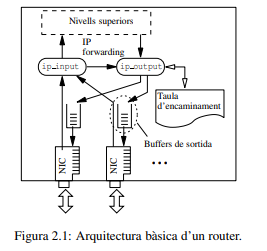
**Tema 2: Xarxes IP**

L’objectiu d’aquest tema és l’estudi de la Internet Protocol (IP) i alguns altres protocols relacionats amb la configuració i funcionament del nivell de xarxa d’una xarxa TCP/IP. El protocol IP es va dissenyar amb l’objectiu de poder interconnectar xarxes heterogènies. Això s’aconsegueix amb routers que “parlen” el mateix protocol: IP. Els routers tenen la funció d’encaminar les unitats de dades (PDUs en la terminologia OSI) que fa servir el protocol IP: els “datagrames”. La descripció del funcionament bàsic dels routers ens permet descriure les característiques més importants del protocol IP. La figura 2.1 mostra l’arquitectura bàsica d’un router. El seu funcionament el podríem resumir de la següent manera: un router té dues o més “interfícies” connectades a xarxes diferents. Físicament, les interfícies estan formades per targes de comunicacions (Network Interface Cards, NIC) que permeten transmetre o rebre informació a través d’una xarxa física específica: una LAN o una WAN.

****

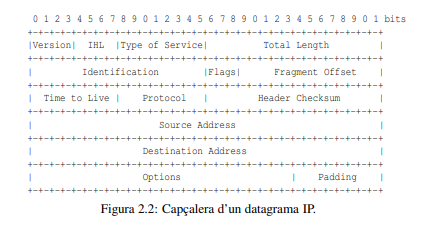
Quan un router rep un datagrama d’una interfície, segueix el següent procés:

1. El datagrama passa a la funció ip input, que mira si va dirigit al mateix router. En cas afirmatiu, el passa als nivells superiors, altrament el passa a la funció ip output. El pas de ip input a ip output es coneix com **IP forwarding** i és la diferència entre un host i un router. En el cas d’un host, l’IP forwarding està desactivat, i si el datagrama no va dirigit al mateix host es descarta.
2. La funció ip output s’encarrega de l’encaminament. Per a l’encaminament un router fa servir una “taula d’encaminament”. En aquesta taula hi ha les xarxes destinació on el router sap arribar. Per a cada xarxa destinació la taula diu la interfície per on ha d’enviar-se el datagrama.
3. Després de consultar la taula, ip output passa el datagrama al driver que controla la NIC per on ha d’enviar-se i aquest el guarda en un “buffer de sortida” a l’espera que la NIC l’agafi i l’enviï a la xarxa física on està connectada.

El procediment de rebre els datagrames, processar-los i emmagatzemar-los fins que es transmeten es coneix com a **store & forward**. Cal destacar que les taules d’encaminament del router han d’estar inicialitzades. Si el router rep un datagrama dirigit a una destinació desconeguda, el descarta. A més, si arriben més datagrames dirigits a una mateixa NIC del que és capaç¸ de transmetre, el buffer de sortida s’omplirà i el router començarà a descartar datagrames.

De la descripció anterior podem deduir les següents característiques del protocol d’IP:

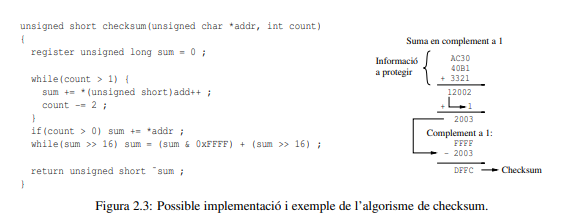
* No orientat a la connexió **connectionless**: és a dir, abans de començar a encaminar datagrames, un router no ha de registrar l’establiment d’una connexió. Això no és així en altres xarxes com ara la xarxa telefònica, on abans de començar a parlar hem d’establir una connexió i esperar que l’altre extrem respongui la trucada. Aquestes xarxes es diuen “orientades a la connexió” (connection oriented).
* Sense estat (**stateless**): és a dir, no guarda informació de les connexions en curs.
* No fiable: per exemple, els routers descarten datagrames si el buffer per on han d’encaminar-lo està ple. Aquest tipus de servei on els routers “fan el millor que poden” per encaminar el datagrama, i si no poden el descarten es coneix amb el nom de **best effort**.

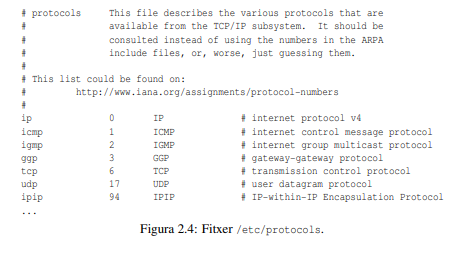
****

**2.1 Capçalera IP**

La figura 2.2 mostra la capçalera d’un datagrama IP. Els camps són els següents:

* Version: versió´ del protocol: 4.
* IHL: IP Header Length, mida de la capçalera en words de 32 bits (si la capçalera no té opcions val 5, és a dir, la capçalera té 5 · 4 = 20 bytes).
* Type of Service: preferència d’encaminament. El format és: xxxdtrc0, on xxx permeten indicar una precedència. Només té significat a l’interior d’una mateixa xarxa i no arreu d’Internet (no sol fer-se servir). L’últim bit es posa a 0 (no es fa servir). Els bits dtrc volen dir:
  + d: delay, optimitzar el retard.
  + t: throughput, optimitzar la velocitat eficaç¸.
  + r: reliability, optimitzar la fiabilitat.
  + c: cost, optimitzar el cost (econòmic).
* Total Length: Mida total del datagrama en bytes.
* Identification, Flags, Fragment Offset: es fan servir en la fragmentació (secció 2.1.1).
* Time to Live (TTL): Temps de vida. Els routers decrementen aquest camp i descarten el datagrama si arriba a zero, és a dir, executen el codi: if(--TTL == 0) { descartar el datagrama }. El motiu és evitar que hi hagi datagrames donant voltes indefinidament per Internet. Això podria passar si hi hagués algun bug en una taula d’encaminament, o si s’enviés un datagrama a una adreça inexistent.
* Protocol: multiplexació del protocol de nivell superior. Els números de protocol estan estandarditzats en el RFC 1700 (Assigned Numbers [36]). En una màquina UNIX es poden consultar en el fitxer /etc/protocols (vegeu la figura 2.4).
* Header Checksum: Permet la detecció d’errors. El checksum és només de la capçalera. L’algorisme de Checksum és el complement a 1 de la suma en complement a 1 de la informació a protegir. És a dir, pel càlcul del checksum es posen els bits del camp de checksum de la capçalera a 0, s’agrupen els bits de la capçalera en words de 16 bits, se sumen en complement a 1 i es fa el complement a 1 de la suma. La suma en complement a 1 consisteix en fer la suma en binari natural i tornar a sumar el possible excés (carry) que es pugui produir. El complement a 1 consisteix a canviar els ‘1’ per ‘0’ i viceversa. La figura 2.3 mostra una possible implementació de l’algorisme de checksum en C i un exemple del càlcul.
* Source Address, Destination Address: adreça font i destinació del datagrama (secció 2.2).
* Options: aquest camp pot portar una o més opcions. Normalment no en porta cap. Algunes de les que hi ha definides són:
  + Record Route: quan hi ha aquesta opció, els routers afegeixen l’adreça IP de la interfície per on encaminen el datagrama.
  + Loose Source Routing: especifica una llista d’adreces IP de routers que ha de travessar el datagrama (pot travessar també altres routers que no siguin a la llista).
  + Strict Source Routing: adreces IP dels únics routers que pot travessar el datagrama.

****



2.1.1 Fragmentació

IP pot fragmentar un datagrama quan la Maximum Transfer Unit (MTU) del nivell d’enllaç, on ha d’encaminar-se té una mida menor que la del datagrama. Això sol passar en els següents casos:

* Quan un router ha d’encaminar un datagrama per una xarxa amb MTU menor que la d’on ha arribat.
* Quan es fa servir UDP i l’aplicació fa una escriptura major que l’MTU de la xarxa.

Quan es produeix la fragmentació, el reensamblatge es fa en la destinació dels datagrames. Per poder fer el reensamblatge es fan servir els camps:

* Identification: El nivell IP del host que genera els datagrames hi posa el valor d’un comptador que incrementa cada cop que es genera un nou datagrama. Aquest número permet identificar fragments d’un mateix datagrama.
* Flags: són 0DM. El primer bit no es fa servir, els altres són:
  + D: Flag de Don’t fragment. Quan està a ‘1’ el datagrama no es pot fragmentar. Si un router necessita fer-ho el descartarà i enviarà un missatge ICMP d’error. Es fa servir en el mecanisme de MTU path discovery (vegeu la secció 2.1.2).
  + M: Flag de More fragments. Si està a ‘1’ vol dir que hi ha més fragments. Tots els fragments d’un datagrama el porten a ‘1’ excepte l’últim.
* Offset: posició del primer byte del fragment en el datagrama original (el primer val ‘0’). Es compta en unitats de 8 bytes (per poder comptar fins a 2 16−1 amb 13 bits).

2.1.2 MTU Path Discovery

El nivell de transport TCP agrupa els bytes que escriu l’aplicació fins a tenir un segment de mida òptima i després l’envia. Normalment la mida òptima és la que permet enviar els segments de mida igual a l’MTU menor de les xarxes que han de travessar fins a la destinació. Això permet enviar segments de mida gran (i així l’overhead de les capçaleres serà petit), però no tant com perquè s’hagin de fragmentar. La fragmentació no és desitjable perquè:

* (i) ralentitza els routers,
* (ii) pot provocar que hi hagi fragments de mida petita que redueixen l’eficiència de la xarxa i,
* (iii) si es perd un sol fragment, s’han de descartar tots els altres quan arriben a la destinació.

Per aconseguir que la mida dels segments sigui l’òptima, les implementacions actuals de TCP fan el mecanisme MTU Path Discovery, que consisteix en el següent: Primer proven d’adaptar la mida dels segments a l’MTU de la xarxa on està connectat el host. Els datagrames s’envien amb el bit de Don’t fragment activat. Si un router intermedi ha d’enviar-los per una xarxa d’MTU menor, descartaran el datagrama i enviaran un missatge ICMP d’error fragmentation needed but DF set (vegeu la secció 2.6). Una de les informacions que porta aquest missatge d’error és la mida de l’MTU que l’ha provocat. Quan el host rep el missatge d’error, TCP redueix la mida dels segments per adaptar-los en aquesta MTU.

Exemple de Fragmentació

Suposem que una aplicació d’un host genera un datagrama UDP de 1980 bytes que ha d’enviar-se per una xarxa Ethernet. El nivell IP construirà un datagrama de 2000 bytes: 20 de capçalera IP + 1980 de payload. Quan IP vegi que el datagrama ha d’encaminar-se per una interfície d’MTU = 1500 bytes, haurà de fragmentar-lo. El payload màxim dels fragments (en unitats de 8 bytes) serà:



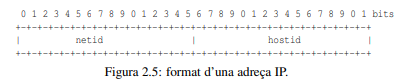
És a dir, 185 · 8 = 1480 bytes. Es generaran doncs 2 fragments: El primer fragment portarà els primers 1480 bytes del payload del datagrama original. El segon fragment portarà els 500 bytes restants. Al fragmentar-se, es copiarà la capçalera del datagrama original en cada fragment i es canviaran els camps:

* 1r fragment: offset = 0, M = 1.
* 2n fragment: offset = 185, M = 0.

A més dels camps anteriors, IP també haurà de canviar convenientment el camp de total length i el camp de checksum de cada fragment.

**2.2 Adreces IP**

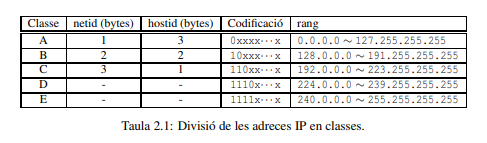
Les adreces IP tenen 32 bits (4 bytes). La figura 2.5 mostra el format d’una adreçà IP.



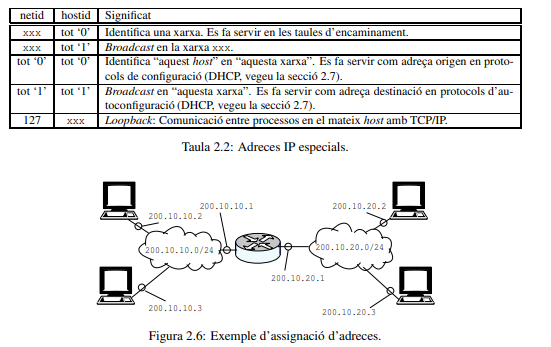
El **netid** identifica la xarxa i el **hostid** identifica un host dintre la xarxa. El límit entre el netid i el hostid és variable. La notació que es fa servir es coneix com a notació amb punts i consisteix a expressar els 4 bytes de l’adreça en decimal separats per punts, per exemple 147.83.34.25. A la taula 2.1 hi ha la divisió en classes de les adreces IP. La classe D és per adreces multicast (per exemple, l’adreça 224.0.0.1 identifica “All Systems on this Subnet” [36]) i la classe E són adreces reservades.

L’assignació de les adreces es fa tenint en compte que:

* Una adreça identifica una interfície (han de ser úniques).
* Totes les adreces d’una mateixa xarxa IP tenen el mateix netid.
* Totes les adreces assignades han de ser diferents.



No totes les adreces es poden fer servir per numerar les interfícies. La taula 2.2 mostra les adreces especials.



La figura 2.6 en mostra un exemple. Cal destacar:

* Totes les interfícies d’una mateixa xarxa tenen el mateix prefix de xarxa (netid).
* No es poden fer servir les adreces especials per numerar les interfícies. Cada xarxa en té dues: la de la xarxa (amb el hostid tot a ‘0’, que és la primera adreça disponible en el rang d’adreces de la xarxa) i la del broadcast en la xarxa (amb el hostid tot a ‘1’, que és l’última adreça disponible en el rang d’adreces de la xarxa).
* El router ha de tenir assignada una adreça en cada interfície. Cada adreça ha tenir el netid de la xarxa on està connectada la interfície. Les adreces han de ser úniques en tot Internet. Per aconseguir-ho, l’organisme d’Internet Internet Assigned Numbers Authority (IANA) assigna blocs d’adreces als Regional Internet Registries (RIR):
  + RIPE: Europa.
  + ARIN: USA.
  + APNIC: ASIA.
  + LACNIC: Amèrica Llatina.

A la vegada, els RIR assignen blocs d’adreces als ISP i aquests als seus abonats. Aquestes adreces s’anomenen públiques, globals o registrades.

2.2.1 Adreces Privades

Per a les xarxes (o hosts) que fan servir TCP/IP i que no tenen una adreça publica, s’han reservat els següents blocs d’adreces (que no són enrutables a Internet):

* 1 adreça de classe A: 10.0.0.0 ∼ 10.255.255.255.
* 16 adreces de classe B: 172.16.0.0 ∼ 172.31.255.255.
* 256 adreces de classe C: 192.168.0.0 ∼ 192.168.255.255.

**2.3 Subnetting**

La divisió en classes és massa rígida i no permet aprofitar bé les adreces. Solució: deixar que el límit entre el netid i hostid sigui variable. Això permet tenir més xarxes de mida diferent. El nombre d’adreces d’una xarxa, però, haurà de continuar sent un múltiple de 2. Per expressar quina és la mida del netid es fa servir una màscara. La màscara és un número de 32 bits amb els bits més significatius que es corresponen amb el netid a ‘1’ i els altres a ‘0’.

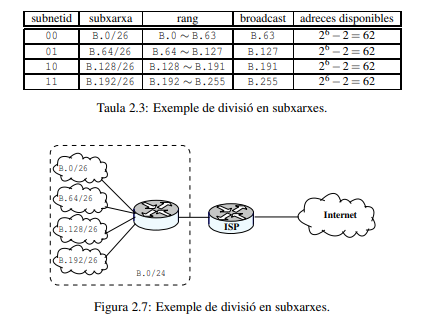
La màscara es representa amb una de les següents notacions: adreça-IP/nombre-de-bits-a-1-de-lamàscara (per exemple 147.83.24.0/24), o amb la notació amb punts (per exemple: 255.255.255.0)

Divisió d’una xarxa en subxarxes. Motivació:

* Eficiència: els routers filtren els broadcasts de L2. Si no fos així, col·lapsarien Internet!
* Seguretat: Els routers aïllen xarxes diferents i només encaminen els datagrames que ho permeten les taules d’encaminament. També permeten afegir llistes i d’altres mecanismes que permeten un control més acurat (això s’explica en la secció 2.11).

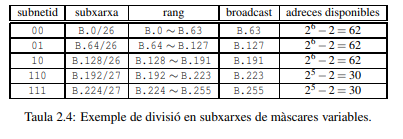
Exemple de divisió en subxarxes: Suposem que una empresa (en l’argot d’Internet es parla d’una SOHO: Small Office Home Office) contracta l’adreça de classe C 200.200.200.0/24 a l’ISP (l’adreça de partida per fer el subnetting l’anomenarem “adreça base”). Suposem que l’empresa vol tenir 4 subxarxes. Per aconseguir-ho, agafem 2 bits del hostid (22 = 4) per fer el subnetting (aquest bits s’anomenen subnetid i han de ser els més significatius del hostid).

La taula 2.3 mostra les subxarxes que resulten. Els pesos dels bits del subnetid són 27 = 128 i 26 = 64. A la taula es fa servir B = 200.200.200. La columna d’adreces disponibles compte el nombre d’adreces que queden al descomptar les adreces especials (la de la xarxa i la de broadcast). Per saber quants hosts es podrien connectar en cada subxarxa també hauríem de descomptar les adreces que consumeixen els routers. La figura 2.7 mostra una possible connexió de les subxarxes.



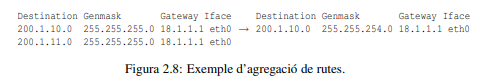
2.3.1 Màscares variables

Si es necessita tenir subxarxes de diferents mides, és possible tenir màscares variables. Per exemple, a la xarxa de la figura 2.7 podríem dividir una de les subxarxes en dues subxarxes més (afegint un altre bit al subnetid) de 25 − 2 = 30 adreces disponibles en cada una. La taula 2.4 mostra les subxarxes que resultarien. Fixeu-vos que perquè la divisió en subxarxes sigui correcta, els rangs d’adreces que pertanyen a cada subxarxa no es poden superposar. Per exemple, no és possible agafar dues subxarxes amb els subnetid 00 (rang B.0 ∼ B.63) i 001 (rang B.32 ∼ B.63), perquè la primera subxarxa tindria adreces que també pertanyen a la segona subxarxa.

****

**2.3.2 Classless Inter-Domain Routing (CIDR)**

Inicialment els routers d’Internet feien l’encaminament per classes. Finalment, però, per evitar el creixement desmesurat de les taules d’encaminament, es varen eliminar les classes de les taules d’encaminament. Actualment en les taules d’encaminament les destinacions sempre es donen en forma d’una adreça i una màscara que dona la mida del netid. El RFC 1519 [33] proposa l’eliminació de l’encaminament per classes i la distribució de blocs d’adreces als ISP de forma “racional” (es a dir, tenint en compte la seva situació geogràfica). La proposta es coneix pel nom de CIDR Classless Inter-Domain Routing. L’objectiu és poder agregar les adreces amb prefix comú. Per exemple, si l’encaminament és el mateix per a les xarxes 200.1.10.0/24 i 200.1.11.0/24, aleshores es poden resumir les dues xarxes en una sola entrada: 200.1.10.0/23 (agregació, vegeu la figura 2.8). D’aquesta manera s’aconsegueix un adreçament jeràrquic. Per exemple, suposeu que s’assigna un bloc d’adreces amb un prefix comú a Europa. D’aquestes, s’assigna un bloc amb un netid major a cada país d’Europa. Això es repeteix per assignar blocs als ISP de cada país, i a la vegada, per assignar blocs als abonats dels ISPs. D’aquesta manera, un ISP als EUA, per exemple, podria resumir totes les xarxes que hi ha a Europa en una sola entrada: la que apuntés cap a l’ISP que encaminés els datagrames cap a Europa. A mida que es va encaminant el datagrama cap a la destinació final, les entrades de les taules d’encaminament es van fent més específiques (amb un netid major).

****

**2.4 Taules d’encaminament i algorisme de lliurament de datagrames**

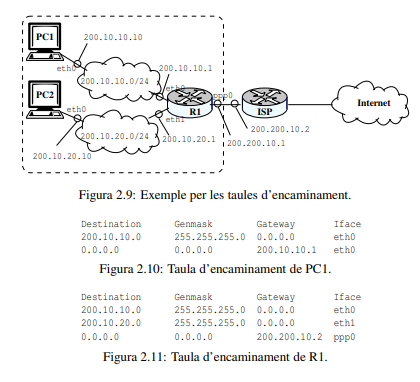
Per “algorisme de lliurament de datagrames” ens referirem a les accions que fa IP quan processa un datagrama pel seu encaminament. L’encaminament pot ser:

* Directe: Quan hi ha una interfície en la mateixa xarxa que la de l’adreça destinació del datagrama.
* Indirecte: En cas contrari. En aquest cas el datagrama s’ha de lliurar a un router que l’encamini cap a la destinació.

Mirem ara com és la taula d’encaminament d’IP. Cal tenir en compte que el nivell IP és el mateix tant si es tracta d’un host com d’un router. Per tant, la taula d’encaminament s’interpreta de la mateixa manera. En aquesta taula hi ha les destinacions a què sap arribar el nivell IP. Si el nivell IP rep un datagrama cap a una destinació que no és a la taula, el descarta. Així doncs, a la taula hi ha tuples que especifiquen {destinacions, com arribar-hi}. Evidentment, si totes les taules d’encaminament haguessin de tenir totes les possibles xarxes destinació d’Internet, les taules serien inviables per la seva complexitat i nombre d’entrades. Aquest problema es resol amb “**l’entrada per defecte**” (default route).

Considerem l’exemple de la figura 2.9. La figura 2.10 mostra com seria la taula d’encaminament de PC1.

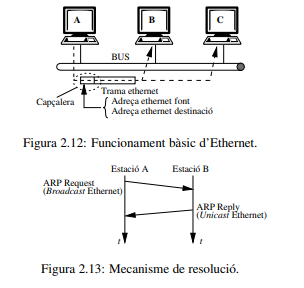
Les dues primeres columnes de la taula d’encaminament de la figura 2.10 (Destination i Genmask) identifiquen les destinacions on sap arribar PC1. Les columnes identifiquen respectivament l’adreça de xarxa i la seva màscara. Les dues altres columnes Gateway i Iface indiquen com arribar a la destinació. La columna de Gateway té l’adreça del router que s’ha de fer servir per encaminar el datagrama cap a la destinació. Aquest valor val 0.0.0.0 si el lliurament és directe (no s’ha de fer servir cap router). La columna Iface identifica la interfície per on ha d’enviar-se el datagrama. En les màquines UNIX es fa servir una etiqueta. En Linux, per identificar els ports Ethernet les etiquetes són eth0, eth1, etc. Aquestes etiquetes tenen la mateixa funció que els noms de dispositius. Per exemple: /dev/hda1 identifica la primera partició del primer disc dur IDE d’un PC Linux. A diferència dels dispositius, les interfícies de comunicacions no són nodes del directori /dev. Les interfícies de comunicacions es creen quan es carrega el driver que les controla i podem llistar-les amb la comanda ifconfig -a. La primera entrada és la xarxa on està connectat. Si aquesta entrada no hi fos, PC1 no podria enviar datagrames a altres estacions de la seva mateixa xarxa, encara que tingués assignada una adreça a la interfície que té en aquesta xarxa. Podem identificar que el lliurament és directe perquè la columna de Gateway té el valor 0.0.0.0.

****

La segona entrada de la taula d’encaminament de la figura 2.10 és l’entrada per defecte. Aquesta entrada sempre té la xarxa destinació 0.0.0.0 i màscara 0.0.0.0. El motiu és el següent: per consultar la taula d’encaminament el nivell IP ordena les entrades de més a menys específiques, és a dir, de més a menys bits de la màscara a ‘1’. El nivell IP segueix aquest ordre per buscar la xarxa destinació (Longest Prefix Match). Per això fa l’operació AND bit a bit entre l’adreça destinació que hi ha en el datagrama i la màscara de l’entrada de la taula d’encaminament (d’aquí el nom de màscara). El resultat el compara amb la xarxa destinació (columna Destination). Si hi ha coincidència, es para la cerca i l’entrada es fa servir per encaminar el datagrama. D’aquesta manera l’entrada per defecte serà sempre l’última en mirar i el resultat de la comparació serà sempre cert. L’entrada per defecte sempre és indirecta (en l’exemple, el gateway 200.10.10.1). La figura 2.11 mostra la taula d’encaminament del router R1. En aquest cas R1 resumeix la resta d’Internet amb l’entrada per defecte, però ha d’especificar en la taula totes les altres xarxes que “pengen” dels altres ports (en l’exemple són dues: 200.10.10.0/24 i 200.10.20.0/24. Si alguna de les xarxes que pengen dels ports que no són el que té l’entrada per defecte no es trobessin en la taula, R1 no sabria arribar-hi (els enviaria per l’entrada per defecte cap a Internet!).

**2.5 Address Resolution Protocol (ARP)**

Motivació: adreces de xarxa versus adreces “físiques”. En una LAN es fan servir adreces per identificar l’estació transmissora i receptora. Per exemple, a Ethernet es fan servir adreces de 6 bytes (48 bits). El funcionament d’Ethernet s’explica en detall en el tema 4. La figura 2.12 mostra el funcionament bàsic. Ethernet fa servir un “medi compartit” que funciona com un BUS: quan una estació envia una trama, aquesta arriba a totes les estacions connectades a la xarxa Ethernet. En cada estació la tarja Ethernet rep la trama que s’ha enviat al medi. Totes les targes miren l’adreça Ethernet destinació que hi ha en la trama. Si l’adreça destinació és la de la tarja, aleshores aquesta interromp la CPU i transfereix el contingut de la trama rebuda per DMA a la memòria del computador. En cas contrari, la tarja descarta la trama rebuda. Quan el computador vol enviar una trama Ethernet, el driver que controla la tarja ha de passar-li la informació que vol enviar (per exemple, el datagrama IP) i l’adreça Ethernet de la tarja del computador on es vol enviar.

****

Així doncs, després de mirar la taula d’encaminament, tal com s’ha explicat en la secció anterior, el nivell IP pot necessitar fer una conversió de l’adreça IP a l’adreça física. El terme “adreça física” es refereix a l’adreça de la xarxa física per on s’enviarà el datagrama, per exemple Ethernet. Aquesta conversió no sempre és necessària. Per exemple, en un enllaç punt a punt no es necessita una adreça física, perquè hi ha un únic destinatari possible: l’altre extrem de l’enllaç¸. El mecanisme que s’encarrega de fer la conversió adreça IP – adreça física s’anomena Address Resolution Protocol, ARP.

El procediment que dispara el mecanisme ARP és el següent:

* IP determina l’adreça IP on ha d’enviar-se el datagrama (que pot ser l’adreça destinació del datagrama si el lliurament és directe o l’adreça d’un router si el lliurament és indirecte).
* Si la interfície per on ha d’enviar-se el datagrama no necessita adreça física (per exemple, un enllaç punt a punt), aleshores IP passa el datagrama al driver per a la seva transmissió. Altrament:
  + IP sol·licita al mòdul ARP l’adreça “física” que correspon a l’adreça IP.
  + ARP té una taula on guarda les tuples {adreça IP; adreça física}. Si l’adreça IP sol·licitada no es troba en la taula, aleshores el mòdul ARP inicia un “procediment de resolució”. Quan acaba aquest procediment, ARP retorna al mòdul IP l’adreça física sol·licitada.
  + Mentre IP espera la resposta del mòdul ARP, guarda temporalment el datagrama en un buffer. Aquest buffer té un time-out. Si salta el time-out i ARP encara no ha resolt l’adreça, es descarta el datagrama.
  + Quan IP rep la resposta del mòdul ARP, aleshores passa el datagrama al driver juntament amb l’adreça física on s’ha d’enviar.

2.5.1 Funcionament del mecanisme de resolució

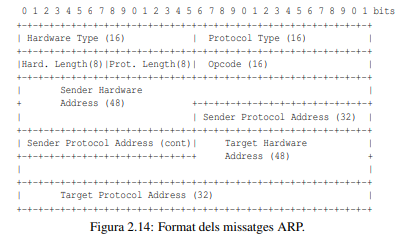
Suposem que el host A de la figura 2.12 vol resoldre l’adreça del host B. Quan es fa el boot dels PCs les caches ARP estan buides. La figura 2.13 mostra les trames ARP Request i ARP Reply que s’envien durant el mecanisme de resolució. Els detalls són els següents:

* L’estació A envia un missatge ARP Request amb una adreça Ethernet destinació broadcast. Aquesta trama porta l’adreça IP que es vol resoldre (la de l’estació B). Com que l’adreça destinació és broadcast, totes les targes interrompran la CPU quan rebin la trama. Després el driver passarà el contingut de la trama al mòdul ARP.
* El mòdul ARP de l’estació B veurà que es vol resoldre la seva adreça i enviarà una trama ARP Reply. L’adreça destinació de l’ARP Reply és unicast (és l’adreça Ethernet de l’estació A). Els mòduls de les altres estacions descartaran l’ARP Request.
* Les dues estacions implicades (A i B) actualitzaran la cache ARP amb les adreces {IPB; ethB} i {IPA; ethA} respectivament. El motiu que només les estacions implicades actualitzin la cache ARP és que convé mantenir les cache petites perquè les consultes siguin el més ràpid possible (la cache ARP s’ha de consultar cada vegada que s’envia un datagrama!).

Les entrades de la cache ARP tenen associades un time-out. Cada vegada que una entrada es fa servir, el time-out es refresca. Si salta el time-out, l’entrada s’esborra de la cache. El motiu és, també, mantenir les cache només amb les entrades que es fan servir perquè siguin petites.

2.5.2 Format dels missatges ARP

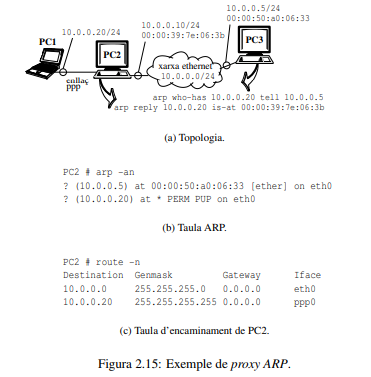
La figura 2.14 mostra el format dels missatges ARP pel cas particular de resoldre adreces IPv4 en xarxes Ethernet.

****

* Hardware Type: indica el tipus d’adreça física (Ethernet).
* Protocol Type: Indica el protocol de l’adreça que es vol resoldre (IPv4).
* Hardware Length: Indica el nombre de bytes de l’adreça física.
* Protocol Length: Indica el nombre de bytes de l’adreça que es vol resoldre (IPv4).
* Opcode: Tipus de missatge (Request/Reply).
* Sender Hardware Address/Sender Protocol Address: adreça física i del protocol que es vol resoldre, del que envia. En el Reply hi ha l’adreça física buscada.
* Target Hardware Address/Target Protocol Address: adreça física i del protocol que es vol resoldre, de l’estació on s’envia. En el Request el camp Target Hardware Address és irrellevant (és l’adreça buscada).

2.5.3 Proxy ARP

Idea : una estació contesta un ARP request en nom d’una altra. La figura 2.15(a) mostra un possible escenari. El host PC2 fa de router al host PC1 per accedir a la xarxa 10.0.0.0/24. Les figures 2.15(b) i 2.15(c) mostren respectivament la taula ARP i la taula d’encaminament que podria tenir PC2 (el bolcat és el que s’obtindria en una màquina Linux). La segona entrada de la taula ARP (figura 2.15(b)), és la que s’afegiria manualment perquè PC2 fes proxy arp del host PC1 (adreça 10.0.0.20). Un possible motiu per connectar el host PC1 a la seva xarxa a través de PC2 podria ser que no hi hagués una roseta disponible per connectar PC1. Un altre motiu podria ser que la xarxa fos, per exemple, Ethernet, i que PC1 no tingués una interfície Ethernet, de manera que la connexió entre PC1 i PC2 podria ser, per exemple, a través del port sèrie amb un cable mòdem null i un enllaç ppp. En l’exemple de la figura 2.15, si PC3 vol enviar un datagrama a PC1, enviarà un ARP request perquè és a la mateixa xarxa. L’ARP request el contestarà PC2, de manera que PC3 enviarà els datagrames destinats a PC1 al host PC2. PC2 ha d’estar configurat com a router i tenir una entrada en la taula d’encaminament que enviï els datagrames de PC1 a través de l’enllaç PC1-PC2, com mostra la figura 2.15(c).

****

2.5.4 Gratuitous ARP

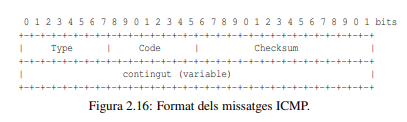
El gratuitous ARP consisteix en un ARP request que envia un host per resoldre la seva mateixa adreça després de la fase de boot. Els motius són els següents:

* Per detectar adreces IP duplicades en la mateixa xarxa. Per exemple, si un altre host tingués la mateixa adreça que el que envia el gratuitous ARP contestaria l’ARP request. Això permetria detectar al host que envia el gratuitous ARP que la seva adreça IP ja l’està fent servir un altre host.
* Per actualitzar les caches ARP. Per exemple, si una mateixa adreça IP es canvia d’un host A a un altre diferent B (per exemple, perquè el host A es dona de baixa i s’assigna la seva adreça al host B), les estacions que tinguessin l’entrada ARP del host A continuarien enviant paquets a l’adreça Ethernet del host A després del canvi. El gratuitous ARP que envia el host B el rebran totes les estacions de la xarxa, i aquelles que tinguessin l’adreça IP en la cache actualitzarien l’adreça física a la del host B.

**2.6 Internet Control Message Protocol (ICMP)**

Senyalitza missatges d’error o atenció. Característiques:

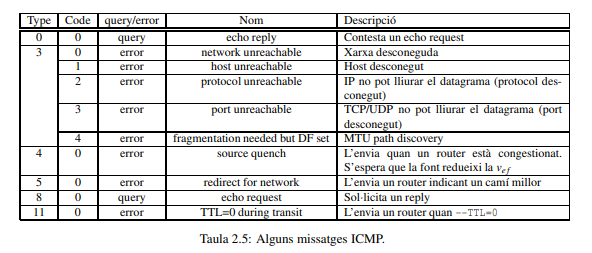
* Va sobre IP (no TCP/UDP).
* Els pot activar IP/TCP/UDP o un procés d’usuari.
* Poden ser (i) query o (ii) error.
* Els missatges ICMP d’error no poden generar un altre ICMP d’error (per evitar bucles).



La figura 2.16 mostra el format d’un missatge ICMP:

* Type/Code: identifiquen el missatge (vegeu la taula 2.5).
* El checksum és de tot el missatge ICMP.
* El contingut depèn del Type/Code. A més,
* Els missatges de tipus query porten un camp anomenat identifier que serveix per poder fer correspondre els request amb els reply (el reply sempre porta el mateix valor que el request que l’ha generat).
* Els missatges d’error retornen sempre la capçalera IP del datagrama causant del missatge i els 8 primers bytes del seu payload. Si el datagrama causant de l’error porta encapsulat un segment TCP o un datagrama UDP, els ports origen i destinació queden inclosos en aquests primers 8 bytes. Aquesta informació és necessària en alguns casos. Per exemple, quan es fa servir MTU Path Discovery (vegeu la secció 2.1.2), TCP necessita conèixer els ports origen i destinació per identificar la connexió que ha de reduir la mida dels segments.

Exemple: traceroute. Aquesta comanda envia successivament paquets amb TTL=1, 2, ... i un port arbitrari. Identifica els routers intermedis pels ICMP d’error TTL==0 i la destinació per l’ICMP port unreachable.



**2.7 Dynamic Host Configuration Protocol (DHCP)**

L’assignació d’adreces IP pot ser:

* Manual, per exemple amb un script que executa les comandes de configuració de la xarxa (ifconfig eth0 ...).
* Automàtica, a través del protocol IPCP en un enllaç ppp o amb el protocol DHCP en una LAN (abans BOOTP).

DHCP és un protocol que segueix el model client-servidor amb l’objectiu de subministrar als clients:

* Paràmetres per a la configuració de la xarxa.
* Assignació temporal o permanent d’adreces IP.

L’assignació d’adreces IP pot ser:

* Dinàmica: en el servidor s’especifica un conjunt (pool) d’adreces IP que es poden assignar als clients i un temps de “lloguer” (leasing).
* Automàtica: el temps de lloguer és il·limitat.
* Manual: En el servidor s’especifica l’adreça de cada client (p.e., associant l’adreça IP per a cada adreça Ethernet).

Altres paràmetres:

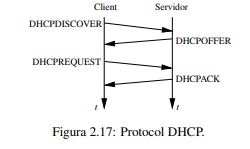
* Domini (p.e. ac.upc.es) i hostname del client.
* Adreça IP del Servidor DNS (DNS s’explica en la secció 2.9).
* Router per defecte.
* Màscara.
* Etc.

Detalls del protocol: Fa servir UDP. El port del servidor és 67 i el del client és 68. Fixeu-vos que el client té un port well known i no un port efímer. Això no és un problema perquè en un mateix host només hi pot haver un client DHCP. En la mateixa xarxa pot haver-hi més d’un servidor DHCP. La figura 2.17 descriu el protocol:

* El client envia un missatge DHCPDISCOVER amb adreça IP font 0.0.0.0 i destinació broadcast (255.255.255.255), port UDP font 68 i destinació 67. Aquest missatge pot suggerir les opcions que desitgi el client, com ara l’adreça IP i el temps de leasing.
* Cada servidor respon amb un missatge DHCPOFFER amb l’oferta dels paràmetres de configuració. Un dels paràmetres d’aquest missatge és un identificador del servidor.
* El client respon amb un missatge DHCPREQUEST broadcast (perquè ho rebin tots els servidors) amb l’identificador del servidor que hi havia en el missatge DHCPOFFER que ha escollit.

El servidor escollit confirma la configuració enviant un missatge DHCPACK.

Els missatges DHCPOFFER i DHCPACK poden tenir una adreça de nivell físic i IP unicast si el client pot rebre paquets unicast abans de tenir la xarxa configurada, o broadcast en cas contrari. En el missatge DHCPDISCOVER hi ha un flag de broadcast que permet al client triar el tipus d’adreça. Si el client recorda l’adreça IP assignada en una sessió anterior, pot enviar directament un missatge DHCPREQUEST i estalvia els dos primers missatges del diagrama de la figura 2.17. El client també pot enviar un DHCPREQUEST per prolongar el temps de leasing.



**2.8 Network Address Translation (NAT)**

Motivació: Aprofitar millor les adreces públiques.

Funcionament: La figura 2.18 mostra el funcionament de NAT. En l’interior de la xarxa privada els hosts tenen adreces privades (per exemple, de la xarxa 10.0.0.0). Amb aquestes adreces les estacions no poden accedir a Internet (el servidor on intentessin accedir no podria contestar a l’adreça privada). Perquè això sigui possible, el router NAT fa el següent:

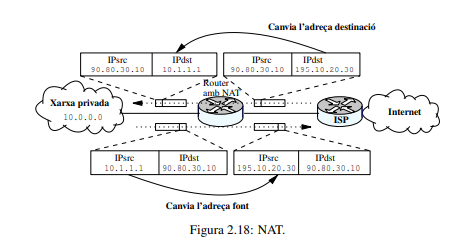
* Als datagrames que surten de la xarxa privada els canvia l’adreça font privada per una adreça pública (una de les adreces contractades a l’ISP).
* Als datagrames que entren a la xarxa privada en resposta als datagrames anteriors els desfà el canvi. És a dir, es canvia l’adreça destinació pública per l’adreça destinació privada del host.

Els canvis que fa el router NAT són transparents als hosts. És a dir, els hosts no s’adonen del canvi, ni tan sols saben quina adreça pública se’ls hi està posant quan surten a Internet. Per poder desfer el canvi el router manté una “Taula NAT” amb les adreces privades i les corresponents adreces públiques. Avantatges de NAT:

* Estalvia adreces públiques perquè no hem d’assignar una adreça pública a cada un dels hosts de la xarxa privada (serà, per tant, més econòmic perquè hem de contractar menys adreces a l’ISP!).
* No hem d’esperar a tenir un ISP per assignar adreces a la xarxa privada.
* No han de reassignar-se les adreces privades si es canvia d’ISP.
* Afegeix seguretat: L’entrada/sortida de la xarxa està controlada pel router NAT (per exemple, els hosts als quals el router no canviï l’adreça no podran accedir a Internet).

Tipus de translacions:

* NAT estàtic: s’associa una adreça pública per a cada adreça privada. Només poden accedir a Internet les adreces privades que tinguin una adreça pública associada. Necessitem tantes adreces públiques com hosts volem que puguin accedir a Internet.
* NAT dinàmic: hi ha un “conjunt” (pool) d’adreces públiques que s’assignen dinàmicament: quan un host inicia una connexió amb Internet, s’afegeix una entrada en la taula NAT amb la seva adreça privada i l’adreça pública que se l’hi ha assignat. L’entrada té un temporitzador, de manera que quan el host deixa de fer servir l’entrada durant un temps, s’esborra i un altre host pot reutilitzar la mateixa adreça pública. Necessitem tantes adreces públiques com hosts volem que puguin accedir simultàniament a Internet.



2.8.1 PAT

El mecanisme NAPT (o PAT): (Network Address and Port Translation) es fa servir quan desitgem que més d’un host faci servir la mateixa adreça pública simultàniament per accedir a Internet. El funcionament bàsic és el següent: quan un host inicia una connexió amb Internet, en la taula NAT es posa la tupla: (adreça privada, port local), (adreça pública, port extern). Com que una mateixa adreça pública s’assigna a diferents adreces privades, es fa servir el port per poder desfer el canvi. El port que es fa servir és el port font dels datagrames de sortida, que és el de destinació en els datagrames d’entrada. Quan un datagrama surt es busca en la taula NAT fent servir les entrades (adreça privada, port local). Quan un datagrama entra es busca en la taula fent servir les entrades (adreça pública, port extern). Com que les translacions han de ser unívoques, totes les tuples (adreça privada, port local) han de ser diferents, i han de tenir una tupla (adreça pública, port extern) també diferent. Aquest motiu fa que el port també necessiti ser canviat en alguns casos: Per exemple, quan dos hosts diferents accedeixen a Internet amb el mateix port font fent servir la mateixa adreça pública. Per aquest motiu cal guardar el port local i el port extern. El mecanisme explicat anteriorment és vàlid per a TCP/UDP. Què passa per altres protocols? Per exemple, en el cas d’ICMP els paquets de query tenen el camp identifier, que porta un número arbitrari que serveix per relacionar request/reply. Aquest camp es fa servir en la taula NAT de forma anàloga al port. En el cas de missatges ICMP d’error, el router NAT busca el port en els 8 bytes del payload del datagrama que causa l’error que s’afegeixen en el missatge ICMP (vegeu la secció 2.6). Això permet, per exemple, que els hosts en la xarxa interna facin MTU Path Discovery.

2.8.2 DNAT

El mecanisme NAT/PAT explicat anteriorment permet sortir de la xarxa privada. Què passa si volem que es pugui entrar (volem tenir algun servidor accessible des d’Internet)? Per aconseguir-ho necessitem fer el mateix que abans: canviar l’adreça destinació dels datagrames que entren, i l’adreça font dels que surten. Els efectes, però, són molt diferents: Permetre que s’iniciïn connexions de la xarxa privada cap a l’exterior, o viceversa.

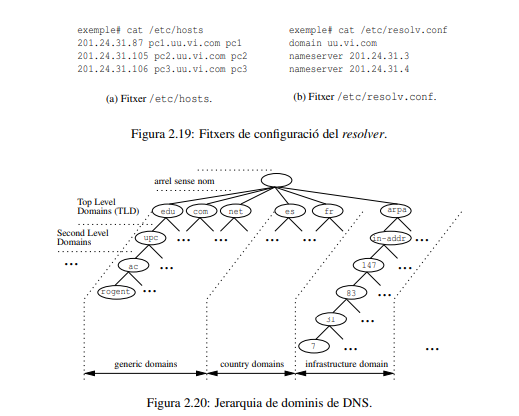
Per aquest motiu, per referir-se al segon cas es sol parlar de DNAT Destination Network Address Translation. En el cas de DNAT la iniciativa és d’un client extern, i primer es canviarà l’adreça destinació, perquè els datagrames que envia el client puguin entrar en la xarxa interna (d’aquí el nom de DNAT). Quan els datagrames del client extern arriben al router, la taula NAT ja ha d’estar configurada amb l’adreça dels servidors de la xarxa interna als que es pot accedir. Per això, en cas de fer servir DNAT la configuració haurà de ser estàtica. Exemple típic: accés a un servidor de web intern (port web: 80). En la taula DNAT s’afegiria l’entrada estàtica (adreça privada, 80), (adreça pública, 80), on adreça privada seria l’adreça IP del servidor de web en la xarxa privada i adreça pública seria l’adreça IP del servidor vista des d’Internet. Quan un datagrama arriba des d’Internet al router DNAT amb l’adreça pública i port destinació 80, el router canvia l’adreça destinació per l’adreça privada (el port no es canvia). Quan un datagrama surt amb l’adreça privada del servidor i el port font 80, es canvia per l’adreça pública del servidor.

**2.9 Domain Name System (DNS)**

Objectiu: Permetre que els usuaris d’Internet puguin fer servir noms en comptes d’adreces IP. Per exemple, que es pugui executar Telnet rogent en comptes de Telnet 147.83.34.35. Les característiques bàsiques de DNS són:

* Segueix el paradigma client/servidor amb nivell de transport TCP/UDP amb port well-known 53.
* Hi ha una base de dades amb els noms i les adreces per poder fer la resolució.
* El sistema de noms està organitzat en una jerarquia que permet distribuir la base de dades arreu d’Internet, en comptes d’haver de mantenir-la centralitzada en un únic servidor.

Exemple de funcionament en una màquina UNIX: Les aplicacions que fan servir el sistema DNS es linken amb les funcions de la llibreria de resolució de noms (resolver). Aquestes funcions són gethostbyname(), que retorna l’adreça IP d’un nom i gethostbyaddr(), que fa la resolució inversa: Retorna el nom que correspon a una l’adreça IP. Quan es crida el resolver, primer mira el fitxer /etc/hosts on hi ha noms i adreces IP (vegeu la figura 2.19(a)). Si no pot resoldre el nom amb la informació d’aquest fitxer intenta contactar amb un servidor de noms. En el fitxer /etc/resolv.conf (vegeu la figura 2.19(b)) hi ha el domini local i l’adreça IP del servidor primari i secundari del domini.



La figura 2.20 mostra la jerarquia de noms de DNS [34]. Aquesta jerarquia està organitzada en “dominis” (domains) o “zones”. El domini arrel no té nom i d’ell pengen els top level domains (TLD). Cada un dels TLD té un administrador (registrar en l’argot DNS) que delega part del domini en “sub-dominis”. Per exemple, del TLD edu pengen universitats. D’aquí penja el sub-domini upc, l’administració del qual està delegada a la Universitat Politècnica de Catalunya. La informació de contacte dels registrats dels TLD (per exemple edu).

El nom s’escriu començant pel host i separant els dominis per punts fins el TLD. El nom es pot fer acabar en un punt per indicar que s’especifiquen tots el dominis (fully qualified domain name). Per exemple: rogent.ac.upc.edu. Cada administrador d’un domini (o subdomini) ha de mantenir part de la base de dades de DNS en un “servidor primari” i un o més “servidors secundaris” (de backup). En l’argot DNS, aquests servidors també es coneixen com a “autoritat” (authority) del domini. En aquests servidors hi ha d’haver el nom i adreça dels hosts que pengen del seu domini i el nom i adreça dels servidors primaris i secundaris de les autoritats dels subdominis que hagi delegat. Actualment hi ha 13 servidors que tenen les adreces dels TLD. Aquest servidors s’anomenen root-servers, estan distribuïts arreu del món i tenen per nom a.root-server.net, ..., m.root-server.net.

**2.10 Algorismes d’encaminament**

Tenen per objectiu afegir entrades a les taules d’encaminament. Poden ser:

1. Estàtics: són aquelles entrades que s’afegeixen manualment o mitjançant scripts o protocols de configuració (com DHCP). Una vegada establertes no canvien el seu valor (d’aquí el nom “d’entrades estàtiques”).
2. Adaptatius: en aquest cas hi ha un “protocol d’encaminament” (routing protocol) encarregat de calcular i afegir entrades a les taules. El protocol s’estableix entre els routers, els quals s’intercanvien informació relativa a la topologia de la xarxa perquè s’afegeixin entrades a les taules d’encaminament de forma automàtica. Si la topologia de la xarxa canvia (s’afegeixen o es donen de baixa noves xarxes) els protocols d’encaminament actualitzen automàticament les taules.

Hi ha diferents tipus de protocols d’encaminament adaptatius. Per a classificar-los és convenient introduir el concepte de “sistema autònom” (Autonomous System, AS), que s’explica tot seguit:

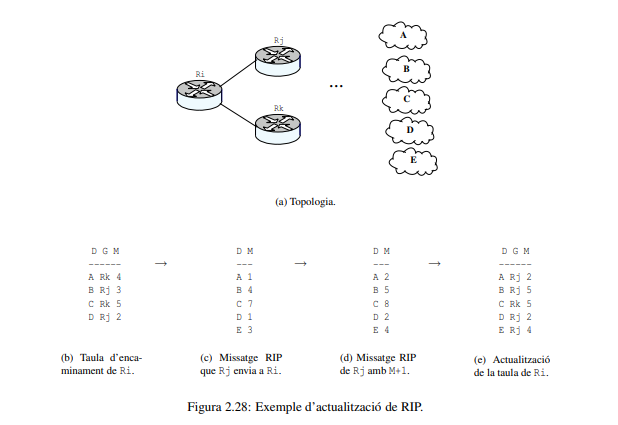
1. Interior Gateway Protocol (IGP): El protocol d’encaminament s’estableix entre els routers d’un mateix AS.
2. Exterior Gateway Protocol (EGP): El protocol d’encaminament s’estableix entre els routers de diferents ASs.

Els protocols IGP podríem dir que tenen com a objectiu el “millor encaminament possible”. Per això fan servir una “mètrica” per avaluar el cost d’una ruta, i a l’hora de calcular les entrades de les taules ho fan de manera que les mètriques siguin les mínimes. És a dir, que les entrades que permeten arribar a les diferents destinacions siguin les òptimes. D’entre els protocols IGP que més es fan servir en la pràctica cal destacar: Routing Information Protocol (**RIP**), que estudiarem en la pròxima secció; Open Shortest Path First (**OSPF**) o el protocol propietari de CISCO: Interior Gateway Routing Protocol (**IGRP**). En el cas dels EGP, hi ha altres condicions de tipus comercial que es tenen en compte a l’hora de calcular les rutes. Per exemple, segons les relacions comercials que hi ha entre dos ISPs, pot passar que un d’ells no desitgi encaminar tràfic de l’altre ISP cap a una certa destinació. Per això, els protocols EGP fan servir “polítiques d’encaminament” (routing policies) per intercanviar-se informació i decidir les entrades que s’afegeixen en les taules d’encaminament. D’aquesta manera, es defineix un AS com a la interconnexió d’un conjunt de xarxes amb una mateixa “política d’encaminament”. El protocol EGP que s’ha convertit en l’estàndard “de facto” a Internet és Border Gateway Protocol (**BGP**).

2.10.1 Routing Information Protocol (RIP)

Les característiques d’aquest protocol són:

* La mètrica és el nombre de salts fins a la destinació: 1 si la destinació és una xarxa directament connectada, 2 si s’ha que passar per un router, etc.
* Els routers envien periòdicament (cada 30 segons) un missatge RIP amb les destinacions i mètriques conegudes. Aquest missatge s’envia amb UDP, amb port font i destinació 520. El missatge s’envia amb adreça destinació broadcast per cada interfície on es vulgui fer servir RIP.
* Si es deixen de rebre missatges RIP d’un router veí (durant 180 segons), aleshores s’assumeix que el router veí ha “caigut” i les entrades que el fan servir com a gateway es posen amb “mètrica infinit” (per marcar les destinacions com inaccessibles).
* La mètrica infinit val 16.
* RIP versió 2 afegeix les següent modificacions: S’afegeix la màscara a les destinacions (en RIP versió 1 no s’envien les màscares i, per tant, no poden ser variables). En RIP versió 2 també es pot fer servir opcionalment l’adreça destinació multicast 224.0.0.9, en comptes de l’adreça de broadcast.

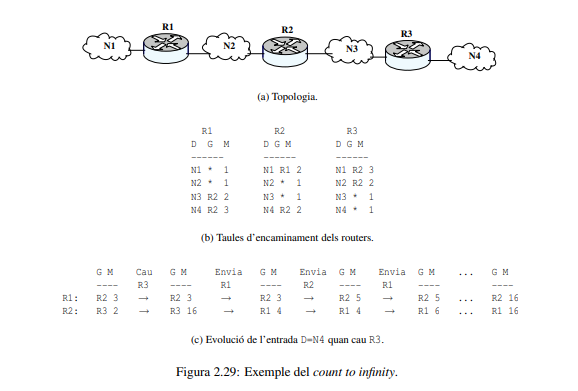


Les taules d’encaminament s’actualitzen com mostra el següent exemple. L’algorisme s’anomena Bellman-Ford (o distance vector algorithm). Suposem la topologia de la figura 2.28(a). Suposem que el Router Ri té la taula d’encaminament de la figura 2.28(b). Per simplicitat, farem servir noms en comptes d’adreces IP i màscares. En les taules, D es refereix a la destinació, G al gateway i M a la mètrica. Suposem ara que el router Rj envia el missatge de la figura 2.28(c). Quan Ri rep el missatge de Rj primer incrementa les mètriques amb 1, tal com mostra la figura 2.28(d). Aquesta mètrica incrementada és la que costaria a Ri arribar a les destinacions a través de Rj. Després compara les destinacions amb les que té a la taula:

* Si la mètrica és menor, com passa per a la destinació A, canvia l’entrada fent servir Rj.
* Si feia servir el mateix gateway, posa sempre la nova mètrica, encara que sigui major, com passa per a la destinació B. Això voldria dir que hi ha hagut un canvi de topologia i que fent servir el mateix gateway passa a costar més arribar a la destinació.
* Si la mètrica és major que fent servir un altre gateway (com passa per a la destinació C), aleshores no es canvia.
* Si hi ha una nova destinació (com passa amb la destinació E), s’afegeix a la taula.

*Count to infinity*

El principal problema de RIP és el temps de convergència: és a dir, el temps que passa des que hi ha un canvi en la topologia de la xarxa fins que les taules d’encaminament s’estabilitzen. Aquest temps de convergència en alguns casos pot ser especialment gran, com mostra el següent exemple. Suposem la xarxa de la figura 2.29(a). Quan s’ha assolit la convergència, els routers tindran les taules que mostra la figura 2.29(b). Suposem ara que el router R3 es queda penjat, que R2 ho detecta, però que al mateix temps R1 envia el seu missatge RIP. La figura 2.29(c) mostra com evolucionaria l’entrada amb destinació D=N4 en les taules dels routers R1 i R2. Quan R2 detecta que R3 deixa d’enviar missatges RIP, posa la destinació D=N4 amb mètrica infinit (M=16). Si en aquest moment R1 envia la taula, aleshores R2 es pensa que pot arribar a la destinació D=N4 a través de R1 i posa en la seva taula que pot arribar a D=N4 a través de R1 amb mètrica M=3. A mesura que R1 i R2 es van enviant missatges RIP, l’entrada convergeix fins a la mètrica infinit.



*Split horizon*

El problema de l’exemple anterior s’ha produït perquè R2 ha pensat que podia arribar a la destinació D=N4 a través de R1, quan en realitat R1 necessita R2 per arribar a D=N4. Per resoldre aquest problema RIP modifica els missatges de forma que quan fa un broadcast en una interfície, elimina les entrades que tinguin un gateway en la mateixa interfície. Aquesta tècnica es coneix com Split horizon. En l’exemple anterior, si s’hagués fet servir Split horizon R1 no hauria enviat l’entrada D=N4 al fer el broadcast del missatge RIP en la interfície on hi ha R2. D’aquesta manera, R2 no hauria pensat que pot arribar a D=N4 a través de R1. A continuació, quan R2 hagués enviat el missatge RIP cap a d’interfície on hi ha R1, R2 sí que posaria l’entrada {D=N4, M=16} i R1 actualitzaria correctament la seva taula. L’Split horizon no resol completament el problema del count to infinity de RIP. Es poden trobar exemples de topologies més complexes que la de l’exemple anterior, on continuaria apareixent el count to infinity tot i fer servir el Split horizon. Una modificació que en alguns casos pot reduir encara més aquest problema consisteix a enviar les entrades que tinguin un gateway en la mateixa interfície on s’envia el missatge RIP posant mètrica infinit. En aquest cas es parla de **Split horizon with poisoned reverse**. Finalment, per accelerar encara més la convergència, quan canvia la mètrica d’una entrada, RIP envia immediatament el missatge RIP sense esperar el període de 30 segons. Aquesta tècnica es coneix com a **triggered updates**.

2.10.2 Open Shortest Path First (OSPF)

Per a xarxes petites se sol fer servir RIP per la seva senzillesa. Per a xarxes grans i amb canvis de topologia freqüents, el problema de convergència de RIP pot ser un inconvenient. En aquest cas sol fer-se servir OSPF, que té una convergència molt més ràpida que RIP. El funcionament bàsic d’OSPF és el següent:

* Els routers mantenen una base de dades amb l’estat de tota la xarxa (link state data-base).
* Cada router monitoritza les xarxes directament connectades i dels seus veïns. Aquesta informació s’envia a la resta de routers de la xarxa (Link State Advertisements, LSA). A diferència de RIP no envia totes les destinacions, sinó només les directament connectades.
* Els LSA necessiten un encaminament especial perquè arribin a tots els altres routers de la xarxa. Aquest encaminament s’anomena **flooding** i consisteix a enviar els LSA per totes les interfícies, excepte la d’on ha arribat. Com adreça destinació es fa servir l’adreça multicast All OSPF Routers (224.0.0.5).
* Els missatges OSPF van encapsulats directament sobre IP (no fan servir TCP/UDP), amb camp protocol d’IP igual a 89.
* A diferència de RIP, els missatges LSA no s’envien periòdicament, altrament només quan es produeix algun canvi o si es rep un missatge Link State Request.
* A més dels missatges LSA, els routers executen un protocol de hello per descobrir i mantenir la relació dels routers veïns. El protocol de hello envia missatges multicast All OSPF Routers periòdicament per totes les interfícies. Si un router “cau”, els veïns ho detecten per l’absència dels missatges de hello.
* Tots els routers guarden la informació rebuda en el link state data-base i fan servir l’algorisme Shortest Path First (anomenat també algorisme de Dijkstra) per calcular les taules d’encaminament òptimes.
* La mètrica es calcula en funció de les velocitats de transmissió, retards, etc. Té, per tant, un significat molt més acurat que en RIP.
* OSPF no té el problema de count to infinity de RIP i, per tant, no s’ha de fer servir una mètrica petita per a la mètrica infinit.

2.10.3 Border Gateway Protocol (BGP)

BGP és un protocol d’encaminament EGP, és a dir, es fa servir entre sistemes autònoms (ASs). Cada AS té un identificador únic de 16 bits anomenat Autonomous System Number (ASN). IANA ha assignat blocs d’ASN als Regional Internet Registries (RIR), que a la vegada els distribueixen als ISPs.

Un AS pot ser un ISP (o conjunt d’ISPs) i els seus abonats. El conjunt de prefixos de xarxa que administren estan identificats pel seu ASN. La informació d’encaminament intercanviada per BGP fa servir els ASNs. El tràfic en un AS es classifica en:

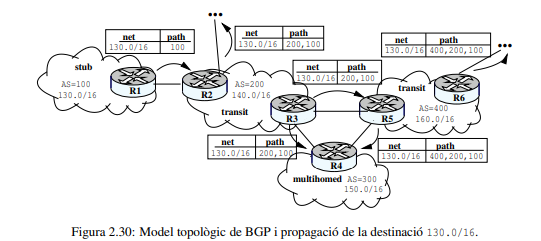
* Tràfic local (local traffic): és el tràfic que s’origina o acaba en el AS.
* Tràfic de trànsit (transit traffic): En cas contrari.

En funció del tipus de tràfic d’un AS, BGP defineix el model topològic de la figura 2.30:

* stub AS: AS que només té una connexió a un altre AS. Òbviament, només té tràfic local.
* multihomed AS: AS que té connexions a més d’un AS, però` que només porta tràfic local (típicament, les múltiples connexions es tenen per tolerància a falles).
* transit AS: AS que té connexions a més d’un AS i que suporta tràfic de trànsit (sota certes routing policies).

El funcionament bàsic de BGP és el següent:

* Entre cada parell de routers BGP veïns (BGP peers) s’estableix una connexió punt a punt (TCP) al port 179.
* Cada router BGP envia als seus veïns les xarxes destinació per a les quals vol encaminar tràfic (de forma anàloga a RIP). Per a cada xarxa destinació, BGP envia la llista d’ASN pels que s’ha que passar per arribar-hi. La llista d’AS que s’ha que passar per arribar a una destinació permet a BGP evitar bucles i el problema de count to infinity que té RIP. Per exemple, si un enllac¸ entre dos AS cau, BGP no triarà un camí alternatiu que també tingui aquest enllac¸. La figura 2.30 mostra un exemple dels missatges de BGP. El router R1 envia a R2 la destinació 130.0/16 amb el path 100. Quan aquest missatge arriba al router R2, aquest el distribueix als altres routers del mateix AS, i tots ells l’envien als seus peers d’altres ASs afegint el seu ASN al path (és a dir, el path passa a ser 200,100).
* Amb els missatges que s’intercanvien, els routers BGP van aprenent els ASN i les possibles destinacions d’Internet. Els missatges BGP també´ tenen altres atributs que depenen de les routing policies. També es fan servir mecanismes d’agregació de rutes per reduir el nombre de destinacions possibles. Els routers BGP guarden la informació que reben dels seus peers en una Routing Information Base (RIB). Amb tota aquesta informació BGP calcula les entrades que s’afegeixen a la taula d’encaminament.



**2.11 Seguretat en Xarxes IP**

Hi ha molts motius, entre ells legals, pels que és necessari introduir polítiques de seguretat en els negocis que fan servir Internet. Típicament, els objectius de la seguretat en una xarxa són:

* Confidencialitat: protecció de l’accés no autoritzat a dades.
* Integritat: protecció de la modificació no autoritzada de dades.
* Disponibilitat: garantia de funcionament dels serveis.

Les vulnerabilitats d’una xarxa poden tenir diferents orígens:

* Tecnològics: deguts als protocols (TCP/IP), sistemes operatius, equips de la xarxa (hubs)...
* De configuració: servidors “massa permissius”, comptes insegurs (anonymous), llistes d’accés´ mal configurades ...
* Falta de polítiques de seguretat: utilitzar aplicacions poc segures, passwords fàcils d’endevinar, absència de proteccions (firewalls) etc.

Els tipus d’atacs poden ser:

* De reconeixement: té com a objectiu trobar les vulnerabilitats de la xarxa i és el pas previ a un atac. Els passos solen ser:
  + Descobrir les adreces IP de la xarxa accessibles (amb pings).
  + Descobrir els ports “vius” (servidors) en cada adreça.
  + Connectar-se als servidors per descobrir els tipus, versions i sistemes operatius del sistema.

Un altre mecanisme de reconeixement és l’eavesdropping (que es podria traduir per “escoltar darrera de les portes”). Aquest mecanisme consisteix a capturar datagrames IP i decodificar el contingut.

* Accés: accés d’un intrús a un compte o servei.
* Denegació´ de servei (Denial of Service, DoS): té com a objectiu deshabilitar o corrompre un servei o xarxa perquè no hi puguin accedir els usuaris legítims.
* Cucs worms, virus i cavalls de Troia (Trojan horses): software maliciós que s’autoreplica, danya o corromp el sistema atacat.

A continuació hi ha una introducció a dues eines bàsiques per introduir seguretat en una xarxa: Firewalls i “Xarxes privades virtuals” (Virtual Private Networks, VPN).

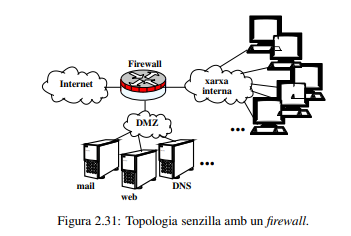
2.11.1 Firewalls

Un firewall (tallafocs) és genèricament un dispositiu que separa una xarxa que es desitja protegir, on hi ha els possibles “intrusos” (típicament la resta d’Internet). El nom de tallafocs ve de l’analogia amb aquests dispositius, fets de materials ignífugs, que impedeixen la propagació´ del foc en cas d’incendi. Hi ha diferents tipus de firewalls: Des dels més senzills que simplement filtren segons les adreces IP o ports fins a altres que són capaços de seguir i filtrar segons l’estat de les connexions i el tipus de missatges del nivell d’aplicació.

La figura 2.31 mostra una possible configuració per a una xarxa d’una oficina o petita empresa. Per referir-se a aquest tipus de xarxa a vegades es fa servir l’acrònim SOHO (Small Office – Home Office) o “xarxa corporativa” (corporate network). En la xarxa hi ha una zona anomenada DMZ (De-Militarized Zone). DMZ és un terme militar que es fa servir per identificar una zona neutral entre dos bàndols en conflicte. Anàlogament, en la zona DMZ hi ha els únics hosts (típicament servidors) que es desitja siguin accessibles des de l’exterior. Típicament el firewall restringeix l’accés des de l’exterior a la DMZ, i només a alguns ports dels servidors. Les eines bàsiques són:

* En la xarxa interior es fan servir adreces privades. El firewall fa servir NAT perquè els hosts de la xarxa interior puguin accedir a l’exterior.
* El firewall filtra els paquets que venen de l’exterior i no compleixen certes condicions. El filtrat es fa amb “llistes de control d’accés” (Access Control List, ACL). Les ACL s’apliquen a l’entrada o sortida d’una interfície. Si hi ha definida una ACL, tots els datagrames es comparen amb les regles de l’ACL. Quan es compleix alguna de les regles, el datagrama s’accepta o es descarta. Les regles d’una ACL es llegeixen en seqüencia i es deixen de mirar quan es compleix alguna de les regles. Per exemple, en la interfície d’entrada de la DMZ hi podria haver la següent ACL:
  + 1. Deixa passar els datagrames amb adreça destinació 200.200.10.10 i port 80.
  + 2. Descarta tot.

El resultat d’aquesta ACL és evident: només es podria accedir al servidor de web (port 80) del servidor 200.200.10.10. Amb qualsevol altra adreça destinació o port la primera regla no es compliria i es passaria a executar la segona, que ho descarta tot.



2.11.2 Xarxes privades virtuals

L’objectiu d’una xarxa privada virtual (Virtual Private Network, VPN) és el de proveir connectivitat als usuaris remots (Remote Access VPNs) o xarxes remotes (LAN-to-LAN VPNs) com si estiguessin connectats a la xarxa interna. La figura 2.32 en mostra un exemple. Les xarxes de les dues delegacions i el “tele-treballador” estan en punts geogràfics distants, però tenen accés a la xarxa com si estiguessin connectats a la mateixa LAN. Una solució alternativa per interconnectar les diferents xarxes o accessos remots és l’ús de línies d’operadors de telecomunicacions. Els operadors ofereixen diferents tecnologies per aquest tipus d’interconnexions: la xarxa telefònica, RSDI, Frame Relay o ATM en són alguns exemples. Per exemple, el tele-treballador podria fer servir un mòdem i la xarxa telefònica per trucar directament a una delegació i així tenir connectivitat sense passar per Internet. De forma anàloga, les dues delegacions també podrien estar connectades per una línia llogada Frame Relay. Aquest tipus de connexions, però, són bastant cares i, tanmateix, possiblement les xarxes desitgin tenir connexió a Internet. El repte d’una VPN és la seguretat. És a dir, aconseguir que l’accés a la xarxa interna sigui només possible als usuaris o xarxes autoritzades. Per aconseguir aquest objectiu les VPNs solen fer servir les tècniques següents:

* Autentificació: per reconèixer els usuaris autoritzats.
* Encriptació: per evitar l’eavesdropping.
* Túnels: per aïllar els enllaços remots de la resta d’Internet.

La figura 2.33 mostra un exemple de túnel. La figura 2.33(a) mostra la topologia i la figura 2.33(b) mostra la taula d’encaminament del router R1. El túnel en aquest exemple apareix identificat pel dispositiu tunl0. R1 encamina els datagrames dirigits a la xarxa 10.0.1.0 cap al túnel (vegeu la figura 2.33(b),). A l’entrada del túnel els datagrames s’encapsulen afegint una nova capçalera IP amb les adreces dels extrems del túnel. A la sortida del túnel els datagrames es desencapsulen, és a dir, s’elimina la capçalera externa i s’encaminen segons l’adreça destinació de la capçalera interna. En la pràctica hi ha diferents protocols que permeten assolir els objectius d’una VPN. Alguns d’ells són:

* IP over IP: túnel amb un encapsulament bàsic d’un datagrama IP dintre d’un altra datagrama IP (com el que mostra la figura 2.33).
* Generic Routing Encapsulation, GRE: túnel on s’afegeix una capçalera addicional, entre la capçalera IP externa i la interna, que permet altres funcionalitats com ara encapsular protocols diferents d’IP.
* Point-to-Point Tunneling Protocol (PPTP): túnel que té com a objectiu estendre la funcionalitat d’un enllaç ppp entre dos routers remots d’una VPN.
* IPsec: els túnels anteriors no estan encriptats i típicament es fan servir conjuntament amb altres protocols que encripten el canal, com ara ssh. IPsec pretén ser la solució estàndard per implementar autentificació i encriptació a nivell IP.

