Configuració i administració de protocols dinàmics

Eduard García Sacristán

Índex

Introducció

La divisió de les xarxes en mides depenent de les diferents classes (A, B i C) és molt estricta. Això feia que s'haguessin de contractar més xarxes de les necessàries i que aquestes estiguessin infrautilitzades.

Per solucionar aquest problema va sorgir el concepte de subxarxa, segons el qual una xarxa es pot dividir en subxarxes d'una mida més petita que poden donar serveis a diferents segments de la xarxa. A partir d'aquesta solució es va optar per flexibilitzar la mida de les xarxes en general creant subxarxes de mida variable i encaminament sense classe.

En l'apartat "Subaxarxes i CIDR" veureu com es fa a mà la divisió de subxarxes amb la mateixa mida, subxarxes de mida variable i, finalment, l'encaminament sense classe. Aquest apartat té activitats molt interessants amb les quals podreu fer exercicis típics de divisió de subxarxes (i veure'n la solució).

L'apartat "Encaminament dinàmic" tracta dels diferents algorismes i protocols d'encaminament dinàmic. En primer lloc, veureu conceptes teòrics d'encaminament i, posteriorment, alguns exemples dels protocols d'encaminament més populars en l'actualitat i com es configuren per als encaminadors.

Per treballar correctament amb aquesta unitat, seria interessant fer servir un simulador de xarxes per anar provant les diferents configuracions d'encaminadors. És recomanable que instal·leu un simulador de xarxa i aneu provant les diferents ordres que us presentem en els materials. Feu proves amb xarxes dins del simulador i intenteu canviar la configuració dels encaminadors per comprovar com canvia el funcionament de la xarxa.

Resultats d'aprenentatge

En finalitzar aquesta unitat l'alumne/a:

1. Realitza tasques avançades d'administració de xarxa analitzant i utilitzant protocols dinàmics d'encaminament.
    - Configura el protocol d'encaminament RIPv1.
    - Configura xarxes amb el protocol RIPv2.
    - Realitza el diagnòstic d'errors en una xarxa que utilitza RIP.
    - Valora la necessitat d'utilitzar màscares de longitud variable en IPv4.
    - Divideix una xarxa principal en subxarxes de diferents mides amb VLSM.
    - Realitza agrupacions de xarxes amb CIDR.
    - Habilita i configura OSPF en un encaminador.
    - Estableix i propaga una ruta per defecte usant OSPF.

1. Subxarxes i CIDR

Per tenir connectivitat IP, s'ha de tenir configurada una adreça IP en cadascuna de les interfícies dels *hosts* i els dispositius de xarxa. La ICANN (Internet Corporation for Assigned Names and Numbers, la Corporació d'Internet per a l'Assignació de Noms i Números), l'organització responsable d'assignar els espais d'adreces IP, assignava originalment espais d'adreces de classes A, B i C. Aquestes classes tenen mides molt dispars, de l'ordre de milions, de desenes de milers i 256 adreces respectivament. Per obtenir divisions de les xarxes originals, es va crear el mode de funcionament de les subxarxes, les quals poden ser de la mateixa mida o de mida variable.

El CIDR (*classless interdomain routing*, encaminament entre dominis sense classe) es va presentar com a solució a alguns dels problemes que van sorgir amb l'augment exponencial de la mida d'Internet.

1.1 Subxarxes

La divisió de l'espai d'adreces IP en diferents classes permet que es puguin demanar als ISP (*Internet service provider*, proveïdor de serveis d'Internet, és a dir, l'empresa de telecomunicacions que ens dóna accés a Internet) rangs d'adreces IP públiques corresponents a les classes existents (classes A, B i C). Amb les adreces assignades amb una d'aquestes classes, es poden cobrir les necessitats de la xarxa.

La taula 1.1 resumeix les característiques de les diferents classes de xarxes existents:

TAULA 1.1. Classes de xarxa

Classe	Valor dels primers bits	Màscara	Bits xarxa /host	Nombre de xarxes	Nombre d'adreces	Rang d'adreces
A	0--	255.0.0.0	7/24	128	16777216	0.0.0.0 a 127.255.255.255
B	10-	255.255.0.0	14/16	16384	65536	128.0.0.0 a 191.255.255.255
C	110-	255.255.255.0	21/8	2097152	256	192.0.0.0 a 223.255.255.255

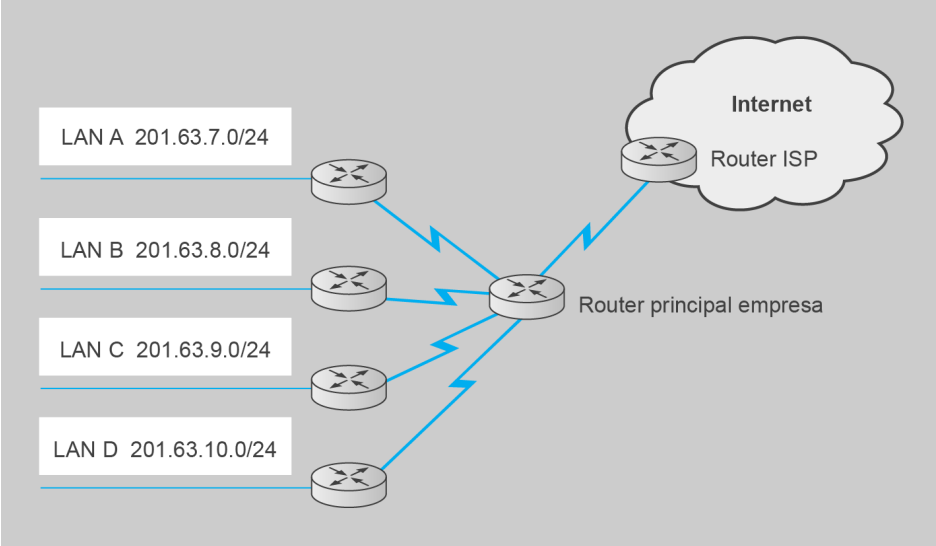
Les classes tenen diferents quantitats de bits de host i de xarxa que en determinen les característiques

Però aquesta divisió dels rangs d'adreces en classes és molt estricta i no sempre s'ajustarà a les necessitats de cada xarxa. A més a més, la diferència de mida entre les classes és molt gran. Per exemple, d'una xarxa de classe C amb 256 adreces passem a una de classe B amb 65.536 adreces. Què passa si es necessita

una quantitat d'adreces entre aquestes dues mides de xarxa?

Imagineu el cas d'una empresa que té quatre oficines diferents amb cinquanta ordinadors a les quals vol assignar xarxes distintes. Amb la forma de funcionament per classes, l'única solució que té l'empresa és demanar quatre xarxes diferents de classe C, una per cada xarxa de l'empresa, com es pot veure en la figura 1.1.

FIGURA 1.1. Empresa sense subxarxes



S'ha d'assignar una classe C diferent per cada LAN

En aquest cas, l'empresa ha demanat al seu ISP quatre xarxes de classe C, i han estat assignades les xarxes de classe C:

- 201.63.7.0/24
- 201.63.8.0/24
- 201.63.9.0/24
- 201.63.10.0/24

Tot i que amb aquesta configuració l'empresa obté les adreces necessàries per aconseguir connectivitat en les quatre LAN, aquesta solució presenta dos problemes:

- Cadascuna de les classes C assignades a les LAN està infrautilitzada, ja que s'ha assignat una xarxa amb 256 adreces IP a una LAN que en realitat únicament en necessita cinquanta. Hi ha més de dues centes adreces IP assignades que no es fan servir.
- L'encaminador de l'ISP necessita afegir quatre rutes a la seva taula de rutes per donar encaminament a l'empresa des d'Internet. Si bé en l'actualitat la capacitat de les memòries ha augmentat molt i el seu preu s'ha abaixat molt respecte a dècades anteriors, originàriament el preu de la memòria dels encaminadors era molt alt. Els encaminadors de la xarxa troncal o *backbone* (els que connecten una gran quantitat de xarxes entre elles) tenen milers de

rutes en les seves taules, i l'augment de la quantitat de rutes va resultar un greu problema al seu dia.

Per solucionar aquest problema es va crear el mecanisme de subxarxes. Les **subxarxes** constitueixen un nivell intermedi entre l'espai de xarxes i l'espai de *hosts*. Les subxarxes són cadascuna de les diferents parts en què es pot dividir una xarxa.

Les **subxarxes** permeten dividir una xarxa en porcions més petites.

En l'exemple que hem donat, l'empresa podria demanar una xarxa de classe C (per exemple, 201.63.7.0/24), i dividir-la en quatre subxarxes per assignar a les diferents oficines. Si la xarxa original de classe C tenia 256 adreces, les quatre subxarxes resultants seran de seixanta-quatre adreces. Aquesta divisió resol els dos problemes que hem plantejat més amunt:

- L'espai d'adreces IP està més ben aprofitat. En aquest cas, cada subxarxa té assignades seixanta-quatre adreces, per cinquanta ordinadors que tindran.
- Externament, per a l'encaminador de l'ISP i els encaminadors d'Internet, totes les xarxes es poden considerar com una sola (la 201.63.7.0/24) i es poden encaminar conjuntament. Únicament cal una entrada a la taula de rutes dels encaminadors per poder-se encaminar a les quatres subxarxes.

A continuació veurem com es faria aquesta divisió de la xarxa original en subxarxes.

El primer pas per treballar amb xarxes i subxarxes és passar l'adreça IP i la màscara de xarxa a la seva representació binària:

```
IP decimal: 201.63.7.0        201   .    63    .     7    .    0
IP binari:                 11001001 . 00111111 . 00000111 . 00000000

Màscara decimal:             255        255        255         0
Màscara binària:            11111111 . 11111111 . 11111111 . 00000000
```

La màscara defineix quina part de l'adreça IP correspon a la xarxa i quina part correspon al *host* de la manera següent:

- Si el bit de la màscara val 1, vol dir que el bit corresponent a l'adreça IP és de xarxa.
- Si el bit de la màscara val 0, vol dir que el bit corresponent a l'adreça IP és de *host*.

Els tres primers bytes de la màscara tenen tots els seus bits a 1 (valor decimal 255), i l'últim té tots els seus bits a 0 (valor decimal 0). Per això sabem que els tres primers bytes de l'adreça IP (201.63.7.x) corresponen a la part de xarxa, i l'últim byte (x.x.x.0) correspon a la part de *host*.

```
<————————   Xarxa   ————————> <— Host —>
IP decimal: 201.63.7.0        201   .    63    .     7    .    0
IP binari:                  11001001 . 00111111 . 00000111 . 00000000

Màscara decimal:             255        255        255        0
Màscara binària:            11111111 . 11111111 . 11111111 . 00000000
```

La xarxa original (201.63.7.0) consta de 256 adreces, equivalents a totes les combinacions dels bits corresponents a la part de *host*. Així, partint del primer valor (el valor binari 00000000, 0 en decimal) fins a l'últim (el valor binari 11111111, 255 en decimal) s'obtenen totes les adreces de la xarxa, com es pot veure en la taula 1.2.

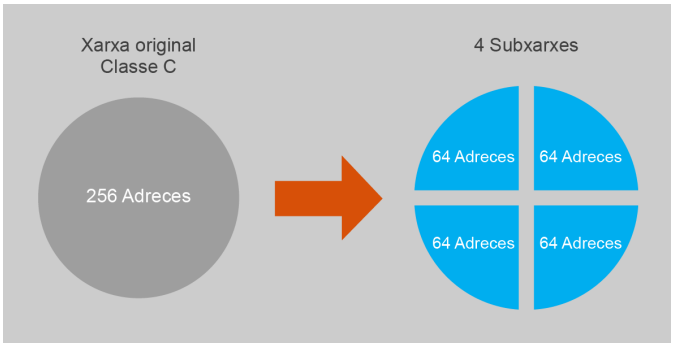
Taula 1.2. Classes de xarxa

Valor dels bits de host	Valor en decimal	Adreça IP
00000000	0	201.63.7.0
00000001	1	201.63.7.1
00000010	2	201.63.7.2
00000011	3	201.63.7.3
00000100	4	201.63.7.4
00000101	5	201.63.7.5
00000110	6	201.63.7.6
00000111	7	201.63.7.7
...	...	...
11111100	252	201.63.7.252
11111101	253	201.63.7.253
11111110	254	201.63.7.254
11111111	255	201.63.7.255

Calculant totes les combinacions dels bits de host s'obtenen totes les adreces de la xarxa

Recordeu que la primera i la darrera adreça d'una xarxa no es poden utilitzar per assignar a *hosts*, ja que corresponen a l'adreça de la pròpia xarxa i a l'adreça *broadcast* en la xarxa.

Figura 1.2. Divisió de subxarxes



Cadascuna de les subxarxes és més petita, però la quantitat total d'adreces es manté

Per fer subxarxes, el que farem és dividir l'espai de *hosts* en diferents parts. Aquestes parts, evidentment, seran individualment més petites que les originals,

però en conjunt la quantitat d'adreces es mantindrà, com es pot veure en la figura 1.2.

En quantes parts s'ha de dividir la xarxa? Analitzant l'exemple veiem que es necessiten quatre subxarxes. Quants bits es necessitaran per comptar totes les subxarxes? Es necessiten 2 bits per poder comptar quatre subxarxes, ja que $2^2 = 4$. En aquest cas les quatre subxarxes serien 00, 01, 10 i 11.

Depenent de la quantitat de subxarxes que necessiteu fer, haureu d'agafar més o menys bits per poder-les identificar. En la taula 1.3 podeu veure la quantitat de bits que es necessiten per comptar diferents quantitats de subxarxes.

TAULA 1.3. Bits necessaris per a les diferents necessitats de subxarxes

Nombre de bits de subxarxa	Quantitat de subxarxes
1	$2^1 = 2$ subxarxes
2	$2^2 = 4$ subxarxes
3	$2^3 = 8$ subxarxes
4	$2^4 = 16$ subxarxes
5	$2^5 = 32$ subxarxes
6	$2^6 = 64$ subxarxes
...	...
n	2^n subxarxes

Com més subxarxes, més bits cal fer servir per poder-les identificar

Quan es creen subxarxes, s'ha de definir una nova màscara. Com sempre, els bits a 1 de la màscara identifiquen la part de xarxa (en aquest cas, la part de xarxa + subxarxa) i els bits a 0 corresponen a la part de *host*. Així, en aquest cas, agafarem 2 bits de la part de *host* per anomenar les subxarxes, i deixarem els 6 bits restants per identificar els *hosts* dins de la subxarxa.

La primera subxarxa (la 00) es calcularia de la manera següent:

```
IP decimal: 201.63.7.0        201   .    63    .     7    .    0
IP binari:                 11001001 . 00111111 . 00000111 . 00 000000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . ss hhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de //host//

Màscara decimal:             255        255        255        192
Màscara binària:            11111111 . 11111111 . 11111111 . 11000000
```

Els tres primers bytes de l'adreça de xarxa corresponen als bits de xarxa. De l'últim byte, els dos primers bits corresponen a la identificació de la subxarxa, i els últims 6 bits a la identificació dels *hosts* dins d'aquesta subxarxa. Aquesta subxarxa té 6 bits de *host* i, per tant, tindrà un total de $2^6 = 64$ adreces. Consulteu la taula 1.4.

TAULA 1.4. Bits necessaris per a la part de host

Número de bits de host	Quantitat de hosts
1	$2^1 = 2$ *hosts*
2	$2^2 = 4$ *hosts*
3	$2^3 = 8$ *hosts*
4	$2^4 = 16$ *hosts*
5	$2^5 = 32$ *hosts*
6	$2^6 = 64$ *hosts*
...	...
n	2^n hosts

Com més bits de host, més hosts es poden identificar

Fixeu-vos que en aquest cas la màscara de xarxa (ara anomenada *màscara de subxarxa*) indica els bits de l'adreça IP que corresponen a la part de xarxa i subxarxa. Per això, de la màscara de xarxa original (255.255.255.0) amb 24 bits de xarxa i 8 bits de *host*, hem passat a la màscara de subxarxa (255.255.255.192) amb 26 bits de xarxa + subxarxa i 6 bits de *host*.

Com podeu comprovar, la taula 1.4 és idèntica a la taula 1.3. Això és així perquè de manera general:

> Amb ***n*** bits es poden comptar 2^n diferents elements.

No importa si parlem de subxarxes o de *hosts*, la fórmula per calcular la necessitat de bits és la mateixa. Per exemple: per tenir vuit subxarxes necessitarem 3 bits de subxarxa, i per tenir xarxes de 1.024 *hosts* necessitarem 10 bits de *host*.

En la taula 1.5 podeu veure un resum dels diferents valors de màscara per fer divisions en l'últim byte d'una xarxa de classe C, amb les diferents quantitats de subxarxes i els *hosts* que corresponen a cada subxarxa.

Xarxa buida?

No té sentit fer subxarxes amb una màscara 255.255.255.254, ja que, en aquestes subxarxes, únicament queden dues adreces IP. De cada subxarxa es perden dues adreces: la primera (que identifica la xarxa) i l'última (que identifica l'adreça *broadcast* en la subxarxa). Per tant, es perden les dues adreces i no queda cap adreça per assignar als *hosts*.

TAULA 1.5. Subxarxes possibles en una xarxa de classe C

Màscara	Últim byte de la màscara (en binari)	Bits de subxarxa	Nre. de subxarxes	Bits de *host*	Nre. de *hosts* en cada subxarxa
255.255.255.0	00000000	0	0	8	256
255.255.255.128	10000000	1	$2^1 = 2$	7	$2^7 = 128$
255.255.255.192	11000000	2	$2^2 = 4$	6	$2^6 = 64$
255.255.255.224	11100000	3	$2^3 = 8$	5	$2^5 = 32$
255.255.255.240	11110000	4	$2^4 = 16$	4	$2^4 = 16$
255.255.255.248	11111000	5	$2^5 = 32$	3	$2^3 = 8$
255.255.255.252	11111100	6	$2^6 = 64$	2	$2^2 = 4$
255.255.255.254	11111110	7	$2^7 = 128$	1	$2^1 = 2$
255.255.255.255	11111111	8	$2^8 = 256$	0	0

La quantitat de hosts total de totes les subxarxes es manté en tots els casos en 256 adreces.

En la taula 1.6 podeu veure totes les subxarxes i màscares que es poden definir per a una classe B.

TAULA 1.6. Subxarxes possibles en una xarxa de classe B

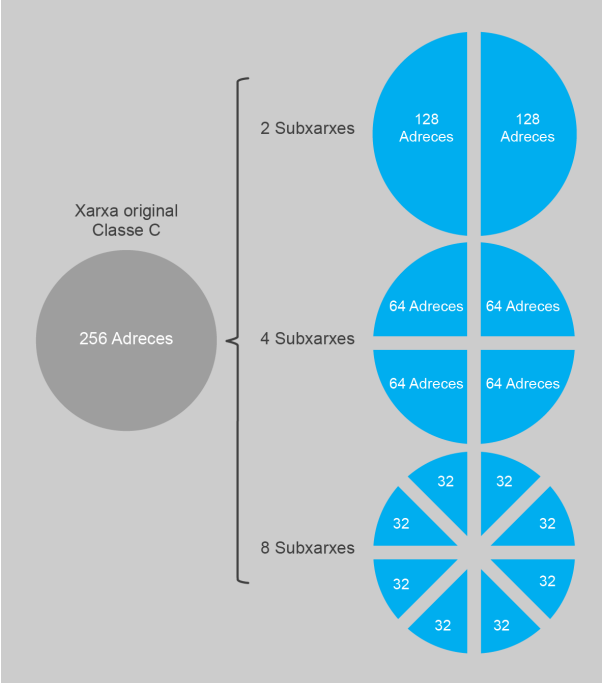
Màscara	Últims bytes de la màscara (en binari)	Bits de subxarxa	Nre. de subxarxes	Bits de *host*	Nre. de *hosts* en cada subxarxa
255.255.0.0	00000000 00000000	0	0	16	65536
255.255.128.0	10000000 00000000	1	2	15	32768
255.255.192.0	11000000 00000000	2	4	14	16384
255.255.224.0	11100000 00000000	3	8	13	8192
255.255.240.0	11110000 00000000	4	16	12	4096
255.255.248.0	11111000 00000000	5	32	11	2048
255.255.252.0	11111100 00000000	6	64	10	1024
255.255.254.0	11111110 00000000	7	128	9	512
255.255.255.0	11111111 00000000	8	256	8	256
255.255.255.128	11111111 10000000	9	512	7	128
255.255.255.192	11111111 11000000	10	1024	6	64
255.255.255.224	11111111 11100000	11	2048	5	32
255.255.255.240	11111111 11110000	12	4096	4	16
255.255.255.248	11111111 11111000	13	8192	3	8
255.255.255.252	11111111 11111100	14	16384	2	4
255.255.255.254	11111111 11111110	15	32768	1	2
255.255.255.255	11111111 11111111	16	65536	0	0

La quantitat total de hosts de totes les subxarxes es manté en tots els casos en 65.536 adreces.

Independentment de les subxarxes que es creïn, la quantitat d'adreces IP es manté, com s'exemplifica en la figura 1.3.

Tots els *hosts* de la subxarxa tindran els mateixos bits de xarxa i subxarxa (ja que tots pertanyen a la mateixa xarxa i subxarxa), l'únic que identificarà al *host* respecte dels altres de la seva subxarxa seran els bits de *host*. Les adreces dels *hosts* de la subxarxa 00 seran, doncs, les que es poden veure en la taula 1.7.

Figura 1.3. Diferents subxarxes



La quantitat d'adreces es manté

Taula 1.7. Càlcul d'adreces de la subxarxa 00

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
00000000	00	000000	0	201.63.7.0
00000001	00	000001	1	201.63.7.1
00000010	00	000010	2	201.63.7.2
00000011	00	000011	3	201.63.7.3
00000100	00	000100	4	201.63.7.4
...	...	...	...	...
00111100	00	111100	60	201.63.7.60
00111101	00	111101	61	201.63.7.61
00111110	00	111110	62	201.63.7.62
00111111	00	111111	63	201.63.7.63

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits de host

Com passava amb les xarxes, amb les subxarxes hi ha algunes adreces que estan reservades:

- L'adreça amb tots el bits de *host* a 0 (valor de l'últim byte 00000000, en decimal 0) correspon a l'adreça que identifica a la pròpia subxarxa.
- L'adreça amb tots el bits de *host* a 1 (valor de l'últim byte 00111111, en decimal 63) correspon a l'adreça que identifica l'adreça *broadcast* dins de la subxarxa.

Per tant, en cada subxarxa es perden dues adreces, una que identifica la mateixa subxarxa, i una altra que identifica l'adreça *broadcast* dins de la subxarxa.

Així, finalment la nostra primera subxarxa tindria aquestes característiques:

- Subxarxa: 201.63.7.0
- Màscara: 255.255.255.192
- Subxarxa i màscara (notació abreviada): 201.63.7.0/26
- Adreça *broadcast* de la subxarxa: 201.63.7.63
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.1 fins a la 201.63.7.62

El càlcul de les adreces de les altres subxarxes és igual, però en cadascuna els valors dels bits de subxarxa coincidiran amb la subxarxa.

El càlcul de les adreces de la **subxarxa 01**:

```
IP decimal: 201.63.7.64       201   .    63    .     7    .    64
IP binari:                 11001001 . 00111111 . 00000111 . 01 000000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . ss hhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        192
Màscara binària:            11111111 . 11111111 . 11111111 . 11000000
```

En la taula 1.8 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa.

TAULA 1.8. Càlcul d'adreces de la subxarxa 01

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
01000000	01	000000	64	201.63.7.64
01000001	01	000001	65	201.63.7.65
01000010	01	000010	66	201.63.7.66
01000011	01	000011	67	201.63.7.67
01000100	01	000100	68	201.63.7.68
...	...	...	...	...
01111100	01	111100	124	201.63.7.124
01111101	01	111101	125	201.63.7.125
01111110	01	111110	126	201.63.7.126
01111111	01	111111	127	201.63.7.127

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host.

Les característiques de la subxarxa 01 serien:

- Subxarxa: 201.63.7.64
- Màscara: 255.255.255.192
- Subxarxa i màscara (notació abreviada): 201.63.7.64/26

- Adreça *broadcast* de la subxarxa: 201.63.7.127
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.65 fins a la 201.63.7.126

El càlcul de les adreces de la **subxarxa 10**:

```
IP decimal: 201.63.7.128      201   .    63    .     7    .    128
IP binari:                 11001001 . 00111111 . 00000111 . 10 000000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . ss hhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        192
Màscara binària:            11111111 . 11111111 . 11111111 . 11000000
```

En la taula 1.9 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa 10.

TAULA 1.9. Càlcul d'adreces de la subxarxa 10

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
10000000	10	000000	128	201.63.7.128
10000001	10	000001	129	201.63.7.129
10000010	10	000010	130	201.63.7.130
10000011	10	000011	131	201.63.7.131
10000100	10	000100	132	201.63.7.132
...	...	...	...	...
10111100	10	111100	188	201.63.7.188
10111101	10	111101	189	201.63.7.189
10111110	10	111110	190	201.63.7.190
10111111	10	111111	191	201.63.7.191

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host.

Les característiques de la subxarxa 10 serien:

- Subxarxa: 201.63.7.128
- Màscara: 255.255.255.192
- Subxarxa i màscara(notació abreviada): 201.63.7.128/26
- Adreça *broadcast* de la subxarxa: 201.63.7.191
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.129 fins a la 201.63.7.190

Finalment, el càlcul de les adreces de la **subxarxa 11**:

```
IP decimal: 201.63.7.192      201   .    63    .     7    .    192
IP binari:                  11001001 . 00111111 . 00000111 . 11 000000
Significat del bit:         xxxxxxxx . xxxxxxxx . xxxxxxxx . ss hhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:              255        255        255        192
Màscara binària:             11111111 . 11111111 . 11111111 . 11000000
```

En la taula 1.10 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa 11.

TAULA 1.10. Càlcul d'adreces de la subxarxa 11

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
11000000	11	000000	192	201.63.7.192
11000001	11	000001	193	201.63.7.193
11000010	11	000010	194	201.63.7.194
11000011	11	000011	195	201.63.7.195
11000100	11	000100	196	201.63.7.196
...	...	...	...	...
11111100	11	111100	252	201.63.7.252
11111101	11	111101	253	201.63.7.253
11111110	11	111110	254	201.63.7.254
11111111	11	111111	255	201.63.7.255

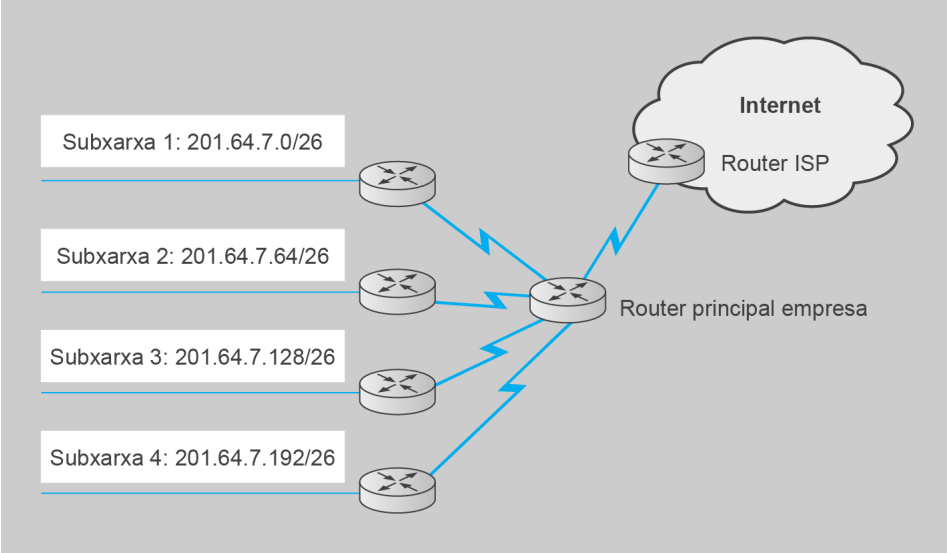
Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host.

Les característiques de la subxarxa 11 serien:

- Subxarxa: 201.63.7.192
- Màscara: 255.255.255.192
- Subxarxa i màscara(notació abreviada): 201.63.7.192/26
- Adreça *broadcast* de la subxarxa: 201.63.7.255
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.193 fins a la 201.63.7.254

La divisió de subxarxes seria la que es veu en la figura 1.4.

FIGURA 1.4. Configuració amb subxarxes



Cada LAN té assignada una subxarxa

També hem solucionat el problema que abans l'encaminador de l'ISP (i els del *backbone*) havia d'afegir quatre rutes per donar encaminament cap a les quatre LAN. Ara les quatre subxarxes es poden encaminar amb una única entrada que correspon a la xarxa original (la 201.63.7.0/24). Una ruta dirigida a 201.63.7.0/24 és una ruta que s'adreça a tots els *hosts* d'aquesta xarxa. Tots els possibles valors de *host* (l'últim byte) van des de 201.63.7.0 fins a 201.63.7.255 que corresponen a tots els *hosts* de totes les subxarxes de l'empresa.

El **treball amb subxarxes** sempre implica treballar en binari. El mode de treball habitual és:

1. Passar la xarxa i la màscara a binari
2. Treballar amb la xarxa i la màscara en binari
3. Passar els resultats a decimal

1.2 Subxarxes de mida variable

La divisió en subxarxes no ha de ser necessàriament de forma homogènia per a totes les subxarxes, com hem fet fins ara. Es poden fer divisions de diferents mides de la xarxa.

Seguint amb l'exemple proposat, començant per la xarxa 201.63.7.0/24 imagineu que, en lloc de tenir quatre xarxes locals amb cinquanta ordinadors, l'empresa té les xarxes locals amb aquestes mides:

- Dues xarxes locals amb 20 ordinadors (A i B)

- Una xarxa local amb 50 ordinadors (C)
- Una xarxa local amb 100 ordinadors (D)

En aquest cas les divisions que heu vist fins ara no serveixen, ja que assignen la mateixa quantitat d'adreces a cada subxarxa. El primer pas serà calcular les subxarxes que necessitem i la seva mida. És fàcil veure que necessitem quatre subxarxes, però, quina mida han de tenir?

No es poden crear subxarxes de qualsevol mida. Per exemple, no podeu fer subxarxes de mida 50 o 100 com les que es necessiten en l'exemple.

Atès que per fer subxarxes heu d'agafar bits de la part de *host*, i que la mida de les subxarxes depèn de la quantitat de bits que tenen, únicament podeu escollir subxarxes amb mida potència de 2 (en una classe C: 2, 4, 8, 16, 32, 64, 128, 256), com es pot veure en la taula 1.11.

TAULA 1.11. Bits necessaris per a la part de host

Nombre de bits de *host*	**Quantitat de *hosts***
1	$2^1 = 2$ *hosts*
2	$2^2 = 4$ *hosts*
3	$2^3 = 8$ *hosts*
4	$2^4 = 16$ *hosts*
5	$2^5 = 32$ *hosts*
6	$2^6 = 64$ *hosts*
7	$2^7 = 128$ *hosts*
8	$2^8 = 256$ *hosts*

Com més bits de host, més hosts es poden identificar

Per tant, heu d'escollir una mida de subxarxa suficientment gran per allotjar la xarxa local corresponent. En l'exemple:

- Les xarxes locals de 20 ordinadors (A i B) estaran dins de subxarxes de 32 adreces cadascuna.
- La xarxa local de 50 ordinadors (C) estarà dins d'una subxarxa de 64 adreces.
- La xarxa local de 100 ordinadors (D) estarà dins d'una subxarxa de 128 adreces.

Una vegada sabeu la mida de les xarxes, podeu saber quants bits de *host* necessitarà cada subxarxa, ho podeu consultar en la taula 1.11:

- Les subxarxes de 32 adreces (A i B) necessiten 5 bits de *host*.
- La subxarxa de 64 adreces (C) necessita 6 bits de *host*.
- La subxarxa de 128 adreces (D) necessita 7 bits de *host*.

La manera més senzilla de calcular les subxarxes sempre és començar per la subxarxa més gran, i anar calculant les subxarxes de les més grans a les més petites. Així, començarem per la subxarxa D, de 128 adreces.

```
IP decimal: 201.63.7.0        201   .    63    .     7    .     0
IP binari:                 11001001 . 00111111 . 00000111 . 0 0000000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . s hhhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        128
Màscara binària:            11111111 . 11111111 . 11111111 . 10000000
```

En la taula 1.12 podeu veure el càlcul de les adreces dels *hosts* dins de la **subxarxa D**.

Taula 1.12. Càlcul d'adreces de la subxarxa D

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
00000000	0	0000000	0	201.63.7.0
00000001	0	0000001	1	201.63.7.1
00000010	0	0000010	2	201.63.7.2
00000011	0	0000011	3	201.63.7.3
00000100	0	0000100	4	201.63.7.4
...	...	...	...	...
01111100	0	1111100	124	201.63.7.124
01111101	0	1111101	125	201.63.7.125
01111110	0	1111110	126	201.63.7.126
01111111	0	1111111	127	201.63.7.127

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits de host.

Les característiques de la subxarxa D serien:

- Subxarxa: 201.63.7.0
- Màscara: 255.255.255.128
- Subxarxa i màscara (notació abreviada): 201.63.7.0/25
- Adreça *broadcast* de la subxarxa: 201.63.7.127
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.1 fins a la 201.63.7.126

La **subxarxa C** té una mida diferent, per tant, la màscara que farem servir serà diferent. Amb 64 *hosts*, necessitarà 6 bits de *host*, i dos de subxarxa. Ara és molt important que triem valors dels bits de subxarxa que no s'hagin escollit per a cap de les subxarxes anteriors (en aquest cas, per a la subxarxa D). Si, per exemple, escolliu per als bits de subxarxa el valor 00, el rang d'adreces assignat a la subxarxa C seria des del següent:

```
primera IP (binari):       11001001 . 00111111 . 00000111 . 00 000000
primera IP (decimal):         201   .    63    .     7    .     0


última IP (binari):        11001001 . 00111111 . 00000111 . 00 111111
última IP (decimal):          201   .    63    .     7    .     63
```

Aquest rang d'adreces (de la 0 a la 63) col·lisiona amb el rang d'adreces assignat anteriorment a la subxarxa D (de la 0 a la 127). L'adreça 201.63.7.15, per exemple, estaria assignada a dues subxarxes alhora, i això és incorrecte.

> Els rangs d'adreces IP de les subxarxes no es poden superposar.

Per tant, heu d'escollir un valor dels bits de subxarxa per a la subxarxa C que no se superposi amb les adreces de la subxarxa D.

El valor dels bits de subxarxa 01 dóna com a resultat el rang 201.63.7.64 fins a 201.63.7.127, que també se superposa amb la subxarxa D. Això passa perquè el primer bit de l'últim byte de la subxarxa D és un 0 (és el valor del bit de subxarxa que hem escollit anteriorment per a la subxarxa D). La solució més fàcil és escollir un 1 per al primer bit de subxarxa de C per evitar que se superposi. Els valors 10 i 11 serien vàlids.

Escollim 10 i calculem les adreces de la subxarxa C:

```
IP decimal: 201.63.7.0        201   .    63    .     7    .    128
IP binari:                 11001001 . 00111111 . 00000111 . 10 000000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . ss hhhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        192
Màscara binària:            11111111 . 11111111 . 11111111 . 11000000
```

En la taula 1.13 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa C.

Les característiques de la subxarxa C serien:

- Subxarxa: 201.63.7.128
- Màscara: 255.255.255.192
- Subxarxa i màscara(notació abreviada): 201.63.7.128/26
- Adreça *broadcast* de la subxarxa: 201.63.7.191
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.129 fins a la 201.63.7.190

Taula 1.13. Càlcul d'adreces de la subxarxa C

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
10000000	10	000000	128	201.63.7.128
10000001	10	000001	129	201.63.7.129
10000010	10	000010	130	201.63.7.130
10000011	10	000011	131	201.63.7.131
10000100	10	000100	132	201.63.7.132
...	...	...	...	...
10111100	10	111100	188	201.63.7.188
10111101	10	111101	189	201.63.7.189
10111110	10	111110	190	201.63.7.190
10111111	10	111111	191	201.63.7.191

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host

Finalment, el càlcul de les subxarxes A i B. De nou heu de fer atenció a no superposar-les amb les subxarxes anteriors. Les subxarxes A i B tenen 3 bits de subxarxa, i veiem que les subxarxes anteriors:

- La subxarxa D comença per 0.
- La subxarxa C comença per 10.

Per no superposar les subxarxes A i B començaran per 11, per tant, escollim 110 per a la subxarxa A, i 111 per a la subxarxa B.

Càlculs de la **subxarxa A**:

```
IP decimal: 201.63.7.0        201   .    63    .     7    .    192
IP binari:                 11001001 . 00111111 . 00000111 . 110 00000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . sss hhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        224
Màscara binària:            11111111 . 11111111 . 11111111 . 1110000
```

En la taula 1.14 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa A.

Taula 1.14. Càlcul d'adreces de la subxarxa A

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
11000000	110	00000	192	201.63.7.192
11000001	110	00001	193	201.63.7.193
11000010	110	00010	194	201.63.7.194
11000011	110	00011	195	201.63.7.195
11000100	110	00100	196	201.63.7.196
...	...	...	...	...

TAULA 1.14 (continuació)

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
11011100	110	11100	220	201.63.7.220
11011101	110	11101	221	201.63.7.221
11011110	110	11110	222	201.63.7.222
11011111	110	11111	223	201.63.7.223

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host

Les característiques de la subxarxa A serien:

- Subxarxa: 201.63.7.192
- Màscara: 255.255.255.224
- Subxarxa i màscara(notació abreviada): 201.63.7.192/27
- Adreça *broadcast* de la subxarxa: 201.63.7.223
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.193 fins a la 201.63.7.222

I, finalment, els càlculs de la **subxarxa B**:

```
IP decimal: 201.63.7.0        201   .    63    .     7    .    224
IP binari:                 11001001 . 00111111 . 00000111 . 111 00000
Significat del bit:        xxxxxxxx . xxxxxxxx . xxxxxxxx . sss hhhhh

x = bit de xarxa
s = bit de subxarxa
h = bit de host

Màscara decimal:             255        255        255        224
Màscara binària:            11111111 . 11111111 . 11111111 . 1110000
```

En la taula 1.15 podeu veure el càlcul de les adreces dels *hosts* dins de la subxarxa A.

TAULA 1.15. Càlcul d'adreces de la subxarxa B

Bits de l'últim byte	Part de subxarxa	Part de *host*	Valor decimal	Adreça IP completa
11100000	111	00000	224	201.63.7.224
11100001	111	00001	225	201.63.7.225
11100010	111	00010	226	201.63.7.226
11100011	111	00011	227	201.63.7.227
11100100	111	00100	228	201.63.7.228
...	...	...	...	...
11111100	111	11100	252	201.63.7.252
11111101	111	11101	253	201.63.7.253
11111110	111	11110	254	201.63.7.254
11111111	111	11111	255	201.63.7.255

Els bits de subxarxa es mantenen per a tots els hosts de la mateixa subxarxa. Únicament es modifica el valor dels bits del host

Les característiques de la subxarxa B serien:

- Subxarxa: 201.63.7.224
- Màscara: 255.255.255.224
- Subxarxa i màscara(notació abreviada): 201.63.7.224/27
- Adreça *broadcast* de la subxarxa: 201.63.7.255
- Rang vàlid d'adreces per assignar a *hosts*: des de la 201.63.0.225 fins a la 201.63.7.254

Les xarxes de mida variable també es coneixen com a **VLSM** (*variable length subnet masking*, màscara de subxarxa de longitud variable) i van ser una de les mesures presses pel CIDR.

1.3 CIDR

Internet ha tingut un creixement exponencial. En la taula 1.16 podeu veure la quantitat de *hosts* registrats a Internet en diferents anys:

Llei de Moore

La llei de Moore (per Gordon E. Moore, cofundador d'Intel) descriu una tendència a llarg termini en la història del maquinari de computació. La capacitat d'integració de transistors en un circuit integrat es duplica aproximadament cada divuit mesos. Això passa des de l'any 1958 (any d'invenció del circuit integrat) fins a l'actualitat, i s'espera que continuï fins a l'any 2015 o 2020.

TAULA 1.16. Nombre de hosts a Internet

Any	Nombre de *hosts*
1981	213
1984	1024
1987	28.174
1992	1,136.000
1997	19,540.000
2002	162,128.493
2007	489,774.269
Gener 2011	818,374.269

Font: Internet System Consortium, https://www.isc.org/solutions/survey/history

El creixement accelerat d'Internet als anys vuitanta i noranta va crear diversos problemes. El més important va ser l'esgotament de l'espai d'adreces IP. El segon, l'augment de la mida de les taules d'encaminament als encaminadors. Al principi dels anys noranta, el creixement d'Internet era tal que el nombre de xarxes connectades es duplicava cada nou mesos, mentre que la tecnologia únicament podia duplicar la capacitat i la potència dels encaminadors cada divuit mesos. Segons els càlculs fets per l'IETF (*Internet engineering task force*, grup de treball sobre enginyeria a Internet) el 1993, d'haver continuat creixent Internet al mateix ritme que ho havia fet fins aquell moment, s'hauria col·lapsat l'any 1998.

Backbone

La **xarxa troncal** (en anglès *backbone* vol dir espina dorsal, columna vertebral) d'Internet es refereix a les rutes principals de dades entre xarxes grans, estratègicament interconnectades, i els encaminadors principals d'Internet. Aquestes rutes de dades estan suportades per xarxes d'alta capacitat comercials, governamentals i acadèmiques i intercanvien trànsit entre països i continents per mitjà de connexions per fibra òptica (algunes transoceàniques).

Aquest creixement el va provocar principalment la disparitat en la mida de les classes de xarxes A, B i C. Moltes organitzacions havien d'escollir entre sol·licitar

una xarxa de classe C (amb 256 adreces) o una de classe B (amb 65.536). Si amb una classe C no en tenien prou(per exemple, 256 adreces és totalment insuficient per al campus d'una universitat), en molts casos optaven per demanar una xarxa de classe B, encara que moltes de les adreces quedaven sense utilitzar. Per aquest motiu, les xarxes de classe B es van esgotar ràpidament, i es va acordar crear grups de xarxes de classe C. D'aquesta manera, les organitzacions podien optar a xarxes de mides a mig camí entre les classes B i C. Una organització amb 4.096 adreces podia demanar un grup de setze xarxes de classe C. Amb aquesta solució es reduïa el problema del ràpid esgotament d'adreces IP, però es va generar un altre problema: el creixement de les taules de rutes. L'organització de l'exemple hauria d'afegir setze entrades a les taules de rutes dels encaminadors per ser visible (ja que caldria una entrada per a cada xarxa assignada). Això hauria provocat un creixement exponencial de les taules de rutes dels encaminadors que formen part del *backbone* d'Internet, els quals treballen al límit de la tecnologia.

Els dos problemes descrits (esgotament d'espai d'adreces IP i creixement de les taules de rutes) es van resoldre conjuntament a l'any 1993 amb l'adopció d'un sistema anomenat **CIDR** (*classless interdomain routing*, o encaminament entre dominis sense classe), que definia una nova forma d'assignació de blocs d'adreces IP i nous mètodes d'encaminament de paquets. Es va publicar una nova especificació de CIDR l'any 2006.

Es va presentar el concepte de **superxarxa** oposat al concepte de subxarxa. Si una subxarxa consistia en la divisió d'una xarxa de classe A, B o C en diferents parts, una superxarxa consisteix en una xarxa formada per la combinació de dues o més xarxes amb una màscara comuna.

Es diu que amb la divisió d'adreces IP la frontera entre la part de xarxa i la de *host* era per bytes (la part de xarxa tenia 1, 2 o 3 bytes o, el que és el mateix 8, 16 o 24 bits). Amb el CIDR aquesta frontera és per bit, ja que la màscara pot agafar qualsevol valor.

Així, per exemple, es pot formar una superxarxa de 1.024 adreces amb quatre xarxes de classe C. La subxarxa resultant tindrà una màscara de 22 bits (per tant, 10 bits de *host*, suficients per allotjar $2^{10} = 1.024$ *hosts*). Aquesta superxarxa únicament ha d'aparèixer una vegada en les taules de rutes dels encaminadors.

Amb el CIDR la part de l'adreça IP que correspon a la xarxa i la part que correspon a *hosts* està especificada pel valor de la màscara exclusivament, i no té significat la classificació tradicional en classes A, B i C d'acord amb el valor dels primers bits.

Únicament es respecta el significat dels primers bits en les classes D (*multicast*) i E (reservat).

El procés de crear una superxarxa s'anomena **supernetting** o **agregació de rutes**. Els beneficis de la creació de superxarxes són la conservació de l'espai d'adreces i la millora de l'eficiència en els encaminadors en termes d'espai d'emmagatzematge i càlcul de rutes.

2. Encaminament dinàmic

La tasca principal de la capa de xarxa és trobar el millor camí entre l'origen i la destinació de la comunicació. Aquesta tasca es du a terme pels dispositius de capa de xarxa, els encaminadors. Els encaminadors analitzen l'adreça IP de destinació al paquet IP i, fent ús de les taules d'encaminament que tenen en memòria, decideixen per quina de les seves interfícies retransmetran el paquet. La informació de la taula d'encaminament la pot calcular manualment l'administrador de la xarxa i introduir la configuració als encaminadors de manera estàtica. Aquestes configuracions són vàlides però no es poden adaptar als canvis en la xarxa. Per exemple, si un encaminador de la xarxa es desconnecta, la configuració de rutes de la resta d'encaminadors es mantindrà invariable, encara que les seves rutes enviïn paquets per l'encaminador desconnectat. A més a més, en xarxes molt grans (com Internet), el cost de calcular les rutes fa inviable que l'administrador ho faci manualment.

Per aquests dos motius es van crear els **protocols d'encaminament dinàmic**, en què els encaminadors intercanvien informació sobre la topologia de la xarxa i ells mateixos poden configurar les rutes que tindran configurades. Aquesta operació es du a terme periòdicament de manera que, si hi ha cap canvi en la xarxa (per exemple, s'apaga un encaminador), la propera vegada que s'executi l'algorisme recalcularà les rutes amb la nova informació de la topologia de la xarxa.

2.1 Algorismes d'encaminament

La funció principal dels encaminadors és calcular el camí entre l'origen de la comunicació i la destinació en la xarxa. En realitat, els encaminadors no solen conèixer la topologia sencera de la xarxa i, per tant, no calculen el camí sencer fins a la destinació. Penseu en la grandària de la xarxa d'Internet i penseu que el cost de tenir tota la informació de xarxes i camins seria excessiva per a la majoria dels encaminadors. En realitat, el que calculen els encaminadors és simplement el salt següent del paquet IP, és a dir, la interfície del mateix encaminador per on han d'enviar els paquets rebuts perquè arribin a la destinació.

En les xarxes orientades a connexió, aquest procés té lloc en el moment d'establir el circuit virtual. A partir d'aquest moment, els paquets es commuten entre els diferents circuits virtuals, per això se'ls coneix sovint com a *commutadors de capa 3*. La commutació de paquets és una tasca senzilla i ràpida, ja que no cal fer càlculs complicats.

En les xarxes no orientades a connexió, la decisió de per on s'envien els paquets es pren per a cada paquet i per a cada encaminador.

Atenent a quina informació fan servir els encaminadors per prendre la decisió de per on s'han d'enviar els paquets, els algorismes d'encaminament es divideixen en:

- algorismes d'encaminament estàtics,
- algorismes d'encaminament dinàmics.

2.1.1 Algorismes d'encaminament estàtics

Els **algorismes d'encaminament estàtics** calculen les rutes fent servir informació de la xarxa recopilada amb anterioritat. Per exemple, es poden fer servir les dades de velocitat de la interfície, la mitjana de trànsit o l'històric d'errades en la transmissió per l'enllaç. Els càlculs de rutes no es fan dinàmicament sobre els encaminadors, sinó que es fan una vegada i es carreguen en la configuració dels encaminadors. Per això, aquest càlculs poden ser tan complicats com es vulgui o requerir una gran quantitat de recursos o memòria. Una vegada acabats els càlculs, el resultat de les rutes es carrega de manera estàtica en les taules d'encaminament dels encaminadors. No cal cap tipus de protocol d'encaminament, ja que els encaminadors no intercanvien informació per calcular les rutes. La part negativa és que aquests algorismes no responen a canvis en les condicions de la xarxa i, per tant, no es poden adaptar a possibles problemes: encara que falli una connexió o un dels enllaços estigui patint un trànsit excessiu, l'encaminador no canviarà les decisions d'encaminament.

2.1.2 Algorismes d'encaminament dinàmic

Els **algorismes d'encaminament dinàmic** fan servir informació sobre l'estat de la xarxa per calcular la millor ruta cap a la destinació. Amb aquest tipus d'algorismes d'encaminament, els encaminadors intercanvien paquets d'informació sobre l'estat de la xarxa contínuament i, per tant, cal definir un protocol d'encaminament. Amb la informació actualitzada de l'estat de la xarxa, l'encaminador calcula la ruta òptima cap a la destinació. Per tant, es diu que aquest algorismes són **autoadaptatius**, responen als canvis de la xarxa per obtenir sempre el millor camí. Per exemple, si un encaminador troba que una de les interfícies està desconnectada o funciona molt lentament, aquesta informació s'actualitzarà en l'algorisme d'encaminament, el qual és probable que esculli una altra interfície com el camí a seguir pels paquets cap a la seva destinació. Aquests algorismes d'encaminament dinàmics s'executen en els encaminadors en temps real, per tant, no han de ser gaire complexos, atesa la limitació de recursos de CPU i memòria dels encaminadors.

Els algorismes d'encaminament **estàtics** fan servir informació invariable per determinar la ruta. No s'adapten als canvis en l'estat de la xarxa.
Els algorismes d'encaminament **dinàmics** fan servir informació que recopilen en temps real per determinar la ruta. Poden modificar la ruta dinàmicament i necessiten un protocol d'encaminament perquè els encaminadors intercanviïn informació.

Mètrica

Els algorismes d'encaminament intenten trobar el camí més curt entre dos *host* en una xarxa però, què entenem per *camí més curt*? En les xarxes de computadors la distància física és un factor poc important per determinar el cost d'arribar fins a la destinació.

Per exemple, en els viatges per carretera es pot calcular el cost d'arribar a la destinació en funció d'alguns d'aquests criteris o una combinació d'ells:

- Distància física
- Temps de trajecte
- Trànsit de la carretera
- Cost de peatge
- Tipus de carretera (autopista, nacional, regional, etc.)

En les xarxes de computadors els criteris són diferents:

- Velocitat de la interfície
- Quantitat d'errors de la connexió
- Retard
- Trànsit mitjà de l'enllaç

Aquest diferents criteris es poden combinar amb una mètrica o fórmula que determinarà la manera de calcular la distància per al càlcul de rutes.

La **mètrica** determina com es calcula el cost de travessar un enllaç i serveix per calcular el camí òptim.

La mètrica més senzilla i que de vegades es fa servir és el nombre de salts, en què cada enllaç té un cost d'1.

2.2 Encaminament per inundació

L'encaminament per inundació (*flooding* en anglès) és la tècnica d'encaminament més senzilla. Consisteix a retransmetre el paquet per totes les interfícies excepte per aquella per la qual s'ha rebut.

El seu problema principal és que multiplica el paquet per totes les interfícies, la qual cosa genera una gran quantitat de paquets duplicats, per tant, és molt ineficient i fa servir molta amplada de banda de la xarxa. A més, en cas que en la topologia hi hagi bucles, els paquets es poden quedar donant voltes indefinidament en la xarxa.

S'han de prendre mesures per limitar el procés de duplicació de paquets. Aquestes són dues tècniques:

- Una opció seria introduir en cada paquet un comptador de salts. Aquest comptador decreix cada vegada que el paquet travessa un encaminador. Quan el comptador arriba a zero, el paquet s'elimina de la xarxa. Aquesta és la tècnica que es fa servir en TCP/IP. La capçalera IP té el camp TTL (*time to live* o temps de vida), que indica el nombre màxim de salts que fa el paquet.

- I una altra opció pot ser que els encaminadors tinguin una llista de paquets enviats. D'aquesta manera l'encaminador no propagarà dues vegades el mateix paquet i s'evita el problema dels paquets que es queden donant voltes als bucles.

L'**encaminament per inundació** consisteix a enviar el paquet per totes les interfícies excepte per la que l'ha rebut. S'han de prendre mesures per evitar la multiplicació indiscriminada de paquets.

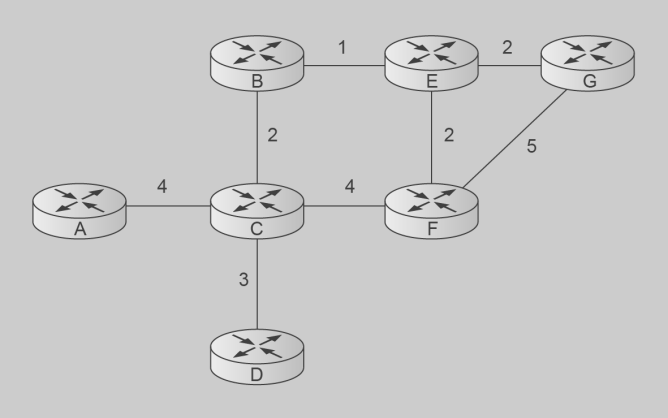
La inundació no és pràctica en la majoria dels casos, però té alguns usos. Per exemple, aquest és el mètode que fan servir els ponts i els commutadors quan reben una trama que no tenen en la taula d'adreces MAC. Els encaminadors també s'envien la informació d'encaminament per inundació, al cap i a la fi aquesta informació l'envien abans de tenir les rutes, per tant, no poden saber els camins per arribar a la resta d'encaminadors.

Fins i tot es pot fer servir la inundació per comparació amb altres algorismes d'encaminament. De fet, la inundació sempre troba el camí més curt cap a la destinació (perquè els prova tots) encara que genera molt trànsit. Per això es pot comprovar si un algorisme ha trobat el millor camí comparant el resultat amb el primer paquet que ha arribat per inundació.

2.3 Algorisme de vector distància

En l'encaminament per vector distància cada encaminador té una taula interna o vector que indica la distància mínima per arribar a cada possible destinació de la xarxa i per quina interfície de l'encaminador s'han de retransmetre els paquets per arribar a la seva destinació. L'encaminador retransmet aquesta taula als encaminadors veïns. Tots els encaminadors actualitzen el contingut de la seva taula amb la informació que han rebut dels veïns. Per exemple, mireu la figura 2.1.

FIGURA 2.1. Encaminament per vector distància



Al començament de l'algorisme, l'encaminador A únicament sap el cost d'arribar a C amb cost 4. Quan rebi el vector distància dels seus veïns (en aquest cas, C) rebrà la informació del cost d'arribar als veïns de C. El vector distància de C serà el que es veu en la taula 2.1.

TAULA 2.1. Vector distància de C

Destinació	A	B	C	D	E	F	G
Cost	4	2	0	3	-	4	-

L'encaminador A pot calcular fàcilment el cost d'arribar als veïns de C. Per exemple, si a A li costa 4 arribar a C, i l'encaminador C anuncia que té B a una distància de 2, l'encaminador A pot deduir que pot arribar a B amb una distància de 4 + 2 = 6.

Durant la primera execució de l'algorisme no tots els encaminadors tenen la informació sobre com s'arriba a tots els encaminadors de la xarxa. Per exemple, com que C no sap com arribar a G, A no ho sabrà des del començament. Però passades unes quantes iteracions el resultat convergeix. Per exemple, C haurà rebut l'actualització del vector distància de B, que haurà actualitzat el seu vector amb la informació del vector distància d'E, el qual sí que inclou com s'arriba a G; finalment aquesta informació també arribarà a A i D mitjançant l'actualització del vector distància de C.

La mètrica pot ser el nombre de salts, el retard, la taxa d'errors, etc. o una combinació d'aquests. Cada encaminador únicament sap directament el cost

d'arribar als seus encaminadors veïns, la resta de costos l'obté per càlcul amb la informació del vector distància dels veïns.

Compte a infinit

A vegades els llibres descriuen el problema del compte a infinit com "l'algorisme de vector distància les notícies bones es propaguen ràpidament i les dolentes lentament".

L'algorisme de vector distància té el problema del **compte a infinit**, per al qual no s'ha trobat una solució definitiva. Aquest consisteix en el fet que quan es troba un camí millor per anar a un destinació, la xarxa convergeix ràpidament en la nova ruta. Però quan es passa d'un camí millor a un altre de més llarg (per exemple, perquè un dels encaminadors del camí original ha deixat de funcionar), a la xarxa li costa molt tornar a calcular la ruta, i els encaminadors incrementen els seus comptadors fins que arriben al valor màxim.

L'algorisme de vector distància es va fer servir en l'ARPANET original. També es va utilitzar en diversos protocols de xarxes (DECNET de Digital, IPX de Netware i AppleTalk d'Apple). Actualment es fa servir en el RIP (*routing information protocol*, protocol d'informació d'encaminament), que fins l'any 1988 era l'únic protocol d'encaminament d'Internet. També es fa servir en els protocols IGRP (*interior gateway routing protocol*, protocol d'encaminament interior a l'encaminador) i EIGRP (*enhanced IGRP*, IGRP millorat) de Cisco.

ARPANET

ARPANET (*advanced research projects agency network*, xarxa de l'agència de projectes d'investigació avançada) va ser creada pel Departament de Defensa dels Estats Units i va ser la primera xarxa en què es van provar els protocols TCP/IP. Finalment, quan altres xarxes es van anar afegint a ARPANET van formar Internet.

2.4 Estat de l'enllaç

Paquet d'eco

Els paquets d'eco (***echo request*** i ***echo reply***, petició d'eco i resposta d'eco respectivament) són paquets enviats per comprovar una connexió o el *temps d'anada i tornada* d'aquesta (el temps que tarda un missatge en travessar dues vegades la connexió). L'emissor envia una petició d'eco i mesura el temps que tarda en arribar la resposta del receptor. Així calcula el *temps d'anada i tornada* de la connexió.

Aquest algorisme també es coneix com l'**algorisme de Dijkstra** o **SPF** (*shortest path first*, primer el camí més curt). Aquest algorisme es va dissenyar per solucionar les limitacions dels algorismes basats en el vector distància. L'algorisme de Dijkstra es pot explicar en cinc passos. Cada encaminador ha de fer el següent:

1. Descobrir els seus veïns i conèixer-ne les adreces de xarxa. Això ho aconsegueix enviant uns paquets anomenats paquets *HELLO*.

2. Mesurar el retard o cost de cadascun dels veïns. La manera més senzilla de fer-ho és enviar un paquet d'eco, al qual respondrà l'encaminador veí.

3. Construir un paquet amb tota aquesta informació. Aquests paquets s'anomenen LSP (*link state packet*, paquet d'estat de l'enllaç) i contenen entre altres coses la identificació de l'encaminador, els seus veïns i el retard per arribar-hi.

4. Enviar aquest paquet a tota la resta d'encaminadors (es pot fer per inundació, o fer servir una versió més eficient de l'algorisme).

5. Calcular el camí més curt a la resta d'encaminadors. Amb la informació rebuda dels LSP de la resta d'encaminadors de la xarxa, l'encaminador va

creant un graf en el qual va afegint els encaminadors. Fent servir l'algorisme de Dijsktra el graf queda en forma d'arbre d'expansió (sense bucles) amb el camí més curt per arribar a cada encaminador.

El funcionament de l'algorisme de l'estat de l'enllaç és realment oposat al del vector distància, ja que el vector distància envia la informació del cost per arribar a tots els encaminadors de la xarxa als seus veïns, i l'algorisme d'estat de l'enllaç envia la informació de cost als seus veïns i a tots els encaminadors de la xarxa.

Mitjançant l'algorisme de l'estat de l'enllaç l'encaminador pot conèixer l'arbre d'expansió de la xarxa i, per tant, saber tota la ruta que faran servir els paquets fins arribar a la seva destinació.

L'algorisme de vector distància es fa servir en els protocols d'encaminament IS-IS (*intermediate system to intermediate system*, de sistema intermedi a sistema intermedi) i OSPF (*open shortest path first*, obre primer el camí més curt), que és el protocol d'encaminament estàndard d'Internet (encara que també se'n fan servir d'altres).

2.5 Protocols d'encaminament

Hi ha multitud d'algorismes d'encaminament, la majoria basats en l'algorisme de vector distància o estat de l'enllaç.

Es defineix un **sistema autònom** (AS o *autonomous system*) o una xarxa administrada i gestionada per una organització, es fa servir un protocol d'encaminament en concret. Així, Internet està formada per una sèrie d'AS connectats entre ells. Els algorismes d'encaminament també es poden classificar en:

- Algorismes d'encaminament interior o **IGP** (*interior gateway protocol*): que tracten l'encaminament intern a un AS.

- Algorismes d'encaminament exterior o **EGP** (*exterior gateway protocol*): que tracten l'encaminament entre diferents AS.

2.5.1 Protocols d'encaminament interior

A Internet es fan servir multitud d'algorismes d'encaminament interior. Els podem classificar entre els protocols que fan servir l'algorisme d'estat de l'enllaç, i els que fan servir l'algorisme de vector distància.

Protocols d'encaminament interior basat en el vector distància

Aquests protocols fan servir el protocol de vector distància i no tenen consciència completa de la topologia de la xarxa:

- **RIP** (*routing information protocol*, protocol d'informació d'encaminament). És un dels protocols d'encaminament més antics que existeixen. Originalment la mètrica que feia servir estava basada en el nombre de salts i no permetia fer servir múltiples rutes. L'any 1993 es va publicar la versió 2 del RIP, que afegia subxarxes i màscares de mida variable. El RIP és un protocol senzill que es pot fer servir en xarxes petites, però no és recomanable fer-lo servir en xarxes grans (de més de dotze encaminadors).
- **IGRP** (*interior gateway routing protocol*, protocol d'encaminament interior a l'encaminador). L'IGRP és un protocol propietari de Cisco i, per tant, únicament es pot fer servir en els seus encaminadors, és a dir, tots els encaminadors del sistema autònom han de ser de Cisco. Es va dissenyar per solucionar els problemes del RIP (comptador de salts màxim de quinze i mètrica d'encaminament única). L'IGRP pot fer servir diferents mètriques per a cada ruta: amplada de banda, retard, càrrega, fiabilitat i MTU (*maximum transfer unit*, mida màxima del paquet de transmissió).
- **EIGRP** (*enhanced interior gateway routing protocol*, protocol millorat d'encaminament interior a l'encaminador). Aquest protocol aporta optimitzacions importants a l'IGRP, per minimitzar la inestabilitat de les rutes quan canvia la topologia i també respecte a l'amplada de banda i el consum del processador dels encaminadors que fa servir el protocol.

Protocols d'encaminament interior basats en l'estat de l'enllaç

Aquests protocols tenen un coneixement complet de la topologia de la xarxa i fan servir aquesta informació per trobar la millor ruta per als paquets. Els dos protocols interiors basats en l'estat de l'enllaç més populars són:

- **OSPF** (*open shortest path first*, obre primer el camí més curt). És un protocol autoadaptatiu. La darrera versió (versió 3) està adaptada per treballar amb IPv6 i es va definir l'any 2008. L'OSPF és probablement l'IGP més utilitzat en grans xarxes corporatives. És un estàndard d'Internet i està recomanat per l'IAB (*Internet Architecture Board*, és el comitè encarregat de supervisar el desenvolupament tècnic d'Internet). Aquestes són les seves característiques més importants:
  - És autoadaptatiu, reacciona als canvis de la xarxa automàticament.
  - Suporta múltiples paràmetres per calcular la mètrica.
  - Suporta subxarxes, VLSM i CIDR.
  - Estructura el sistema autònom en àrees per simplificar l'administració i optimitzar el trànsit. Les àrees s'identifiquen per nombres de 32 bits

que es poden representar de manera semblant a les IP, separats per punts.
  - L'àrea 0 (0.0.0.0) s'anomena *backbone* i connecta la resta d'àrees entre elles.
  - Estableix mecanismes de validació als missatges d'encaminament.

- **IS-IS** (*intermediate system to intermediate system*, de sistema intermedi a sistema intermedi). Va ser dissenyat pel protocol DECNET i va ser adoptat finalment per l'OSI. Les seves característiques són molt semblants a l'OSPF. Però l'OSPF es va dissenyar per a xarxes TCP/IP, mentre que IS-IS es va dissenyar de manera neutral respecte al tipus d'adreces que faria servir, per això la seva adaptació d'IPv6 va ser molt senzilla. Una altra diferència és la manera en què l'IS-IS defineix les àrees i la manera en què aquestes en comuniquen entre elles.

Intermediate System

El terme *sistema intermedi* és la manera que té l'OSI d'anomenar els encaminadors. El protocol IS-IS estava pensat per a comunicacions entre encaminadors.

2.5.2 Protocols d'encaminament exterior

Els protocols d'encaminament exterior s'utilitzen per calcular les rutes entre els diferents sistemes autònoms. El primer protocol d'encaminament exterior que es va fer servir va ser l'**EGP** (*exterior gateway protocol*, protocol d'encaminament exterior), especificat l'any 1982. Però aquest protocol es va substituir pel **BGP** (*border gateway protocol*, protocol d'encaminament de frontera). El BGP no fa servir les mètriques tradicionals del protocols interiors, sinó que fa servir mètriques basades en el camí, polítiques de xarxa i regles. Recordeu que es fa servir entre diferents sistemes autònoms, administrats per diferents organitzacions. En aquests casos el camí més curt no sempre és el camí que es vol que porti la informació. Actualment, a tot Internet es fa servir la versió 4 del BGP, definida l'any 1994.

2.6 Configuració dels protocols d'encaminament

Els protocols d'encaminament es poden classificar en protocols de vector-distància i protocols d'estat de l'enllaç. És important tenir en compte les característiques principals d'aquests protocols, la seva classificació i com s'han de configurar en l'encaminador.

2.6.1 Protocols vector-distància

La mètrica per determinar quina és la millor ruta per enviar un paquet de dades és el nombre de salts que ha de fer un paquet des de l'origen fins a la destinació. Els protocols més comuns del tipus vector-distància són els següents:

- **RIP** (*routing information protocol*, protocol d'informació d'encaminament). Fa servir el nombre de salts (*hops* en anglès) com a mètrica d'encaminament.
- **IGRP** (*interior gateway routing protocol*, protocol d'encaminament interior de l'encaminador). Inventat per Cisco, es fa servir pels encaminadors per intercanviar dades d'encaminament dins d'un sistema autònom.
- **EIGRP** (*enhanced interior gateway routing protocol*, protocol millorat d'encaminament interior a l'encaminador). És propietari de Cisco. Es considera un protocol híbrid. És un protocol d'encaminament avançat basat en l'algorisme d'encaminament del vector distància, amb optimitzacions per minimitzar la inestabilitat de l'encaminament després de canvis en la topologia (problema de *compte a infinit*) i l'ús del processador i l'amplada de banda del protocol.

Les característiques principals d'aquests protocols es mostren en la taula 2.2.

TAULA 2.2. Característiques dels protocols de vector-distància

Protocol	Mètrica	Distància administrativa	Nombre de salts	Període d'actualització
RIP	Salts	120	15	30 segons
IGRP	Salts, amplada de banda, retard, càrrega i fiabilitat.	100	255	90 segons
EIGRP	Amplada de banda de la línia Cost Retard	90	-	Variable

Configuració de la versió 1 del protocol RIP

Protocol d'encaminament RIP

El protocol RIP utilitza el nombre de salts com a mètrica per generar les taules d'encaminament i permetre als paquets de dades arribar a la xarxa de destinació. Aquest protocol s'ha de configurar en tots els encaminadors que volem que intercanviïn informació.

El protocol RIP és un protocol d'encaminament de vector-distància que es destina a xarxes petites i mitjanes. La seva característica principal és que no és un protocol propietari, el poden utilitzar tots els fabricants d'encaminadors.

L'encaminador coneix les rutes que té connectades directament, el RIP s'ha de configurar globalment en l'encaminador i s'han d'afegir les xarxes que coneix en cada interfície. El RIP fa les funcions següents:

- Anuncia totes les xarxes que coneix l'encaminador en els encaminadors de la xarxa directament connectats a les seves interfícies.
- Escolta les actualitzacions externes.
- Difon les actualitzacions amb informació de l'estat de la xarxa en totes les interfícies.

L'ordre per configurar el protocol RIP s'executa des del mode de configuració global i és **router rip**, en executar-la accedim al mode de configuració del protocol.

Un cop en aquest mode s'han d'especificar les xarxes per a les quals treballarà (cada xarxa està assignada a una interfície) amb l'ordre **network *IP de la xarxa***. A continuació podeu veure un exemple d'ús d'aquestes ordres.

```
Router(config)#router rip
Router(config–router)#network 192.168.1.0
```

Fixeu-vos que en l'exemple anterior hem indicat a l'encaminador que el protocol RIP envia i rep actualitzacions per la interfície que té configurada una adreça IP de la xarxa 192.168.1.0.

Hi ha una sèrie d'ordres per visualitzar les estadístiques del RIP i els temps d'actualització d'aquestes estadístiques. Són les següents:

- **show ip protocols**. Mostra totes les variables que utilitza el protocol RIP i identifica cada variable de temps que utilitza. Les podeu veure en la taula 2.3.
- **show ip rip database**. Mostra la base de dades del protocol RIP amb tota la informació referent a quines rutes té directament connectades i quin encaminador proporciona les rutes addicionals.
- **debug ip rip**. Proporciona a l'administrador la informació necessària per determinar com està treballant el protocol. És un mode de depuració, i es pot desactivar executant **undebug ip rip**.

debug

L'ordre **debug** permet a l'administrador activar el mode de depuració per un protocol en concret. El mode de depuració mostra missatges que indiquen totes les accions que fa el protocol. Per desactivar el mode de depuració s'utilitza l'ordre **undebug**.

Taula 2.3. Variables del protocol RIP

Variable	Valor	Descripció
UPDATE EVERY	30	S'envien actualitzacions cada 30 segons.
INVALID AFTER	180	Es descarta el paquet després de 180 segons.
SEND	1	Versió del protocol RIP per enviar actualitzacions.
RECV	1,2	Pot rebre paquets de qualsevol versió RIP.
Routing for networks	172.17.0.0 192.168.0.0	Envia actualitzacions d'aquestes xarxes.
Routing inf sources	192.168.0.2	Encaminador que actualitza la nostra taula.

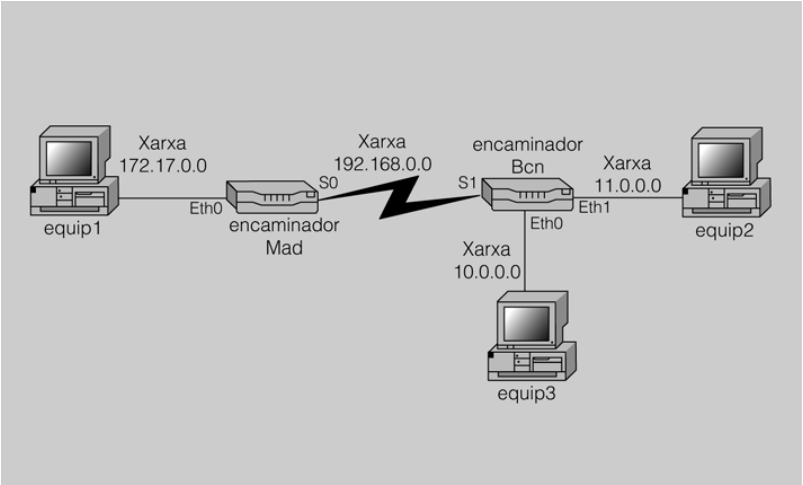
La taula 2.4 mostra la configuració de les interfícies dels encaminadors de Madrid i Barcelona.

Taula 2.4. Adreces dels encaminadors Mad i Bcn

Dispositiu	Interfície Eth0	Interficie Eth1	Serial
Encaminador Mad	172.17.1.1/16	No configurat	192.168.1.1/24
Encaminador Bcn	10.0.0.1/8	11.0.0.1/8	192.168.1.2/24

Amb el disseny de la figura 2.2 fem un exemple de configuració de RIP, en què ja s'han configurat totes les interfícies dels encaminadors amb les seves adreces i s'ha activat el protocol RIP en totes les interfícies excepte en la de l'encaminador Mad.

FIGURA 2.2. Disseny d'una xarxa RIP



Fem la configuració sencera de l'encaminador Mad amb la seqüència d'ordres següent:

1. Configurar adreces en totes les interfícies. Cal activar les interfícies dels encaminadors un cop s'han configurat. Utilitzarem les ordres **ip address** ***IP màscara*** per configurar les interfícies i **no shutdown** per activar-les.

2. Configurar RIP amb l'ordre **router rip** i **network** ***adreça de la xarxa***.

3. Mostrar informació de configuració amb l'ordre **show ip rip database**.

En la figura 2.3 podeu veure la taula d'encaminament de l'encaminador Mad mitjançant l'ordre **show ip route**. Veiem que només coneix les xarxes que té connectades directament.

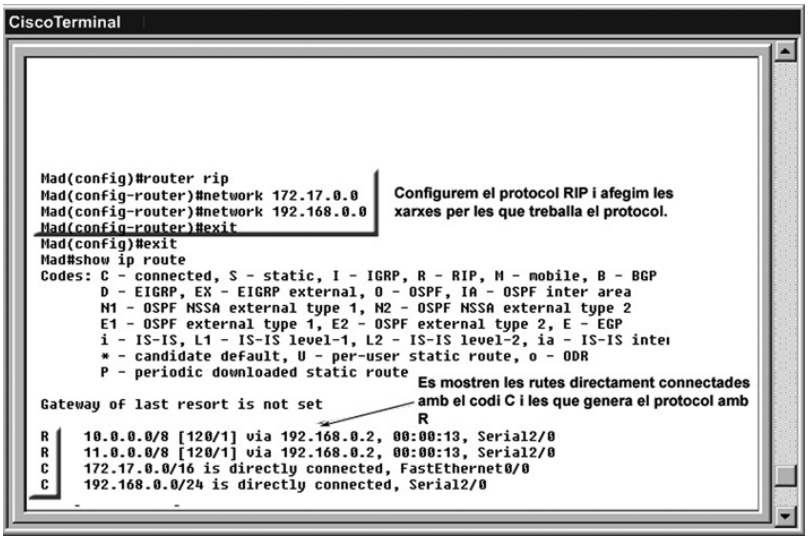
FIGURA 2.3. Configuració d'interfícies i taula d'encaminament

CiscoTerminal

```
Router>enable
Router#conf term
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#hostname Mad
Mad(config)#interface fastethernet 0/0
Mad(config-if)#ip address 172.17.1.1 255.255.0.0
Mad(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed st
Mad(config-if)#exit
Mad(config)#interface serial 2/0
Mad(config-if)#ip address 192.168.0.1 255.255.255.0
Mad(config-if)#clock rate 56000
Mad(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Mad(config-if)#exit
Mad(config)#exit
Mad#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    172.17.0.0/16 is directly connected, FastEthernet0/0
C    192.168.0.0/24 is directly connected, Serial2/0
```

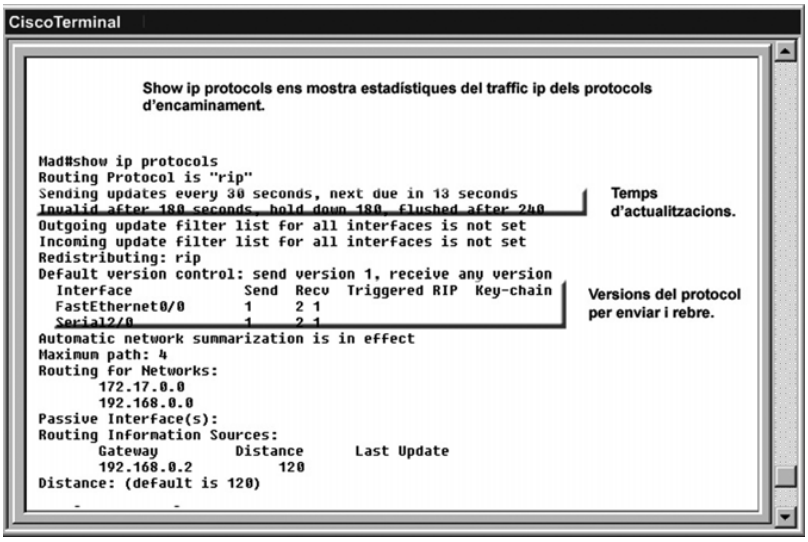
La figura 2.4 mostra com es fa la configuració de l'encaminador per utilitzar el protocol RIP, només cal tornar a observar la taula d'encaminament i veure que ja s'ha actualitzat amb les noves rutes que genera el RIP. El contingut de la taula d'encaminament ens indica amb quin protocol s'ha generat la ruta o si està directament connectada i quin és el salt següent.

Figura 2.4. Resultat d'executar show ip route



En la figura 2.5 ja s'ha configurat el protocol d'encaminament, i amb l'ordre **show IP protocols** podem observar totes les variables que utilitza el protocol RIP, identificar cada variable de temps i analitzar els valors.

Figura 2.5. Resultat d'executar show ip protocol



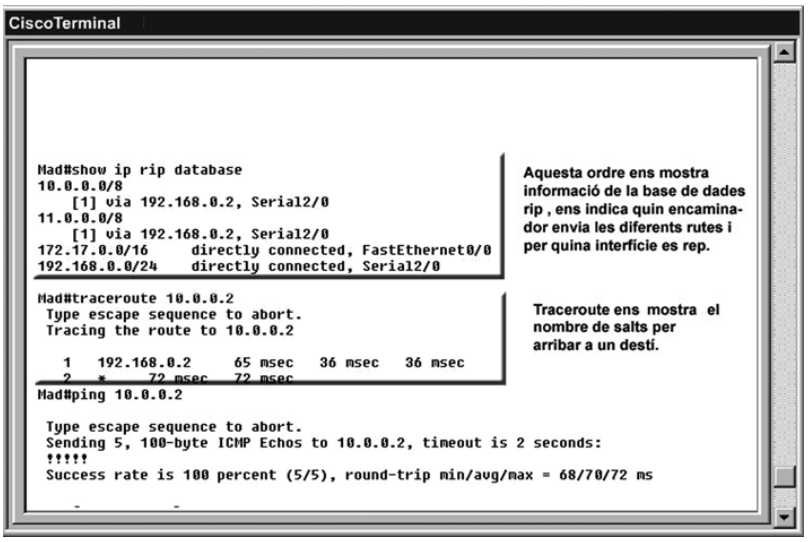
Podeu veure que les variables de temps de la imatge especifiquen els valors següents:

- *Invalid after 180 seconds*. Vol dir que es descarta el paquet després de 180 segons d'espera.

- *Interface send-recv*. Podem veure per a cada interfície la versió del protocol que es fa servir per rebre o enviar actualitzacions. Les dues interfícies d'aquest exemple envien actualitzacions amb la versió 1 del protocol i en reben de la versió 1 i/o 2 del protocol.

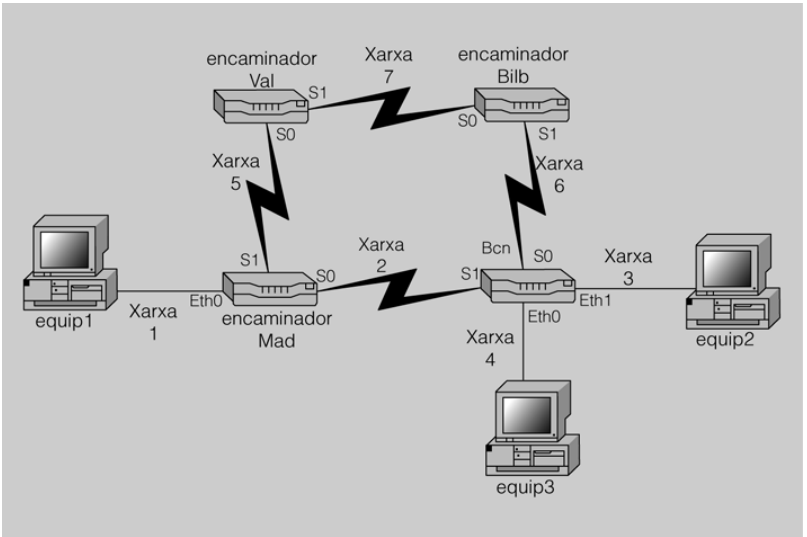
La figura 2.6 proporciona informació de la base de dades del protocol RIP amb totes les rutes connectades. Podeu comprovar el bon funcionament de la configuració efectuant un *ping* en diverses adreces IP de diferents xarxes. Cal executar l'ordre **traceroute** per determinar els salts que fa un paquet de dades fins a la destinació.

FIGURA 2.6. Resultat d'executar show ip rip database



Amb el disseny de xarxa de la figura 2.7 veureu un exemple de redundància de rutes. Un paquet de dades pot optar per utilitzar diferents rutes per arribar a una destinació, i els encaminadors han de seleccionar la millor ruta basant-se en la mètrica. Només s'incorpora en la taula d'encaminament una de totes les rutes possibles per arribar a la destinació.

FIGURA 2.7. Exemple de rutes redundants



Els encaminadors generen la taula d'encaminament publicant i actualitzant les taules periòdicament. Els encaminadors coneixen les xarxes directament connectades, com es mostra en la taula 2.5.

Taula 2.5. Taula de xarxes del disseny de rutes redundants

Encaminador	Xarxes conegudes	Xarxes desconegudes
Mad	1, 2, 5	3, 4, 6, 7
Bcn	2, 3, 4, 6	1, 5, 7
Val	5, 7	1, 2, 3, 4, 6
Bilb	6, 7	1, 2, 3, 4, 5

Les taules d'encaminament descarten múltiples rutes per a una mateixa destinació generades amb un protocol. Si observeu la taula 2.6, podeu deduir com l'encaminador Mad ha seleccionat les rutes en les xarxes basant-se en la mètrica.

Taula 2.6. Taula de rutes de l'encaminador Mad

Destinació	Possibles rutes de destinació	Mètrica	Ruta definitiva
Xarxa 3	Encaminador Bcn Encaminador Val	1 3	Encaminador Bcn
Xarxa 4	Encaminador Bcn Encaminador Val	1 3	Encaminador Bcn
Xarxa 6	Encaminador Bcn Encaminador Val	1 2	Encaminador Bcn
Xarxa 7	Encaminador Bcn Encaminador Val	2 1	Encaminador Val

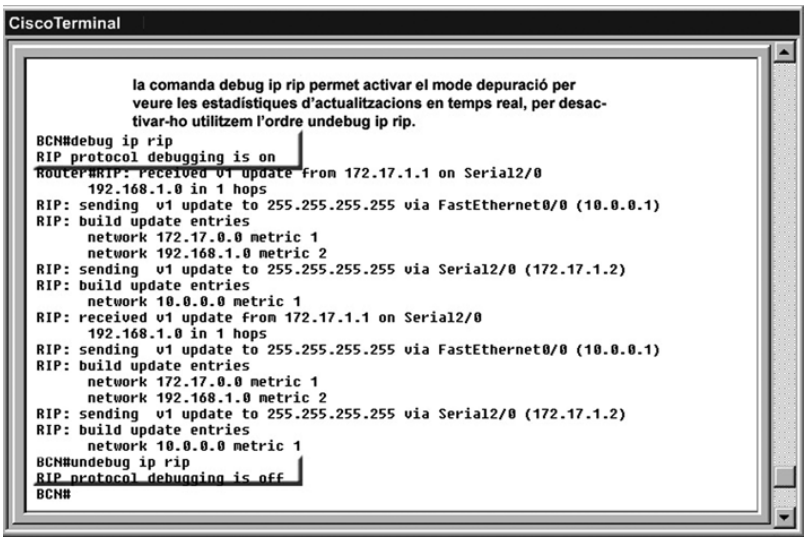
L'encaminador Mad de la taula 2.6 pot arribar a la xarxa 3 per dues rutes: per l'encaminador Bcn o per l'encaminador Val. Per determinar quina de les rutes és la millor, compta el nombre de salts que hauran de fer els paquets de dades i selecciona la ruta amb un valor de mètrica més baix, en aquest cas Bcn.

Si una de les xarxes o un encaminador no està disponible, els protocols d'encaminament s'encarreguen d'actualitzar les taules dels veïns i de seleccionar una altra ruta, encara que tingui una mètrica més elevada.

Quan una de les rutes cau, els protocols d'encaminament permeten que les taules d'encaminament s'actualitzin automàticament. Les actualitzacions dinàmiques poden generar problemes de redundància de rutes, les xarxes han de convergir, això vol dir que tot i que hi ha rutes redundants per arribar a una destinació tots els encaminadors han de tenir taules coherents entre elles.

La figura 2.8 ens mostra el resultat d'executar l'ordre **debug ip rip**, la qual activa el mode de depuració del protocol IP i ens mostra estadístiques d'ús de manera contínua. Per desactivar-lo utilitzem l'ordre **undebug ip rip**.

FIGURA 2.8. Resultat d'executar debug ip rip



Configuració de la versió 2 del protocol RIP

El RIPv2, la versió 2 del protocol d'encaminament RIP, es va crear per solucionar les deficiències de la primera versió del RIP. L'any 1993 es va desenvolupar la versió 2 de RIP, i l'any 1998 es va estandarditzar.

La millora principal era la capacitat de treballar amb subxarxes i CIDR, encara que permetia la compatibilitat amb la versió anterior del RIP. Una altra diferència és que els paquets amb la taula d'encaminament s'envien a l'adreça *multicast* 224.0.0.9 en lloc de l'adreça *broadcast*, com feia el RIPv1, la qual cosa evita la càrrega innecessària dels *hosts* de la xarxa.

Per veure la versió del RIP amb què està treballant l'encaminador, heu de fer servir l'ordre **show ip protocols** des del mode d'execució privilegiat una vegada heu configurat el protocol RIP, per exemple:

```
1  Router>enable
2  Router#conf t
3  Enter configuration commands, one per line.  End with CNTL/Z.
4  Router(config)#router rip
5  Router(config-router)#network 192.168.1.0
6  Router(config-router)#exit
7  Router(config)#exit
8  Router#
9  %SYS-5-CONFIG_I: Configured from console by console
10
11 Router#show ip protocols
12 Routing Protocol is "rip"
13 Sending updates every 30 seconds, next due in 10 seconds
14 Invalid after 180 seconds, hold down 180, flushed after 240
15 Outgoing update filter list for all interfaces is not set
16 Incoming update filter list for all interfaces is not set
17 Redistributing: rip
18 Default version control: send version 1, receive any version
19   Interface             Send  Recv  Triggered RIP  Key-chain
20 Automatic network summarization is in effect
21 Maximum path: 4
22 Routing for Networks:
23   192.168.1.0
24 Passive Interface(s):
25 Routing Information Sources:
```

```
  Gateway         Distance      Last Update
Distance: (default is 120)
Router#
```

Si us hi fixeu, l'ordre **show ip protocols** ens diu que es fa servir el RIP per redistribuir la informació d'encaminament, i que s'envien els paquets de la versió 1 del RIP, però es reben els paquets de qualsevol versió. Aquesta és l'opció per defecte, i serveix per mantenir la compatibilitat amb encaminadors que estiguin fent servir el RIPv1 (com els encaminadors més antics).

Per configurar l'encaminador per treballar amb el RIPv2, heu de fer servir l'ordre **version 2**.

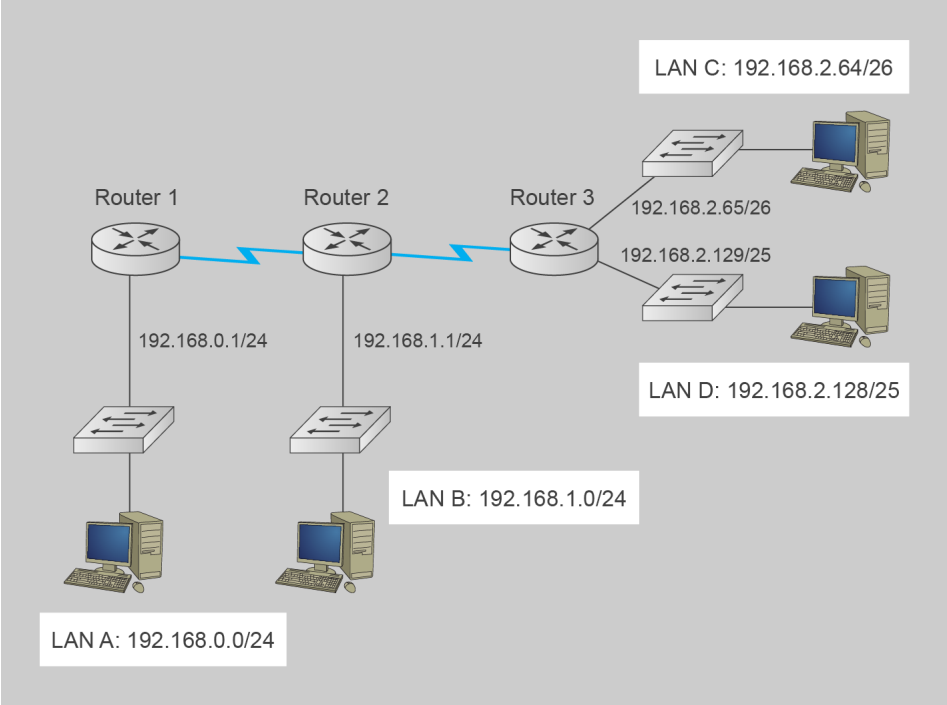
```
Router#enable
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 26 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive 2
  Interface             Send  Recv  Triggered RIP  Key-chain
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  192.168.1.0
Passive Interface(s):
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: (default is 120)
Router#
```

Vegem a continuació un exemple del funcionament del RIPv2. Mireu la xarxa de la figura 2.9.

Activem la configuració del RIPv2 en els tres encaminadors i configurem les xarxes a les quals estan connectats. Per exemple, en l'encaminador Router1

```
Router1>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#router rip
Router1(config-router)#network 192.168.0.0
Router1(config-router)#network 192.168.10.0
```

FIGURA 2.9. Configuració del RIPv2



En el cas de l'encaminador Router3 afegim les dues subxarxes i la xarxa de la línia sèrie:

```
Router3(config)#router rip
Router3(config-router)#network 192.168.2.64
Router3(config-router)#network 192.168.2.128
Router3(config-router)#network 192.168.11.0
```

Si mireu la configuració amb **show ip protocols** veureu que s'ha fet l'agregació automàtica de subxarxes i consta com a xarxa connectada directament la 192.168.2.0 (que inclou les dues subxarxes a les quals està connectat el Router3).

```
Router3#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 27 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive 2
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       2     2
  FastEthernet0/1       2     2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  192.168.2.0
Passive Interface(s):
Routing Information Sources:
  Gateway         Distance      Last Update
Distance: (default is 120)
```

De fet, si consulteu la taula de rutes des del Router1, podreu veure que el RIPv2 ha propagat l'agregació de subxarxes.

```
router3#show ip route
Codes: C – connected, S – static, I – IGRP, R – RIP, M – mobile, B – BGP
       D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
       N1 – OSPF NSSA external type 1, N2 – OSPF NSSA external type 2
       E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
       i – IS–IS, L1 – IS–IS level–1, L2 – IS–IS level–2, ia – IS–IS inter area
       * – candidate default, U – per–user static route, o – ODR
       P – periodic downloaded static route

Gateway of last resort is not set

R    192.168.0.0/24 [120/2] via 192.168.11.1, 00:00:16, Serial0/1/0
R    192.168.1.0/24 [120/1] via 192.168.11.1, 00:00:16, Serial0/1/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.64/26 is directly connected, FastEthernet0/1
C       192.168.2.128/25 is directly connected, FastEthernet0/0
R    192.168.10.0/24 [120/1] via 192.168.11.1, 00:00:16, Serial0/1/0
C    192.168.11.0/24 is directly connected, Serial0/1/0
router3#
```

En cas que es vulgui modificar la forma de funcionament per defecte del RIPv2, es pot cancel·lar l'agregació de subxarxes amb l'ordre **no auto-summary**:

```
router3(config)#router rip
router3(config–router)#no auto–summary
```

En aquest cas, la sortida de l'ordre **show ip protocols** mostra com s'estan encaminant les subxarxes de manera independent:

```
router3#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 14 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive 2
  Interface             Send  Recv  Triggered RIP  Key–chain
  FastEthernet0/0       2     2
  FastEthernet0/1       2     2
  Serial0/1/0           2     2
Automatic network summarization is not in effect
Maximum path: 4
Routing for Networks:
  192.168.2.0
  192.168.11.0
Passive Interface(s):
Routing Information Sources:
  Gateway         Distance      Last Update
  192.168.11.1         120      00:00:12
Distance: (default is 120)
```

Resolució de problemes i encaminament dinàmic

Quan es treballa amb encaminament dinàmic, sembla que en certa manera es perd el control de la xarxa, ja que els encaminadors s'envien informació automàticament i modifiquen les taules de rutes sense la participació de l'administrador. Per això, si alguna cosa no funciona correctament l'administrador no té, *a priori*, pistes sobre on s'ha pogut produir l'error. El millor és fer una comprovació exhaustiva i sistemàtica de la xarxa:

1. En primer lloc, s'ha de comprovar que totes les interfícies tenen una adreça IP i una màscara correctes. Es pot comprovar amb l'ordre **show ip interface brief**.

2. En el cas dels encaminadors s'ha de comprovar que les interfícies estiguin actives. Per defecte les interfícies dels encaminadors no ho estan. Es pot comprovar amb l'ordre **show ip interface brief**.

3. En les línies sèrie s'ha de comprovar que un dels extrems de la línia està configurat com a DCE i l'altre com a DTE, i que s'ha definit la freqüència del senyal de sincronisme (generalment a 56000).

4. S'ha de comprovar que hi ha connectivitat entre les interfícies contigües. Es pot fer servir l'ordre **ping *IP destinació*** per assegurar-se que hi ha connectivitat.

5. Es pot comprovar l'estat de la taula d'encaminament dels encaminadors amb l'ordre **show ip route**. Aquí s'han de mostrar totes les rutes a altres xarxes definides a l'encaminador, tant de manera estàtica com dinàmica.

6. S'ha de comprovar que el RIP estigui habilitat amb **show ip protocols**, també es pot veure la versió del RIP, l'estat de l'agregació automàtica de subxarxes i les xarxes que s'han afegit al protocol d'encaminament.

7. També es pot consultar el fitxer de configuració en execució (*running-config*), ja que dóna molta de la informació de configuració de l'encaminador de manera resumida.

8. S'ha d'activar la depuració del RIP amb l'ordre **debug ip rip**. D'aquesta manera es pot comprovar el contingut de les actualitzacions d'encaminament que l'encaminador envia i rep. Per exemple, amb la informació de depuració, l'administrador de la xarxa pot veure que es rep una ruta però que no s'afegeix a la taula d'enrutament. Això pot passar si hi ha alguna ruta estàtica definida per la mateixa xarxa. Per defecte, una ruta estàtica té una distància més curta que qualsevol ruta de qualsevol protocol d'encaminament i, per tant, tindrà prioritat a l'hora de ser afegida a la taula d'encaminament. Els missatges de depuració són constants, per tant, una vegada s'ha obtingut la informació necessària és habitual desactivar-lo amb l'ordre **no debug ip rip** o **undebug ip rip**.

```
router2#debug ip rip
RIP protocol debugging is on
router2#RIP: received v2 update from 192.168.11.2 on Serial0/1/1

      192.168.2.64/26 via 0.0.0.0 in 1 hops

      192.168.2.128/25 via 0.0.0.0 in 1 hops

RIP: sending  v2 update to 224.0.0.9 via FastEthernet0/0 (192.168.1.1)

RIP: build update entries

      192.168.0.0/24 via 0.0.0.0, metric 2, tag 0

      192.168.2.0/24 via 0.0.0.0, metric 2, tag 0

```

```
      192.168.10.0/24 via 0.0.0.0, metric 1, tag 0

      192.168.11.0/24 via 0.0.0.0, metric 1, tag 0

RIP: sending  v2 update to 224.0.0.9 via Serial0/1/1 (192.168.11.1)

router2#no debug ip rip
RIP protocol debugging is off
```

Configuració del protocol IGRP

El protocol IGRP és un protocol d'encaminament interior propietat de Cisco. Aprofita les característiques del RIP però millora la generació de rutes utilitzant altres mètriques que no són el nombre de salts, sinó les següents:

- **Retard**. És una de les mètriques utilitzades, és la suma de retards de l'origen a la destinació.
- **Amplada de banda**. És una de les mètriques utilitzades i es calcula tenint com a referència l'amplada de banda més baixa de totes les connexions del trajecte.
- **Càrrega**. És un valor comprès entre 1 (ús mínim) i 255 (saturació de l'enllaç), indica la quantitat de trànsit o saturació en un segment de la xarxa durant un període de temps, normalment cinc minuts. Per defecte no està activa i està ajustada a 0.
- **Confiabilitat**. És un valor que funciona al revés de la càrrega, està comprès entre 1 (enllaç totalment ple d'errors) i 255 (enllaç lliure d'errors), analitza el trànsit de la xarxa d'un període de temps, normalment els cinc minuts anteriors per determinar els errors que s'han produït. Per defecte no està activa i està ajustada a 0.

Protocol d'encaminament IGRP

L'IGRP és un protocol d'encaminament de vector-distància. Utilitza com a mètrica paràmetres de l'estat de la línia i el nombre de salts per generar les taules d'encaminament. Aquest protocol s'ha de configurar en tots els encaminadors que volem que intercanviïn informació, a més s'han d'especificar les xarxes per les quals treballarà el protocol i un número de sistema autònom.

Els temps d'actualització entre encaminadors és de 90 segons i s'utilitzen paquets de difusió o *broadcast*.

Per activar l'IGRP s'utilitzen les mateixes ordres que per al RIP, però heu d'indicar un paràmetre addicional, que és el número de sistema autònom. Aquest valor indica el conjunt d'encaminadors que poden intercanviar les taules d'actualització. S'anomena *domini de processos*.

```
Router(config)#router igrp 100
Router(config–router)#network 192.168.1.0
```

En l'exemple anterior, cal que observeu que hem indicat a l'encaminador que el protocol IGRP enviarà i rebrà actualitzacions dels encaminadors que utilitzen el valor de sistema autònom 100 per la interfície que té configurada una adreça IP de la xarxa 192.168.1.0.

Configuració del protocol EIGRP

El protocol EIGRP és una millora del protocol IGRP; també és propietat de Cisco Systems. Utilitza tres taules d'informació per generar les rutes:

- **Veïnat**. Conté els encaminadors que utilitzen el protocol EIGRP i estan directament connectats al nostre dispositiu, són els que transfereixen la seva taula d'encaminament.
- **Topològica**. Mostra les rutes i el seu estat, la mètrica i la distància als encaminadors veïns per a cadascuna de les rutes. Conté informació de tota la topologia de la xarxa.
- **Encaminament**. A partir de la taula topològica es genera la taula de rutes.

Per configurar l'EIGRP utilitzem les mateixes ordres que amb l'IGRP. També s'ha d'especificar un valor de sistema autònom:

```
1 Router(config)#router eigrp 100
2 Router(config–router)#network 192.168.1.0
```

El **sistema autònom** ens indica quins encaminadors poden intercanviar les taules; aquest valor el defineix l'administrador.

Els encaminadors utilitzen els paquets HELLO per conèixer els encaminadors veïns. Són paquets de salutació entre encaminadors que utilitzen l'EIGRP, gràcies als quals el protocol pot construir les taules amb informació dels veïns i la topologia de la xarxa.

Una vegada els encaminadors estan configurats amb el protocol EIGRP, aquests comencen a enviar paquets de salutació o HELLO als veïns de la xarxa i es construeix la taula d'encaminadors veïns amb què intercanviarem actualitzacions. A continuació es genera la taula d'encaminament.

No s'utilitzen actualitzacions cada 30 o 90 segons com en altres protocols, l'encaminador detecta si un veí és actiu mitjançant un missatge de salutació aproximadament cada 60 segons.

Les rutes generades poden tenir diferents estats que es mostren en la taula 2.7.

TAULA 2.7. Estats d'una ruta generada amb l'EIGRP

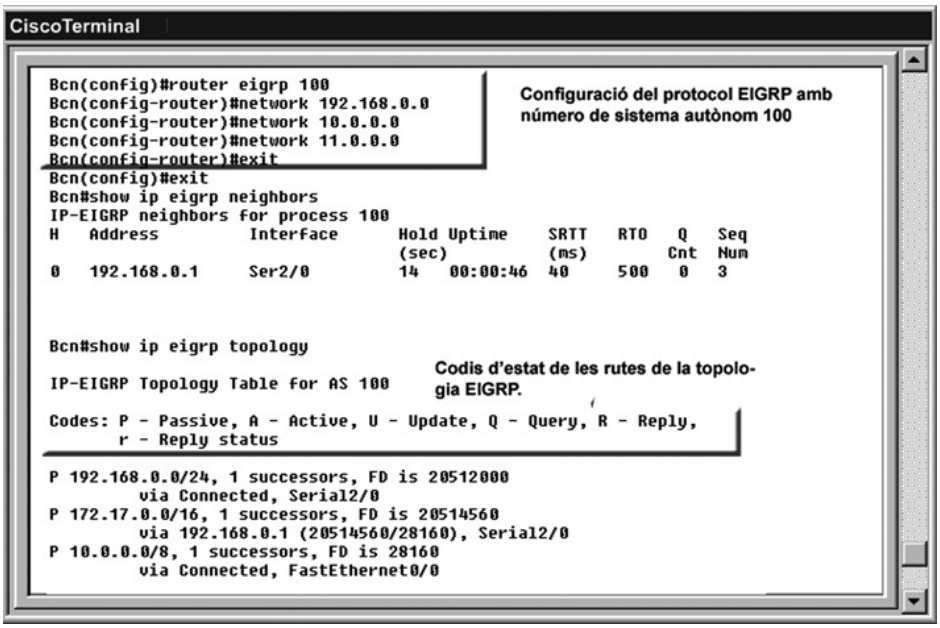
Estat de la ruta	Descripció
P-PASSIVE	Estat normal de la ruta.
A-ACTIVE	La ruta està efectuant càlculs.
U-UPDATE	S'estan enviant actualitzacions.
Q-QUERY	S'està enviant una consulta a aquesta ruta.
R-REPLY	S'està enviant una rèplica a aquesta ruta.
r-REPLY STATUS	S'espera una rèplica.

Per veure el contingut de les taules i la informació dels veïns hi ha les ordres següents. En general, amb l'ordre **show ip eigrp *nom de la taula*** es poden visualitzar les taules que fa servir el protocol EIGRP, per exemple, les taules de veïns o de topologia:

- **show ip eigrp neighbors**. Ens mostra quins encaminadors es consideren veïns i a quina interfície estan connectats.
- **show ip eigrp topology**. Ens mostra la taula de topologia amb totes les rutes incorporades i l'estat en què estan. S'utilitza per escollir successors viables i, per tant, es construeix la taula d'encaminament amb aquesta informació. Aquesta ordre té diversos modificadors per filtrar els resultats que es mostren en la consola.
- **show ip eigrp traffic**. Podem veure les estadístiques de trànsit del protocol EIGRP.

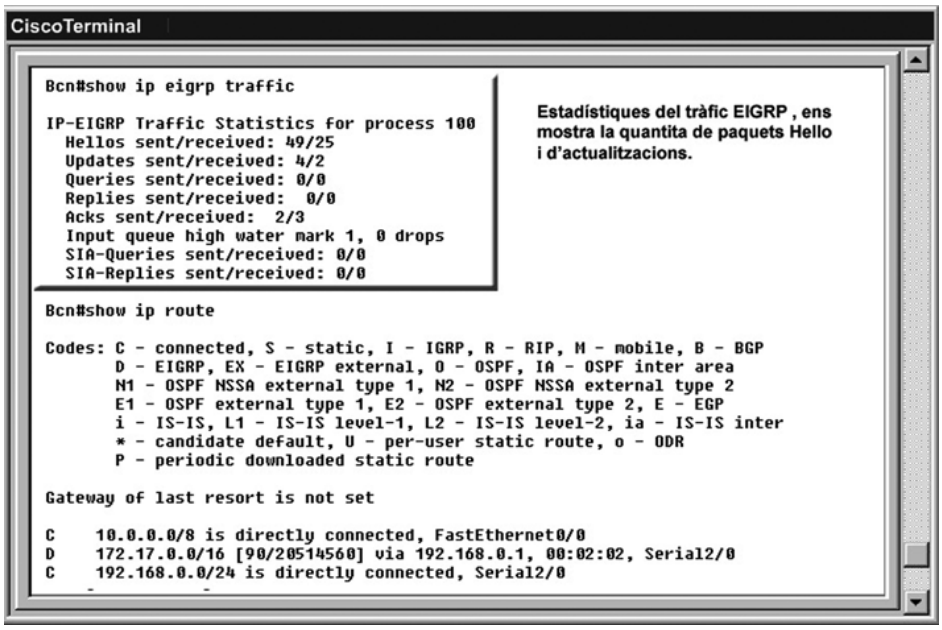
En la figura 2.10 podeu veure com es configura l'EIGRP, l'execució de les ordres **show ip eigrp neighbors** i **show IP eigrp topology**.

FIGURA 2.10. Configuració de l'EIGRP



La figura 2.11 mostra les estadístiques EIGRP i la taula de rutes, en què es pot veure l'ús de les ordres **show ip eigrp traffic** i **show ip route**.

FIGURA 2.11. Resultat d'executar sh ip eigrp traffic



2.6.2 Protocols estat de l'enllaç

Els protocols més comuns de l'estat de l'enllaç és **OSPF** (*Open Shortest Path First*, obre primer el camí més curt). És un protocol d'encaminament autoadaptatiu per a IP. Treballa dins d'un únic sistema autònom. És probablement el protocol interior o **IGP** (*Interior Gateway Protocol*, protocol interior de passarel·la) més utilitzat a xarxes de grans empreses.

Configuració del protocol OSPF

OSPF (*Open Shortest Path First*, obre primer el camí més curt) és un protocol d'encaminament interior que funciona dins d'un sistema autònom. Recopila informació dels encaminadors de la xarxa i construeix un mapa de la topologia de la xarxa, amb la qual es decideix la ruta que prendran els paquets. OSPF està dissenyat per funcionar amb CIDR i VLSM.

OSPF detecta canvis a la topologia (per exemple si un encaminador ha deixat de funcionar) i recalcula les rutes ràpidament. Calcula les rutes fent servir l'algorisme de Dijkstra (algorisme que obre en primer lloc el camí més curt, d'aquí el nom del protocol) que dóna com resultat un arbre d'expansió mínima lliure de bucles.

Les característiques principals d'OSPF són les següents:

- **Redueix el consum de xarxa**. No hi ha actualitzacions periòdiques com en els protocols de vector distància, hem de tenir en compte que tots els encaminadors coneixen el mapa de la xarxa.

- **Temps curt de convergència**. Com que es disposa de tot el mapa de xarxa quan cau una línia automàticament s'estableix una ruta alternativa.
- **Topologia sense bucles**. No es permeten bucles d'encaminament; es fa una selecció de ruta amb l'algoritme SPF (*shortest path first* o primer el camí més curt) que determina el camí més curt a totes les destinacions.
- **Multiplataforma**. Com que OSPF és un estàndard obert, el pot utilitzar qualsevol fabricant en els seus dispositius.

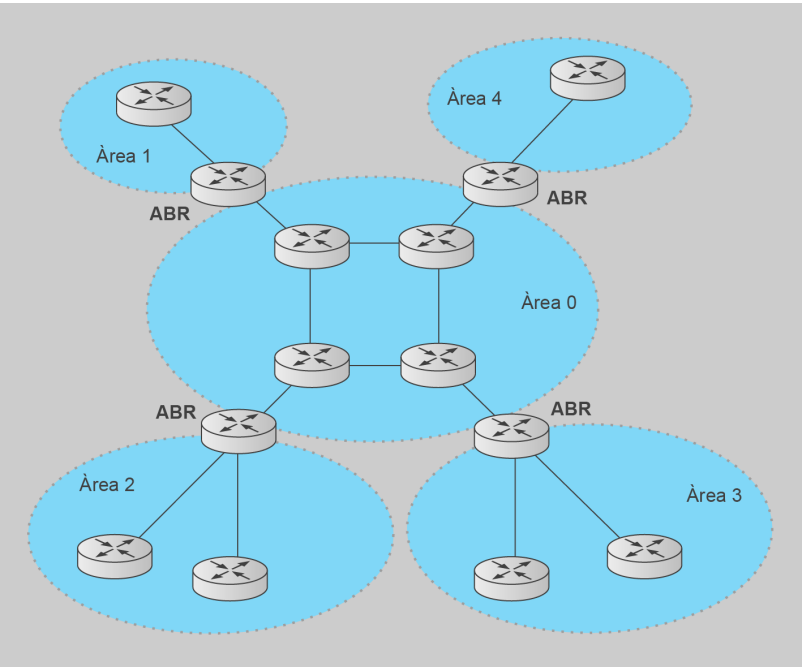
El protocol OSPF utilitza les mateixes taules que EIGRP:

- Taula de veïns
- Taula topològica
- Taula d'encaminament

Una xarxa OSPF es pot dividir en àrea d'enrutament per simplificar l'administració i optimitzar el tràfic. Les àrees estan identificades per números de 32 bits. Es poden expressar en decimal o en decimal separades per punts, semblant a la nomenclatura de les adreces IP (per exemple: 123.21.44.74).

Per convenció l'àrea 0 (o 0.0.0.0) representa l'àrea o regió de backbone a una xarxa OSPF. La identificació de les altres àrees es deixa a selecció de l'administrador (una tècnica habitual és escollir l'adreça IP de l'encaminador principal de l'àrea com identificador d'àrea). Totes les àrees addicionals han de tenir una connexió amb l'àrea del backbone a través d'un encaminador anomenat ABR (*Area Border Router*, encaminador de borde d'àrea), com es pot veure a la figura 2.12.

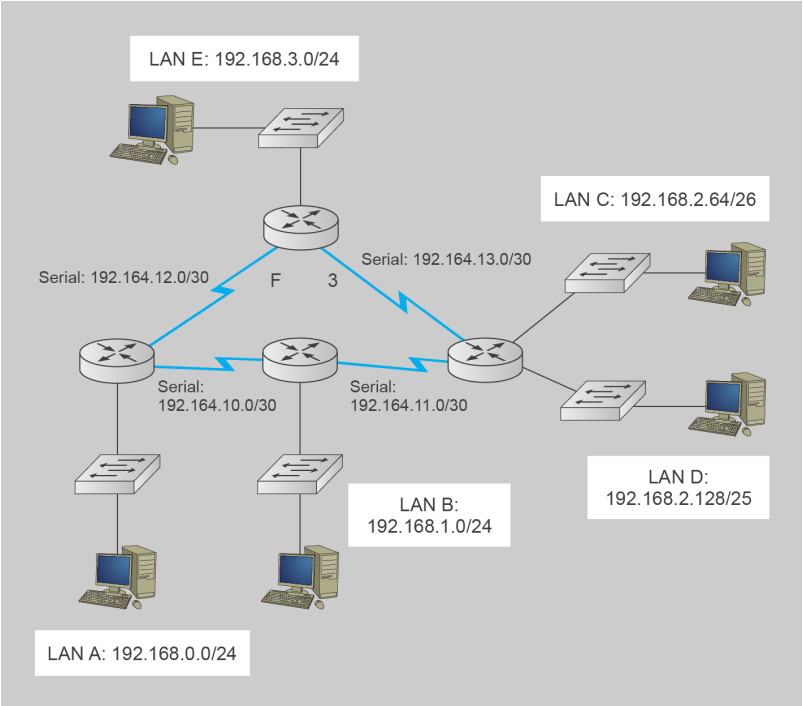
Figura 2.12. Esquema de xarxa OSPF



Totes les àrees han d'estar unides al backbone per un ABR

El protocol OSPF pot funcionar de forma segura entre els encaminadors, fent servir opcionalment diversos mètodes d'autenticació per permetre únicament als encaminadors autenticats participar en l'algorisme d'encaminament.

FIGURA 2.13. Xarxa OSPF



Les línies sèrie tenen una màscara de 30 bits, suficient per configurar dues interfícies

Anem a veure la configuració d'OSPF a través d'una xarxa d'exemple (figura 2.13). A aquesta xarxa tenim tres xarxes de classe C (LAN A, B i E) i una xarxa de classe C dividida en dues subxarxes (LAN C i D). Fixeu-vos com s'han configurat les línies sèrie dels encaminadors amb xarxes de màscara 30. Una xarxa amb màscara de 30 bits té únicament 2 bits de host, que ens deixen 4 adreces de host (00, 01, 10 i 11). Com la primera i la última estan reservades a la identificació de la xarxa i de la broadcast respectivament ens queden únicament dues adreces IP útils per configurar interfícies (1 i 2), suficient per configurar una línia sèrie. Configurar línies sèrie amb xarxes de 30 bits és una pràctica habitual a les xarxes locals.

L'ordre per habilitar OSPF és **router ospf *Identificador de procés***, on l'identificador pot ser un número entre 1 i 65535 escollit per l'administrador. Tot i que no és necessari que sigui el mateix a tots els encaminadors anem a configurar-los tots igual **router ospf 1**. Aquesta ordre passarà al mode de configuració de router.

```
1 R0>enable
2 R0#conf t
3 Enter configuration commands, one per line.  End with CNTL/Z.
4 R0(config)#router ospf 1
```

L'ordre **network** serveix per indicar les xarxes a les que està connectat l'encaminador. En aquest cas l'ordre és una mica més complicada que en altres protocols i la seva sintaxi és **network *IP_Xarxa Màscara_Wildcard* area *ID_àrea***. La màscara wildcard serveix per identificar el rang d'adreces que s'està afegint, i consisteix en la inversió dels bits de la màscara de xarxa. Per exemple una xarxa de màscara 255.255.255.0 tindrà un wildcard de 0.0.0.255 i una màscara

de 255.255.255.192 tindrà un wildcard de 0.0.0.63. L'argument **ID_àrea** és l'identificador de l'àrea OSPF. Tots els encaminadors de la mateixa àrea han de tenir el mateix ID d'àrea configurat. En aquest cas anem a configurar els encaminadors a l'àrea 0 (backbone), ja que anem a treballar amb una única àrea. Per exemple la configuració de l'encaminador R2 serà:

```
R2(config-router)#network 192.168.2.64 0.0.0.63 area 0
R2(config-router)#network 192.168.2.128 0.0.0.127 area 0
R2(config-router)#network 192.168.13.0 0.0.0.3 area 0
R2(config-router)#network 192.168.11.0 0.0.0.3 area 0
```

Si no s'especifica manualment, l'algorisme d'OSPF agafa l'adreça IP més alta de l'encaminador com identificador d'encaminador pels algorismes d'OSPF. Si es vol escollir un altre identificador s'ha de fer servir l'ordre **router-id *ID d'encaminador***. Recordeu que l'identificador d'encaminador és un valor de 32 bits semblant a una adreça IP, per exemple:

```
R0(config-router)#router-id 192.168.10.1
```

Podeu comprovar l'identificador d'encaminador que s'està fent servir amb l'ordre **show ip protocols** des del mode d'execució privilegiat:

```
R0#show ip protocols

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.12.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    192.168.0.0 0.0.0.255 area 0
    192.168.10.0 0.0.0.3 area 0
    192.168.12.0 0.0.0.3 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.11.1         110      00:04:49
    192.168.12.1         110      00:04:32
    192.168.13.1         110      00:04:32
    192.168.13.2         110      00:05:21
  Distance: (default is 110)
```

En aquest cas fixeu-vos que l'encaminador R0 de la xarxa té com ID l'adreça IP més alta que té configurada (192.168.12.1). Aquesta ordre també mostra les xarxes que l'encaminador està publicant als veïns (192.168.0.0, 192.168.10.0 i 192.168.12.0), els encaminadors des d'on l'encaminador rep actualitzacions OSPF (192.168.11.1, 192.168.12.1, 192.168.13.1, 192.168.13.2) i la seva distància administrativa (110).

Per verificar els veïns OSPF de l'encaminador es pot fer servir l'ordre **show ip ospf neighbor**. Per exemple, el resultat per l'encaminador R0

```
R0#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.11.1      0   FULL/  -        00:00:40    192.168.10.2    Serial0/1/0
192.168.13.1      0   FULL/  -        00:00:30    192.168.12.2    Serial0/1/1
```

El resultat de l'ordre mostra:

- **Neighbor ID**: l'identificador de l'encaminador veí.
- **Pri**: prioritat OSPF de l'interfície.
- **State**: estat OSPF de l'interfície. L'estat FULL vol dir que l'encaminador i el seu veí tenen les mateixes bases de dades OSPF.
- **Dead time**: temps d'espera per rebre un missatge de salutació del protocol OSPF per aquesta interfície.
- **Address**: adreça IP de la interfície del veí a la que està connectat l'encaminador.
- **Interface**: interfície de l'encaminador amb la que està connectat al veí.

Amb aquesta ordre podeu comprovar si hi ha hagut algun error en la configuració d'OSPF. Els encaminadors han de poder veure l'estat dels seus veïns amb aquesta ordre i han de tenir l'estat de l'interfície a **FULL** perquè els encaminadors puguin intercanviar informació del protocol.

Una vegada heu configurat OSPF podeu comprovar que s'han afegit les rutes a la taula de rutes de l'encaminador. Anem a observar les taules de rutes de l'encaminador R0:

```
R0#show ip route
Codes: C – connected, S – static, I – IGRP, R – RIP, M – mobile, B – BGP
       D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
       N1 – OSPF NSSA external type 1, N2 – OSPF NSSA external type 2
       E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
       i – IS–IS, L1 – IS–IS level–1, L2 – IS–IS level–2, ia – IS–IS inter area
       * – candidate default, U – per–user static route, o – ODR
       P – periodic downloaded static route

Gateway of last resort is not set

C    192.168.0.0/24 is directly connected, FastEthernet0/0
O    192.168.1.0/24 [110/65] via 192.168.10.2, 00:02:52, Serial0/1/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
O       192.168.2.64/26 [110/129] via 192.168.12.2, 00:02:42, Serial0/1/1
                        [110/129] via 192.168.10.2, 00:02:42, Serial0/1/0
O       192.168.2.128/25 [110/129] via 192.168.12.2, 00:02:42, Serial0/1/1
                         [110/129] via 192.168.10.2, 00:02:42, Serial0/1/0
O    192.168.3.0/24 [110/65] via 192.168.12.2, 00:02:52, Serial0/1/1
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial0/1/0
     192.168.11.0/30 is subnetted, 1 subnets
O       192.168.11.0 [110/128] via 192.168.10.2, 00:02:52, Serial0/1/0
     192.168.12.0/30 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, Serial0/1/1
     192.168.13.0/30 is subnetted, 1 subnets
O       192.168.13.0 [110/128] via 192.168.12.2, 00:02:52, Serial0/1/1
R0#
```

Podeu comprovar que totes les rutes segueixen els camins esperats. Fixeu-vos que les entrades de la taula creades per OSPF tenen una O al començament de línia. Si analitzem les línies del resultat de l'ordre:

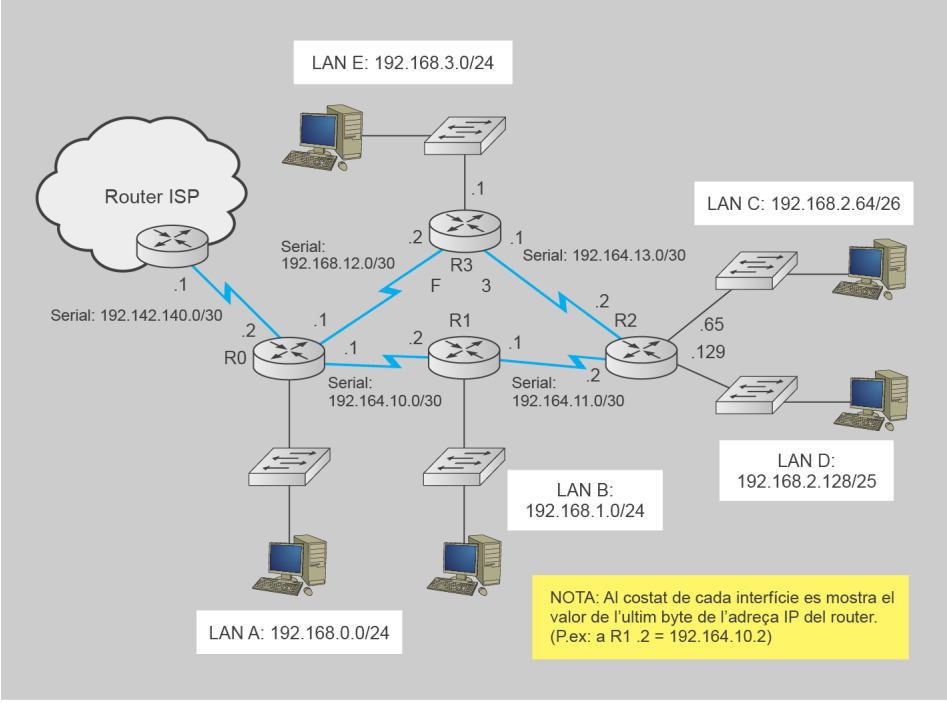
- 192.168.0.0/24 està connectada directament i accedeix a través de la interfície FastEthernet 0/0.

- 192.168.1.0/24 està disponible a través de 192.168.10.2, que es troba per la interfície Serial0/1/0 de l'encaminador.
- 192.168.2.0/24 té dues subxarxes (ha fet agregació de xarxes) que són 192.168.2.64/26 i 192.168.2.128/25. Es poden accedir o bé, per 192.168.12.2 (a través de la interfície Serial0/1/1) o, per 192.168.10.2 (a través de la interfície Serial0/1/0).
- 192.168.3.0/24 s'accedeix via 192.168.12.2 (interfície Serial0/1/1).
- 192.168.10.0 està connectada directament.
- 192.168.11.0 s'accedeix a través de 192.168.10.2 (interfície Serial0/1/0).
- 192.168.12.0 està connectada directament a Serial0/1/1
- 192.168.13.0 s'accedeix a través de 192.168.12.2 (interfície Serial0/1/1).

Configuració d'una ruta per defecte a OSPF

Imagineu que heu connectat la xarxa de l'exemple anterior a Internet a través d'un ISP. La topologia de la xarxa seria ara la que mostra la figura 2.14.

Figura 2.14. Xarxa connectada a un ISP



Ruta per defecte

Una ruta per defecte és com un comodí, qualsevol paquet que no es pugui enrutar per cap de les rutes definides a la taula d'enrutament de l'encaminador s'enviarà per la ruta per defecte.

Sempre que es connecta una xarxa a Internet, s'han de definir rutes per defecte, ja que no és possible definir als encaminadors rutes per totes les xarxes que existeixen a Internet.

La informació d'enrutament se realitzarà entre els encaminadors de la xarxa local, entre els encaminadors que formen part del sistema autònom (AS, *Autonomous System*) de la xarxa. Un encaminador que connecta un sistema autònom amb una

altre o amb una xarxa que no fa servir OSPF (en l'exemple de la figura 2.14 seria l'encaminador R0) s'anomena ASBR (*Autonomous System Boundary Router*) o encaminador frontera de sistema autònom.

Per definir una ruta per defecte, l'administrador ha de configurar-la de manera manual amb l'ordre **ip route 0.0.0.0 0.0.0.0** ***IP destinació*** o **ip route 0.0.0.0 0.0.0.0** ***Interfície de sortida***.

> No definiu la ruta per defecte dues vegades, escolliu un dels dos mètodes i configureu-la una única vegada.

Així l'encaminador R0 per definir la ruta per defecte haurà d'executar:

```
R0(config)#ip route 0.0.0.0 0.0.0.0 Serial 0/0/0
```

en el cas d'estar connectar a l'encaminador de l'ISP per la interfície Serial 0/0/0 o bé:

```
R0(config)#ip route 0.0.0.0 0.0.0.0 143.42.140.1
```

Si mireu les taules d'enrutament veureu que s'ha afegit la ruta per defecte com una ruta estàtica:

```
R0#show ip route
Codes: C – connected, S – static, I – IGRP, R – RIP, M – mobile, B – BGP
       D – EIGRP, EX – EIGRP external, O – OSPF, IA – OSPF inter area
       N1 – OSPF NSSA external type 1, N2 – OSPF NSSA external type 2
       E1 – OSPF external type 1, E2 – OSPF external type 2, E – EGP
       i – IS–IS, L1 – IS–IS level–1, L2 – IS–IS level–2, ia – IS–IS inter area
       * – candidate default, U – per–user static route, o – ODR
       P – periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     142.42.0.0/30 is subnetted, 1 subnets
C       142.42.140.0 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, FastEthernet0/0
O    192.168.1.0/24 [110/65] via 192.168.10.2, 00:02:23, Serial0/1/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
O       192.168.2.64/26 [110/129] via 192.168.12.2, 00:02:23, Serial0/1/1
                        [110/129] via 192.168.10.2, 00:02:23, Serial0/1/0
O       192.168.2.128/25 [110/129] via 192.168.12.2, 00:02:23, Serial0/1/1
                         [110/129] via 192.168.10.2, 00:02:23, Serial0/1/0
O    192.168.3.0/24 [110/65] via 192.168.12.2, 00:02:23, Serial0/1/1
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial0/1/0
     192.168.11.0/30 is subnetted, 1 subnets
O       192.168.11.0 [110/128] via 192.168.10.2, 00:02:23, Serial0/1/0
     192.168.12.0/30 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, Serial0/1/1
     192.168.13.0/30 is subnetted, 1 subnets
O       192.168.13.0 [110/128] via 192.168.12.2, 00:02:23, Serial0/1/1
S*   0.0.0.0/0 is directly connected, Serial0/0/0
Router#
```

Perquè es propagui la ruta per defecte a la resta d'encaminadors del sistema autònom OSFP s'ha d'executar l'ordre **default-information originate** des del mode de configuració d'encaminador.

```
Router(config)#router ospf 1
Router(config-router)#default-information originate
```

Una vegada fet això, podeu comprovar que la ruta per defecte s'ha propagat a la resta d'encaminadors del sistema autònom OSPF. Per exemple a l'encaminador R1:

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 192.168.10.1 to network 0.0.0.0

O    192.168.0.0/24 [110/65] via 192.168.10.1, 00:01:40, Serial0/1/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
O       192.168.2.64/26 [110/65] via 192.168.11.2, 00:01:40, Serial0/1/1
O       192.168.2.128/25 [110/65] via 192.168.11.2, 00:01:40, Serial0/1/1
O    192.168.3.0/24 [110/129] via 192.168.11.2, 00:01:30, Serial0/1/1
                    [110/129] via 192.168.10.1, 00:01:30, Serial0/1/0
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial0/1/0
     192.168.11.0/30 is subnetted, 1 subnets
C       192.168.11.0 is directly connected, Serial0/1/1
     192.168.12.0/30 is subnetted, 1 subnets
O       192.168.12.0 [110/128] via 192.168.10.1, 00:01:40, Serial0/1/0
     192.168.13.0/30 is subnetted, 1 subnets
O       192.168.13.0 [110/128] via 192.168.11.2, 00:01:40, Serial0/1/1
O*E2 0.0.0.0/0 [110/1] via 192.168.10.1, 00:01:00, Serial0/1/0
R1#
```

Rutes OSPF

Les rutes OSPF es classifiquen en tipus 1 i 2, la diferència entre elles és la forma en que es calcula el cost d'OSPF de la ruta.

Fixeu-vos que s'ha configurat com encaminador d'últim recurs (*gateway last resort*) 192.168.10.1 que és justament l'interfície de l'encaminador R0. La ruta per defecte té el codi adicional **E2**,

```
O*E2 0.0.0.0/0 [110/1] via 192.168.10.1, 00:01:00, Serial0/1/0
```

que vol dir que és una ruta externa OSPF de tipus 2.