor va ementent respostes falses, com per exemple donar unes altres IP per als bancs o el correu electrònic, és molt difícil detectar-ho. Aquest tipus d'atacs s'anomenen *Man In The Middle* (MITM). També es pot donar en el cas de xarxes WiFi obertes per a aquest propòsit (els coneguts *honey pots*) que exploten l'avarícia de la gent en obtenir alguna cosa de forma gratuïta. Aquest és un cas del caçador caçat.

Un altre cas encara més difícil de detectar és quan algun servidor DNS ha patit un atac i aquestes respostes es van actualitzant en altres servidors, fent que la informació que contenen en la seva memòria cau sigui fraudulenta. Aquest cas es coneix com **enverinament de la memòria cau DNS** (*DNS cache poisoning*, en anglès).

L'IETF va elaborar un conjunt d'especificacions conegut com a **Domain Name System Security Extensions** (DNSSEC) compatibles amb el protocol DNS.

Aquestes especificacions garanteixen l'autenticitat i la integritat de les dades DNS. No obstant, no garanteixen la confidencialitat, que, de fet, no és important ja que la informació que es transmet és de caràcter públic (les IP de cada domini).

El protocol o extensió **DNSSEC** permet garantir l'**autenticita**t i la **integritat** en les consultes a servidors DNS. Aquestes extensions estan descrites en el RFC 4033, 4034 i 4035.

Aquest sistema utilitza signatures digitals basades en criptografia de clau pública. Cada zona DNS té el seu parell de clau pública-privada amb la qual es van signant les dades de les consultes i respostes DNS a través d'un mecanisme de confiança que arriba fins als servidors arrels (cadena de confiança). La zona arrel del protocol DNS està signada des del 2010, així com la majoria de gTLD i ccTLD.

2. Instal·lació i administració de serveis de configuració automàtica de xarxa

El servei **DHCP** permet la configuració d'adreces IP, màscares, passarel·les per defecte (*gateways*) i moltes altres opcions de configuració de manera totalment dinàmica.

DHCP

DHCP és l'acrònim de Dynamic Host Configuration Protocol, en català, protocol de configuració dinàmica d'equips.

Una manera planera d'entendre el DHCP és imaginar que, en arrencar, els equips clients fan un crit per la xarxa i pregunten "Que hi ha algú?", "Qui soc jo?". El servidor del DHCP els contesta proporcionant-los tota la informació necessària perquè sàpiguen qui són i com han de configurar els seus paràmetres de xarxa.

L'administrador de xarxa té la tasca de configurar els equips que la componen. Això significa configurar els servidors, els equips clients, concentradors, encaminadors... Cada equip de la xarxa s'ha d'identificar amb l'adreça IP corresponent i la màscara de xarxa, i generalment disposarà d'un camí d'accés a Internet.

Tant els usuaris com els serveis requeriran l'accés a altres equips identificant-los pel nom de domini en lloc de fer-ho per l'adreça IP, que és més difícil de recordar. Fer això equip per equip resulta una feina feixuga i repetitiva si no es disposa de serveis de xarxa que la facilitin.

2.1 Configuració automatitzada de xarxa

El servei DHCP proporciona un mecanisme de configuració centralitzat dels equips de la xarxa. En lloc de configurar un per un els equips de xarxa amb adreces i valors estàtics, un servidor DHCP anirà assignant als equips clients els valors que els corresponguin. Aquesta assignació es fa per un període de temps finit, passat el qual caldrà renovar-la.

Els principals avantatges d'utilitzar DHCP són, d'una banda, evitar conflictes d'adreces IP (adreces repetides i adreces errònies), ja que passar equip per equip a canviar la configuració és molt més pesat i propens a l'error que fer-ho editant un sol fitxer de configuració en el servidor DHCP; i, de l'altra, que fer l'administració centralitzada representa un estalvi de temps i de feina.

Avantatges del DHCP

El servei DHCP té diversos avantatges:

- Evita errors i conflictes IP.
- Centralitza l'administració.
- Estalvia temps.
- Simplifica l'administració.

El servei DHCP simplifica l'administració de la configuració dels equips de xarxa fent-la centralitzada, dinàmica i amb concessions per períodes de temps finits.

La concessió dinàmica d'adreces IP i d'altres paràmetres de configuració de xarxa es realitza per un període de temps determinat, que varia en funció de les necessitats del client i del servidor.

Exemples d'ús del servei DHCP

Els següents són alguns exemples d'ús del servei DHCP:

- En una biblioteca que admet connexions Wi-Fi, els clients obtindran concessions per un temps reduït, per exemple, minuts.
- Un usuari d'Internet que rep al seu equip de casa una adreça IP dinàmica del seu proveïdor d'accés a Internet (ISP) tindrà una concessió que segurament serà per hores.
- En la xarxa corporativa d'una empresa que s'ha configurat dinàmicament usant DHCP, els equips rebran concessions dinàmiques per períodes de temps llargs, per exemple, de dies.

2.1.1 Configuració d'un equip de xarxa

Qualsevol equip que pertany a una xarxa requereix que es configuri amb uns paràmetres mínims, que són l'adreça IP, la màscara i la porta d'enllaç per defecte (també anomenada *gateway*). L'adreça IP identifica l'equip de manera única i la màscara permet determinar la xarxa o subxarxa en què es troba l'equip. Amb aquests dos paràmetres n'hi ha prou per tenir connectivitat en la xarxa. Si es vol disposar d'accés fora de la xarxa pròpia (per exemple, a Internet o a la resta de la xarxa corporativa) cal definir també l'encaminador o *gateway*. A part de la configuració bàsica, els equips poden necessitar (de fet, ho necessiten) més paràmetres de configuració, com, per exemple, el nom del *host*, els servidors DNS a usar, el fitxer d'iniciació per a arrencades PXE...

Tot equip de xarxa necessita disposar d'una **adreça IP** que l'identifiqui de manera única a la xarxa. Li cal també una **màscara** per poder distingir en l'adreça IP la part d'**adreça de xarxa** i la d'**adreça de *host***. Finalment, és imprescindible disposar de l'adreça de la **porta d'enllaç predeterminada** o passarel·la per defecte (*gateway*), per disposar d'accés a xarxes externes.

Exemple de configuració de xarxa d'un equip domèstic

La majoria d'usuaris disposen a casa d'un equip (o més) connectat a un encaminador (*router*) que proporciona l'accés a Internet. Aquest equip està configurat com a client DHCP i en iniciar-se rep la configuració de xarxa de l'encaminador. Podeu comprovar a casa quina configuració teniu. Una configuració d'exemple podria ser:

```
1        Adreça IP. . . . . . . . . . . . .: 192.168.1.33
2        Màscara de subxarxa . . . . . . . : 255.255.255.0
3        Porta d'enllaç predeterminada . . : 192.168.1.1
4        Servidor DHCP . . . . . . . . . . : 192.168.1.1
5        Servidors DNS . . . . . . . . . . : 80.58.61.250
6                                            80.58.61.254
```

L'inconvenient de la configuració estàtica

La configuració estàtica implica configurar els equips un a un. Fins i tot encara que es tingui accés remot als equips (per Telnet o SSH), com que cal modificar la configuració de xarxa, no es pot fer assegut des de l'equip de l'administrador, sinó que cal anar equip per equip a modificar la configuració.

Aquest procés de configuració cal que es faci per a cada equip de la xarxa. Fer-ho manualment implica configurar equip per equip sense cometre errades en teclejar les adreces i les màscares. Qualsevol canvi en l'estructura de la xarxa, com per exemple redefinir les subxarxes o modificar algunes adreces IP, significa tornar a configurar manualment els equips implicats. És evident que tota aquesta feinada

no és agradable per a l'administrador de xarxa (és molt avorrida!). Tant si la xarxa corporativa consta de pocs equips com de molts, cal una solució que permeti automatitzar la configuració de xarxa de cada equip de manera centralitzada.

Les opcions de configuració de xarxa es poden assignar a cada equip **estàticament** o es poden configurar de manera **dinàmica** utilitzant DHCP.

Com a administradors de xarxa, la gestió centralitzada que proporciona DHCP ens permet modificar la xarxa afegint, eliminant i redefinint equips amb un cost mínim.

2.1.2 Tipus d'assignacions d'adreces IP

Cada equip de xarxa té assignada una adreça IP que l'identifica de manera única dins de la xarxa. La composició de l'adreça IP i la màscara determinen la xarxa o subxarxa a la qual pertany. A més, es configuren altres paràmetres de xarxa com la porta d'enllaç predeterminada, servidors DNS... Això es pot configurar manualment anant equip per equip i introduint aquesta informació.

Quan l'adreça IP i els altres paràmetres necessaris de configuració de la xarxa es configuren equip per equip, manualment, es diu que tenen una adreça **IP estàtica**.

Reconfiguració d'una xarxa

Imagineu de quin humor estarà l'administrador d'una xarxa corporativa de 1.000 equips amb adreces estàtiques quan cal reconfigurar-la en un cap de setmana!

Quan la configuració de xarxa d'un equip no es fa manualment i localment en l'equip sinó que es fa per mitjà d'un servidor DHCP, es diu que l'equip utilitza una adreça **IP dinàmica**. Per realitzar configuracions de xarxa dinàmicament caldran un o més servidors DHCP (a manera de redundància), que proporcionaran la configuració als equips clients (els que cal configurar). Per tant, serà una estructura client/servidor. Les adreces IP dinàmiques que rep el client les podem classificar en dues categories: **assignació dinàmica de rang** i **assignació fixa**.

El servidor DHCP disposa d'un rang d'adreces que pot assignar als clients que demanen una adreça IP. Quan el servidor assigna una adreça qualsevol del rang al client (a l'atzar) es tracta d'una assignació dinàmica de rang. El client no sap quina adreça IP tindrà i no hi ha manera de predir quina se li concedirà en una futura configuració. A cada nova assignació, l'adreça IP pot ser diferent.

MAC

Cada interfície de xarxa s'identifica de manera única físicament per l'adreça MAC (*media acces control* o adreça d'accés al medi).

Una assignació fixa es produeix quan el servidor DHCP assigna sempre la mateixa adreça al client. Per assignar sempre la mateixa adreça IP al client cal que el servidor pugui identificar inequívocament el client (per l'adreça MAC). El servidor disposa d'una taula amb les correspondències entre les adreces MAC i les adreces IP fixes.

Quan la configuració de xarxa d'un equip es fa per mitjà d'un servidor DHCP es diu que utilitza una adreça IP **dinàmica**. Aquesta adreça pot variar dins d'un **rang** d'adreces disponibles del servidor DHCP o pot ser **fixa**.

DNS dinàmic

Hi ha serveis de DNS dinàmic (DDNS) que permeten assignar un nom de domini a equips amb adreça IP dinàmica.

Els avantatges de disposar d'una adreça IP fixa són que la vostra identificació a Internet (la vostra adreça IP) no varia i tothom us pot identificar sempre per la mateixa IP. Podeu proporcionar serveis a altres equips i els clients us identifiquen sempre amb la mateixa adreça sense haver de recordar en cada moment quina adreça IP teniu avui (com passa en el cas d'una IP dinàmica).

2.2 Funcionament del protocol DHCP

El protocol DHCP està descrit, com la majoria de protocols de xarxa, per un document oficial anomenat RFC. Aquest document ha sofert una evolució al llarg dels anys per anar-se adaptant a les necessitats de cada moment. Tot protocol implica un diàleg entre els equips que intervenen en un procés. Ens caldrà, doncs, analitzar quin és i com es produeix aquest diàleg. Finalment es descriurà el significat de termes tan usuals en el DHCP com *rangs*, *exclusions*, *concessions* i *reserves*.

RFC

Request for Comments (RFC) són memoràndums sobre noves investigacions, innovacions i metodologies relacionades amb les tecnologies d'Internet. Els publica l'Internet Engineering Task Force (IETF) i defineixen a escala mundial els protocols i les seves revisions. És a dir, són les publicacions oficials que descriuen els protocols.

2.2.1 Evolució del protocol DHCP

L'RFC 951 és el document base que descriu el protocol BOOTP.

El servei DHCP és un servei del tipus client/servidor que proporciona la configuració de xarxa als clients que ho sol·liciten. Proporciona els paràmetres bàsics de xarxa com l'adreça IP, la màscara de xarxa, la porta d'enllaç i altres paràmetres necessaris per a la connexió a una xarxa IP. Es tracta d'un protocol de la capa d'aplicació del model TCP/IP.

El protocol DHCP està basat en l'arquitectura de serveis client/servidor i utilitza com a transport el protocol UDP de la pila de protocols TCP/ IP. El servidor DHCP es comunica amb els clients utilitzant paquets UDP, que rep en el seu port 67 i envia al port 68 del client.

La configuració dinàmica d'equips de xarxa es va iniciar amb el protocol BOOTP (BOOT Strap Protocol o protocol d'arrencada). Era un protocol més bàsic que principalment permetia definir l'adreça IP, la màscara de xarxa i la passarel·la per defecte per al client. El BOOTP (RFC 951, any 1985) és un protocol pensat per

proporcionar automàticament la IP a clients de xarxa en el procés d'arrencada. Originàriament s'utilitzava per a estacions de treball sense disc que obtenien la configuració de xarxa del protocol BOOTP i també obtenien el nom d'un fitxer d'arrencada que s'havia de baixar per mitjà del TFTP, que usualment era el sistema operatiu.

El BOOTP va donar pas al protocol DHCP, que n'és una evolució amb moltes més prestacions. El DHCP sorgeix l'octubre de 1993 mitjançant l'RFC 1531. Ràpidament evoluciona gràcies a diversos RFC, com l'RFC 1541 (el mateix 1993), que serà substituït per l'RFC 2131, el març del 1997. Aquest document és la base del protocol DHCP actual. A grans trets, el protocol es descriu en l'RFC 2131 per a xarxes Ipv4, el conjunt d'opcions de configuració de DHCP es descriuen en l'RFC 2132 i l'especificació del DHCP per a xarxes Ipv6 és en l'RFC 8415.

RFC del DHCP

Principals RFC dedicats al DHCP:

- RFC 2131, març 1997: "DHCP: Dynamic Host Configuration Protocol"
- RFC 2132: "DHCP options"
- RFC 3396: "Encoding long options"
- RFC 4361: "Node-specific client identifiers for DHCPv4"
- RFC 8415: "DHCPv6: Dynamic Host Configuration Protocol Ipv6"

2.2.2 El model funcional del protocol DHCP

El protocol DHCP descriu el diàleg que es produeix entre client i servidor per a la concessió de configuracions IP. En una xarxa amb configuració d'equips dinàmica, un o més servidors DHCP escoltaran les peticions dels clients en el port 67. Els clients DHCP sol·licitaran al servidor DHCP una configuració IP i començarà un procés de negociació que ha d'acabar (si tot va bé) amb la concessió d'una adreça IP al client. Els servidors parlen al port 68 dels clients.

La negociació que s'estableix es pot definir a grans trets de la manera següent:

1. El client sol·licita una adreça IP (de fet, una configuració de xarxa).
2. El servidor mira les adreces IP disponibles dins del rang d'adreces dinàmiques de què disposa per concedir i n'ofereix una al client.
3. Si el client l'accepta, envia una sol·licitud al servidor per fer-la seva.
4. Si al servidor li sembla bé, accepta la petició del client i li confirma que pot utilitzar aquesta adreça IP, que l'hi concedeix per un període de temps limitat.

Ports DHCP

El protocol DHCP utilitza UDP en la capa de transport. Utilitza dos ports:

- Port 67, on escolta el servidor.
- Port 68, on escolta el client.

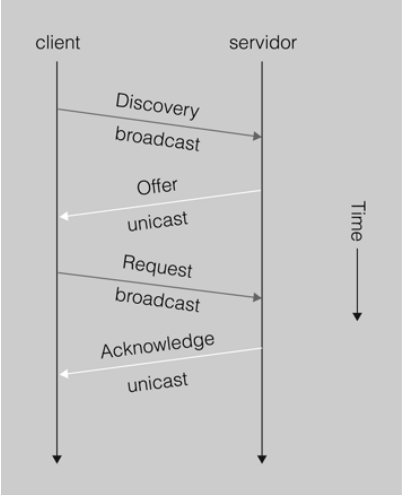
La concessió de l'adreça IP és per un període de temps establert pel servidor. Això significa que, transcorregut aquest període, el client haurà de renegociar la concessió en un procés similar al descrit anteriorment. En la figura 2.1 ("Model funcional del protocol DHCP") es pot veure el diàleg de quatre fases entre el client i el servidor.

UDP en les transmissions DHCP

L'intercanvi d'informació entre client i servidor no és gaire gran (poc volum de dades) i no requereix un flux permanent (una conversa continuada). És per això que el protocol que s'utilitza en les transmissions DHCP és l'UDP.

El procés real, però, és més detallat. El podem repassar. Consta principalment de quatre parts: la petició del client o *discovery*, l'oferta del servidor o *offer*, l'acceptació de l'adreça IP pel client o *request* i la confirmació del servidor o *acknowledgement*. A part d'aquest tipus de missatges, el protocol DHCP en defineix d'altres com el de petició d'informació o *information* i el d'alliberament de l'adreça IP o *releasing*.

FIGURA 2.1. Model funcional del protocol DHCP



Els següents són els tipus de paquets DHCP:

- *DHCP discover*
- *DHCP offer*
- *DHCP request*
- *DHCP ack* / *DHCP nack*
- *DHCP decline*
- *DHCP release*
- *DHCP information*

DHCP 'discover'

En un procés de configuració dinàmica d'un client de DHCP, el paquet *DHCP discover* és el primer que s'envia. L'envia el client per tal de demanar una configuració IP a algun servidor. Generalment, el client s'acaba d'inicialitzar i vol obtenir una configuració dinàmica de xarxa. El client no sap a quina xarxa pertany (no té adreça IP ni màscara de xarxa) ni tampoc sap quins servidors DHCP hi ha en la xarxa (si n'hi ha cap).

Una difusió o *broadcast* s'adreça a la IP 255.255.255.255 o a l'adreça MAC FF:FF:FF:FF:FF:FF, que és acceptada per tots els equips.

Per tant, el client genera un paquet de difusió (*broadcast*) destinat a tots els equips de la xarxa on sol·licita una configuració IP. En la xarxa pot haver-hi cap, un o més d'un servidor DHCP per atendre aquesta petició. És responsabilitat de l'administrador de xarxes configurar correctament l'estructura i els serveis de xarxa, de manera que si defineix clients de DHCP hi hagi servidors DHCP que atenguin les seves peticions.

DHCP 'offer'

En rebre una sol·licitud de configuració d'un client (*DHCP discover*), un servidor DHCP mira d'atendre-la proporcionant una adreça IP del rang d'adreces dinàmiques que gestiona (hi pot haver més d'un servidor DHCP en la mateixa xarxa).

El servidor tracta d'assignar una IP del conjunt o rang (també anomenat *pool*) d'adreces dinàmiques que gestiona. Per fer-ho, ha de mirar quines de les adreces li queden lliures i disponibles per concedir al client. Cada vegada que el servidor concedeix una adreça IP a un client ho anota en un fitxer de registre de les concessions efectuades. Cada vegada que finalitza una concessió el servidor pot tornar a utilitzar l'adreça IP per a un altre client.

Diversos servidors DHCP

Es pot configurar més d'un servidor DHCP, tant per a còpia de seguretat o *backup* com per incrementar el rendiment en compartir la càrrega de les peticions.

Tota **concessió** (o *lease*) DHCP és per un període determinat de temps, i un cop transcorregut cal renovar-la.

El mecanisme que utilitza el servidor per escollir l'adreça IP dins del conjunt d'adreces IP disponibles varia en funció del programa de servidor que s'utilitzi. A més, es poden configurar innumerables opcions del servidor per establir com s'han de fer les concessions. Un cas típic és el de les adreces fixes. A un determinat client se li assigna sempre la mateixa adreça IP. Per això cal disposar de la llista d'adreces MAC dels clients als quals es vol assignar una adreça IP fixa.

El servidor selecciona una adreça IP disponible i la reserva per al client (encara no està assignada). Tot seguit envia un paquet *DHCP offer* (unidestinació o *unicast*) al client amb tota la informació de configuració requerida. L'adreça IP i MAC origen identifiquen el servidor que fa l'oferta. El destinatari s'indica per la seva adreça MAC (que és coneguda). El camp IP del destinatari és l'adreça IP que el servidor ofereix (penseu que el client encara no té adreça IP). Un altre concepte important és per quant temps es realitza la concessió. El paquet inclou camps per completar la resta de configuració de xarxa, per exemple, la porta d'enllaç per defecte, els servidors DNS...

Tipus d'adreçament

Hi ha diversos tipus d'adreçament:

- Unidestinació o *unicast*: a un equip
- Multidestinació o *multicast*: a un conjunt d'equips
- De difusió o *broadcast*: a tothom

DHCP 'request'

Quan el client rep una oferta de configuració IP per part d'un servidor, la pot acceptar o rebutjar. Si el client no accepta l'oferta, simplement realitzarà un nou *DHCP discovery*. Això és suficient perquè el servidor s'adoni que l'oferta ha estat rebutjada.

Si el client accepta l'oferta, ho ha de comunicar al servidor. El mecanisme per fer-ho és mitjançant un paquet *DHCP request* enviat un altre cop per difusió. A hores d'ara, el client encara no disposa de l'adreça IP per utilitzar-la. El servidor l'ha reservat, però encara no ha donat el sí definitiu perquè sigui concedida al client.

El motiu pel qual el client demana quedar-se la concessió (*DHCP request*) que ha rebut utilitzant difusió és fer públic a tothom de la xarxa que ha acceptat una oferta d'un servidor DHCP concret. Recordeu que la petició del client es fa per difusió i, per tant, pot rebre ofertes de diferents servidors DHCP. Quan accepta

una de les ofertes, no ha de dir res als altres servidors que ha refusat. Simplement fent pública quina oferta accepta, la resta de servidors DHCP entenen que la seva oferta s'ha rebutjat.

DHCP 'acknowledgement' (DHCPACK/DHCPNACK)

L'últim pas en una negociació DHCP bàsica el realitza el servidor quan finalment autoritza la concessió enviant el paquet DHCPACK (*DHCP akcnowledgement*). A partir d'aquest moment, el client ja pot fer ús de l'adreça IP i de la configuració de xarxa rebuda. DHCPACK inclou tota la informació referent a la durada de la concessió i les dades necessàries per gestionar quan expira.

ACK i NACK

ACK i NACK són dos acrònims usuals en el món de la informàtica. Signifiquen conformitat (acceptació) i no conformitat (refús) respectivament.

El servidor anotarà en el registre de concessions la que acaba de realitzar i en detallarà tots els aspectes, en especial el temps de concessió. El paquet d'acceptació de la concessió DHCPACK és un paquet unidestinació adreçat a la MAC del client. Recordeu que el client encara no disposa d'una adreça IP vàlida; en disposarà en rebre el DHCPACK.

Exemple de mala configuració d'un equip

Un exemple de mala configuració és la d'un equip que s'ha engegat amb una adreça IP estàtica errònia que se solapa amb les adreces IP que reparteix el servidor. El mecanisme usat per comprovar si l'adreça IP ja està essent utilitzada és un *ping*. Si no respon ningú és que està lliure (segurament).

Quan un servidor DHCP detecta que la IP que havia reservat per a un client i que li anava a concedir ja està en ús (per una configuració incorrecta), el servidor envia al client un paquet DHCPNACK i indica la no autorització de la concessió. El client que rep un DHCPNACK ha de tornar a iniciar tot el procés de negociació començant un altre cop pel *DHCP discovery*.

DHCP 'decline'

Per la seva part, el client també pot examinar l'adreça IP oferta pel servidor per comprovar si està en ús o no. Pot fer altres proves per veure si li sembla correcta o no l'oferta rebuda del servidor. Per exemple, en el cas de renovació d'una adreça IP, el client pot rebre una IP diferent a la que utilitza i no interessar-li. En aquests casos, el client pot enviar un paquet *DHCP decline* al servidor per indicar que la seva oferta ha estat rebutjada.

2.2.3 DHCP 'release'

Quan un client ja no necessita més l'ús de la configuració IP que ha rebut, la pot alliberar enviant al servidor un paquet *DHCP release*. En fer-ho, el servidor afegeix l'adreça IP al conjunt d'adreces dinàmiques que té disponibles. També fa l'anotació pertinent en el registre de concessions (*leases*) per indicar que ha finalitzat l'ús de l'adreça. De totes maneres, molt sovint el client no pot arribar a emetre aquest paquet perquè és apagat per l'usuari sense deixar temps al sistema per alliberar la IP.

DHCP 'information'

En qualsevol moment el client pot sol·licitar més informació sobre la configuració de xarxa al servidor utilitzant un paquet *DHCP information*. En el paquet DHCP *offer* que el servidor envia al client, consten les informacions generals de configuració de xarxa que es trameten en l'oferta: adreça IP, màscara de xarxa, porta d'enllaç predeterminada, servidor DNS, fitxer a baixar per a arrencades PXE i molts altres paràmetres que poden estar configurats per enviar-se en l'oferta. El client pot tornar a demanar al servidor la informació d'aquests paràmetres o pot sol·licitar informació per a la configuració d'altres paràmetres (WINS, NetBIOS, *hostname*...). El client només pot realitzar una petició d'informació *DHCP information* al servidor un cop està configurat.

Petició de renovació/concessió d'una IP concreta

El procés de quatre fases usuals de DHCP consistent en *discovery* / *offer* / *request* / *ack* es produeix quan el client sol·licita una adreça IP de nou. Sabem que les concessions són per un interval de temps finit, passat el qual cal que el client en demani la renovació. Existeix, doncs, un procés de renovació simplificat. El client demana continuar usant la mateixa adreça IP amb un paquet DHCP *request* i el servidor li concedeix o no amb els paquets DHCP ACK/NACK.

Un altre cas és un client que demana usar (renovar) una adreça IP que el servidor no li pot concedir (està en ús, no és del rang que gestiona...). En aquesta situació, el servidor envia un DHCP NACK.

Macchanger

Aquesta ordre GNU/Linux permet emmascarar l'adreça MAC pròpia (*mascarade*), simular que és una altra.

2.2.4 Atacs al funcionament del DHCP

Com qualsevol altre servei de xarxa, el servidor DHCP és susceptible de patir atacs malintencionats. L'atac més fàcil i clàssic és el DoS o denegació de servei. Consisteix a inundar de peticions un servidor per tal de saturar-lo i bloquejar-ne el funcionament. Un client pot realitzar innumerables peticions *DHCP discovery* fingint que són clients diferents (emmascarant la seva MAC) amb la intenció d'esgotar les adreces IP disponibles del servidor o simplement amb la intenció de sobrecarregar-lo amb tantes peticions que no doni a l'abast a atendre-les o que ho faci lentament.

Un altre tipus d'atac consisteix a falsejar la informació que s'envia al client. Recordeu que el client fa una sol·licitud d'IP en forma de difusió (*broadcast*) i la seva petició pot ser atesa per un o més servidors DHCP. Un dels servidors DHCP pot ser un atacant que intenta proporcionar informació de configuració falsa al client, per exemple, indicant un servidor DNS també maliciós. Aquest pot falsejar les identitats de les màquines de la xarxa i que quan el client s'adreci a la seva entitat bancària el servidor DNS en realitat li proporcioni una IP d'una màquina que falseja la de l'entitat bancària.

Tipus d'atacs DNS

- Clients no autoritzats: accés a servidors DNS per part de clients no autoritzats.
- Servidors no autoritzats: servidors DNS impostors que suplanten els vertaders servidors.

Per posar remei a la inseguretat en la comunicació client/servidor DHCP, el protocol permet utilitzar mecanismes d'autenticació i xifratge. Aquests mecanismes queden fora de l'abast d'aquesta explicació.

2.2.5 Conflictes amb les adreces IP

Un dels principals motius per utilitzar DHCP és simplificar el procés de configuració de xarxa i minimitzar els conflictes per encavalcament d'adreces IP. Per desgràcia, això no garanteix que no es puguin produir conflictes. Per exemple, ens podem trobar en situacions en què dues màquines diferents tinguin la mateixa IP per una simple mala configuració del servidor DHCP. Un altre cas típic és el d'un client que s'ha configurat ell mateix una IP estàtica quan en la xarxa ja hi havia un equip que utilitzava la mateixa adreça IP assignada pel servidor DHCP.

Un problema habitual per als administradors poc experimentats és definir una configuració de xarxa local al client (*hostname*, servidor DNS, porta d'enllaç a utilitzar...), però demanar l'adreça IP dinàmicament. La configuració dinàmica no és solament la IP i la màscara sinó que el servidor DHCP pot proporcionar altres paràmetres de xarxa que sobreescriuran els que el client tenia definits localment (aquest és l'objectiu del DHCP!).

La configuració rebuda per DHCP sobreescriu la configuració local del client.

2.2.6 Rangs i concessions

Els clients DHCP obtenen del servidor una configuració de xarxa. Descrivim ara alguns dels termes que apareixen en aquest procés i que formen part de la configuració DHCP.

- **Rang**: anomenen *rang d'adreces IP* el conjunt d'adreces dinàmiques que el servidor té disponibles per assignar als clients. Les adreces IP disponibles s'agrupen per oferir-se a les diverses subxarxes que atén el servidor. Una mateixa subxarxa pot disposar de diversos rangs. Segurament s'entendrà més fàcilment amb un exemple:

```
subnet 140.220.191.0 netmask 255.255.255.0 {
            range 140.220.191.150 239.252.197.250;
}

subnet 239.252.197.0 netmask 255.255.255.0 {
            range 239.252.197.10 239.252.197.107;
            range 239.252.197.113 239.252.197.250;
}
```

En l'exemple anterior s'observa que la primera subxarxa disposa d'un rang de 101 adreces dinàmiques (de la 140.220.191.150 a la 250). La segona subxarxa permet assignar dinàmicament dos rangs d'adreces no correlatius.

- **Exclusions**: entenem per *exclusions* aquelles adreces IP que no s'ofereixen dinàmicament per part del servidor. És a dir, que no formen part de cap rang.

- **Concessions**: l'assignació d'una adreça IP i la resta de paràmetres de xarxa a un client per part del servidor és una concessió o *lease*. Els clients reben les concessions per períodes de temps finits que, en finalitzar, cal renegociar. Tant el client com el servidor s'anoten les concessions, el client la que rep i el servidor les que concedeix. Quan finalitza una concessió, el servidor pot decidir revocar-la o ampliar-la.

El client pot decidir renunciar a la concessió en qualsevol moment. Si el client vol allargar la concessió inicia un diàleg DHCP abreujat amb el servidor que pot acabar amb una renovació o amb la pèrdua de la concessió (sempre pot tornar a començar el procés). Tant el servidor com el client miren normalment les concessions que s'han efectuat entre ells amb anterioritat per tal de, si és possible, repetir la mateixa assignació.

Alliberament d'una concessió

El client pot alliberar (*release* en anglès) una concessió directament des de la línia de comandes. En un entorn Linux:

dhclient -r

En un entorn Windows:

ipconfig /release

- **Reserves**: anomenem *reserves* aquelles adreces IP que s'assignen per DHCP però de manera fixa. És a dir, són adreces que s'assignen dinàmicament però sempre i únicament a un *host* determinat. Fixeu-vos que tot i ser una adreça dinàmica només s'utilitza si el *host* associat en fa ús. Si el *host* està apagat, l'adreça no es pot usar per a altres *hosts*, està reservada. Un exemple de reserva podria ser:

```
subnet 140.220.191.0 netmask 255.255.255.0 {
       host iocserver {
           hardware ethernet 08:00:2b:4c:59:23;
           fixed-address 140.220.191.1;
       }
      range 140.220.191.150 239.252.197.250;
       }
```

En aquest exemple es pot veure que l'adreça 140.220.191.1 és una adreça reservada exclusivament per al *host* iocserver, que s'identifica mitjançant la seva adreça MAC.

2.2.7 DHCP, un servei client/servidor

Com acostuma a passar amb els serveis client/servidor, un equip pot realitzar les dues funcions al mateix temps.

El servei DHCP és un més dels serveis de xarxa que tenen l'estructura client/-servidor. Els servidors DHCP són els equips que tenen en execució el programa servidor. És el programa encarregat d'atendre les peticions dels clients i oferir-los la configuració de xarxa, tot portant el registre de les IP que concedeix i de totes les accions que realitza. Els clients DHCP són aquells equips que realitzen peticions a un servidor DHCP per obtenir una configuració de xarxa.

ISP

ISP: *Internet service provider* o proveïdor de servei/accés a Internet. Ho són, per exemple, les empreses Ono, Vodafone o Jazztel.

Client DHCP

Un equip client DHCP és un equip que sol·licita l'adreça IP i altres paràmetres de configuració de xarxa a un servidor DHCP en lloc de tenir-los definits localment en l'equip.

Si connecteu el vostre equip informàtic a la xarxa Internet per mitjà d'un ISP (*Internet service provider* o proveïdor d'accés a Internet), segurament rebreu una IP dinàmica del vostre proveïdor. Quan es realitzava una trucada telefònica amb mòdem i usant el protocol PPP (Point to Point Protocol o protocol punt a punt), el proveïdor proporcionava una adreça IP dinàmica. Si utilitzeu ADSL i un encaminador o *router*, segurament l'encaminador us proporciona una adreça IP dinàmica privada a l'ordinador de casa. Al mateix temps, l'encaminador obté una adreça IP dinàmica pública del proveïdor. Aquestes adreces IP dinàmiques són fixes (sempre les mateixes) o dinàmiques de rang (pot ser qualsevol adreça IP del conjunt d'adreces IP que té disponibles per concedir el servidor DHCP).

L'encaminador: servidor i client DHCP

Un cas típic en una xarxa privada a casa és disposar d'un encaminador ADSL connectat a un proveïdor ISP. L'encaminador actua com a client DHCP en la seva interfície de xarxa pública (la de l'ADSL), la que connecta a Internet.

Alhora, l'encaminador fa usualment de servidor DHCP per als ordinadors de casa proporcionant-los una adreça IP. En general els ordinadors dels usuaris es configuren com a clients DHCP.

IP pública / IP privada

La diferència entre una IP pública i una IP privada és que la pública és visible per a tots els equips d'Internet, mentre que la privada és visible només dins de la mateixa xarxa local.

El client DHCP ha de tenir en funcionament un dimoni encarregat de la gestió de les tasques DHCP pròpies del client. Realitza la part de negociació encarregada al client (*DHCP discovery*, *request*) i també porta un registre de les concessions (*leases*) rebudes. Aquest registre és el que utilitza el client per tornar a demanar la mateixa IP que tenia anteriorment. Un cop rebuda la concessió, el programa client queda "adormit", pendent de tornar-se a executar automàticament quan calgui renegociar la concessió. Sense intervenció de l'usuari, el programa client s'activa i segueix el procediment necessari per renegociar l'adreça IP cada cop que el temps de la concessió s'exhaureix.

Configuració client

Usualment, les configuracions client es poden fer de tres maneres diferents:

- Fitxer de text: editar directament els fitxers de configuració.
- Menús en mode text: usant algun programa de menús amb interfície de text.
- Aplicació gràfica: usant una aplicació de finestres en l'entorn gràfic.

Els programes client varien d'un sistema operatiu a un altre i la manera d'executar-los també. Generalment es disposa d'un client executable en mode text o ordres i d'una interfície gràfica (GUI, *graphics user interface* o interfície gràfica d'usuari) per a la configuració. No cal dir que els sistemes Windows tendeixen a la configuració gràfica usant finestres i a la configuració i execució interna d'amagat de l'usuari. Normalment, en els sistemes GNU/Linux la configuració es fa usant fitxers de text o opcions que es donen a ordres executables. La interfície gràfica acostuma a ser un *frontend* per cridar l'ordre. Segons sigui el sistema operatiu es pot consultar el fitxer de registre de les concessions rebudes pel client, el fitxer de *leases*, més o menys detalladament.

'Frontend'

Part que està formada per una interfície gràfica i que acostuma a recollir dades interactuant amb l'usuari. És molt comú en els entorns Unix/Linux tenir comandes molt potents amb una varietat d'opcions i disposar d'algun *frontend* per a aquella comanda perquè la interacció sigui més amigable.

Generalment, el programa client es pot configurar per definir com es comunicarà amb el servidor: informació a demanar, informació a proporcionar al servidor, opcions per defecte...

Servidor DHCP

L'administrador de xarxa és l'encarregat de pensar la ubicació del servidor o servidors DHCP en l'estructura corporativa. Com més complicada sigui la topologia de la xarxa, més difícil en serà la gestió. Una xarxa corporativa bàsica pot disposar d'un únic servidor DHCP que ofereix els seus serveis a tots els equips de la xarxa. Els clients poden estar en una mateixa subxarxa o en diverses subxarxes, però totes amb connectivitat amb el servidor DHCP. Aquest també pot ser l'esquema d'una xarxa privada a casa, on un encaminador (el de l'ISP, per exemple) proporciona el servei DHCP a tots els ordinadors de la casa.

Si la xarxa corporativa creix i passa a tenir subxarxes segmentades amb tallafocs, la configuració del servidor DHCP es complica. Si es vol continuar disposant d'un únic servidor per a tota la xarxa, caldrà que els tallafocs (*firewalls*) deixin passar els paquets DHCP entre les subxarxes i el servidor. Una altra opció és posar un servidor DHCP per a cada subxarxa o grups de subxarxes. Fent-ho així, l'administració de cada servidor és més senzilla, però hi ha més servidors a administrar. Una xarxa amb una casuística completa és la que té diversos servidors DHCP per a diverses parts de la xarxa i tallafocs entre clients i servidors que han de permetre el pas de paquets DHCP.

Si el servidor DHCP és l'encarregat de donar adreces IP als clients, qui li proporciona una adreça IP a ell? Ho fa o bé un altre servidor DHCP (i podríem tornar a fer la mateixa pregunta indefinidament) o bé l'administrador. Usualment, en una xarxa corporativa el servidor DHCP utilitza una IP estàtica definida per l'administrador. Això li permet estar sempre disponible per als clients amb la mateixa IP i no el fa dependre d'un servidor extern.

Hi ha diversos programes servidors DHCP que es poden classificar en dos grans grups: els que treballen en mode text i els que ho fan en mode gràfic. Cada administrador treballa amb les seves eines preferides. Les tasques bàsiques per aprendre a utilitzar un servidor DHCP són: observar, fer una llista de la configuració actual, activar/aturar el servei, modificar la configuració, monitorar els *logs* (registre de successos del servei) i, evidentment, saber instal·lar i desinstal·lar l'aplicació servidor.

Els fitxers de *logs* (successos) recullen els esdeveniments que es volen monitorar.

Com la majoria de serveis de xarxa, el servei DHCP s'executa en segon pla en forma de dimoni. El servidor DHCP sempre està engegat escoltant en el port 67 les peticions que rep dels clients. Quan rep una petició entrant, el programa executable del servidor DHCP la processa i posa en marxa tot el mecanisme DHCP pertinent per tornar a escoltar noves peticions. De fet, el servidor sempre escolta peticions i les processa simultàniament (segons la configuració).

Els fitxers del registre del servei, on s'anoten les concessions, mantenen la informació encara que el servei s'aturi o que el servidor s'apagui. En tornar a engegar-lo es llegiran de nou els fitxers de registres per tal de saber quines són les concessions que s'havien realitzat.

Els fitxers de concessions permeten mantenir la coherència de l'assignació d'adreces IP entre aturades del servei.

2.3 Instal·lació del servidor DHCP

El servei de xarxa DHCP està estructurat en forma de servei client/servidor; per tant, caldrà disposar del programari apropiat per interpretar cada un d'aquests rols. El programari que fa la funció de client ja està usualment integrat en el sistema operatiu. És a dir, per disposar de la part client del servei DHCP normalment no cal instal·lar res.

Així, doncs, quan parlem d'instal·lar un servei DHCP fem referència al procés d'instal·lació i configuració del programari del servidor DHCP. Evidentment també caldrà configurar els clients adequadament per fer ús del servei.

La instal·lació del programari que proporciona el servei DHCP es fa de manera molt similar (per no dir idèntica) al programari d'altres serveis de xarxa com DNS, HTTP o FTP. Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada.

Per fer-ho cal fer les reflexions i els passos següents:

1. Preguntarse: Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
2. Obtenir l'aplicació que proporciona el servei DHCP.
3. Observar l'estat de la xarxa actual. Està ja el servei en funcionament? Existeix ja una configuració DHCP activa?
4. Instal·lar l'aplicació servidor.
5. Comprovar que la instal·lació s'ha fet correctament.
6. Configurar el servei en el servidor i activar els clients perquè l'utilitzin.
7. Comprovar que el servei funciona correctament.

2.3.1 Aplicacions servidor DHCP

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha al mercat que ofereixen aquest servei. És feina seva estudiar les característiques de les diverses aplicacions, avaluar-ne l'eficiència, el cost, el que en diuen altres usuaris... La manera més fàcil de fer això és navegar per Internet, consultar les revistes especialitzades o demanar consell a informàtics experts.

Usualment, però, l'administrador acaba utilitzant l'aplicació servidor DHCP que li proporciona el mateix sistema operatiu. Si utilitzeu Windows, l'empresa Microsoft disposa d'una aplicació pròpia, però també en podeu trobar d'altres a

Internet. Igualment, si utilitzeu GNU/Linux segurament la mateixa distribució ja proporciona un servidor DHCP. De totes maneres també en podeu obtenir d'altres a Internet.

2.4 Configuració del servei

Per configurar el servei DHCP primer cal saber observar i manipular la configuració de xarxa existent, i això consisteix a:

- Fer la llista de la configuració de xarxa actual.
- Comprovar l'estat del servei de xarxa.
- Activar/desactivar el servei de xarxa.
- Monitorar el servei i el procés del servidor.

Les tasques principals per configurar un servidor DHCP són les següents:

Exemples de configuració

A GNU/Linux és molt usual que el paquet que proporciona un servei inclogui un fitxer d'exemple de configuració. El servei DHCP inclou el fitxer dhcpd.conf i la pàgina de manual del mateix nom.

- Instal·lar el programari del servidor DHCP.
- Activar/desactivar el servei del DHCP.
- Fer la llista de la configuració actual del servidor DHCP.
- Modificar la configuració del servidor DHCP.
- Monitorar els *logs* del servei DHCP i els fitxers de registre de les concessions (*leases*).

Abans d'endinsar-nos en la configuració del servei és molt útil observar una configuració ja existent. Un exemple de fitxer de configuració del servei DHCP és el que es mostra a continuació:

```
# a) opcions globals del servidor DHCP (usuals)
ddns-update-style interim;
ignore client-updates;

# b) definició de la xarxa a la qual s'ofereix el servei DHCP
subnet 192.168.0.0 netmask 255.255.255.0 {
   # opcions genèriques per a tots els equips de la xarxa
   option routers      192.168.0.1;
   option subnet-mask     255.255.255.0;
   option domain-name     "domain.org";
   option domain-name-servers  192.168.1.1;

   # definició del rang d'IP dinàmiques a usar
   # i dels temps de les concessions
   range dynamic-bootp 192.168.0.128 192.168.0.254;
   default-lease-time 21600;
   max-lease-time 43200;

  # c) opcions d'equips individuals
  # el servidor NS obté sempre una adreça fixa basada en MAC
```

```
  host ns {
   next-server marvin.redhat.com;
   hardware ethernet 12:34:56:78:AB:CD;
   fixed-address 207.175.42.254;
  }
}
```

En aquest fitxer de configuració es pot veure que hi ha tres àmbits diferents de definició:

1. **Opcions globals del servidor DHCP**. Són opcions que indiquen al servidor la manera d'actuar. També són opcions generals que cal aplicar a totes les concessions que es realitzin, independentment de la xarxa o equip.

2. **Definicions i opcions de xarxa**. Es defineixen tantes (sub)xarxes com atén el servidor. Cada definició de subxarxa consta de l'adreça IP de la xarxa i la màscara corresponent. Entre claus s'indiquen totes les opcions específiques per a les concessions de les adreces IP corresponents a la subxarxa. És habitual indicar el rang o *pool* d'adreces dinàmiques a usar, la porta d'enllaç predeterminada, el servidor de noms...

3. **Opcions d'equips individuals**. Dins d'una subxarxa es poden definir opcions per a equips individuals. Cal identificar els equips per la seva adreça MAC i, entre claus, indicar les opcions que els són específiques. Això permet assignar adreces fixes dinàmicament (equivalent al protocol BOOTP) usant les opcions de maquinari Ethernet i *fixed-address*.

Les opcions globals de configuració DHCP es poden redefinir amb valors diferents dins d'un bloc de xarxa concret. Dins d'un equip també es poden definir opcions amb valors diferents als definits per a la xarxa o globalment. Tal com passa en els llenguatges de programació, preval el valor més intern, el de *host* per damunt del de xarxa i el de xarxa per damunt del global.

2.4.1 Configuració bàsica

Per fer funcionar el servidor DHCP cal configurar-lo. Per poder arrencar li cal saber a quina xarxa donarà servei i quin és el rang d'adreces IP que pot usar dinàmicament per a les concessions als clients.

El paquet DHCP conté un fitxer d'exemple al directori /user/share/doc/dhcp*/dhcpd.conf.sample. Aquest fitxer es pot copiar a /etc/dhcpd.conf i passarà a ser la configuració bàsica del servidor DHCP. Podem veure'n el contingut fent:

```
root@server:~# ls -l /usr/share/doc/isc-dhcp-server/examples/dhcpd.conf.example
-rw-r--r-- 1 root root 3496 de des.  11  2018 /usr/share/doc/isc-dhcp-server/
    examples/dhcpd.conf.example
root@server:~# head /etc/dhcp/dhcpd.conf
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
```

```
8
9  # option definitions common to all supported networks...
10 option domain-name "example.org";
11 option domain-name-servers ns1.example.org, ns2.example.org;
12
13 default-lease-time 600;
14 root@server:~#
```

En la configuració per defecte es poden analitzar els diversos elements que es configuren:

- Opcions globals: indiquen al servidor que ignori les actualitzacions dels clients i el tipus de DDNS a usar (actualitzacions dinàmiques de DNS).
- Definició de subxarxa: cal definir tants blocs de subxarxa com subxarxes atengui el servidor DHCP.
- Opcions genèriques de subxarxa: es poden indicar opcions genèriques per als equips d'una subxarxa. Evidentment poden diferir de les opcions d'altres subxarxes.
- Les opcions principals de xarxa a descriure són l'encaminador, la màscara de xarxa, el domini...
- Les opcions principals a descriure del servei DHCP són el rang d'adreces IP dinàmiques a usar i el temps màxim de concessió.
- Perquè un *host* determinat tingui sempre la mateixa adreça IP es poden fer entrades individualitzades per a *hosts* concrets. Els *hosts* s'identifiquen per la seva adreça MAC.
- A un *host* concret se li poden aplicar opcions individualitzades, com per exemple definir el seu nom. Les opcions individuals prevalen sobre les genèriques.

2.4.2 Configuració avançada

El protocol DHCP permet configuracions d'una certa complexitat. Podeu consultar la documentació del DHCP i les pàgines del manual sobre el dimoni dhcpd i el fitxer de configuració dhcpd.conf.

Les característiques principals que s'hi descriuen són l'agrupació d'entrades en grups i classes i la possibilitat que el DHCP es comuniqui amb el DNS (actualitzacions DDNS) per crear entrades DNS quan un equip rep una configuració DHCP.

Vegeu un exemple de configuració amb opcions més avançades:

```
1  # option definitions common to all supported networks...
2  option domain-name "example.org";
3  option domain-name-servers ns1.example.org, ns2.example.org;
4
5  default-lease-time 600;
```

```
max-lease-time 7200;

ddns-update-style none;
authoritative;

subnet 10.5.5.0 netmask 255.255.255.224 {
  range 10.5.5.26 10.5.5.30;
  option domain-name-servers ns1.internal.example.org;
  option domain-name "internal.example.org";
  option routers 10.5.5.1;
  option broadcast-address 10.5.5.31;
  default-lease-time 600;
  max-lease-time 7200;
}

host fantasia {
  hardware ethernet 08:00:07:26:c0:a5;
  fixed-address fantasia.example.com;
}

# You can declare a class of clients and then do address allocation
# based on that.   The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.

class "foo" {
  match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
}

shared-network 224-29 {
  subnet 10.17.224.0 netmask 255.255.255.0 {
    option routers rtr-224.example.org;
  }
  subnet 10.0.29.0 netmask 255.255.255.0 {
    option routers rtr-29.example.org;
  }
  pool {
    allow members of "foo";
    range 10.17.224.10 10.17.224.250;
  }
  pool {
    deny members of "foo";
    range 10.0.29.10 10.0.29.230;
  }
}
```

Base de dades de concessions fetes pel servidor

El servidor desa en una base de dades local (de fet, són fitxers de text) les concessions (*leases*) que realitza. D'aquesta manera en pot seguir la pista en tot moment. Generalment les té a la memòria (per permetre'n un accés més ràpid), però en manté una còpia al disc. Si, per exemple, el sistema o el servei es reinicia, pot saber quines són les concessions que encara estan actives (i, per tant, quines adreces IP no té disponibles).

Usualment el fitxer de concessions és a /var/lib/dhcp. Vegeu-ne el contingut fent:

```
root@server:~# cat /var/lib/dhcp/dhcpd.leases
# The format of this file is documented in the dhcpd.leases(5) manual page.
# This lease file was written by isc-dhcp-4.4.1

# authoring-byte-order entry is generated, DO NOT DELETE
authoring-byte-order little-endian;

root@server:~#
```

2.5 Assignacions estàtiques i dinàmiques

Els clients de xarxa o bé tenen una configuració estàtica on es defineixen els seus paràmetres o bé reben la configuració per DHCP. El procés de configurar un client DHCP és tan senzill com activar aquesta última opció usant algun dels mètodes adients.

La configuració dels clients DHCP consisteix en el següent:

- Observar la configuració de xarxa actual del client.
- Configurar el client per rebre dinàmicament una adreça IP. Es tracta d'activar/desactivar la configuració de xarxa dinàmica o estàtica.
- Sol·licitar una nova IP al servidor DHCP.
- Fer la llista del fitxer de registre de les concessions client rebudes.
- Activar/desactivar el servei de xarxa en el client.

2.5.1 Client dinàmic

Tot equip client de xarxa necessita una configuració apropiada. Si aquesta configuració es defineix element per element en el mateix equip, s'anomena configuració estàtica. Si és així, no cal un servidor DHCP. És quan els clients reben la configuració de xarxa externament que parlem de configuració dinàmica i ens cal un servidor DHCP que la proporcioni.

La configuració del client es pot fer en mode text editant directament els fitxers, utilitzant interfícies de text o gràfiques (*applets*).

Edició dels fitxers de configuració

Es pot editar directament el fitxer de configuració de la interfície de xarxa pertinent i establir l'opció *dhcp* a la directiva *iface* per activar el client DHCP. Si per exemple es vol configurar la interfície de xarxa *enp0s3*, el fitxer hauria de tenir el següent aspecte:

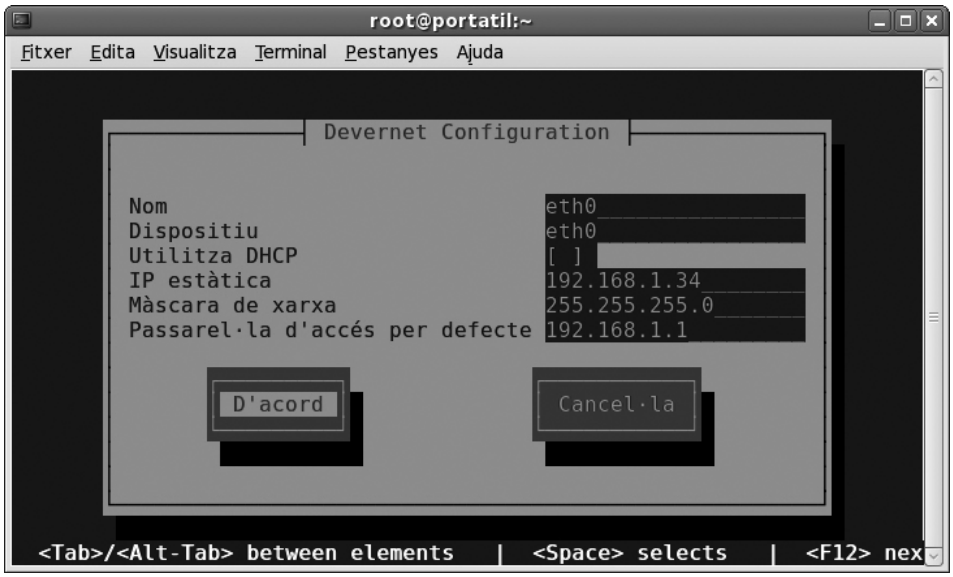
```
root@server:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet dhcp
root@server:~#
```

Menús amb interfície de text

Un altre mecanisme per activar el client DHCP és utilitzar alguna utilitat de menús en entorn de text (varien segons el sistema i se'n poden trobar a Internet), tot i que estan en força desús.

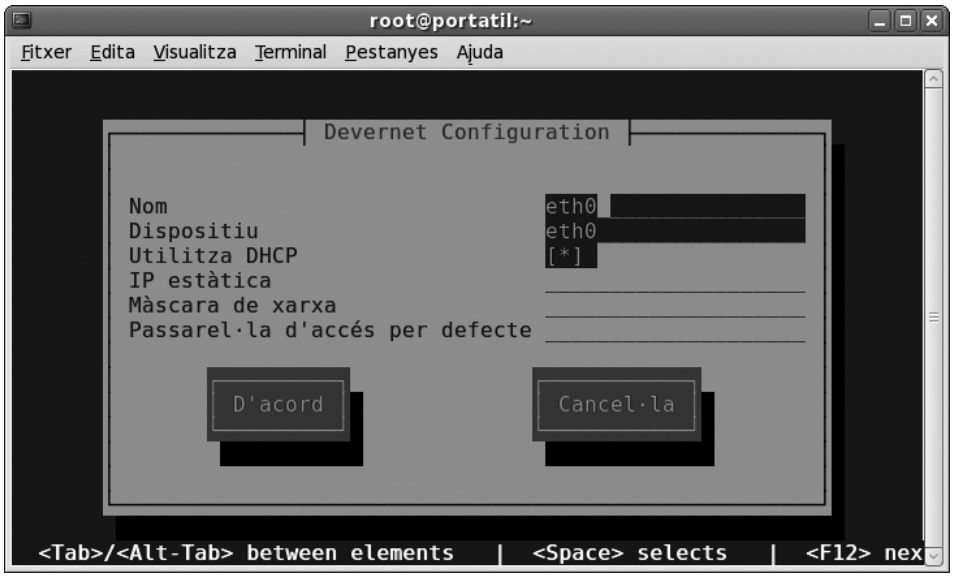
La figura 2.2 que mostra una configuració de client de xarxa estàtica i on es pot veure que la casella que permet activar el client DHCP està desactivada.

Figura 2.2. Configuració estàtica del client DHCP



El procediment per activar el client de xarxa DHCP és molt senzill. N'hi ha prou d'activar l'opció pertinent, tal com mostra la figura 2.3 ("Activació del client DHCP usant menús de text").

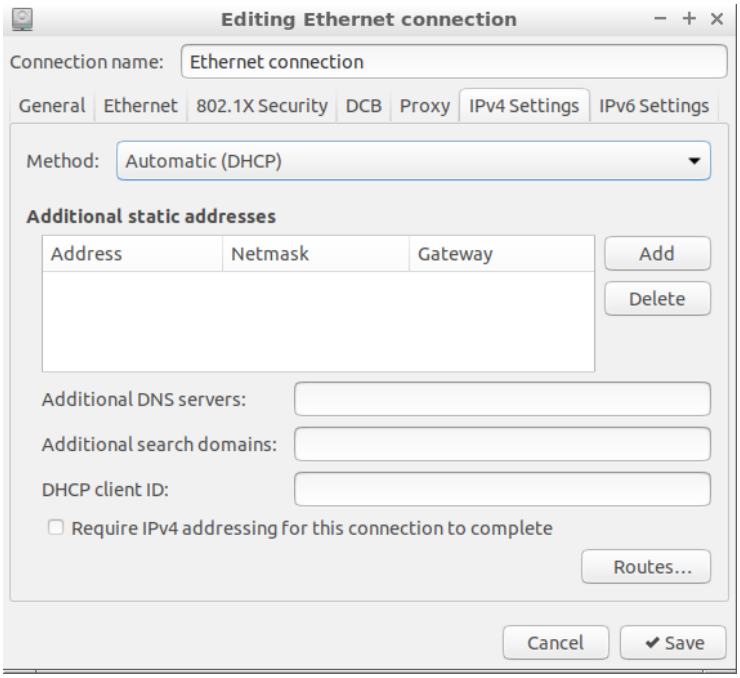
Figura 2.3. Activació del client DHCP usant menús de text



Menús en mode gràfic

En mode gràfic, el sistema també proporciona mecanismes per configurar les interfícies de xarxa i establir el mode d'activació a DHCP. En la figura 2.4 es pot observar que la configuració de la interfície té activada l'opció de configuració de xarxa per DHCP.

FIGURA 2.4. Activació del client DHCP usant l'entorn gràfic



2.5.2 Renovació de l'adreça IP

El client DHCP pot alliberar l'adreça que utilitza quan ho creu pertinent. En fer-ho, el servidor anota la fi de la concessió i si es tracta d'una adreça dinàmica de rang torna a quedar disponible per assignar-la a un altre client. Quan a un client se li està acabant el temps de concessió ha de tornar a negociar una adreça amb el servidor. De totes maneres, si el client vol, en pot tornar a sol·licitar una en qualsevol moment.

El client pot alliberar una adreça (*release*) que està en ús en qualsevol moment. Pot forçar-ho fent, per exemple:

```
root@client:~# dhclient -r -v enp0s3
Killed old client process
Internet Systems Consortium DHCP Client 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:f8:f9:29
Sending on   LPF/enp0s3/08:00:27:f8:f9:29
Sending on   Socket/fallback
DHCPRELEASE on enp0s3 to 10.0.2.3 port 67 (xid=0x3ab21a2a)
```

Per forçar el client a demanar una nova adreça per a la interfície Ethernet *enp0s3* es pot fer:

```
root@client:~# dhclient  -v enp0s3
Internet Systems Consortium DHCP Client 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s3/08:00:27:f8:f9:29
Sending on   LPF/enp0s3/08:00:27:f8:f9:29
Sending on   Socket/fallback
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0xf5a1913d)
DHCPREQUEST of 10.0.2.10 on enp0s3 to 255.255.255.255 port 67 (xid=0x3d91a1f5)
DHCPOFFER of 10.0.2.10 from 10.0.2.3
DHCPACK of 10.0.2.10 from 10.0.2.3
bound to 10.0.2.10 -- renewal in 506 seconds.
```

La comanda *dhclient* amb el paràmetre -v (*verbose*) és molt útil ja que mostra molta informació, i en cas d'error (mala configuració, error de xarxa, etc.) permet afinar molt d'on prové l'error.

2.5.3 Registre de concessions rebudes

El client DHCP porta un registre de les concessions rebudes; d'aquesta manera pot tornar a demanar una concessió abans que expiri l'actual. Aquest registre també serveix per demanar al servidor una adreça IP concreta. Les concessions o *leases* del client es desen en un fitxer anomenat /var/lib/dhcp/dhclient.leases. Podem veure'n el contingut fent:

```
root@client:~# tail /var/lib/dhcp/dhclient.leases
  option subnet-mask 255.255.255.0;
  option routers 10.0.2.1;
  option dhcp-lease-time 1200;
  option dhcp-message-type 5;
  option domain-name-servers 192.168.1.1;
  option dhcp-server-identifier 10.0.2.3;
  renew 6 2020/07/11 14:20:25;
  rebind 6 2020/07/11 14:28:39;
  expire 6 2020/07/11 14:31:09;
}
root@client:~#
```

En l'apartat "Funcionament del protocol DHCP" podeu observar detalladament com és el diàleg entre el client i el servidor.

En l'apartat "Renovar l'adreça IP" s'explica com forçar el client a demanar una nova configuració.

Podeu manipular vosaltres mateixos la captura del trànsit de xarxa DNS carregant el fitxer de captura del Wireshark que es lliura com a material complementari. Aquest fitxer el trobareu en la secció "Annexos" del web del mòdul.

2.5.4 Comprovació del funcionament

La millor manera de comprovar el funcionament del DHCP és simplement posant-lo en marxa, és a dir, creant una xarxa amb diversos clients DHCP i un servidor que els atengui. Per saber si el servei funciona cal mirar un per un cada client i comprovar que han rebut la configuració de xarxa correcta. El problema, però, és què fer si els clients no es configuren correctament.

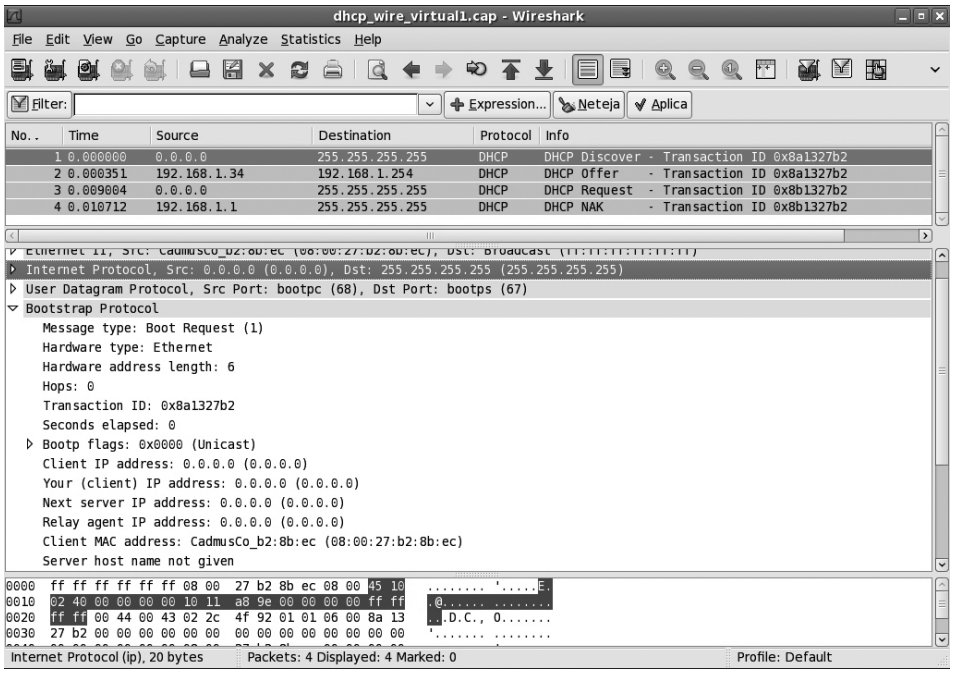
Els passos més usuals a seguir per a la resolució de problemes són:

- Comprovar que la xarxa està correctament connectada físicament, és a dir, cables, connectors, interfícies...
- Mirar si existeix connectivitat entre els equips, per exemple, usant una configuració estàtica. Això permetrà descartar que els problemes siguin deguts a altres causes. Si el DHCP no va és que no està configurat correctament.
- Repassar la configuració del client i del servidor DHCP, especialment la del servidor. Es pot començar fent la configuració tan senzilla com sigui possible. Un cop funciona es pot anar avançant en la seva complexitat.
- Examinar els fitxers de concessions, tant el del client com el del servidor, per detectar-hi anomalies.
- Quan la comunicació client/servidor no funciona correctament i no sabem per què, és molt útil monitorar el trànsit de xarxa mitjançant alguna eina d'anàlisi dels paquets que viatgen per la xarxa.

Centrem-nos, doncs, en el monitoratge del trànsit de xarxa per tal de comprovar que el diàleg entre el client i el servidor és l'apropiat. Existeixen moltes eines al mercat (que podeu trobar per Internet) que fan aquesta funció. Una de les més recomanables és Wireshark. Amb aquesta aplicació hem de poder observar l'intercanvi dels paquets *DHCP discover*, *DHCP offer*, *DHCP request* i *DHCP ack* que es produeix quan tot el procés DHCP funciona correctament. Si aquest intercanvi no es produeix és que hi ha algun problema.

En la figura 2.5 ("Captura d'un diàleg DHCP client/servidor") podeu observar una captura de trànsit DHCP feta amb Wireshark. La captura s'ha fet al servidor i s'ha forçat al client a demanar de nou una configuració de xarxa amb la utilitat *dhclient*.

Figura 2.5. Captura d'un diàleg DHCP client



servidor

2.6 Opcions addicionals de configuració

Com en la majoria de serveis actuals, la quantitat d'opcions de configuració és impressionant, per no dir aterridora. L'administrador de xarxa ha de conèixer les opcions bàsiques per configurar el servei DHCP, que inclouen l'assignació d'una configuració de xarxa bàsica, els temps de les concessions, assignacions dinàmiques i fixes, i fitxers d'arrencada via PXE.

Existeixen centenars d'opcions de configuració que permeten especificar la configuració del client fins al mínimin detall. Més important encara, existeixen diversos mecanismes per agrupar les opcions per *host*, subxarxa, classe, *pool*..., fet que premet definir "prototipus" de configuració per aplicar a *hosts* segons si tenen o no unes característiques determinades.

També existeixen extensions DHCP que permeten usar expressions i llenguatges de programació per poder realitzar configuracions complexes que permeten decidir quin tipus de configuració assignar al client.

2.6.1 Opcions de configuració del servidor i àmbit d'aplicació

Les opcions de configuració DHCP són múltiples i comprenen molts àmbits. Algunes permeten la compatibilitat amb sistemes antics, d'altres, amb altres tipus de xarxes... No és imaginable que un administrador de xarxes les conegui totes a fons. Normalment fa ús d'un conjunt reduït d'opcions que és més que suficient per administrar la majoria de xarxes.

La configuració DHCP es pot definir tant en el client com en el servidor, tot i que usualment es fa en el servidor. La tasca principal és configurar un servidor per tal de proporcionar les opcions apropiades a cada subxarxa. De totes maneres, però, un client també pot disposar d'un fitxer de configuració en el qual es defineixen quins són els seus requeriments i com ha de ser el diàleg amb el servidor. Per exemple, es defineixen quines opcions ha de sol·licitar, valors per defecte de determinades opcions (per si el servidor no en proporciona). El client també pot definir informació que proporcionarà al servidor per tal que aquest prengui decisions dinàmicament.

Caldrà, doncs, entendre quins són els àmbits (*scope*) de definició de sentències i opcions, com s'agrupen les subxarxes i els *hosts*, quines són les opcions globals, com es realitzen les definicions condicionals i molts altres detalls.

Àmbit de definició

Els clients es poden agrupar en diversos àmbits per tal de definir les opcions que han de rebre. El mateix servidor DHCP pot actuar de manera diferent segons quin

sigui l'àmbit de definició.

Alguns dels conceptes a tractar són:

- *Subnets*
- Període de concessió
- Adreces fixes o reservades: identificació de *hosts*
- PXE: protocol d'arrencada via xarxa
- Àmbit d'aplicació
- *Pool*

Les sentències d'àmbit d'aplicació més usuals són *subnet* i *host*, que permeten identificar una subxarxa i un host concret respectivament. Les subxarxes es poden agrupar en *shared-network* i els clients es poden agrupar usant la sentència ***group***. Les opcions es poden definir en funció de determinats requisits que compleixi el client mitjançant la sentència ***class*** i les **declaracions condicionals**.

El servei DHCP es pot configurar amb multitud de sentències que es poden repassar a l'RFC 2131 i a la pàgina de manual dhcpd.conf(5). Els clients DHCP reben del servidor la configuració de xarxa. Usualment parlem de l'adreça IP i la màscara, però de fet poden rebre gran quantitat de paràmetres de configuració de xarxa i informació sobre diversos serveis de xarxa disponibles. El client, per la seva part, pot sol·licitar paràmetres concrets al servidor. Quan configura el servei DHCP, l'administrador de xarxa no ha d'especificar totes les opcions possibles (de fet són moltíssimes), sinó només les que siguin necessàries per a cada client. Algunes opcions prenen valors per defecte i no cal especificar-les, d'altres no poden ser alterades pel servidor.

Una de les possibilitats que ofereix el DHCP és configurar les opcions de xarxa en funció de qui i de com és el client. És a dir, assignar al client una configuració de xarxa o una altra en funció de la informació que proporciona. Fixeu-vos que no es tracta d'entrades *host* estàtiques per a cada client, sinó que un mateix client tindrà una o altra configuració segons la informació que proporcioni.

2.7 Documentació de procediments

Una de les feines més desconegudes en el món de la informàtica és la confecció de manuals i documentació de suport. Com a clients, molt sovint ens queixem que ens falta informació o que està mal redactada. Com a administradors de xarxa, en canvi, no trobem mai temps per anotar les coses. Mentre les tenim al cap no creiem necessari fer la documentació, i després ja ens és impossible fer-ho, i sovint és just quan ens faria falta haver-ho fet.

Cal tenir clara la informació que cal documentar, tant per a l'usuari com per a l'administrador.

El client ha de saber:

- Com contactar amb el servidor DHCP. Quin programari ha d'utilitzar i com l'ha de configurar per fer ús del servei.
- Quina és la informació que obtindrà via DHCP. Cal saber consultar aquesta informació i saber què significa, per a què serveix.

La documentació de l'usuari ha de descriure el procés per activar el client DHCP, amb l'ajut de captures de pantalla. Calen també exemples de llista de concessions rebudes: on són i com es poden consultar. La part més important és mostrar un exemple de configuració de xarxa rebuda en el qual es detalli el significat de cada element i explicar a l'usuari com fer aquesta consulta.

Serveis web i de transferència de fitxers

Eduard Canet i Ricart

Índex

Introducció

Segurament els serveis més populars que estudiarem en el mòdul *Serveis de xarxa i Internet* són els serveis d'HTTP i FTP tractats en aquesta unitat. Aquests són els serveis que permeten la creació de llocs web i de servidors de descàrrega de fitxers.

El servei més popular avui en dia a Internet és el servei web, que utilitza HTTP. La seva popularitat, basada en el tractament d'hipertext que ha acabat incloent vídeo, àudio i multimèdia en general (hipermèdia), l'ha convertit en una eina a l'abast de tothom. L'ús dels navegadors web i HTTP ha eclipsat molts dels altres protocols d'Internet, que han acabat veient com les seves funcionalitats s'integraven en el servei web (els usuaris baixen fitxers pel web en lloc de per l'FTP). El servidor intermediari (*proxy server*) és un servei HTTP que proporciona capacitats de memòria cau i filtratge dels continguts web que sol·liciten els clients.

En l'apartat **"Instal·lació i administració de servidors web"** es descriuen els fonaments i protocols en els quals es basa el funcionament d'un servidor web, el protocol HTTP. S'explica la sintaxi d'aquest protocol i es descriu un diàleg petició/resposta entre un client (per exemple, un navegador) i un servidor web. També es mostra com instal·lar i configurar servidors web i s'examina la configuració per defecte.

La funcionalitat del servidor es pot ampliar a través de mòduls dinàmics. Així doncs, es mostra com activar i configurar mòduls dinàmics, com per exemple els que proporcionen SSL, PAM, estadístiques i monitorització del servidor, etc. S'explica com crear i configurar llocs web virtuals, segurament un dels elements més importants de la configuració de servidors web. Els llocs virtuals permeten disposar de múltiples llocs web en un mateix servidor.

Un altre aspecte molt important és com gestionar l'accés als llocs web, qui té o no té permís per accedir a on. En aquest apartat aprendreu a instal·lar i configurar els mecanismes d'autenticació i control d'accés del servidor. Una de les preocupacions principals a Internet és la seguretat. Es mostra com obtenir i instal·lar certificats digitals i com establir mecanismes per assegurar les comunicacions entre el client i el servidor.

També es presenta com realitzar proves de monitoratge del servei, analitzar els registres del servei per a l'elaboració d'estadístiques, resoldre incidències i generar documentació.

En l'apartat **"Instal·lació i administració de serveis de transferència de fitxers"** s'estudien els serveis FTP i TFTP, que permeten penjar i baixar fitxers en la xarxa. L'FTP utilitza TCP i proporciona fiabilitat en les transferències. Permet l'accés tant d'usuaris identificats com d'anònims. El TFTP utilitza UDP i és un mecanisme sense fiabilitat, però molt usat per a descàrregues en àrees locals. Els clients lleugers o els sistemes que s'inicien de xarxa utilitzen TFTP per transferir la informació.

Es realitza una descripció del protocol i s'analitza un diàleg complet client/servidor. També es mostra com instal·lar i configurar servidors de transferència de fitxers, examinar la configuració per defecte i personalitzar-la per tal de satisfer els requeriments del lloc FTP.

Un altre aspecte és aprendre a gestionar els usuaris i l'accés als recursos. Es mostra com crear usuaris i grups per a l'accés remot al servidor i com configurar l'accés anònim. També s'indica com establir limitacions en els diferents modes d'accés.

Així mateix, es repassa exhaustivament cada un dels modes de connexió, tant en mode actiu com en mode passiu. I també es realitzen proves amb clients en línia d'ordres i amb clients en mode gràfic. Especialment es tracta la utilització del navegador com a client del servei de transferència de fitxers.

Per oferir seguretat, integritat i confidencialitat als serveis que originàriament no en proporcionen, s'han desenvolupat tècniques com l'SSL i el TLS, que han donat lloc als serveis HTTPS i FTPS. També han sorgit protocols com l'SSH, que permeten un model de transferència d'informació xifrada.

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'estudiant:

1. Administra servidors web aplicant criteris de configuració i assegurant el funcionament del servei.
    - Descriu els fonaments i protocols en els quals es basa el funcionament d'un servidor web.
    - Instal·la i configura servidors web.
    - Amplia la funcionalitat del servidor activant i configurant mòduls.
    - Crea i configura llocs web virtuals.
    - Configura els mecanismes d'autenticació i control d'accés del servidor.
    - Obté i instal·la certificats digitals.
    - Estableix mecanismes per assegurar les comunicacions entre el client i el servidor.
    - Realitza proves de monitoratge del servei.
    - Analitza els registres del servei per a l'elaboració d'estadístiques i la resolució d'incidències.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'ús del servei.
2. Administra serveis de transferència de fitxers assegurant i limitant l'accés a la informació.
    - Estableix la utilitat i el mode d'operació del servei de transferència de fitxers.
    - Instal·la i configura servidors de transferència de fitxers.
    - Crea usuaris i grups per a l'accés remot al servidor.
    - Configura l'accés anònim.
    - Estableix limitacions en els diferents modes d'accés.
    - Comprova l'accés al servidor, tant de manera activa com passiva.
    - Realitza proves amb clients de línia d'ordres i amb clients gràfics.
    - Utilitza el navegador com a client del servei de transferència de fitxers.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'ús del servei.

1. Instal·lació i administració de servidors web

L'**HTTP** (*Hypertext Transfer Protocol* o **protocol de transferència d'hipertext**) és un protocol de capa d'aplicació que proporciona transferència de documents d'hipertext al web. El protocol HTTP és omnipresent: el World Wide Web (WWW) permet baixar hipertext, multimèdia i altres tipus de dades.

L'HTTP és un protocol sense estat i no orientat a la connexió permanent.

L'HTTP està basat en un esquema client/servidor en què el client es connecta al port 80 del servidor i fa una sol·licitud (una pàgina web, per exemple), i el servidor emet la resposta corresponent i tanca la connexió. Es tracta, per tant, d'un protocol sense estat. La connexió entre client i servidor sovint s'inicia i es tanca en cada petició/resposta. L'HTTP utilitza habitualment el protocol de transport TCP per obtenir fiabilitat en la comunicació.

L'**HTTP** és un protocol de capa d'aplicació que proporciona transferència de documents d'hipertext a la web. Utilitza un mecanisme client/servidor al port 80 basat en TCP.

L'especificació actual del protocol HTTP és la 2 (HTTP/2), descrita al document RFC 7540, de maig de 2015 (que en descriu l'estàndard), tot i que hi ha l'esborrany de la versió 3 (HTTP/3). Aquesta especificació és un alternativa a l'anterior especificació, la 1.1 (descrita al document RFC 2616, de juny de 1999) i que no deixa obsoleta. L'HTTP sorgeix als anys noranta com a protocol per transferir documents hipermèdia "en cru" per Internet (versió 0.9). La versió HTTP 1.0 (RFC 1945) permet el pas de missatges utilitzant un format tipus MIME (usat en el transport de correu). Originàriament, el contingut dels documents a transferir era text, però amb la popularització del WWW s'ha acabat convertint en un protocol de transport de contingut multimèdia i no únicament hipertext. A més, l'HTTP s'utilitza sovint com a protocol de comunicacions entre clients i altres sistemes d'Internet diferents del WWW, com per exemple NEWS, SMTP, NNTP, FTP, Ghoper, servidors intermediaris (*proxies*) i d'altres, per accedir a aquests recursos.

MIME

El *Multipurpose Internet Mail Extension*, o extensió de propòsit múltiple per al correu, és el mecanisme utilitzat per descriure el contingut dels fitxers, saber si són una imatge, un full de càlcul, un executable o altres, i permetre al navegador obrir l'aplicació pertinent.

En parlar d'HTTP ens venen immediatament al cap els navegadors web (tipus Firefox, Google Chrome, Safari...), que permeten visualitzar des d'entorns gràfics contingut d'hipertext i multimèdia, també anomenat hipermèdia. De fet, però, també existeixen navegadors en mode text (no gràfics) per a contingut únicament de text (per exemple Lynx). Habitualment usem els navegadors per obtenir contingut HTTP, però la majoria d'ells ens permeten accedir a recursos d'altres tipus.

Per accedir als documents publicats en el WWW o a documents interns de la xarxa corporativa, cal un mecanisme d'adreçament universal. L'URI (Uniform Resource Identifier o identificador uniforme de recursos) és el mecanisme d'identificació de recursos universal i té la sintaxi *schema:identifier* (esquema:identificador).

URI

Sovint s'utilitzen indistintament URI i URL, tot i que no són el mateix. Un URI es pot classificar com un URL, un URN o ambdós. L'URI permet identificar elements globalment, l'URL localitzar-los i l'URN proporciona un mecanisme d'assignació de noms únic.

L'esquema descriu la sintaxi utilitzable per l'identificador i pot ser HTTP, HTTPS, FTP o Ghoper, entre d'altres. L'identificador permet determinar el recurs concret dins d'aquest esquema. Usualment, en HTTP s'utilitza un subconjunt de l'URI anomenat URL (Uniform Resource Locator o localitzador uniforme de recursos) per localitzar un recurs. Així, en les barres de navegació trobem URL com http://www.uoc.es o ftp://ftp.rediris.es, per exemple.

La identificació de recursos es realitza mitjançant URI, URL o URN segons correspongui. La sintaxi és la següent:
URI = Uniform Resource Identifier (esquema:identificador)
URL = Uniform Resource Locator (http://www.escoladeltreball.org)
URN = Uniform Resource Name (ietf:rfc:2616)

1.1 Funcionament del servei web

El protocol HTTP estructura el diàleg client/servidor en un esquema molt bàsic de petició/resposta. Fins a la versió HTTP 1.0, cada petició/resposta implicava una connexió que s'obria i es tancava en finalitzar la resposta. Amb les millores introduïdes en la versió HTTP 1.1, s'introdueix un mecanisme de connexions persistents. La connexió establerta es pot mantenir un temps oberta per realitzar més peticions dins de la mateixa connexió (per exemple, baixar altres components de la pàgina). La versió 2 incorpora millores per tal de reduir la latència en la càrrega de pàgines web, a apart de mantenir una alta compatibilitat amb la versió anterior.

Connexions persistents

Les connexions persistents permeten que els múltiples elements d'una pàgina web (que es troben en fitxers diferents) es puguin baixar sense que calgui una connexió per a cada element.

El protocol HTTP és un protocol sense estat (*statless*). Ni client ni servidor mantenen un estat de sessió a nivell de protocol. Segurament us heu connectat a un servidor de correu web (*webmail*) i heu establert una sessió d'usuari mentre consulteu el correu. Aquesta sessió no s'implementa a nivell del protocol HTTP, sinó que és responsabilitat del desenvolupador web mantenir l'estat. Això es fa generalment utilitzant tècniques com l'ús de galetes (*cookies*), passant paràmetres per l'URL, amb camps ocults (típic dels formularis)...

L'**HTTP** és un protocol sense estat que usualment tanca la connexió per a cada petició/resposta.

1.1.1 Descripció del diàleg petició/resposta

En els diàlegs HTTP el client usualment emet una petició al servidor indicant algun dels mètodes que permet el servidor (no necessita implementar-los tots). El servidor emet una resposta i l'acompanya d'un valor d'estatus que indica el tipus de resposta (OK, error...).

Petició ('request')

El diàleg HTTP s'inicia quan un client fa una petició (usualment d'una pàgina web) a un servidor (usualment al port 80). Aquest missatge de petició consta d'una primera línia anomenada *línia de petició*, seguida de capçaleres, una línia en blanc i el cos de la petició:

Components d'una petició

Els components d'un missatge de petició HTTP són quatre:

- Línia de petició
- Capçaleres
- CRLF
- Cos

- **Línia de petició**: la primera línia d'una petició sempre té la mateixa estructura, per exemple: GET /docums/fitxa.html HTTP/1.1. El primer camp és el mètode a usar (GET significa "petició"), el camp següent és el document a obtenir i el tercer indica la versió del protocol HTTP que s'utilitza. Aquesta primera línia ha d'acabar sempre amb els caràcters CRLF.
- **Capçaleres (*headers*)**: a continuació es troben les capçaleres de la petició. Les capçaleres permeten descriure opcions del client i opcions preferibles del servidor. Per exemple, el client pot indicar el sistema operatiu i el navegador que utilitza, i el servidor ho pot tenir en compte a l'hora d'efectuar la resposta. Hi ha multitud de capçaleres i es recomana consultar el document RFC 2616 (que descriu l'estàndard HTTP) per ampliar-ne la informació. La capçalera *Host:* és obligatòria en HTTP 1.1 i indica l'URL del servidor al qual s'adreça la petició.

Diferència entre HTTP1.0 i HTTP1.1

Una de les diferències entre HTTP/1.0 i HTTP/1.1 és que en HTTP/1.1 hi ha una capçalera obligatòria (Host: <nom_servidor>) i en HTTP/1.0 no. Això li permet al servidor saber si la petició és per a ell, i permet implementar seus virtuals.

* **CRLF (línia en blanc)**: una línia en blanc separa la part de capçaleres de la petició de la part del cos. Aquest mecanisme està manllevat del format dels missatges de correu, on també s'utilitza una línia en blanc per separar les capçaleres del cos dels missatges.

- **Cos (body)**: el cos del missatge és opcional i no s'utilitza usualment en les peticions.

Mètodes de les peticions

Les peticions HTTP contenen un mètode en el primer camp de la primera línia. Aquest acostuma a ser GET o POST en les peticions, però n'hi ha més:

- **HEAD**: igual que GET però únicament sol·licita la capçalera del document. S'utilitza per comprovar l'existència del document.
- **GET**: petició al servidor per obtenir el document sol·licitat.
- **POST**: envia al servidor informació que ha d'incorporar al recurs de destinació especificat. Un ús habitual és en els formularis, on les dades es passen per POST perquè el servidor les incorpori en el document de destinació indicat.
- **PUT**: permet posar en el servidor el document indicat. En lloc de baixar un document, és un mètode per penjar un document en el servidor.

- **DELETE**: elimina el document indicat del servidor. Si es deixa, és clar.
- **TRACE**: el servidor retorna com a missatge una còpia del missatge tal com li ha arribat. És molt útil per al monitoratge del servei per part del client, ja que pot veure quines transformacions ha patit la seva petició en creuar passarel·les o *gateways*, servidors intermediaris...
- **OPTIONS**: és una sol·licitud d'informació de les opcions de transferència del servidor. El servidor contesta indicant quines són les seves capacitats.

Resposta

El servidor respon les peticions del client amb missatges que tenen una estructura similar a les peticions. Consten d'una primera línia, anomenada *línia d'estatus*, seguida de les capçaleres, una línia en blanc i la resposta, que va al final:

- **Línia d'estatus**: la primera línia d'una resposta té sempre un format com *HTTP/1.1 403 Accés prohibit*. El primer camp indica el protocol HTTP usat. El segon camp és un valor numèric de tres dígits que indica el tipus de resposta donada. Hi ha una llista exhaustiva de valors d'estatus i de significats. L'últim camp és un text descriptiu de l'estatus.
- **Capçaleres (*headers*)**: la resposta conté totes les capçaleres que el servidor consideri oportú incloure.
- **CRLF**: una línia en blanc separa les capçaleres del cos de la resposta.
- **Cos (*body*)**: aquesta part conté el contingut "real" de la resposta pròpiament dit. Així si per exemple s'ha sol·licitat una pàgina web, el contingut a mostrar es troba aquí (tota la pàgina web, no us confongueu amb les etiquetes HEADER i BODY del llenguatge HTML).

Estatus de les respostes

Les respostes contenen un primer camp amb un valor numèric d'estatus. En el document RFC 2616 se'n pot trobar la llista completa, però segons quin sigui el primer dígit es pot fer la classificació següent:

- 1xx: informació genèrica
- 2xx: acció amb èxit, *successful*
- 3xx: redirecció
- 4xx: error del client
- 5xx: error del servidor

1.1.2 Exemples de connexions HTTP

Tot el diàleg client/servidor té forma d'ordres i respostes, com en l'exemple següent de connexió HTTP 1.0 al servidor local. Vegeu com es realitza una connexió HTTP per mitjà d'un Telnet al port 80 d'un servidor HTTP per fer una petició GET d'una pàgina web:

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 07:59:09 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Mon, 07 Sep 2020 20:01:44 GMT
ETag: "a8-5aebeb015df61"
Accept-Ranges: bytes
Content-Length: 168
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Pàgina principal</title>
    </head>
    <body>
        <h1>Pàgina principal</h1>
        <p>Servidor Apache</p>
    </body>
</html>
Connection closed by foreign host.
root@server:~#
```

En l'exemple es pot fer un seguiment dels elements que intervenen en una comunicació HTTP. La petició client és una petició GET, seguida d'una línia en blanc i sense cos (el GET no en requereix). La resposta del servidor comença amb una primera línia d'estatus (el valor 200 indica "OK"), seguida de vuit capçaleres i finalment el cos. El cos de la resposta és la pàgina web HTML que visualitzarà el navegador.

Vegeu ara un exemple on el client demana al servidor quines són les ordres o mètodes que implementa:

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
OPTIONS /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 08:01:06 GMT
Server: Apache/2.4.38 (Debian)
Allow: GET,POST,OPTIONS,HEAD
Content-Length: 0
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

```
root@server:~#
```

Finalment vegeu la simulació d'una petició POST. S'ha emplenat un formulari amb uns camps (nom, cognom1 i cognom2) i aquests valors es transfereixen per POST al servidor, segurament a un script tipus CGI, JavaScript o ASP.

```
root@server:~# telnet www.ioc.cat 80
Trying 10.0.0.2...
Connected to www.ioc.cat.
Escape character is '^]'.
POST /cgi-bin/script-06.sh HTTP/1.1
Host: www.ioc.cat
Content-Type: text/html
Content-Length: 33

nom=pere&cognom1=pou&cognom2=prat
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 08:01:06 GMT
Server: Apache/2.4.38 (Debian)
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
<h1> Llistat dels arguments rebuts</h1>
<h2> POST arguments rebuts per sdtin <h2>
nom=pere&cognom1=pou&cognom2=prat

Connection closed by foreign host.
```

1.2 Instal·lació i configuració de servidors web

El protocol HTTP està estructurat en forma de servei client/servidor. Per tant, cal disposar del programari apropiat per representar cada un d'aquests rols. El programari que fa la funció de client usualment ja està disponible en el sistema operatiu amb aplicacions com, per exemple, els navegadors gràfics Firefox i Chrome o navegadors d'entorn de text com Lynx. És a dir, per disposar de la part client del servei HTTP normalment no cal instal·lar res, perquè tots els sistemes operatius proporcionen almenys un navegador.

Així, quan parlem d'instal·lar el servei HTTP fem referència al procés d'instal·lació i configuració del programari del servidor. La instal·lació del programari que proporciona el servei HTTP es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS o FTP). Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada. Senzill, oi?

Per fer això cal fer les reflexions i passos següents:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei HTTP.
- Observar l'estat de la xarxa actual. El servei està ja en funcionament? Existeix ja un servidor HTTP instal·lat i actiu?

- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i comprovar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

1.2.1 Aplicacions de servidor HTTP

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha al mercat que ofereixen aquest servei. És feina seva estudiar les característiques de les diverses aplicacions, com per exemple avaluar-ne l'eficiència, el cost, el que en diuen altres usuaris... Això es pot fer navegant per Internet, consultant les revistes especialitzades o demanant consell a un expert.

Usualment, però, l'administrador acaba utilitzant l'aplicació de servidor HTTP que li proporciona el mateix sistema operatiu. Si utilitzeu Windows, l'empresa Microsoft ofereix una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor HTTP o bé n'existeix algun de clàssic provinent d'Unix. De totes maneres en podeu obtenir d'altres a Internet.

Cerca d'HTTP a Internet

Usualment, l'administrador s'informa per mitjà del seu cercador preferit (per exemple, Google) i de webs com la Viquipèdia. Proveu de buscar "HTTP" o "HTTP server" en aquests serveis.

L'**Apache Server** és un programari de servidor HTTP omnipresent en tots els sistemes operatius avui en dia. Tot i que està basat en GNU/Linux, també és utilitzat pels sistemes operatius de Mac i Windows.

Podeu trobar tota la informació d'aquest servidor a **www.apache.org**.

1.2.2 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per Internet paquets de servidor HTTP usant eines com yum o apt-get i els repositoris de paquets apropiats segons la distribució que utilitzin. A més, sempre es pot recórrer als cercadors per localitzar el que faci falta.

Un cop instal·lat el programari caldrà identificar què s'ha instal·lat. Quins paquets i què contenen. A vegades no s'instal·laran paquets sinó fitxers .tar, el contingut dels quals també caldrà saber examinar. És important saber identificar quins dels components instal·lats corresponen a fitxers executables, quins a fitxers de configuració i quins a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant, caldrà saber gestionar l'estat del servei (engegar, aturar, recarregar...) i definir l'estat que ha de tenir en els diferents *runlevels* (nivells d'execució) del sistema.

En definitiva, el procediment d'instal·lar inclourà usualment:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo utilitzant l'eina apropiada segons quina sigui la distribució que utilitzem.
- Examinar el sistema per identificar quin programari, quins paquets, hi ha instal·lats relacionats amb el servei.
- Identificar els components del servei. Quins són els fitxers executables, quins els de configuració i quins els de documentació.
- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

1.2.3 Configuració per defecte

El servei HTTP té, en instal·lar-se, una configuració per defecte que acostuma a ser l'apropiada per a un servidor web bàsic. De vegades té els fitxers de configuració buits, de manera que caldrà editar-los abans de posar el servei en funcionament.

En qualsevol cas, cal saber identificar cadascun dels conceptes que es descriuen a continuació (es mostren els valors apropiats per al servidor Apache):

- Nom del servei: apache2, localitzat a /etc/init.d/apache2.
- Fitxer de configuració: /etc/apache2/apache2.conf.
- Directori de configuracions particulars de mòduls externs: /etc/apache2/mods-available i /etc/apache2/mods-enabled.
- Directori de treball del servidor: /etc/apache2.
- Directori de publicació del servidor: /var/www/html.
- Ubicació de fitxers d'exemple, documentació i pàgines de manual d'on poder obtenir una configuració inicial bàsica.

La configuració d'un servidor web pot ser molt senzilla o terriblement complexa, tot depèn dels objectius que ens proposem. Per publicar un senzill web estàtic no cal fer altra cosa que copiar els fitxers al directori indicat i utilitzar la configuració per defecte del servidor. Si volem utilitzar diverses seus webs virtuals amb certificats digitals per permetre connexions segures i amb contingut dinàmic, la configuració del servidor esdevé una mica més entretinguda.

La configuració del servidor web Apache s'estructura en:

- Secció 1: configuració global
- Secció 2: configuració de la seu web principal
- Secció 3: configuració de seus virtuals

Configuració global

En aquesta secció es descriuen aspectes generals del funcionament del servidor, entès com un servei (com un dimoni) del sistema. S'hi descriuen les característiques següents, entre d'altres:

- Definir l'arrel on hi ha els fitxers de configuració Apache.
- Localitzar i observar on es troba el fitxer del PID.
- Definir per quines adreces IP i ports escolta el servidor.
- Carregar els mòduls dinàmics.
- Definir l'usuari i grup amb el qual s'executa Apache.

Les principals directives del servidor de configuració global del servei són:

- **ServerRoot**: descriu el directori de treball del servei. Dins d'aquest directori és on hi ha els fitxers de configuració i on s'han generat enllaços simbòlics que permeten enllaçar amb els mòduls, el PID, els *logs* i els fitxers de configuració particulars de mòduls externs.

```
1 ServerRoot "/etc/apache2"
```

- **Include**: descriu el directori per defecte on hi ha més fitxers de configuració a incloure. En lloc de generar un fitxer de configuració molt gran, es crea un fitxer particular per a cada aspecte addicional que cal configurar. Generalment n'hi ha un per a cada mòdul extra que es carrega, per exemple per al SSL, l'LDAP...

```
1 Include conf-enabled/*.conf
```

- **Listen**: indica les adreces IP i els ports pels quals escolta el servidor. Per defecte, el servidor escolta pel port 80, corresponent al protocol HTTP, però també se sol fer pel port 8080, pel port 443 en comunicacions HTTPS i per qualsevol altre port diferent que es vulgui usar. Per defecte, escolta per totes les adreces IP del servidor.

```
Listen *:80
```

Es poden indicar plantilles per a les adreces IP i per als ports usant el caràcter *. Així, *10.0.0.1:** indica escoltar per qualsevol port per a la IP indicada. En canvi, **:8080* significa escoltar pel port 8080 per a totes les adreces IP del servidor. Es poden posar tantes directives *Listen* com facin falta.

```
Listen *:80              #escoltar pel port 80 per a totes les adreces IP
Listen 10.0.0.1:*        #escoltar per tots els ports per a aquesta IP
Listen 192.168.1.30:443  #escoltar pel port del protocol HTTPS per a l'adreça
    IP indicada
```

- **User** i **Group**: permeten definir l'usuari i el grup amb els quals s'executa el servidor. En aquest cas s'executa com a usuari Apache i grup Apache.

```
User apache
Group apache
```

Vegeu l'estructura dels directoris del servidor amb:

```
oot@server:~# tree /etc/apache2
/etc/apache2
|-- apache2.conf
|-- conf-available
|   |-- charset.conf
|   |-- javascript-common.conf
|   |-- localized-error-pages.conf
|   |-- other-vhosts-access-log.conf
|   |-- security.conf
|   '-- serve-cgi-bin.conf
|-- conf-enabled
|   |-- charset.conf -> ../conf-available/charset.conf
|   |-- localized-error-pages.conf -> ../conf-available/localized-error-pages.
    conf
|   |-- other-vhosts-access-log.conf -> ../conf-available/other-vhosts-access-
    log.conf
|   |-- security.conf -> ../conf-available/security.conf
|   '-- serve-cgi-bin.conf -> ../conf-available/serve-cgi-bin.conf
|-- envvars
|-- magic
|-- mods-available
|   |-- access_compat.load
|   |-- actions.conf
|   |-- actions.load
<retallat>
|-- mods-enabled
|   |-- access_compat.load -> ../mods-available/access_compat.load
|   |-- alias.conf -> ../mods-available/alias.conf
|   |-- alias.load -> ../mods-available/alias.load
|   |-- auth_basic.load -> ../mods-available/auth_basic.load
<retallat>
|-- ports.conf
|-- sites-available
|   |-- 000-default.conf
|   '-- default-ssl.conf
|-- sites-enabled
|   |-- 000-default.conf -> ../sites-available/000-default.conf
|   '-- default-ssl.conf -> ../sites-available/default-ssl.conf
'-- ssl
    |-- apache.crt
    '-- apache.key

```

```
7 directories, 196 files
root@server:~#
```

Configuració de la seu web principal

La segona secció del fitxer de configuració descriu les opcions de funcionament i de publicació de la seu web per defecte o principal del servidor. Les directives usades aquí afecten al web per defecte i s'hereten per a totes les altres seus web (virtuals) que es defineixin en el servidor. S'ha de tenir clar que el servidor web pot servir múltiples seus webs anomenades seus **virtuals** o *vhosts*. Existeix sempre una seu que és la seu web **principal** o per defecte, a part de les altres seus virtuals que es puguin definir.

El servidor web configura sempre almenys una seu web **principal** amb independència del fet que es configurin altres seus **virtuals**.
Les opcions definides en aquesta segona secció:

- Afecten a la seu web principal.
- Les hereten per defecte totes les altres seus webs virtuals.

S'hi descriuen les característiques següents, entre d'altres:

DocumentRoot: estableix el direcori de publicació de la seu web principal o per defecte. Els clients que es connectin a l'URL indicat per *ServerName* podran accedir al contingut de *DocumentRoot*. És dins d'aquest directori que hi haurà el típic fitxer index.html i la resta de fitxers i directoris que formen el web principal.

```
DocumentRoot "/var/www/html"
```

Directory: per a cada directori que calgui configurar es pot definir un bloc d'opcions de configuració agrupades en aquesta directva. Evidentment, les opcions afecten al directori i també s'hereten per als seus subdirectoris. Cal parar atenció en el fet que el directori a indicar és una ruta absoluta de l'arbre de directoris físic del sistema i no una ruta relativa lògica de l'estructura del web.

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Options +Includes
    XBitHack On
</Directory>
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
    XBitHack On
</Directory>
<IfModule mod_userdir.c>
    UserDir disabled
</IfModule>
```

DirectoryIndex: en l'exemple següent es pot veure com es defineixen els documents a mostrar per defecte quan se solicita un URL i no s'especifica el document.

```
DirectoryIndex index.html index.html.var
```

htaccess: a banda de les directives *Directory* es pot usar un altre mètode per establir opcions de configuració per a un directori determinat i per a tots els seus subdirectoris. Consisteix a posar un fitxer de configuració .htaccess a cada directori a configurar específicament. El fitxer conté les opcions específiques per al directori i els seus subdirectoris. Evidentment, aquest fitxer s'ha de protegir perquè no sigui descarregat pels clients.

```
AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

mime: indica com s'identifiquen els tipus MIME.

```
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
```

Logs: defineix el fitxer de registre o *logs*, el nivell dels *logs* o *loglevel* i el format en el qual s'hi han de desar les entrades.

```
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
    combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log combined
```

cgi-bin: defineix el directori que conté els scripts executables CGI i n'especifica les opcions de funcionament.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

server-status i **server-info**: activen i defineixen el funcionament del monitoritge integrat en el servidor web Apache. Permeten observar detalladament la configuració del servidor i el seu estat actual. Cal activar aquestes funcionalitats per a cada seu web de la qual es vulgui fer el seguiment.

```
<Location /server-status>
    SetHandler server-status
```

```
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat portatil localhost
</Location>
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat portatil localhost
</Location>
```

Depenent de la versió d'Apache i distribució de Linux, ens hem d'assegurar que aquests mòduls estan habilitats. Es pot consultar i habilitar amb la comanda *a2enmod*.

```
root@server:/etc/apache2# a2enmod info
Enabling module info.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:/etc/apache2# a2enmod status
Module status already enabled
root@server:/etc/apache2#
```

Configuració de seus virtuals

Les seus virtuals o *virtualhosts* es descriuen àmpliament en l'apartat "Creació i configuració de llocs web virtuals".

En aquesta secció es descriuen les seus virtuals que ha d'atendre el servidor. Cal una entrada VirtualHost per a cada web a servir.

VirtualHost: descriu una seu web virtual indicant la seva adreça IP i port associats. Es defineix el nom del servei i el directori de publicació per a aquest servei.

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/html
    ServerName www.ioc.cat
    ErrorLog logs/www.ioc.cat-error_log
    CustomLog logs/www.ioc.cat-access_log common
</VirtualHost>
```

1.2.4 Exemple de configuració bàsica

Un cop instal·lat el servidor és molt fàcil posar en funcionament la seu web principal del servidor. Simplement cal:

- Establir el lligam o *bind* amb la directiva *Listen* per indicar les IP i ports per on servir. De fet, es pot deixar el valor per defecte "*:80" si es vol atendre per totes les IP.
- Indicar el nom del servei amb la directiva *ServerName*.
- Poblar el directori de publicació amb els continguts del web.

- Assegurar-se que la resolució de noms DNS identifica correctament el nom del servei amb alguna de les IP del servidor.

Per fer proves es pot usar la resolució de noms locals via /etc/hosts:

```
root@server:~# cat /etc/hosts
192.168.1.30  ioc www.ioc.cat
# ping www.ioc.cat
```

Configuració del servidor:

```
Listen *:80
ServerAdmin root@localhost
ServerName www.ioc.cat:80
UseCanonicalName Off
DocumentRoot "/var/www/html"
```

Arrencada del servei:

```
root@server:~# service apache2 start
root@server:~# service apache2 status•
 apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
        enabled)
   Active: active (running) since Tue 2020-09-15 08:48:33 CEST; 6s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 3711 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/
       SUCCESS)
 Main PID: 3715 (apache2)
    Tasks: 55 (limit: 1138)
   Memory: 10.0M
   CGroup: /system.slice/apache2.service
           |--3715 /usr/sbin/apache2 -k start
           |--3716 /usr/sbin/apache2 -k start
           '--3717 /usr/sbin/apache2 -k start

de set. 15 08:48:32 server.ioc.cat systemd[1]: Starting The Apache HTTP Server
     ...
de set. 15 08:48:33 server.ioc.cat apachectl[3711]: AH00557: apache2:
     apr_sockaddr_info_get() failed for
de set. 15 08:48:33 server.ioc.cat apachectl[3711]: AH00558: apache2: Could not
      reliably determine the se
de set. 15 08:48:33 server.ioc.cat systemd[1]: Started The Apache HTTP Server.
root@server:/etc/apache2#
```

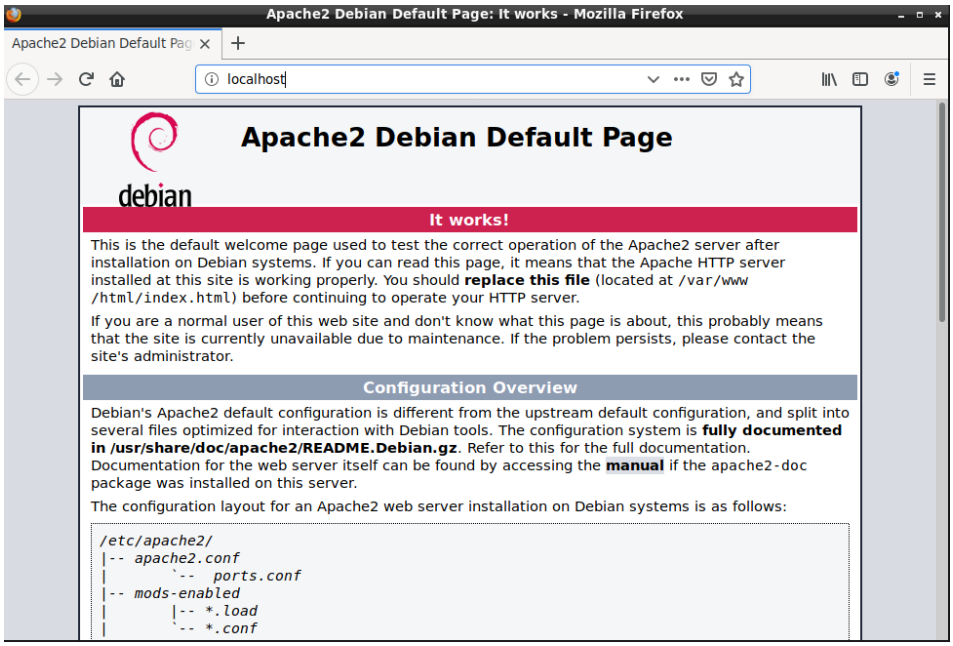
Comprovació del funcionament

Per validar el funcionament n'hi ha prou d'utilitzar qualsevol navegador client i connectar-se localment a qualsevol de les adreces IP del servidor o al *ServerName* definit per al servidor principal (l'únic configurat actualment).

```
[user@host]# telnet www.ioc.cat 80
```

Per defecte, el servidor web Apache mostra una pàgina preparada expressament per quan encara no hi ha contingut web en el directori de publicació. Aquesta pàgina serveix de test per verificar el funcionament del servidor, tal com es pot observar en la figura 1.1. Aquesta pàgina es mostra quan es contacta el servidor i encara no s'ha definit cap seu web pròpia.

FIGURA 1.1. Pàgina per defecte del servidor web Apache



Pàgina de prova pròpia

Finalment, es pot realitzar una pàgina HTML de prova pròpia per verificar que el servidor accedeix al directori de publicació i la mostra correctament. La pàgina ha de tenir el nom *index.html* o un dels noms definits com a nom de document per defecte.

El llistat de la pàgina i la seva ubicació:

```
[root@host]# cat /var/www/html/index.html
<html>
<head><title>Prova</title></head>
<body>
  <h1>Això és una prova de pàgina web</h1>
  <p>Aquí es pot escriure un paràgraf molt més interessant que aquest.<p>
</body>
</html>
```

1.3 Mòduls dinàmics

En fer la instal·lació del servidor s'han identificat els fitxers de configuració i l'executable del servei, httpd.conf i a l'httpd respectivament. Però aquests no són els únics fitxers de configuració i programari executable del servidor web. Sovint la funcionalitat del servidor web s'incrementa afegint-li noves funcions, com per exemple l'autenticació d'usuaris via PAM o LDAP, la incorporació de certificats digitals, comunicacions segures amb SSL... Cada una d'aquestes noves funcionalitats pot requerir programari addicional i noves directives de configuració.

Antigament, els fitxers de configuració creixien i creixien fins a "rebentar", cosa que dificulta la capacitat de l'administrador per governar-los i sobretot per tenir-los estructurats i fàcilment modificables. Avui en dia la majoria de serveis permeten estendre la seva funcionalitat en mòduls separats i amb fitxers de configuració que es mantenen a part i es carreguen mitjançant un *Include* en el fitxer de configuració principal.

Els **mòduls** permeten estendre la funcionalitat del servidor web proporcionant noves "peces" de programari.
La configuració del servei per mitjà de mòduls permet:

- Carregar peces de programari, mòduls encarregats de fer funcions específiques que extenen les funcionalitats del servidor web.
- Disposar de fitxers de configuració separats per a cada mòdul, cosa que facilita l'organització estructurada de la configuració.

S'han d'entendre els mòduls com un mecanisme de bocins de programari (a la manera del joc de construcció Lego) que es poden afegir i treure de la configuració actual per tal de seleccionar les prestacions i funcions que es volen proporcionar pel servidor. Podem dividir els mòduls en dues categories:

- **Estàtics**: el servidor web Apache que s'ha posat en funcionament ja té diversos mòduls carregats i executant-se des de bon principi. De fet, l'executable del servei, el dimoni httpd, s'ha compilat i se li han incorporat uns determinats mòduls (els responsables de fabricar el paquet per a la distribució que s'estigui utilitzant són qui els han seleccionat). Si es volgués disposar d'altres mòduls caldria compilar de nou l'executable del servidor.

```
# Llistat dels mòduls compilats
root@server:~# apache2 -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
root@server:~#
```

- **Dinàmics**: a part dels mòduls estàtics que incorpora el servidor es poden afegir els mòduls dinàmics o **Dinamyc Shared Objects** que calguin. Des del fitxer de configuració global es poden afegir mòduls i també es poden afegir des del directori de configuracions específiques.

```
# Llistat de mòduls carregats: estàtics i dinàmics
root@server:~# source /etc/apache2/envvars
root@server:/root# apache2 -M
Loaded Modules:
```

```
 core_module (static)
 so_module (static)
 watchdog_module (static)
 http_module (static)
 log_config_module (static)
 logio_module (static)
 version_module (static)
 unixd_module (static)
 access_compat_module (shared)
 alias_module (shared)
 auth_basic_module (shared)
 authn_core_module (shared)
 authn_file_module (shared)
 authz_core_module (shared)
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 deflate_module (shared)
 dir_module (shared)
 ...
```

És convenient saber usar les eines que proporciona el servei per interrogar-lo. Hem de ser capaços de:

- Identificar la versió del servidor.
- Identificar les opcions amb què s'ha compilat el servidor.
- Llistar els mòduls estàtics.
- Llistar els mòduls dinàmics.
- Llistar les directives actives.
- Monitorar tot el servei usant el recurs web propi *server-status*.

```
# Versió d'HTTP
root@server:~# apache2 -v
Server version: Apache/2.4.38 (Debian)
Server built:   2019-10-15T19:53:42

# Llistat de la versió de servidor i les opcions amb les quals s'ha compilat
root@server:~# apache2 -V
[Tue Sep 15 09:17:49.993943 2020] [core:warn] [pid 5272] AH00111: Config
    variable ${APACHE_RUN_DIR} is not defined
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf:
    DefaultRuntimeDir must be a valid directory, absolute or relative to
    ServerRoot
Server version: Apache/2.4.38 (Debian)
Server built:   2019-10-15T19:53:42
Server's Module Magic Number: 20120211:84
Server loaded:  APR 1.6.5, APR-UTIL 1.6.1
Compiled using: APR 1.6.5, APR-UTIL 1.6.1
Architecture:   64-bit
Server MPM:
Server compiled with....
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D DYNAMIC_MODULE_LIMIT=256
```

```
 -D HTTPD_ROOT="/etc/apache2"
 -D SUEXEC_BIN="/usr/lib/apache2/suexec"
 -D DEFAULT_PIDLOG="/var/run/apache2.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="mime.types"
 -D SERVER_CONFIG_FILE="apache2.conf"

# Llistat de les directives
root@server:~# source /etc/apache2/envvars
root@server:/root# apache2 -L
<Directory (core.c)
  Container for directives affecting resources located in the specified
      directories
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<Location (core.c)
  Container for directives affecting resources accessed through the specified
      URL paths
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<VirtualHost (core.c)
  Container to map directives to a particular virtual host, takes one or more
      host addresses
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<Files (core.c)
  Container for directives affecting files matching specified patterns
  Allowed in *.conf anywhere and in .htaccess
  when AllowOverride isn't None
...
```

1.3.1 Examinar els mòduls dinàmics

En instal·lar els paquets del servei s'han instal·lat tot de mòduls en el directori específic de mòduls del servei httpd. Usualment aquest directori és /usr/lib/apache2/modules. Es pot fer un llistat d'aquest directori per observar quins mòduls externs hi ha instal·lats en el sistema.

```
root@server:/# ls -l /usr/lib/apache2/modules | head -5
total 4040
-rw-r--r-- 1 root root  15742 Oct 15  2019 httpd.exp
-rw-r--r-- 1 root root  14384 Oct 15  2019 mod_access_compat.so
-rw-r--r-- 1 root root  14384 Oct 15  2019 mod_actions.so
-rw-r--r-- 1 root root  18480 Oct 15  2019 mod_alias.so
root@server:/#
```

Tots aquests mòduls estan carregats al servidor? No necessàriament. Estan instal·lats, però que estiguin actualment en funcionament en el servidor depèn de si s'han carregat o no des de la configuració del servidor. La directiva **LoadModule** permet carregar mòduls dinàmics des d'algun dels fitxers de configuració del servei.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

Aquest és un extracte dels mòduls carregats en el fitxer de configuració principal apache2.conf. Podem observar, per exemple, que es carreguen els mòduls d'autenticació bàsica, *digest*, *file*, LDAP...

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

```
LoadModule auth_digest_module modules/mod_auth_digest.so
```

No tots els mòduls que es carreguen s'indiquen en el fitxer de configuració principal apache2.conf. Per facilitar l'administració del servei, el fitxer de configuració es pot repartir en petits fitxers que en configurin aspectes concrets. Dividir la configuració per funcionalitats separades és molt pràctic perquè li permet a l'administrador governar cada aspecte per separat i perquè evita que el fitxer de configuració principal esdevingui un fitxer massa extens per ser manipulat amb facilitat.

Tal com s'ha pogut observar en fer la instal·lació, existeixen dos directoris, **apache2/conf-available** i **apache2/conf-enabled**, que contenen els fitxers de configuració de mòduls i de funcionalitats que s'han segregat del fitxer de configuració principal, essent el darrer el de la configuració activa. No només té fitxers de configuració de mòduls, sinó que l'administrador també pot decidir segregar en fitxers a part aquells aspectes que vol governar per separat. Això li permet la flexibilitat d'incorporar-los o no a la configuració en execució simplement incloent-los o no.

Apache proporciona un sistema disponible/actiu (available/enabled) que permet tenir diferents configuracions preparades per a configuracions, mòduls i seus virtuals que permet activar i desactivar d'una manera més còmode (en comptes d'esborrar, reanomenar, etc.). al mateix fitxer de configuració principal està explicat com a comentari, i les respectives comandes per activar i desactivar:

```
# * Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/
#   directories contain particular configuration snippets which manage modules,
#   global configuration fragments, or virtual host configurations,
#   respectively.
#
#   They are activated by symlinking available configuration files from their
#   respective *-available/ counterparts. These should be managed by using our
#   helpers a2enmod/a2dismod, a2ensite/a2dissite and a2enconf/a2disconf. See
#   their respective man pages for detailed information.
```

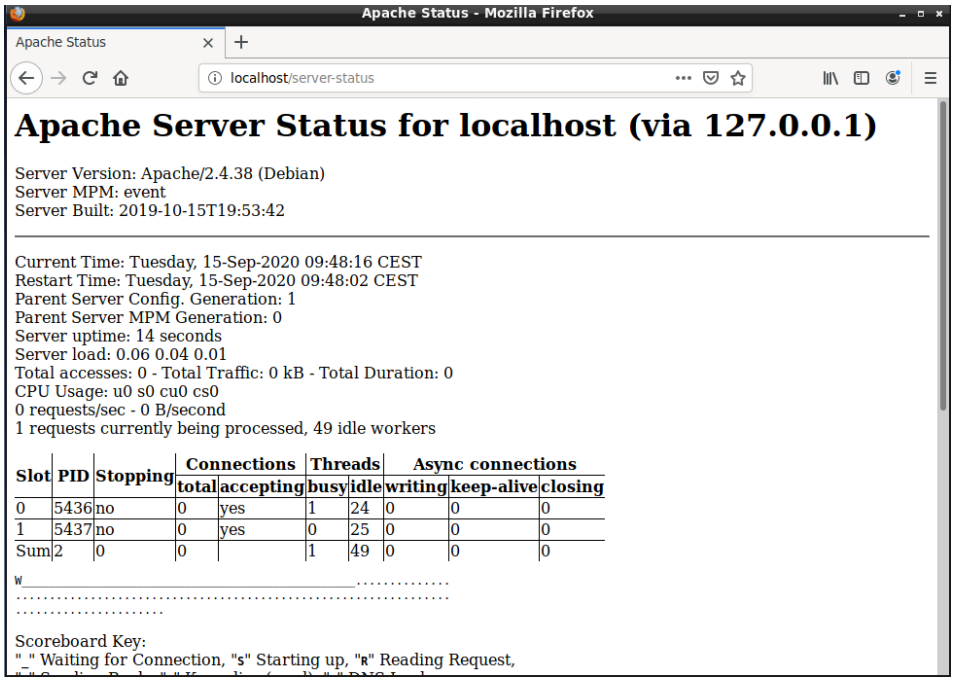
La directiva *Include* del fitxer de configuració global és l'encarregada de carregar tots els fitxers de configuració extres que hi ha al directori apache2/conf-enabled. En l'exemple ve amb la directiva *IncludeOptional* que fa el mateix que *Include*, però que si hi ha algun error en no trobar els fitxers de configuració, ho ignora.

```
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf
```

En resum, podem dir que els mòduls **instal·lats** es troben en un directori específic tipus *usr/lib/apache2/modules*. Que estiguin instal·lats no significa que estiguin carregats i en funcionament.
Els mòduls es carreguen directament des del fitxer de configuració apache2.conf mitjançant directives **LoadModule**, indicant el nom del mòdul i la seva ubicació.
També es poden carregar des de fitxers de configuració específics, situats típicament en el directori apache2/conf-enabled. Els fitxers que conté poden configurar diversos aspectes de la funcionalitat del servidor i també poden, si cal, carregar mòduls usant la directiva *LoadModule*.

El millor mecanisme per observar els mòduls carregats i la configuració de les directives i opcions que proporcionen és consultar el mateix web de monitoratge que proporciona el servidor web a l'adreça ***server-status***, tal com podeu veure en la figura 1.2.

FIGURA 1.2. Pantalla d'informació de l'estat del servidor



1.4 Creació i configuració de llocs web virtuals

El servidor web s'ha configurat mitjançant el fitxer de configuració global per escoltar per un conjunt de ports i per a un conjunt d'adreces IP amb la directiva *Listen* i ha rebut un nom mitjançant la directiva *ServerName*. Aquest és el nom amb el qual s'identifica el servei principal o per defecte. La majoria de servidors en tenen prou amb una configuració com aquesta, ja que disposen d'una sola seu web. Ara bé, el servidor pot tenir més d'una seu web, ja sigui perquè té múltiples adreces IP o perquè té una adreça IP amb múltiples seus web.

En la terminologia d'Apache s'anomena ***virtual host*** o *vhost* a cada un dels servidors virtuals que hi ha en funcionament a banda del servidor principal o per defecte.

- Quan s'assignen servidors virtuals diferents a adreces IP diferents es parla de **servidors virtuals basats en IP** o *IP-Based vhosts*.
- Quan s'assignen múltiples seus virtuals a una mateixa adreça IP es parla de **servidors virtuals basats en nom** o *Name-Based vhosts*.

Per a cada seu virtual que es defineix cal utilitzar un bloc de configuració de la directiva *VirtualHost*:

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@www.ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    ServerName ioc.cat
    ErrorLog logs/ioc.cat-error_log
    CustomLog logs/ioc.cat-access_log common
</VirtualHost>
```

Cal recordar que a part de les seus virtuals que es defineixin hi ha sempre una seu global o principal. Existeixen mecanismes per desactivar-la, però no els tractarem aquí.

Fem una anàlisi detallada de les principals opcions de configuració necessàries per definir una seu virtual:

- **VirtualHost**: aquesta directiva és la que fa el *bind*, el lligam amb l'adreça IP i port assignats a la seu virtual. Tot i que, per claredat, en l'exemple s'ha indicat un nom d'amfitrió (*host*) en lloc d'una adreça IP, Apache recomana usar sempre l'adreça IP.
- **ServerAdmin**: indica el nom de l'administrador de la seu virtual. De fet, n'indica el correu electrònic.
- **DocumentRoot**: defineix el directori de publicació de la seu web virtual. El directori que s'indica és una ruta absoluta del sistema físic de fitxers, no una ruta relativa del servidor web.
- **ServerName**: és el nom virtual amb el qual es reconeix aquest web, el nom que els clients han de referenciar per poder accedir al web.
- **errorLog** i **CustomLog**: aquestes dues directives especifiquen la ubicació dels fitxers de registre o *logs* de monitoratge de l'activitat d'aquesta seu web. Les rutes que s'hi indiquen són relatives i s'utilitza el directori de *logs* definit en la configuració global.

El problema dels fitxers de registre

En els exemples es pot observar que per a cada seu web es defineixen dos fitxers de registre (en poden ser tants com calgui). Si el servidor té en funcionament força seus

virtuals amb un trànsit de xarxa normal, pot succeir que de tants fitxers de *log* com té s'acabin esgotant els *file descriptors* del sistema.

El nombre de fitxers que pot tenir oberts un sistema és limitat. Si se supera, el sistema es bloqueja. És a dir, que realment no és difícil que això acabi provocant un problema.

Vegeu a l'apartat "Monitoratge del servei" com gestionar aquestes situacions.

L'exemple següent mostra un llistat de seus web virtuals d'un mateix servidor. Observeu el següent:

- La seu virtual www.ioc.cat està lligada a les adreces 11.0.0.3:80, 11.0.0.2:80, 11.0.0.1:80 i 10.0.0.1:80.
- La seu virtual www.inf.ioc.cat està lligada a l'adreça 10.0.0.2:80.
- L'adreça IP 10.0.0.3:80 conté diverses seus virtuals *Name-Based*. Concretament, www.virtual.cat i www.ioc-virtual.cat. La primera actua com a seu per defecte per a aquesta adreça IP.
- L'adreça IP 10.0.0.1:80 té també diverses seus virtuals *Name-Based*. Són www.ioc.cat i www.institut.cat. El servidor sempre utilitza la primera que s'ha definit en el fitxer de configuració com a seu per defecte de l'adreça IP.

```
root@server:/# apache2 -S
VirtualHost configuration:
11.0.0.3:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

10.0.0.2:80            www.inf.ioc.cat (/etc/apache2/apache2.conf:1050)

10.0.0.3:80            is a NameVirtualHost
         default server www.virtual.cat (/etc/apache2/apache2.conf:1067)
         port 80 namevhost www.virtual.cat (/etc/apache2/apache2.conf:1067)
         port 80 namevhost www.ioc-virtual.cat (/etc/apache2/apache2.conf:1075)

11.0.0.2:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

11.0.0.1:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

10.0.0.1:80            is a NameVirtualHost
         default server www.ioc.cat (/etc/apache2/apache2.conf:1022)
         port 80 namevhost www.ioc.cat (/etc/apache2/apache2.conf:1022)
         port 80 namevhost www.institut.cat (/etc/apache2/apache2.conf:1031)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
```

1.4.1 Seus virtuals basades en IP

El servidor web té la capacitat d'oferir serveis webs diferents a adreces IP diferents. De fet, pot oferir serveis diferents per a tantes combinacions IP:port com faci falta. Caldrà un bloc de configuració *VirtualHost* per a cada seu web. Si a cada una

d'aquestes diferents combinacions IP:port s'hi vol accedir amb un nom de seu web caldrà que la resolució DNS es faci apropiadament (globalment amb DNS o localment amb /etc/hosts).

Per definir servidors virtuals, *vhosts* en la terminologia Apache, basats en les adreces IP, cal usar la directiva:

```
<VirtualHost adreça_ip:port>
... configuració de la seu virtual ...
</VirtualHost>
```

adreça-IP: es recomana escriure l'adreça IP i no el nom de la seu web per indicar a quina adreça es lliga aquest *vhost*. També és vàlid escriure el nom de la seu, però això implica una doble resolució. Es pot usar el metacaràcter asterisc, (*), per indicar que s'escolta per totes les adreces IP, tot i que sembla un contrasentit, ja que precisament s'estan definint *IP-Based vhosts*. Usar l'asterisc pot tenir sentit si s'utilitza conjuntament amb ports diferents que permetin generar combinacions IP:port diferents.

port: indica el port associat al servidor virtual per l'adreça IP donada. També es pot usar el metacaràcter * per indicar qualsevol port. En aquest cas les diferents seus virtuals han de diferir d'adreça IP.

Per a cada seu virtual o **vhost** diferent que es vol implementar caldrà una directiva ***VirtualHost***.
La combinació **adreça-IP:port** permet establir l'associació de la seu virtual amb una combinació d'adreça IP més port. Es poden especificar múltiples associacions i es pot usar el metacaràcter *.

Es poden combinar múltiples expressions del tipus *adreça-IP:port* en cada sentència *VirtualHost*.

Alguns exemples de combinacions possibles són:

- **<VirtualHost 10.0.0.1:*>**: Seu virtual associada (*bind*) a qualsevol port de l'adreça 10.0.0.1.
- **<VirtualHost www.ioc.cat:*>**: Seu virtual associada a qualsevol port de l'adreça IP amb la qual es resolgui el nom www.ioc.cat.
- **<VirtualHost 10.0.0.2:80>**: Seu virtual associada exclusivament al port 80 de l'adreça IP 10.0.0.2.
- **<VirtualHost 10.0.0.2:443>**: Seu virtual associada al port 443 (el del protocol HTTPS) de l'adreça 10.0.0.2. Examinant aquest exemple i l'anterior es pot observar que es mostren seus virtuals diferents si se sol·licita l'adreça 10.0.0.2 via HTTP o via HTTPS.
- **<VirtualHost *:80>**: Seu virtual associada al port 80 de totes les adreces IP del servidor. És a dir, sigui quina sigui la IP, si és pel port 80 es mostrarà aquest *vhost*.

- **<VirtualHost *:443>**: El mateix que en l'exemple anterior, però en aquest cas associat exclusivament al port de l'HTTPS.
- **<VirtualHost *:*>**: Aquesta expressió no té sentit, ja que indica qualsevol port per a qualsevol IP. Bé, sí que té sentit, però no cal fer un amfitrió virtual per implementar aquest servei, es pot fer directament des del servei web principal.
- **<VirtualHost 10.0.0.3:80 10.0.0.3:8080 192.168.1.30:* 192.168.1.31:443>**: Estableix que aquesta seu web està associada als ports 80 i 8080 de l'adreça IP 10.0.0.3. També està lligada a qualsevol port de l'adreça IP 192.168.1.30, i finalment també està associada al port 443 de l'adreça IP 168.168.1.31.
- **<VirtualHost www.ioc.cat:80 www.ioc.cat:8080 www.xtec.cat:8080>**: Aquesta directiva lliga cada una de les adreces IP amb les quals es resolen els noms de seu web indicats i el seu port corresponent. Cal recordar que la documentació recomana usar les adreces IP en lloc dels noms d'amfitrió.

Exemple d'implementació (local) de seus virtuals basades en IP

Tot seguit implementarem tres seus virtuals "inventades" lligades a tres adreces falses en un servidor (per exemple el nostre mateix PC). Els passos a seguir són:

1. Crear les adreces IP falses. Per facilitar el monitoratge amb eines tipus Wireshark es faran les tres adreces al *loopback*.
2. Assignar noms d'amfitrió localment a cada adreça IP imitant noms de domini de seus web.
3. Crear i omplir de contingut els directoris de publicació de cada seu virtual.
4. Crear les entrades corresponents a cada *VirtualHost*.
5. Comprovar-ne el funcionament.

Creació les adreces IP falses al *loopback* i verificar-les:

```
# Creació les IP falses
root@server:/# ip address add 10.0.0.1/24 dev lo
root@server:/# ip address add 10.0.0.2/24 dev lo
root@server:/# ip address add 10.0.0.3/24 dev lo
```

Comprovem que s'han creat i que comuniquen:

```
root@server:/# ip address show lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.1/24 scope global lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.2/24 scope global secondary lo
       valid_lft forever preferred_lft forever
```

```
    inet 10.0.0.3/24 scope global secondary lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
root@server:/# ping 10.0.0.1 -c 2
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.059 ms

--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 0.019/0.039/0.059/0.020 ms
root@server:/# ping 10.0.0.2 -c 2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.061 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 21ms
rtt min/avg/max/mdev = 0.020/0.040/0.061/0.021 ms
root@server:/# ping 10.0.0.3 -c 2
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.058 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 22ms
rtt min/avg/max/mdev = 0.021/0.039/0.058/0.019 ms
root@server:/#
```

Cal configurar la resolució de noms local per assignar noms de seu web (falsos) a cada una de les adreces IP creades. Els noms de seu que s'assignen són www.ioc.cat per a la primera adreça IP, www.virtual.cat per a la segona i www.secret.cat per a la tercera.

```
root@server:/# cat /etc/hosts
10.0.0.1  www.ioc.cat
10.0.0.2  www.virtual.org
10.0.0.3  www.secret.cat
root@server:/#
```

Evidentment, si es vol disposar de diverses seus virtuals és per publicar coses diferents a cada una. Cal crear els seus directoris de publicació i posar-hi contingut. Aquest contingut ha de ser **diferent** per poder observar fàcilment amb quina seu es contacta, quina seu mostra el servidor.

Els directoris de publicació tindran el mateix nom que la seu web i es trobaran dins del directori global WWW. El contingut de cada seu web pot ser la mateixa pàgina índex amb el títol modificat per mostrar el nom de la seu web.

```
# Creació dels tres directoris de publicació
root@server:/# mkdir /var/www/{www.ioc.cat,www.virtual.cat,www.secret.cat}

# Creació de la pàgina índex per a cada seu web. Per a la primera es pot fer:
root@server:/# cat /var/www/www.ioc.cat/index.html
<html>
    <head>
        <title>Seu virtual basada en IP</title>
    </head>
    <body>
        <h1>Seu virtual basada en IP</h1>
    </body>
</html>
```

```
root@server:/#
```

Un cop està tot a punt, cal fer la configuració apropiada en el servidor. S'han d'afegir les tres seus virtuals indicant la configuració de cada una. Un cop fet això caldrà reiniciar el servei (o recarregar la configuració). Aquest és l'aspecte del fitxer de configuració apache2.conf:

```
# Seu virtual ip-based "www.ioc.cat" port 80
<VirtualHost 10.0.0.1:80>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.ioc.cat
    ServerName www.ioc.cat
</VirtualHost>

# Seu virtual ip-based "www.virtual.cat" qualsevol port
<VirtualHost 10.0.0.2:*>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.virtual.cat
    ServerName www.virtual.org
</VirtualHost>

# Seu virtual ip-based "www.secret.cat"
<VirtualHost 10.0.0.3:443>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.secret.cat
    ServerName www.secret.cat
    ... configuració SSL ...
</VirtualHost>
```

Per recarregar el servei sense aturar-lo cal fer:

```
root@server:/# service apache2 reload
```

Finalment, cal verificar el funcionament de les tres seus virtuals. Evidentment, el sistema més senzill és verificar des d'un navegador cada una de les seus web i observar que es mostra la pàgina inicial que s'ha definit per a cada seu. A continuació es mostra un altre mecanisme de verificació "en text", usant utilitats de comandes com Telnet per HTTP i Curl o OpenSSL per HTTPS:

```
# Verificar l'accés a la seu web virtual www.ioc.cat pel port 80
root@server:/# telnet 10.0.0.1 80
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 15 Sep 2020 13:38:56 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Tue, 15 Sep 2020 13:30:16 GMT
ETag: "97-5af5a26d14299"
Accept-Ranges: bytes
Content-Length: 151
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Seu virtual basada en IP</title>
    </head>
    <body>
        <h1>Seu virtual basada en IP</h1>
```

```
    </body>
</html>
Connection closed by foreign host.
root@server:/#
```

Es pot observar que el Telnet permet connectar al port 80 de l'adreça 10.0.0.1. La petició GET s'ha fet usant el protocol HTTP 1.0, de manera que no cal posar cap capçalera addicional per fer una petició de pàgina web.

En l'exemple següent es valida el funcionament de la segona seu virtual. Observeu que la petició GET s'ha fet utilitzant el protocol HTTP 1.1, que requereix obligatòriament la capçalera *Host: nomSeu*. Aquesta capçalera indica realment quina és la seu virtual a la qual es vol accedir:

```
root@server:/# telnet 10.0.0.2 80
Trying 10.0.0.2...
Connected to 10.0.0.2.
Escape character is '^]'.
GET / HTTP/1.1
host: www.virtual.cat

HTTP/1.1 200 OK
Date: Tue, 15 Sep 2020 13:38:59 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Tue, 15 Sep 2020 13:30:46 GMT
ETag: "97-5af5a26d14299"
...
```

Finalment, cal verificar el funcionament de la tercera seu virtual, que s'ha configurat per usar connexions segures HTTPS via SSL. En l'exemple no s'han inclòs totes les directives necessàries ni la gestió dels certificats digitals per tal de poder generar una seu web segura. Tot això serà tractat més endavant. Les eines **OpenSSL** i **Curl** permeten comprovar-ne el funcionament en mode text:

```
root@server:/# openssl s_client -connect www.secret.cat:443 -state -debug
GET / HTTP/1.0
...
HTTP/1.1 200 OK
```

```
root@server:/# curl https://www.secret.cat -kv
...
HTTP/1.1 200 OK
```

1.4.2 Seus virtuals basades en nom

El servidor web pot oferir serveis webs diferents associats a una mateixa adreça IP. Això vol dir que una adreça IP pot tenir més d'una seu web associada. De fet, en pot tenir tantes com facin falta. Igual que passa amb les seus virtuals basades en nom també caldrà un bloc de configuració *VirtualHost* per a cada seu web a publicar. Si a cada una d'aquestes **seus virtuals basades en nom** s'hi vol accedir amb un nom de seu web caldrà que la resolució DNS es faci apropiadament (globalment amb DNS o localment amb /etc/hosts).

Per tant, cal que la petició HTTP tingui una capçalera *Host: nomSeu*, que determina quina de les seus associades es demana. Si la petició HTTP no conté aquesta capçalera, el servidor web es veu incapaç de determinar la seu virtual i contacta amb la seu per defecte associada a l'adreça IP:port (el primer dels *vhosts* definits per a cada ip:port és la seu per defecte).

El protocol **HTTP 1.0** no utilitza la capçalera *host*. Per tant, no permet diferenciar entre diverses seus virtuals d'una combinació IP:port. En aquest cas es contacta amb la seu per defecte.
El protocol **HTTP 1.1** requereix obligatòriament la capçalera **Host: nomSeu** per indicar a quina seu s'intenta accedir. Aquesta capçalera és la que determina a quin servei web s'accedirà. Aquest nom ha de coincidir amb un dels *ServerName* declarats.

Exemples d'implementació de seus virtuals basades en nom

Alguns exemples d'implementació són els següents:

- Diverses seus web basades en nom sobre una única adreça IP.

```
# Apache escolta al port 80
Listen 80

<VirtualHost *:80>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
    # Altres directives...
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/www/exemple2"
    ServerName www.exemple.org
    # Altres directives...
</VirtualHost>
```

- Diverses seus web basades en nom sobre amb més d'una adreça IP.

```
Listen 80

# El servidor principal s'executa a 172.20.30.40
ServerName server.exemple.com
DocumentRoot "/www/mainserver"

<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
    # Altres directives...
</VirtualHost>

<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple2"
    ServerName www.exemple.org
    # Altres directives...
</VirtualHost>
```

- Diferents seus web en diferents ports.

```
Listen 80
Listen 8080

<VirtualHost 172.20.30.40:80>
    ServerName www.exemple.com
    DocumentRoot "/www/domini-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.exemple.com
    DocumentRoot "/www/domini-8080"
</VirtualHost>

<VirtualHost 172.20.30.40:80>
    ServerName www.exemple.org
    DocumentRoot "/www/domini-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.exemple.org
    DocumentRoot "/www/domini-8080"
</VirtualHost>
```

- Combinació de seus web i ports.

```
Listen 172.20.30.40:80
Listen 172.20.30.40:8080
Listen 172.20.30.50:80
Listen 172.20.30.50:8080

<VirtualHost 172.20.30.40:80>
    DocumentRoot "/www/exemple1-80"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    DocumentRoot "/www/exemple1-8080"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.50:80>
    DocumentRoot "/www/exemple2-80"
    ServerName www.exemple.org
</VirtualHost>

<VirtualHost 172.20.30.50:8080>
    DocumentRoot "/www/exemple2-8080"
    ServerName www.exemple.org
</VirtualHost>
```

- Combinació de seus web basades en nom i en IP.

```
Listen 80

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple2"
```

```
    ServerName www.exemple.org
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple3"
    ServerName www.exemple.net
</VirtualHost>

# IP-based
<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple4"
    ServerName www.exemple.edu
</VirtualHost>

<VirtualHost 172.20.30.60>
    DocumentRoot "/www/exemple5"
    ServerName www.exemple.gov
</VirtualHost>
```

Es poden trobar molts més exemples a la documentació oficial d'Apache.

1.5 Autenticació

Fins ara hem vist com crear i configurar diverses seus web accessibles per tothom que tingui accés al servidor. Hi ha ocasions en que es vol restringir l'accés a una part del web o a tot el web però només per a uns usuaris concrets. En aquest apartat es descriuen diverses formes de realitzar-ho.

El servei web incorpora mecanismes bàsics per verificar els usuaris que volen accedir a àrees restringides. Però a més a més la flexibilitat dels mòduls fa que es puguin afegir nous mecanismes que puguin sorgir més endavant tot i que no hàgin estat desenvolupats per Apache. Així, per validar l'accés a un directori amb material dels professors en un web d'una escola segurament n'hi ha prou amb el mecanisme bàsic de verificació d'usuaris i grups. En canvi, per accedir a un web ultrasecret d'una agència governamental potser cal incorporar mecanismes addicionals, basats per exemple en l'empremta òptica i el registre de veu.

Primerament cal analitzar els mecanismes de validació d'usuaris generals que permet el servidor web:

- **Autenticació bàsica amb fitxers**: el mecanisme més simple per implementar el control d'accés a recursos d'una seu web és utilitzar fitxers d'**usuaris** i **grups** propis del servidor. Apache proporciona eines per crear-los. L'avantatge principal d'aquest mètode és la facilitat d'administració. L'inconvenient és que comporta una gestió diferenciada dels usuaris del servei web i dels del sistema. De fet, això pot ser un inconvenient o un avantatge, si el que interessa és tenir-los segregats.
- **Autenticació mitjançant PAM**: en els sistemes GNU/Linux actuals l'autenticació dels usuaris es realitza via PAM (*Pluggable Authentication Module*). El PAM comprovarà el directori /etc/passwd, el LDAP, el Kerberos, les empremtes dactilars o el que calgui. Usar el lligam amb el mòdul del PAM

és un bon mecanisme per validar els usuaris del servei web igual que es validen els usuaris del sistema.

- **Autenticació mitjançant LDAP**: un dels mecanismes més populars actualment per a l'autenticació (i per a altres tasques) és el LDAP. Usar el mòdul del LDAP permet passar la validació dels usuaris a l'encarregat de gestionar l'autenticació LDAP dels usuaris del sistema. També es pot tenir en funcionament un servei LDAP específic per a les validacions del servei web.

Tot seguit es descriuen alguns conceptes clau relacionats amb el control d'accés al servidor:

- **Autenticació**: el procés d'autenticació és el que determina si un usuari és qui diu ser. En cap moment governa quins drets té, què pot fer i què no, simplement s'encarrega de comprovar que l'usuari és qui diu que és. Per implementar l'autenticació hi ha innumerables sistemes, des dels fitxers d'usuaris i contrasenyes fins a sofisticats mecanismes d'empremtes dactilars, òptiques, dades biomètriques o llapis USB (sense el llapis l'usuari no es pot identificar).
- **Autorització**: un cop s'ha identificat un usuari (és qui diu ser), què pot fer?, a quins recursos pot accedir?, a quins no? Això és l'autorització: determinar els drets d'utilització dels recursos.
- **Recurs amb accés restringit**: el control d'accés al servidor busca determinar quins recursos són accessibles per quins usuaris. Pot restringir l'accés a tota una seu web de manera que només els usuaris autoritzats puguin accedir als seus continguts. Sovint es restringeixen àrees concretes de la seu web, per exemple directoris que són accessibles només per un conjunt d'usuaris (els empleats, o els professors, en el web de l'escola). En aquest cas parlem de directoris amb accés restringit.
- **Reialme**: en una seu web hi poden haver diverses àrees restringides a perfils d'usuari diferents. Els reialmes permeten definir quines àrees restringides comparteixen el mateix grau d'accés. Tornem a l'exemple d'una seu web d'una escola on hi ha tot de continguts públics accessibles per tothom. El directori Notes és un recurs restringit on només hi poden accedir els alumnes de l'escola. Els directoris Programacions i Registres de Treball són accessibles només pels professors. Un professor que, per exemple, s'autentica per entrar a l'àrea Programacions introduïnt el seu identificador d'usuari i contrasenya, si vol entrar a l'àrea Registres de Treball s'hauria de tornar a identificar entrant de nou l'usuari i la contrasenya. Els reialmes permeten declarar que diversos llocs restringits tenen el mateix nivell d'accés, de manera que si un usuari s'ha autenticat en un està autenticat en tots els recursos que formen part del reialme.
- **Web amb inici de nom d'usuari/contrasenya**: un error molt típic és confondre l'autenticació a nivell de servidor amb l'autenticació a nivell de programari que realitzen les seus webs. Quan un usuari es valida en un

entorn web com per exemple Yahoo o Google, no està usant l'autenticació amb el servidor web. Està usant un usuari i una contrasenya de l'empresa web a la qual es connecta i la gestió d'aquesta sessió d'usuari per consultar el seu correu es realitza mitjançant la programació en les mateixes pàgines web que visita. Això **no** té res a veure amb el control d'accés al servidor que es tracta en aquest apartat.

Autenticar els usuaris és determinar de forma veraç si un usuari és qui diu ser. **Autoritzar** és indicar quins usuaris tenen dret a accedir a quins recursos. Les seus web i els directoris que limiten l'accés a un conjunt restringit d'usuaris s'anomenen **recursos restringits**. Els recursos restringits que implementen la mateixa política de seguretat es poden agrupar en **reialmes**.

1.5.1 Els mòduls de control d'accés

Mòduls

Es pot obtenir la llista de mòduls relacionats amb l'autenticació i el control d'accés consultant la pàgina corresponent de la documentació d'Apache.

Apache gestiona l'autenticació i el control d'accés al servidor mitjançant mòduls propis (a part que es poden incorporar mòduls externs). Cada mòdul consta d'un conjunt de directives que permeten configurar el funcionament de l'autenticació i control d'accés implementats. Aquests mòduls es poden classificar en tres categories segons la seva funcionalitat:

- **Tipus d'autenticació**: l'autenticació pot ser de tipus *basic* o *digest*. En aquests exemples s'utilitzarà autenticació bàsica. L'autenticació *digest* implica comunicacions xifrades. Aquests mòduls s'implementen amb la directiva ***AuthType***.

```
1 AuthType Basic
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **auth** d'*authentication*.

```
1 mod_auth_basic
2 mod_auth_digest
```

- **Proveïdor d'autenticació**: indica quin és el mecanisme usat per realitzar l'autenticació. Són els mòduls que permeten autenticar usant fitxers de contrasenyes o el mòdul PAM o el de LDAP... Es poden identificar els mòduls d'aquesta família perquè inclouen en el seu nom la cadena **authn** d'*authentication*.

```
1 mod_authn_file
2 mod_authn_alias
3 mod_authnz_ldap
4 ...
```

- **Autorització**: els mòduls d'aquesta família proporcionen autorització a nivell d'usuaris, de grups, del LDAP o del que convingui. Aquests mòduls es determinen segons el valor que prengui la directiva ***Require***.

```
Require user valid-user
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **authz** d'*authorization*.

```
mod_authz_user
mod_authz_group
mod_authz_owner
mod_authz_ldap
...
```

1.5.2 Autenticació bàsica amb fitxers

El mecanisme més senzill per implementar l'autenticació en el servidor és l'autenticació bàsica amb fitxers d'usuaris i grups específics per al servidor web. Això es pot interpretar com un desavantatge perquè obliga a portar una gestió d'usuaris a més de la gestió d'usuaris del sistema. Però al mateix temps és un avantatge si el que volem és segregar aquets dos conjunts d'usuaris i administrar-los per separat.

Amb l'autenticació bàsica utilitzant fitxers es poden validar els usuaris utilitzant un **fitxer d'usuaris**, que conté els comptes d'usuaris i les seves contrasenyes.
També es poden validar grups d'usuaris amb un **fitxer de grups**, que indica quins usuaris formen part de cada grup.

El procés més simplificat per implementar la verificació d'usuaris i grups mitjançant fitxers de text pla amb contrasenyes requereix els passos següents:

1. Crear el fitxer d'usuaris en què s'indica la contrasenya corresponent a cada usuari.
2. Crear el fitxer de grups assignant a cada grup els usuaris que en formen part.
3. Identificar (o crear) el recurs que ha de tenir l'accés restringit.
4. Definir les directives apropiades per restringir l'accés al recurs als usuaris autoritzats.

L'exemple següent crearà un directori anomenat *privat* en la nostra web, al qual només hi podran accedir els usuaris autoritzats.

Primer cal crear el fitxer d'usuaris. Es tracta d'un fitxer de text pla en el qual s'emmagatzemen l'identificador i la contrasenya, que pot ser en text pla o xifrada,

de cada usuari. Per crear el fitxer i cada nou usuari s'utilitza l'ordre ***htpasswd***, proporcionada pel paquet del servidor. En el primer exemple s'utilitza l'opció *-c*, que crea el fitxer de nou.

```
root@server:/# htpasswd -c /var/www/passwd usuari
New password: usuari
Re-type new password: usuari
Adding password for user usuari
root@server:/# htpasswd /var/www/passwd usuari2
root@server:/# htpasswd /var/www/passwd usuari3
root@server:/# htpasswd /var/www/passwd usuari4
usuari:DeSaz54k9YRJU
usuari2:N0OF27Bcygc..
usuari3:okdHbmg.0G.Xo
...
```

A continuació cal posar en cada grup (de moment no n'hi ha cap) els usuaris que n'han de formar part. De fet, és tan senzill com crear un fitxer de text pla cada línia del qual consta del nom del grup, el delimitador dos punts (:) i la llista d'usuaris separats per espais.

```
root@server:/# vim /var/www/group
usuaris: usuari usuari2 usuari3 usuari4
```

Ara cal generar el **recurs restringit**, l'accés al qual només es permetrà als usuaris autoritzats. En aquest cas serà un directori anomenat *privat* a la nostra web.

```
root@server:/# mkdir /var/www/html/privat
root@server:/# vim /var/www/html/privat/index.html
... creació de la pàgina ....
```

Finalment s'assignen al directori local les directives apropiades per convertir-lo en un recurs d'accés restringit. Cal modificar el fitxer de configuració global apache2.conf (o el fitxer corresponent si es fa com a seu virtual) i definir un bloc de configuració usant la directiva ***Directory***. En aquesta directiva cal indicar la ruta absoluta corresponent al sistema de fitxers real del servidor (no es possible usar rutes relatives al servei web).

```
<Directori path-absolut-filesystem>
... opcions de configuració ...
</Directory>
```

Un exemple complet de configuració és el que es mostra a continuació, en el qual únicament es permet accedir al recurs a usuaris del grup anomenar *usuaris*:

```
<Directory /var/www/html/privat>
    AuthType Basic
    AuthName "Fitxers restringits"
    AuthBasicProvider file
    AuthUserFile /var/www/passwd
    AuthGroupFile /var/www/group
    Require group usuaris
</Directory>
```

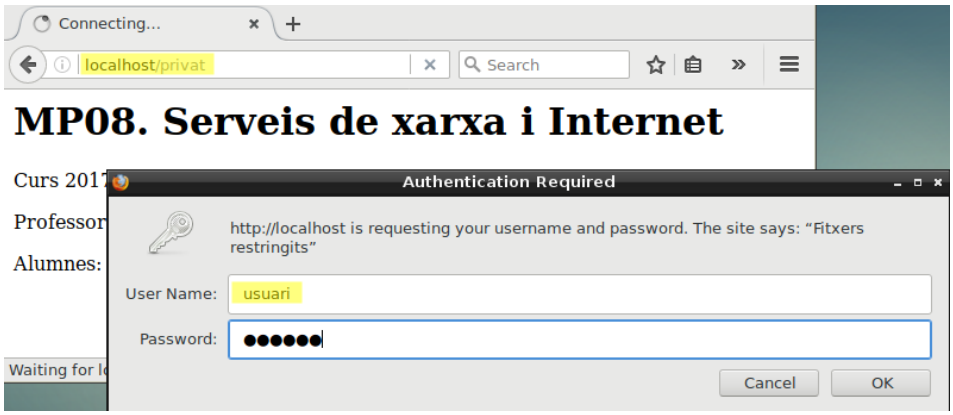
Vegeu les directives que s'utilitzen:

- ***AuthType***: indica que el tipus d'autenticació és bàsica (en lloc de *digest*).

- ***AythName***: declara el reialme al qual pertany el recurs restringit. Això permet que si hi ha altres recursos restringits associats a aquest reialme l'usuari que ja s'ha autenticat en un d'ells no ho hagi de fer en els altres. El nom del reialme el posa l'administrador web.
- ***AuthBasicProvider***: indica el mètode d'autenticació a usar. Pot prendre valors tipus *ldap*, *pam*, *dbm*, *bdb*, *file* i d'altres. El valor *file* significa que s'utilitzarà un fitxer d'usuaris i opcionalment un de grups.
- ***AuthUserFile***: indica quin és el fitxer que conté els comptes dels usuaris locals del servidor Apache. És el fitxer que s'ha creat en l'exemple anterior.
- ***AuthGroupFile***: indica quin és el fitxer de grups en el qual consta quins grups d'usuaris hi ha i quins usuaris pertanyen a cada grup.
- ***Require***: aquesta directiva és la que determina quina és la autorització a realitzar. En l'exemple es permet que qualsevol usuari del grup *usuaris* tingui accés al recurs.

Finalment, cal verificar que l'accés al directori local és concedit únicament als membres del grup profes. Evidentment el mecanisme més senzill és verificar des d'un navegador l'accés al recurs privat i observar que es demana l'autenticació. En la figura 1.3 es pot observar una petició d'autenticació d'usuari realitzada des d'un navegador Firefox.

Figura 1.3. Petició d'autenticació d'usuari



Exemples de mecanismes d'autorització

La directiva *Require* és la que defineix l'autorització d'accés al recurs, és a dir, qui pot accedir-hi. Aquests en són alguns exemples d'ús:

- *Require user valid-user*: permet l'accés a qualsevol usuari autenticat.
- *Require user usuari usuari2*: permet l'accés als usuaris indicats (usuari i usuari2).
- *Require group usuaris*: permet l'accés als usuaris que són membres d'algun dels grups indicats.

1.6 Comunicacions segures

El protocol HTTP pateix els mateixos problemes de seguretat que els seus companys dels inicis d'Internet (FTP, TFTP, SMTP...). Tota la informació viatja en text net i és fàcilment monitorable per altres. Quan un usuari es connecta a un web i indica l'usuari i la contrasenya, aquestes dades viatgen sense cap mena de protecció i qualsevol les pot capturar. Si el que es transmet són dades bancàries, llistes d'amistats íntimes o qualsevol tipus de dada privada, és desaconsellable fer-ho per HTTP.

El primer mecanisme de seguretat que es va implementar per a HTTP va ser el protocol SSL (Secure Socket Layer o capa de sòcol segur), desenvolupat per Netscape. L'SSL proporciona una capa entre la capa de transport TCP i la capa d'aplicació HTTP en què les dades viatgen xifrades. L'HTTPS solament és un esquema URI que indica la utilització d'HTML més algun mecanisme de transport xifrat, com SSL o TLS.

Quan s'utilitza HTTP amb un protocol xifrat com SSL o TCL s'anomena HTTPS (secure HTTP). Utilitza el port 443.

El protocol SSL es va enviar a l'IETF (Internet Engineering Task Force o Equip d'Enginyeria d'Internet, l'òrgan rector d'Internet) per a l'estandardització i després de diversos canvis va sorgir el protocol TLS (Transport Layer Security, Seguretat de capa de transport). El TLS proporciona les mateixes condicions de confidencialitat i autenticació en les transmissions HTTP que SSL.

L'HTTPS garanteix el trànsit de dades xifrat i el certificat del servidor. "En principi", autentica el web, però veurem que això depèn de si es confia o no amb el certificat usat.

Un dels avantatges de l'HTTPS és que permet la confidencialitat entre tots dos extrems de la comunicació encara que només sigui un dels extrems el que s'ha autenticat. Aquest model és molt pràctic quan, per exemple, un client anònim compra en un web autenticat. Quan es volen pagar els bitllets d'avió, interessa que les dades de la targeta de crèdit viatgin xifrades i que el receptor sigui la companyia aèria i no un web fals.

L'ús dels certificats no és exclusiu per autenticar el servidor. Si cal, els clients poden ser autenticats. Per exemple, un web pot requerir que els clients disposin del certificat que els atorga dret a accedir-hi (expedit, per exemple, per la mateixa entitat).

Els passos necessaris per implementar comunicacions segures que permeten a un navegador client (o a un client, sigui qui sigui) connectar-se via HTTPS a una seu web són:

- **Certificats digitals**: el servidor web ha de disposar d'una clau privada i d'un certificat digital.
- **Mòdul *mod_ssl***: cal tenir instal·lat el paquet de programari que proporciona les prestacions SSL al servidor i que la configuració activa en carregui els mòduls pertinents.

- **Configurar la seu web segura**: finalment cal establir les directives SSL apropiades a la seu web que es vol configurar per fer-la accessible via SSL.

L'objectiu de les explicacions següents és implementar connexions segures HTTPS a la seu web www.ioc.cat utilitzant SSL com a mecanisme de transport xifrat.

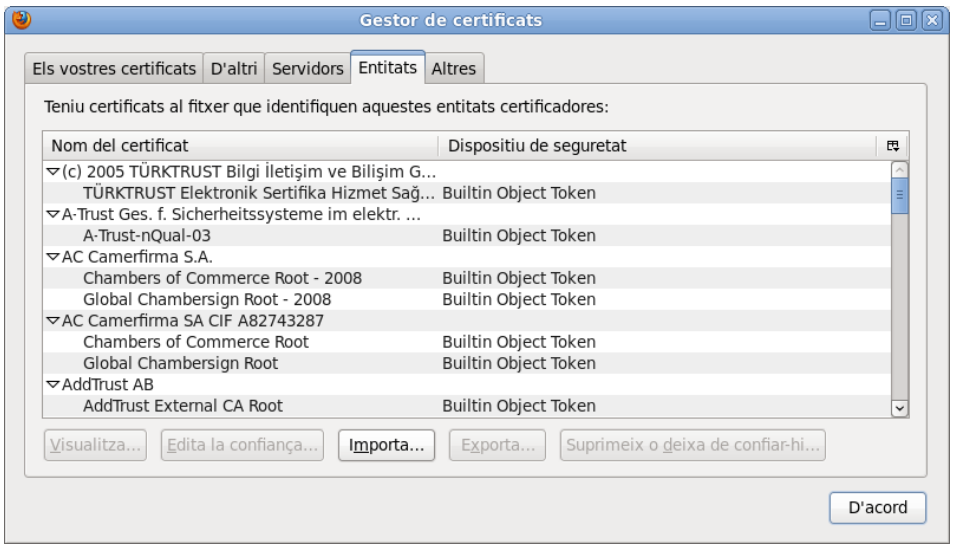
1.6.1 Els certificats del servidor

Suposarem que el servidor disposa ja d'una clau privada i d'un certificat, amb independència de com s'hagin obtingut. En concret, en el subdirectori certs del directori base del servei web hi ha:

- **server.crt**: el fitxer corresponent al certificat o clau pública del servidor. Aquest fitxer assegura als clients que es connecten a la seu web que el servidor és qui realment diu ser.
- **server.key**: és el fitxer amb la clau privada del servidor. Aquest fitxer s'ha codificat amb una *passphrase* o frase de pas de manera que cada vegada que s'inicialitzi el servidor web caldrà entrar aquesta frase.

La figura 1.4 mostra la pantalla típica de gestió de certificats de Firefox. En aquesta pantalla es poden llistar els certificats preinstal·lats en el navegador, observar-ne les seves propietats i també es poden afegir i eliminar certificats.

FIGURA 1.4. Pantalla de gestió de certificats de Firefox



Cal recordar que els navegadors clients validaran la confiança que els mereix el certificat contrastant el seu emissor amb la llista d'entitats certificadores que tenen carregada. Si l'emissor del certificat no hi és, caldrà fer passos per incorporar el certificat al navegador. Aquests passos poden ser:

- Admetre el certificat com a vàlid quan el navegador presenta "l'excepció de seguretat".
- Obtenir el certificat de l'entitat CA (Certification Authority o autoritat de certificació) que l'ha generat i incorporar l'entitat al llistat d'entitats en què el navegador confia.

Generar un certificat autosignat

A mode de recordatori ràpid, es pot generar una clau privada i un certificat autosignat fent:

```
# openssl req -new -x509 -nodes -out server.crt -keyout server.
    key
```

1.6.2 Configuració d'Apache per usar SSL

El servidor web podrà usar SSL si disposa dels mòduls que en proporcionen la capacitat. En cas de no tenir-los, cal buscar en els repositoris de programari habitual un paquet que proporcioni el mòdul apropiat, instal·lar-lo i examinar-ne el contingut. Usualment, tant el paquet com el mòdul que proporcionen les prestacions de trànsit segur SSL s'anomenen **mod_ssl**.

Per habilitar-lo a Apache:

```
root@server:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create
    self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:~#
```

A la carpeta `/etc/apache2/mods-available` hi ha els diferents mòduls que es poden activar, entre els quals es troba el mòdul SSL. Es poden distingir dos fitxers, un que s'encarrega de la configuració i l'altre de la càrrega del mòdul:

```
root@server:/# ls /etc/apache2/mods-available/ssl.*
/etc/apache2/mods-available/ssl.conf  /etc/apache2/mods-available/ssl.load
root@server:/# cat /etc/apache2/mods-available/ssl.load
# Depends: setenvif mime socache_shmcb
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
root@server:/# cat /etc/apache2/mods-available/ssl.conf
<IfModule mod_ssl.c>

  # Pseudo Random Number Generator (PRNG):
  # Configure one or more sources to seed the PRNG of the SSL library.
  # The seed data should be of good random quality.
  # WARNING! On some platforms /dev/random blocks if not enough entropy
  # is available. This means you then cannot use the /dev/random device
  # because it would lead to very long connection times (as long as
  # it requires to make more entropy available). But usually those
```

```
  # platforms additionally provide a /dev/urandom device which doesn't
  # block. So, if available, use this one instead. Read the mod_ssl User
  # Manual for more details.
  #
  SSLRandomSeed startup builtin
...
```

En executar la comanda *a2enmod*, hem fet que es crei un enllaç simbòlic des de la carpeta `/etc/apache2/mods-enabled` a la carpeta `/etc/apache2/mods-available` per a aquest mòdul. Aquest és el mecanisme que fa servir Apache per activar i desactivar els mòduls. En cas que es vulgui desactivar, cal fer servir la comanda *a2dismod*. També es poden fer servir sense paràmetres, i indicaran quins mòduls volem activar o desactivar dels que estan desactivats o activats respectivament:

```
root@server:/# a2enmod
Your choices are: access_compat actions alias allowmethods asis auth_basic
     auth_digest auth_form authn_anon authn_core authn_dbd authn_dbm authn_file
      authn_socache authnz_fcgi authnz_ldap authz_core authz_dbd authz_dbm
     authz_groupfile authz_host authz_owner authz_user autoindex brotli buffer
     cache cache_disk cache_socache cern_meta cgi cgid charset_lite data dav
     dav_fs dav_lock dbd deflate dialup dir dump_io echo env expires ext_filter
      file_cache filter headers heartbeat heartmonitor http2 ident imagemap
     include info lbmethod_bybusyness lbmethod_byrequests lbmethod_bytraffic
     lbmethod_heartbeat ldap log_debug log_forensic lua macro md mime
     mime_magic mpm_event mpm_prefork mpm_worker negotiation proxy proxy_ajp
     proxy_balancer proxy_connect proxy_express proxy_fcgi proxy_fdpass
     proxy_ftp proxy_hcheck proxy_html proxy_http proxy_http2 proxy_scgi
     proxy_uwsgi proxy_wstunnel ratelimit reflector remoteip reqtimeout request
      rewrite sed session session_cookie session_crypto session_dbd setenvif
     slotmem_plain slotmem_shm socache_dbm socache_memcache socache_shmcb
     speling ssl status substitute suexec unique_id userdir usertrack
     vhost_alias xml2enc
Which module(s) do you want to enable (wildcards ok)?
^C
root@server:/# a2dismod
Your choices are: access_compat alias auth_basic authn_core authn_file
     authz_core authz_host authz_user autoindex deflate dir env filter mime
     mpm_event negotiation reqtimeout setenvif status
Which module(s) do you want to disable (wildcards ok)?
^C
root@server:/#
```

1.6.3 Configuració de la seu web amb SSL

Cal aplicar a la nostra seu web les directives SSL apropiades per fer possible l'accés a aquesta seu web per HTTPS. El llistat de la directiva *VirtualHost* és:

```
<VirtualHost www.ioc.cat:443>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    SSLEngine On
    SSLProtocol all -SSLv3
    SSLCertificateKeyFile /var/www/certs/server.key
    SSLCertificateFile /var/www/certs/server.crt
</VirtualHost>
```

Vegeu les directives usades:

- **Port 443**: aquest és el port usual per a les connexions segures HTTP. Si la seu web només escolta per aquest port, només es podrà accedir al seu contingut per HTTPS. Si es volen seus diferents per al trànsit xifrat i per al no xifrat, n'hi ha prou de crear una altra seu virtual amb un altre port.
- ***SSLEngine On***: aquesta directiva indica que cal activar el trànsit SSL per a aquesta seu web.
- ***SSLProtocol all -SSLv3***: en aquesta directiva s'indiquen quins protocols es poden usar per generar el trànsit xifrat. Les opcions *all* i *-SSLv3* indiquen que s'accepten tots els protocols vàlids excepte el protocol SSLv3.
- ***SSLCertificateKeyFile* <clau privada del servidor>**: aquesta directiva indica el fitxer amb la clau privada del servidor.
- ***SSLCertificateFile* <certificat>**: indica el fitxer que conté el certificat del servidor. Aquest és el certificat que els navegadors clients veuran i del qual hauran de decidir si hi confien o no.
- ***SSLCACertificateFile* <certificat-CA>**: aquesta directiva és opcional i permet indicar quin és el fitxer que conté el certificat públic que ha emès l'entitat de certificació CA.

Un cop configurada apropiadament la seu virtual, cal reiniciar el servei. Des de qualsevol navegador s'ha de poder accedir a la seu usant HTTPS. Ara bé, es generarà una excepció de seguretat perquè el navegador desconeix la procedència del certificat. Si l'usuari accepta confiar en la seu web, el certificat s'incorporarà al navegador i accedirà de forma xifrada a la seu web.

No obstant, Apache porta ja una seu web predeterminada (en forma de plantilla) per activar la seu web amb SSL. Aquest fitxer és el `/etc/apache2/sites-available/default-ssl.conf`.

1.6.4 Verificació de les connexions SSL

Els problemes principals que es poden trobar els navegadors en connectar amb seus web amb certificats són els següents:

- Amb un certificat autosignat no cal definir cap CA. El navegador client mostrarà la típica pantalla d'excepció de seguretat i caldrà indicar que s'accepta el certificat de servidor per a la nostra entitat. És un certificat emès per la mateixa entitat.
- Amb un certificat generat per una CA local cal incorporar manualment el certificat al navegador. Un cop fet això el navegador serà capaç de validar el certificat del servidor amb la CA que l'ha expedit (*issuer*).

A més dels navegadors, existeixen eines d'entorn de text per verificar connexions SSL, de la mateixa manera que s'utilitza *telnet host 80* per verificar connexions

HTTP. D'una banda, es pot usar el mateix **OpenSSL** i, de l'altra, es pot instal·lar la utilitat **Curl**, que permet fer un ampli seguiment del diàleg SSL.

```
root@server:/# openssl s_client -connect www.ioc.org:cat
root@server:/# curl https://www.ioc.cat -kv
```

1.7 Monitoratge del servei

El servei web incorpora diverses eines per monitorar el funcionament del servei, algunes de configurades per defecte i d'altres que s'hi poden afegir. Les dues utilitats tractades en aquest apartat són *server-status* i *server-info*. Cal afegir el codi corresponent per tal d'indicar a quines seus web es vol fer el monitoratge per tal d'activar-los. A més a més, cal assegurar que aquests mòduls estan habilitats, ja que depenent de la versió d'Apache i distribució de Linux no sempre és així.

```
root@server:/etc/apache2# a2enmod info
Enabling module info.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:/etc/apache2# a2enmod status
Module status already enabled
root@server:/etc/apache2#
```

El fragment de codi següent mostra la configuració que permet monitorar les seus corresponents al nom de host *server* (la seu principal o per defecte) i la seu www.ioc.cat.

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat server
</Location>
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat server
</Location>
```

Per accedir als continguts del monitoratge simplement cal indicar des d'un navegador client els URL apropiats:

```
www.ioc.cat/server-info
www.ioc.cat/server-status
```

1.7.1 Utilitat de server-status

La informació bàsica que es mostra en consultar el *server-status* de la seu web www.ioc.cat és la següent:

```
Apache Server Status for www.ioc.cat (via 10.0.2.15)

Server Version: Apache/2.4.38 (Debian)
Server MPM: event
Server Built: 2019-10-15T19:53:42

Current Time: Monday, 21-Sep-2020 19:50:49 CEST
Restart Time: Monday, 21-Sep-2020 19:49:40 CEST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 1 minute 9 seconds
Server load: 0.00 0.03 0.06
Total accesses: 2 - Total Traffic: 14 kB - Total Duration: 15
CPU Usage: u0 s0 cu0 cs0
.029 requests/sec - 207 B/second - 7.0 kB/request - 7.5 ms/request
1 requests currently being processed, 49 idle workers

Slot  PID Stopping  Connections   Threads Async connections
total accepting busy  idle  writing keep-alive  closing
0 7866  no  1 yes 1 24  0 0 0
1 7867  no  0 yes 0 25  0 0 0
Sum 2 0 1   1 49  0 0 0

_____________W_____________________________.............
..........................................................
......................

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process
Srv PID Acc M CPU   SS  Req Dur Conn  Child Slot  Client  Protocol  VHost
    Request
0-0 7866  0/1/1 _   0.00  12  15  15  0.0 0.01  0.01  server  http/1.1  server.
    ioc.cat:80 GET /server-info HTTP/1.1
0-0 7866  0/1/1 _   0.00  12  0 0 0.0 0.00  0.00  server  http/1.1  server.ioc.
    cat:80 GET /favicon.ico HTTP/1.1
0-0 7866  0/0/0 W   0.00  0 0 0 0.0 0.00  0.00  10.0.2.15 http/1.1  server.ioc.
    cat:80 GET /server-status HTTP/1.1
Srv Child Server number - generation
PID OS process ID
Acc Number of accesses this connection / this child / this slot
M Mode of operation
CPU CPU usage, number of seconds
SS  Seconds since beginning of most recent request
Req Milliseconds required to process most recent request
Dur Sum of milliseconds required to process all requests
Conn  Kilobytes transferred this connection
Child Megabytes transferred this child
Slot  Total megabytes transferred this slot
Apache/2.4.38 (Debian) Server at www.ioc.cat Port 80
```

Els elements més destacats de la informació d'estatus són:

- Versió del servidor i data de compilació.
- Últims cops que s'ha engegat el servei i temps que fa que està actiu.
- Ús de CPU, detalls del trànsit i estadístiques sobre les peticions realitzades.

1.7.2 Utilitat de server-info

El servei de monitoratge *server-info* proporciona informació detallada de la configuració del dimoni del servidor, els valors obtinguts de processar cada un dels fitxers de configuració, els mòduls carregats i les configuracions de cada mòdul.

- **server settings**: descriu la versió del servidor i les opcions amb què s'ha compilat. En particular, podeu observar els valors que descriuen l'arrel de l'estructura de fitxers del servidor i el fitxer de configuració a usar.

```
Server Root: /etc/apache2
Config File: /etc/apache2/apache2.conf
```

- **configuration files**: descriu detalladament tots els valors de configuració, obtinguts dels diferents fitxers de configuració. Mostra el número de línia, la directiva i el valor que pren. Es pot observar que s'analitzen tots els fitxers actualment carregats en la configuració, el fitxer global apache2.conf i els inclosos en el directori `/etc/apache2`.

```
Configuration:
    In file: /etc/apache2/apache2.conf
      87: PidFile /var/run/apache2/apache2.pid
      92: Timeout 300
      98: KeepAlive On
     105: MaxKeepAliveRequests 100
     111: KeepAliveTimeout 5
     115: User www-data
     116: Group www-data
     126: HostnameLookups Off
     134: ErrorLog /var/log/apache2/error.log
     143: LogLevel warn
    In file: /etc/apache2/mods-enabled/alias.conf
      14: Alias /icons/ "/usr/share/apache2/icons/"
      16: <Directory "/usr/share/apache2/icons">
      17:   Options FollowSymlinks
      18:   AllowOverride None
      19:   Require all granted
        : </Directory>
    In file: /etc/apache2/mods-enabled/autoindex.conf
      ...
```

- **Llistat dels mòduls**: fa un llistat de tots els mòduls carregats actualment en la memòria. Es poden observar quins mòduls estan carregats estàticament i quins dinàmicament.

```
Server Module List
    core.c
    event.c
    http_core.c
...
```

- **Informació detallada de cada mòdul**: segurament aquesta és una de les opcions més interessants, perquè permet observar les directives proporcionades per un mòdul i els valors que té assignats. D'aquesta manera, es pot validar fàcilment si el mòdul adopta els valors apropiats o si s'ha comès alguna errada en la configuració.

1.8 Registres del servei

Els fitxers de *logs* enregistren tots els successos que es produeixen en el servei web i que s'ha declarat que s'han d'enregistrar. El seu funcionament és idèntic al dels *logs* del sistema. El servidor web Apache utilitza un fitxer que enregistra els errors que es produeixen i un altre que enregistra els accessos al servidor web. Aquests fitxers enregistren el servei web global o per defecte. Per a cada seu web virtual o *virtual host* es poden definir els fitxers de *log* que es considerin oportuns. El contingut que s'enregistra per a cada succés és configurable, de manera que l'administrador pot personalitzar al seu gust quina informació s'emmagatzema i en quin format.

> Els fitxers de ***log*** enregistren els successos del servidor web. S'anoten aquells successos que s'han **declarat** per ser enregistrats i en un **format** de missatge configurable.
> El servei web principal utilitza un fitxer d'**errors** i un de **accés** en la configuració predeterminada. Cada una de les diferents seus virtuals defineix els seus propis fitxers de *log*.

El següent és un recull de les directives relacionades amb la gestió de *logs* que hi ha al fitxer de configuració global apache2.conf:

```
1 ErrorLog ${APACHE_LOG_DIR}/error.log
2 LogLevel warn
3 LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
      vhost_combined
4 LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
      combined
5 LogFormat "%h %l %u %t \"%r\" %>s %O" common
6 LogFormat "%{Referer}i -> %U" referer
7 LogFormat "%{User-agent}i" agent
```

Els conceptes principals que descriuen aquestes directives són:

- ***ErrorLog***: defineix quin és el fitxer en què s'han d'enregistrar els successos d'error.
- ***LogLevel***: defineix quin és el nivell (*verbosity*) dels successos que s'han d'enregistrar. Aquest poden ser: emerg, alert, crit, error, warn, notice, info, debug, trace1... trace8. Estan ordenats de menys informació a més, doncs cal tenir en compte d'ajustar bé aquest paràmetre si no volem tenir fitxers

de *log* molt grans i que alenteixin el servei web, sobretot en entorns de producció (un cop ja testejats).

- ***CustomLog***: defineix el fitxer en què s'han de desar els accessos al servei web. Aquest fitxer contindrà el registre de totes les sol·licituds i respostes gestionades pel servidor. El format dels missatges de *log* que s'enregistraran per defecte és *combined*. Aquesta directiva es pot usar per definir tants fitxers de *log* com es cregui pertinent, cal una directiva per a cada fitxer a generar.
- **LogFormat**: cada una d'aquestes directives defineix un format de missatge de *log*. En l'exemple es defineixen cinc formats: *vhost_combined*, *combined*, *common*, *referer* i *agent*. Amb aquesta directiva l'administrador pot personalitzar el format que han de tenir els missatges de *log* al seu gust.
- **Directori de logs**: no es mostra explícitament en cap de les directives, però es pot veure que està definit en la variable d'entorn APACHE_LOG_DIR definida al fitxer `/etc/apache2/envvars`.

```
root@server:/# cat /etc/apache2/envvars | grep APACHE_LOG_DIR
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX
root@server:/#
```

Aquesta, al seu torn, també està formada per una altra variable d'entorn (SUFFIX), que s'afegeix en el cas que s'hagi de tenir múltiples instàncies del servidor Apache:

```
# for supporting multiple apache2 instances
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
        SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
        SUFFIX=
fi
```

- **Seus virtuals**: cada una de les seus virtuals defineix els seus propis fitxers de *log*. És usual assignar a aquests fitxers el nom de la seu virtual, tal com es pot veure en l'exemple següent. Si una seu virtual no declara fitxers de *log* propis amb les directives *ErrorLog* i *CustomLog*, s'utilitzen els fitxers globals.

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    ServerName www.ioc.cat
    ErrorLog logs/www.ioc-error_log
    CustomLog logs/www.ioc-access_log common
</VirtualHost>
```

Així, seguint les indicacions estàndard, el directori de *logs* contindrà una parella de fitxers anomenats **error_log** i **access_log** per a cada seu virtual i per a la seu global. A més, altres mòduls, com per exemple SSL, poden generar els seus propis fitxers de *log*. També es mantenen històrics dels registres, de manera que es poden trobar múltiples versions del mateix fitxer corresponents a períodes de temps diferents (tasca duta a terme per *log-rotate*).

2. Instal·lació i administració de serveis de transferència de fitxers

Seguretat de la xarxa

En els orígens d'Internet, els usuaris eren part implicada i no els va passar pel cap que hi poguessin haver usuaris amb ànim d'"atacar" altres sistemes.

L'**FTP** (File Transfer Protocol o **protocol de transferència de fitxers**) és el protocol que proporciona el servei de transferència de fitxers més usat. Es basa en una arquitectura client/servidor i utilitza el protocol de transport TCP. Permet la transferència de fitxers de qualsevol tipus entre dos equips. El servidor actua a mode de repositori de fitxers i el client s'hi connecta per baixar (*download*) o pujar (*upload*) fitxers.

El **protocol FTP** (File Transfer Protocol o protocol de transferència de fitxers) és un protocol TCP que permet la transferència de fitxers en ambdós sentits entre un client i un servidor. Aquesta transferència és independent de la plataforma usada i del tipus de fitxers manejats.

La necessitat d'un mecanisme de transferència de fitxers sorgeix des de bon principi a Internet i el 1980 ja apareix la primera especificació de l'FTP. L'octubre de 1985 es publica l'RFC 959, que és la base del protocol actual. A aquest RFC se n'hi han afegit posteriorment d'altres per incorporar seguretat, internacionalització... Tractant-se d'un protocol tan "vell", no és estrany que sigui insegur. De fet, la majoria de protocols originaris d'Internet ho són (HTTP, SMTP...).

Una de les virtuts del model FTP és que permet transferir fitxers (també modificar-los, esborrar-los, afegir-los...) independentment de la plataforma i del sistema on resideixen. És a dir, l'FTP amaga els detalls d'implementació. Un client (que funcioni amb sistema operatiu Unix, per exemple) baixa un fitxer de text d'un servidor FTP sense saber el tipus de màquina ni el tipus de sistema de fitxers del servidor (Windows, per exemple).

Els objectius del servei FTP són els següents:

- Compartir fitxers tant de dades com de programes.
- Permetre prosseguir les transferències de fitxers un cop interrompudes (reprendre les baixades).
- Transferir dades de manera eficient i fiable.

La debilitat principal del protocol FTP és la falta de seguretat. Tot el flux de dades viatja en text pla, sense encriptar, fins i tot els noms d'usuari i les contrasenyes. Això ha obligat a adoptar noves estratègies per proporcionar confidencialitat a les transmissions.

2.1 Servei de transferència de fitxers

El servei FTP es basa en l'arquitectura client/servidor, de manera que caldrà descriure'l en cada un d'ells. En aquest apartat es descriuen els diferents modes d'operació d'un servidor FTP, les seves funcionalitats i els tipus d'accés permesos. Els servidors poden oferir els seus serveis a tota la comunitat d'Internet o a un entorn restringit, com per exemple una xarxa local. En el primer cas es parla d'un servidor d'accés públic global, en el segon cas, d'un servidor local o corporatiu. El servei pot ser ofert a qualsevol usuari, incloent usuaris anònims, o es pot restringir a usuaris i grups determinats.

El servidor FTP es pot classificar segons sigui:

- D'accés **públic /** d'accés **corporatiu**
- D'accés amb usuari **identificat /** d'accés amb usuari **anònim**
- De mode de transferència **ASCII /** de mode de transferència **binari**

De forma inusual, el servei FTP utilitza dos ports del sistema. A través del port 21 es realitza la interpretació de les instruccions. El port 20 és destinat a la transferència de dades, tot i que es pot utilitzar un altre port dinàmic en el seu lloc. El mode real de funcionament de la transmissió de dades i els ports implicats depenen del mode de funcionament, que pot ser actiu o passiu.

El mode de funcionament de la transferència FTP pot ser actiu o passiu. La funcionalitat del servei es classifica en:

- Mode intèrpret del protocol.
- Mode de transferència de dades.

2.1.1 Tipus de clients i servidors

El **servidor FTP** és una màquina que executa un programari determinat que proporciona el servei FTP a clients FTP. Usualment, en entorns GNU/Linux aquest programa és un dimoni, anomenat usualment *ftpd* o similar. En funció del tipus d'usuaris que permet connectar i de l'àmbit d'accés que permet, el podem classificar de maneres diferents.

Segons el tipus de clients que accepta, podem classificar els servidors FTP de la manera següent:

- **Usuari identificat**. El servidor requereix un nom d'usuari i una contrasenya vàlids per accedir al servei. Els comptes d'usuari poden ser gestionats directament per l'aplicació del servidor o se'n pot delegar l'autenticació al sistema operatiu.
- **Accés anònim**. Un servidor que permet accessos anònims permet que qualsevol usuari pugui accedir al repositori de fitxers. Usualment cal indicar com a nom d'usuari "anonymous" i com a contrasenya s'accepta qualsevol text, però per convenció s'escriu el correu electrònic de l'usuari.

Alguns servidors FTP permeten l'accés anònim acceptant qualsevol nom d'usuari i qualsevol contrasenya o fins i tot sense contrasenya.

Segons l'àmbit del servei que proporciona, podem classificar els servidors FTP de la manera següent:

- **Servidor públic**. Molts servidors FTP a Internet ofereixen servei d'accés anònim a mode de repositoris de programari perquè els usuaris el puguin utilitzar. N'hi ha que actuen com a rèpliques (miralls, *mirrors*) d'altres repositoris per tal d'apropar les baixades a l'usuari. Aquest servei és usualment només de lectura (pel client) i en sistemes GNU/Linux s'ubica sovint en els directoris /var/ftp o /var/ftp/pub.
- **Servidor corporatiu**. No cal oferir per força els serveis FTP a Internet; l'administrador de xarxa pot configurar el servidor FTP per oferir els serveis als equips que cregui oportuns. Dins d'una xarxa corporativa es pot disposar d'un o més servidors FTP que permeten l'accés als usuaris de la xarxa (tant a usuaris identificats com a usuaris anònims de la xarxa corporativa).

Exemple de mirall

Si us voleu baixar un GNU/Linux Live, en lloc de contactar el servidor de Fedora o Ubuntu, ho podeu fer en un mirall de RedIRIS, que és més proper.

Evidentment, el servidor FTP pot combinar els models anteriors i proporcionar accés tant a usuaris identificats com a usuaris anònims, i pot diferenciar els recursos que ofereix en funció de si són usuaris interns de la xarxa corporativa o d'Internet.

El **client FTP** és el programari que s'utilitza per establir una connexió amb el servidor per tal de poder baixar o pujar fitxers al servidor. El client pot llistar, modificar i afegir fitxers al servidor, a més de realitzar altres accions, sempre que hi estigui autoritzat. L'aplicació client pot ser basada en text o d'entorn gràfic, però en qualsevol cas ha de poder establir connexió amb el servidor.

URL

Acrònim d'*Uniform Resource Locator* (en català, localitzador uniforme de recursos). Sovint és usat com a sinònim d'URI (*Uniform Resource Identifier*, identificador uniforme de recursos), tot i que no és el mateix.

La connexió FTP es pot indicar mitjançant un URL del tipus ftp://servidor/fitxer, on *fitxer* pot ser una trajectòria. De fet, l'URL pot ser més detallat:

```
1 ftp://usuari:contrasenya@servidor:port/fitxer
```

En aquest format s'indica l'usuari i la contrasenya, el servidor, el port d'accés i el fitxer al qual es vol accedir. En determinats sistemes operatius es pot usar aquesta sintaxi en la línia d'ordres per indicar un fitxer remot, igual que es faria per indicar un fitxer local.

El **servidor** proporciona un repositori de fitxers i una aplicació que permet que els **clients** s'hi connectin i facin ús dels fitxers (baixar-los o pujar-ne). L'URL per accedir a un fitxer per FTP es pot expressar així: **ftp://usuari:contrasenya@servidor:port/fitxer**.

2.1.2 Funcionament del servei FTP

El port 21 és el port del canal de control. El port 20 és el del canal de dades.

L'FTP és un protocol d'aplicació basat en TCP/IP que utilitza TCP com a capa de transport. El servidor escolta connexions entrants de clients pel port 21 i inicia una sessió si l'autenticació s'estableix correctament. El servidor pot funcionar com un servei per si mateix (*stand-alone*) o pot estar configurat per funcionar dins d'un superservei de xarxa com, per exemple, inetd o xinetd. Si funciona en mode de servei propi és el servidor qui escolta les connexions entrants i les atén. Si s'executa dins del superservei de xarxa, aquest és qui detecta les connexions entrants i activa el dimoni de l'FTP perquè les atengui. Un cop ateses, el dimoni de l'FTP acaba i torna a ser el superservidor de xarxa el que es queda escoltant.

Tipus de client

- Identificat: té accés al sistema de fitxers complet.
- Anònim: és engabiat en un punt de l'arbre de fitxers.

Serveis autònom i xinetd

En el servei autònom (*stand-alone*) el servidor escolta per si mateix les connexions entrants. En el servei xinetd o inetd, el servidor està dins d'un superservei de xarxa. inetd és un superdimoni de xarxa que escolta connexions entrants de diferents protocols i executa el dimoni del servei corresponent en rebre una connexió. xinetd n'és la versió millorada.

L'accés als recursos del servidor varia usualment en funció de si el client és anònim o està identificat. Els clients identificats poden navegar per l'estructura de fitxers segons els seus permisos. Els clients anònims usualment estan engabiats (*chroot*) en una part de l'arbre de fitxers i no en poden sortir. Usualment, el servidor fa correspondre el directori de publicació (/var/ftp o /srv/ftp en sistemes GNU/Linux) amb el directori arrel (/) d'accés del client. El client pot descendir a partir d'aquest punt, però no pot anar a directoris superiors.

Chroot

En sistemes GNU/Linux, *chroot* és una utilitat o una tècnica consistent a "engabiar" serveis en una part del disc com si fos el disc sencer, de manera que es fa correspondre un directori particular (on hi ha el servei) a una arrel de disc virtual. Els usuaris del servei creuen que naveguen per tot el disc, però en realitat estan "engabiats" en una estructura virtual.

ASCII

Acrònim d'American Standard Code for Information Interchange. En català, codi estàndard americà per a l'intercanvi d'informació.

Els modes en què es transfereixen els fitxers entre el client i el servidor poden ser múltiples. Els dos més importants són aquests:

- **ASCII**. El fitxer es transmet caràcter a caràcter. Els caràcters han de correspondre als caràcters del codi bàsic ASCII. Si el fitxer conté caràcters ASCII no vàlids, la transferència fallarà. Per tant, es tracta d'un mode vàlid únicament per transferir text net. El receptor farà les conversions de caràcter necessàries per desar les dades en el format que requereixi.

Format del salt de línia

Penseu en el típic problema del salt de línia que varia segons el sistema operatiu. El servidor, per exemple, envia text usant el caràcter LF com a salt de línia (usa Unix) i el receptor els desa usant la combinació de caràcters CR+LF (que és el format usat pel seu sistema operatiu diferent d'Unix).

- **Binari** (*binary*). Quan el mode de transferència és binari, el fitxer s'envia bit a bit sense interpretació de cap mena. És el mode que cal usar per transmetre programes, imatges, vídeo, so, dades binàries...

2.1.3 Especificació del protocol FTP

El protocol FTP és un protocol de capa d'aplicació basat en TCP com a capa de transport. Utilitza el port 21 per al canal de control i el port 20 per al canal de dades. És un dels pocs protocols actuals que encara utilitzen més d'un port per a la comunicació. El port 21 s'utilitza com a canal de comunicació entre client i servidor. És per on es transmeten les ordres, però no els fitxers. Aquests es transmeten per una connexió diferent, per un canal diferent, que en principi utilitza el port 20 del servidor.

Tant en el client com en el servidor hi ha dues entitats clarament diferenciades:

- **Intèrpret del protocol**: és l'encarregat de l'intercanvi d'ordres i respostes entre client i servidor. Utilitza el canal de control establert entre el port de sortida del client i el port 21, on escolta el servidor. És l'encarregat d'interpretar les ordres de l'aplicació client convertint-les en instruccions FTP, executar-les en el servidor i retornar les respostes al client. No s'encarrega de la transferència de fitxers.
- **Transferència de dades**: és la part encarregada d'intercanviar els fitxers i directoris entre client i servidor. En el funcionament bàsic, el canal de dades s'estableix entre un nou port del client (port dinàmic i específic per a la transmissió del fitxer) i el port 20 (*ftp-data*) del servidor.

Modes de connexió FTP

- Actiu: el servidor usa el port 20. Correspon a l'ordre PORT del protocol.
- Passiu: el servidor usa un port dinàmic (no el 20). Correspon a l'ordre PASV del protocol.

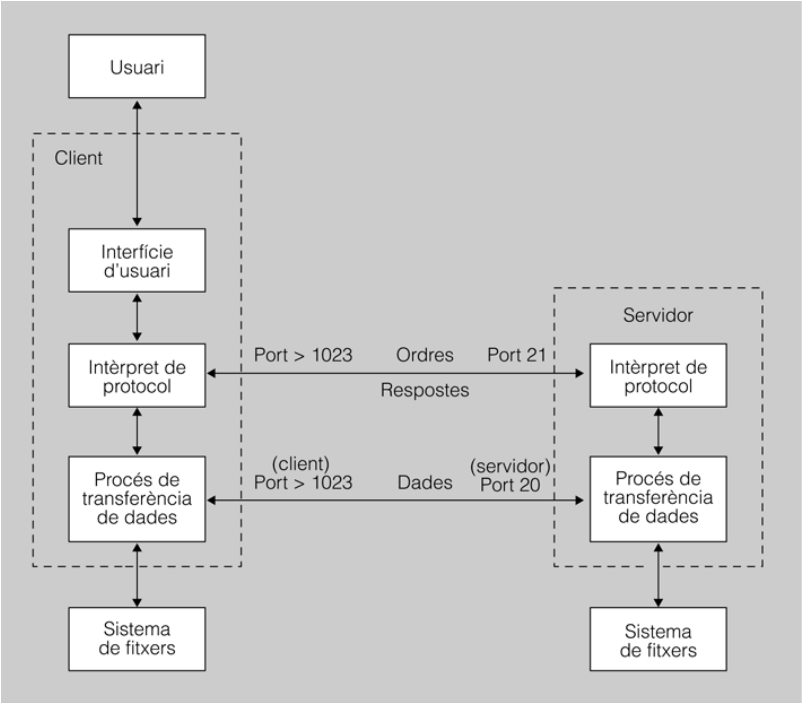
La connexió TCP del canal de dades entre client i servidor es pot establir de dues maneres diferents:

- **Mode actiu**: generalment és el mode per defecte. Abans de fer una sol·licitud al servidor que impliqui transferir dades pel canal de dades, el client indica al servidor el port dinàmic que utilitzarà. Per tant, el canal de dades s'estableix entre aquest port dinàmic del client i el port 20 del servidor. Servidor i client estableixen una nova connexió TCP per aquest canal.
- **Mode passiu**: el client fa una sol·licitud de mode passiu al servidor. Aquest respon enviant el seu port dinàmic, per on s'establirà el canal de dades (en lloc del port 20). Llavors el client inicia una nova connexió TCP entre un port dinàmic nou seu i el port dinàmic del servidor. Aquest és el canal de dades.

En resum, podríem dir que el client es posa en contacte amb el servidor connectant-se al port 21. Mitjançant aquest canal de comunicació, client i servidor governen tota la sessió FTP. Les transferències de fitxers es realitzen per una altra connexió

que es pot crear i destruir al llarg de la sessió (per exemple, per a cada baixada o segons els períodes d'inactivitat). En la figura 2.1 es pot veure l'esquema de connexió i de ports entre un client i un servidor FTP.

FIGURA 2.1. Model funcional del protocol FTP



Ordres i respostes FTP

En el document RFC 959 hi ha la llista d'ordres i la taula de codis/missatges de resposta del protocol FTP. Aquest document és l'especificació de l'estàndard FTP.

El protocol FTP descriu diferents categories d'ordres que el client pot realitzar (no les confongueu amb les ordres concretes de l'aplicació client). Aquestes s'agrupen en tres grups:

1. **Ordres de control d'accés**: les que gestionen l'accés al servei FTP. Per exemple, inici i finalització de la sessió, validació de l'usuari i la contrasenya, canvis de directori i de sistemes de fitxers...
2. **Ordres de paràmetres de transferència**: gestionen les opcions relacionades amb la transferència de fitxers com, per exemple, el mode de transferència binari o ASCII, els ports, el tipus de mode passiu o actiu...
3. **Ordres de servei FTP**: són les ordres d'allò que es vol fer en una sessió FTP com, per exemple, baixar un fitxer, pujar-lo, modificar-ne el nom...

Per a cada ordre del client, el servidor emet una resposta pel canal de control en què indica l'estatus de l'execució de l'ordre rebuda. Per exemple, el client envia el seu nom d'usuari i contrasenya, el servidor els valida i retorna una resposta positiva. Si l'ordre implica la transferència de dades, aquesta es realitza pel canal de dades.

Poseu observar en els exemples de sessió FTP d'aquest mòdul els codis de resposta que es donen a les diverses ordres que realitza el client. Els trobareu a l'apartat "2.6 Modes d'accés al servidor".

El protocol FTP defineix un conjunt de respostes FTP consistents en un codi numèric de tres xifres i un text descriptiu de la resposta, com per exemple "250 Directory successfully changed". Tot el diàleg client/servidor té forma d'ordres i respostes.

2.2 Instal·lació i configuració del servidor

Hi ha moltes aplicacions FTP en el mercat, tant per a clients com per a servidors. Al mateix temps, hi ha versions de text i gràfiques per a cada cas. Hi ha moltes aplicacions que són de font pública i que es poden baixar gratuïtament.

Aplicacions FTP recomanables

Més que fer una enumeració de les aplicacions FTP actuals, us suggerim que busqueu a Internet quines aplicacions estan "de moda" i esbrineu les característiques que les fan especials.

La majoria de sistemes GNU/Linux proporcionen l'aplicació client **ftp** o **lftp** (*lftp* és una versió més nova i simple). També disposen d'una aplicació servidor, entre d'altres, anomenada **vsftpd** (*very secure FTP daemon* o dimoni FTP molt segur).

Una de les eines més usades per fer baixades FTP de repositoris públics són els navegadors web. Els navegadors web permeten utilitzar el protocol FTP per realitzar baixades, però no proporcionen totes les prestacions que pot arribar a tenir un client FTP específic. Per accedir a un servidor FTP amb un navegador, n'hi ha prou d'indicar l'URL del protocol i el servidor al qual es vol accedir (per exemple: ftp://ftp.rediris.es).

Així, doncs, quan parlem d'instal·lar el servei FTP fem referència al procés d'instal·lació i configuració del programari del servidor. Això es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS, HTTP...): es tracta d'instal·lar els paquets o *tarballs* (fitxers .tar) de l'aplicació servidor i fer-ne la configuració apropiada.

Per fer això cal plantejar-se els passos següents:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei FTP.
- Observar l'estat de la xarxa actual. Està ja el servei en funcionament? Existeix ja un servidor FTP instal·lat i actiu?
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i comprovar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

Cerca d'FTP a Internet

Usualment, l'administrador s'informa mitjançant el seu cercador preferit, per exemple Google, i de webs com la Viquipèdia. Proveu a buscar "FTP" o "FTP server" en aquests llocs web.

Usualment l'administrador acaba utilitzant l'aplicació servidor que li proporciona el mateix sistema operatiu que està utilitzant. Si utilitzeu Windows, l'empresa Microsoft ofereix una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor FTP o bé n'existeix algun de clàssic provinent d'Unix. De totes maneres en podeu obtenir d'altres a Internet.

L'eina que s'utilitzarà en aquesta unitat per oferir el servei FTP és l'aplicació **vsftpd** o *very secure FTP daemon*.

2.2.1 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per Internet paquets de servidor FTP usant eines com *yum* o *apt-get* i els repositoris de paquets apropiats segons quina sigui la distribució que utilitzin. A més, sempre poden usar els cercadors web per localitzar tot allò que els faci falta.

Un cop instal·lat el programari caldrà identificar què s'ha instal·lat. Quins paquets i què contenien. A vegades no s'instal·laran paquets sinó fitxers *tarball*, el contingut dels quals també caldrà saber examinar. És important saber identificar quins dels components instal·lats corresponen a fitxers executables, quins a fitxers de configuració i quins a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant, caldrà saber gestionar l'estat del servei (engegar, aturar, reiniciar...) i definir l'estat que ha de tenir en els diferents *runlevels* (nivells d'execució) del sistema.

En definitiva, el procediment d'instal·lar inclourà usualment:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo utilitzant l'eina apropiada segons quina sigui la distribució que s'utilitzi.
- Examinar el sistema per identificar quin programari, quins paquets, hi ha instal·lats relacionats amb el servei.
- Identificar els components del servei. Quins són els fitxers executables, quins els de configuració i quins els de documentació.
- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

2.3 Creació d'usuaris i grups

L'accés a un servidor FTP es pot considerar des de diferents angles. Des del punt de vista de la configuració de xarxes (ports, tallafocs, *TCP wrappers*...) es poden filtrar els amfitrions (*hosts*) que poden accedir al servidor FTP. Des del punt de vista dels usuaris, poden ser usuaris anònims, locals i virtuals.

Els **usuaris** es poden classificar en:

- Usuaris anònims
- Usuaris locals del sistema
- Usuaris propis del servei FTP, anomenats usuaris *virtuals*

L'accés anònim al servidor permet que qualsevol client pugui accedir a l'àrea pública del servidor FTP com a usuari "anonymous". Aquest dret d'accés de lectura es pot concedir o restringir. També se li pot proporcionar a l'usuari anònim el dret a publicar documentació dins de directoris amb els permisos apropiats.

Podeu trobar més informació sobre els usuaris anònims en l'apartat "Configuració de l'accés anònim".

L'accés al servei com a usuaris locals del sistema permet accedir al servei FTP als usuaris que ho són també del sistema on s'executa el servei. És per això que s'anomenen *usuaris locals*, perquè són locals a l'amfitrió on s'executa el servei. Aquest accés es pot permetre o restringir segons es desitgi.

Finalment hi ha l'accés d'usuaris virtuals. Es tracta d'usuaris identificats (no anònims) que no són (o no necessàriament) usuaris del sistema. Són usuaris propis del servei FTP i caldrà portar-ne la gestió pròpia amb les típics fitxers d'usuaris i contrasenyes.

2.3.1 Usuaris locals

Permetre l'accés al servei FTP d'usuaris locals és ben simple: simplement cal activar la directiva *local_enable*. Fet això, els usuaris locals tindran accés de descàrrega. Per permetre'ls pujar documents caldrà configurar més directives.

Els usuaris del sistema accedeixen al seu directori d'inici (*home*), que és el directori de publicació, i en general tenen dret a navegar per tot el sistema de fitxers de manera similar a com ho farien en una sessió d'usuari del sistema. Es pot engabiar (*chroot*) els usuaris en el seu directori d'inici, de manera que aquest passarà a ser l'arrel del sistema de fitxers al qual poden accedir via FTP i, per tant, no en podran sortir.

Els usuaris **locals** accedeixen al seu directori per defecte de publicació. Poden moure's únicament pel subarbre del seu *home* o per tot el sistema de fitxers segons si estan **engabiats** o no.

2.3.2 Usuaris virtuals

Generalment, tot servidor FTP permet un tercer tipus d'usuaris que no són ni els anònims ni els usuaris del sistema. Es tracta d'usuaris propis del servei FTP. L'avantatge d'aquest model és la separació entre els usuaris del servei i els usuaris del sistema. Això permet l'accés identificat a serveis FTP sense necessitat de crear comptes d'usuari en el sistema. També permet un grau més gran de portabilitat, ja que el servei es pot traslladar a un altre entorn (amb uns altres usuaris del sistema) i continuar mantenint els usuaris propis del servei. L'inconvenient d'aquest model és que generalment implica dur l'administració d'una gestió d'usuaris paral·lela a la del sistema i crear els fitxers o una base de dades d'usuaris FTP amb els noms i contrasenyes de cada un d'ells.

Els servidors FTP generalment permeten administrar **usuaris específics** del servei FTP (ni locals ni anònims). Això té l'avantatge de la independència respecte dels usuaris del sistema i presenta l'inconvenient de requerir una administració pròpia d'aquests usuaris.
En el servidor vsftpd aquests usuaris s'anomenen ***virtual users***.

Els usuaris virtuals són comptes d'usuaris que no existeixen realment com a usuaris del sistema. Això proporciona un grau major de seguretat, ja que un compte d'usuari virtual compromès únicament pot explotar les debilitats del servei FTP. En canvi, un compte del sistema compromès pot intentar explotar debilitats de tot el sistema. Una de les finalitats principals de la possibilitat de crear usuaris virtuals és permetre l'accés a contingut que ha de ser accessible a usuaris no validats en el sistema, però que no es vol fer públic per a tothom.

El procés que cal seguir per generar usuaris virtuals en el servidor vsftpd és el següent:

1. Crear la base de dades d'usuaris virtuals.
2. Crear/editar el fitxer de configuració PAM per usar la base de dades creada anteriorment.
3. Generar el directori de publicació de l'usuari virtual tot creant aquest usuari.
4. Generar el fitxer de configuració apropiat per permetre l'ús d'usuaris virtuals.

2.4 Configuració de l'accés anònim

El client que accedeix a un servidor FTP ho pot fer com a usuari identificat (del sistema o del servidor) o amb accés anònim. Tradicionalment, els usuaris anònims

tenen accés a un directori de publicació des d'on poden realitzar descàrregues però no pujar documents al servidor (tot i que es pot configurar per permetre-ho). Aquests usuaris s'identifiquen normalment amb el nom "anonymous" i com a contrasenya és costum que introdueixin el seu correu electrònic (vàlid o inventat).

L'**accés d'usuaris anònims** es fa usualment amb el nom d'usuari **anonymous** i una adreça de **correu electrònic** com a contrasenya. Sovint, els comptes anònims s'usen únicament per descarregar fitxers del directori de publicació, però també se'ls pot permetre publicar documents.

Quan els usuaris anònims poden publicar documents cal determinar en nom de qui ho fan, és a dir, quin és l'usuari, el grup i els permisos que s'assignen en el sistema de fitxers als documents pujats per aquests usuaris. També es pot configurar si se'ls permet crear directoris dins de l'arbre de publicació o no.

En general els elements que cal configurar en un servidor FTP relacionats amb els usuaris anònims són:

- Determinar si es permet l'accés als usuaris anònims.
- Determinar si se'ls concedeix el dret a pujar documents.
- Concedir-los o no el dret a crear directoris en el servidor (en l'àrea de publicació).
- Determinar l'usuari, el grup i la màscara amb els quals pujar els documents (si se'ls permet fer-ho).
- Engabiar (*chroot*) o no l'usuari en el seu accés al sistema de fitxers.
- Establir si cal demanar contrasenya als usuaris anònims.
- Generar una llista de contrasenyes (correus electrònics) no acceptades pel sistema.
- En el sistema de fitxers del servidor, establir els permisos a fitxers i directoris des dels quals es permeti descarregar documents i als quals es puguin pujar documents.

En fer-se la instal·lació del servidor FTP, usualment es crea de manera automatitzada un usuari anomenat **ftp**, que és l'usuari del sistema que representarà els usuaris anònims. Aquest usuari no té dret a realitzar sessions interactives amb el sistema, no té *shell* (**nologin**). Usualment el directori d'inici d'aquest usuari és el directori de publicació del servei FTP. Així, en el cas del servei vsftpd, aquest usuari ftp té com a directori d'inici **/srv/ftp** (tot i que pot variar en segons quines distribucions). És costum que dins d'aquest directori existeixi un directori anomenat **pub**, en el qual hi ha tota la documentació d'accés públic (les descàrregues permeses a tothom).

El codi següent mostra la línia que defineix l'usuari ftp en el fitxer d'usuaris del sistema:

```
root@server:/# grep "ftp" /etc/passwd
ftp:x:116:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
root@server:/#
```

L'usuari del sistema **ftp** és l'usuari usat en les connexions anònimes. Aquest usuari té el directori d'inici **/srv/ftp**.

2.5 Limitacions d'accés

Directives

El llistat de totes les directives es pot consultar amb l'ordre:

man 5 vsftpdconf.

Existeixen diverses directives que permeten establir múltiples aspectes del funcionament del servei FTP. Aquest apartat se centra en les directives que afecten als aspectes següents:

- Rendiment
- Mode de transferència
- Seguretat
- Mode del servei: autònom o xinetd
- Gestió dels *logs*
- Bàners globals i missatges de directori

2.5.1 Rendiment

El rendiment o *performance* permet configurar aspectes com el nombre màxim de connexions que pot atendre el servidor concurrentment, el nombre de connexions que es permet a un mateix client, l'ample de banda màxim permès en les connexions locals i en les anònimes... Aquestes són algunes de les directives relacionades amb el rendiment del servei:

- ***accept_timeout*** (60): estableix el *timeout* (o temps d'espera) en segons que té un client per establir connexions segures. Entre parèntesi se n'indica el valor per defecte
- ***anon_max_rate*** (0): estableix la taxa màxima de transferència que es permet als clients anònims. S'estableix en *bytes* per segon i per defecte val zero, que significa il·limitat.
- ***connect_timeout*** (60): estableix el temps màxim que té un client per respondre a una connexió de tipus PORT.

- ***data_connection_timeout*** (300): estableix el nombre màxim de segons en què una sessió de transferència de dades pot estar inactiva. Si se supera aquest límit sense progressar en la transferència la sessió es tanca.
- ***delay_failed_login*** (1): indica el nombre de segons de pausa abans d'indicar un error d'inici de sessió.
- ***delay_successful_login*** (0): indica el nombre de segons de pausa abans de permetre una connexió correcta.
- ***idle_session_timeout*** (300): estableix el nombre màxim de segons que una sessió pot estar inactiva. Passat aquest temps la sessió es tanca.
- ***local_max_rate*** (0): especifica la taxa màxima de transferència (en *bytes* per segon) que pot usar un usuari local. Per defecte val zero, que significa il·limitada.
- ***max_clients*** (0): estableix el nombre màxim de clients que es poden connectar concurrentment. Els clients següents rebran un missatge d'error informant que no es permeten més connexions. Per defecte pren el valor zero, que significa connexions il·limitades.
- ***max_login_fails*** (3): indica el nombre màxim d'intents d'inici de sessió fallits. Quan s'arriba a aquest número es tanca la sessió.
- ***max_per_ip*** (0): indica el nombre màxim de clients que es poden connectar simultàniament des de la mateixa adreça IP. Això permet establir límits al nombre de connexions d'un sol client basant-se en la seva adreça IP.
- ***one_process_model*** (YES): activa el model de funcionament *one process per connection*, que proporciona més velocitat de processament de les peticions client.

2.5.2 Mode d'accés

Els servidors FTP permeten treballar en mode ASCII i binari. Actualment, el servei es configura en mode binari per defecte. El mode ASCII pot presentar algunes vulnerabilitats de seguretat en fer un ús més intensiu dels recursos. Aquestes són algunes de les directives que controlen els modes d'accés:

- ***ascii_upload_enable*** (YES): estableix si es permet o no la transferència de dades en mode ASCII a més del mode binari. Activant aquesta directiva es permet pujar documents al servidor en mode ASCII.
- ***ascii_download_enable*** (YES): és la directiva anàloga a l'anterior per permetre o no les descàrregues en mode ASCII.
- ***async_abor_enable*** (NO): si s'activa, permet a clients FTP antics cancel·lar descàrregues a mig fer. D'altra manera, aquests clients es bloquejaven.

2.5.3 Seguretat

Hi ha diverses qüestions relacionades amb la seguretat d'un servei de xarxa a més de les connexions anònimes i els usuaris locals o virtuals. Algunes són aspectes com el rang de ports a usar, les connexions PASV i PORT, el permís per fer llistats, la configuració PAM a usar, els *TCP wrappers*...

Algunes de les principals directives que tracten aspectes relacionats amb la seguretat són:

- ***pasv_min_port*** (0): indica el mínim port permès per a connexions tipus PASV. Si s'utilitza conjuntament amb la opció *pasv_max_port* permet definir un rang (mínim-màxim) amb els valors de port dinàmic que es poden utilitzar en connexions passives.
- ***pasv_max_port*** (0): indica el port màxim que es pot usar en connexions de tipus PASV. En aquestes connexions s'utilitza un port dinàmic que sempre serà igual o inferior a l'especificat aquí.
- ***pasv_enable*** (YES): si es defineix a *NO*, no es permetran les connexions PASV, de tipus passiu. Per defecte pren el valor *YES*.
- ***port_enable*** (YES): si s'estableix a *NO* no es permetrà la transferència de dades en mode actiu, és a dir, usant el mètode PORT. Per defecte pren el valor *YES*.
- ***tcp_wrappers*** (NO): aquesta opció permet activar la seguretat de les connexions de xarxa usant *TCP wrappers*. Permet establir regles de connexió en funció dels noms d'amfitrions o de les seves adreces.
- ***ls_recurse_enable*** (NO): per defecte es desactiva la propietat de poder fer llistats recursius amb l'ordre *ls -R*. Això generalment es fa per evitar riscos de seguretat, ja que aquesta ordre implica un gran comsum de recursos que pot fer caure el servidor.
- ***local_umask*** (077): estableix el valor per defecte de la màscara de permisos per a aquells elements que publiquin els usuaris locals.
- ***connect_from_port_20*** (NO): indica si les connexions de transferència de dades de tipus PORT han d'usar obligatòriament el port 20 o no.
- ***dirlist_enable*** (YES): especifica si es concedeix el permís de fer llistats. Si s'assigna el valor *NO*, els clients no podran llistar els directoris.
- ***download_enable*** (YES). indica si es permet o no la descàrrega de continguts. Evidentment, el cas general és permetre la descàrrega, de manera que per defecte s'estableix a *YES*.
- ***hide_ids*** (NO): aquesta opció permet ocultar la informació d'usuari i grup en els llistats dels directoris de manera que es mostri sempre el nom *ftp*.

- ***hide_file*** (none): aquesta opció permet indicar patrons de noms de fitxer que seran exclosos dels llistats dels directoris. És a dir, els noms de fitxers i de directoris que coincideixin amb els patrons especificats no seran vistos pels clients.
- ***pam_service_name*** (vsftpd): indica el nom del servei PAM que s'utilitza. Si usualment la configuració PAM utilitza el directori /etc/pam.d i aquesta directiva s'estableix a vsftpd, significa que espera trobar un fitxer amb aquest nom dins del directori.
- ***ftp_username*** (ftp): especifica el nom de l'usuari real del sistema que s'utilitzarà quan es realitzin connexions anònimes. És a dir, els clients anònims són mapejats a aquest usuari.

2.5.4 Mode del servei: autònom o xinetd

El servidor vsftpd pot funcionar en mode autònom (*stand-alone*) o dins del superservei de xarxa xinetd. A més, es poden executar diverses instàncies del servidor per atendre diferents seus virtuals o per atendre amb més eficiència les peticions. Els següents són exemples de directives del mode autònom:

```
# Standalone mode
listen=YES

This tells vsftpd to run in standalone mode. Do NOT try and run vsftpd from
an inetd with this option set – it won't work, you may well get 500 OOPS:
could not bind listening socket.

One further note on standalone mode, regarding virtual IPs. This is very
easy – just run one copy of vsftpd per virtual IP (remembering to give each
a separate config file on the command line).
Distinguish which vsftpd is for which virtual IP with a setting like this
in the vsftpd.conf:

listen_address=192.168.1.2
```

- ***listen*** (YES): és la directiva que indica al servidor que ha de funcionar en mode autònom.
- ***listen_address*** (adreça IP): indica per quina adreça IP en concret escolta el servidor. Es poden engegar diversos servidors, cada un amb el seu propi fitxer de configuració i atenent a una adreça IP concreta. Així es poden implementar seus virtuals.

El fragment de codi següent mostra un exemple de fitxer de configuració del servei vsftpd dins del superdimoni de xarxa xinetd. Usualment, els serveis es configuren amb un fitxer amb el mateix nom que el servei dins del directori apropiat de xinetd, usualment /etc/xinetd.d.

```
# vsftpd is the secure FTP server.
service ftp
{
```

```
        disable                 = no
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/local/sbin/vsftpd
        per_source              = 5
        instances               = 200
        no_access               = 192.168.1.3
        log_on_success          += PID HOST DURATION
        log_on_failure          += HOST
}
```

- ***disable*** (no): indica que el servei ha de ser escoltat per el xinetd. Si pren el valor *YES*, el servei queda inhabilitat.
- ***socket_type*** (stream): indica que utilitza TCP.
- ***wait*** (no): indica si el servei és amb un sol *thread* o *multithread*. És a dir, si s'executa una sola instància del servei per atendre totes les peticions o si xinetd pot llançar múltiples instàncies. El valor *no* permet ser multithread.
- ***user*** (root): el servei s'executa en nom de l'usuari *root*.
- ***server*** (/usr/local/sbin/vsftpd): especifica el nom de l'executable del servidor.
- ***per_source*** (5): estableix un màxim de 5 connexions simultànies des d'un mateix client.
- ***instances*** (200): permet un màxim de 200 connexions concurrents, un màxim de 200 clients simultànis.
- ***no_access*** (192.168.1.3): denega l'accés al servidor a clients amb aquesta adreça IP origen.
- ***log_on_success*** (+= PID HOST DURATION): especifica el format dels *logs* quan es realitzen connexions satisfactòries. Es desa l'adreça IP, el PID i el temps invertit en la connexió.
- ***log_on_failure*** (+= HOST): especifica el format dels missatges de *log* per a les connexions fallides. Simplement s'enregistre l'adreça del client.

2.5.5 Logs

Els aspectes principals a descriure en les directives de *logs* són indicar si cal generar els *logs* de les connexions clients o no i establir el format que han de tenir els missatges. El següent és un llistat de les principals directives implicades:

```
# Activate logging of uploads/downloads.
xferlog_enable=YES

# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
```

```
7
8  # If you want, you can have your log file in standard ftpd xferlog format.
9  # Note that the default log file location is /var/log/xferlog in this case.
10 xferlog_std_format=YES
```

- ***xferlog_enable*** (YES/NO): indica si cal generar *logs* de les connexions que es realitzen.
- ***xferlog_file*** (/var/log/vsftpd.log): indica la ubicació i el nom del fitxer on es desaran els *logs*.
- ***xferlog_std_format*** (YES/NO): especifica que el format que han de tenir els missatges de *log* ha de ser l'estàndard.

2.5.6 Bàners i missatges

En el servidor FTP es poden especificar diversos bàners o missatges de capçalera, que es mostren segons convingui. Per exemple, és habitual mostrar un bàner de benvinguda als usuaris que inicien sessió en el servidor.

```
1  # You may fully customise the login banner string:
2  # This  string option allows you to override the greeting banner displayed by
3  # vsftpd when a connection first comes in.
4  ftpd_banner=Welcome to blah FTP service.
5
6  # This  option is the name of a file containing text to display when someone
7  # connects to the server. If set, it overrides the banner string provided by
       the
8  # ftpd_banner option.
9  banner_file=
```

Un element diferent dels bàners són els missatges (*messages*). En els directoris de publicació es poden posar fitxers anomenats *.message* que poden contenir, per exemple, una descripció dels continguts del directori. El servidor FTP mostra automàticament el contingut d'aquests fitxers en forma de capçalera quan l'usuari accedeix al directori.

```
1  # Messages, banners
2
3  # Activate directory messages - messages given to remote users when they
4  # go into a certain directory.
5  dirmessage_enable=YES
6
7  # This option is the name of the file we look for when a new directory is
8  # entered. The contents are displayed to the remote user. This option is only
9  # relevant if the option dirmessage_enable is enabled.Default: .message
10 message_file=
```

2.6 Modes d'accés al servidor

Verificar el funcionament del servidor és tan senzill com tractar de connectar des d'un client FTP al servidor i fer diverses sol·licituds en una mateixa sessió. Des de l'entorn de text, el mecanisme més simple per verificar el funcionament sempre és:

- Usar un client FTP text i realitzar les comprovacions.
- Tractar de simular un diàleg FTP usant una sessió Telnet. Com que el protocol FTP té la particularitat que utilitza dos ports, això serà una mica més difícil. Caldrà un *telnet* per al diàleg de control i un altre per a cada transferència de dades.

A part de realitzar una sessió FTP de prova per verificar el funcionament del servei, un administrador també ha de saber:

- Observar l'estat dels ports amb Nmap.
- Observar l'estat del servei.
- Monitorar el trànsit FTP amb utilitats tipus *ss*, *netstat* i IPTraf.
- Monitorar el trànsit amb Wireshark.

2.6.1 Sessió FTP

En la secció "Annexos" del web d'aquest mòdul trobareu una captura del trànsit de xarxa corresponent a una comunicació (diàleg) FTP.

En aquest apartat es mostra un exemple de sessió FTP usant el client text FTP i un exemple de sessió simulant el diàleg amb diverses connexions TCP amb *telnet*. En una mateixa sessió es poden realitzar diverses ordres de consulta, descàrrega i pujada de fitxers. També es combina la transferència de dades en mode actiu i passiu.

Exemple de diàleg FTP

El diàleg client-servidor té forma d'ordres i respostes, tal com es pot veure a continuació, on es realitza una connexió FTP i s'executen diverses ordres. S'utilitza el client FTP per defecte i un servidor vsftpd instal·lat al *localhost*. En aquesta sessió es realitza transferència de dades tant en mode **passiu** com en mode **actiu**.

```
1 Ordre      ports    comanda/resposta
2 [root@portatil ~]# ftp localhost
3     c49962 – s21    ... establiment de la connexió TCP...
4     s21 – c4992     Connected to localhost (127.0.0.1).
5     s21 – c4992     220 (vsFTPd 2.0.5)
```

```
6
7  Name (localhost): anonymous
8      c4992 – s21     ... enviar el nom d'usuari al servidor...
9                      USER anonymous <CRLF>
10     s21 – c4992     331 Please specify the password.
11
12 Password:
13     c49962 – s21    ... enviar la contrasenya al servidor...
14                     PASS usuari@fpoberta.cat
15     s21 – c4992     230  Login successful.
16     s21 – c4992     Remote system type is UNIX.
17     s21 – c4992     Using binary mode to transfer files.
18
19 ftp> cd pub
20     c49962 – s21    ... canviar al directori pub...
21                     CWD pub
22     s21 – c4992     250 Directory successfully changed.
23
24 ftp> dir
25     c49962 – s21     ... indicar el port dinàmic client ...
26                      PORT 127,0,0,1,137,108 <CRLF>
27     s21 – c49962     200 Port command succesfull
28     c49962 – s21     ... llistar el directori ...
29                      LIST <CRLF>
30 *   s20 – c35180     ... establiment connexió canal dades...
31 *   c35180 – s20     ... connexió TCP client/servidor...
32     s21 – c49962     150 Here comes the directory listing
33 *   s20 – c35180     total 2
34                      –rw–r––r–– 1 0 0 110 May 31 12:15 llistat
35                      –rw–r––r–– 1 0 0 362047 Feb 23 16:12 services
36                      ... tancament de la connexió de dades...
37     s21 – c4992      226 Directory send OK.
38
39 ftp> get carta.txt
40     c49962 – s21     ... descarregar el fitxer carta.txt...
41     c49962 – s21     ... demanar el mode passiu al server...
42                      PASV
43     s21 – c49962     227 Entering Passive Mode (127,0,0,1,142,120).
44     c49962 – s21     RETR carta.txt
45     s21 – c49962     150 Opening BINARY mode data connection
46                      for carta.txt (110 bytes)
47 *   c435181 – s36472 ... establiment connexió canal dades...
48 *   s36472 – c435181 ... connexió TCP client/servidor...
49 *                    ... transferència del contingut del fitxer...
50 *                    ... tancament de la connexió de dades...
51     s21 – c4992      226 File send OK
52
53 ftp> quit
54     c49962 – s21     QUIT <CRLF>
55     s21 – c49962     221 Goodbye.
```

El seguiment de la sessió és el següent:

- El client FTP es connecta al servidor via *localhost* i s'identifica com a usuari *anonymous* amb la contrasenya usuari@fpoberta.cat. Observeu que el servidor respon amb "230 Login successful": indica que el sistema operatiu és Unix i que el mode de transferència és binari.
- El client canvia al directori pub amb l'ordre d'usuari *cd pub*. Aquesta ordre correspon a l'ordre *CWD pub* del protocol FTP.
- El client demana fer un llistat amb l'ordre *dir*. Internament, el client aplica aquesta ordre fent dues ordres al servidor: PORT i LIST. Amb l'ordre *PORT 127,0,0,1,137,108*, el client indica al servidor quin és el seu port dinàmic. La

transferència del llistat es realitzarà del port 20 del servidor al port dinàmic del client. El mode de transferència és **actiu**.

- Tot seguit l'usuari client demana descarregar un fitxer amb l'ordre *GET carta.txt*. El client FTP transforma aquesta petició GET en dues ordres al servidor: PASV i RETR. La primera demana al servidor que operi en mode **passiu** i el servidor ho fa responent "227 Entering Passive Mode (127,0,0,1,142,120)". És a dir, el servidor informa de quin és el port (dinàmic) que utilitzarà per a la transferència de dades en lloc del port per defecte 20. Així, serà el client qui haurà de prendre la iniciativa d'iniciar la connexió de dades. L'ordre RETR fa que el servidor envii al client el fitxer sol·licitat. La comunicació, per tant, és en mode **passiu** del port dinàmic del servidor al port dinàmic del client.
- Finalment el client tanca la sessió amb l'ordre *quit*.

Mode actiu i mode passiu

Per activar el mode actiu s'utilitza l'ordre PORT A, B, C, D, PH, PL.

El client indica el seu port de dades especificant la pròpia adreça IP: A.B.C.D i el port. El port s'indica en dos octets PL i PH tals que el número del port correspon a l'expressió *port = PH 256 + PL*.

Per activar el mode passiu s'utilitza l'ordre PASV.

El client sol·licita al servidor treballar en mode passiu. El servidor comunica quin és el seu port de dades (que usarà en lloc del port 20). Emet una resposta amb la informació A, B, C, D, PH, PL, que indica la IP del servidor (A.B.C.D) i el port del servidor en el format *port = PH 256 + PL* (el valor del port s'obté dels octets PL i PH aplicant l'expressió indicada).

Exemple de diàleg FTP usant Telnet

La majoria de protocols TCP inicials d'Internet es poden simular usant Telnet al port pertinent i simulant manualment les ordres del protocol. En el cas del protocol FTP això és molt difícil, ja que requereix dues connexions, una de control i una de dades.

Si el servidor treballa en mode actiu és ell el que estableix la comunicació de dades del seu port 20 a un port dinàmic del client. Per poder recrear aquest diàleg caldria una aplicació que escoltés en el port del client (un Telnet escoltant per aquest port).

Si el servidor treballa en mode passiu és el client el que genera la nova connexió de dades al port especificat pel servidor. Això sí que es pot recrear amb un altre Telnet.

Calen dos Telnets per simular una transferència FTP en mode passiu:

- Un pel port 21, per realitzar el control de la comunicació (del client al servidor).
- L'altre (també del client al servidor) per realitzar la transferència d'un port dinàmic del client a un port dinàmic del servidor.

Vegem-ne un exemple:

Sessió 1: canal de control

```
[user@host ~]# telnet localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 (vsFTPd 2.0.5)

USER pere
331 Please specify the password.

PASS pere
230 Login successful.

PWD
257 "/home/pere"

LIST
425 Use PORT or PASV first.

PORT 127,0,0,1,231,175
200 PORT command successful. Consider using PASV.

LIST
150 Here comes the directory listing.
426 Failure writing network stream.

PASV
227 Entering Passive Mode (127,0,0,1,202,67)
*** iniciar sessió 2 en un altre terminal ***

LIST
150 Here comes the directory listing.
226 Directory send OK.

PASV
227 Entering Passive Mode (127,0,0,1,165,50)
*** iniciar sessió 3 en un altre terminal ***

RETR pere.info
150 Opening BINARY mode data connection for pere.info (70 bytes).
226 File send OK.

QUIT
221 Goodbye.
Connection closed by foreign host.
```

Sessió 2: llistat

```
[user@host ~]# telnet localhost 51779
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
-rw-r--r--    1 500      501          1695 Jun 01 17:19 nou.odt
-rw-rw-r--    1 500      501            70 Jun 01 17:14 pere.info
```

```
Connection closed by foreign host.
```

Sessió 3: '*get*' del fitxer

```
[user@host ~]# telnet localhost 42290
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
per crear un usuari com per exemple pere:
# useradd pere
# passwd pere
Connection closed by foreign host.
```

Per saber quines són les comandes possibles del protocol FTP cal consultar l'RFC del protocol.

El seguiment del diàleg és el següent:

- Fer un Telnet al port 21 del servidor, identificar-se amb les ordres del protocol USER i PASS. Esbrinar quin és el directori per defecte amb l'ordre PWD.
- En intentar fer un llistat del directori amb l'ordre LIST, el servidor contesta que primer cal executar l'ordre PORT o PASV. És a dir, que el client indiqui quin és el seu port de dades o que li mani al servidor treballar en mode passiu (perquè sigui el servidor qui indiqui el seu port dinàmic).
- El client indica el seu port de dades amb l'ordre *PORT 127,0,0,1,231,175*. Aquesta ordre indica la IP del client (127.0.0.1) i el port de dades a usar, el 59311 (231 * 256 + 175 = 59.311). En aquest punt el servidor, intenta una nova connexió TCP entre el seu port 20 i el port indicat pel client. Com que el client no atén aquesta connexió, es produeix un error i el servidor no realitza la transferència.
- En el tercer bloc del llistat Telnet, el client executa l'ordre PASV, que provoca que el servidor respongui indicant el port dinàmic de dades que utilitzarà en lloc del port 20. El servidor respon "227 Entering Passive Mode (127,0,0,1,202,67)". Indica la seva IP (127,0,0,1) i el port que utilitzarà: 202 * 256 + 67 = 51.779.
- En un altre terminal del client es pot iniciar una segona sessió Telnet a aquest port, tal com es pot veure a "Sessió 2: llistat": *telnet localhost 51779*. En aquesta connexió es rebrà la transferència del canal de dades de l'ordre que es realitzi a continuació (si és de transferència).
- En el canal de control es realitza l'ordre LIST i automàticament en la sessió 2 apareix el contingut del llistat i es tanca la connexió per part del servidor.
- Tot seguit es realitza el mateix procés per descarregar el fitxer pere.info. El client realitza l'ordre PASV en el canal de control perquè el servidor indiqui el port de dades que utilitzarà.
- Establir una tercera sessió fent un *telnet* al servidor al port que ha indicat (65 * 256 + 50 = 42.290). En fer l'ordre *RETR pere.info* (equivalent al GET del fitxer), el contingut del fitxer es transfereix a la sessió 3. El llistat que es mostra després de l'establiment de connexió correspon al contingut del fitxer.

- Per acabar, es tanca la connexió del canal de control amb l'ordre QUIT (equivalent al BYE).

2.7 Comunicacions segures

Una característica fonamental del protocol FTP és que quan es va dissenyar no es va tenir en compte la seguretat, més enllà de l'usuari i la contrasenya (que és pròpia del sistema operatiu). Com en altres protocols, la informació que porten els paquets viatja per la xarxa com a text pla, sense encriptar. És a dir, que qualsevol persona amb uns mínims coneixements de xarxes, pot capturar aquests paquets i espiar les dades (usuaris, contrasenyes, informació sensible...). Si, a més, fem servir un medi sense fils, encara s'és més vulnerable.

Com d'altres protocols, s'han afegit extensions per tal de xifrar les comunicacions. En aquest cas hi ha dues maneres de fer-ho: a través del protocol FTPS i el SFTP.

2.7.1 El protocol FTPS

El protocol **FTPS** (FTP sobre SSL/TLS, *FTP over SSL/TLS*) utilitza el mateix mecanisme que altres protocols (HTTPS, etc.). Utilitza la capa de seguretat SSL/TLS per fer que les comunicacions siguin xifrades i proporcionar la confidencialitat i autenticació necessàries per a les comunicacions FTP. Utilitza els mateixos ports que el protocol FTP (20 i 21).

El programari *vsftpd* ja porta preparats uns certificats per tal de poder xifrar les comunicacions, tot i que es poden substituir per d'altres. Per tal d'activar el protocol FTPS cal descomentar les següents línies i reiniciar el servei:

```
ssl_enable=yes
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

Vegem amb més detalls les directives:

- ***ssl_enable***: habilita SSL per tal d'activar el protocol FTPS.
- ***rsa_cert_file***: conté el certificat públic.
- ***rsa_private_key_file***: conté la clau privada.

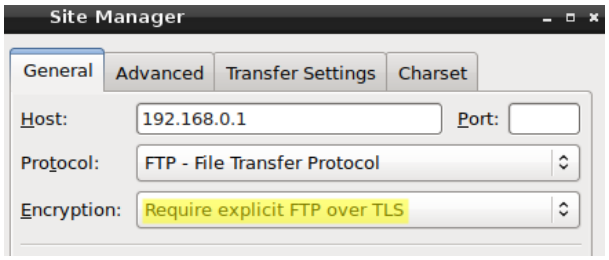
El client normal d'FTP que porten la majoria de distribucions no suporta SSL. Cal instal·lar algun client de text que ho suporti, com *ftp-ssl* o *lftp*.

```
usuari@client:~$ ftp 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 3.0.3)
```

```
Name (10.0.2.15:usuari): usuari
530 Non-anonymous sessions must use encryption.
Login failed.
421 Service not available, remote server has closed connection
ftp> quit
usuari@client:~$ ftp-ssl 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 3.0.3)
Name (10.0.2.9:usuari): usuari
234 Proceed with negotiation.
[SSL Cipher ECDHE-RSA-AES256-GCM-SHA384]
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Amb els clients gràfics també cal indicar que s'usa SSL, tal com es pot veure en la figura 2.2.

Figura 2.2. Connexió a un servidor amb suport FTPS



2.7.2 El protocol SFTP

El protocol **SFTP** és una implementació diferent del protocol FTP. De fet, és una extensió del protocol SSH (*Secure SHell*), que és qui realment ofereix el xifrat. En aquest cas, el protocol utilitza el port 22, que és el d'SSH. Aquest protocol **no** és compatible amb l'FTPS.

No s'ha de confondre amb el protocol *Simple File Transfer Protocol* (protocol simple per a la transferència de fitxers), ja que coincideixen les sigles SFTP. Aquest era una versió lleugera del protocol FTP que utilitzava el port 115 (RFC 913) i que avui en dia no s'utilitza, ja que en el seu lloc es fa servir el TFTP (Trivial FTP).

Per poder utilitzar el protocol SFTP cal instal·lar un altre programari, com per exemple la *suite openSSH*, que incorpora un propi servidor FTP.

```
root@server:/# apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openssh-sftp-server
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass ufw
The following NEW packages will be installed:
  openssh-server openssh-sftp-server
0 upgraded, 2 newly installed, 0 to remove and 182 not upgraded.
```

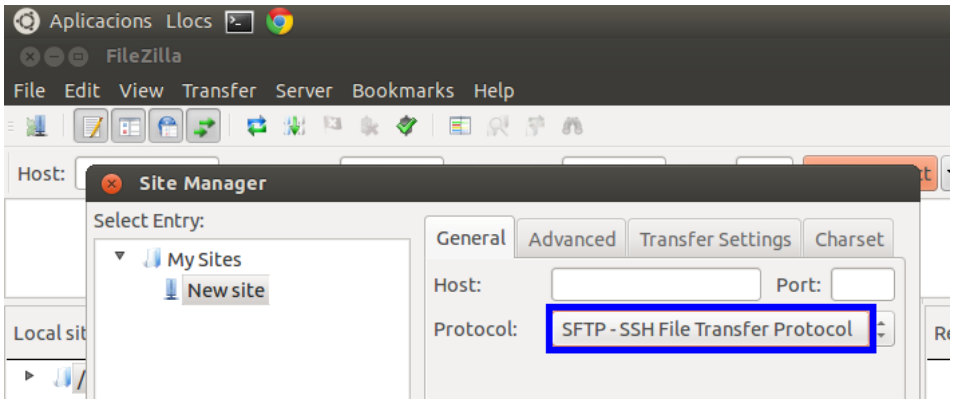
```
Need to get 397 kB of archives.
After this operation, 1609 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

I des d'un client ja ens podem connectar al servidor amb la comanda *sftp*:

```
usuari@client:~$ sftp 10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:2cqYKzks+fFtaQutgx8jLfvy8X08lEzxgPdkXYg2DKw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
usuari@10.0.2.15's password:
Connected to 10.0.2.15.
sftp>
```

En el cas de voler-se connectar amb un client gràfic com, per exemple, FileZilla, també cal indicar que s'usa SFTP, tal com es pot veure en la figura 2.3:

Figura 2.3. Connexió a un servidor amb suport SFTP



2.8 Clients gràfics i de text

Abans de la popularització del World Wide Web, el servei FTP era profusament usat per a l'intercanvi d'informació i la transferència de fitxers, generalment en mode text o consola, ja que aquest era el mode més comú de treball. Més endavant van anar apareixent eines gràfiques que permetien gestionar còmodament, amb pocs clics, les descàrregues. Avui en dia les eines gràfiques han quedat superades pels omnipresents navegadors, la interfície amb la qual la majoria d'usuaris dialoguen amb el món.

2.8.1 Clients de text

Alguns dels clients més populars per treballar en mode text són:

- **ftp**: el client FTP, utilitat incorporada en tots els sistemes GNU/Linux del món.

- **wget**: una de les eines actualment més populars per a la descàrrega de continguts d'FTP i HTTP (entre d'altres).
- **SFTP**: eina del paquet ofimàtic SSH que permet la connexió FTP segura.

Client FTP

Un client FTP es pot connectar molt fàcilment des del mode d'ordres de la majoria de sistemes operatius. Únicament s'ha d'executar l'ordre *ftp <servidor>*, i automàticament s'inicia una connexió entre el client i el servidor indicat. Usualment, l'aplicació client permet un ús interactiu i es poden obrir i tancar sessions i treballar en diferents servidors al gust del client.

Vegeu un exemple de sessió client:

```
$ ftp ftp.uoc.net            # inicia una sessió al servidor ftp.uoc.net
> quit                       # finalitza la sessió al servidor ftp.uoc.net

$ ftp                        # engega l'aplicació client
> open ftp.rediris.es        # connecta al servidor FTP de RedIRIS
> get carta.txt              # descarrega el fitxer carta.txt
> get file1.txt /tmp/nou.txt # desa file1.txt a /tmp amb el nom nou.txt
> quit                       # finalitza la sessió al servidor atual
> open servidor_nou          # inicia una connexió a un altre servidor
> put /tmp/nou.txt           # puja el fitxer al directori actiu amb el nom nou
    .txt
> cd dir1                    # canvia el directori de destinació a dir1
> put carta.txt              # puja el fitxer carta.txt al directori actual (
    dir1)
```

El client disposa de les ordres FTP que el servidor implementi. No necessàriament s'implementen totes les ordres descrites en el protocol. A més, l'usuari pot disposar de més ordres si el client i el servidor permeten extensions del protocol (utilitats addicionals).

Les ordres més usuals són ***get***, ***mget***, ***put*** i ***mput*** per baixar i pujar fitxers; ***cd*** i ***ls*** per canviar i llistar directoris; ***ascii*** i ***binary*** per indicar el mode de transferència; ***!ordre*** per executar una ordre de sistema operatiu en el servidor, i ***help*** per mostrar la llista d'ordres.

A continuació es mostra la llista d'ordres que implementa el servidor al qual ens hem connectat:

```
ftp> help
Commands may be abbreviated. Commands are:
! debug mdir  sendport  site
$       dir     mget    put     size
account disconnect mkdir pwd    status
append  exit    mls     quit    struct
ascii   form    mode    quote   system
bell    get     modtime recv    sunique
binary  glob    mput    reget   tenex
bye     hash    newer   rstatus tick
case    help    nmap    rhelp   trace
cd      idle    nlist   rename  type
cdup    image   ntrans  reset   user
chmod   lcd     open    restart umask
close   ls      prompt  rmdir   verbose
cr      macdef  passive runique ?
```

```
delete  mdelete proxy   send
```

Aquest és un altre exemple de sessió de text amb un client FTP:

```
[pere@host ~]# ftp localhost
Name (localhost:root): pere
Password:

ftp> pwd
257 "/home/pere"
ftp> ls
-rw-rw-r-- 1 500 501 70 Jun 01 17:14 pere.info

ftp> cd /tmp
ftp> ls
-rw-rw-rw- 1 0 0 1695 Jun 01 17:15 fitxa.odt

ftp> !pwd
/root

ftp> get fitxa.odt
local: fitxa.odt remote: fitxa.odt

ftp> cd ~
250 Directory successfully changed.

ftp> pwd
257 "/home/pere"

ftp> !pwd
/root
ftp> get pere.info
local: pere.info remote: pere.info

ftp> put fitxa.odt nou.odt
local: fitxa.odt remote: nou.odt

ftp> put pere.info /tmp/pere.txt
local: pere.info remote: /tmp/pere.txt

ftp> cd pub
ftp> put pere.info
local: pere.info remote: pere.info
553 Could not create file.

ftp> bye
221 Goodbye.
```

El seguiment del diàleg mostra les accions següents:

Ús del servei FTP

L'ús del servei FTP exigeix una bona gimnàstica mental, ja que l'usuari ha de ser conscient en tot moment de quin és el seu sistema de fitxers local i quin és el sistema remot. La majoria d'ordres tenen una versió per al sistema de fitxers local (*lcd* o "local cd", per exemple) i una altra per al sistema remot (*cd*).

- Connectar al servidor com a usuari identificat: pere. El directori actiu en el servidor FTP és el dirctori d'inici de l'usuari (en aquest exemple /home/pere), com mostra l'ordre ***pwd***.
- Canviar el directori actiu en el servidor amb l'ordre ***cd***. Observar que en el client el directori actiu és un altre, de fet en el client sembla que som l'usurari *root* en el seu dirctori d'inici. L'ordre "*!pwd*" executa en el *shell* del client l'ordre que se li mani.
- La instrucció *get fitxa.odt* descarrega el fitxer d'aquest nom del servidor i el desa en el directori actiu del client amb el mateix nom.

- Es torna a canviar de directori actiu en el servidor (es retorna al dirctori d'inici) i es descarrega el fitxer pere.info. Es desa novament amb el mateix nom en el directori actiu en el client.
- L'ordre *put fitxa.odt nou.odt* permet desar el fitxer fitxa.odt que hi ha en el directori actiu del client amb un nom nou en el directori actiu del servidor.
- Es pot pujar un fitxer indicant tant la ubicació origen en el client com la ubicació de destinació en el servidor. L'ordre *put pere.info /tmp/pere.txt* desa el fitxer pere.info que hi ha en el directori actiu del client a /tmp en el servidor amb el nom pere.txt.
- L'últim exemple mostra que no es pot pujar un fitxer a una ubicació en què no es disposa de permisos per fer-ho.

Per obtenir informació d'una ordre en concret del client FTP es pot consultar la pàgina del manual del client, o també es pot usar l'ordre *help* dins de l'intèrpret client.

```
[user@host ~]$ ftp
ftp> help get
get         receive file
ftp> help mget
mget        get multiple files
```

Client Wget

La utilitat principal de Wget és descarregar un fitxer o un conjunt de fitxers d'un servidor FTP (també d'altres protocols) de forma desatesa, és a dir, sense fer-ho interactivament fitxer a fitxer. Si no indiquem l'usuari i la contrasenya amb els quals ens volem connectar, es realitza una connexió anònima.

```
root@server:/# wget ftp://ftp.rediris.es/welcome.msg
--2020-09-26 14:41:11--  ftp://ftp.rediris.es/welcome.msg
           => 'welcome.msg'
Resolving ftp.rediris.es (ftp.rediris.es)... 130.206.13.2, 2001:720:418:cafd::2
Connecting to ftp.rediris.es (ftp.rediris.es)|130.206.13.2|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE welcome.msg ... 93
==> PASV ... done.    ==> RETR welcome.msg ... done.
Length: 93 (unauthoritative)

welcome.msg
     100%[=============================================================>]      93
       --.-KB/s    in 0s

2020-09-26 14:41:12 (3.42 MB/s) - 'welcome.msg' saved [93]

root@server:/#
```

El llistat següent mostra, d'entre totes les opcions de wget, les referides al servei FTP:

```
root@server:/# wget -h
GNU Wget 1.10.1, un baixador de xarxa no interactiu.
```

```
Forma d'ús: wget [OPCIÓ]... [URL]...
... output suprimit ...

Opcions de FTP:
       --ftp-user=USUARI       estableix l'usuari d'FTP a USUARI.
       --ftp-password=PASS     estableix la contrasenya d'FTP a PASS.
       --no-remove-listing     no suprimir els fitxers «.»listing.
       --no-glob               inhabilita l'ús de comodins de fitxers per a FTP
            .
       --no-passive-ftp        inhabilita el mode de transferència passiu.
       --preserve-permissions  manté els permisos del fitxer remot
       --retr-symlinks         en mode de recursió, baixa els fitxers
                               apuntats per enllaços simbòlics que no siguen
                               directoris
... output suprimit ...
```

Client SFTP

La utilitat SFTP és molt potent i és utilitzada per navegadors de fitxers (per exemple, Nautilus) per poder mostrar sistemes de fitxers remots connectats per FTP. Aquest és el seu format:

```
root@server:/# sftp -h
usage: sftp [-46aCfpqrv] [-B buffer_size] [-b batchfile] [-c cipher]
          [-D sftp_server_path] [-F ssh_config] [-i identity_file] [-l limit]
          [-o ssh_option] [-P port] [-R num_requests] [-S program]
          [-s subsystem | sftp_server] destination
```

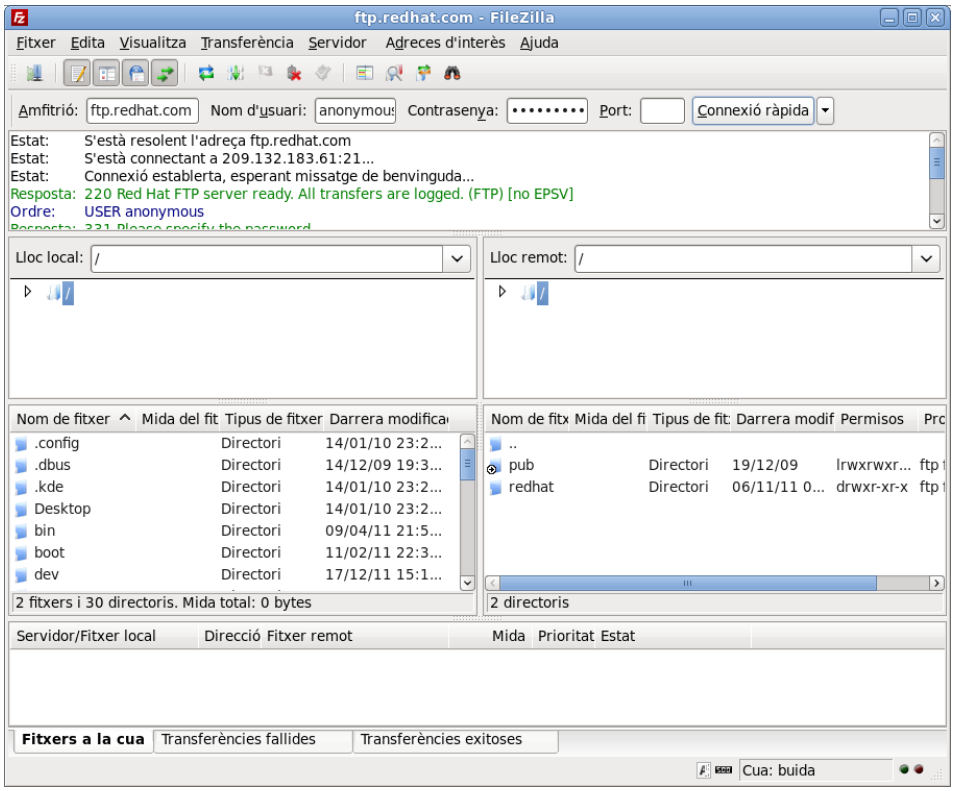
Vegeu un exemple de sessió client usant SFTP per realitzar una transferència segura de fitxers:

```
root@server:/# sftp pere@localhost
Connecting to localhost...
pere@localhost's password:

sftp> pwd
Remote working directory: /home/pere

sftp> ls
nou.odt    pere.info

sftp> put pere.info /tmp/pere.bak
Uploading pere.info to /tmp/pere.bak
pere.info                                 100%   70     0.1KB/s   00:00

sftp> get /tmp/pere.bak
Fetching /tmp/pere.bak to pere.bak
/tmp/pere.bak                             100%   70     0.1KB/s   00:00
```

2.8.2 Clients gràfics

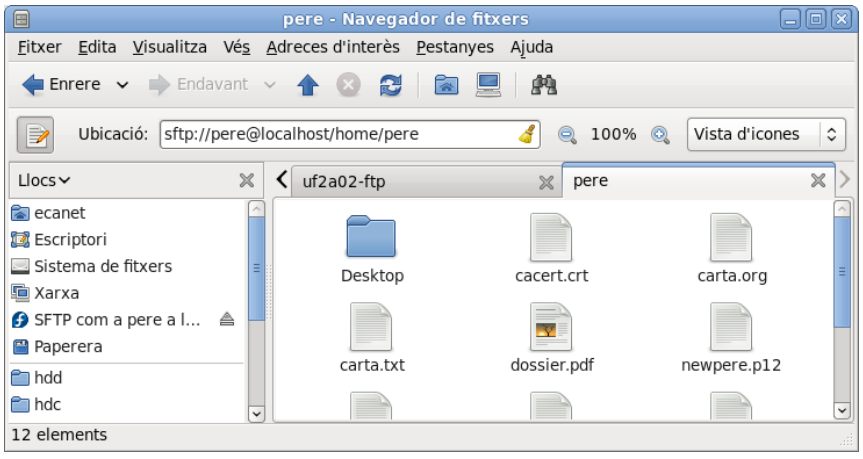
Actualment hi ha una gran varietat de clients gràfics que tenen més o menys èxit. Un dels més populars és FileZilla. En la figura 2.4 podem observar el client FileZilla connectat a la seu ftp.redhat.com.

FIGURA 2.4. Pantalla del client gràfic FileZilla



En el sistema GNU/Linux usualment hi ha algun tipus d'aplicació per navegar pel sistema de fitxers, com Konqueror o Nautilus, depenent de l'escriptori instal·lat. Aquestes eines permeten no només navegar per sistemes de fitxers, sinó també connectar a recursos externs per FTP o SFTP. La figura 2.5 mostra un exemple del funcionament:

FIGURA 2.5. Pantalla del client Nautilus connectat a un recurs per SFTP



2.8.3 El navegador com a client

Avui dia una de les eines imprescindibles per als usuaris d'Internet és el navegador (normalment considerat navegador web). Cada usuari utilitza el que prefereix. Alguns dels més destacats actualment són Firefox i Chrome. Els navegadors

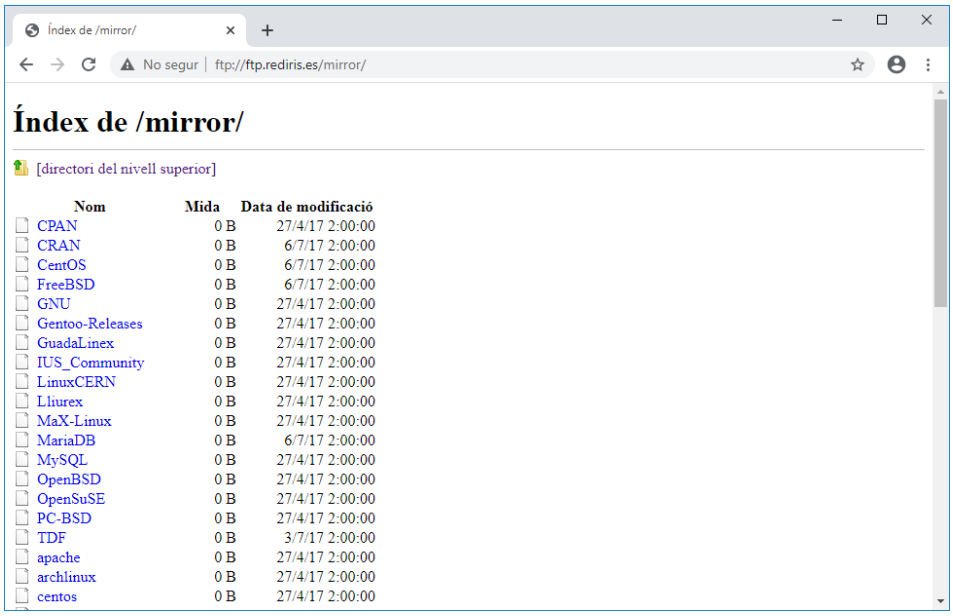
permeten accedir a molts tipus de continguts diferents, entre els quals s'inclouen recursos FTP.

Simplement cal introduir l'URL indicant l'esquema (*schema*) adequat:

```
ftp://<url>
```

La figura 2.6 mostra un exemple de mirall (*mirror*) de descàrregues per FTP per a diferents distribucions de Linux. Els usuaris poden descarregar-se sistemes operatius usant un client navegador qualsevol, com Firefox o Chrome.

FIGURA 2.6. Pàgina de descàrregues per FTP de diferents distribucions de Linux



Correu electrònic i missatgeria

Eduard Canet i Ricart

Índex

Introducció

En aquesta unitat del mòdul *Serveis de xarxa i Internet* es presenta de manera molt exhaustiva el funcionament del correu electrònic o *e-mail*. Es mostren tots els elements que intervenen en aquest tipus de comunicacions; es detalla el funcionament dels protocols SMTP, POP i IMAP; s'estudia el format dels missatges, i també els missatges MIME. A part del correu electrònic, s'expliquen altres formes de comunicació com les llistes de correu i de notícies o la missatgeria instantània, que són molt populars.

En molts aspectes el correu electrònic imita el funcionament del correu postal. És un sistema distribuït que permet als usuaris enviar missatges a un destinatari final. El correu electrònic ha tingut una gran evolució des dels primers sistemes, que únicament permetien intercanviar missatges de text ASCII, fins als correus electrònics amb continguts multimèdia d'avui en dia.

L'apartat "Instal·lació i administració del servei de correu electrònic" mostra el funcionament i els elements que participen en l'enviament de missatges de correu. En el servei de correu es diferencia molt clarament entre el mecanisme de transport dels missatges i els missatges. S'analitzen els mecanismes dels protocols SMTP, POP i IMAP per tal de comprendre amb profunditat el seu funcionament, i s'aprendrà a instal·lar i configurar programari capaç d'implementar aquests protocols.

Un altre aspecte tractat és el de la seguretat dels missatges i de la comunicació. S'exposen els mecanismes per xifrar i autenticar missatges, proporcionant les prestacions de xifrat, autenticació, integritat i no-repudi. Per obtenir aquestes propietats cal familiaritzar-se amb els certificats digitals. Les comunicacions segures entre dos interlocutors es basen principalment en la utilització de capes de transport segur com SSL i TLS. S'exposarà com configurar servidors SMTP, POP i IMAP per permetre connexions segures a través dels certificats digitals.

En l'apartat "Instal·lació i administració de serveis de missatgeria instantània, notícies i llistes de distribució" es descriu el funcionament i la implementació i configuració de cada un d'aquests tipus de serveis. Es realitza una comparativa entre cada servei per entendre bé la seva funcionalitat, el seu propòsit. També s'explica com instal·lar i configurar el servei de missatgeria instantània i com utilitzar clients gràfics i de text. El servei de missatgeria es basa en el protocol XMPP, també anomenat JABBER, del qual se n'expliquen les característiques més rellevants.

Així mateix, es mostra com instal·lar i configurar el servei de llistes de distribució o llistes de correu i els mecanismes que s'han anat utilitzant al llarg del temps (des d'àlies fins als servidors actuals). Es classifiquen les llistes segons la seva funcionalitat o tipus i l'accés que permeten. També es donen exemples de funcionament i administració.

La immensa popularitat del WWW i la utilització de llistes de correu han fet perdre importància al servei de notícies. Basat en el protocol NNTP, el servei de notícies permetia publicar articles com qui publica anuncis en un tauler. És un mecanisme per publicar i compartir informació sense que calgui indicar els destinataris.

La majoria d'usuaris no han utilitzat mai el servei de notícies, però segurament no n'hi ha cap que no hagi xatejat en algun moment o altre. Els sistemes de missatgeria instantània, començant pel mòbil i acabant pels televisors actuals que permeten connexions amb Skype, són àmpliament utilitzats. Permeten una comunicació immediata entre usuaris identificats, fins i tot amb àudio i vídeo. Es mostra com realitzar una apropiada gestió dels comptes d'usuari i verificar l'accés als serveis de missatgeria instantània, notícies i llistes de distribució. També s'explica com elaborar documentació relativa als procediments tractats al llarg de l'apartat.

Per assolir els objectius d'aquesta unitat és molt recomanable que feu les activitats i els exercicis d'autoavaluació proposats en cadascun dels apartats de la unitat.

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'estudiant:

1. Administra servidors de correu electrònic, aplicant criteris de configuració i garantint la seguretat del servei.
    - Descriu els diferents protocols que intervenen en l'enviament i recollida del correu electrònic.
    - Instal·la i configura un servidor de correu electrònic.
    - Crea comptes d'usuari i en verifica l'accés.
    - Estableix i aplica mètodes per impedir usos indeguts del servidor de correu electrònic.
    - Instal·la serveis per permetre la recollida remota del correu en les bústies d'usuari.
    - Usa clients de correu electrònic per enviar i rebre correu des dels comptes creats en el servidor.
    - Utilitza la signatura digital i el correu xifrat.
    - Configura el servidor de correu com a servei segur.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'utilització del servei.
2. Administra serveis de missatgeria instantània, notícies i llistes de distribució, verificant i assegurant l'accés dels usuaris.
    - Descriu els serveis de missatgeria instantània, notícies i llistes de distribució.
    - Instal·la i configura el servei de missatgeria instantània.
    - Utilitza clients gràfics i de text de missatgeria instantània.
    - Instal·la i configura el servei de notícies.
    - Instal·la i configura el servei de llistes de distribució.
    - Determina el tipus de llista i els modes d'accés permesos.
    - Crea comptes d'usuari i en verifica l'accés als serveis de missatgeria instantània, notícies i llistes de distribució.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'ús dels serveis de missatgeria instantània, notícies i llistes de distribució.

1. Instal·lació i administració del servei de correu electrònic

En molts aspectes el correu electrònic imita el funcionament del correu postal. És un sistema distribuït que permet als usuaris enviar missatges a un destinatari final. El correu electrònic ha evolucionat molt dels primers sistemes que únicament permetien intercanviar missatges de text ASCII als correus electrònics amb continguts multimèdia d'avui en dia.

ASCII

Acrònim amb el qual es coneix l'American Standard Code for Information Interchange, el codi de caràcters establert com a estàndard americà el 1963 i que va esdevenir *de facto* l'estàndard mundial.

En el servei de correu es diferencia molt clarament entre el mecanisme de transport dels missatges i els missatges. El mecanisme de transport dels missatges és el protocol SMTP, i és independent del format i el contingut del missatge. Els missatges originals eren en text pla ASCII de 7 bits, però actualment en un missatge es permet tot tipus de contingut. Això és possible gràcies als tipus MIME, que descriuen i codifiquen els missatges.

El mateix disseny del sistema de correu ha evolucionat a mesura que ha avançat la tecnologia a internet. En el model bàsic de transport per SMTP s'exigeix que el receptor disposi de connexió permanent i que es connecti al servidor de correu localment per tal de consultar-lo. Quan els usuaris tenen accés a internet mitjançant un ISP (proveïdor de serveis d'internet) volen poder baixar tot el correu de cop i examinar-lo un cop tancada la connexió (per no pagar connexió telefònica). El protocol POP proporciona el mecanisme per descarregar del servidor de correu els missatges de l'usuari.

Amb la popularització d'internet s'abaixen els costos de connexió. L'usuari s'acostuma a baixar el correu des de llocs diferents, però usant el POP té l'inconvenient que el correu li queda repartit per diverses màquines. Cal un mecanisme que permeti accedir i gestionar el correu i les bústies directament en el servidor. El protocol IMAP ho fa possible.

Tot això ha canviat amb la popularització d'internet a tots els nivells i usant tota mena de dispositius. La major part del correu electrònic funciona actualment per web, gràcies a proveïdors *webmail* com els coneguts Gmail o Yahoo.

1.1 Protocols de correu electrònic

El correu electrònic és un dels primers serveis que es va utilitzar a les xarxes i un dels més populars a internet. Ha evolucionat molt des dels primers sistemes, que podien intercanviar únicament missatges de text ASCII (7 bits), fins als portals web usats avui dia per milions d'usuaris per enviar-se continguts multimèdia.

Per obtenir més informació sobre l'especificació del protocol SMTP en els RFC 821, 822, 2821 i 2822, aneu a la secció "Adreces d'interès" del web del mòdul.

El 1982 es van desenvolupar els estàndards que defineixen el correu electrònic. Es descriuen en els documents RFC 821, que explica el protocol de transmissió,

Per conèixer més detalls de l'estàndard MIME, consulteu la secció "Els tipus MIME".

i RFC 822, que descriu el format dels missatges. Aquests dos estàndards han evolucionat i actualment s'utilitzen els documents RFC 2821 i RFC 2822. A més, per permetre missatges multimèdia s'ha definit l'estàndard MIME en el document RFC 2231.

L'especificació original distingeix molt clarament entre el mecanisme de transport dels missatges i els missatges. El mecanisme de transport dels missatges és el **protocol SMTP**.

El protocol SMTP (*Simple Mail Transport Protocol* o **protocol simple de transport de correu**) és independent del format i el contingut del missatge. El missatge es compon del **sobre** (*envelope*) i el **contingut**, format per les capçaleres i el cos del missatge.

El correu electrònic és un sistema distribuït que permet als usuaris enviar missatges a un destinatari final. Hi intervenen diversos agents:

Ambigüitat dels agents

Els agents que intervenen en el sistema de correu electrònic fan sovint més d'un paper, fet que provoca ambigüitat en la seva definició.

Servidor SMTP

Sovint el programari de servidor SMTP (per exemple, Sendmail) fa tant la funció de client (emissor de missatges) com de servidor (receptor de missatges).

- **MUA** (*Mail User Agent* o **agent de correu d'usuari**). L'usuari utilitza un MUA per redactar, rebre i manipular correus electrònics. Un MUA és un programari que permet aquestes capacitats, que poden ser aplicacions en línia d'ordres (per exemple, l'ordre *mail* d'Unix), aplicacions de text (per exemple, Mutt o Pine), interfícies gràfiques (com Thunderbird) i portals de correu web (com Gmail o Yahoo). L'usuari interactua usualment amb un MUA. En el cas d'enviar un correu, el MUA lliura el missatge al sistema de transport (SMTP) per fer-lo arribar al destinatari. En el cas de recepció de correu, el MUA obté el missatge d'una bústia de correu (on l'ha dipositat l'SMTP) i el mostra a l'usuari.
- **MTA** (*Mail Transport Agent* o **agent de transport de correu**). L'agent de transport de correu és l'encarregat de transportar els missatges al destinatari indicat. Aquesta tasca la fa el protocol **SMTP**. L'MTA rep el missatge d'un MUA i s'encarrega del seu transport fins al destinatari final. Generalment realitzen la funció de client/servidor o emissor/receptor al mateix temps. La funció que es realitza en cada cas és:
  - **MTA client SMTP (emissor)**. S'anomena client de correu o emissor (segons l'arquitectura client/servidor) el servidor SMTP (fixeu-vos en l'ambigüitat) que envia el correu al destinatari. És qui envia el correu utilitzant el protocol SMTP. Estableix les connexions amb els servidors/receptors SMTP.
  - **MTA servidor SMTP (receptor)**. S'anomena servidor de correu o receptor el programari de servidor SMTP que rep els missatges de correu entrant i els lliura a la bústia del destinatari si es tracta d'un lliurament local, o els reenvia a un altre servidor SMTP si va destinat a un sistema remot. El fet que un receptor MTA rebi un correu no significa que el missatge hagi arribat al destinatari final.
- **MDA** (*Mail Delivery Agent* o **agent de lliurament de correu**). Un altre element en l'estructura de correu és el MDA. És l'encarregat de fer el

lliurament final del missatge a la bústia del destinatari. En el procés pot realitzar diverses accions segons un conjunt de regles *.forward* definibles per l'usuari. Un exemple d'MDA és el programa procmail, que permet filtrar els missatges entrants per posar-los en una bústia o una altra, esborrar-los, marcar-los com a correu brossa (*spam*), fer-ne còpies, reenviar-los a altres bústies i a altres destinataris... Usualment, en sistemes de correu que no disposen d'MDA és el mateix MTA el que diposita el missatge a la bústia del destinatari final.

@ (arrova)

Significa "at" en anglès o "a" en català. Usualment la composició d'una adreça de correu electrònic es descriu com a *usuari@màquina* (usuari tal a la màquina qual), però el nom del domini no correspon necessàriament al nom de la màquina. És més correcte dir *usuari@domini*.

Per exemple, pere@gmail.com indica el compte de correu d'en Pere en la màquina gmail.com. Però, de fet, no hi ha cap màquina que es digui així, sinó que és el compte de correu d'en Pere en el domini gmail.com. En realitat, Gmail té diverses màquines que responen a aquest nom de domini.

També hi ha altres conceptes que intervenen en el sistema de correu electrònic:

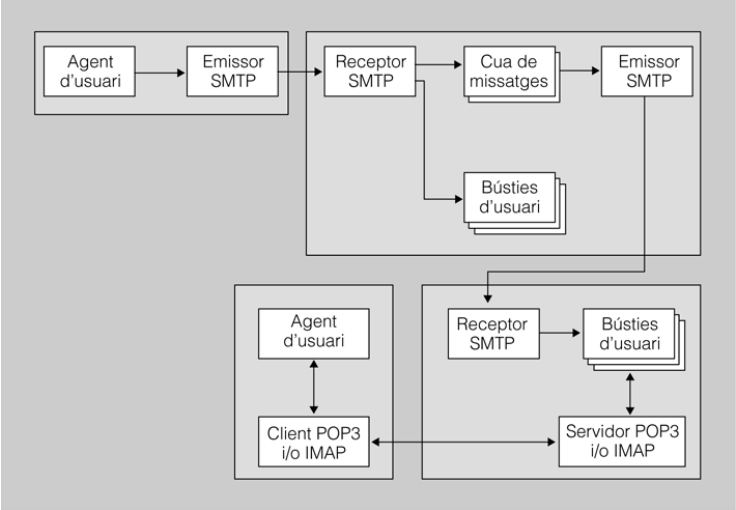
- **Adreça de correu**. Els usuaris que volen utilitzar el sistema de correu electrònic han de disposar d'una bústia en un servidor de correu. Els missatges s'adrecen utilitzant la coneguda nomenclatura **usuari@domini-servidor-correu**, que es llegeix com a "compte de correu de l'usuari tal en el domini qual". Així, per exemple, a un usuari amb un compte de correu de nom pere en el domini ioc.cat se li poden adreçar missatges a l'adreça pere@ioc.cat.

- **Bústia de correu** o *mailbox*. Els usuaris tenen bústies en un servidor de correu. Quan el servidor de correu MTA rep un missatge destinat a un usuari amb compte de correu en el mateix servidor, el diposita a la bústia de correu corresponent (si no hi ha un MDA pel mig). Fixeu-vos que dipositar el missatge en la bústia de l'usuari no garanteix que l'usuari el llegeixi. Cal un altre pas, que és la recuperació del missatge de la bústia per part de l'usuari. Aquest pas es realitza des d'un MUA i sovint empra protocols com **POP** o **IMAP**, fora de l'abast de les explicacions del correu SMTP.

- **Llista de correu i àlies**. Els àlies i les llistes de correu es tradueixen en adreces de comptes de correu. Si la llista de correu es gestiona localment, el MUA local l'expandirà en el conjunt d'adreces d'usuari corresponents i enviarà el correu electrònic a cada una. Si la llista d'usuaris és remota, s'envia el correu electrònic al sistema remot i serà l'MTA remot el que l'expandirà i enviarà un correu electrònic a cada membre de la llista. Fixeu-vos que si la llista conté usuaris d'altres dominis de correu (on sigui del món), farà arribar una còpia a aquests usuaris.

El model funcional del protocol SMTP, que mostra els elements que intervenen en una comunicació d'aquest tipus, es pot veure a figura 1.1.

FIGURA 1.1. Model funcional del protocol SMTP



Exemple de funcionament del correu electrònic

El correu web té un funcionament similar al correu electrònic.

Per exemple, un usuari de Gmail utilitza com a MUA el web de Gmail per enviar un missatge a un usuari de Yahoo. Gmail transfereix el missatge per SMTP al servidor de correu de Yahoo i el missatge es diposita a la bústia del destinatari. Aquest, quan li sembla, consulta el correu utilitzant el lloc web de Yahoo com a MUA.

Exemples de programes que implementen SMTP: Sendmail, Exim, Postfix, MS Exchange Server.

Per tant, vist en conjunt, un MUA (Thunderbird, per exemple) permet a l'usuari crear un correu electrònic i lliurar-lo a l'MTA del sistema (per exemple, Sendmail) perquè el faci arribar al destinatari final. Usant el protocol SMTP, l'MTA s'encarrega de fer el lliurament al sistema final (pot ser amb una connexió directa o encaminant-se a través de diversos MTA) i el missatge es diposita en la bústia de correu del receptor. En aquest procés de deixar el missatge en la bústia del receptor hi pot haver un MDA que "postprocessi" el missatge o ho pot fer l'MTA directament. Quan ho considera oportú, el receptor recupera els missatges de la bústia utilitzant un MUA i un mecanisme d'accés adequat (per exemple, amb Thunderbird, usant el protocol POP o IMAP).

1.1.1 Format dels missatges

El protocol SMTP s'encarrega del transport de missatges de correu amb independència del format i del contingut. Els missatges es componen de diferents elements que es descriuen en l'especificació SMTP (corresponent al document RFC 821) i en l'especificació pròpia dels missatges d'internet (document RFC 822).

El missatge es desglossa en els elements següents:

Els camps FROM i RCPT del sobre no porten dos punts (:) mentre que les capçaleres *From* i *To* del contingut sí que en porten.

Alguns exemples de capçaleres: *From:*, *To:*, *Date:*, *Subject:*

- **Sobre** o *envelope*. Com passa en el correu postal, per fer arribar un missatge cal un sobre en el qual s'indiquin el destinatari i el remitent. L'especificació de l'SMTP (document RFC 821) descriu com a sobre el conjunt de dades necessàries per al transport del missatge (emissor, receptor, prioritat, nivell

de seguretat...). Generalment, el sobre consta tan sols dels camps FROM (emissor) i RCPT (receptor). L'MTA utilitza el sobre per encaminar el missatge. De fet, la separació entre sobre i contingut és confusa i l'MTA obté les dades del sobre a partir de les capçaleres del contingut.

- **Contingut**. El contingut d'un missatge és el que està descrit en el document RFC 822 *Standard for ARPA Internet Text Messages* (estàndard per als missatges de text d'internet). Tot contingut consta d'un conjunt de capçaleres (*headers*), una línia en blanc i un cos (*body*) del missatge:
  - **Capçaleres** (*headers*). El missatge conté capçaleres del tipus *clau: valor*, cada una en una línia independent.
  - **Línia en blanc**. Les capçaleres se separen del cos del missatge amb una línia en blanc.
  - **Cos del missatge**. El cos del missatge conté el missatge que es vol fer arribar al receptor. L'especificació inicial només permet text ASCII de 7 bits (sense símbols internacionals). El cos del missatge acaba amb una línia que conté a l'inici únicament un punt.

Les capçaleres descrites en l'especificació inicial tenen com a objectiu descriure clarament l'emissor i el receptor o receptors del missatge i la data, i permetre identificar el missatge (ID únic), entre d'altres.

En l'exemple següent podeu veure els elements que formen el contingut i les **capçaleres usuals** d'un correu electrònic:

```
Received: by 10.100.195.12 with HTTP; Sun, 11 May 2008 10:11:38 -0700 (PDT)
Message-ID: <7b4e8fcc0805111011g83da6b0rbdb4f63409024720@mail.gmail.com>
Date: Sun, 11 May 2008 19:11:38 +0200
From: "Pere Puig" <puig@correu.fp-oberta.org>
To: ppuig@correu.fp-oberta.org
Subject: =?ISO-8859-1?Q?Exemple_de_missatge_de_correu_amb_capçaleres
Delivered-To:ppuig@correu.fp-oberta.org

Hola,
Això és un exemple de missatge de correu.
Conté les capçaleres usuals.
S'ha generat des del web de Gmail i s'envia també a Gmail.

Pere
```

Aquestes són algunes de les capçaleres estàndard:

- ***From***: indica l'adreça de correu de l'emissor del missatge.
- ***Sender***: indica l'adreça de qui ha enviat el missatge. No s'utilitza si qui ha enviat el missatge és l'emissor del missatge. Serveix per diferenciar entre qui envia el missatge físicament i en nom de qui ho fa.
- ***To*** i ***Cc***: serveixen per indicar els destinataris del missatge. La idea original era posar un destinatari en el *To* i la resta en el *Cc*, però amb la utilització dels MUA actuals i la utilització de llistes d'usuaris generalment es posen tots els destinataris en el To.

- ***Bcc***: prové de l'anglès *blind carbon copy* o còpia oculta. S'indiquen els destinataris que han de rebre el missatge però que no han d'aparèixer a la llista de destinataris. Serveix per evitar que els altres destinataris sàpiguen qui n'ha rebut una còpia.

- ***Reply-to***: indica l'adreça de retorn del missatge al remitent. L'emissor pot voler que si el missatge es retorna o es respon, l'adreça a la qual es dirigeix la resposta sigui diferent de la indicada en el camp *From*. És útil per concentrar les respostes en un compte de correu quan l'emissor en té més d'un.

- ***Received***: cada MTA que processa un missatge afegeix una entrada de tipus *Received* en el missatge. És una manera de realitzar el seguiment o la traçabilitat dels MTA pels quals ha passat el missatge. La informació afegida descriu l'emissor (*From*) i el receptor (*By*), el mecanisme físic (*Via*), l'identificador del missatge (ID) i la data i hora (*Date*).

- ***Date***: indica la data i hora en què s'ha generat el missatge. L'hi afegeix el primer MTA que rep el missatge del MUA.

- ***Message-ID***: és l'identificador únic del missatge. Cada missatge s'ha de poder referenciar de manera única a tot el món. Això permet que les respostes indiquin a quin missatge es refereixen. S'utilitzen els noms de domini i un identificador numèric únic que genera l'MTA que rep el missatge per enviar.

- ***Subject***: descriu el propòsit del missatge o assumpte. És un petit text explicatiu.

- ***In-reply-to***: quan un missatge és una resposta a un missatge anterior, aquest camp indica a quin missatge original fa referència.

- ***Keywords***: és la llista separada per comes de paraules clau descriptives del missatge.

- ***Comments***: és el text de comentari del missatge que no interfereix en el contingut.

- ***References***: quan un missatge fa referència a altres missatges anteriors, es pot indicar mitjançant aquesta capçalera.

- ***Encrypted***: indica el tipus d'encriptació que s'ha utilitzat per al missatge. L'especificació del format dels missatges de correu (descrita en el document RFC 822) no indica cap tipus d'encriptació, simplement reserva una capçalera per indicar-ne el tipus.

- ***Return-path***: identifica el camí de retorn cap a l'origen. Aquesta informació l'ha de posar l'MTA receptor. Actualment està en desús, de manera que normalment conté l'adreça de l'emissor.

- ***X-<userDefined>***: els usuaris poden crear les pròpies capçaleres amb el nom que vulguin però començant per *X-*. D'aquesta manera s'assegura que si apareixen noves capçaleres oficials en el futur, no xocaran amb capçaleres definides pels usuaris.

1.1.2 Bústies de correu

Les **bústies de correu** són el sistema que permet l'emmagatzematge dels correus electrònics. Estan ubicades en l'espai de disc del servidor que allotja el servei de correu electrònic. Els dos principals formats són el mbox i el Maildir.

El format tradicional d'UNIX per a les bústies de correu és l'**mbox**. Les bústies dels usuaris s'emmagatzemen normalment a la carpeta */var/mail* o */var/spool/mail* (a vegades una és un enllaç simbòlic de l'altra). En aquesta carpeta hi ha un sol fitxer per usuari que conté tots els seus correus, concatenats un darrere l'altre. Aquest fet fa que sigui molt ràpid fer una cerca en la correspondència d'un usuari, tot i que aquest sistema no és gens escalable. Un dels grans problemes són els bloquejos, ja que per afegir un nou correu cal bloquejar el fitxer i això el fa inaccessible per a la cerca. El RFC 4155 dona informació sobre aquest tipus de bústia, però en cap cas es tracta d'unes especificacions.

La principal innovació del format **Maildir** és que té un fitxer per a cada correu i estan estructurats en carpetes, i això fa que no es produeixin bloquejos (només a nivell d'un sol correu). Els tres principals directoris són:

- *new*: és la carpeta on van a parar els correus nous. Un cop llegits passen a la carpeta cur.
- *cur*: és on es troben els correus que ja no són nous.
- *tmp*: és una carpeta temporal que, entre d'altres coses, serveix per rebre correctament els missatges abans de ser moguts a la carpeta *new*.

Aquest sistema és més estable, ràpid i escalable que el tradicional mbox. I el problemes de corrupció de fitxer afecten notablement menys a aquest sistema ja que tots els correus estan separats.

No obstant, també hi ha altres formats de bústia, més minoritaris. Aquests són:

- **dbox**: format de bústia d'alt rendiment per a Dovecot. Té dues variants:
  - **sdbox** (*single dbox*): semblant al Maildir, un missatge per correu.
  - **mdbox** (*multidbox*): múltiples correus per fitxers, però no a l'estil de mbox.
- **mbx/mix**: format de bústia del programari UW-IMAP de la Universitat de Washington. L'anterior format era el mbx, que s'ha substituït pel mix, que permet un millor rendiment.
- **Mailstore**: format de bústia originari del programari Exim. Consta de dos fitxers per correu amb les extensions .env i .msg, un per al sobre (*envelope*) i l'altre per al missatge.
- **Pst** (*Personal Storage Table*): format obert propietari de Microsoft. És utilitzat per Microsoft Exchange Server i Microsoft Outlook.

1.1.3 Funcionament de l'SMTP

'Push'

Es diu que l'SMTP és un protocol que fa *push* (lliura), però no *pull* (agafa). Els usuaris finals han d'usar altres mecanismes per accedir remotament als seus comptes de correu.

El funcionament del protocol SMTP imita el correu postal en molts aspectes. L'SMTP és un protocol d'emmagatzemament i enviament que funciona igual que es fa amb les cartes de correu, que es lliuren en una oficina postal, d'allà a una altra, i així successivament fins arribar al destinatari final. De fet, les cartes es lliuren a la bústia del destinatari final i aquest les ha de recollir.

El **servidor SMTP** és una aplicació distribuïda que permet enviar missatges electrònics. Utilitza el protocol de transport TCP i el port 25.

MX

En la base de dades d'un servidor DNS, els equips que fan de servidors de correu d'un domini s'identifiquen per les entrades tipus MX. Si un domini no disposa d'entrades MX, s'utilitza l'amfitrió que defineix el domini.

En l'esquema original en què es va desenvolupar l'SMTP, una organització disposa d'un servidor SMTP (un MTA) que rep correu electrònic de fora de l'organització i el diposita en les bústies de correu locals del servidor. També recull el correu intern de l'organització i l'envia fora.

Cada organització disposa d'una o més màquines encarregades de gestionar el correu. Així, quan s'envia un correu electrònic a l'usuari pere@ioc.cat, cal que l'organització o domini ioc.cat disposi de màquines que fan la funció de servidors de correu. ¿Com trobarà l'SMTP a quin servidor de correu ha de lliurar els correus electrònics destinats a un domini? Utilitzant el protocol DNS (*domain name system*, sistema de noms de domini) i fent una consulta de tipus MX obtindrà la màquina o màquines que fan la funció de servidors de correu del domini consultat.

El client SMTP o emissor estableix una connexió TCP amb el port 25 del servidor SMTP o receptor. En una mateixa connexió l'emissor pot enviar un o més missatges al receptor. Si el mateix missatge va destinat a diversos receptors del sistema final, el missatge s'envia un sol cop i l'MTA receptor el replica a cada destinatari.

El client SMTP o emissor disposa d'una cua de missatges per enviar i una llista de destinataris per a cada missatge. Els destinataris poden ser en destinacions diferents (evidentment) i, per tant, li caldrà connectar-se als diferents servidors de destinació per fer-los arribar els missatges.

Llistes negres de servidors de correu

Els servidors de correu que accepten correus electrònics de tots els clients a totes les destinacions són inclosos en llistes negres perquè poden ser generadors de correu brossa.

Correu brossa o 'spam'

Correu brossa, correu no desitjat o no sol·licitat. És un correu que es rep insistentment i que bombardeja les bústies dels usuaris de manera mecànica.

Quan un destinatari no és accessible, el missatge es pot tornar a posar a la cua de missatges pendents d'enviar o es pot descartar (segurament després de diversos intents infructuosos) tot notificant-ne l'emissor.

Avui en dia el servidor de correu pot ser a qualsevol lloc del món i no cal que cada organització en tingui un. Es pot utilitzar el del proveïdor ISP o el de qualsevol servei extern de correu (per exemple, Google permet externalitzar el correu a empreses tot mantenint el domini propi de l'empresa). Això significa que el servidor SMTP ha de verificar si accepta o no peticions d'enviar correu d'un client. Es pot verificar el client mitjançant l'adreça IP o mitjançant altres mecanismes d'autenticació i seguretat. Evidentment, disposar d'un servidor SMTP que accepta peticions de clients sense verificar qui són és una porta oberta a permetre correu

brossa. Normalment els servidors SMTP restringeixen qui pot fer ús del servei (quins clients) i a quines destinacions.

Un cop un servidor SMTP accepta un correu electrònic per fer-ne el lliurament (d'un MUA com Thunderbird, per exemple) tot validant que accepta rebre correus electrònics d'aquest client, estableix una connexió TCP al port 25 del servidor SMTP destinatari (ha obtingut l'adreça IP fent la resolució DNS de la part del domini de l'adreça de correu).

Ordres/respostes SMTP

L'emissor sempre porta el control de la comunicació i inicia la connexió amb el receptor. El diàleg consisteix en un intercanvi d'ordres i respostes que segueixen les especificacions de Telnet:

S'entén per CRLF una línia en blanc. Ve de l'anglès *Carriage Return - Line Feed* (retorn de carro - salt de línia).

- **Ordres**. Són codis de quatre caràcters (HELO, MAIL, DATA...) i arguments opcionals separats per espais i acabats amb <CRLF>. Per a cada ordre es rep una resposta del receptor.
- **Respostes**. Són codis numèrics de tres dígits, un espai i un missatge descriptiu que pot variar segons la implementació.

Un diàleg bàsic entre emissor i receptor SMTP podria ser el següent:

- **HELO <domini> / EHLO <domini>**. Un cop connectat, l'emissor s'ha d'identificar amb l'ordre HELO i indicar el domini al qual es connecta. Actualment, els servidors SMTP utilitzen extensions i l'ordre preferida per identificar-se és EHLO (significa extended HELO).
- **MAIL FROM: <emissor>**. Identifica l'emissor del missatge i genera la capçalera *From* del missatge. El receptor comprova que l'emissor sigui un usuari vàlid, és a dir, que accepti missatges d'aquest origen. Si no el pot validar, envia una resposta denegant-li la comunicació. Els equips amb el *relay* configurat per permetre enviar missatges de tothom són els principals generadors de correu brossa.
- **RCPT TO: <destinatari>**. Indica el destinatari del missatge. Aquesta ordre es pot repetir tantes vegades com destinataris tingui el missatge. També cal que el receptor accepti el destinatari, que pot ser un destinatari local, o que accepti fer el reenviament si és un destinatari remot. Aquesta ordre genera la capçalera *To* en el missatge.
- **DATA**. Indica que a continuació s'enviarà el missatge. Tot el que es transmet a continuació és el contingut del missatge, que finalitzarà en trobar una línia que només inclou un punt (.<CRLF>). El contingut segueix les especificacions del document RFC 822; per tant, pot contenir capçaleres a l'inici, una línia en blanc a manera de separador i el cos. No es pot enviar un missatge (l'ordre DATA) fins que el receptor no ha confirmat que accepta almenys un destinatari. Això evita transmetre missatges que es descartarien en la destinació.

- **QUIT**. L'emissor envia l'ordre per indicar al receptor que vol finalitzar la comunicació. El receptor confirma la recepció i llavors tots dos poden finalitzar la transmissió.

En la secció "Annexos" del web d'aquest mòdul teniu captures dels diferents diàlegs SMTP, POP i IMAP.

En l'exemple següent podeu veure un diàleg client/servidor SMTP mitjançant ordres i respostes Telnet:

```
[root@host ~]# telnet www.escola.org 25
Trying 22.170.21.168...
Connected to www.escola.org.
Escape character is ,^]'.
220 escola.org ESMTP Sendmail 8.13.8/8.13.8; Sat, 26 Apr 2008
19:56:05 +0200
EHLO escola.org
250-escola.org Hello 106.Red-71-92-14.dynamicIP.rima-tde.net
71.92.14.106], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE 10000000
250-DSN
250-ETRN
250-AUTH DIGEST-MD5 CRAM-MD5 LOGIN PLAIN
250-DELIVERBY
250 HELP

MAIL FROM: pere@xtec.cat
250 2.1.0 pere@xtec.cat... Sender ok
RCPT TO: pere@correu.escola.org
250 2.1.5 pere@correu.escola.org... Recipient ok

DATA
354 Enter mail, end with "." on a line by itself
Hola,
Aquest és un missatge de prova per enviar un
correu usant Telnet al servidor SMTP de l'escola.
S'envia una còpia a dos usuaris locals al servidor.
S'ha denegat fer relaying i enviar una còpia a
l'exterior.
Pere
.
250 2.0.0 m3QHu5B3012660 Message accepted for delivery

QUIT
221 2.0.0 escola.org closing connection
Connection closed by foreign host.
```

Ordres SMTP

Per obtenir la llista d'ordres del protocol podeu consultar el document RFC 2821. Atès que és un servidor concret, podeu consultar les ordres que implementa amb l'ordre HELP.

Amb els camps MAIL FROM i RCPT TO, el protocol SMTP obté les dades necessàries per generar el sobre o *envelope*.

A part de les ordres bàsiques mostrades anteriorment, hi ha altres ordres en el protocol SMTP, com ara les següents:

- **RSET**. L'emissor pot interrompre l'enviament de missatges.
- **NOOP**. Aquesta ordre no fa res ('no operate'), però força el receptor a enviar una resposta afirmativa. Serveix per confirmar que la connexió encara és oberta.
- **HELP**. Fa una llista de les ordres que implementa el servidor. Els servidors SMTP no implementen necessàriament totes les ordres descrites pel protocol.

- **VRFY <destinatari>**. L'emissor pot verificar l'existència del destinatari.
- **EXPN <destinatari>**. Permet a l'emissor verificar l'existència d'una llista de correu i obtenir-ne els noms dels membres.
- **SEND, SOML, SAML**. Permeten enviar els missatges tant a les bústies de correu com als terminals.
- **TURN**. Permet intercanviar els papers entre emissor i receptor. El receptor hi ha d'estar d'acord.

Els servidors SMTP no implementen necessàriament totes les ordres, però hi ha un conjunt d'ordres mínim definit pel protocol que tot servidor SMTP ha d'implementar.

El **conjunt d'ordres mínim** que tot servidor SMTP ha d'implementar és el següent:
HELO <domini>
MAIL FROM: <emissor>
RCPT TO: <destinatari>
DATA
RSET
NOOP QUIT

El protocol SMTP permet treballar amb missatges ASCII de 8 bits i amb extensions del protocol, és a dir, afegir als servidors SMTP funcionalitats extres segons el programari de servidor utilitzat. El client pot sol·licitar al receptor la llista de les extensions que implementa i fer-li saber que les vol utilitzar. El mecanisme consisteix a fer que el client enviï un **EHLO** en lloc del HELO estàndard. Si el receptor implementa extensions, respondrà afirmativament i en farà una llista; si no les implementa, respondrà negativament. Llavors l'emissor pot fer un HELO estàndard.

Les respostes es poden classificar en quatre grans grups. El primer dígit del codi numèric de tres dígits de la resposta indica el grup al qual pertany:

- **Positiva (2xx)**. L'acció que ha sol·licitat l'emissor és acceptada pel receptor. L'emissor pot fer una nova sol·licitud. Les respostes d'aquest grup comencen totes pel dígit 2. En els llistats de codi es pot observar que, per exemple, el valor 250 correspon a OK o acció realitzada correctament.
- **Intermèdia positiva (3xx)**. L'acció sol·licitada s'ha acceptat, però està suspesa pendent de rebre informació addicional que l'emissor haurà de proporcionar.
- **Negativa transitòria (4xx)**. La sol·licitud no s'ha acceptat i l'acció no s'ha realitzat, però es tracta d'un error temporal i es pot tornar a intentar més tard. L'emissor pot tornar a fer la sol·licitud més endavant.
- **Negativa pertinent (5xx)**. L'ordre no s'ha realitzat i, per tant, la sol·licitud no ha estat acceptada.

1.1.4 MIME

Els missatges de correu tenen el format definit en l'RFC 822 (actualment, RFC 2822), que únicament permet missatges de text net ASCII de 7 bits. No es permeten els caràcters accentuats, caràcters internacionals (ASCII de 8 bits) i molt menys la transferència de dades binàries com imatges, àudio, aplicacions, PDF o altres. Però tot això i molt més s'envia avui en dia per correu electrònic.

El juny del 1992 es va definir el que es coneix com a **MIME** (*Multipurpose Internet Mail Extension* o **extensió de correu d'internet per a ús múltiple**) en l'RFC 1341, que actualment ha evolucionat en els RFC 2045 i RFC 2049. El MIME utilitza missatges RFC 822, però afegint una estructura al cos del missatge i regles de codificació per a missatges no ASCII. El gran avantatge del MIME és que permet seguir utilitzant les mateixes eines de l'SMTP que fins ara i només cal modificar els MUA perquè apliquin MIME. A l'MTA, el cos del missatge li és absolutament indiferent (per tant, pot estar codificat), ja que només utilitza el sobre per enviar el missatge i el contingut s'envia com un tot.

El MIME es basa en tres elements per permetre qualsevol tipus de contingut en un missatge de correu:

- **Capçaleres MIME**. Es creen cinc noves capçaleres de correu per definir informació del cos del missatge. No totes són obligatòries.
- **Formats de contingut**. Es defineixen diferents formats de contingut que permeten als MUA receptors interpretar el contingut de manera adequada i saber si reben un full de càlcul, un vídeo...
- **Esquemes de codificació de transferència**. Es realitza una transformació de les dades a un format manipulable per al transport SMTP (que només permet caràcters ASCII de 7 bits).

Podeu veure els components d'un missatge amb contingut MIME en l'exemple següent:

```
From root@tftp.server.cat  Fri Jun 13 17:26:31 2012
Return-Path: <root@tftp.server.cat>
Received: from tftp.server.cat (localhost [127.0.0.1])
 by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DFQTH7003922
 for <pere@tftp.server.cat>; Fri, 13 Jun 2012 17:26:30 +0200
Received: (from root@localhost)
  by tftp.server.cat (8.14.1/8.14.1/Submit) id m5DFQSIq003918
  for pere@tftp.server.cat; Fri, 13 Jun 2012 17:26:28 +0200
Date: Fri, 13 Jun 2012 17:26:27 +0200
From: root <root@tftp.server.cat>
To: pere@tftp.server.cat
Subject: missatge amb atchment
Message-ID: <20080613152627.GA3909@portatil.local.lan>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="zYM0uCDKw75PZbzx"
Content-Disposition: inline
User-Agent: Mutt/1.5.17 (2007-11-01)
Status: O

```

```
20 --zYM0uCDKw75PZbzx
21 Content-Type: text/plain; charset=us-ascii
22 Content-Disposition: inline
23
24 missatge de root a l'usuari pere
25 conté adjunt un pdf i jpeg
26 adéu!
27
28 --zYM0uCDKw75PZbzx
29 Content-Type: application/pdf
30 Content-Disposition: attachment;
31 filename="informatica_AX_ud2.pdf"
32 Content-Transfer-Encoding: base64
33 ... output suprimit (contingut del pdf codificat en base64) ...
34 --zYM0uCDKw75PZbzx
35 Content-Type: image/jpeg
36 Content-Disposition: attachment; filename="cd15_11_puerto-
37 madrin.jpg"
38 Content-Transfer-Encoding: base64
39 ... output suprimit (contingut del jpeg codificat en base64) ...
40 --zYM0uCDKw75PZbzx--
```

Capçaleres MIME

Les cinc capçaleres que defineix l'especificació MIME aporten informació del contingut del missatge.

Aquestes capçaleres són les següents:

- ***MIME-version***. Identifica el tipus MIME del missatge. Si indica 1.0, es tracta d'un missatge MIME; si no, es tracta d'un missatge ASCII.
- ***Content-description***. És un text que descriu el tipus de contingut. No és obligatori i no té cap funcionalitat més enllà de la merament descriptiva.
- ***Content-ID***. Identifica el contingut de manera única, igual que ho fa el camp *Message-ID*.
- ***Content-transfer-encoding***. És el mecanisme de codificació utilitzat en el missatge per poder-lo transmetre. El contingut que no és ASCII de 7 bits es codifica per poder ser transmès.
- ***Content-type***. Descriu el tipus de contingut segons la taula de tipus MIME. Això permet a un MUA obrir l'aplicació pertinent per gestionar el contingut. Si, per exemple, el tipus és *image/jpeg*, permet al MUA saber que en pot manipular el contingut amb una aplicació de gestió d'imatges.

Tipus MIME

Es defineix un conjunt de tipus i subtipus MIME amb un esquema tipus/subtipus. Originàriament es van definir els set tipus que es descriuen a continuació, però en l'actualitat n'hi ha molts més.

- **text/native**. Text net en format ASCII de 7 bits.

- **multipart/<subtipus>**. El missatge conté múltiples parts independents. Un delimitador (o *boundary*) indica la separació de cada part. El delimitador és únic i no apareix en el cos de les parts. El delimitador es troba a l'inici i al final de cada part i comença amb dos guions. L'última part acaba amb un delimitador que comença i acaba amb dos guions. Cada part pot ser qualsevol cosa!
- **multipart/parallel**. Múltiples parts, en ordre. És a dir, les parts s'han de mostrar en el receptor en l'ordre indicat.
- **multipart/mixed**. Múltiples parts. No es defineix cap ordre.
- **multipart/alternative**. Les parts són versions alternatives del mateix contingut en ordre creixent de fidelitat. El receptor escull la més apropiada. Per exemple, un text s'envia com a text pla en una primera part i com a PDF en una segona; si el receptor no disposa de PDF podrà usar la part en text net.
- **multipart/digest**. Cada part és un missatge de correu individual. S'utilitza quan un correu electrònic conté diversos correus electrònics en el seu interior (per exemple, reenviaments).
- **message/rfc822**. El cos és un missatge de correu complet, amb capçaleres i cos. Pot ser un missatge MIME tot i que al nom hi digui "rfc822".
- **message/partial**. Permet fragmentar un missatge llarg en diferents missatges. Cada fragment ha de disposar d'un identificador, número de fragment i nombre total de fragments.
- **message/external body**. Les dades del cos del missatge no estan en el missatge sinó que cal baixar-les a part. En la capçalera Content-type es descriu el tipus de contingut i el tipus d'accés, que pot ser FTP, TFTP, anon-FTP (FTP anònim), *local-file*, AFI i *mail-server*. Per exemple, el contingut pot ser una imatge no inclosa en el missatge sinó que calgui baixar d'un servidor FTP.
- **image/jpeg**. Imatge codificada JPEG
- **image/gif**. Imatge GIF
- **video/mpeg**. Vídeo en format MPEG (*Moving Picture Experts Group*, grup d'experts d'imatges en moviment)
- **audio/basic**. Àudio en format estàndard
- **application/postscript**. Dades binàries en format PostScript. Per exemple, PDF.
- **application/octet-stream.** Dades binàries

Codificació de transferència

Les dades binàries i els caràcters internacionals (que no pertanyen al conjunt ASCII de 7 bits) no es poden enviar per correu electrònic. Per poder-ho fer, cal codificar-los en un altre format.

L'especificació MIME defineix els tipus de codificacions següents:

- **7bit**. Indica que les dades es transfereixen en ASCII de 7 bits. No es realitza cap codificació.
- **8bit**. No es realitza cap codificació i les dades es transmeten en ASCII de 8 bits. Evidentment, cal que receptor i emissor permetin la transferència a 8 bits (una extensió d'SMTP).
- **Binary**. Es transmeten les dades en binari tal com són, sense cap codificació ni control de la longitud de les línies. Si s'envien dades en binari (en cru), no es garanteix que la transmissió sigui correcta.
- **X-token**. Indica la utilització d'un esquema de codificació de transport no estàndard, un esquema propi. Emissor i receptor han de compartir aquest esquema de codificació.
- **Quoted-printable**. Quan la majoria de caràcters del missatge són imprimibles excepte una petita part, és més eficient utilitzar aquesta codificació que Base64. Aquest esquema codifica els caràcters no imprimibles amb un signe igual (=) i el codi hexadecimal del caràcter. Es garanteix que les línies tenen una longitud no superior a 36 caràcters mitjançant salts de línia reversibles.
- **Base64**. És l'esquema de codificació més usat per a la transferència d'informació binària. Converteix l'entrada en un conjunt de caràcters imprimibles i, per tant, immunes al transport per SMTP. Consta d'un conjunt de 63 caràcters imprimibles i un més de farciment (2 ^ 6 = 64 caràcters). Cada 24 bits de l'entrada binària (3 *bytes*) es codifica en quatre blocs de 6 bits (4 * 6 = 24 bits). A cada bloc de 6 bits li correspon un caràcter imprimible que es posa en 1 *byte*. Per tant, per cada 24 bits d'entrada binària, s'utilitzen 32 bits de transmissió (4 * 1 *byte*).

Base64

Per aprendre el funcionament de Base64 podeu consultar la Viquipèdia:

en.wikipedia.org/wiki/Base64

Exemple de codificació en Base64

Aquest és un petit exemple extret de la Viquipèdia on s'observa que el text "Man" original (3 *bytes* = 24 bits) acaba codificat en Base64 com a "TWFu" (4 *bytes*).

Text content M a n

ASCII 77 97 110

Bit pattern 01001101 01100001 01101110 (8 bits * *byte*)

Bit pattern 010011 010110 000101 101110 (divisió en blocs de 6 bits)

Index 19 22 5 46

Base64-encoded T W F u

1.2 Instal·lació d'un servidor

Per obtenir més informació sobre els servidors de correu actuals, consulteu l'activitat titulada: Quota de mercat dels servidors de correu.

Sendmail

Per conèixer els orígens i l'evolució de Sendmail us recomanem consultar l'article sobre el servei a la Viquipèdia:

en.wikipedia.org/wiki/Sendmail

Hi ha diverses aplicacions de servidor de correu en el mercat i moltíssims clients de correu de tota mena, tant en versió gràfica com d'entorn de text. Algunes d'aquestes aplicacions són de font pública i es poden baixar gratuïtament d'internet.

La majoria de sistemes GNU/Linux proporcionen l'aplicació client Mail i sovint també Mutt, que és una versió de Mail amb pantalles en mode text. Els sistemes GNU/Linux i Unix també disposen d'una aplicació servidor omnipresent anomenada Sendmail.

Quan es parla d'instal·lar el servei de correu es fa referència al procés d'instal·lació i configuració del programari del servidor. Això es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS, HTTP o FTP): es tracta d'instal·lar els paquets o *tarballs* de l'aplicació servidor i fer-ne la configuració apropiada.

Per fer això cal plantejar-se els passos i reflexions següents:

- Preguntar-se i buscar l'aplicació més adient: *Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?*
- Observar l'estat de la xarxa actual: *El servei ja està en funcionament? Existeix ja un servidor de correu instal·lat i actiu?*
- Obtenir l'aplicació que proporciona el servei de correu.
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha fet correctament.
- Configurar el servei en el servidor i comprovar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

L'eina utilitzada en aquest mòdul per estudiar els serveis de servidor de correu és **Sendmail**. Podeu trobar tota la informació sobre aquest servidor a **www.sendmail.org**.

Usualment, l'administrador acaba utilitzant l'aplicació servidor que li proporciona el seu sistema operatiu. Si utilitzeu Windows, l'empresa Microsoft disposa d'una aplicació pròpia, però en podeu trobar d'altres a internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució proporciona un servidor de correu o bé n'existeix algun de clàssic provinent d'Unix. De totes maneres, en podeu obtenir d'altres a internet.

1.2.1 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per internet paquets de servidor de correu Sendmail usant eines com yum o apt-get i els repositoris de paquets

apropiats segons la distribució que utilitzin. A més, sempre es pot recórrer a Google per localitzar tot allò que faci falta.

Un cop instal·lat el programari, cal identificar què s'ha instal·lat, els paquets i el contingut. A vegades no s'instal·len paquets sinó fitxers *tarball*, el contingut dels quals també cal saber examinar. És important identificar els components instal·lats que corresponen a fitxers executables, els que corresponen a fitxers de configuració i els que corresponen a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i s'ha posar en marxa. Per tant, cal saber gestionar l'estat del servei (engegar, aturar, recarregar...) i definir l'estat que ha de tenir en els diferents *runlevels* del sistema.

En definitiva, el procediment d'instal·lació inclou usualment:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo amb l'eina apropiada segons la distribució que s'utilitzi.
- Examinar el sistema per identificar el programari, els paquets, instal·lat relacionats amb el servei.
- Identificar els components del servei: fitxers executables, fitxers de configuració i fitxers de documentació.
- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

Verificació de l'accés als comptes de correu

Un servidor de correu realitza la funcionalitat de client i de servidor SMTP. Com a client s'encarrega d'enviar els missatges que hi ha a la cua de missatges al servidor apropiat. És a dir, si un missatge està dirigit a un usuari del domini gmail.com, s'encarrega de fer arribar el missatge a algun dels servidors de correu d'aquest domini.

Com a servidor SMTP té la funció d'escoltar les peticions entrants que li fan els clients SMTP i atendre-les. Això significa "rebre" els missatges i fer els passos necessaris per fer-los arribar a la bústia del client. Si el sistema de correu no utilitza un MDA propi (ho són programes com procmail o SpamAssassin), el mateix Sendmail farà aquesta funció.

> Quan actua com a **servidor SMTP** el servidor de correu també pot realitzar la funcionalitat de MDA (*Mail Delivery Agent*) i encarregar-se de deixar els missatges a la bústia de cada usuari local.

De fet, si la configuració de correu actual utilitzada és la que es crea per defecte en instal·lar Sendmail, no hi ha cap MDA específic sinó que és el mateix Sendmail el que fa aquesta funció. Això significa que també s'ha d'encarregar de crear les bústies de correu dels usuaris i de gestionar-les (si no és que ja ho fa el mateix sistema operatiu).

Si no hi ha altres agents tipus MDA en funcionament, la creació i la gestió de les **bústies d'usuari** és missió del servidor de correu.
Un **compte de correu** no és altra cosa que disposar d'una bústia de correu en el sistema.

Usuaris locals

Si es vol que un usuari local no pugui iniciar una sessió d'usuari en el sistema, se li pot assignar com a *shell* /sbin/nologin.

Tot usuari de sistemes GNULinux disposa d'un compte local en la màquina on té el compte d'usuari. Així, un usuari de nom pere en un amfitrió anomenat pc-jocs/ pot rebre correu a l'adreça pere@pc-jocs.

És evident que cada usuari que vol tenir un compte de correu ha de disposar d'una **bústia** pròpia en la qual el servidor ha de poder desar el correu destinat a l'usuari. L'usuari accedeix a la seva bústia per consultar el seu correu.

Des del punt de vista de la creació de bústies es fa la classificació següent:

- **Usuaris locals del sistema**. En sistemes GNU/Linux tots els usuaris del sistema disposen d'una bústia de correu local. Què cal fer perquè els usuaris d'un servidor tinguin correu local? Res. Tots els usuaris locals d'un *host* tenen una bústia pròpia i tothom s'hi pot adreçar indicant **usuari@host**. Aquest mecanisme obliga a generar comptes d'usuaris locals en el sistema per tal de poder disposar dels comptes de correu.
- **Usuaris del servei**. Hi ha servidors de correu que permeten crear comptes de correu sense necessitat de crear comptes d'usuari locals en el sistema. És a dir, es tracta d'usuaris que existeixen només per al servidor de correu però no per al sistema operatiu.

1.2.2 Usos indeguts del servidor de correu

El problema principal del correu electrònic és el correu brossa o *spam*, és a dir, el correu no desitjat. Des del punt de vista del client, convé saber filtrar el correu per detectar el correu brossa i informar-ne al servidor (generalment és un *webmail*). Des del punt de vista del servidor, cal saber filtrar el correu brossa i cal establir mecanismes per no participar en la seva difusió.

Actualment la majoria de serveis de correu del tipus *webmail* incorporen les dues prestacions següents:

- **Filtrat automàtic de correu brossa**. Gmail, per exemple, filtra els correus i intenta detectar quins són brossa i els posa directament en una carpeta amb aquest nom. Gmail aplica regles complexes de filtrat per detectar correus brossa segons el seu criteri. Els usuaris el poden ajudar informant-lo dels missatges que consideren brossa.
- **Cerca automàtica de virus**. S'aplica un antivirus als continguts que s'adjunten als fitxers. D'aquesta manera s'evita la propagació indiscriminada de continguts maliciosos.

1.3 Accés remot al correu

La manera com els usuaris accedeixen al seu correu ha anat evolucionant al mateix pas que ho ha anat fent la tecnologia. Originalment s'utilitzava simplement l'SMTP (Sendmail, per exemple) i els usuaris havien d'accedir al servidor iniciant una sessió d'usuari per consultar el correu amb eines com Mail. És a dir, els usuaris havien d'anar físicament on hi havia la màquina servidor i iniciar-hi una sessió o bé connectar-s'hi via Telnet.

Però els usuaris volien poder descarregar i enviar des de casa els missatges de correu. El protocol POP d'accés remot a les bústies de correu va proporcionar aquest servei. Els usuaris es connectaven per mòdem, es connectaven al servidor de correu i es descarregaven tot el correu de cop i aprofitaven també per enviar missatges. En aquest model, la gestió dels missatges es feia a casa, el servidor simplement els acumulava per permetre'n la descàrrega. La tecnologia d'accés a internet per mòdem implicava pagar per les trucades. Per tant, l'usuari tenia interès en baixar tot el correu, finalitzar la trucada (per no seguir pagant) i examinar tranquil·lament els missatges sense connexió a internet.

Amb l'aparició de les tarifes planes, els usuaris ja no s'han de preocupar de fer una connexió curta al servidor i poden estar connectats permanentment. El protocol IMAP d'accés a bústies remotes permet l'accés dels clients a les seves bústies realitzant totes les gestions (carpetes, etiquetat, filtrat...) en el mateix servidor. Això resol un dels problemes típics del POP, que és que baixant els missatges en màquines diferents el correu quedava repartit per diferents llocs.

Sempre que es configura un client de correu cal indicar:

- Servidor de **correu entrant**: un servidor POP o IMAP des d'on es descarreguen els missatges de l'usuari.
- Servidor de **correu sortint**: el servidor SMTP a qui cal lliurar el correu que genera l'usuari per tal que sigui enviat al destinatari.

Actualment, la majoria de clients de correu utilitzen serveis *webmail* com Gmail, Yahoo o altres. Els clients es connecten al servidor i accedeixen a la seva bústia utilitzant una interfície web. Tota la gestió del correu es fa des d'un navegador.

1.3.1 Servei POP

POP3 és la versió actual del protocol POP. Aquí usem tots dos noms indistintament.

En el model de transport de correu SMTP s'exigeix que el receptor disposi de connexió permanent a internet. Està pensat per a correu entre organitzacions connectades a la xarxa i que disposen d'un servidor de correu que conté les bústies dels usuaris locals de l'organització. Això obliga els usuaris a treballar localment

en el servidor per accedir a les seves bústies. Amb la popularització d'internet sorgeix el problema dels usuaris que hi accedeixen per ISP i que no tenen connexió permanent (per exemple, amb mòdem).

POP3 i el correu postal

El POP3 és un mecanisme similar al correu postal. El carter deixa les cartes a la nostra bústia i les recollim quan ens sembla.

S'idea un mecanisme per a l'accés remot als comptes de correu, de manera que l'usuari es connecta quan vol, accedeix a la bústia de correu per recuperar els missatges i finalitza la connexió. POP3 i IMAP són protocols que permeten l'accés remot de clients a les bústies de correu.

POP3 (*Post Office Protocol* o **protocol d'accés simple a les bústies de correu**) és un protocol de capa d'aplicació de la pila de protocols TCP/IP (port 110) definit en l'RFC 1939. Permet a un client de correu (MUA) obtenir remotament el correu dipositat en la bústia de l'usuari en un servidor POP3.

Dispersió del correu

Si l'usuari baixa correu POP des de màquines diferents, li queda dispersat. En cada màquina queda desat localment el que s'hi ha baixat.

Normalment, l'usuari utilitza una aplicació client de POP3 (per exemple, Thunderbird) i baixa el correu del servidor POP. Els missatges que es baixen es desen a la màquina de l'usuari (localment) i s'esborren del servidor (es pot configurar si s'esborren o no). Finalment es tanca la connexió.

Funcionament del POP3

Amb el protocol POP3 el client fa una connexió TCP/IP al port 110 del servidor, baixa el correu i tanca la connexió. En aquest procés, client i servidor passen per tres estats (autorització, transacció i actualització) i s'intercanvien ordres i respostes seguint el model de diàleg de Telnet:

- **Ordres**. Són ordres de text de quatre caràcters seguides d'espais i els arguments que requereixin. Finalitzen amb un <CRLF>.
- **Respostes**. Són una cadena de caràcters que comença per **+OK** o **-ERR** més una descripció. Les respostes afirmatives comencen per +OK, i les d'error per -ERR.

Baixar missatges del servidor POP3

Alguns MUA bàsics baixen tots els missatges del servidor de cop. Si es deixen els missatges ja llegits en el servidor, es tornaran a baixar cada cop que s'hi accedeix.

Vegeu una llista de les ordres utilitzables en el protocol POP3 agrupades segons l'estat:

1. **Autorització**
   - **USER <nomUsuari>**. El client s'identifica en el servidor POP indicant el nom d'usuari, que ha de correspondre a una bústia de correu del servidor.
   - **PASS <password>**. El client s'ha d'autenticar indicant un nom d'usuari i una contrasenya vàlids. L'ordre PASS permet indicar aquesta contrasenya en text net.
   - **APOP <nomUsuari> password-md5**. Per proporcionar més seguretat en el procés d'autorització, l'usuari pot fer servir l'ordre APOP, que té com a arguments el nom d'usuari i la contrasenya encriptada usant una funció resum o *hash*, com per exemple *MD5*.

2. **Transacció**

   - **STAT**. Demana l'estat de la bústia. El servidor retorna el nombre de missatges que conté i el total de *bytes* que ocupa.
   - **LIST [msg]**. Llista els missatges o un missatge concret. No en llista el contingut sinó que llista el número de missatge i el nombre de *bytes* que ocupa cada missatge.
   - **RETR msg**. Baixa un missatge concret del servidor. Els missatges es poden indicar pel número de missatge.
   - **DELE msg**. Marca el missatge indicat per ser esborrat. No l'esborra immediatament, només el marca; l'esborra en l'estat final d'actualització. En els MUA que actuen de clients POP és típic permetre configurar si es deixen els missatges en el servidor o s'eliminen. Usualment s'eliminen perquè només hi quedin els nous.
   - **NOOP**. Aquesta ordre força el servidor a emetre una resposta positiva. Serveix per comprovar que la connexió encara és oberta.
   - **RSET**. Si s'executa aquesta ordre abans de passar a l'estat d'actualització, RSET desmarca tots els missatges que estaven marcats per esborrar.
   - **TOP msg nLin**. En lloc de baixar el missatge sencer com fa l'ordre RETR, l'ordre TOP permet baixar-ne les línies inicials. Baixa les capçaleres i les línies inicials (*nLin*). Aquesta ordre és útil per baixar només les capçaleres (remitents, assumptes...) i per filtrar els missatges a baixar i marcar-los per esborrar-los directament sense baixar-los.
   - **UIDL [msg]**. Els missatges s'identifiquen pel seu número d'ordre (com fa l'ordre LIST), però aquest número pot variar entre connexions si s'esborren els missatges que el precedeixen. Per identificar de manera única un missatge independentment de la posició que ocupa es pot usar el UID. El UID és únic per a cada missatge d'una bústia. L'ordre UIDL pot llistar els UID i els números d'ordre de tots els missatges o els d'un missatge concret.
   - **QUIT**. Indica al servidor que el client vol finalitzar la connexió. El servidor passa de l'estat de transacció al d'actualització.

3. **Actualització**

   - En aquest estat no hi ha ordres. El servidor elimina els missatges marcats per ser esborrats, emet una resposta positiva al client i tots dos tanquen la connexió.

L'exemple següent correspon a un diàleg mitjançant Telnet entre client i servidor usant el protocol POP. S'hi poden veure les ordres i respostes del protocol:

```
[root@portatil ~]# telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is ,^]'.
+OK POP3 localhost 2007a.104 server ready
```

```
USER pere
+OK User name accepted, password please
PASS pere
+OK Mailbox open, 1 messages

STAT
+OK 1 480
NOOP
+OK No-op to you too!
LIST
+OK Mailbox scan listing follows
1 480
.
RETR 1
+OK 480 octets
Return-Path: <root@localhost.localdomain>
Received: from tftp.server.cat (localhost [127.0.0.1])
 by tftp.server.cat (8.14.1/8.14.1) with ESMTP id m5DI4ig8005681
 for pere@tftp.server.cat; Fri, 13 Jun 2008 20:06:12 +0200
Date: Fri, 13 Jun 2008 20:04:44 +0200
From: root <root@localhost.localdomain>
Message-Id: <200806131806.m5DI4ig8005681@tftp.server.cat>
Status:

Aquest és un e-mail qualsevol,
el text s'escriu fins acabar
amb una línia que només conté un punt
.

QUIT
+OK Sayonara
Connection closed by foreign host.
```

Hi ha diverses implementacions de servidors POP i cada una compta amb un conjunt d'ordres i extensions pròpies. L'especificació POP3 requereix que s'implementin almenys les ordres STAT, LIST, RETR, DELE, NOOP i REST.

El model POP3

El POP3 és un protocol que permet l'accés remot a les bústies de correu dels usuaris, com l'IMAP. Ni el POP3 ni l'IMAP transporten el correu, aquesta funció la fa l'SMTP, sinó que ofereixen els mecanismes per a que un usuari amb el seu MUA pugui accedir a la seva bústia.

Com la majoria de serveis de xarxa a internet, el protocol POP3 s'estructura seguint l'esquema client/servidor. Aquests són els agents que intervenen en una comunicació POP3:

- **MUA** (*Mail User Agent* o **agent d'usuari de correu**). L'usuari interactua amb un agent d'usuari per accedir al correu mitjançant POP3. Són agents d'usuari programes com Thunderbird, GetMail, Fetchmail, MS Outlook Express, Eudora, Gmail... Les aplicacions MUA poden ser de text, gràfiques i fins i tot interfícies web. Aquestes aplicacions incorporen el programari necessari per actuar de clients POP3.
- **Client POP3**. La part pròpiament encarregada de comunicar amb el servidor POP3 per obtenir els missatges de la bústia de correu de l'usuari

és el client POP3. Client i servidor POP3 parlen un llenguatge comú, que és el protocol POP3.

- **Servidor POP3**. Per poder implementar l'accés remot al correu cal disposar d'un servidor POP3 en funcionament. Aquest servidor POP3 conté les bústies dels usuaris o hi accedeix. Els missatges es reben mitjançant SMTP, i és un MTA o un MDA el que els diposita a la bústia. El servidor POP3 atén les peticions dels clients POP3 per baixar el correu.

Un concepte que ajuda a diferenciar el funcionament de POP3 i IMAP és que en l'esquema de POP3 es considera que l'emmagatzematge del correu es realitza en la màquina de l'usuari. El servidor acumula els missatges nous i aquests es baixen tots a l'amfitrió (*host*) de l'usuari i s'eliminen del servidor. Per tant, gestionar-los és responsabilitat de l'usuari. Si bé és cert que els missatges es poden desar en el servidor (sense esborrar), no hi ha eines per gestionar-los, tots es troben en una mateixa carpeta. El servidor POP3 ofereix poques funcionalitats: baixar missatges, baixar les capçaleres i esborrar els missatges.

Accés POP3 al correu web

Molts serveis de correu web o *webmail* permeten baixar correu d'altres serveis usant el protocol POP3 o IMAP. Per exemple, des de Gmail es poden baixar missatges de Yahoo, i viceversa.

Una sessió POP3 passa per tres estats clarament diferenciats:

1. **Autorització**. Un cop feta la connexió TCP/IP pel port 110 entre el client i el servidor POP3, s'entra en l'estat d'autorització. Cal que el client s'identifiqui davant del servidor POP3 indicant el nom d'usuari i la contrasenya.
2. **Transacció**. Un cop el client ha estat autoritzat pel servidor, s'entra en l'estat de transacció. En aquest estat el client demana accions (dona ordres) al servidor i aquest les atén. És a dir, en aquest estat el client descarrega el correu, marca missatges per esborrar, demana les capçaleres dels missatges, en fa una llista per ordre... El client finalitza l'estat de transacció utilitzant l'ordre QUIT.
3. **Actualització**. El servidor entra en l'estat d'actualització en rebre l'ordre QUIT del client. Elimina els missatges marcats per esborrar (que fins ara no s'havien eliminat) i envia un OK al receptor. Ara tots dos poden finalitzar la comunicació.

1.3.2 Servei IMAP

IMAP (*Internet Message Access Protocol* o **protocol d'accés a missatges d'Internet**) és un protocol de capa d'aplicació del model TCP/IP que proporciona a l'usuari accés remot a la seva bústia de correu. L'IMAP sorgeix com a resposta al problema d'accés al correu des de diferents ordinadors utilitzant POP.

El POP és un protocol pensat per baixar el correu del servidor al PC local de l'usuari i poder-lo manipular després sense connexió a internet. Usant POP es considera que el correu resideix en l'equip de l'usuari, que baixa tot el correu de cop cada vegada que es connecta al servidor.

Quan els usuaris es van acostumar a consultar el correu remotament, ho van començar a fer des d'equips diferents: a casa, a la feina, de vacances... Cada cop que ho feien deixaven part del seu correu en llocs diferents. El que s'havien baixat a casa no es podia consultar a la feina, i viceversa. Un cop els usuaris es van acostumar a disposar de connexió d'internet més assíduament, calia un mecanisme més evolucionat d'accés remot al correu.

Ni l'IMAP ni el POP són protocols de transmissió de correu. Usualment és el protocol SMTP qui fa aquesta funció.

L'IMAP presenta un enfocament diferent. Els missatges de correu es dipositen en el servidor i allà s'emmagatzemen en carpetes (o *folders* o *mailboxes*) i on es manipulen. L'usuari els pot baixar localment, però com a còpia temporal. Per tant, tota la gestió dels missatges de correu té lloc en el servidor. Això fa de l'IMAP un protocol més complex que el POP.

Per obtenir més informació sobre l'especificació del protocol IMAP en els RFC 1064 i 3501, aneu a la secció "Adreces d'interès" del web d'aquest crèdit.

L'**IMAP** és un protocol de capa d'aplicació de la pila de protocols TCP/IP (port 143) definit en el document RFC 1064. Permet a un client de correu (MUA) obtenir remotament el correu dipositat en la bústia de l'usuari en un servidor IMAP.

L'IMAP sorgeix el 1986 amb el nom d'*Interim Mail Access Protocol*, que en la versió següent es canvia per *Interactive Mail Access Protocol* (document RFC 1064), i que finalment serà *Internet Mail Access Protocol*. L'evolució actual és IMAP versió 4 revisió 1 (març de 2003), corresponent al document RFC 3501 (del qual també s'han fet actualitzacions i extensions) que s'ha creat sota els auspicis de l'IETF.

Difusió del servei IMAP

Avui en dia l'IMAP està molt estès, però no és estrany trobar ISP i portals de correu web que permeten baixar el correu únicament mitjançant el POP.

El protocol IMAP està pensat per tenir en el servidor tot el correu de l'usuari organitzat en carpetes jeràrquiques de manera indefinida. Es permet la manipulació remota de les carpetes i els missatges. Tant les unes com els altres es poden crear, modificar i suprimir. Els missatges no s'esborren si no ho indica explícitament l'usuari. A més, aporta la funcionalitat de cerca i filtratge de missatges directament en el servidor. És a dir, no cal baixar els missatges per buscar els que compleixen unes condicions determinades. El protocol permet l'accés concurrent de diversos usuaris a la mateixa bústia, i el servidor pot notificar l'arribada de correu nou. Els missatges multipart es poden baixar parcialment, es poden buscar parts i baixar només les que interessin. Tots els missatges i les bústies tenen indicadors d'estat que descriuen, per exemple, si el missatge s'ha llegit, si s'ha contestat, si és nou...

El model IMAP

Com la majoria de serveis de xarxa a internet, el protocol IMAP s'estructura seguint l'esquema client/servidor. Aquests són els agents que intervenen en una comunicació IMAP:

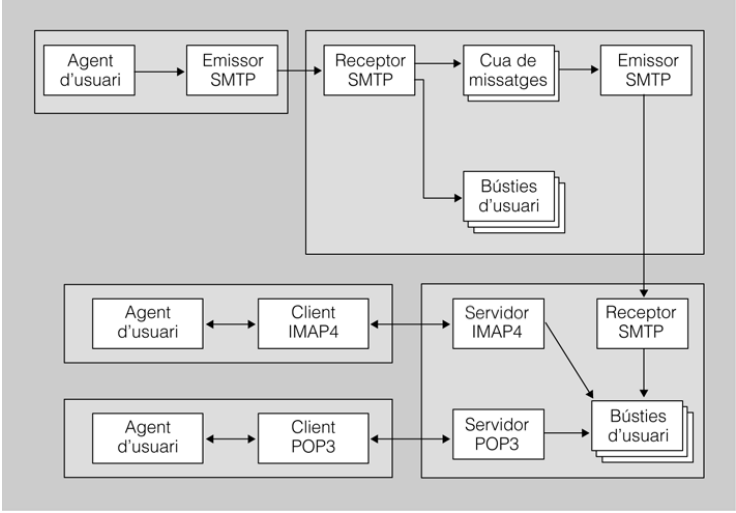
- **MUA** (*Mail User Agent* o **agent d'usuari de correu**). L'usuari interactua amb un agent d'usuari per accedir al correu mitjançant l'IMAP. Són agents d'usuari programes com Thunderbird, GetMail, Fetchmail, MS Outlook Express, Eudora o Gmail. Les aplicacions MUA poden ser de text, gràfiques

i fins i tot d'interfície web. Aquestes aplicacions disposen del programari necessari per actuar com a clients IMAP.

- **Client IMAP**. La part pròpiament encarregada de comunicar amb el servidor IMAP per obtenir els missatges de la bústia de correu de l'usuari és el client IMAP. Client i servidor IMAP parlen un llenguatge comú: el protocol IMAP.
- **Servidor IMAP**. Per implementar l'accés remot al correu cal disposar d'un servidor IMAP en funcionament. Aquest servidor IMAP conté les bústies dels usuaris o hi accedeix. Els missatges es reben per SMTP i és un MTA o un MDA els que els diposita a la bústia. El servidor IMAP atén les peticions dels clients IMAP per gestionar el correu.

Observeu el model funcional del protocol IMAP en la figura 1.2, on mitjançant el seu MUA un usuari descarrega el correu del servidor amb el POP o IMAP.

FIGURA 1.2. Model funcional del protocol IMAP



En el protocol IMAP s'observen quatre estats:

1. **No autenticat**. Quan s'estableix la connexió TCP/IP entre el client i el servidor s'entra en aquest estat. El client s'ha d'autenticar en el servidor, ha d'acreditar ser un usuari vàlid. Per fer-ho ha d'indicar el nom d'usuari i la contrasenya.
2. **Autenticat**. Un cop autenticat i abans de poder manipular missatges, ha de seleccionar la bústia (carpeta, *folder* o *mailbox*) amb la qual operarà. En aquest estat pot manipular les carpetes (crear-ne, esborrar-les, modificar-les i veure'n l'estat), però no els missatges fins que no se n'ha seleccionada una.
3. **Seleccionat**. Un cop s'ha seleccionat una carpeta, s'entra en aquest estat, que permet la manipulació dels continguts de la carpeta.
4. ***Logout***. En aquest estat es tanca la connexió. S'hi pot arribar tant per petició del client com per decisió unilateral del servidor.

El servidor IMAP emmagatzema permanentment els missatges de l'usuari. Per fer-ho utilitza un sistema de bústies o carpetes jeràrquiques i atributs que descriuen tant l'estat de les bústies com dels missatges:

Carpetes

Hi ha una bústia o *mailbox* que és la de l'usuari. Dins d'aquesta bústia, s'hi poden crear carpetes indicades de manera relativa, com els directoris d'una estructura de fitxers. Les carpetes disposen almenys de dos atributs:

- ***Next UID*** (UID següent). Indica l'UID que s'assignarà al missatge següent.
- ***UID Validity Value*** (UIDVALIDITY). És un valor d'identificador únic assignat a la carpeta seleccionada. La combinació de nom de carpeta UIDVALIDITY i UID identifica de manera perpètua un missatge en el servidor.

Atributs de missatge

Els missatges tenen atributs que s'emmagatzemen en les pròpies bústies que en faciliten la gestió:

- **UID**. Identificador únic del missatge. És un número de 32 bits que s'assigna ascendentment a mesura que arriben missatges (no és necessàriament correlatiu). Això permet al servidor saber, en una bústia, a partir de quin número de missatge hi ha els missatges nous (en POP això no és possible).
- **Número de seqüència**. Número relatiu del missatge dins de la bústia (de 1 a *n* per a *n* missatges). Els números de seqüència s'assignen correlativament segons el UID en ordre ascendent i varien en esborrar-se i afegir-se nous missatges.
- **Indicadors**. Els indicadors, *flags* o banderes informen de l'estat del missatge. Per exemple, si s'ha llegit o esborrat. Els indicadors són: *Seen* (llegit), *Answered* (respost), *Flagged* (marcat), *Deleted* (esborrat), *Draft* (esborrany) i *Recent* (nou).
- **Data interna**. Data i hora d'arribada del missatge al servidor IMAP (no és la data i hora de l'emissió del missatge que hi ha en la capçalera *Date*).
- **Longitud**. Nombre de *bytes* del missatge.
- **Estructura del sobre**. Representació analitzada de les capçaleres del missatge.
- **Estructura del cos**. Representació analitzada de l'estructura MIME del cos del missatge.
- **Parts de text del missatge**. Per permetre la cerca de les diferents parts de text del missatge. Es pot fer l'accés segons la part de capçaleres, cos, part del cos MIME i capçalera MIME.

Funcionament de l'IMAP

Diferència entre POP i IMAP

En el protocol POP el servidor només pot respondre a peticions del client, però no pot prendre la iniciativa. En el protocol IMAP sí.

Amb el protocol IMAP el client fa una connexió TCP/IP al port 143 del servidor i s'inicia un diàleg entre el client i el servidor en què tots dos poden prendre la iniciativa. En aquest procés se succeeixen els quatre estats del protocol IMAP (no autenticat, autenticat, seleccionat i *logout*) i s'intercanvien ordres i respostes seguint el model de diàleg de Telnet:

- **Ordres**. Són ordres de text que inclouen un *tag* (o etiqueta curta) inicial, l'ordre i els seus arguments. Cada ordre comença amb una etiqueta inicial diferent per diferenciar-la de les altres ordres i aquesta és obligatòria. Quan el servidor emeti la resposta final de l'ordre i indiqui si s'ha realitzat correctament o no, ho farà mostrant l'etiqueta de l'ordre a la qual respon. Per exemple, es pot usar *a001* per a la primera ordre, *a002* per a la segona... El client pot enviar ordres sense esperar que finalitzin les anteriors.
- **Respostes**. El servidor pot enviar dades al client tant com a resposta a una ordre com de manera unilateral (per exemple, per informar que hi ha correu nou). El client ha d'estar en tot moment a punt per rebre aquestes dades. Que el servidor enviï dades al client no significa que l'execució de l'ordre del client hagi finalitzat. Només es dona per finalitzada quan el servidor emet una resposta amb la mateixa etiqueta que l'ordre del client. El servidor pot processar una ordre abans d'acabar de processar l'ordre anterior.

Vegeu les ordres utilitzables en el protocol IMAP agrupades segons l'estat:

Avantatges de l'IMAP respecte al POP

Un dels avantatges de l'IMAP respecte al POP és que el servidor sap a partir de quin número de missatge hi ha els missatges nous (RECENT).

1. **Ordres generals** (qualsevol estat)
   - **Capability**. Llista les capacitats del servidor. Permet al client saber quines són les prestacions del servidor.
   - **Noop**. Exigeix una resposta afirmativa del servidor. Permet al client saber si encara es manté la connexió.
   - **Logout**. És la notificació del client al servidor per fer-li saber que vol finalitzar la connexió.
2. **No autenticat**
   - **Authenticate <tipus>**. Indica al servidor el mecanisme d'autenticació a utilitzar.
   - **Login <user> <password>**. El client s'identifica en el servidor indicant el nom d'usuari i la contrasenya. El format varia (text net, *hash* MD5...) segons el tipus d'autenticació utilitzat.
3. **Autenticat**
   - **Select <bústia>**. Selecciona la bústia amb què ha d'operar. En fer-ho, el servidor emet una resposta no etiquetada en què informa dels atributs (*flags*) de la bústia, del nombre de missatges que conté (*exists*) i del nombre de missatges recents (*recent*). També pot indicar el

número del primer missatge no llegit (*nseen*) i la llista d'atributs que es poden modificar (*permanent flags*).

- **Examine <bústia>**. Realitza la mateixa funció que l'ordre *Select* però només de lectura.
- **Create <bústia>**. Crea la bústia amb el nom indicat.
- **Delete <bústia>**. Esborra la bústia indicada.
- **Rename <bústia> <nomNou>**. Permet assignar un nom nou a la bústia.
- **Subscribe <bústia>**. Les bústies poden estar actives o no. Aquesta ordre les activa.
- **Unsubcribe <bústia>**. Permet desactivar una bústia.
- **List <bústia> <criteri>**. Llista les bústies que compleixen el criteri indicat dins de la bústia seleccionada.
- **Lsub <bústia> <criteri>**. Realitza la mateixa funció que l'ordre anterior però només per a les bústies actives.
- **Status <bústia> <atributs>**. Permet conèixer l'estat d'una bústia per mitjà dels seus atributs. Els atributs són els següents:
  - *MESSAGE*: nombre de missatges dins la bústia
  - *RECENT*: nombre de missatges recents (nous)
  - *UIDNEXT*: UID assignat al següent missatge que arribi a la bústia
  - *UIDVALIDITY*: valor UID de la bústia
  - *UNSEEN*: nombre de missatges no llegits
- **Append bústia [atributs] [data-hora] literal**. Permet afegir un text al final de la bústia com si es tractés d'un missatge nou. El missatge s'afegeix amb la data, hora i atributs indicats.

4. **Seleccionat**

- **Check**. El client sol·licita al servidor que es faci un control de la bústia en un moment determinat.
- **Close**. Tanca la bústia i elimina tots els missatges que conté que tenen l'indicador d'esborrament (*deleted*) activat.
- **Expugne**. Permet esborrar els missatges que tenen l'atribut d'esborrament (*deleted*) activat sense que calgui tancar la bústia.
- **Search [charset] criteri**. Permet buscar missatges dins de la bústia que compleixen el criteri de cerca indicat.
- **Fetch dades atributsRecuperació**. Permet recuperar un conjunt de missatges totalment o parcialment segons els atributs de recuperació indicats.
- **Store conjuntMissatges atributs**. Permet alterar les dades d'atributs associats a un conjunt de missatges en una bústia.
- **Copy conjuntMissatges bústia**. Copia un conjunt de missatges al final de la bústia indicada.

- **UID ordre**. Retorna l'UID d'un missatge. S'utilitza conjuntament amb les ordres COPY, FETCH, STORE o SEARCH per retornar els UID manipulats.

5. **Experimental**

   - **X<ordre>**. Es poden desenvolupar ordres experimentals fora de l'especificació IMAP. Per fer-ho cal que les ordres comencin amb *XnomOrdre*. D'aquesta manera s'evita que es produeixin conflictes amb ordres futures.

En l'exemple següent podeu veure tot el diàleg client/servidor d'una sessió IMAP utilitzant Telnet:

```
[root@portatil ~]# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is ,^]'.
* OK [CAPABILITY IMAP4REV1 LITERAL+ SASL-IR LOGIN-REFERRALS
STARTTLS] localhost IMAP4rev1 2007a.403 at Sat, 14 Jun 2008
13:16:47 +0200 (CEST)

a003 LOGIN pere pere
a003 OK [CAPABILITY IMAP4REV1 LITERAL+ IDLE UIDPLUS NAMESPACE
CHILDREN MAILBOX-REFERRALS BINARY UNSELECT ESEARCH WITHIN SCAN
SORT THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User
pere authenticated

a004 SELECT inbox
* 4 EXISTS
* NO Mailbox vulnerable - directory /var/spool/mail must have
1777 protection
* 4 RECENT
* OK [UIDVALIDITY 1213385060] UID validity status
* OK [UIDNEXT 6] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft
\Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/pere
* NO Mailbox vulnerable - directory /var/spool/mail must have
1777 protection
a004 OK [READ-WRITE] SELECT completed

a005 FETCH 1 rfc822.text
* 1 FETCH (RFC822.TEXT {63}
exemple de missatge de la usuària anna
a l'usuari pere
adéu
)
a005 OK FETCH completed

a006 LOGOUT
* BYE portatil IMAP4rev1 server terminating connection
a006 OK LOGOUT completed
```

1.3.3 Clients de correu

L'accés a les bústies de correu electrònic a través de les ordres i comandes que ofereixen els protocols IMAP i POP és complex i per a un usuari normal, impracticable. No obstant, existeix tot un conjunt de programari que automatitza totes

aquestes tasques i les presenta en un entorn més amigable. Aquest programari s'anomena **client de correu**. Existeixen clients de correu de característiques molt diverses.

En general es poden classificar en:

- **Clients de text**. Utilitats Unix i GNU/Linux de tota la vida que permeten generar missatges de correu i accedir a la bústia de correu pròpia. Són utilitats de consola, és a dir, de text. N'hi ha de format ben simple, com l'ordre *mail*, i evolucions que permeten treballar amb programes de menús en color també en entorn de consola, com Mutt, Alpine...
- **Clients gràfics**. Amb la popularització dels entorns gràfics van sorgir programes client de correu com Eudora, Evince, Outlook o Thunderbird. Durant un temps aquests programes eren l'eina més usada pels usuaris per accedir a les seves bústies de correu.
- **Client web**. Actualment la majoria de comptes de correus dels usuaris són de tipus correu web, principalment en serveis gratuïts com Gmail, Yahoo o Hotmail. Fins i tot els entorns corporatius utilitzen correu web hostatjat en serveis externalitzats.

1.4 Correu encriptat i signat

Les comunicacions per correu electrònic s'han tornat cada vegada més importants en la nostra vida diària, no només pels correus amb acudits, presentacions gracioses o crítiques al govern. També són imprescindibles per al funcionament de moltíssimes empreses i organitzacions. De fet, molta documentació que abans es feia per escrit ara es fa telemàticament per correu electrònic.

Això fa necessari poder estar segurs de la identitat de l'emissor i del receptor dels correus. En entorns de seguretat més exigents fins i tot es pot requerir l'encriptació del contingut per evitar que sigui accessible per tercers.

El concepte clau per a la confiança en les comunicacions per correu electrònic és la **signatura digital**, que permet assegurar que un emissor és qui diu ser i garanteix també que el contingut del missatge no s'ha alterat per tercers. Per implementar la signatura digital i l'encriptació cal utilitzar **certificats digitals**.

Consulteu a "Annexos" l'explicació més àmplia de conceptes globals de seguretat i documents específics per a PGP, S/MIME i certificats digitals.

És important tenir clars els conceptes generals de seguretat per poder diferenciar els termes relacionats amb la seguretat dels quals no es coneix el significat amb exactitud.

Els puntals de la seguretat en el correu són: autenticació, integritat, no repudi i encriptació. Es poden crear parelles de claus públiques i privades (certificats digitals) i hi ha diversos formats de claus existents.

L'MTA i els servidors de correu transporten missatges independentment del fet de que estiguin encriptatts o signats. Són els **clients de correu** els que han de proporcionar a l'usuari la capacitat d'**encriptar** i **signar** missatges .

Per poder gestionar correu signat i encriptat cal utilitzar clients de correu que permetin fer aquestes funcions. Cada usuari ha de disposar dels certificats digitals apropiats. Per al protocol de transport de correu SMTP i al servidor de correu (per exemple, Sendmail) que el missatge que processen sigui encriptat i signat és indiferent, igual que al carter no li afecta que el contingut d'una carta estigui encriptat amb una clau secreta o que la carta porti imprès el segell d'una entitat. Són els clients de correu, com Thunderbird, els que han de tenir la capacitat de gestionar aquest tipus de correu. Per exemple, a Thunderbird cal afegir-li programari addicional (Open PGP o S/MIME, per exemple) per permetre-li signar i encriptar missatges.

Les dues tecnologies de signat i encriptat de correu tractades aquí són:

- **Open PGP**. Aquesta tecnologia permet generar missatges de correu encriptats i signats i, lògicament, desxifrar-ne el contingut i verificar les signatures. Es basa en el programa PGP (*Pretty Good Privacy*), desenvolupat per Phil Zimmermann. Utilitza el model de seguretat anomenat *web of trust*, en què cada usuari és el responsable de la gestió de la confiança en els certificats dels altres usuaris.
- **S/MIME**. Aquest estàndard proporciona les mateixes característiques de signatura digital i encriptació (i a la inversa) que Open PGP, però requereix un altre format per als certificats digitals. El model de seguretat que utilitza és el PKI (*Public Key Infrastructure*), en què existeix una estructura piramidal d'entitats de certificació (*Certification Authority*, CA) que determinen la confiança en els certificats digitals. Aquest PKI és el model en què es basen la majoria de protocols de seguretat d'internet, com HTTPS o SMTPS, que utilitzen SSL o TLS. Usualment els navegadors web incorporen per defecte els certificats de les CA més destacades del món.

1.4.1 Seguretat en el correu

L'intercanvi de missatges de correu es produeix utilitzant protocols SMTP, POP i IMAP, que són insegurs, és a dir, que transporten el contingut en forma de text pla. Per tant, qualsevol intermediari en la xarxa pot monitorar (usant un *sniffer* o rastrejador) el contingut dels missatges. Qualsevol eina tipus Wireshark permet fer un seguiment dels continguts de qualsevol protocol TCP en text pla.

Monitoratge de contingut

Un exemple típic de monitoratge és monitorar el diàleg entre dos amfitrions usant Wireshark. Permet fer un seguiment de les trames de xarxa i utilitzant l'opció de seguiment del flux TCP anomenada *Follow TCP Stream* es pot observar clarament el text de tot el diàleg efectuat.

Això significa que quan dos interlocutors intercanvien missatges per qualsevol xarxa el contingut del seu diàleg pot ser monitorat per tercers. I no només això, sinó que la conversa pot ser falsejada per aquests tercers (el conegut com a *man-in-the-middle*). És a dir, un usuari pot creure que està dialogant amb la seva entitat bancària quan en realitat ho està fent amb uns impostors.

1.4.2 Propietats de seguretat

Els aspectes de seguretat desitjables en la comunicació entre dos interlocutors són:

- Confidencialitat (encriptació)
- Autenticitat
- No-repudi
- Integritat

Un error comú es barrejar aquests conceptes i pensar que tots quatre van junts, i no sempre és així. Cal aplicar a cada cas el tipus de seguretat que faci falta.

Confidencialitat

Si un emissor vol enviar un missatge secret a un receptor de manera que únicament aquest el pugui entendre, ha d'encriptar el missatge. El destinatari ha de conèixer la clau o ha de disposar d'un mecanisme per desencriptar el missatge i obtenir-ne el contingut original.

> **Encriptar** és codificar el missatge utilitzant algun tipus de clau o mecanisme per fer inaccessible el missatge a qualsevol usuari que no sigui el destinatari.

Que el missatge estigui encriptat garanteix que només l'emissor és capaç d'entendre'n el contingut (se suposa que han acordat el mecanisme per desencriptar). Ara bé, pot estar segur el destinatari que el missatge prové realment de qui creu que prové? Pot estar segur que el missatge no ha estat alterat?

Encriptar un missatge proporciona **confidencialitat**, però no és un mecanisme que ofereixi la seguretat que l'emissor és qui diu ser ni que el missatge és tal com era en l'origen (sense modificacions posteriors).

Autenticitat

Quan un receptor rep un missatge del banc informant-lo que ha de fer un pagament o que ha d'enviar el número de la seva targeta de crèdit per una raó determinada, però no pot estar segur el receptor que el missatge procedeix realment del banc.

L'**autenticació** és la característica de seguretat que permet a un receptor estar segur que el missatge prové de qui diu provenir. Alhora, proporciona a l'emisor la garantia que el receptor sap del cert que el missatge li ha enviat ell.

No-repudi

Si un emisor envia un missatge autenticat a un emissor no té manera legal de desdir-se d'haver-lo enviat. Imagineu que un directiu d'una empresa envia un missatge inapropiat a un treballador. Si el missatge és autenticat, el directiu no pot negar legalment que l'ha enviat. El mateix pot passar si l'empresa A envia un correu electrònic a un client oferint-li els productes a meitat de preu, i després intenta retractar-se'n dient que el correu no era seu. Si el missatge és autenticat, queda legalment demostrat que l'empresa A n'és l'emissora.

Aquesta característica que impedeix que l'emissor pugui negar haver enviat el missatge s'anomena **no-repudi**.

Cal un mecanisme de seguretat que permeti a un emissor enviar missatges de manera que els receptors tinguin la certesa absoluta que l'emissor és qui diu ser.

Una confusió habitual és creure que per establir seguretat també cal encriptar. Això no és cert. Una administració pública, per exemple, pot enviar missatges als ciutadans garantint l'autenticitat del missatge, però no li cal encriptar els missatges, no li cal fer-los secrets.

Integritat

No n'hi ha prou amb l'encriptació i l'autenticació per establir una comunicació cent per cent segura. S'ha d'assegurar que el missatge no ha estat alterat, és a dir que no s'han eliminat o afegit parts, ni tampoc que se n'hagin modificat. S'ha d'assegurar que el missatge està íntegre. L'encriptació i l'autenticació asseguren que el missatge procedeix de qui diu procedir i que no ho veu ningú més, però no s'assegura que no hagi estat modificat.

La **integritat** és la propietat de seguretat que garanteix que el missatge no ha estat alterat i que arriba al destinatari tal com s'ha generat en l'origen.

Per implementar integritat no cal encriptar els missatges, són coses independents. En canvi, la integritat i l'autenticitat van juntes, és a dir, tenint l'una també s'obté l'altra (i viceversa).

1.4.3 Implementació de seguretat

Per implementar seguretat en la comunicació avui en dia s'utilitzen habitualment els mecanismes de certificats digitals basats en la criptografia de clau asimètrica. És a dir, basats en el fet que cada interlocutor disposa d'una clau privada (coneguda i accessible només per ell) i d'una clau pública o **certificat** (coneguda per tots els interlocutors).

Els certificats són les claus públiques **signades**, avalades, per una entitat de certificació o CA (*Certification Authority*).

Per implementar seguretat els interlocutors disposen de: una **clau privada** i un **certificat**, o clau pública signada per una CA.

Vegeu com s'implementa la seguretat de clau pública/privada:

- Encriptació
- Sigatura o certificat digital

En realitat, la qüestió de la seguretat és més complexa si s'hi afegeixen les CA, els anells de clau, el sistema PKI... En aquesta unitat es tracta la versió més simple d'aquest model i s'estudia només allò estrictament necessari per a la comunicació segura entre dos interlocutors.

Cal tenir molt present que la clau privada és això, privada: només la coneix i només hi pot accedir el seu propietari. Mai es comunica a un tercer. En canvi, la clau pública és coneguda per totes les parts que intervenen en la comunicació. De fet, si els interlocutors no la coneixen cal enviar-los-la per tal que la tinguin. Si no fos així, el sistema de clau pública/privada no funcionaria.

La parella clau pública/privada permet encriptar i signar. Quan se n'utilitza una per fer una acció, cal l'altra per desfer-la. Funcionen conjuntament: si una fa, l'altra desfà, i a l'inrevés.

Mala interpretació del funcionament de les claus

Un error típic és creure que cada clau només pot fer una cosa i dir que "la clau privada sempre encripta i signa i la pública desencripta i verifica". No va així. Segons l'acció a fer se n'utilitza una o l'altra (i per desfer l'acció sempre cal aplicar la contrària).

Encriptació

Un **emissor A** vol enviar un missatge encriptat a un **destinatari B**, de manera que únicament aquest tingui la capacitat de conèixer el contingut real del missatge.

Per fer-ho, l'emissor encripta el missatge amb la clau pública del destinatari. De fet, qualsevol emissor del món ho pot fer, precisament perquè la clau pública del destinatari és pública.

Un cop encriptat el missatge, únicament el pot desencriptar qui tingui la clau privada associada a la clau pública usada per encriptar-lo. És a dir, només podrà desencriptar el missatge el destinatari.

Publicació de la clau pública

La clau pública ha de ser coneguda per tots els participants en la comunicació. Els mecanismes per donar-la a conèixer són:

- Publicar-la en un servidor públic de claus.
- Enviar la clau als destinataris als quals s'adreça l'emissor (amics, coneguts i saludats).
- Adjuntar la clau pública als missatges.

Què coneix l'emissor del destinatari? La seva clau pública. De manera que l'emissor encripta el missatge amb la clau pública del destinatari. De fet, qualsevol emissor del món ho pot fer, precisament perquè la clau pública del destinatari és pública.

Un cop encriptat el missatge, la resta (inclòs l'emissor) no poden desencriptar-lo. Únicament el pot desencriptar qui tingui la clau privada associada a la clau pública usada. És a dir, només podrà desencriptar el missatge qui disposi de la clau privada del destinatari B, és a dir que només el destinatari podrà fer-ho.

Qualsevol pot **encriptar** un missatge usant la **clau pública del destinatari**, que precisament és pública. Únicament el destinatari pot **desencriptar** el missatge usant la seva **clau privada**.

Signatura

Signar un missatge proporciona **integritat**, **autenticació** i **no-repudi** simultàniament. Quan un missatge està signat per l'emissor és irrefutable que el missatge procedeix d'ell i, a més, garanteix que no s'ha modificat per tercers. El receptor pot **verificar** així que el missatge és autèntic.

El procés físic de signar el missatge consisteix a afegir al missatge el certificat digital de l'emissor. De vegades el fet de signar un missatge s'anomena certificar-lo o es parla de missatge amb certificat digital.

El funcionament és força similar al de l'encriptació, però en aquest cas l'emissor **signar** un missatge utilitzant la seva pròpia **clau privada**, i el receptor utilitza la **clau pública** de l'emissor per **verificar** el missatge.

Com que les parelles de claus pública/privada només funcionen conjuntament, la verificació només serà correcta si el missatge s'ha encriptat amb la clau privada corresponent. És a dir, la verificació amb la clau pública de l'emissor A només funcionarà si el missatge s'ha signat amb la clau privada de l'emissor A.

Qualsevol pot **verificar** la signatura d'un missatge usant la **clau pública de l'emissor**. Únicament l'emissor pot **signar** el missatge usant la seva **clau privada**.

El procés de signatura és complementari al de l'encriptació. El primer proporciona integritat, autenticació i no-repudi, mentre que el segon proporciona confidencialitat.

Que un missatge estigui **signat** no significa que sigui secret. Perquè sigui **secret** també ha d'estar **encriptat**.

El procés mecànic de signar un missatge consisteix a aplicar la clau privada al missatge. Això comporta els processos següents:

- **Integritat**: no es codifica tot el missatge amb la clau privada, ja que això implica un sobrecost de temps i esforç de càlcul. El procés tècnic que es fa és generar un ***hash*** o resum del missatge i signar-lo. Si el missatge es modifiqués, el resum no coincidiria amb el missatge.
- **Autenticitat**: per autenticar el missatge s'adjunta el certificat o clau pública de l'emissor avalat per una autoritat de certificació o CA. Aquest certificat i el resum van codificats amb la clau privada. No tot el missatge, només aquesta part. Així el receptor verifica amb la clau pública de l'emissor que el certificat de l'emissor és vàlid i que el *hash* també.

Tècniques de 'hash' o resum

Una funció resum (*hash*) és una funció que permet reduir qualsevol informació de qualsevol mida a un valor de mida fixa. Entre les seves utilitats hi ha la validació de la integritat de fitxers (tant per temes d'autenticitat com de comprovació d'errors en la transmissió), autenticació amb signatura digital, i en programació i base de dades per a la indexació de les dades.

Són funcions exhaustives, i moltes vegades s'anomenen de direcció única, ja que la funció inversa no dona un únic resultat. A més, el càlcul de la funció inversa és costós. Per exemple, per calcular el valor d'origen normalment es recorre a atacs de força bruta.

Algunes funcions resum són els CRC (*Cyclic Redundancy Checks*), els MD (*Message Digest*) i els SHA (*Secure Hash Algorithm*).

En definitiva, el procés de signar un missatge o incorporar al missatge una signatura digital consta dels passos següents:

1. Es genera un *hash*, *message digest* o *fingerprint* del missatge.
2. S'encripta el resum amb la clau privada de l'emissor. Això és la signatura digital.
3. El destinatari rep el missatge i fa un nou resum, és a dir, torna a fer el *hash* basant-se en el contingut del missatge rebut.
4. La signatura digital rebuda (resum encriptat) es desencripta amb la clau pública de l'emissor.

Per tant:

- Els dos resums han de ser iguals: això garanteix l'autenticació i la integritat.

- Es garanteix l'autenticació: només l'emissor pot haver generat el missatge si es pot desencriptar per la clau pública de l'emissor.
- Es garanteix la integritat: ningú més pot haver modificat el missatge perquè això hauria modificat el resum i no hi ha ningú més que tingui la clau privada de l'emissor per tornar-lo a signar.
- No cal encriptar tot el missatge, només el resum, que és la part que garanteix que no s'ha modificat.

1.5 Servidor de correu segur

Els protocols de correu analitzats en aquesta documentació són tots protocols de transport d'informació en text pla. Qualsevol comunicació SMTP, POP o IMAP es fa en text pla i pot ser monitorada per tercers que tinguin accés als nodes de xarxa per on passen aquestes comunicacions. De fet, s'han vist exemples de monitoratge de les converses TCP utilitzant Wireshark. Aquesta feblesa no és exclusiva dels protocols de correu, sinó que és comuna a la majoria de protocols d'Internet, com HTTP, FTP o TFTP.

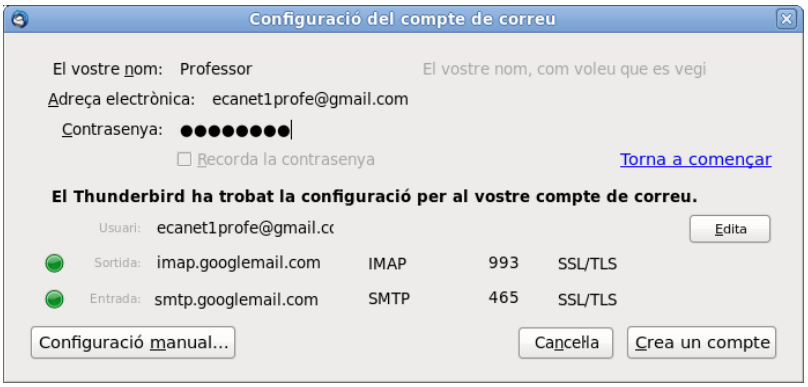
Els protocols de correu tenen mecanismes similars als usats en HTML per establir canals de comunicació segurs:

- **SMTPS**: no és un protocol diferent de l'SMTP ni una extensió seva, és simplememnt una manera d'anomenar l'ús d'SMTP a través d'SSL o TLS. Així com l'usuari estableix comunicacions HTTP segures usant HTTPS, pot establir un transport de correu segur amb SMTPS. El protocol és el mateix, però viatja per SSL, que li proporciona seguretat. Per usar SMTPS cal comunicar-se per un port diferent del 25, en concret el 465. Això representava un problema, ja que el sistema de correu a Internet es basa majoritàriament en la utilització del port 25, però aquest problema es va resoldre amb la utilització d'STARTTLS
- **POPS**: com passa amb HTTPS i SMTPS, POPS no és un protocol pròpiament dit, sinó tan sols la utilització de l'accés remot a bústies POP amb transport segur SSL o TLS. Utilitza el port 995 en lloc del port 110 clàssic de POP
- **IMAPS**: és també la manera d'anomenar el protocol IMAP quan viatja per una capa de transport segur com SSL o TLS. El port utilitzat per IMAP sobre SSL és el 993

Els acrònims SMTPS, POPS i IMAPS indiquen la utilització del protocol usant una **capa de transport segura** SSL o TLS, cosa que permet una comunicació encriptada que no pot ser monitorada per tercers (com passa amb HTTP sobre SSL, que s'anomena HTTPS).

En la figura 2.2 es pot veure la pantalla de Thunderbird de creació d'un compte de correu de Gmail. Observeu que el correu entrant utilitza IMAP sobre SSL/TLS al port 993 del servidor imap.googlemail.com, que és el servidor de correu de Gmail. També es configura com a correu sortint el servidor SMTP sobre SSL/TLS de Google. S'utilitza el port 465 del servidor smtp.googlemail.com, que és el nom de l'amfitrió de Gmail que fa la funció de servidor SMTP.

FIGURA 1.3. Configuració de compte de correu a Gmail



L'exemple següent mostra com s'inicia un diàleg en mode consola amb un servidor SSL. S'utilitza una de les utilitats de l'ordre *openssl*, que permet actuar com a client SSL. En aquest exemple es contacta el servidor IMAP de Gmail:

```
1 [root@host ~]# openssl s_client -crlf -connect imap.gmail.com:993
2 [root@host ~]# openssl s_client -crlf -connect smtp.googlemail.com:465
```

2. Instal·lació i administració de serveis de missatgeria instantània, notícies i llistes de distribució

Els serveis de missatgeria instantània, notícies i llistes de distribució complementen el sistema de missatgeria tradicional basat en el correu electrònic, cadascun amb les seves pròpies característiques.

La missatgeria instantània permet la comunicació en temps real, i aquesta és una de les principals diferències amb el servei de correu electrònic. Aquest tipus de comunicació permet incloure més d'un interlocutor i en diferents formats multimèdia com el text, l'àudio i/o el vídeo.

Les llistes de distribució permeten rebre missatges de correu electrònic sobre un tema determinat. La diferència entre el correu electrònic i les llistes de distribució rau en el fet que no coneixem el destinatari, és a dir, quan enviem un missatge a la llista, aquesta la reenvia als seus subscriptors en el cas de llistes de discussió. Un altre cas són les llistes de publicació on els missatges són unidireccionals (només rebem missatges).

El servei de notícies funciona de manera molt similar a un tauler d'anuncis on es poden penjar i consultar notícies. En aquest cas tampoc cal especificar cap destinatari, ja que són els mateixos usuaris els que van a aquest tauler. Aquest servei es realitza a través d'un protocol específic, el NNTP (*Network News Transfer Protocol*).

2.1 Missatgeria instantània

L'evolució de les tecnologies, l'abaratiment de costos i la popularització d'internet han provocat que avui en dia una gran quantitat de dispositius permetin navegar i comunicar-se per internet. No només mitjançant els ordinadors, sinó també amb els mòbils, les tauletes tàctils i fins i tot els televisors. Una forma popular de comunicar-se entre usuaris és per xats i missatgeria instantània, que permeten la transmissió en temps real de text, àudio i vídeo.

La **missatgeria instantània** o **IM** (*Instant Messaging*) proporciona comunicació entre dos o més usuaris en temps real. Aquesta comunicació pot ser en format de text, àudio i vídeo.

Per entendre millor el concepte d'IM, cal contrastar-lo amb altres formes de comunicació i la seva evolució.

Una de les formes clàssiques de comunicació entre dos interlocutors és la **correspondència**: enviar-se cartes. En les comunicacions informàtiques aquest

mecanisme s'implementa amb el correu SMTP. La diferència del correu electrònic amb l'IM és que el primer és una comunicació **asíncrona**, mentre que en els diàlegs IM (comunicacions de xat en línia) la comunicació és **síncrona**. Els missatges s'envien en temps real i els interlocutors emeten i reben missatges amb immediatesa, construint una comunicació fluida.

Seguint l'evolució històrica de la comunicació social, després del correu postal una altra forma popular de comunicació va ser el **telèfon**. Aquest proporciona una comunicació **síncrona** entre dos interlocutors que dialoguen en temps real. Un primer mecanisme per implantar en les xarxes informàtiques un equivalent al telèfon va ser l'aplicació **talk**, que permetia el diàleg de text en temps real entre dos interlocutors. De fet, talk és un predecessor dels sistemes d'IM actuals. Tant el telèfon com talk permeten la comunicació entre dos interlocutors que coneixen d'antuvi la identitat l'un de l'altre (el número de telèfon o l'adreça usuari@host del destinatari).

Amb l'expansió d'internet va sorgir un tipus de comunicació anomenat **xat**, en el qual podem incloure la missatgeria instantània, tot i algunes particularitats. Els xats permeten la comunicació entre dos o més clients en "sales" d'un servei de xat. Els usuaris s'identifiquen per un sobrenom (*nickname*) que generalment no té perquè ser un nom real (permet un cert anonimat). Molts xats funcionen per interfície web i també amb programes clients específics.

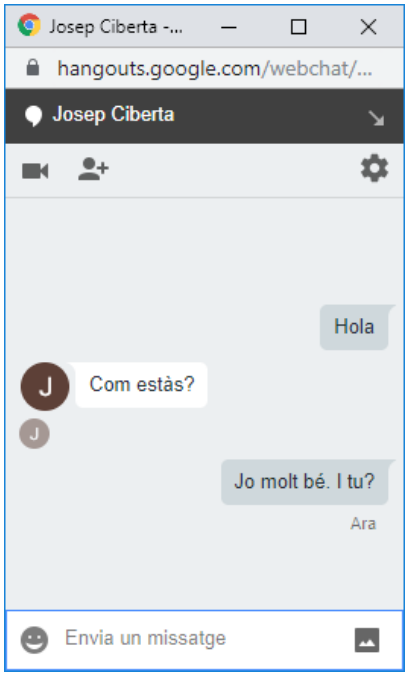
El sorgiment del telèfon mòbil i els **missatges SMS** van ser el següent pas evolutiu en aquest camp. Els missatges SMS són una forma de missatgeria instantània per text, on temps real i **síncron** és una mica més dubtós.

Actualment, els telèfons mòbils presten la majoria dels serveis de comunicació (navegació web, xats i missatgeria instantània, entre d'altres) a través d'internet. També els televisors incorporen prestacions d'Internet com l'accés a continguts a la carta, la navegació web i la missatgeria instantània. Sí, no és estrany trobar televisors que incorporen una càmera i un micròfon i que permeten utilitzar eines de comunicació com Skype.

La **comunicació per missatgeria instantània** (IM) es diferencia del correu i de les llistes de distribució pel fet que és comunicació **en temps real** o **síncrona**. És una comunicació de tipus xat, però es diferencia en el fet que és una comunicació no anònima, sinó que es produeix entre dos o més **interlocutors coneguts**.

En la figura 2.1 es pot veure un diàleg xat entre dos usuaris que usen el client de missatgeria instantània Google Hangouts. Observeu que les prestacions d'aquest servei de missatgeria inclouen el xat en text i vídeo i la realització de trucades telefòniques.

FIGURA 2.1. Exemple de diàleg de missatgeria instantània



En resum, la missatgeria instantània té com a finalitat comunicar dos o més interlocutors. Es pot dir que "xategen", ja sigui en format de text, àudio o vídeo. Aquests usuaris es volen comunicar específicament entre ells. Per exemple, en Pere vol parlar amb l'Anna per acordar qui farà el sopar avui o a qui li toca recollir els nens. Sovint la comunicació es dona en una finestra de diàleg que s'obre en rebre la notificació que algú altre vol parlar amb nosaltres o quan establim una connexió seleccionant algun del programes client d'IM existents. Tot i que aquestes comunicacions s'anomenen xat, cal diferenciar-les del xat basat en l'**IRC** (*Internet Relay Chat* o conversa interactiva per internet), en què usuaris (anònims o no, desconeguts o no) s'agrupen en sales per debatre temes i aficions o simplement per xerrar.

Funcionalitats: els sistemes de missatgeria instantània actuals permeten les comunicacions no només en format de text (el xat clàssic) sinó també usant àudio i vídeo. Depenent de les aplicacions utilitzades es proporcionen més funcionalitats i tot. Algunes d'elles són:

- **Text**: les comunicacions entre els interlocutors es produeixen amb missatges de text que s'escriuen usualment en finestres de diàleg. A mesura que un interlocutor va escrivint les frases apareixen en la finestra de tots els interlocutors.
- **Àudio**: una primera evolució del xat de text va ser incorporar àudio. D'aquesta manera els interlocutors podien parlar entre ells utilitzant la comunicació IM. Evidentment, cal disposar de micròfon i d'altaveus per poder establir aquest tipus de comunicació.
- **Vídeo**: actualment la majoria d'equips informàtics disposen de càmera, micròfon i sistema de so, cosa que permet establir comunicacions audiovisuals

en temps real. En aquestes comunicacions els interlocutors no només parlen sinó que es veuen com si fossin l'un davant de l'altre.

- **Transferència de fitxers**: una característica afegida que proporcionen sovint els clients IM és la transferència de fitxers. La transferència no es realitza a través del servidor ni és un protocol especialment dissenyat per a aquesta finalitat. Simplement, les aplicacions client permeten que en una comunicació directa entre dos interlocutors es puguin transferir fitxers usant el canal de comunicació establert.

- **Compartir escriptori**: de la mateixa manera que es comparteixen fitxers entre dos clients es poden realitzar transferències de tota mena de dades directes. Un cas concret és compartir un escriptori. Això significa que un usuari pot atorgar el permís perquè el seu escriptori sigui visible i també governable, si vol, per l'altre interlocutor. Tot i els perills de seguretat que això comporta, és un mecanisme molt pràctic per solucionar problemes a distància i per ensenyar als usuaris coses que poden fer amb el seu ordinador.

- **Ubicació en el mapa**: un servei més que es pot oferir als usuaris amb la missatgeria instantània és indicar la posició de l'usuari i localitzar la dels altres usuaris via GPS. L'usuari pot fins i tot satisfer la curiositat de veure "qui hi ha per aquí", és a dir, detectar usuaris geogràficament propers.

- **Trucades telefòniques**: ha esdevingut molt popular fer trucades telefòniques per Internet mitjançant serveis com Skype o usant clients IM que disposen d'aquesta prestació. Es pot trucar a telèfons fixos i mòbils. Usualment aquest servei és de pagament.

2.1.1 Funcionament de la missatgeria

Per entendre el funcionament dels serveis de missatgeria instantània, cal tenir clar el funcionament dels xats i les seves diferències respecte de la missatgeria instantània.

Xat

Les comunicacions de xat permeten converses de text i àudio entre usuaris i grups d'usuaris en les clàssiques sales de xat que els servidors posen a disposició dels clients. Una sala de xat és un espai virtual on els usuaris poden dialogar entre ells en temps real. En aquestes comunicacions, el protocol usat és l'**IRC** o conversa interactiva per Internet (Internet Relay Chat). Els servidors IRC proporcionen sales virtuals on els usuaris es poden trobar i on poden xatejar usant aplicacions client de xat o el mateix navegador web. Els clients es connecten a un servidor indicant un sobrenom o *nickname*. L'accés pot ser obert a tothom (públic) o reservat a usuaris registrats. Un cop connectats, els clients poden veure quines sales hi ha i entrar en una o crear-ne de noves. Tots els usuaris d'una sala formen un xat, de manera que els missatges que generen es mostren a tots els participants.

Sovint es permet també la possibilitat d'establir una comunicació privada entre dos usuaris, en aquest cas es parla d'un *xat privat*.

En els xats IRC els usuaris s'identifiquen per sobrenoms i es troben en sales on xategen plegats. Tot i que es permet la comunicació vis-a-vis, l'objectiu principal és la comunicació col·lectiva amb els altres usuaris de la sala, "fer-la petar", "xafardejar". Un client es connecta a un servidor i pot comunicar-se amb els altres usuaris identificant-se pel seu sobrenom en les sales on entri. Si uns amics volen fer un xat un divendres a la tarda, han d'acordar prèviament en quin servidor i en quina sala es trobaran. Si volen establir una conversa privada cal que coneguin els sobrenoms dels seus interlocutors.

En els **xats IRC**, els usuaris s'identifiquen per **sobrenoms** i es troben en sales on xategen plegats o en diàlegs a dues bandes. Les converses permeten un cert grau d'anonimat en tant que es realitzen entre usuaris identificats per un sobrenom i no pel nom real.

Un dels problemes d'aquest model és la falta d'identificació dels usuaris, la repetició de *nicknames* i la impossibilitat de saber qui hi ha contactat. Metafòricament, es pot entendre el xat IRC com un mecanisme que permet a un usuari passejar per un edifici i parar-se a parlar amb gent que troba en les diferents sales. El seu propòsit no és la comunicació concreta amb un destinatari concret.

Missatgeria instantània

La missatgeria instantània (IM) permet als usuaris identificar-se com a tals i mantenir converses amb un o més destinataris concrets. A diferència del model de xat IRC, no es tracta d'entrar, per exemple, a la sala "Amics de la sardana" per fer-la petar amb qui hi hagi allà, sinó que es tracta d'establir una conversa en temps real amb un destinatari o destinataris desitjats.

Per poder usar l'IM, els usuaris s'han d'identificar en el servidor; d'aquesta manera es pot consultar quins usuaris hi ha connectats al servei i establir contacte amb un d'ells o més. El servidor proporciona el servei de missatgeria instantània als seus clients, als membres registrats del seu servei, però no a altres clients, si bé es poden establir passarel·les (*gateways*) entre serveis.

El servei de **missatgeria instantània** permet als usuaris registrar-se en el servidor facilitant la identificació dels usuaris connectats i permetent el diàleg directe entre un o més usuaris escollits.

En una primera fase d'implantació de la missatgeria instantània, els proveïdors d'aquest servei utilitzaven protocols propis i privats implantats als seus servidors. Els usuaris estaven obligats a usar les seves aplicacions clients i no es podien connectar amb usuaris d'altres serveis. De fet, no existia un protocol IM com a tal, sinó que cada un anava pel seu cantó. Això va esdevenir un problema evident a mesura que les comunitats de clients van anar creixent i formaven bosses d'usuaris aïllats els uns dels altres. Molts usuaris optaven per disposar de més d'un

compte en proveïdors de missatgeria diferents per tal d'abastar el màxim possible d'interlocutors. L'inconvenient és que per a cada compte calia usar una aplicació client pròpia i, per exemple, s'havien de tenir oberts alhora els diferents clients.

Tres canvis significatius han facilitat la comunicació dels usuaris independitzant-la del servei de missatgeria usat:

- **Clients multiplataforma**: permeten als usuaris connectar-se a diferents serveis sense necessitat d'usar un client específic per a cada servei. L'aplicació client "parla" el llenguatge particular usat com a protocol per cada proveïdor de servei propietari.
- **Passarel·les**: alguns proveïdors de serveis, davant de la necessitat dels seus clients de contactar amb clients d'altres proveïdors, han establert acords de comunicació de manera que els clients d'uns i altres es poden comunicar usant passarel·les (*gateways*) que són transparents (imperceptibles) per a l'usuari. Evidentment, hi ha proveïdors que ofereixen aquest servei als seus usuaris i d'altres més preocupats a mantenir-los captius i no facilitar-los la comunicació amb usuaris d'altres serveis.
- **Jabber** o **XMPP**: és un protocol obert de missatgeria instantània àmpliament utilitzat i s'ha convertit en un estàndard de compatibilitat entre serveis. Quan un proveïdor de servei de missatgeria instantània diu que és Jabber o XMPP significa que permet comunicacions obertes.

Els mecanismes per comunicar usuaris clients de diferents proveïdors de serveis de missatgeria instantània poden residir en l'aplicació client o l'aplicació servidor.

- **Aplicació client**: té la capacitat de "parlar" protocols diferents. S'hi configuren els comptes que connecten amb els comptes a usar.
- **Aplicació servidor**: el proveïdor d'internet proporciona la capacitat de comunicar amb altres servidors de manera transparent per l'usuari.

Protocol XMPP o Jabber

En la secció "Annexos" del web d'aquest mòdul teniu captures dels diferents diàlegs XMPP.

Sovint al protocol de missatgeria instantània, i fins i tot al servei, se l'anomena Jabber. Aquest és el nom que es va donar al protocol desenvolupat el 1999 per proporcionar serveis de missatgeria instantània, informació de presència i manteniment de llistes de contacte. Es tractava d'un protocol obert desenvolupat per la comunitat **Jabber Open Source**. Aquest protocol va esdevenir molt popular en tractar-se d'un protocol obert, i finalment l'IETF el va convertir en un dels seus estàndards amb el nom d'**XMPP**.

El protocol **XMPP** o **Extensible Messaging and Presence Protocol** (protocol de presència i missatgeria extensible) és l'estandardització de l'IETF del protocol Jabber. Es tracta d'un protocol obert basat en XML (*eXtensible Markup Language* o llenguatge de marques extensible) que proporciona serveis de missatgeria instantània i de presència.

Una confusió important en el món de la missatgeria instantània és el terme Jabber, que s'usa tant per descriure el protocol XMPP com el programari de servidor *jabberd* (el dimoni), com una de les organitzacions que presten el servei d'IM a clients de manera gratuïta, Jabber.org.

L'organització que va desenvolupar originalment Jabber s'ha reconvertit amb el nom XMPP Standards Fundation i continua desenvolupant versions del protocol. Per conèixer a fons la seva especificació es poden consultar els RFC 3920 a 3923, emesos inicialment per l'IETF. Actualment s'han refet amb les especificacions RFC 6120, 6121 i 6122; i aquesta darrera ha estat actualitzada per la RFC 7622.

El primer servidor que va implementar el protocol XMPP va ser Jabber.org. En tractar-se d'un protocol obert, altres proveïdors l'han anat implementant i s'ha convertit *de facto* en el protocol estàndard de passarel·la entre proveïdors que utilitzen protocols particulars.

Les seves caractéristiques principals són:

- **Estàndard obert**: és un protocol obert, qualsevol pot utilitzar-lo sense infringir normes de propietat intel·lectual. Això ha permès que s'utilitzi àmpliament.
- **Distribuït**: es tracta d'un protocol descentralitzat. No té un servidor central, sinó que cada organització pot tenir en funcionament el seu propi servidor. Els usuaris d'un servidor es poden comunicar directament entre ells, però també amb usuaris d'altres servidors.
- **Extensible**: el protocol permet definir extensions de manera que es puguin afegir noves funcionalitats al servei. Per exemple, Jingle, que és una extensió compatible amb SIP (*Session Initiation Protocol*) per a veu, vídeo, la transferència de fitxers i altres aplicacions
- **Seguretat**: permet l'establiment de connexions segures usant SSL/TLS, STARTTLS i autenticació segura usant SASL.

El model funcional

Quan es parla d'un servidor en realitat poden ser diverses les màquines que presten el servei dins d'un nom de domini.

El protocol XMPP utilitza una estructura client/servidor descentralitzada. No existeix un servidor únic per a tota la comunitat Jabber, sinó que cada domini pot tenir en funcionament el seu propi servidor. Això contrasta amb altres models d'IM, com per exemple AOL, que té un servidor centralitzat per a tots els usuaris.

La diferència amb l'XMPP és que els usuaris de Jabber d'un servei es poden comunicar amb usuaris de Jabber d'un altre servei sense que existeixi un servidor centralitzat que aglutini tota la informació. Una analogia és dir que els clients d'una companyia telefònica poden trucar sense cap problema a clients d'una altra companyia (no estan limitats a trucar només als clients de la seva mateixa companyia; el preu és un altre tema...), tan sols cal conèixer l'identificador

(número de telèfon) del destinatari. Només cal tenir un compte per poder accedir a tota la comunitat Jabber a internet. Evidentment, cada usuari es pot crear tants comptes com desitgi, igual com es fa amb els comptes de correu. En aquest model, diferents organitzacions no relacionades entre elles poden oferir el servei de Jabber als seus usuaris.

Cada usuari té un compte únic anomenat **JID** o **Jabber ID**, que l'identifica com a usuari en un servidor concret. Aquest JID es vàlid per comunicar-se amb qualsevol altre usuari Jabber, sigui quin sigui el seu servidor. El format de compte JID és user@server.cat.

Exemple d'ús de Jabber

L'empresa X ofereix missatgeria instantània als seus treballadors amb un servidor propi. També l'ofereixen la universitat Y i l'escola Z.

Els usuaris d'aquestes tres entitats es poden comunicar no només entre si, sinó amb tots els altres usuaris de la comunitat Jabber, simplement coneixent l'identificador JID dels destinataris.

La **comunitat Jabber** està formada per tots aquells usuaris que tenen un compte (JID) en servidors compatibles i oberts al protocol XMPP. Això permet a tots aquests usuaris comunicar-se independentment del servidor en el qual tenen el compte.
Les passarel·les o *gateways* permeten comunicar amb usuaris d'altres xarxes d'IM com si també fossin usuaris XMPP.

És a nivell de servidor que s'implementa el transport o *gateway* entre xarxes de missatgeria instantània diferents i amb protocols diferents. En el model Jabber no és l'aplicació client de missatgeria instantània la que "parla" diversos llenguatges o protocols, sinó que és el servidor el que estableix comunicacions amb servidors que implementen protocols diferents. Aquests servidors executen serveis de passarel·la o *gateway* que permeten la comunicació amb els servidors Jabber. És a dir, XMPP proporciona una interfície comuna amb la qual comunicar-se. Això permet que els usuaris es puguin comunicar amb els usuaris d'aquestes altres xarxes com si fossin també usuaris XMPP.

En resum, la comunicació entre usuaris XMPP pot ser:

- **Local**: els usuaris registrats d'un mateix servei Jabber es comuniquen entre ells a través del servidor. Aquests usuaris s'han registrat en aquest servidor. Si, per exemple, es diuen Pere i Anna i el servidor s'anomena ioc.cat, els seus JID són pere@ioc.cat i anna@ioc.cat. Cadascú es connecta al seu servidor i la comunicació es realitza a través del servidor o es pot generar una connexió individual entre ells (per exemple, per a les transferències de fitxers).
- **Altres servidors Jabber**: si, per exemple, pere@ioc.cat vol establir una sessió IM amb l'usuari jordi@edt.cat, el servidor del qual (edt.cat) també

proporciona servei XMPP, la comunicació l'estableix cada usuari amb el seu servidor. Els servidors es comuniquen entre ells per transferir la comunicació, sempre que l'administrador permeti aquests tipus de comunicació.

- **Serveis externs**: per comunicar usuaris que pertanyen a xarxes de missatgeria instantània diferents i que utilitzen protocols de comunicació diferents calen mecanismes de *gateway* entre els servidors. Com sempre, els usuaris inicien sessions en el seu servidor (cada un es connecta al seu) i són els servidors els que es comuniquen entre si realitzant una conversió dels seus protocols al format obert XMPP.

Suport XMPP

Quan va aparèixer el protocol XMPP molts dels serveis de missatgeria instantània del moment van veure una oportunitat pels avantatges que oferia un protocol obert i van començar a donar suport a XMPP, és a dir, els seus clients de missatgeria instantània van començar a ser compatibles amb aquest protocol.

Destaquen els següents:

- Google Talk
- AIM (AOL Instant Messenger)
- Yahoo! Messenger
- Skype
- Facebook

Fins i tot, Whatsapp utilitza una versió modificada del protocol XMPP, almenys quan va ser comprada per Facebook.

No obstant, per raons diverses, totes aquestes grans empreses han deixat de donar servei amb aquest protocol. Google ho va fer quan va passar de Google Talk a Google Hangouts; AOL (America Online) ha donat suport limitat a XMPP fins al 2017; Yahoo! Messenger es va substituir al 2018 per un nou servei anomenat Yahoo Together, que no va arribar a l'any de vida; Microsoft manté un suport molt limitat, i Facebook va deixar de donar-hi suport al 2014.

A part de raons comercials i estratègiques, també cal comentar alguna raó tècnica: XMPP està molt centrat en la missatgeria instantània de text, tot i que a través d'extensions permet afegir serveis tals com trucades de veu i videotrucades.

2.1.2 Clients de missatgeria

Existeixen clients de missatgeria instantània per a tots els gustos i de tots els colors. Inicialment, el proveïdors de serveis requerien la seva pròpia aplicació client,

per tant, cada proveïdor disposava d'un client diferent. Posteriorment han anat apareixent clients multiplataforma que permeten connectar amb serveis diferents "parlant" el protocol apropiat en cada cas.

Clients XMPP

Llista actualitzada de clients XMPP:

xmpp.org/software/clients.html

Les distribucions de GNU/Linux acostumen a incorporar clients gràfics multiplataforma. Els més coneguts són:

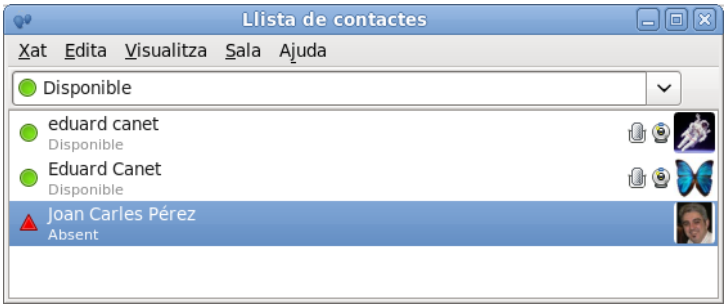
- Empathy
- Pidgin
- Gossip
- Psi
- Gajim

Empathy

Un client de missatgeria instantània actualment popular en sistemes GNU/Linux és Empathy, que permet a un mateix usuari crear múltiples comptes de missatgeria de diferents tipus. Empaty és un exemple de la capacitat multiplataforma que tenen la majoria de clients IM, ja que és capaç d'establir connexions de missatgeria instantània amb multitud de serveis que utilitzen protocols diferents.

La figura 2.2 mostra l'estat i la llista de contactes d'un usuari determinat.

FIGURA 2.2. Client de missatgeria Empathy



Les funcionalitats principals d'aquest client són:

- Veure l'estat del compte.
- Configurar un compte.
- Llistar de contactes.
- Crear grups de contactes.
- Unir-se a sales.
- Gestionar les sales preferides.
- Realitzar xats.

Estat del compte

L'usuari pot indicar quin és el seu estat o *status* en el menú desplegable. D'aquesta manera determina quina és la visibilitat que vol tenir. Pot escollir entre estar visible i disponible, no disponible o fins i tot no visible.

Els estats possibles són:

- **Disponible**: els altres usuaris poden contactar amb aquest usuari.
- **Ocupat**: l'usuari és visible per als altres usuaris, però no hi poden contactar perquè està ocupat.
- **Amagat**: els altres usuaris no poden saber quin és el seu estat, no saben si està connectat o no.
- **Absent**: l'usuari disponible amb un temps d'inactivitat en el sistema passa a estar absent (igual que passa amb l'ordinador quan entra en mode d'estalvi d'energia).
- **Desconnectat**: l'usuari no està connectat al sistema de missatgeria.
- **Missatge personalitzat**: es pot personalitzar el missatge que es vol mostrar en cada un dels estats, de manera que en lloc de veure paraula predeterminada els altres usuaris veuen un missatge personalitzat. Així, per exemple, algú molt optimista pot personalitzar l'estat *Disponible* per *A punt per a tot!*, algú altre molt enfeinat pot personalitzar l'estat *Ocupat* per un missatge com *De bòlit...*

Llista de contactes

La finestra principal que es mostra en obrir Empathy és la llista de contactes de l'usuari, com es pot observar en la figura 2.2. Aquesta llista es pot editar afegint-hi i eliminant-ne contactes. Per eliminar-ne simplement cal seleccionar *suprimir*. Per afegir nous contactes cal indicar l'identificador, el JID del contacte a afegir. Es poden observar les dades de cada un dels contactes configurats seleccionant l'opció *informació*.

L'aspecte del llistat dels contactes es pot configurar amb el menú *Visualitza*, que permet indicar si el llistat és de tipus compacte, normal o amb icones, si està ordenat per nom o estat i si es mostren o no els contactes desconnectats.

La figura 2.3 mostra les dades del contacte d'un usuari.

FIGURA 2.3. Informació d'un contacte

Configuració del compte

L'element més important en la utilització d'un client de missatgeria instantània és configurar correctament el compte de l'usuari. Antigament, els usuaris utilitzaven un client propi per a cada servidor amb el qual volien connectar. Un pas posterior va ser usar clients multiprotocol (com Empathy) per poder connectar amb servidors i protocols diferents. Actualment, molts dels serveis són compatibles amb Jabber, de manera que amb un sol compte Jabber es pot accedir als contactes, sigui quin sigui el seu proveïdor de servei.

L'usuari pot crear tants **comptes de missatgeria instantània** associats al seu usuari com desitgi. Per a cada compte que crea es connecta amb el servidor pertinent i n'obté la **llista de contactes**. Un compte permet accedir a tots els contactes d'aquell servidor i de tots els altres servidors amb acords o compatibles amb **XMPP**.

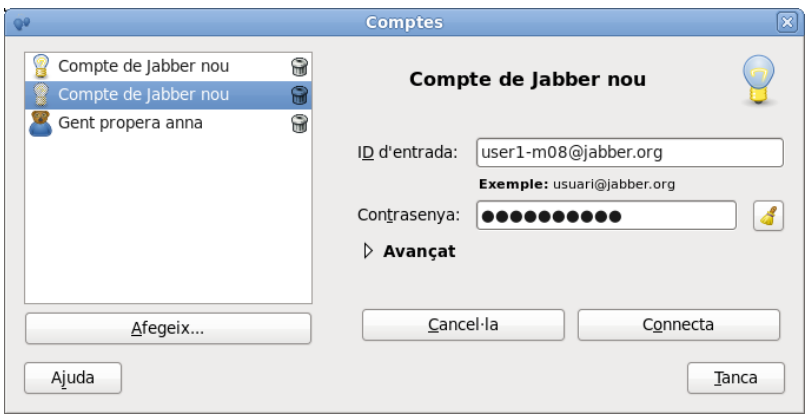
Us recordem que no cal crear un compte per a cada servidor si aquests serveis tenen entre ells un acord de connectivitat o accepten l'accés extern via XMPP.

La configuració de comptes permet:

- **Usar un compte ja existent**: si l'usuari disposa ja de comptes de missatgeria en proveïdors públics pot configurar directament el compte indicant el seu JID i les dades que siguin necessàries per accedir al servidor. Sovint n'hi ha prou amb el JID, però es pot afegir informació addicional com qüestions de seguretat, un port o nom de servidor diferent...
- **Crear un compte nou**: cal seleccionar en el menú desplegable quin tipus de compte es vol crear. Empathy permet crear comptes nous en determinats servidors de missatgeria instantània.

La figura 2.4 mostra el procés de creació d'un compte de Jabber al servidor Jabber.org. És tan senzill com indicar un JID vàlid i la contrasenya que es vol usar. Recordeu que cal fer clic a *Connecta* per crear el compte. Assegureu-vos també de seleccionar l'opció *habilitat* per poder-ne fer ús.

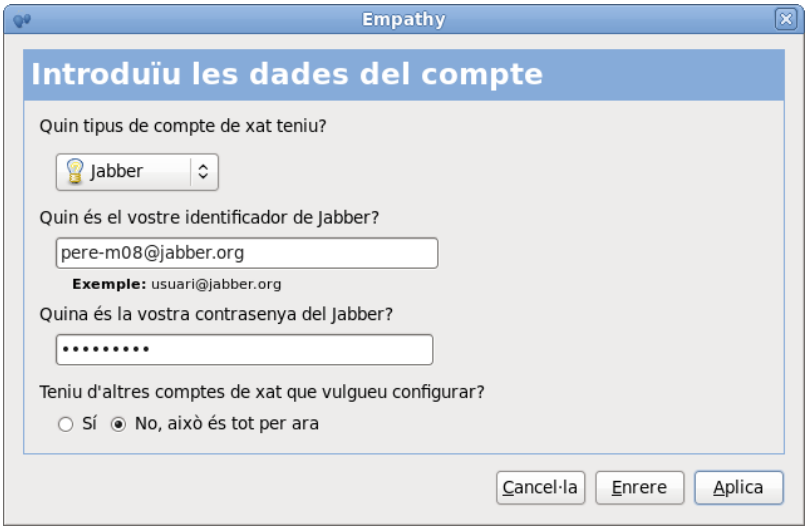
Figura 2.4. Creació d'un compte a Jabber.org



Quan un usuari posa en marxa per primera vegada el client Empathy de missatgeria instantània, se li permet configurar un o més comptes, ja siguin nous o preexistents

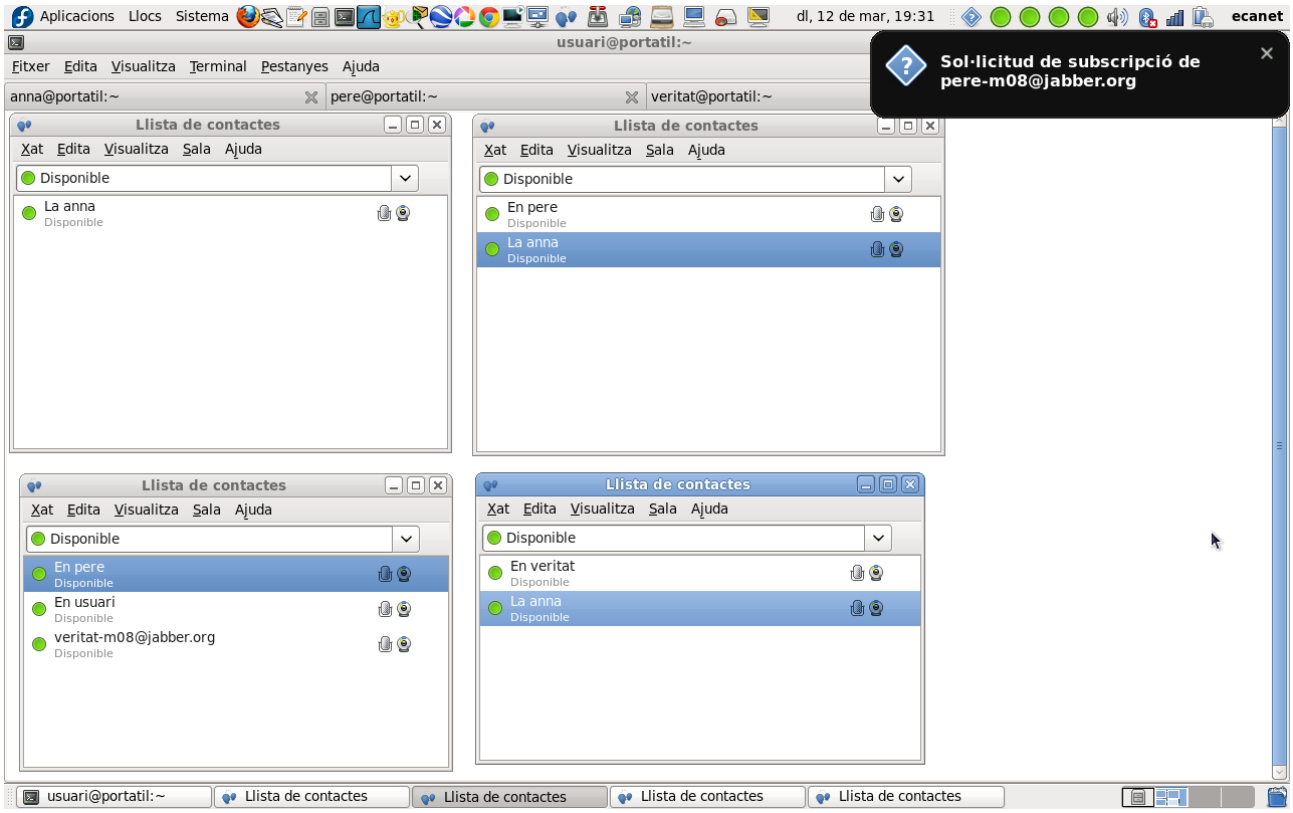
en proveïdors d'IM. La figura 2.5 mostra el procés d'utilització d'un compte Jabber ja existent iniciat quan l'usuari entra al programa per primer cop.

Figura 2.5. Assistent de configuració inicial



La figura 2.6 mostra quatre aplicacions client Empathy obertes des de sessions de *bash* per als usuaris "anna", "pere", "veritat" i "usuari". Es pot observar a la part superior dreta de notificació dels quatre indicadors verds d'Empathy, un per a cada client. També s'observa el missatge de notificació que rep l'usuari "usuari" informant-lo que pere l'ha posat a la seva llista de contactes i demanant-li si vol fer el mateix (incorporar "pere" a la seva llista).

Figura 2.6. Pantalla amb quatre usuaris



Xat

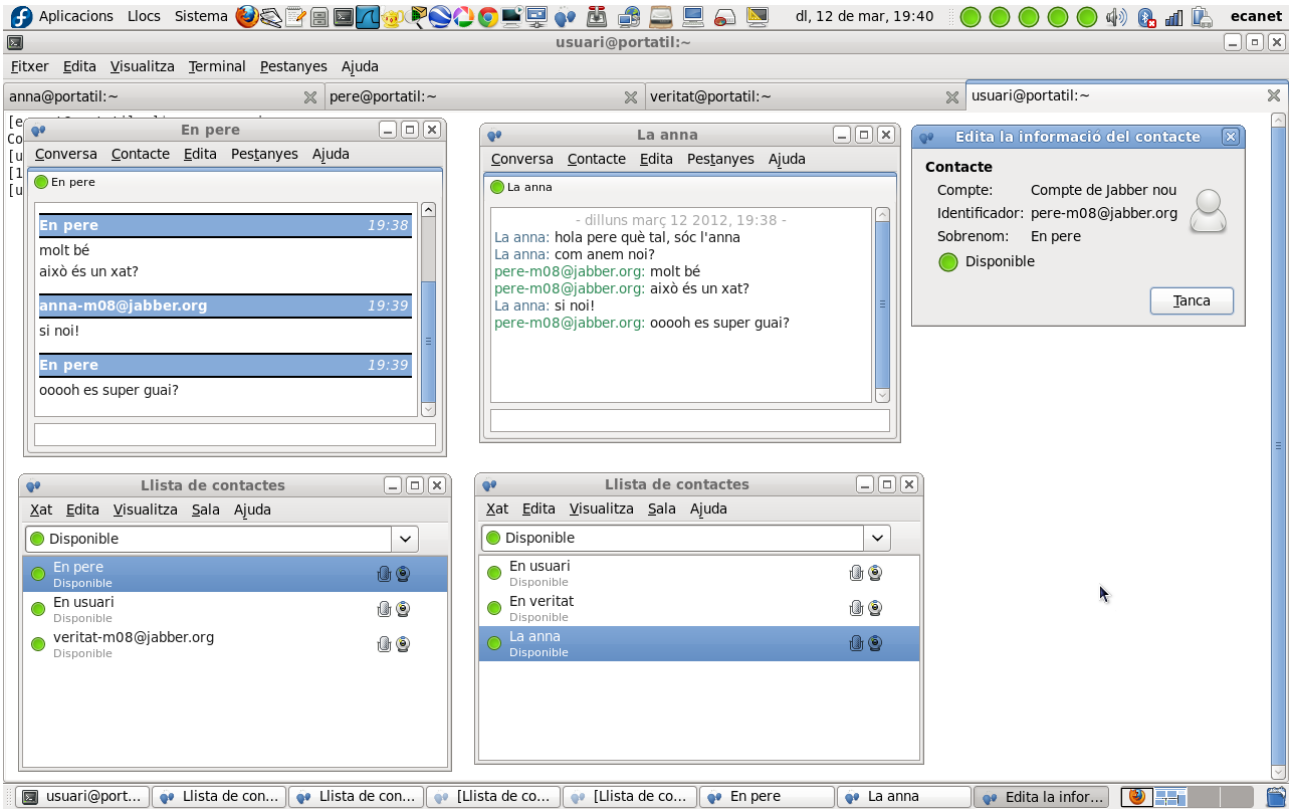
Evidentment, la principal funció d'un client de missatgeria instantània és permetre l'usuari parlar amb els altres contactes, xatejar. Els xats es poden fer seleccionant un per un els contactes amb qui parlar o seleccionant directament un grup de contactes.

Els xats poden ser de:

- Text
- Àudio
- Vídeo
- Transferència de fitxers
- Compartir l'escriptori

La figura 2.7 mostra el diàleg de text entre dos usuaris. S'ha seleccionat també l'opció que mostra la informació del contacte "pere".

FIGURA 2.7. Xat entre anna i pere



Un usuari pot obrir múltiples xats. Per a cada xat s'obre una pestanya nova en la qual es produeix la conversa. La figura 2.8 mostra com la finestra de xat de la usuària "anna" té dues pestanyes: en una dialoga amb "veritat-m08" i en l'altra amb "usuari".

FIGURA 2.8. L'Anna manté dues converses

2.2 Llistes de distribució

Una llista de distribució és una llista de membres (usuaris, clients, empleats, simpatitzants, socis, adeptes, col·legues, amics de classe, de la "mili", jugadors de futbol, afiliats a Metges Sense Fronteres...) que reben missatges sobre un tema determinat. Algunes llistes només permeten als inscrits rebre missatges i d'altres els permeten també participar-hi enviant missatges.

Així, per exemple, quan els clients d'un supermercat es fan la targeta client esdevenen membres d'una llista del supermercat, que els envia periòdicament publicitat a casa o per correu electrònic. Si et fas soci del zoo, també reps periòdicament a casa informació d'aquesta institució, possiblement una revista trimestral i correus electrònics amb novetats del zoo i notícies sobre fauna.

Hi ha altres possibles llistes de distribució, com per exemple "Alumnes d'M08", "Kernel Linux" o "El festeig de l'elefant marí en l'època de zel", els membres de les quals poden participar-hi generant missatges que s'envien a la resta de destinataris de la llista.

Una **llista de correu** (o *email mailing list*) és una llista d'adreces de correu dels **subscriptors** que reben els mateixos missatges. Les llistes poden ser només de **publicació** (*announcement list*) o de **discussió** (*discussion list*).

Un mecanisme per crear llistes electròniques o *elists* són els àlies de correus o àlies de llistes d'adreces, que permeten els clients de correu com, per exemple, Thunderbird. Un àlies és un nom identificatiu assignat a un compte de correu o a un conjunt com a nom de grup. El pas següent és organitzar aquesta llista. Per fer-ho un dels membres fa la tasca de gestor o administrador, decideix qui en forma part i qui no, quins missatges s'hi poden enviar i quins no. Per facilitar la gestió de les llistes es van dissenyar programes que n'automatitzaven el funcionament. Aquests permeten gestionar la publicació i la distribució de

missatges, la subscripció i baixa i altres aspectes del seu funcionament. Això es fa amb missatges a la llista que contenen ordres adreçades al programari.

És típic que en el peu dels missatges d'una llista de distribució s'inclogui un text que expliqui com donar-se d'alta o de baixa. Són els típics missatges com "Escriu un missatge a aquesta adreça i passaràs a ser membre de la llista" o "Respon a aquesta adreça per deixar de formar part de la llista". Actualment, les aplicacions que gestionen llistes de correu acostumen a incorporar un frontal web per a l'administració de les llistes.

Els grups de debat, discussions o fòrums poden realitzar-se en formats diferents (llistes de distribució, *news*, web...) i permeten als subscriptors la publicació de missatges. Usualment, les discussions s'emmagatzemen en servidors, s'**arxiven**. Els servidors n'indexen el contingut i en permeten la recerca posterior.

Els debats que es produeixen en les llistes de discussió normalment s'emmagatzemen o **arxiven** permanentment a internet en servidors que n'organitzen i indexen els continguts. És a dir, la majoria de les discussions, debats i missatges que publiquen els subscriptors de les llistes es conserven. És per això que quan busquem al Google "problema instal·lar sendmail en Debian" ens apareixen múltiples converses d'usuaris en fòrums de debat.

Es pot configurar un grup de discussió per rebre les **publicacions individuals**, una a una, o agrupades per períodes de temps o quantitat. Per exemple, es pot demanar un resum diari o un resum de cada 10 publicacions. En aquest cas es diu que l'usuari rep un **resum o *digest***.

2.2.1 Creació d'un gestor de llistes

Existeixen diversos mecanismes per crear llistes de distribució, des de programari especialitzat fins als àlies de correu.

Els principals mecanismes per implementar llistes són:

- Àlies
- Grups de correu de serveis d'Internet: Google Groups
- Àlies de Sendmail
- Servidor de llistes

Si una entitat vol gestionar una o més llistes de distribució sense externalitzar-les i d'una manera més avançada i eficient que els simples àlies, ha d'usar algun dels

paquets de programari que fan de servidor de llistes de distribució. Existeixen diverses aplicacions que fan aquesta tasca:

- GNU Mailman: és un programari de codi lliure realitzat bàsicament amb el llenguatge de programació Python. Ha estat elaborat per programadors de GNU.
- Majordomo: és un programari que va ser àmpliament utilitzat, però que actualment no té tant seguiment.
- Listserv: és el pare de tots els programes de llistes de distribució. El seu creador va ser pioner en el desenvolupament d'aquest tipus de programes. Actualment no és de llicència pública.

El procés per instal·lar el programa que gestionarà el servidor de llistes de distribució és similar al procés seguit per a la instal·lació de la majoria de serveis del sistema. En resum, cal:

- Identificar els paquets de programari que contenen l'aplicació, descarregar-los i instal·lar-los. Es poden localitzar en els repositoris de paquets apropiats segons la distribució de GNU/Linux que s'estigui utilitzant o es pot descarregar el fitxer .tar original.
- Un cop instal·lat, cal determinar l'estat que ha de tenir el servei en cada *runlevel* (nivell d'execució) del sistema. És a dir, cal automatitzar si ha d'estar engegat o parat per defecte en cada un d'ells. Si el servei es deixa apagat per defecte, evidentment caldrà posar-lo en funcionament manualment. En el cas de GNU Mailman es tracta d'un servei autònom (*stand-alone*).
- Identificar els components de l'aplicació instal·lada. S'ha de saber determinar quins són els fitxers executables, quins els de documentació i quins els de configuració.
- Identificar el PID del servei.
- Identificar i monitorar els fitxers de registre (*log*) del servei.

El programa GNU Mailman s'acompanya d'una extensa documentació en format PDF i web disponible a /usr/share/doc/mailman. En concret, convé que examineu el contingut de:

- Guia d'instal·lació
- Guia d'administració de llistes
- Guia d'usuari membre d'una llista

Per a la instal·lació de Mailman, primer cal haver instal·lat com a prerequisits un servidor de correu i un servidor web. En aquest cas s'han fet servir **Postfix** i

Apache. S'ha d'habilitar també el mòdul **CGI** (Common Gateway Interface) per a Apache. Aquest mòdul permet al servidor web executar programes del servidor com si fossin aplicacions web i obtenir pàgines de manera dinàmica.

Per habilitar el mòdul i reiniciar el servei:

```
root@server:~# a2enmod cgi
Your MPM seems to be threaded. Selecting cgid instead of cgi.
Enabling module cgid.
To activate the new configuration, you need to run:
  systemctl restart apache2

root@server:~# service apache2 restart
root@server:~# service apache2 status•
 apache2.service – The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
        enabled)
   Active: active (running) since Mon 2019–12–02 16:30:09 CET; 4s ago
  Process: 4663 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/
      SUCCESS)
  Process: 4669 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/
      SUCCESS)
 Main PID: 4674 (apache2)
    Tasks: 56 (limit: 4915)
   CGroup: /system.slice/apache2.service
           4674 /usr/sbin/apache2 –k start
           4675 /usr/sbin/apache2 –k start
           4676 /usr/sbin/apache2 –k start
           4677 /usr/sbin/apache2 –k start

des 02 16:30:09 server.ioc.cat systemd[1]: Stopped The Apache HTTP Server.
des 02 16:30:09 server.ioc.cat systemd[1]: Starting The Apache HTTP Server...
des 02 16:30:09 server.ioc.cat systemd[1]: Started The Apache HTTP Server.
```

Després s'instal·la el programa de gestió de llistes de distribució GNU Mailman:

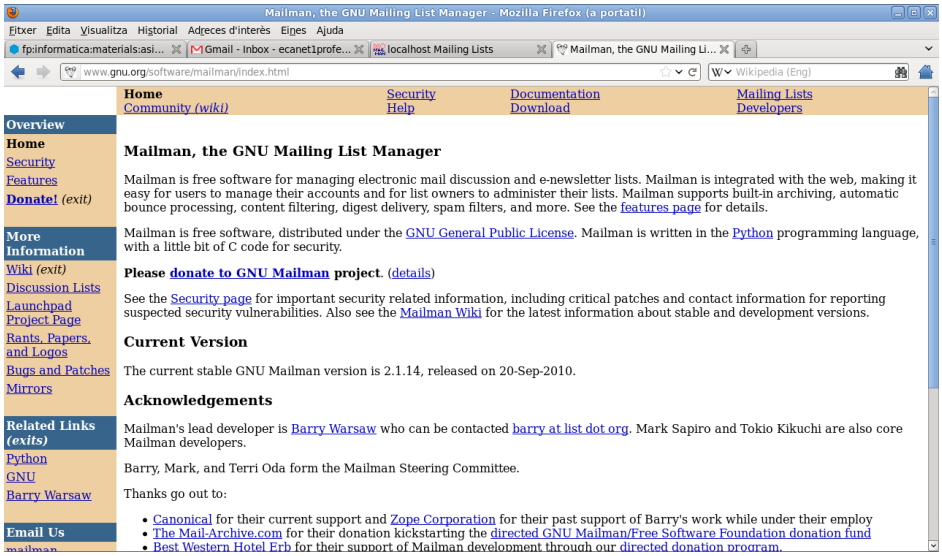
```
root@server:~# apt–get install mailman
S'està llegint la llista de ...paquets Fet
S'està construint l'arbre de dependències
S'està llegint la informació de l'...estat Fet
S'instal·laran els següents paquets extres:
  python–dnspython
Paquets suggerits:
  spamassassin lynx listadmin
S'instal·laran els paquets NOUS següents:
  mailman python–dnspython
0 actualitzats, 2 nous a instal·lar, 0 a suprimir i 337 no actualitzats.
S'ha d'obtenir 4568 kB d'arxius.
Després d'aquesta operació s'empraran 39,8 MB d'espai en disc addicional.
Voleu continuar? [S/n]
```

Observeu que el servei no arranca bé, ja que falta configurar-lo:

```
root@server:~# service mailman status•
 mailman.service – LSB: Mailman Master Queue Runner
   Loaded: loaded (/etc/init.d/mailman; generated; vendor preset: enabled)
   Active: active (exited) since Mon 2019–12–02 16:33:51 CET; 25s ago
     Docs: man:systemd–sysv–generator(8)

des 02 16:33:51 server.ioc.cat systemd[1]: Starting LSB: Mailman Master Queue
    Runner...
des 02 16:33:51 server.ioc.cat mailman[7555]: Site list for mailman missing (
    looking for list named 'mailman'). ... (warning).
des 02 16:33:51 server.ioc.cat mailman[7555]: Please create it; until then,
    mailman will refuse to start. ... (warning).
des 02 16:33:51 server.ioc.cat systemd[1]: Started LSB: Mailman Master Queue
    Runner.
```

La figura 2.9 mostra la pàgina principal del web de GNU Mailman, on es pot trobar tota la documentació i versions de l'aplicació.

FIGURA 2.9. Pàgina principal del web de GNU Mailman



Arrancada del servidor Mailman

Un cop realitzat el procés d'instal·lació cal assegurar-se que tots els components estan configurats apropiadament i realitzar petites tasques de configuració abans de poder posar en marxa el servei i utilitzar les llistes de distribució.

S'aconsella realitzar els passos següents:

1. Comprovar usuaris i grups creats.
2. Identificar el directori base.
3. Comprovar els permisos de fitxers i directoris.
4. Comprovar la configuració del Mailman en el servidor web (en aquest cas, Apache).
5. Reiniciar el servei d'Apache.
6. Crear la llista principal o per defecte.
7. Assignar la contrasenya d'administració del servei.
8. Engegar el servei.
9. Verificar l'accés via web a l'administració de les llistes.

1. Es crea un usuari i un grup anomenats *list* amb els quals s'executa el servei.

```
root@server:~# grep "list" /etc/passwd
mailman:x:38:38:GNU Mailing List Manager:/usr/lib/mailman:/sbin/nologin

root@server:~# grep "list" /etc/group
mailman:x:38:
```

2. El directori base que conté l'aplicació és:

```
/usr/lib/mailman
```

Dins d'aquest directori hi ha una estructura de subdirectoris amb tots els elements necessaris per al seu funcionament, tant la part d'aplicació web com la configuració o els executables en la línia d'ordres.

Aquesta és l'estructura de directoris que neix del directori base (es mostra només un nivell de profunditat):

```
root@server:~# tree -L 1 /usr/lib/mailman/
/usr/lib/mailman/
├── bin
├── cron
├── mail
├── Mailman
└── scripts

5 directories, 0 files
```

3. El següent pas és comprovar els permisos dels fitxers i directoris:

```
root@server:~# /usr/lib/mailman/bin/check_perms
```

Si sorgeixen problemes es pot seguir la indicació i usar l'opció *-f* per intentar solucionar-los.

```
root@server:~# /usr/lib/mailman/bin/check_perms -f
Problemes trobats: 14
Re-executa com mailman (o root) amb la senyal -f per a fixar
```

setgid (set group ID)

Flag d'accés d'un fitxer que dona permís a l'usuari a executar un executable amb els permisos d'execució del grup al qual pertany el fitxer. L'equivalent per a l'usuari és el **setuid** (set user ID).

No obstant, aquesta opció no sempre acaba resolent tots els problemes. En aquest cas hi ha problemes de fitxers que no pertanyen al grup corresponent i no tenen activat el setgid:

```
root@server:~# chgrp list /usr/lib/cgi-bin/* -R
root@server:~# chgrp list /var/lib/mailman//* -R
root@server:~# chmod g+s /var/lib/mailman/cgi-bin/* -R
root@server:~# /usr/lib/mailman/bin/check_perms
No s'han trobat problemes
```

4. Cal enllaçar GNU Mailman amb Apache. A /etc/mailman/apache.conf hi ha un exemple de configuració. La part que no està comentada s'afegeix a /etc/apache2/apache2.conf just després del darrer <Directory>.

```
...
ScriptAlias /cgi-bin/mailman/ /usr/lib/cgi-bin/mailman/
Alias /pipermail/ /var/lib/mailman/archives/public/
Alias /images/mailman/ /usr/share/images/mailman/

```

```
<Directory /usr/lib/cgi-bin/mailman/>
    AllowOverride None
    Options ExecCGI
    AddHandler cgi-script .cgi
    Require all granted
</Directory>
<Directory /var/lib/mailman/archives/public/>
    Options FollowSymlinks
    AllowOverride None
    Require all granted
</Directory>
<Directory /usr/share/images/mailman/>
    AllowOverride None
    Require all granted
</Directory>
...
```

5. Cal reiniciar el servei web perquè tinguin efecte els canvis:

```
root@server:~# service apache2 restart
root@server:~# service apache2 status•
 apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
        enabled)
   Active: active (running) since Mon 2019-12-02 16:43:25 CET; 21ms ago
  Process: 7834 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/
       SUCCESS)
  Process: 7840 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/
       SUCCESS)
 Main PID: 7846 (apache2)
    Tasks: 56 (limit: 4915)
   CGroup: /system.slice/apache2.service
           7846 /usr/sbin/apache2 -k start
           7849 /usr/sbin/apache2 -k start
           7850 /usr/sbin/apache2 -k start
           7852 /usr/sbin/apache2 -k start

des 02 16:43:25 server.ioc.cat systemd[1]: Stopped The Apache HTTP Server.
des 02 16:43:25 server.ioc.cat systemd[1]: Starting The Apache HTTP Server...
des 02 16:43:25 server.ioc.cat systemd[1]: Started The Apache HTTP Server.
```

6. Cal crear una llista principal per al funcionament del servei. La llista s'anomena usualment *mailman*.

7. Cal crear la contrasenya de l'administrador del servei (el *root* del servei).

```
root@server:~# newlist mailman
Introduïu l'adreça electrònica de l'encarregat de la llista: admin@server.ioc.
    cat
Contrasenya inicial de mailman: peremailman
Haureu d'editar el fitxer /etc/aliases (o equivalent) per a finalitzar la
creació de la vostra llista de correu. Hi haureu d'afegir les línies
següents i possiblement executar el programa «»newaliases:

## Llistes de correu mailman
mailman:              "|/usr/lib/mailman/mail/mailman post mailman"
mailman-admin:        "|/usr/lib/mailman/mail/mailman admin mailman"
mailman-bounces:      "|/usr/lib/mailman/mail/mailman bounces mailman"
mailman-confirm:      "|/usr/lib/mailman/mail/mailman confirm mailman"
mailman-join:         "|/usr/lib/mailman/mail/mailman join mailman"
mailman-leave:        "|/usr/lib/mailman/mail/mailman leave mailman"
mailman-owner:        "|/usr/lib/mailman/mail/mailman owner mailman"
mailman-request:      "|/usr/lib/mailman/mail/mailman request mailman"
mailman-subscribe:    "|/usr/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe:  "|/usr/lib/mailman/mail/mailman unsubscribe mailman"

Premeu la tecla de retorn per a notificar el propietari de mailman...
```

S'ha creat automàticament un conjunt de llistes. La finalitat de cada una d'elles es descriurà posteriorment. L'administrador del servei serà l'usuari admin i la contrasenya d'administració que s'ha establert és admin (caldria posar una contrasenya més segura).

8. Arribats a aquest punt, el servei ja es pot posar en funcionament:

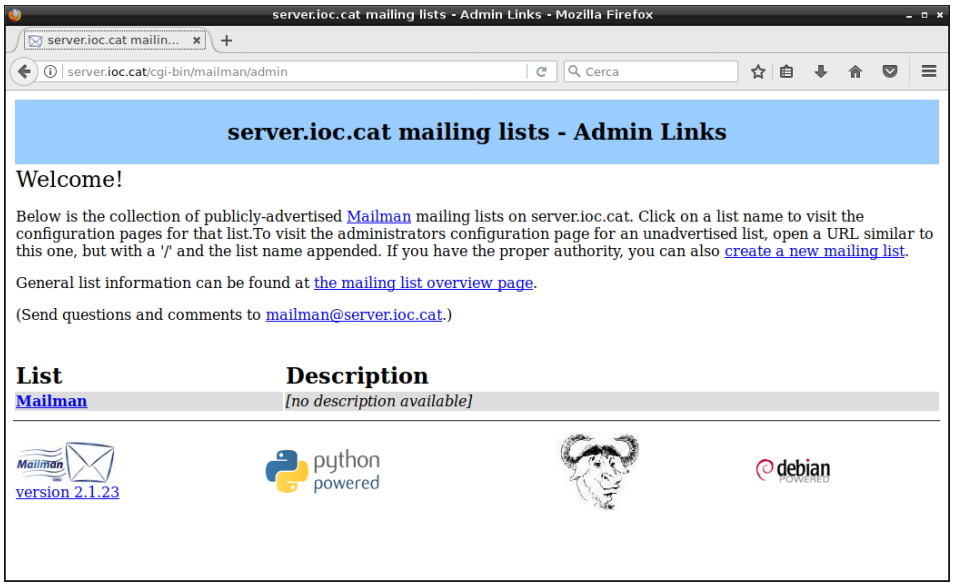
```
root@server:~# service mailman restart
root@server:~# service mailman status•
 mailman.service – LSB: Mailman Master Queue Runner
   Loaded: loaded (/etc/init.d/mailman; generated; vendor preset: enabled)
   Active: active (running) since Mon 2019–12–02 16:43:40 CET; 72ms ago
     Docs: man:systemd–sysv–generator(8)
  Process: 7917 ExecStop=/etc/init.d/mailman stop (code=exited, status=0/
      SUCCESS)
  Process: 7922 ExecStart=/etc/init.d/mailman start (code=exited, status=0/
      SUCCESS)
    Tasks: 9 (limit: 4915)
   CGroup: /system.slice/mailman.service
           7930 /usr/bin/python /usr/lib/mailman/bin/mailmanctl –s –q start
           7931 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               ArchRunner:0:1 –s
           7932 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               BounceRunner:0:1 –s
           7933 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               CommandRunner:0:1 –s
           7934 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               IncomingRunner:0:1 –s
           7935 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               NewsRunner:0:1 –s
           7936 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               OutgoingRunner:0:1 –s
           7937 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               VirginRunner:0:1 –s
           7938 /usr/bin/python /var/lib/mailman/bin/qrunner ––runner=
               RetryRunner:0:1 –s

des 02 16:43:40 server.ioc.cat systemd[1]: Stopped LSB: Mailman Master Queue
     Runner.
des 02 16:43:40 server.ioc.cat systemd[1]: Starting LSB: Mailman Master Queue
     Runner...
des 02 16:43:40 server.ioc.cat mailman[7922]: Starting Mailman master qrunner:
     mailmanctl.
des 02 16:43:40 server.ioc.cat systemd[1]: Started LSB: Mailman Master Queue
     Runner.
```

9. Vegeu a la figura 2.10 l'accés al lloc web local d'administració del servei Mailman. La ruta és:

```
http://server.ioc.cat/cgi–bin/mailman/admin
```

En aquesta pàgina es poden crear noves llistes i administrar el servei.

FIGURA 2.10. Web d'administració de Mailman



2.2.2 Creació i utilització de llistes

Hi ha diversos passos necessaris per a la creació i administració de llistes de distribució amb el programa d'entorn web de gestió llistes Mailman. Es poden dur a terme els processos següents:

- Creació d'una llista
- Subscripció a la llista
- Utilització de la llista

Exemple de creació d'una llista

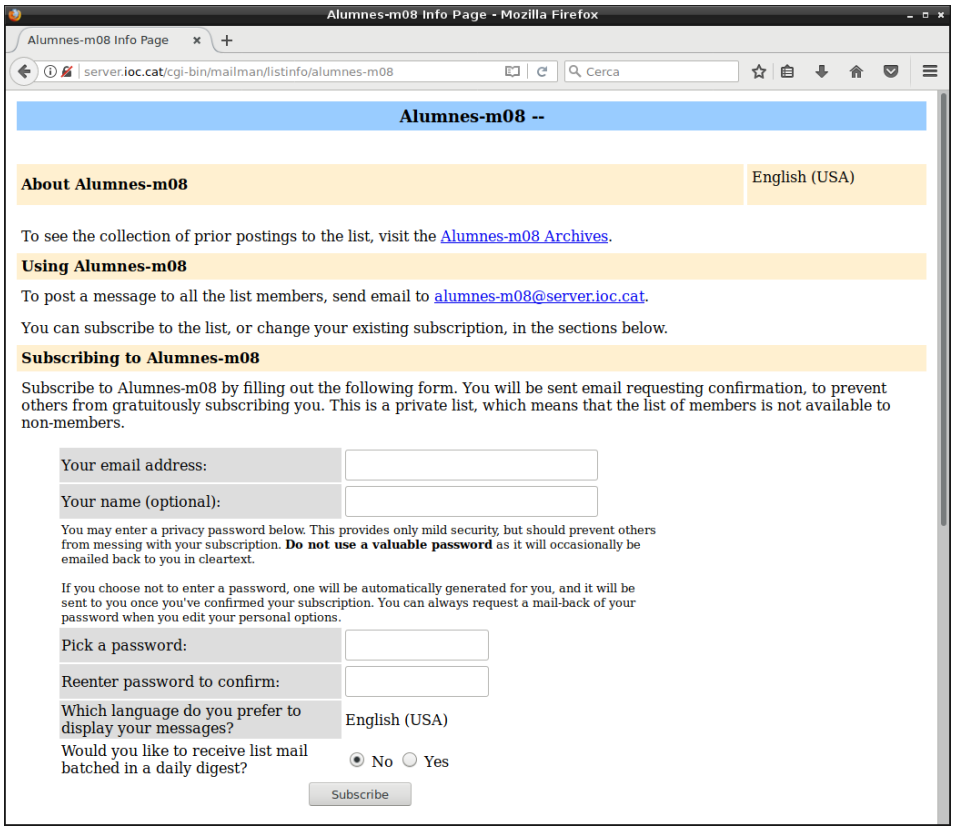
En aquest primer exemple es crearà una llista anomenada "alumnes-m08":

```
root@server:~# newlist alumnes-m08
Introduïu l'adreça electrònica de l'encarregat de la llista: jciberta@server.
    ioc.cat
Contrasenya inicial de alumnes-m08:
Haureu d'editar el fitxer /etc/aliases (o equivalent) per finalitzar la
creació de la vostra llista de correu. Hi haureu d'afegir les línies
següents i possiblement executar el programa «»newaliases:

## Llista de correu alumnes-m08
alumnes-m08:              "|/var/lib/mailman/mail/mailman post alumnes-m08"
alumnes-m08-admin:        "|/var/lib/mailman/mail/mailman admin alumnes-m08"
alumnes-m08-bounces:      "|/var/lib/mailman/mail/mailman bounces alumnes-m08"
alumnes-m08-confirm:      "|/var/lib/mailman/mail/mailman confirm alumnes-m08"
alumnes-m08-join:         "|/var/lib/mailman/mail/mailman join alumnes-m08"
alumnes-m08-leave:        "|/var/lib/mailman/mail/mailman leave alumnes-m08"
alumnes-m08-owner:        "|/var/lib/mailman/mail/mailman owner alumnes-m08"
alumnes-m08-request:      "|/var/lib/mailman/mail/mailman request alumnes-m08"
alumnes-m08-subscribe:    "|/var/lib/mailman/mail/mailman subscribe alumnes-m08
    "
alumnes-m08-unsubscribe:  "|/var/lib/mailman/mail/mailman unsubscribe alumnes-
    m08"

```

```
20 Premeu la tecla de retorn per notificar el propietari de alumnes-m08...
```

Un cop creada la llista es pot accedir via web a la informació de la pàgina. La figura 2.11 mostra part de la pàgina d'informació de la llista "alumnes-m08". La URL és http://server.ioc.cat/cgi-bin/mailman/listinfo/alumnes-m08.

FIGURA 2.11. Pàgina d'informació de la llista alumnes-m08

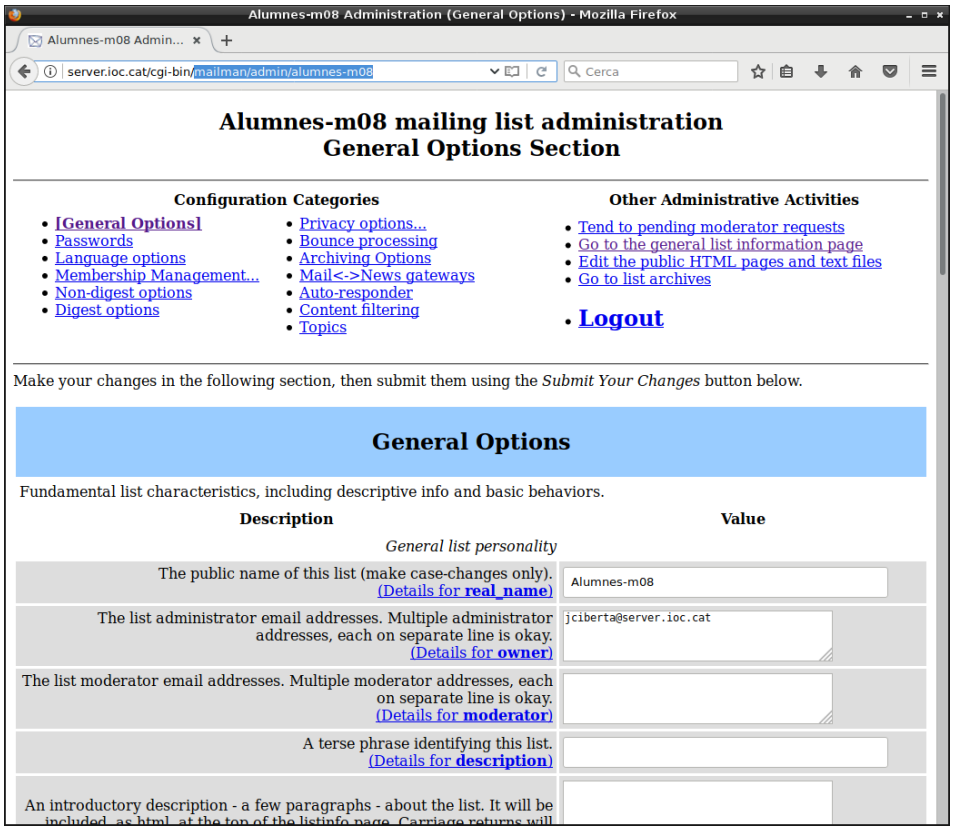


Aquesta pàgina està dividida en quatre parts:

- *About* permet accedir a la llista de missatges (*posts*) publicats en la llista.
- *Using* mostra l'adreça de correu que cal per publicar. És a dir, els subscriptors de la llista han d'enviar els seus missatges a l'adreça indicada, que en aquest cas és alumnes-m08@localhost.localdomain.
- *Subscribing* permet fer-se subscriptor de la llista. Per fer-ho cal indicar una adreça de correu i una contrasenya. Als subscriptors se'ls permet seleccionar l'idioma i decidir si volen rebre els missatges un a un o en forma de *digest* (agrupats per períodes de temps o nombre de missatges).
- *Subscribers* són opcions només vàlides per a subscriptors. Permeten visualitzar la llista de subscriptors, anul·lar la subscripció o modificar-ne les propietats.

L'administrador de la llista i l'administrador global poden accedir a la pàgina d'administració de la llista i configurar-ne nombrosos aspectes de funcionament. La figura 2.12 mostra algunes de les opcions de configuració.

FIGURA 2.12. Administració de la llista alumnes-m08



Observeu com els respectius administradors de les diferents llistes han rebut un correu indicant la creació de la llista:

```
root@server:~# su admin
admin@server:/root$ mail
"/var/mail/admin": 1 message 1 new
>N   1 mailman-owner@serv dl des  2 16:43   46/2633  Your new mailing list:
    mailman
? q
Held 1 message in /var/mail/admin
admin@server:/root$ su jciberta
Contrasenya:
jciberta@server:/root$ mail
"/var/mail/jciberta": 1 message 1 new
>N   1 mailman-owner@serv dt des  3 15:28   46/2702  Your new mailing list:
    alumnes-m08
? q
Held 1 message in /var/mail/jciberta
jciberta@server:/root$
```

Subscripció a la llista

Els usuaris es poden subscriure a la llista accedint al web de cada llista, per correu electrònic (a l'adreça de subscripció) o si l'administrador de la llista els inscriu.

El procés consta dels passos següents:

1. Consulta les llistes del lloc.
2. Fa la petició de subscripció.

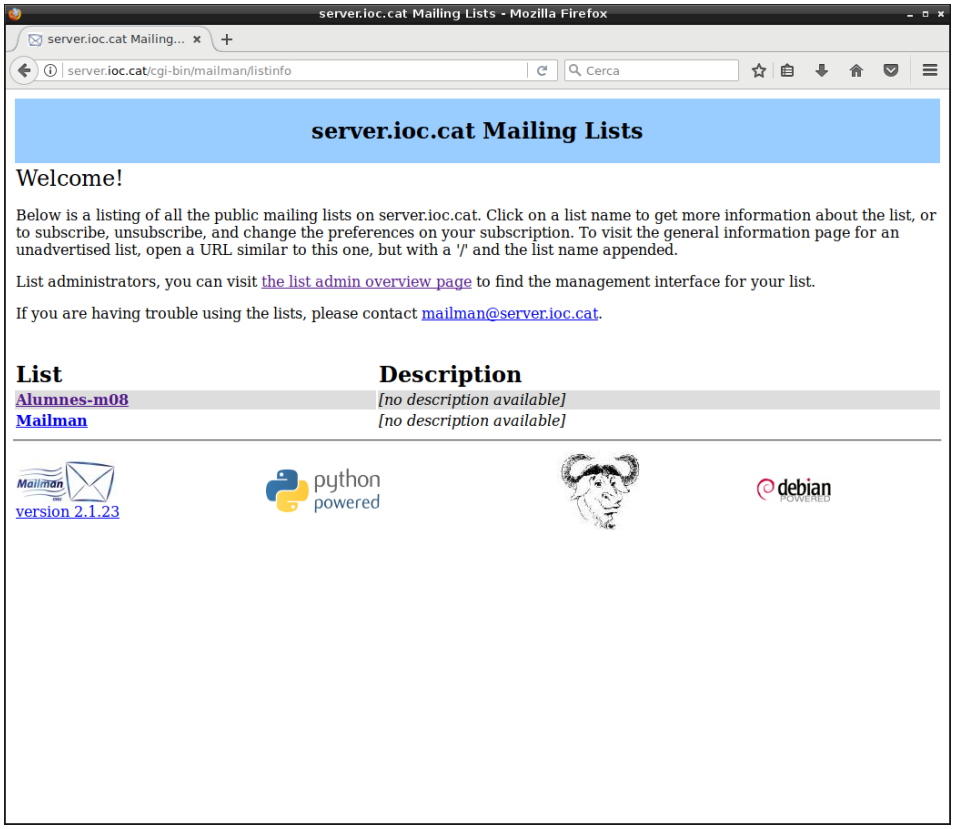
3. Rep un correu demanant la confirmació de la subscripció.
4. Confirma la subscripció per una de les tres vies proposades.
5. Estableix o modifica les condicions de subscripció.
6. Un cop confirmat, l'usuari pot començar a utilitzar la llista i a enviar missatges.

1. Es pot accedir via web a l'índex de les llistes de distribució d'un lloc web:

```
1 http://server.ioc.cat/cgi-bin/mailman/listinfo
```

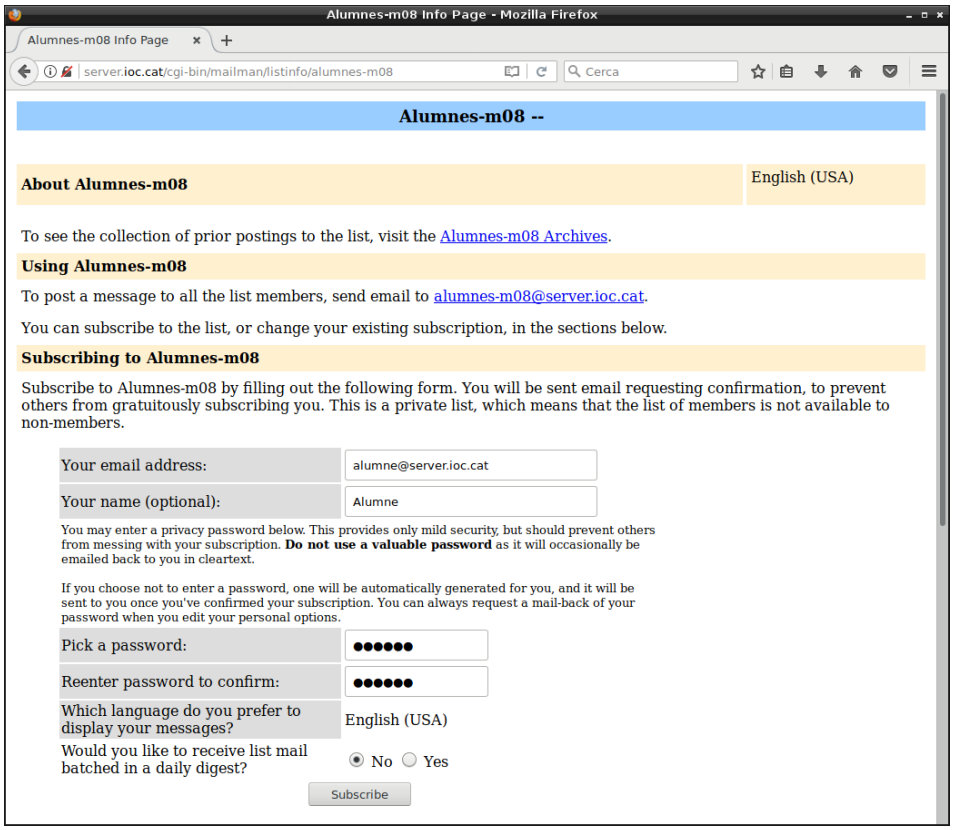
La figura 2.13 mostra que s'han creat dues llistes. Els usuaris subscrits poden utilitzar-les per intercanviar informació realitzant els seus *posts*. Els usuaris que encara no en són subscriptors poden accedir al web per subscriure-s'hi o poden enviar un correu electrònic amb una petició de subscripció.

FIGURA 2.13. Índex de llistes públiques d'un lloc web



2. L'usuari "alumne" ha omplert una petició de subscripció via web (vegeu figura 2.14).

FIGURA 2.14. Subscripció a alumnes-m08

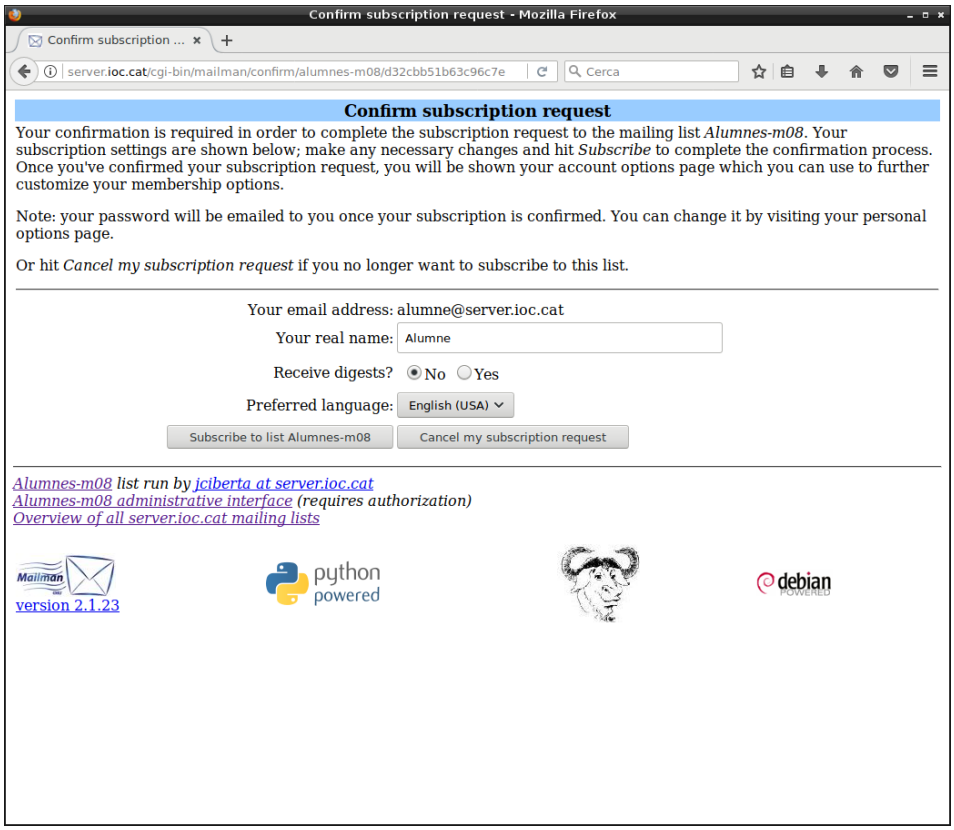


3. L'usuari "alumne" ha rebut a la seva bústia el correu electrònic de confirmació. De fet, el procés de confirmació depèn de com el configuri l'administrador de la llista. Demanar una confirmació a l'usuari és una prevenció contra la utilització fraudulenta de comptes d'altres usuaris. Quan Mailman envia un correu electrònic a l'usuari i exigeix una resposta, es verifica realment que l'usuari és qui ha fet la petició de subscripció.

```
root@server:~# su alumne
Contrasenya:
alumne@server:/root$ mail
"/var/mail/alumne": 1 message 1 unread
>U   1 alumnes-m08-reques dt des  3 15:52   48/2470  confirm
    d32cbb51b63c96c7e60ed4a4b48d783d0efb074f
? 1
```

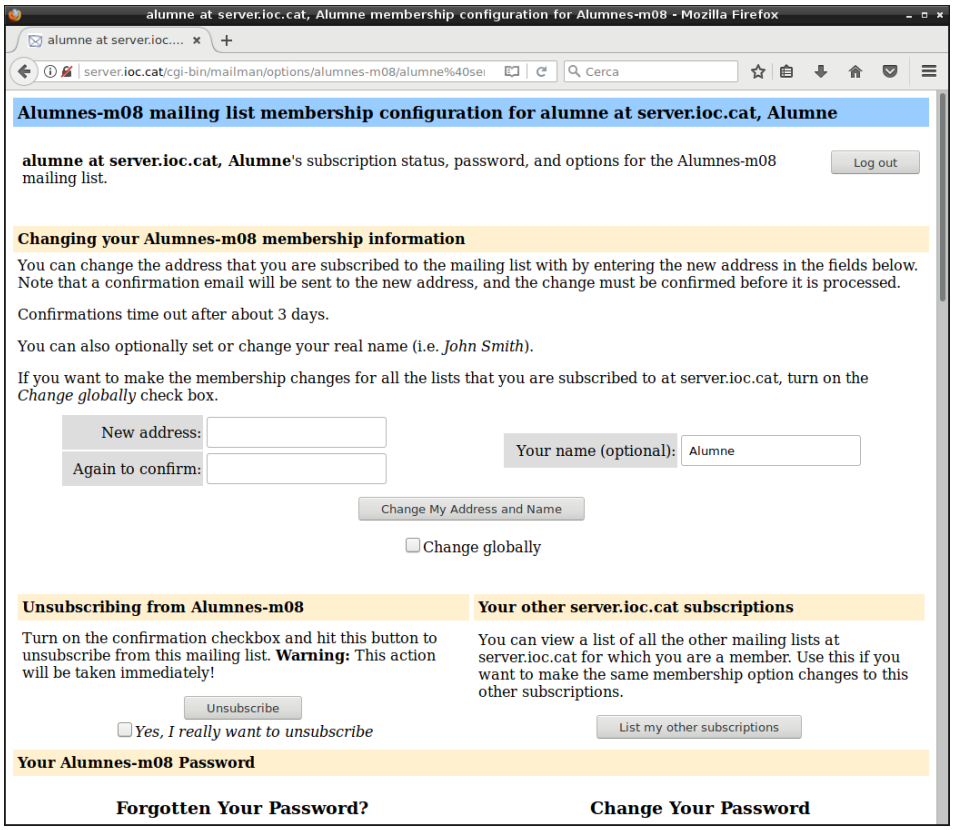
4. L'alumne confirma la seva subscripció utilitzant algun dels procediments indicats en el correu electrònic del llistat anterior. La figura 2.15 mostra la pantalla de confirmació usada per l'alumne.

FIGURA 2.15. Confirmació de la subscripció



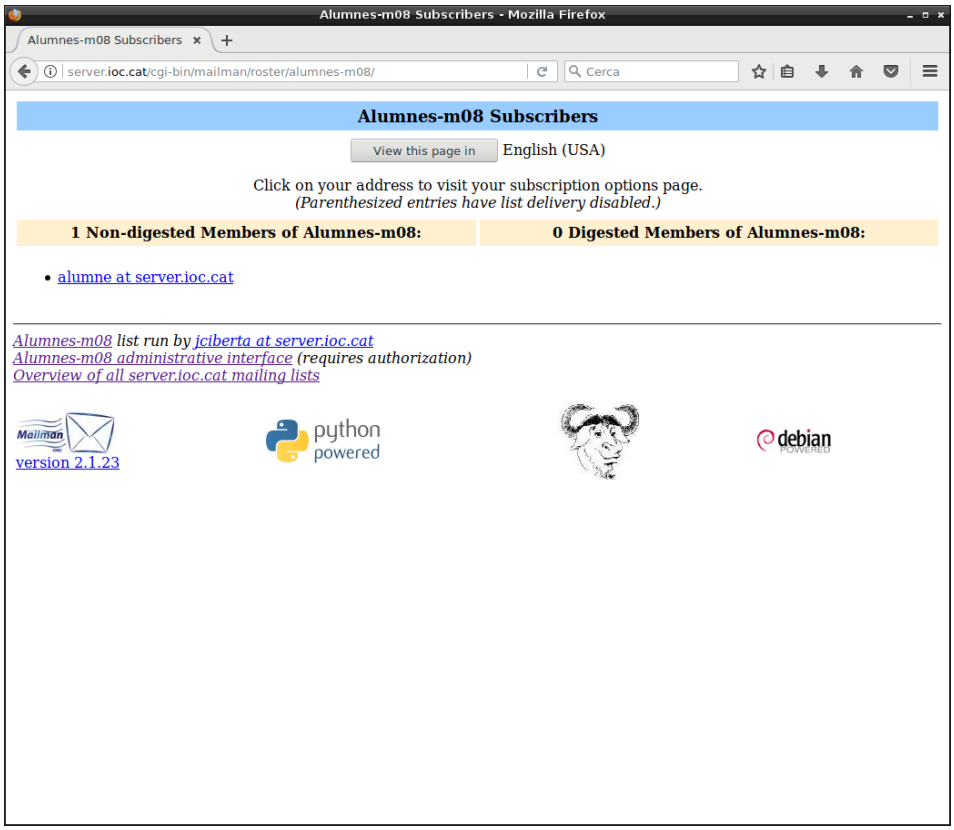
5. Tot usuari se subscriu a una llista amb unes condicions o configuració determinada. Inicialment s'aplica la configuració de subscripció per defecte, però l'usuari pot canviar aquesta configuració per a cada una de les llistes a les quals està subscrit. La figura 2.16 mostra part d'aquestes opcions.

FIGURA 2.16. Configuració de subscripció de l'alumne a alumnes-m08



6. L'alumne ja és subscriptor de la llista i pot usar-la al seu gust. Pot publicar missatges, examinar els *posts*, consultar les llistes del lloc i modificar la configuració de cada una de les seves subscripcions. La figura 2.17 mostra que l'usuari "alumne" és subscriptor de la llista i rep els missatges a mesura que es creen. També s'observa que no hi ha cap usuari que rebi els missatges en format de resum diari.

FIGURA 2.17. Llistat dels subscriptors de la llista alumnes-m08



En completar el procés de subscripció a una llista, l'usuari rep un correu electrònic del servei Mailman donant-li la benvinguda. A continuació podeu observar el missatge rebut per l'usuària "anna":

```
1  Date: Tue, 03 Dec 2019 16:01:09 +0100
2  From: alumnes-m08-request@server.ioc.cat
3  To: alumne@server.ioc.cat
4  Subject: Welcome to the "Alumnes-m08" mailing list
5
6  Welcome to the Alumnes-m08@server.ioc.cat mailing list!
7
8  To post to this list, send your message to:
9
10   alumnes-m08@server.ioc.cat
11
12 General information about the mailing list is at:
13
14   http://server.ioc.cat/cgi-bin/mailman/listinfo/alumnes-m08
15
16 If you ever want to unsubscribe or change your options (eg, switch to
17 or from digest mode, change your password, etc.), visit your
18 subscription page at:
19
20   http://server.ioc.cat/cgi-bin/mailman/options/alumnes-m08/alumne%40server.ioc
       .cat
21
22
23 You can also make such adjustments via email by sending a message to:
24
25   Alumnes-m08-request@server.ioc.cat
26
27 with the word 'help' in the subject or body (don't include the
28 quotes), and you will get back a message with instructions.
29
30 You must know your password to change your options (including changing
```

```
the password, itself) or to unsubscribe without confirmation.  It is:
...
```

D'aquest missatge es pot extreure la informació següent:

- Correu electrònic per fer apunts (*posts*): alumnes-m08@server.ioc.cat.
- Informació de la llista: http://server.ioc.cat/cgi-bin/mailman/listinfo/alumnes-m08
- Cancel·lació de la subscripció: http://server.ioc.cat/cgi-bin/mailman/options/alumnes-m08/alumne%40server.ioc.cat
- Adreça d'ordres: Alumnes-m08-request@server.ioc.cat.

Utilització de la llista

Els subscriptors poden enviar missatges a la llista de distribució, poden modificar la configuració de la seva subscripció (per cada llista on estan subscrits) i poden anul·lar la subscripció. Depenent de la configuració establerta pel propietari de la llista, els usuaris no subscriptors també podran escriure en una llista. Si la llista és de tipus *anunncement only*, només el propietari o els membres autoritzats poden escriure-hi, mentre que els subscriptors poden veure els apunts però no escriure'n.

- Fer un apunt

En el cas de la llista alumnes-m08, els subscriptors poden fer *posts* enviant missatges a:

```
alumnes-m08@server.ioc.cat
```

- Consultar la configuració de l'usuari "alumne" a la llista alumnes-m08:

```
http://server.ioc.cat/cgi-bin/mailman/options/alumnes-m08/alumne--at--server.
    ioc.cat
```

- Llegir els apunts:

```
http://server.ioc.cat/pipermail/alumnes-m08/
```

- Anul·lar una subscripció:

```
http://server.ioc.cat/cgi-bin/mailman/options/alumnes-m08/alumne@server.ioc.cat
```

- Enviar ordres al motor d'ordres de Mailman:

```
alumne@server:~$ mail -s "comanda" Alumnes-m08-request@server.ioc.cat
unsubscribe alumne
.
```

2.3 Servei de notícies

El **servei de notícies** o **NNTP** (*Network News Transfer Protocol* o **protocol de transferència d'articles**, o més senzillament *news*) està pensat per proporcionar una funcionalitat similar als taulers d'anuncis, on tothom pot publicar i llegir els missatges que hi ha penjats. En molts aspectes s'assembla al servei de correu electrònic, però el diferencia que aquí no cal especificar un destinatari.

L'objectiu que persegueix l'NNTP és difondre articles arreu del món sense que qui els publica n'hagi d'enviar una còpia als destinataris. Els articles es publiquen en servidors que els propaguen a altres servidors. Els usuaris que volen accedir als articles utilitzen un client *news* per connectar per NNTP amb els servidors locals o remots. Els articles s'organitzen en grups segons la temàtica o àmbit territorial, per exemple, per facilitar-ne la recerca.

L'**NNTP** és un protocol pensat per a la distribució, consulta, cerca i publicació d'articles mitjançant TCP (port 119). Està basat en el model client/servidor.

El servei de notícies es coneix indistintament com a servei *news*, servei NNTP o servei **Usenet**. Usenet és el nom de la xarxa original d'Unix en què es va basar el primer servei de notícies. Utilitzava l'**UUCP** (*Unix to Unix Copy Protocol* o **protocol de còpia d'Unix a Unix**) per copiar els articles d'una màquina Unix a una altra. Els usuaris feien connexions locals (mitjançant trucada telefònica) als servidors locals per accedir als articles existents i deixar-hi els seus.

Per obtenir més informació sobre l'especificació del protocol NNTP en els RFC 977, 1036 i 3977 consulteu la secció "Adreces d'interès" del web d'aquest mòdul.

El protocol NNTP es descriu originàriament en el document RFC 977 de l'any 1986. La versió actual correspon a l'RFC 3977 de l'any 2006. El format dels articles es descriu en l'RFC 1036 de l'any 1987 i es basa principalment en el mateix format que els missatges de correu (document RFC 822). També es pot fer servir l'IMAP per gestionar el servei de notícies.

Els serveis de notícies NNTP estan en retrocés a causa del gran èxit del servei web (HTTP) a internet. Cada cop s'utilitzen menys els servidors de notícies i més els fòrums web, que proporcionen una funcionalitat equivalent.

2.3.1 Descripció general

Hi ha diversos mecanismes per distribuir articles a usuaris repartits per internet. Potser el més evident és el correu electrònic, mitjançant les **llistes de distribució** o *Internet mailing lists*. L'usuari envia l'article per correu electrònic a una llista d'usuaris que creu que hi estaran interessats. L'inconvenient d'aquest model és l'ús ineficient de l'amplada de banda de la xarxa. Requereix enviar una còpia a cada destinatari. A més, es poden produir duplicacions, els usuaris pertanyen a diverses llistes i el reben diverses vegades, canvien d'ubicació o de correu electrònic... No hi ha un mecanisme de selecció de què es propaga i de què es vol rebre.

El servei de notícies és una evolució millorada del servei Usenet original que funcionava sobre la xarxa UUCP. Aquest mecanisme obligava els usuaris a realitzar una connexió als servidors Unix amb servei de notícies per iniciar-hi una sessió local, i així poder accedir als articles. Normalment es tractava de connexions per mòdem amb trucada telefònica al servidor i exigia a l'usuari disposar d'un compte en la màquina.

El servei de notícies NNTP utilitza un repositori central d'emmagatzemament d'articles que es distribueix de manera descentralitzada a altres servidors. L'arquitectura client/servidor permet als usuaris connectar-se als servidors per gestionar els articles.

L'NNTP utilitza el model client/servidor:

- **Client**. El client demana al servidor l'article que vol veure (en lloc de baixar-los tots). Client i servidor parlen en llenguatge NNTP. L'aplicació client normalment és un programari lector/generador d'articles.
- **Servidor**. El servidor rep i envia notícies als subscriptors i dels subscriptors, i en propaga a altres servidors. En el servidor hi ha programari que permet als subscriptors seleccionar els ítems que volen. Hi ha mecanismes d'indexació, selecció, referències creuades i expiració. El servidor ofereix servei a una àrea d'influència com una LAN, un campus, una ciutat, un país... Normalment, el servidor és un programari que treballa en segon pla (*background*) en forma de dimoni. Els articles sovint s'emmagatzemen en una cua (*spool*) a la qual els subscriptors accedeixen per obtenir-los i dipositar-los. Hi ha la figura del servidor esclau (*slave server*), que manté una memòria cau de notícies per donar servei a la seva àrea.

En el **protocol NNTP** només els articles no duplicats i desitjats són transferits.

Els elements i les funcions que intervenen en el servei de notícies són:

- **Distribució**. Usenet UUCP (el model antic) utilitzava la distribució d'articles per inundació d'amfitrió *host* a amfitrió. Es feien còpies de tot a tots els

amfitrions, evidentment amb una utilització ineficient dels recursos. Amb l'NNTP es fa la selecció de què es vol rebre i enviar i a qui. El subscriptor demana la llista de novetats i baixa el que li interessa (no tot). El client també informa el servidor dels articles que vol publicar, i el servidor els accepta si no són duplicats (pot filtrar). No es garanteix que un article arribi a tots els servidors del món ni a un de concret. S'eviten els bucles controlant el camí per on es distribueixen els articles. No hi ha un únic repositori mundial dels articles, sinó que formen una base de dades distribuïda (com la informació DNS).

- **Articles**. Un article o *news* és un text que un subscriptor publica en un servidor per tal que altres usuaris el puguin llegir (tipus tauler d'anuncis). Els articles porten associat un temps d'expiració (publicació per un període de temps limitat) que pot ser establert pel redactor, el servidor o el moderador. Per facilitar la cerca d'informació, els articles s'organitzen en grups segons el tema de manera jeràrquica. Un article pot pertànyer a diversos temes. La nomenclatura dels grups va de nivell més ampli a més concret separats per un punt. Per exemple: comp.os.linux (ordinadors, sistemes operatius, Linux).

- **Administració d'articles i grups**. Els articles i els grups que hi ha a Usenet poden ser administrats per un o més **moderadors**. Hi ha grups sense administració, grups en què les decisions es prenen de manera assembleària i grups moderats. El moderador decideix si permet la publicació de l'article o no, i en pot decidir la data d'expiració. A Usenet hi ha regles i votacions per decidir la gestió (creació, eliminació, modificació) de grups de notícies a nivell global.

- **Grups**. Els articles publicats en el sistema de notícies es van organitzar jeràrquicament en grups. Es van definir set grups en el nivell principal que posteriorment es van convertir en vuit. Són els següents:

  - *Comp*: temes relacionats amb la informàtica
  - *Misc*: temes no classificables en els altres apartats
  - *News*: articles sobre el mateix sistema de notícies
  - *Rec*: activitats recreatives, aficions, jocs...
  - *Sci*: temes científics
  - *Soc*: temes socials, culturals i humanístics
  - *Talk*: debats, opinions, discussions
  - *Humanities*: discussions de temes d'humanitats, literatura, filosofia

Hi ha una jerarquia alternativa a l'estàndard en què hi ha grups que no es podien crear en la jerarquia oficial. Són una mena de grup sense normes:

- *Alt*: jerarquia alternativa a l'oficial, que conté de tot i sense normes.

L'NNTP és un protocol basat en TCP que utilitza el port 119. Com tots els protocols "vells" d'internet, no ofereix cap mena de xifratge ni privacitat en

la informació que transporta. Es pot utilitzar NNTP per SSL (modalitat que anomenem NNTPS), la qual cosa permet connexions segures. Utilitza el port 563.

El format dels articles NNTP es basa en el document RFC 1036, dedicat a Usenet. Al seu torn, aquest format es basa en el format dels missatges de correu (document RFC 822). És a dir, els articles de Usenet són en concepte similars als correus electrònics. Els articles tenen una estructura basada en un conjunt de capçaleres, una línia en blanc de separació (CRLF) i el cos o text de l'article, igual que els correus electrònics.

Serveis d'àudio i vídeo

Josep Ciberta Tirado, Oriol Torres Carrió

Índex

Introducció

En el mòdul *Serveis de xarxa i Internet* s'estudia i practica la instal·lació i configuració de diversos serveis de xarxa. Molts d'aquests serveis són molt coneguts i es destinen a proporcionar serveis d'internet a l'usuari final (per això són populars): per exemple, el servei web (HTTP), el de transferència de fitxers (FTP), el de correu, el d'àudio i de vídeo... Tanmateix, hi ha altres serveis que, tot i ser imprescindibles a internet, són menys coneguts pels usuaris. Es tracta de serveis com DHCP, DNS o SMTP, que, tot i ser omnipresents, no són tan coneguts perquè no van destinats a l'usuari final, sinó a la configuració de les xarxes, a fer que les xarxes funcionin correctament.

En l'apartat **"Instal·lació i administració del servei d'àudio"** s'explica com mantenir i administrar adequadament els servidors de reproducció en temps real d'informació multimèdia. A grans trets, es mostren els paràmetres que ha de tenir en compte l'administrador a l'hora de definir els serveis de difusió en temps real de clips d'àudio i/o ràdio.

Cal que us familiaritzeu amb el format d'àudio digital, amb especial rellevància per a la conversió analogicodigital, per acabar centrant l'atenció en el format MP3. També s'aprofundeix en les principals diferències entre el *podcasting* i la reproducció en temps real.

La reproducció en temps real (*streaming*) transmet informació multimèdia en temps real. Cal saber definir i configurar el servidor de difusió en temps real d'àudio. S'expliquen els aspectes que cal tenir en compte a l'hora de configurar el servidor, com compartir la informació amb els diferents usuaris, etc.

En l'apartat **"Instal·lació i administració del servei de vídeo"** s'explica com mantenir i administrar adequadament els servidors de reproducció en temps real d'informació multimèdia. A grans trets, es mostra quins paràmetres ha de tenir en compte l'administrador a l'hora de definir els serveis adients per a la difusió en temps real d'arxius de vídeo.

S'estudien els principis fonamentals en què es basa el vídeo digital. Partint del fet que una seqüència de vídeo digital consisteix en una sèrie d'imatges que, en reproduir-se unes rere les altres, creen una sensació de moviment, es tracten els diferents formats d'imatge digital i s'aprofundeix en els formats de vídeo digital.

També es tracta l'entorn de l'ordinador tot estudiant la importància dels còdecs per, finalment, entrar a conèixer els principals paràmetres dels servidors de vídeo i de continguts continus. Així mateix, es veu la subscripció de vídeo fins a enllaçar amb un dels fenòmens més importants d'internet com és Youtube i el protocol en què es basa (*pseudostreaming*). Coincidint amb la importància de Youtube, s'exposa una eina que ha anat guanyant rellevància tant des del punt de vista empresarial com personal: les videoconferències.

Els dos temes tractats en aquesta unitat estan força relacionats. Amb tot, és preferible que feu una primera lectura global dels diferents apartats i que en una segona lectura aneu practicant *in situ* els passos descrits. Aquest procés pràctic es pot ampliar al mateix temps seguint els apunts i les activitats contingudes en el material web.

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'estudiant:

1. Administra serveis d'àudio identificant les necessitats de distribució i adaptant els formats.
    - Descriu la funcionalitat del servei d'àudio.
    - Instal·la i configura un servidor de distribució d'àudio.
    - Instal·la i configura el client per a l'accés al servidor d'àudio.
    - Reconeix i utilitza formats d'àudio digital.
    - Utilitza eines de reproducció d'àudio en el client.
    - Utilitza serveis d'àudio mitjançant el navegador.
    - Utilitza tècniques de sindicació i subscripció d'àudio.
    - Elabora documentació relativa a la instal·lació i administració del servidor d'àudio.
2. Administra serveis de vídeo identificant les necessitats de distribució i adaptant els formats.
    - Descriu la funcionalitat del servei de vídeo.
    - Instal·la i configura un servidor de vídeo.
    - Configura el client per a l'accés al servidor de vídeo.
    - Reconeix i utilitza formats de compressió de vídeo digital.
    - Utilitza tècniques de sindicació i subscripció de vídeo.
    - Descriu les característiques i els protocols utilitzats en el servei de videoconferència.
    - Instal·la i configura eines gràfiques per fer videoconferències.
    - Utilitza eines gràfiques i navegadors per fer videoconferències.
    - Elabora documentació relativa a la instal·lació i l'administració del servidor de vídeo i del servei de videoconferència.

1. Instal·lació i administració del servei d'àudio

La manera com escoltem música o mirem la televisió ha canviat molt en els darrers anys. De fet, fins i tot s'ha canviat el verb: ara tot aquest tipus de continguts multimèdia es *consumeix*.

En el cas del món de la música (i de l'àudio en general) la transformació ha estat enorme. De fet, ha patit diverses transformacions. Una va ser la transformació de tota una era analògica cap a un món digital amb la irrupció de les primeres tecnologies digitals (el *compact disc* o disc compacte) que aporten una millora en la qualitat del so.

Una altra transformació va ser la creació del format MP3, juntament amb el creixement d'internet, pel que fa a l'àmbit domèstic. Va revolucionar la manera de distribuir la música, amb la qual cosa el model anterior va quedar totalment obsolet.

Una darrera transformació ha estat l'aparició d'una nova tecnologia anomenada *streaming*, la reproducció en temps real, que fa que l'usuari ja ni disposi de fitxers MP3 per tal d'escoltar la seva música preferida, sinó que la consumeixi directament des d'internet, sense passar per un emmagatzematge previ.

Totes aquestes transformacions han anat impactant per generacions i han fet que les darreres generacions ja no estiguin tan familiaritzats amb conceptes com àudio analògic, disc, etc.

1.1 Servidors de reproducció en temps real

La reproducció en temps real s'acostuma també a anomenar amb el terme anglosaxó *streaming*, que vol dir 'corrent' o 'flux'. Al cap i a la fi, és una metàfora per indicar com flueixen les dades que s'estan transmetent.

La reproducció en temps real o *streaming* s'associa normalment amb continguts multimèdia, és a dir, a la transmissió i la distribució principalment d'àudio i vídeo. No obstant això, l'*streaming* també es pot utilitzar per a la transferència en temps real d'altres tipus de dades, com per exemple monitoratges o simulacions. De totes maneres, en la distribució d'àudio i vídeo és on ha agafat més rellevància.

La **reproducció en temps real** o *streaming* consisteix a anar consumint (reproduint, en el cas dels continguts multimèdia) les dades que es transmeten sense emmagatzemar-les.

És a dir, es tracta de consumir dades mentre es van descarregant, tal com es pot veure en la figura 1.1. La tècnica consisteix a tenir un *buffer*, que és una memòria intermèdia, que permeti emmagatzemar temporalment el que es reproduirà. D'aquesta manera sempre es tenen les dades amb anterioritat a la reproducció en condicions de transmissió constants (ample de banda).

FIGURA 1.1. Procés de reproducció en temps real

Els protocols principals per a la transmissió de dades en temps real són el protocol de transport en temps real (RTP) i el protocol de control RTP (RTCP). Aquest serveixen com a base per a altres protocols, com el protocol *Real Time Streaming Protocol* (RTSP). Aquests solen usar el protocol de transport *User Datagram Protocol* (UDP), ja que és un protocol no orientat a connexió i no fiable que fa augmentar la velocitat de transmissió en no demanar reconeixement (*acknowledge*) que les dades han arribat al receptor. No obstant això, també pot treballar amb el protocol de transport (TCP, *Transmission Control Protocol*).

Protocol no fiable

Un protocol és no fiable quan no comprova si les dades han arribat al destí. A vegades la fiabilitat la dona un protocol d'una capa superior, o a vegades no cal.

Darrerament, algunes empreses especialitzades en la distribució de continguts multimèdia han desenvolupat altres protocols per a la distribució de dades en temps real que tenen en compte altres característiques. Una d'aquestes consisteix a detectar la disponibilitat d'ample de banda per part del servidor i el client per tal d'ajustar el flux (*stream*) i la seva qualitat. Aquests protocols s'anomenen protocols d'*streaming* de *bitrate* adaptatiu, i els més coneguts són HLS, HDS, Smooth Streaming i MPEG-DASH.

1.2 Àudio digital

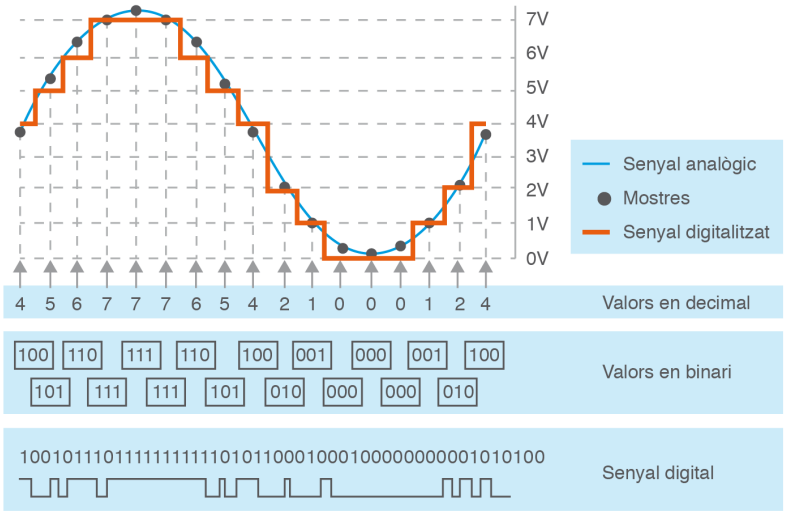
L'àudio digital és aquell que utilitza senyals binaris per a l'emmagatzematge i la reproducció del so. La transformació d'un senyal sonor (ona mecànica) a un senyal elèctric (ona electromagnètica) se sol fer a través d'un transductor (micròfon). Per tant, aquest senyal sonor es convertirà en un senyal elèctric binari, és a dir, en tot un seguit de 0 i 1, que quedaran ben definits per dos voltatges principals (per exemple, 0 volts per al valor 0 i 5 volts per al valor 1).

El senyal elèctric (analògic) pot patir un processat abans de la conversió analogicodigital que pot incloure reducció del soroll, l'amplificació, l'eliminació de freqüències i l'equalització, entre altres processos.

La conversió d'un senyal analògic a digital (en aquest cas, un senyal d'àudio) es fa mitjançant un procediment anomenat modulació per impulsos codificats

(PCM, *Pulse Code Modulation*) a través d'un conversor analogicodigital (ADC, A/D o *analog-to-digital converter*). Consta de tres etapes principals: mostreig, quantificació i codificació, tal com es pot veure en la figura 1.2:

FIGURA 1.2. Conversió d'un senyal analògic a digital



El **mostreig** és l'etapa que consisteix a agafar mostres (valors del senyal) depenent d'una freqüència prefixada, anomenada freqüència de mostreig. Com més alta sigui aquesta freqüència, més qualitat tindrà l'àudio digital. No obstant això, l'oïda humana no percep freqüències superiors als 20 KHz aproximadament.

El **teorema de Nyquist** (teorema de mostreig de Nyquist-Shannon) indica que per obtenir un senyal digital que contingui components fins a una certa freqüència cal mostrejar com a mínim al doble d'aquesta freqüència.

Per tant, mostrejar per sobre d'aquest valor no aportarà més informació per a les freqüències més baixes (demostració matemàtica fora de l'abast), és a dir, el senyal digital contindrà la mateixa informació que el senyal analògic per a aquestes freqüències.

Altres freqüències de mostreig

- 8 KHz per a telefonia digital
- 22,05KHz per a ràdio digital

Per exemple, la freqüència de mostreig en un CD de música (*Compact Disc Digital Audio* o CDDA) és de 44.100 Hz, és a dir, es prenen 44.100 mostres per segon.

La **quantificació** és l'etapa que assigna valors discrets a les mostres (valors analògics). En aquest pas es produeix una pèrdua de qualitat del senyal respecte a l'original que és inherent a l'àudio digital. Aquesta discretització es fa dins una escala determinada que determinarà la **precisió** de la mostra discreta.

Valor analògic: valor que pot prendre un nombre infinit de valors.

Valor discret: valor que pot prendre un nombre finit de valors.

La **codificació** és l'última etapa que assigna una seqüència de bits al valor quantificat. La longitud d'aquesta seqüència dependrà de l'escala que s'hagi fet

servir en la quantització, i s'anomena **resolució** (nombre de bits per mostra). Per exemple, si el rang de l'escala en què s'ha quantificat el senyal digital té 7 valors, llavors es necessitaran 3 bits per codificar-la, ja que $2^3 \geq 7$ Per exemple, en un CD de música la codificació és de 16 bits, és a dir, el rang és de 65.536 valors discrets diferents.

A més a més, un àudio digital pot tenir diversos canals. Per exemple, el so estereofònic té dos canals (dret i esquerre), la qual cosa fa que en escoltar el so l'experiència sigui més natural en provocar que l'àudio provingui de diferents fonts. Altres exemples són el sistema 5.1, que disposa de 6 canals. Tot el procés de conversió d'un senyal analògic a digital es realitza per a cada canal.

Per tant, els principals paràmetres que s'han de tenir en compte en la conversió d'un senyal analògic a digital són:

- Freqüència de mostratge
- Resolució (nombre de bits per mostra)
- Nombre de canals

1.2.1 Formats d'àudio digital

Els formats d'àudio digital es poden classificar en 3 grans grups:

- Formats d'àudio no comprimits
- Formats d'àudio comprimits sense pèrdua (*lossless compression*)
- Formats d'àudio comprimits amb pèrdua (*lossy compression*)

MIDI

Els arxius MIDI no contenen àudio en si. Realment contenen una seqüència de música enregistrada amb un conjunt de números que indiquen com s'ha de reproduir.

Hi ha algun altre format d'àudio que no prové d'un senyal digital d'àudio. És el cas del format MIDI (*Musical Instrument Digital Interface*, interfície digital d'instruments musicals), que és un estàndard de comunicació entre components musicals electrònics tals com instruments electrònics o ordinadors.

Els formats d'àudio no comprimits no s'usen per a la distribució d'àudio a causa de la seva mida, sinó per a l'edició d'àudio digital (per la rapidesa a l'hora de processar el senyal). Posteriorment, si s'ha de distribuir per algun mitjà que requereixi alguna limitació, com per exemple un ample de banda concret, es codifiquen en un format comprimit.

Els formats d'àudio comprimits basen la reducció de la seva mida en algorismes que tenen en compte les característiques de l'oïda humana.

Una altra cosa que cal tenir en compte quan es parla de formats, i sovint s'usa indistintament o de forma genèrica, és la distinció entre còdecs d'àudio i formats contenidors (en el cas d'àudio, però també és vàlid per a vídeo).

Un **còdec** és un codificador que permet la compressió del senyal digital per al seu emmagatzematge o transmissió i també el descodificador per a la seva reproducció.

De fet, *còdec* és l'abreviació de codificador-descodificador. Alguns còdecs tenen el mateix nom que el fitxer contenidor que els conté (per exemple, MP3) i d'altres no (AAC/MP4). Fins i tot hi ha variants en les extensions dels fitxers contenidors (MP4, M4A, etc.).

Formats d'àudio no comprimits:

- **WAV** (*Waveform Audio File Format*): format desenvolupat per Microsoft i IBM.
- **AIFF** (*Audio Interchange File Format*): format desenvolupat per Apple.
- **AU**: format desenvolupat per la desapareguda Sun Microsystems (absorbida per Oracle el 2010).

Formats d'àudio comprimits sense pèrdua:

- **FLAC** (*Free Lossless Audio Codec*): format obert i lliure de *royalties* mantingut per la fundació Xiph.Org.
- **ALAC** (*Apple Lossless Audio Codec*): format desenvolupat per Apple. El còdec que dona nom al format, tot i que inicialment era propietari, actualment està publicat sota una llicència de codi obert i lliure de *royalties*.
- **WMA lossless** (*Windows Media Audio*): format propietari desenvolupat per Microsoft.

Formats d'àudio comprimits amb pèrdua:

- **MP3** (*MPEG Audio layer III*): format molt popular desenvolupat pel Moving Picture Experts Group.
- **WMA lossy** (*Windows Media Audio*): format propietari desenvolupat per Microsoft.
- **OGG/Opus**: Opus és un còdec obert i lliure de *royalties* desenvolupat per la fundació Xiph.Org i estandarditzat per Internet Engineering Task Force (IETF) com a RFC6716. Aquest còdec sol anar dins del format contenidor obert OGG (Ogg Vorbis).
- **AAC** (*Advanced Audio Coding*): dissenyat per ser el successor de l'MP3, aconsegueix taxes de compressió superiors a aquest tot mantenint la mateixa qualitat. Com a contenidor usa el format MP4 (a vegades l'extensió és M4A per identificar clarament que és un fitxer d'àudio). És el còdec/format usat per iTunes i YouTube, entre altres grans distribuïdors.

- **AC3** (*Audio Codec 3*): també és conegut com a Dolby Digital. És un format molt estès en el cinema i en els DVD, entre d'altres.
- **RA** (*Real Audio*): format molt utilitzat per a la reproducció d'àudio en temps real (es va reproduint mentre es realitza la descàrrega). Avui en dia ha perdut força popularitat.

1.2.2 El format més popular: l'MP3

L'MP3 (també conegut de manera formal com a MPEG-1 Audio Layer III i MPEG-2 Audio Layer III) és un format de codificació d'àudio digital. Aquest és un format de compressió d'àudio amb pèrdua, fet que permet una reducció considerable de la mida d'un fitxer al mateix temps que garanteix una alta qualitat sonora.

El format MP3 es basa en la reducció o eliminació de certs components del so que no són percebuts per la majoria dels humans. El mètode es basa en dos algorismes matemàtics: la transformada discreta del cosinus modificada (MDCT, *Modified Discrete Cosine Transform*), que és el nucli de l'MP3, i la transformada ràpida de Fourier (FFT, *Fast Fourier Transform*). Aquestes permeten fer l'anàlisi espectral del so i filtrar-lo de forma digital. Amb aquestes tècniques es redueix de mitjana entre un 75 i un 95% la mida d'un fitxer.

MP3 forma part dels estàndards MPEG, tant dels estàndards MPEG-1 com del MPEG-2. L'estàndard MPEG-1 part 3 (que inclou l'MP3) va ser publicat el 1993, i l'estàndard MPEG-2 part 3, el 1995.

Fraunhofer-Gesellschaft

Organisme que agrupa tots els centres (instituts) de recerca d'Alemanya especialitzats en diversos camps de la ciència aplicada. El seu nom ve del físic alemany Joseph von Fraunhofer.

Els inventors que apareixen a la patent americana són Bernhard Grill, Karlheinz Brandenburg, Thomas Sporer, Bernd Kurten i Ernst Eberlein. No obstant això, el que ha passat a la història per ser el pare de l'MP3 és Karlheinz Brandenburg, que va liderar la recerca al Fraunhofer Institute d'Alemanya.

A l'inici, el format MP3 estava patentat, i permetia la reproducció (descodificació). Aquest fet va facilitar la distribució d'àudios en format MP3, concretament de música. Un dels reproductors més conegut d'aquella època (mitjan anys noranta) va ser el WinAmp, de la companyia NullSoft, que juntament amb l'expansió d'internet va fer que el format MP3 esdevingués un dels formats més populars.

No obstant això, per a la creació (codificació) dels MP3 s'havia de pagar la patent, fins que un estudiant australià, utilitzant tècniques d'enginyeria inversa, va crear un codificador i el va publicar. Aquest fet, i l'aparició de xarxes d'igual a igual (*peer-to-peer*), va significar l'inici d'una una nova era en la indústria de la música (infraccions dels drets d'autor, demandes, pirateria, etc.) que sempre quedarà associat al format MP3 i que va abocar el sector a un nou model radicalment diferent.

Actualment, la tecnologia MP3 és lliure de patents a Europa des de 2012 i als Estats Units des de 2017. Tot i que encara continua gaudint de certa popularitat, està sent desplaçada pel nou format MP4.

Un CD té 44.100 mostres per segon, codificat amb 16 bits i 2 canals; això dona 1411200 bit/s, és a dir 1.411 kbit/s

Pel que fa a la compressió, el format és capaç de codificar un disc digital compacte, passant d'un *bitrate* de 1.411 kbit/s a *bitrates* de 320 kbit/s o inferiors. Els *bitrates* més típics són 192 i 128, tot i que n'hi ha d'altres (160, 144, etc.).

Les característiques més rellevants del format MP3 són:

- Compressió elevada
- Qualitat acceptable
- Facilitat en la distribució

1.2.3 Llistes de reproducció

La reproducció en temps real permet la reproducció contínua d'elements multimèdia. Per exemple, en el cas de la reproducció de cançons per a un fil musical es pot enllaçar la finalització d'una cançó amb l'inici de la següent. Això es fa gràcies a l'ús de llistes de reproducció.

Les **llistes de reproducció** (*playlists*, en anglès) són unes llistes de fitxers d'àudio (també poden ser de vídeo) preparades per ser reproduïdes en algun mitjà tant de forma seqüencial com aleatòria. Les entrades de fitxer d'àudio poden estar referenciades de forma externa, és a dir, fer referència a una cançó en un altre servidor.

A més, les *playlists* poden contenir informació addicional, com ara títol, autor, disc, etc. Hi ha diferents formats per a les llistes de reproducció, i la seva interpretació depèn del programari on es reprodueixin.

Els programaris més coneguts són:

- **m3u** (MP3 URL): format desenvolupat originalment pel Fraunhofer Institute, és un dels formats més populars i més suportats. Té una sintaxi senzilla en forma de text pla.
- **m3u8**: versió *unicode* de m3u. *Unicode* és un estàndard per a la codificació de caràcters que permet codificar la totalitat d'alfabets actuals, entre altres característiques.
- **pls**: un altre format de *playlist* una mica més complet que m3u.
- **smil** (*Synchronized Multimedia Integration Language*): format desenvolupat pel W3C (World Wide Web Consortium) per a la presentació de continguts multimèdia.

- **asx** (*Advanced Stream Redirector*): format desenvolupat per Microsoft que emmagatzema les llistes de reproducció en XML.
- **xspf**: format XML desenvolupat per la fundació Xiph.org encarregada de crear i promocionar formats multimèdia lliures (còdecs, *playlists*, etc.)
- **wpl** (Windows Media Player Playlist): format propietari desenvolupat per Microsoft basat en smil.

Alguns exemples són:

- M3u:

```
#EXTM3U

#EXTINF:123,Artista – Títol
C:\Documents and Settings\usuari\My Music\Exemple.mp3

#EXTINF:321,Artista – Títol
http://www.web.com/~usuari/Exemple.mp3
```

- Pls:

```
[playlist]
File1=http://www.web.com:8020/
Title1=Estacio de ràdio
File2=Exemple.mp3
Title2=Artista – Títol
Length2=120
NumberOfEntries=2
```

- Xspf:

```
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/">
  <trackList>
    <track>
      <title>Windows Path</title>
      <location>file:///C:/music/Exemple.mp3</location>
    </track>
    <track>
      <title>Linux Path</title>
      <location>file:///media/music/Exemple.mp3</location>
    </track>
    <track>
      <title>Relative Path</title>
      <location>music/Exemple.mp3</location>
    </track>
    <track>
      <title>External Example</title>
      <location>http://www.web.com/music/Exemple.ogg</location>
    </track>
  </trackList>
</playlist>
```

Cal tenir en compte que no tots els reproductors i servidors admeten totes les llistes de reproducció. Per exemple, el servidor Ampache suporta els següents formats: m3u, m3u8, asx, pls i xspf.

1.3 Subscripció d'àudio

En els últims anys, la ràdio i la televisió s'han transformat a causa d'internet i les noves tecnologies. La digitalització d'ambdues tecnologies ha estat constant, si bé ha estat la darrera la que ha patit la transformació més gran, ja que no transmet de forma analògica. La ràdio, en canvi continua emetent-se en analògic a través de les modulacions AM i FM, i la digitalització en aquest cas ha estat menor. L'**RDS** (*Radio Data System*) és un estàndard que permet enviar informació digital sobre ones analògiques. D'aquesta manera, cada emissora pot enviar informació com el nom de la cançó o la freqüència alternativa per quan el senyal perdi potència , entre altres característiques.

No obstant això, la part de la ràdio que més s'ha transformat és la d'internet. Han aparegut una multiplicitat de ràdios en línia per la facilitat de muntar servidors i la no-necessitat d'una estructura com la de la ràdio convencional (llicència per emetre, antenes, etc.). També ha canviat la manera com s'escolta la ràdio: en viu (en temps real) o en forma de *podcast*, que permet descarregar l'arxiu per escoltar-lo més tard.

1.3.1 'Podcast'

El terme *podcast* va ser usat primer cop pel periodista Ben Hammersley en un article al diari britànic *The Guardian* per intentar descriure el fenomen de les descàrregues automàtiques de programes d'àudio el 2004. No obstant això, Adam Curry, juntament amb Dave Winer, passa per ser el gestador de la idea i de la creació del primer programari per a la descàrrega automàtica d'arxius, anomenat iPodder. Per fer-ho possible van utilitzar les especificacions RSS, en el desenvolupament de les quals havia participat Dave Winer.

La paraula ***podcast*** és la combinació d'iPod, conegut reproductor portable, i *broadcast*, que en anglès significa 'difusió'.

> Un ***podcast*** és un programa (de veu o musical) o conjunt de programes preparats per ser descarregats de forma automàtica a través d'internet mitjançant una subscripció, tot i que també es poden descarregar de forma individual i poden ser reproduïts posteriorment en qualsevol altre moment.

La subscripció es fa de diverses maneres:

- A través d'una extensió del navegador.
- A través d'un lector de RSS web. Google Reader va ser un lector molt popular, tot i que ja està obsolet. Actualment hi ha alternatives com Feedly.

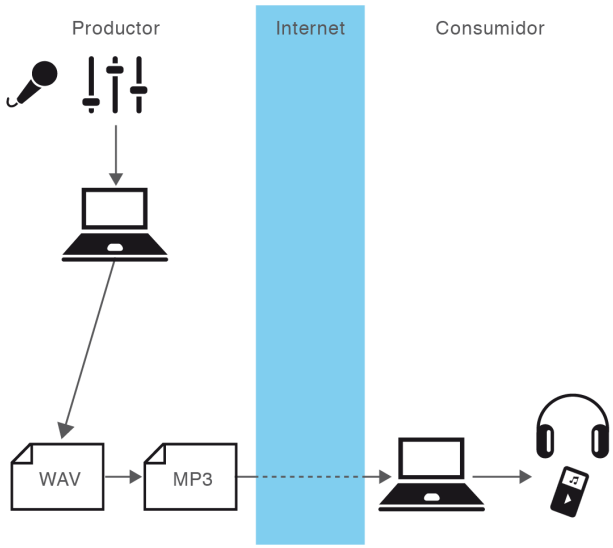
- Per a ordinadors, a través d'un programari de *podcasts*, com per exemple VLC o gPodder.
- Per a mòbils i tauletes, a través d'alguna aplicació de *podcasts*. En el cas de dispositius Apple, iTunes ja fa aquesta tasca.

Òbviament, també existeix l'opció cancel·lar la subscripció en el cas que no es desitgi rebre més actualitzacions automàtiques dels programes.

Els clients que són capaços de descarregar *podcasts* se'ls anomena ***podcatchers***. Tot i que generalment descarreguen àudio, també són capaços de descarregar vídeo, notícies, text i imatges.

En la figura 1.3 podeu observar un esquema del recorregut que fa un *podcast* des de la seva creació fins que arriba a l'oient:

FIGURA 1.3. Creació d'un podcast



1.3.2 Diferències entre 'podcasting' i reproducció en temps real

Tot i les similituds entre *podcasting* i reproducció en temps real, aquests difereixen en la seva concepció. La principal diferència és que la reproducció en temps real està pensada per al consum multimèdia de forma immediata, mentre que el *podcasting* està pensat per a un consum posterior.

La **sindicació** és una de les principals característiques del *podcasting*, i permet la descàrrega automàtica d'arxius, principalment d'àudio, però també poden ser de vídeo, notícies, text o imatges.

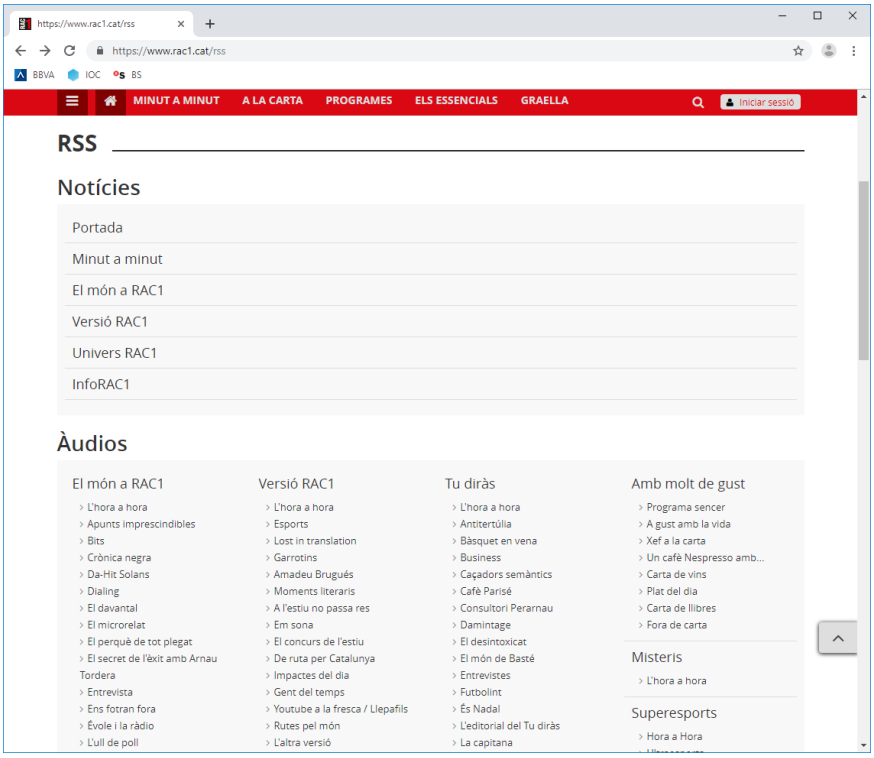
Atom

Estàndard de sindicació que intenta resoldre alguns dels problemes de RSS, augmentar-ne les capacitats i comportar-se com un estàndard tradicional (RFC 4287 i 5023).

El sistema de sindicació utilitzat és l'**RSS** (*Really Simple Syndication*), tot i que alguns també permeten l'ús d'Atom.

En la figura 1.4 podeu veure la pàgina de sindicació d'una coneguda ràdio que no tan sols ofereix sindicació per als àudios, sinó també per a les notícies.

FIGURA 1.4. RSS d'una ràdio comercial



Vegeu un exemple de fitxer RSS (retallat). Dins de les especificacions RSS només hi pot haver un únic canal (*channel*), i a través dels elements *item* es van publicant els nous continguts:

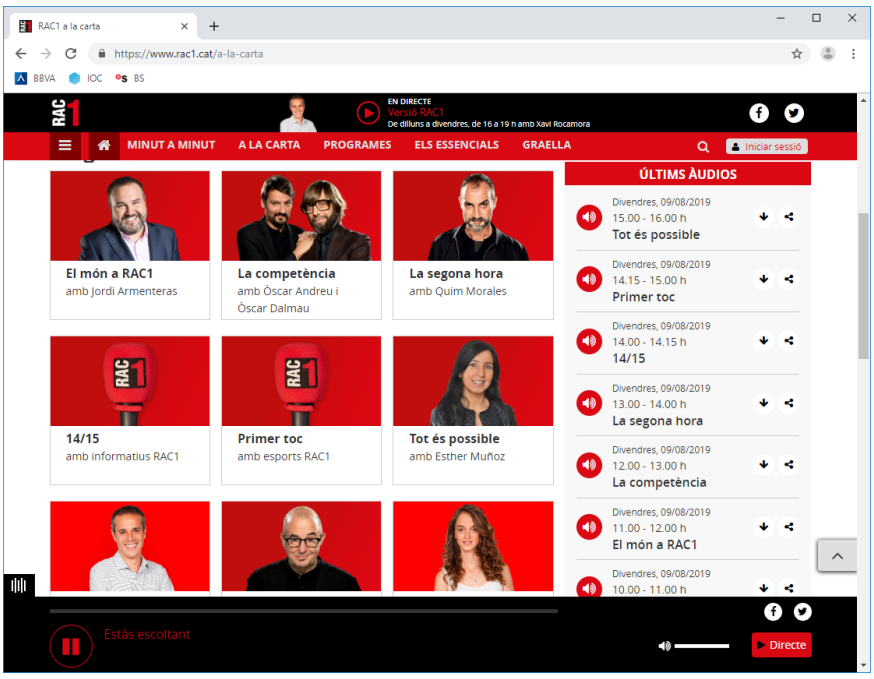
```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:atom="http://www.w3.org/2005/Atom" xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd" version="2.0">
  <channel>
    <title>La competència - Programa sencer</title>
    <link>http://www.rac1.org/lacompetencia/</link>
    <description>Podcasts del programa La competència - Programa sencer</description>
    <item>
      <title>La competència Divendres 2019-08-09 12:00</title>
      <link>https://audioserver.rac1.cat/get/be39fc26-0f7f-46e0-ae82-3444ec2977d0/1/2019-08-09-la-competencia-12h.mp3?source=RSS</link>
      <pubDate>Fri, 09 Aug 2019 10:00:00 GMT</pubDate>
    </item>
    <item>
      <title>La competència Dijous 2019-08-08 12:00</title>
      <link>https://audioserver.rac1.cat/get/2d21c240-9a1e-422e-a0e8-963774f1728f/1/2019-08-08-la-competencia-12h.mp3?source=RSS</link>
      <pubDate>Thu, 08 Aug 2019 10:00:00 GMT</pubDate>
    </item>
    <item>
      <title>La competència Dimecres 2019-08-07 12:00</title>
      <link>https://audioserver.rac1.cat/get/da49f168-4cf1-41e6-af4b-d2615a3fbde2/1/2019-08-07-la-competencia-12h.mp3?source=RSS</link>
      <pubDate>Wed, 07 Aug 2019 10:00:00 GMT</pubDate>
    </item>
    ...
  </channel>
</rss>
```

Les principals diferències que hi ha entre la reproducció en temps real (*streaming*) i el *podcasting* són:

- En la reproducció en temps real no es descarrega inicialment un fitxer per ser seguidament reproduït, sinó que l'arxiu d'àudio es va reproduint (es pot anar escoltant) mentre aquest es descarrega. Un cop consumit l'arxiu (finalitzada la reproducció), si es vol tornar a escoltar cal tornar a realitzar l'*streaming*, és a dir, es torna a descarregar l'arxiu. En canvi, en el *podcasting* l'arxiu es descarrega un sol cop i es pot reproduir tantes vegades com es vulgui.
- El *podcasting* és passiu. La distribució de programes és automàtica i gràcies a la sindicació es descarreguen els nous arxius quan estan disponibles. Posteriorment es pot reproduir quan es desitgi. La reproducció en temps real és activa. Cal que l'usuari faci alguna acció per tal de reproduir un àudio.
- La reproducció en temps real és dependent de la connexió a internet. Si aquesta és deficient o hi ha problemes puntuals, es veuran afectats en la qualitat de la transmissió. En el *podcasting*, com que l'arxiu està descarregat localment, no es produeixen aquests incidents.
- La reproducció en temps real és més simple per a l'usuari, ja que no ha de pensar a subscriure's a cap canal ni ha de saber on estan emmagatzemats els fitxers. Fent una analogia, es podria comparar amb la ràdio tradicional (*streaming*) i els CD de música (arxius descarregats).

En la figura 1.5 podeu veure les opcions en una ràdio comercial. A baix de tot es pot escoltar la ràdio en directe, i a la columna de la dreta es pot escoltar la ràdio a la carta o baixar el programa per escoltar-lo més endavant.

FIGURA 1.5. Opcions de servei per internet



1.4 Reproducció en temps real

La reproducció en temps real consisteix a anar mostrant el contingut mentre es va descarregant, normalment contingut multimèdia. Això sol ser degut a la grandària dels arxius que es volen mostrar. Si s'haguessin de descarregar completament abans, aquest tipus de continguts no haurien estat tan populars (per exemple, si cada vegada que es visualitza un vídeo de Youtube el consumidor s'hagués d'esperar perquè es descarregués completament, segurament aquesta plataforma no hauria tingut tanta repercussió).

Per tal de poder mostrar aquests continguts s'ha de transmetre aquest flux de manera constant, és a dir, a la mateixa velocitat. Això es coneix com a **taxa de bits**.

Bits/bytes

Cal anar alerta per no confondre bits (símbol b) amb bytes (símbol B), ja que bytes és una magnitud 8 vegades superior. És un error típic confondre l'ample de banda de les línies de telefonia.

La **taxa de bits** (en anglès, *bitrate*) es defineix com la freqüència amb què es transmeten les dades per un canal, o el nombre de bits que es transmeten per segon i que defineixen l'ample de banda. Les unitats en el sistema internacional són els bits/segon (a vegades escrit bps, bits per segon), tot i que depenent de la magnitud s'usen múltiples unitats (Kb/s, Mb/s).

Cal tenir en compte que aquests són múltiples de 1.000 i no de 1.024 (com passa amb els bytes).

Aquest terme s'usa també per a la codificació i la compressió d'àudio i vídeo. Hi ha dos tipus principals de tècniques a l'hora de codificar o comprimir un arxiu multimèdia: CBR i VBR.

La **taxa de bits constant** (CBR, *Constant Bit Rate*) estableix el *bitrate* de forma numèrica (amb mètodes estadístics) i es manté constant per a tota la duració de l'arxiu. Aquest mètode fa que l'arxiu resultant sigui bastant extens, però és molt útil a l'hora de transmetre continguts multimèdia per determinats canals on l'ample de banda té poca capacitat.

La **taxa de bits variable** (VBR, *Variable Bit Rate*) estableix un *bitrate* de mitjana, però que va variant en funció de les característiques del senyal, és a dir, hi ha parts d'un arxiu multimèdia que necessiten una taxa de bits més alta per representar aquella porció d'informació perquè té un nivell més alt de detall. Per exemple, en una cançó, un tall que tingui molts instruments i presenti una multiplicitat de freqüències diferents amb molts harmònics necessitarà molts més bits per unitat de temps per codificar aquesta informació que un altre tall més simple. Aquest mètode aconsegueix una qualitat més alta, i la mida de l'arxiu pot variar considerablement.

Per tant, a l'hora de codificar arxius multimèdia s'ha d'escollir entre un mètode i l'altre. Aquesta elecció dependrà de les característiques que es vulguin tenir per a l'arxiu resultant en funció de la qualitat, l'emmagatzematge i la distribució.

1.4.1 Protocols bàsics

Per poder realitzar aquest tipus de transmissió s'usen uns protocols especials que fan que no calgui usar tot l'ample de banda disponible. Una característica comuna a tots aquests protocols és la de descarregar inicialment un *buffer* (memòria intermèdia temporal) per tal de corregir possibles fluctuacions en el senyal (*jitter*). D'aquesta manera, la part descarregada sempre va per davant de la part que s'està reproduint.

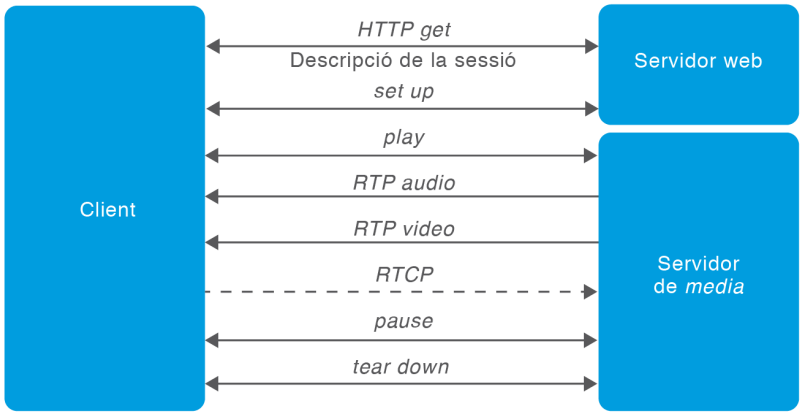
Els protocols més coneguts per a la transmissió de dades en temps real són l'RTP, juntament amb l'RTCP.

El protocol de transport en temps real (**RTP**) proporciona serveis de transport per a aplicacions que transmetin en temps real, com poden ser d'àudio, vídeo o dades per a simulacions. L'adreçament d'aquests serveis pot ser tant *unicast* com *multicast*. Normalment, RTP usa UDP, però també està preparat per treballar amb altres protocols de transports o de xarxa. No proporciona mecanismes per assegurar el lliurament en un temps concret ni proveeix cap garantia de qualitat del servei, sinó que confia en les capes subjacents. Aquest protocol està molt relacionat amb l'TCP (protocol de control RTP) i treballen estretament. La versió segura d'aquest protocol és l'SRTP (*Secure Real-time Transport Protocol*), que ofereix confidencialitat i autenticació de missatges, entre altres característiques.

El protocol de control RTP (**RTCP**) es basa en la transmissió periòdica de paquets de control a tots els participants. Una de les funcions principals és proporcionar dades sobre la qualitat de la transmissió (distribució de dades).

En la figura 1.6 podeu veure un esquema del funcionament dels protocols RTP/RTCP.

FIGURA 1.6. Esquema del funcionament dels protocols RTP/RTCP



1.4.2 Reproducció en temps real d'informació multimèdia

Per a la reproducció en temps real de continguts d'àudio i vídeo es pot utilitzar el protocol **RTSP** (*Real Time Streaming Protocol*), que estableix i controla un o diversos fluxos sincronitzats tals com àudio o vídeo. Aquests fluxos poden incloure fonts de dades en viu o clips emmagatzemats. Pot treballar per sobre de RTP (no és obligatori), però també per sobre d'UDP o de TCP directament.

Les URL RSTP tenen la següent forma:

```
rtsp://media.exemple.com:554/disc/pista
rtsp://media.exemple.com:554/disc
```

En el primer cas fa referència a un arxiu (àudio o vídeo), mentre que en el segon identifica una presentació composta de fluxos d'àudio o vídeo.

Un altre protocol utilitzat és l'**RTMP** (*Real-Time Messaging Protocol*). Aquest és un protocol propietari desenvolupat inicialment per Macromedia (actualment, Adobe) per a l'*streaming* d'àudio i vídeo entre les diferents plataformes Flash. Tot i ser un protocol privat, les especificacions són públiques amb una llicència específica (*RTMP Specification License*). El port usat per defecte per a aquest protocol és el 1935.

Aquest protocol disposa de diverses versions segures:

- **RTMPE**: les dades són encriptades amb un algorisme d'encriptació conegut i se centra en la velocitat de l'encriptació (segons Adobe, requereix un 15% de procés que RTMP).
- **RTMPS**: RTMP sobre TLS/SSL.
- **RTMFP**: les dades són encriptades amb un algorisme de xifratge per blocs.

> L'***streaming* de *bitrate* adaptatiu** (*adaptive bitrate streaming*) és una tècnica actual usada en la reproducció de continguts multimèdia. Consisteix a detectar la disponibilitat d'ample de banda per part del servidor i el client per tal d'ajustar el flux (*stream*) i la seva qualitat. Treballen gairebé exclusivament sobre HTTP i estan dissenyats per treballar de forma eficient.

Entre les característiques es destaca un temps d'inici de reproducció ràpid, el poc *buffering* i una bona experiència per a diferents amples de banda. Els protocols d'*streaming* de *bitrate* adaptatiu més coneguts són:

- **HLS** (*HTTP Live Streaming*): protocol desenvolupat per Apple que inicialment permet l'*streaming* des d'un servidor web qualsevol a dispositius basats en iOS (iPhone, iPad, iPod touch, macOS i Apple TV). No obstant això, Apple ha fet públic el protocol en format de RFC (RFC 8216).

- **HDS** (*HTTP Dynamic Streaming*): protocol desenvolupat per Adobe que permet l'*streaming* des d'un servidor web qualsevol a clients que suportin la reproducció de contingut Flash. Les especificacions d'aquest protocol estan publicades en la pàgina web d'Adobe.
- **Smooth Streaming**: protocol propietari de Microsoft per a l'*streaming* de contingut multimèdia. Aquest protocol s'instal·la com a extensió al servidor web de Microsoft, l'Internet Information Services (IIS). Els clients han de suportar Silverlight, *plugin* per als navegadors web que permet la visualització d'aquests continguts.
- **MPEG-DASH** (*Motion Pictures Expert Group Dynamic Adaptive Streaming over HTTP*): protocol de codi obert que intenta solucionar els inconvenients dels anteriors protocols (llicències, clients específics, etc.).

D'altra banda, es pot classificar la manera com es reprodueixen els continguts multimèdia en funció del seu origen:

- **Reproducció en temps real en directe**: permet veure esdeveniment que estan succeint en el mateix moment en què s'estan reproduint els continguts. Tots els clients veuen i escolten el mateix.
- **Reproducció en temps real a la carta** (VoD, Video On Demand): permet a l'usuari seleccionar els continguts que han estat prèviament gravats i emmagatzemats en un servidor. El que veu i escolta cada usuari és independent dels altres.

1.4.3 Difusió en temps real a adreces de multidestinació

Unicast (unidifusió): enviament d'informació a un únic destinatari.

Multicast (multidifusió selectiva): enviament d'informació a múltiples destinataris.

Broadcast (multidifusió general): enviament d'informació a tots els destinataris (d'una xarxa).

En el cas de la reproducció en temps real en directe, si cada client estableix la seva pròpia connexió amb el servei i inicia el seu propi flux (comunicació *unicast*) hi ha un ús considerable de l'ample de banda.

Una solució és la transmissió d'aquests continguts amb una comunicació multicast, en la qual aquest flux és únic i compartit per tots els participants. No obstant això, no sempre és possible aplicar aquesta tècnica.

1.4.4 Difusió en temps real de ràdio

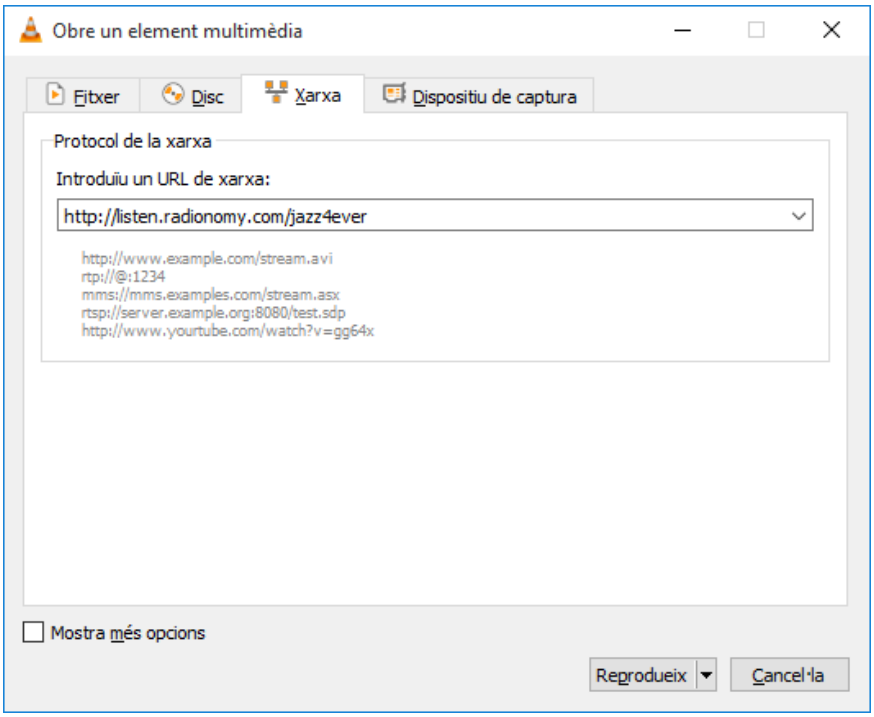
Un dels sectors que s'ha vist revolucionat i que ha experimentat bastants canvis ha estat el món de la ràdio. La mateixa forma de consumir els continguts ha variat (ara es pot escoltar per internet), i també es disposa de serveis a la carta d'àudio, bàsicament musicals, que competeixen directament amb les emissores de ràdio musicals tradicionals. Les facilitats per muntar les pròpies emissores, en les quals els costos es veuen absolutament reduïts a la infraestructura de servidor i domini

(n'hi ha de gratuïts), així com les facilitats per a l'oient d'escoltar emissores de tot el món, han fet canviar el paradigma de la radiodifusió tal com era conegut.

Per escoltar la ràdio per internet es pot fer de diverses maneres. Normalment les emissores es poden escoltar des de la mateixa pàgina web. No obstant això, a vegades es pot aconseguir la referència de la font d'àudio i reproduir-la amb algun programari amb l'ajuda de la URL.

Per exemple amb el programari VLC. Se selecciona l'opció *Obre un flux de xarxa* de l'apartat *Fitxer multimèdia*, i s'introdueix la URL (vegeu la figura 1.7). Es clica a *Reprodueix* i seguidament es començarà a escoltar l'emissora.

Figura 1.7. URL



1.5 Programari de servidors de difusió en temps real d'àudio

Com que l'*streaming* és una tecnologia relativament nova, actualment hi ha una gran diversitat de servidors d'*streaming*, alguns d'especialitzats i d'altres de propòsit general. Algunes solucions són propietàries i ofereixen continguts conjuntament. També hi ha solucions de programari lliure que tot i que a vegades no ofereixen les mateixes característiques, la majoria de vegades són suficients.

1.5.1 Servidor Darwin

QuickTime és una plataforma multimèdia d'Apple. Aquest sistema comprèn tant la part de servidor com la part de client, a més d'un format propi. Algunes vegades s'ofereix una versió de part del programari en codi obert (s'allibera) que

disposa generalment de les mateixes característiques que la versió tancada, excepte aquelles que tenen patents o llicències restrictives. És el cas de la part servidor de QuickTime, que s'anomena Darwin.

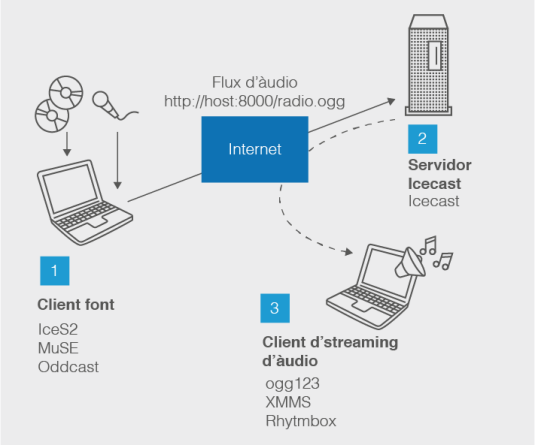
Com que el codi és obert, el servidor d'*streaming* Darwin (DSS, *Darwin Streaming Server*) disposa de versions per a Windows, Mac i Linux. Usa el protocol RTSP per a l'*streaming* multimèdia i és compatible amb els formats més actuals, com MP4 i 3GP.

1.5.2 'Streaming' d'àudio des de la consola: IceCast

IceCast és un servidor d'*streaming* multimèdia patrocinat per la fundació Xiph.Org que es pot utilitzar per crear estacions de ràdio. Està disponible per a Linux i Windows. El servidor funciona des de la línia de comandes (això facilita l'emissió de ràdio de forma desatesa), tot i que permet el monitoratge via web. Aquest programari està distribuït sota la llicència GNU GPL, versió 2.

Vegeu el funcionament en la figura 1.8.

FIGURA 1.8. Arquitectura de l'IceCast



IceCast consta dels diferents components:

- **IceCast** és el servidor. Bàsicament distribueix els fluxos d'àudio de les diferents fonts cap als diferents clients. Els fluxos d'entrada els rep a través dels punts de muntatge que s'han definit en el servidor. Els fluxos de sortida s'emeten a través d'un port especial, especificant el punt de muntatge (estació de ràdio).
- **Client font**. A través de diferents programaris (IceS, Ezstream, etc.) s'envia l'*stream* d'àudio cap al servidor per tal que sigui reemès posteriorment per la xarxa a través d'IceCast.
- Els **clients d'*streaming* d'àudio** són els clients que reproduiran l'*streaming*. Poden ser aplicacions específiques o el mateix navegador (si suporta la reproducció d'àudio).

1.5.3 Servidor Ampache

Ampache és una aplicació web dissenyada inicialment per a *streaming* de música, però que també incorpora *streaming* de vídeo. Està disponible en format de codi obert sota una llicència AGPLv3 (*GNU Affero General Public License v3*).

Entre les característiques cal destacar que disposa d'un organitzador per a la col·lecció de música a través d'una senzilla interfície web i que es pot escoltar l'*streaming* a través de la majoria de reproductors i en diferents dispositius: ordinador, telèfon i TV.

1.5.4 Subsonic/LibreSonic/Airsonic

Subsonic és un servidor web d'*streaming* multimèdia força popular desenvolupat en Java. Inicialment de codi obert, el codi va ser tancat a partir de la versió 6. A partir de la darrera versió del codi obert es va crear un *fork* anomenat LibreSonic, però per posteriors discussions entre els col·laboradors van acabar creant un segon *fork* anomenat AirSonic.

Fork

Creació d'un projecte paral·lel i que evoluciona de forma deslligada a partir d'aquest punt. És comú en projectes de programari lliure a causa de desavinences en els objectius, el lideratge, etc.

Com que està basat en Java, es pot executar en una multitud de plataformes. Libresonic i Airsonic són programari lliure i es distribueixen sota la llicència GPLv3 (*GNU General Public License v3*).

1.5.5 GNUMP3d

GNUMP3d és un altre servidor d'*streaming* per a àudio i vídeo. El seu codi és obert i forma part del programari GNU mantingut per la *Free Software Foundation* (FSF).

Free Software Foundation

La *Free Software Foundation* (FSF) és una organització sense ànim de lucre a nivell mundial que promou l'ús de programari lliure per tal de defensar el dret dels usuaris a l'ús de programari de qualitat i lliure de patents.

Segons la web oficial, les característiques principals són:

Podeu consultar les opcions de configuració del servidor consultant la secció "Annexos" del web d'aquest mòdul.

- És reduït, estable i segur.
- És senzill d'instal·lar, configurar i utilitzar.
- És portable a diferents varietats de Unix/Linux i plataformes Windows.

2. Instal·lació i administració del servei de vídeo

Els serveis de vídeo han revolucionat el panorama actual de la indústria de la televisió i del cinema. Un dels grans iniciadors d'aquest canvi de paradigma ha estat sens dubte YouTube, ja que la popularització d'aquest servei ha permès que l'usuari final sigui també creador de continguts.

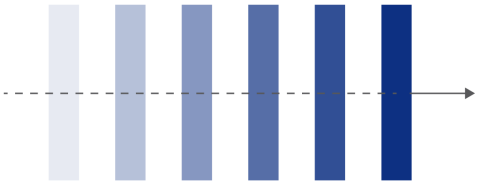
No obstant això, també grans empreses o noves han començat a crear plataformes de continguts (Netflix, Amazon, Apple, Disney, etc.) tot disputant-se una part important del pastís de l'entreteniment mundial.

A part de tota aquesta concurrència, cal tenir en compte una dada important: una gran part de l'ample de banda d'internet es deu al consum de tots aquests serveis d'*streaming* de vídeo.

2.1 Vídeo digital

El principi bàsic del vídeo digital és semblant a l'analògic, un conjunt d'imatges en sèrie, és a dir, una darrera l'altra per tal de produir una sensació de moviment si es passen prou ràpidament (vegeu la figura 2.1). Aquestes imatges s'anomenen **fotogrames** (en anglès, *frames*).

FIGURA 2.1. Exemple de seqüència de fotogrames



En l'apartat "Còdecs de vídeo" s'explica el funcionament de la codificació del vídeo i es tracten els diferents còdecs que hi ha.

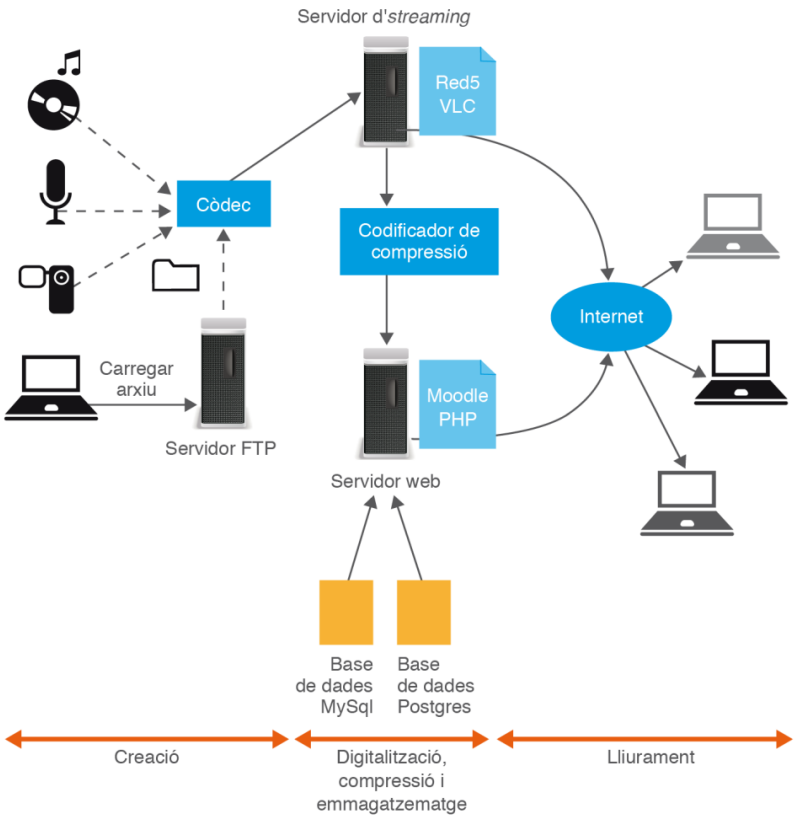
El problema d'aquest sistema és que encara que les imatges estiguin comprimides, la suma total d'imatges és molt gran i fa que la mida del fitxer de vídeo no sigui apta per al maneig o la transmissió. Aquí és on entren els còdecs (codificadors-descodificadors), que redueixen considerablement la mida d'aquests fitxers de vídeo.

Un altre concepte important per tal de produir la sensació de moviment en anar passant fotogrames un darrere l'altre és la freqüència, és a dir, quantes imatges es visualitzen per unitat de temps. És el que s'anomena velocitat de reproducció (*frame rate*), i la seva unitat de mesura són els fotogrames per segon (*frames per second*, FPS). El seu valor pot variar, però normalment s'usa la xifra de 24 FPS per a la reproducció de vídeo digital, tot i que també s'usen altres valors. Per

exemple, per a animacions senzilles s'usen 12 FPS i per a televisió, depenent de la part del món, s'usen 25 o 30 FPS, ja que s'aprofita la freqüència de la xarxa elèctrica (alterna), i l'electrònica per fer funcionar aquests aparells pren aquesta freqüència com a referència.

En la figura 2.2 es mostra l'arquitectura bàsica d'un servei de vídeo, el qual inclou tres fases principals: la creació de continguts, l'emmagatzematge i la distribució.

Figura 2.2. Arquitectura del sistema



2.1.1 Formats d'imatge

Els dos principals tipus d'imatges digitals són el mapa de bits i les imatges vectorials.

Es defineix la **profunditat** del color com el nombre de bits necessaris per codificar un color. Com més bits codifiquin un color, més qualitat tindrà una imatge.

Píxel
Unitat mínima (indivisible) d'una imatge digital. Cada píxel representa un color.

Les imatges de **mapa de bits** consisteixen bàsicament en un conjunt de píxels formant una imatge rectangular on en cada píxel s'hi especifica un color. Depenent de la quantitat de bits que s'usen per codificar el color (profunditat), s'obté un nombre màxim de colors disponibles per a aquella imatge i en determina la qualitat. Per exemple, si s'usa una profunditat de 8 bits, es poden tenir com a màxim 256

colors; en canvi, amb 24 bits s'obté una gamma de més de 16 milions de colors. Aquests tipus d'imatges s'usen en fotografia, tractament d'imatges, etc.

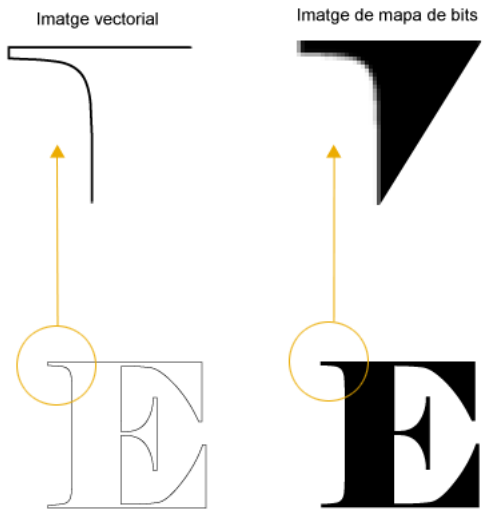
Vector

Representació del desplaçament entre dos punts en un espai euclidià.

Les imatges **vectorials** consisteixen en un conjunt de vectors que defineixen formes geomètriques. Aquests tipus de gràfics són escalables, és a dir, que es poden augmentar i reduir sense perdre qualitat (a diferència del mapa de bits), ja que les imatges es reconstrueixen amb els vectors donats i un factor de multiplicitat. Aquests tipus d'imatges són molt utilitzades en la cartografia, el disseny CAD i la tipografia.

Vegeu en la figura 2.3 la diferència en l'escalabilitat d'ambdós tipus:

FIGURA 2.3. Imatges: vectorial i en mapa de bits



Els formats d'imatges més estesos són:

- **JPEG** (*Joint Photographic Experts Group*): format gràfic de compressió amb pèrdues, però que manté una altra qualitat. El paràmetre de compressió es pot ajustar permetent tenir un balanç qualitat/mida desitjat. Les imatges tenen una profunditat de 24 bits i es fa servir en fotografia digital.
- **GIF** (*Graphics Interchange Format*): format gràfic de compressió sense pèrdua amb una profunditat de 8 bits. L'algorisme de compressió és propietari d'Unisys. Permet gràfics animats.
- **PNG** (*Portable Network Graphics*): format gràfic lliure de compressió sense pèrdua. Les seves especificacions es troben en l'RFC 2083. Va aparèixer com a alternativa al GIF.
- **BMP** (*Windows BitMaP*): format gràfic sense compressió desenvolupat per Microsoft. Permet diverses profunditats de color (fins a 24 bits) i la mida del fitxer és bastant considerable en funció d'aquesta profunditat, la qual cosa fa que no s'utilitzi per a transmissió de dades.
- **TIFF** (*Tagged Image File Format*): format gràfic creat per Aldus, que va ser adquirida posteriorment per Adobe. Aquest format permet la compressió de dades i manté tota una sèrie de metadades (principalment geomètriques) que

facilita l'edició i la impressió de la imatge. És un format molt popular en el món del disseny gràfic. No obstant això, aquest format requereix molt d'espai.

- **SVG** (*Scalable Vector Graphics*): format gràfic realitzat en XML que permet definir gràfics vectorials en dues dimensions. És un estàndard obert desenvolupat pel W3C (World Wide Web Consortium).

Tots els formats d'imatge són del tipus mapa de bits excepte l'SVG, que és vectorial.

2.1.2 Formats contenidors de vídeo

De forma general, un vídeo no conté només imatges, sinó que també incorpora àudio. Fins i tot, pot contenir algun altre tipus de contingut multimèdia, com per exemple subtítols. El fitxer que emmagatzema totes aquestes dades s'anomena contenidor.

Un **format contenidor** (o, simplement, format) és un tipus de fitxer que pot contenir diversos tipus de dades.

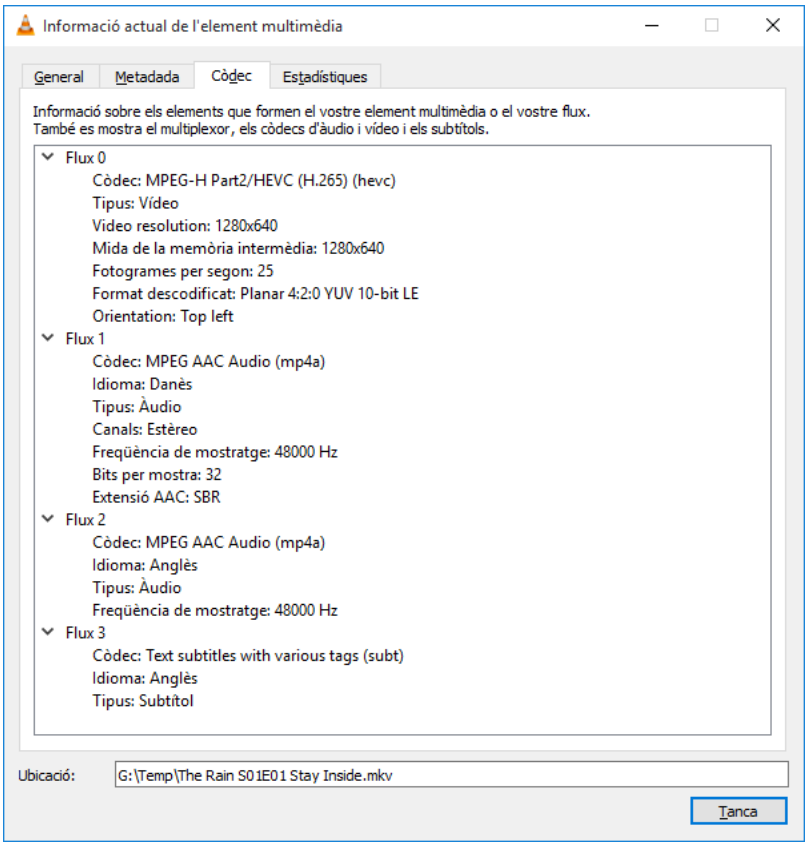
En els contenidors de vídeo, el més normal és que a dins hi hagi també dades d'àudio, de vegades múltiples. Per exemple, àudio en diferents idiomes o en diferents formats (estèreo, 5.1, etc.). Alguns també admeten la possibilitat d'incloure-hi dades de text, com per exemple els subtítols.

Els formats contenidor més coneguts són:

- **AVI** (Audio Video Interleave, intercalat d'àudio i vídeo): contenidor desenvolupat per Microsoft.
- **WMV** (Windows Media Video): contenidor de Microsoft per a determinats còdecs propietaris seus.
- **FLV** (Flash Video): contenidor específic per al Flash d'Adobe. El format és propietari i es necessita Adobe Flash Player per reproduir-lo.
- **MOV**: contenidor de QuickTime d'Apple.
- **MP4**: contenidor estàndard per a continguts multimèdia en MPEG-4.
- **MKV** (Matroska): contenidor obert i gratuït. El nom ve donat per les nines russes i la idea és que un contenidor en pot contenir d'altres alhora, i així successivament si s'escau.
- **3GPP** (3rd Generation Partnership Project): format contenidor utilitzat en telèfons mòbils de tercera generació. S'usen còdecs adaptats als requeriments mòbils (ample de banda, emmagatzematge, etc.).

En la figura 2.4 podeu veure els còdecs utilitzats per un arxiu multimèdia. En concret, hi ha 4 fluxos: un de vídeo, dos d'àudio i un de subtítols.

FIGURA 2.4. Diferents fluxos



2.1.3 Còdecs de vídeo

Una manera molt senzilla d'emmagatzemar un senyal de vídeo és en forma d'imatges, que s'anomenen fotogrames (*frames*). L'ull humà no és capaç de distingir les imatges a partir d'una freqüència de 25 FPS (fotogrames per segon) i té la sensació de continuïtat entre les imatges, és a dir, no percep salts d'una imatge a l'altra i el que veu és moviment entre una imatge i l'altra. Encara que la imatge estigui comprimida, la suma del que ocupen totes les imatges és molt considerable. Si, a més, aquest senyal de vídeo s'ha de transmetre en temps real a través de la xarxa, el flux de dades per unitat de temps que hauríem de transmetre és pràcticament inviable per a resolucions actuals.

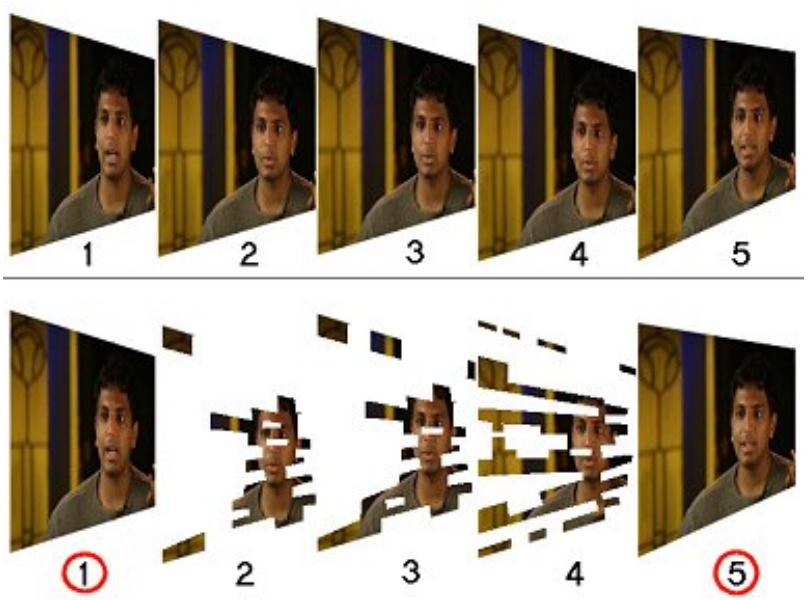
Per tal de reduir la quantitat de dades transmeses o emmagatzemades, igual que en els serveis d'àudio, les dades es codifiquen per tal de comprimir-les i poder reduir l'ample de banda o espai utilitzat.

El principi bàsic en què es fonamenten la majoria de còdecs de vídeo és a emmagatzemar les diferències d'un *frame* a un altre, és a dir, de l'anterior al següent, en les quals en la majoria de les vegades varia un percentatge molt petit de la imatge (imaginem una persona parlant: a grans trets, la part on hi ha més canvis és a la cara). Això fa que es redueixi moltíssim la mida d'un fitxer.

Evidentment hi ha certs refinaments, com que cada cert temps tornem a tenir tota la imatge. Si no fos així, per reconstruir un fotograma en concret necessitaríem

un munt de fotogrames anteriors. En la figura 2.5 podeu veure un exemple dels diferents fotogrames que s'emmagatzemen en un fitxer de vídeo. Els fotogrames sencers s'anomenen fotogrames clau (*key frames*) o **I-frames** (*intra-coded frame*), i els altres s'anomenen *delta frames*. D'aquests n'hi ha de dos tipus: **P-frame** (*predicted picture*) i **B-frame** (*bidirectional predicted picture*). En els *P-frame* s'usen els fotogrames anteriors per reconstruir la imatge, mentre que en els *B-frame* s'usen els anteriors i posteriors. Aquests darrers permeten encara estalviar més espai de disc a costa de més complexitat en la descodificació.

FIGURA 2.5. Fotogrames clau



Els còdecs de vídeo més usats són:

ITU

La International Telecommunication Union és un organisme de les Nacions Unides encarregat de la regulació de les telecomunicacions a escala mundial, entre diferents administracions i proveïdors.

ISO/IEC

La International Organization for Standardization és és una organització no governamental que es dedica a l'elaboració d'estàndards a nivell mundial. La International Electrotechnical Commission és una organització internacional que es dedica a la normalització en els camps elèctric i electrònic i tecnologies relacionades. Moltes normes han estat desenvolupades conjuntament per les dues organitzacions.

- **H.264**: desenvolupat conjuntament per la ITU i l'ISO/IEC. Aquest còdec es troba a les especificacions de l'ISO/IEC com a **MPEG-4, Part 10**, tot i que també se'l coneix amb el nom més genèric d'**AVC** (Advanced Video Codec). H.264 té un rendiment superior que els altres, però amb un consum més alt de CPU. Una implementació amb llicència GPL d'aquest estàndard és el x264.

- **H.265**: també conegut com a **HEVC** (High Efficiency Video Coding). Aquest ha estat dissenyat per ser el successor del H.264. Desenvolupat també per la ITU i l'ISO/IEC, aconsegueix codificar amb un *bitrate* menor que l'H.264, tot mantenint la qualitat. Una implementació d'aquest estàndard és el x265, amb una llicència doble (GPLv2 i comercial).

- **AV1** (AOMedia Video 1): còdec obert i lliure de *royalties* desenvolupat per l'Alliance for Open Media (AOMedia) que inclou companyies com Amazon, Cisco, Google, Intel, Microsoft, Mozilla, Netflix, etc. Competeix directament amb el còdec HVEC.

- **VP6**: còdec propietari desenvolupat per On2 Technologies. És el que s'usa en Adobe Flash Player.

- **VC-1**: és un estàndard de compressió de vídeo SMPTE (Society of Motion Picture and Television Engineers), concretament l'**SMPTE 421M**. Una implementació d'aquest estàndard és el WMV9 (Windows Media Video) de Microsoft.
- **MPEG-4 Part 2**: desenvolupat per l'ISO/IEC sota la designació d'**ISO/IEC 14496-2**. Implementacions d'aquest estàndard són els populars DivX i XviD.
- **MPEG-2 Part 2**: còdec que també rep el nom de **ITU H.262**, **ISO/IEC 13818-2** i **MPEG-2 Video**. S'usa en els DVD, en els SVC (Super VCD) i en els sistemes de distribució per cable.
- **DV** (Digital Video): còdec de vídeo utilitzat principalment per les càmeres de vídeo digital domèstiques.

Altres còdecs no tan utilitzats són:

- **H.261**: desenvolupat per la ITU, és dels més antics. S'usa per donar compatibilitat amb productes anteriors.
- **H.263**: desenvolupat per la ITU, és un estàndard de compressió dissenyat per a comunicacions amb un *bitrate* baix.
- **MPEG-1 Part 2**: s'usa per als CD de vídeo (VCD). La qualitat és semblant a la d'un vídeo analògic VHS.
- **RealVideo**: desenvolupat per RealNetworks, aquest còdec propietari ja no gaudeix de tanta popularitat.

2.2 Servidors de vídeo

Quan parlem de servidors de vídeo ens referim a un ample espectre de serveis, bàsicament serveis que permeten la transmissió de vídeo i àudio. La font d'aquests continguts pot ser heterogènia, des de càmeres i micròfons fins a continguts emmagatzemats en un disc dur.

Dintre d'aquest ventall de serveis hi ha els circuits tancats de televisió (CCTV) (*Closed Circuit Television*), normalment destinats a temes de seguretat, que avui en dia, i gràcies a internet, es poden connectar mitjançant el protocol TCP/IP.

No obstant això, els serveis de vídeo que s'han popularitzat més recentment són els servidors de reproducció en temps real o *streaming*, que moltes vegades engloben tota una plataforma tecnològica que ofereix una gran diversitat de serveis.

2.2.1 Servidors de continguts continus

Els servidors de continguts en temps real no difereixen gaire dels servidors d'*streaming* d'àudio. El principi bàsic és el mateix: anar reproduint el vídeo a mesura que es va descarregant, sense passar per cap emmagatzematge local. Tot això, amb l'ajuda d'una memòria intermèdia anomenada *buffer*, permet compensar les petites variacions del flux de dades.

Vegeu a "Reproducció en temps real d'informació multimèdia" de l'apartat dedicat a "Instal·lació i administració del servei d'àudio" els protocols usats per a la retransmissió dels continguts digitals.

Darrerament, cada cop es fan servir més protocols en ***streaming* de *bitrate* adaptatiu** (*adaptive bitrate streaming*) per tal de detectar la disponibilitat d'ample de banda per part del servidor i el client, i així ajustar el flux (*stream*) i la qualitat. Un bon exemple d'aquests protocols són HLS (HTTP Live Streaming), HDS (HTTP Dynamic Streaming), Smooth Streaming i MPEG-DASH (Motion Pictures Expert Group Dynamic Adaptive Streaming over HTTP).

2.2.2 Descàrrega progressiva ('pseudostreaming')

Una tècnica diferent de la reproducció en temps real és la descàrrega progressiva, que consisteix a anar descarregant parts d'un fitxer multimèdia perquè, un cop descarregades, es reprodueixin tot seguit, produint el mateix efecte de flux o *streaming*.

El **pseudostreaming** (també anomenat descàrrega progressiva) és una tècnica basada en Flash que consisteix a descarregar un fitxer multimèdia en parts petites que ja es poden anar reproduint. Una de les característiques més destacades és que permet cercar una posició concreta de l'arxiu multimèdia.

Un exemple típic de posicionament seria:

```
1 http://exemple.com/video.mp4?start=212.34
```

Els fitxers que permeten fer aquest posicionament han de contenir les metadades que aporten informació per a la indexació del fitxer. Si el fitxer no aporta aquestes dades, alguns servidors la poden recrear, però a costa de carregar el servidor. Per tant, és una bona idea preparar els continguts multimèdia per al *pseudostreaming*.

Una altra diferència amb l'*streaming* (anomenat també *streaming* pur, per diferenciar-lo) és que el *pseudostreaming* emmagatzema de forma temporal el contingut multimèdia que es rep, mentre que l'*streaming* pur es reprodueix directament sense passar pel disc dur.

Aquesta tècnica usa el protocol HTTP i la majoria dels servidors web (Apache, IIS, Lighttpd, Nginx) la suporten, normalment a través d'algun mòdul. També la suporten diverses plataformes de continguts digitals, entre les quals hi ha YouTube.

YouTube és una plataforma per a la compartició de vídeos creada el 2005 per Chad Hurley, Steve Chen i Jawed Karim, tres extreballadors de la companyia PayPal. El primer vídeo pujat va ser *Me at the zoo*, per un dels fundadors. El mateix any, l'empresa ja va rebre una injecció de capital important i va experimentar un creixement vertiginós fins a l'octubre del 2006, quan va ser comprada per Google. Amb aquesta adquisició, YouTube es protegia amb millors aliats de possibles infraccions de drets d'autor. El creixement ha continuat i s'ha diversificat molt l'oferta, oferint tot un ventall de serveis que ha vegades són específics per a països (per tal d'adaptar-se a les diferents legislacions).

Pel que fa a la seva infraestructura, tot i que inicialment tenia la seva pròpia, ara està integrada en els *data centers* de Google com a un més dels seus serveis. El sistema operatiu és un Debian modificat migrat progressivament d'un RedHat a partir del 2013. El servidor web és un servidor propi, el *Google Web Server* (o *gws*), que es creu que és una versió modificada d'Apache, tot i que Google no dona cap informació.

Si es fa un *nmap* al lloc web de YouTube es pot veure que dona servei als ports 80 (HTTP) i 443 (HTTPS), i es pot comprovar que el servei web ja es cataloga com a *gws*.

```
# nmap -T4 -A youtube.com
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-16 08:03 Romance Standard
    Time
Nmap scan report for youtube.com (172.217.168.174)
Host is up (0.23s latency).
rDNS record for 172.217.168.174: mad07s10-in-f14.1e100.net
Not shown: 998 filtered ports
PORT    STATE SERVICE   VERSION
80/tcp  open  http      gws
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.0 200 OK
|     Date: Sat, 16 Nov 2019 07:03:54 GMT
<retallat>
| http-server-header:
|   YouTube Frontend Proxy
|_  gws
|_http-title: Did not follow redirect to https://youtube.com/
443/tcp open  ssl/https gws
| http-server-header:
|   YouTube Frontend Proxy
|_  gws
|_http-title: Did not follow redirect to https://www.youtube.com/
| ssl-cert: Subject: commonName=*.google.com/organizationName=Google LLC/
    stateOrProvinceName=California/countryName=US
| Not valid before: 2019-11-05T07:46:16
|_Not valid after:  2020-01-28T07:46:16
|_ssl-date: 2019-11-16T07:04:16+00:00; -3s from scanner time.
<retallat>

OS and Service detection performed. Please report any incorrect results at
    https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 70.22 seconds
```

Vegeu "Subscripció d'àudio" de l'apartat dedicat a "Instal·lació i administració del servei d'àudio".

2.2.3 Subscripció de vídeo

La subscripció de vídeo tampoc no és gaire diferent de la d'àudio. Es fa a través de la sindicació.

La **sindicació** és una tècnica que permet la descàrrega automàtica d'arxius, que poden ser d'àudio, vídeo, notícies, text o imatges a través de diferents estàndards, com pot ser el RSS o l'Atom. No obstant, en el cas del vídeo molts d'aquests arxius no són realment descarregats, sinó que es descarrega l'URL per posteriorment poder visualitzar el vídeo en *streaming*.

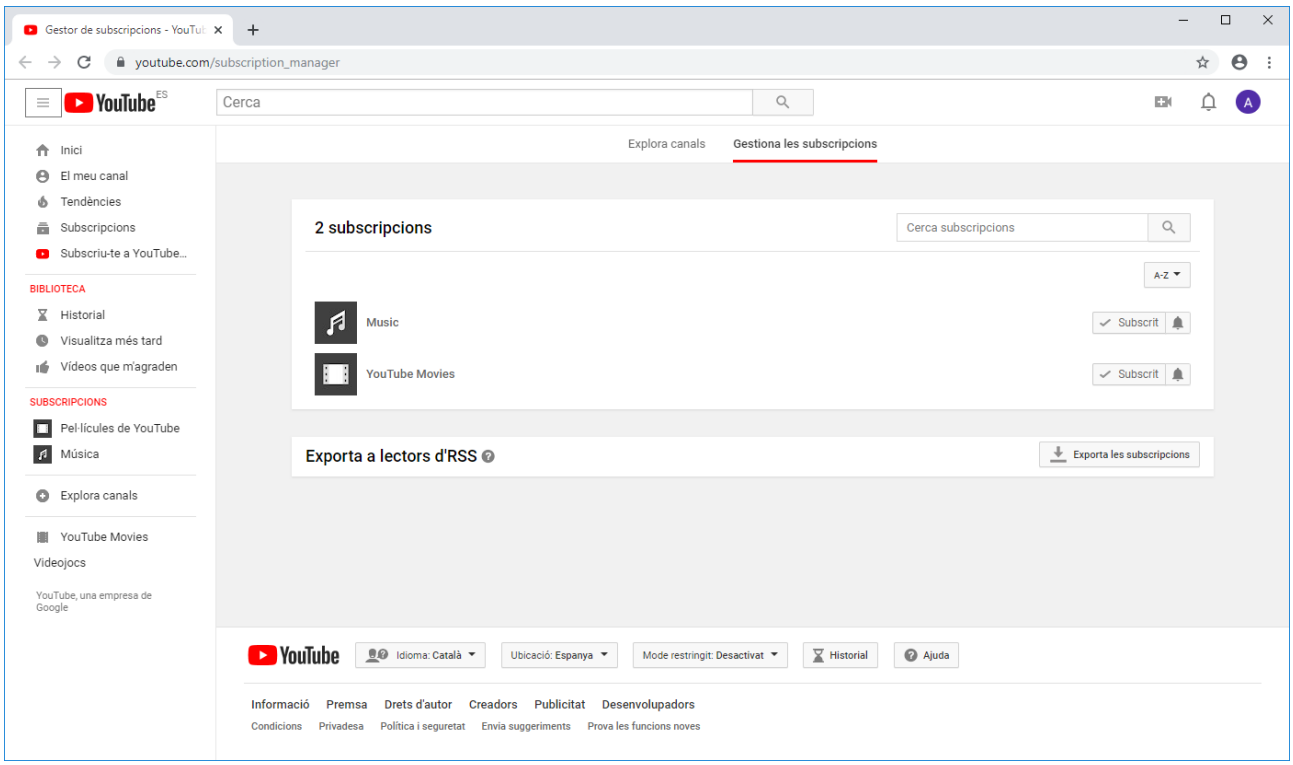
JSON
Format d'intercanvi de dades més lleuger que l'XML i més senzill d'interpretar.

Això fa que algunes plataformes disposin de formats propis. Per exemple, les subscripcions a YouTube utilitzen un format propi basat en **JSON** (JavaScript Object Notation), que es pot consultar a la documentació de l'API de Google (igual per a les llistes de reproducció). Malgrat tot, es permet exportar totes les subscripcions en format **OPML** (Outline Processor Markup Language), que posteriorment es pot importar en una altra aplicació, com per exemple Feedly.

OPML és un format de fitxer XML que permet importar/exportar totes les subscripcions d'un determinat canal a partir de les seves sindicacions (RSS majoritàriament).

En la figura 2.6 podeu veure l'administrador de subscripcions de Youtube. En l'apartat *Exporta a lectors d'RSS* es poden descarregar les subscripcions en format OPML.

FIGURA 2.6. Administrador de subscripcions de Youtube



2.3 Videoconferències

Un dels altres serveis de vídeo que ha augmentat molt el creixement ha estat el de les videoconferències, gràcies a l'increment de l'ample de banda que ha experimentat internet. En aquest cas no és un usuari final qui reprodueix un senyal de vídeo, sinó que tant l'emissor com el receptor són generadors i consumidors. Això fa que els còdecs utilitzats també hagin de codificar en temps real (en els casos anteriors només calia que la descodificació fos en temps real, permetent optimitzar més els còdecs, ja que no importava si el temps de codificació era més gran).

La **videoconferència** és la comunicació de vídeo i al mateix temps so. La videoconferència és un mitjà pel qual els individus a distància es poden trobar cara a cara i en temps real a través de la pantalla. Es necessita un ordinador amb un programari, una càmera de vídeo i una connexió a internet de bona qualitat.

El primer servei de videotelefonia públic es va utilitzar per comunicar les oficines de correu alemanyes (Reichpost) el 1936 entre Berlín i Leipzip, cobrint una distància d'uns 160 km, i es va utilitzar cable coaxial. No obstant això, el primer servei comercial que es va presentar va ser un producte d'AT&T, el Picturephone, el 1964, dins el marc d'una fira a Nova York. No obstant això, aquest servei no va tenir gaire èxit a causa de les dificultats tècniques i l'alt cost de l'època.

Amb l'arribada de l'era digital, i l'augment d'internet tant pel que fa a nombre d'usuaris com a la capacitat de transmissió, juntament amb el desenvolupament de còdecs que permetien reduir considerablement la mida dels paquets enviats a través d'aquesta xarxa, va fer possible l'aparició de programari de videoconferències, fins i tot en l'àmbit d'usuari domèstic. Una de les aplicacions que va fer molt popular el servei de videotrucades és Skype, vigent encara avui en dia, tot i que amb altres competidors (Google Hangouts, Whatsapp, etc.).

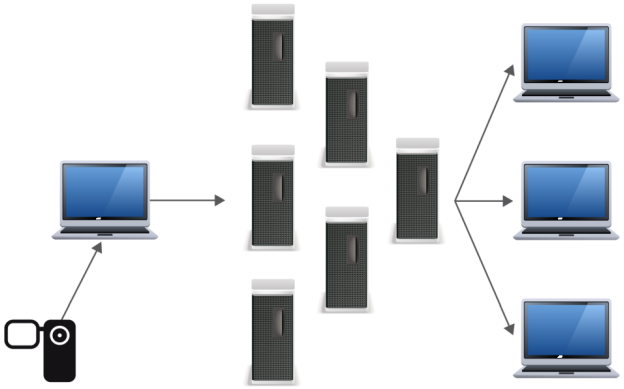
2.3.1 Funcionament

La principal funció de les videoconferències és permetre la comunicació a un determinat nombre de persones que poden estar ubicades en diferents llocs a través de la comunicació simultània i en temps real de serveis d'àudio i de vídeo, tot i que pot incloure altres serveis com xat, transmissió de documents, etc. D'aquesta manera, es permet que la situació geogràfica no sigui un impediment per a reunions de qualsevol tipus, permetent així un estalvi econòmic important tant per a les empreses com per als individus en particular.

En la figura 2.7 podeu veure un esquema del funcionament de les videoconferències. Els elements que intervenen als extrems són dispositius que incorporen com

a mínim una càmera i un micròfon per a l'emissió de les dades i uns altaveus i una pantalla per a la recepció. Els equips intermedis són els que s'encarreguen de transportar les dades.

FIGURA 2.7. Funcionament de les videoconferències



Depenent del nombre de participants en una videoconferència, les videoconferències són punt a punt o multipunt:

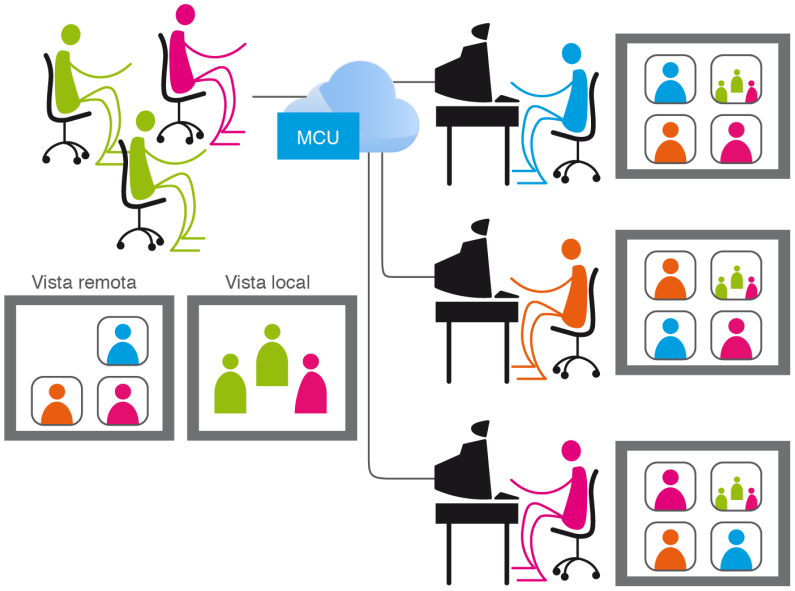
- **Punt a punt**: és la forma més simple de videoconferència. Bàsicament un usuari busca en el seu directori un altre usuari i inicia una trucada per fer la videoconferència (vegeu la figura 2.8). Actualment es poden fer des de qualsevol tipus de dispositiu amb connexió a internet: ordinador, telèfon intel·ligent, tauleta, etc. Un exemple seria una trucada d'Skype o una trucada de vídeo de Whatsapp.

FIGURA 2.8. Modalitat punt a punt

- **Multipunt**: quan hi ha més de dos participants el sistema requereix algun dispositiu que actuï com a nucli per tal de rebre i redistribuir el

senyal. Aquests dispositius s'anomenen unitat de control multipunt (MCU) (*Multipoint Control Unit*), i poden ser tant de programari com de maquinari (vegeu la figura 2.9).

Figura 2.9. Modalitat multipunt



El tipus de videoconferència multipunt es pot desglossar en dos grans subtipus:

Webinar ('web-based seminar')

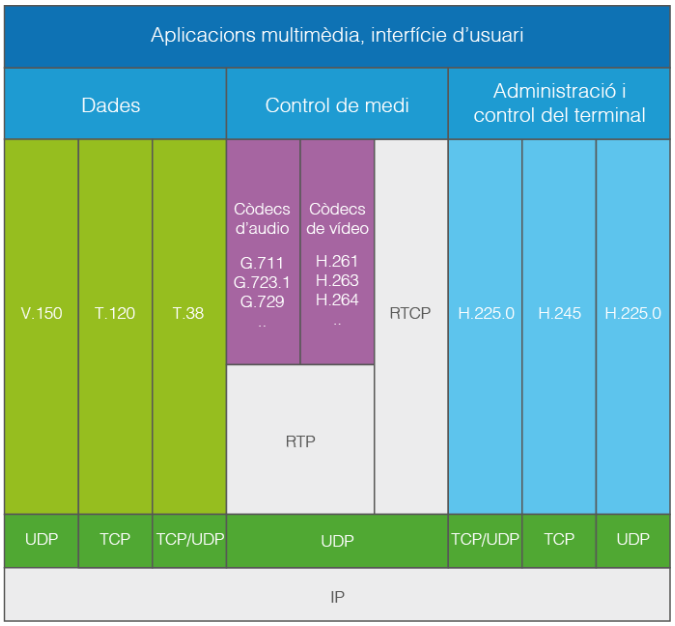
Videoconferència de múltiples participants bàsicament encarada a la formació (cursos) on es dona la possibilitat als assistents d'interactuar amb la resta de participants. S'aplica a la formació a distància per tal d'augmentar-ne la qualitat.

- **Videoconferència programada**. Aquest tipus de videoconferència permet a l'usuari programar i organitzar els recursos. També poden anar vinculats a algun sistema de calendari, tant al núvol com en una aplicació. Un exemple són els webinars.
- **Videoconferència sense reserva**. És l'alternativa a la videoconferència programada, que permet la llibertat de crear una videoconferència de forma immediata sense especificar els participants ni la duració. Un exemple és Google Hangouts (tot i que permet videoconferències punt a punt).

El principals conjunts de normatives/estàndards que es fan servir per a les videoconferències són l'estàndard H.323 i el protocol SIP:

- **H.323**: normativa realitzada per la ITU (International Telecommunication Union) el 1996 que fixa els estàndards per a la comunicació de veu i vídeo en temps real sobre el protocol IP (admet altres protocols de xarxa). Es fixen els estàndards de l'establiment de la trucada, la compressió d'àudio i vídeo i la gestió de l'ample de banda, i utilitza els protocols RTC i RTCP per a la transferència dels continguts multimèdia. En la figura 2.10 podeu veure alguns dels protocols que formen part de l'especificació H.323:

FIGURA 2.10. Pila de protocols de l'especificació H.323



- **SIP** (*Session Initiation Protocol*): és un protocol més senzill desenvolupat per l'IETF MMUSIC (Multiparty Multimedia Session Control). És un protocol de senyalització per a la gestió de sessions en temps real que incloguin àudio i vídeo, entre d'altres. Les seves especificacions es troben definides en l'RFC 3261. SIP treballa amb altres protocols com l'SDP (Session Description Protocol), que és usat per descriure la inicialització d'una sessió multimèdia, RTP i RTCP.

2.3.2 Programari

En els darrers temps hi ha hagut una proliferació de programari per a la realització de videoconferències. Molts són solucions particulars o propietàries (Cisco Webex, Adobe Connect, Zoom, etc.) que ofereixen diferents plans depenent del nombre de participants, la necessitat d'emmagatzematge o altres característiques.

No obstant això, també hi ha un conjunt important de programari de codi obert que porta a terme les principals tasques. Alguns són de propòsit general i d'altres estan encarats a solucions en particular, com per exemple l'ensenyament a distància.

GDPR
Reglament de la UE per tal unificar tots els criteris per a la protecció de dades. Inclou un marc regulador per a les multinacionals i per a la transferència fora de la UE.

Un altre aspecte que cal tenir en compte a l'hora d'escollir el programari de videoconferències és que compleixi el Reglament General de Protecció de Dades de la Unió Europea (**GDPR**).

Learning Management System (LMS)
Programari dissenyar per a la gestió integral de cursos en línia.

Algunes de les plataformes de codi lliure més destacades són les següents:

- OpenMeetings: es distribueix amb llicència Apache i permet fer videoconferències, edició col·laborativa de documents, missatgeria instantània, pissarra i compartició d'escriptori.

- BigBlueButton: es distribueix amb llicència LGPL i ofereix videoconferències, presentacions, missatgeria instantània, etc. Està encarat a l'aprenentatge en línia i alguns LMS com Moodle tenen *plugins* per a la seva integració.
- Jitsi: es distribueix amb llicència Apache i el seu punt fort és la varietat de plataformes en que està disponible.