Les capes de la xarxa de computadors

Xavier Vilajosana Guillén

PID_00147705

Universitat Oberta de Catalunya
www.uoc.edu

Índex

Introducció

En aquest mòdul veurem en detall les característiques i el funcionament dels diferents nivells de la xarxa. L'objectiu del mòdul és donar una visió general del funcionament intern d'una xarxa de computadors, tant d'àrea local com a gran escala, com és Internet. L'aproximació als nivells de la xarxa serà des dels nivells més propers a les aplicacions fins als nivells més específics del maquinari.

Com hem vist al mòdul "Conceptes de xarxes de computadors", les xarxes de computadors han estat estructurades en diferents nivells que abstrauen als nivells superiors les complexitats dels nivells inferiors. El funcionament del nivell d'aplicació, els seus serveis i protocols els veurem en detall al mòdul "El nivell d'aplicació". Abans, i per a entendre alguns dels conceptes del nivell d'aplicació, passarem pel nivell de transport, que ofereix fiabilitat a la xarxa tot permetent l'accés de diferents aplicacions a un únic medi de transmissió, la xarxa. Seguidament aprofundirem en el coneixement de la capa de xarxa. El nivell de xarxa és cabdal per al funcionament d'Internet, ja que ens permet adreçar i identificar els nodes d'una xarxa. En aquest mòdul coneixerem el funcionament actual de la xarxa i també les noves tecnologies de xarxa que esdevindran estàndards en un futur. La capa d'enllaç abstrau els nodes de la xarxa del medi físic sobre el qual transmeten la informació. Aquesta capa s'encarrega d'adreçar la informació entre nodes físics adjacents i garantir transmissió fiable entre ells. Cal destacar la tasca de la subcapa de control d'accés al medi que permet transmetre informació entre dos nodes independentment de la tecnologia de xarxa utilitzada, ja sigui sense fil, o cablejada. Finalment, el nivell físic s'encarrega dels aspectes de modulació i codificació dels senyals físics que són dependents directament de la tecnologia i el medi de transmissió.

El mòdul, en definitiva, us permetrà fer-vos una idea general i àmplia del funcionament d'una xarxa de computadors.

Objectius

L'estudi d'aquest mòdul us ha de permetre assolir els objectius següents:

1. Aprofundir en el coneixement de cada una de les capes de les xarxes de computadors.
2. Conèixer les funcions principals de la capa de transport.
3. Conèixer el funcionament del protocol TCP i del protocol UDP.
4. Conèixer el serveis oferts per la capa de xarxa.
5. Entendre l'adreçament en les xarxes de computadors.
6. Conèixer els estàndards de xarxa actuals.
7. Tenir una visió global del funcionament de la capa d'enllaç i física.

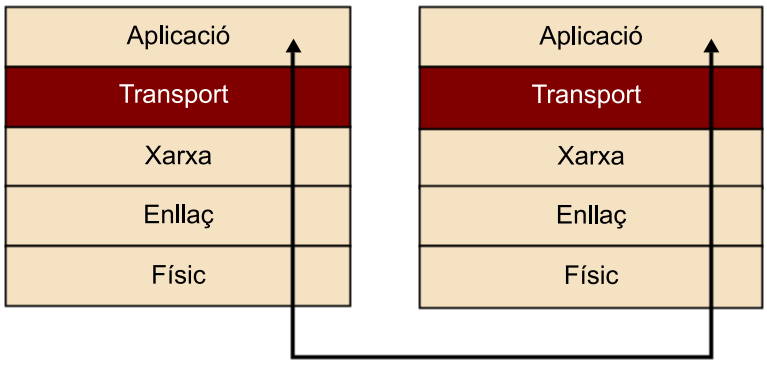
1. El nivell de transport

El nivell de transport abstrau la complexitat de la xarxa a les aplicacions. En aquest mòdul en veurem el funcionament amb detall.

1.1. Objectius

La capa de transport s'encarrega de proveir la comunicació extrem a extrem entre processos d'aplicacions ubicades en diferents amfitrions. Des del punt de vista de l'aplicació, és com si els amfitrions estiguessin directament connectats, però en realitat podrien estar en llocs oposats del planeta, connectats mitjançant múltiples encaminadors i tipus d'enllaços diferents. La capa de transport ofereix les funcionalitats bàsiques per a assolir aquest nivell de comunicació lògica, sense que les aplicacions s'hagin de preocupar de la infraestructura de xarxa subjacent.

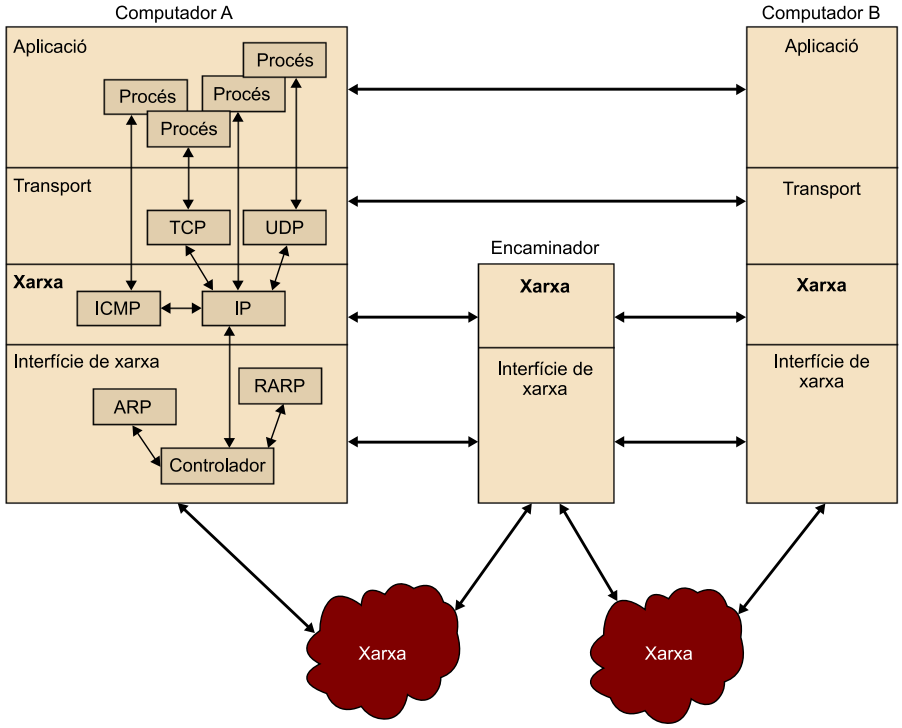
Missatges extrem a extrem



Els protocols de la capa de transport s'implementen en els dispositius extrems de la xarxa i no pas en els encaminadors ni dispositius intermedis. La informació transmesa per les aplicacions és convertida a paquets de la capa de transport, coneguts com a **segments de la capa de transport**. Els segments de la capa de transport es construeixen dividint la informació que vol transmetre l'aplicació en segments d'una mida determinada, afegint-hi una capçalera. Cada segment és enviat a la capa de xarxa, on seran inclosos en els paquets del nivell de xarxa coneguts com a **datagrames**. A partir de la capa de xarxa, els datagrames són enviats als destinataris passant per diferents dispositius que només examinaran la informació corresponent a la capa de xarxa. Només els receptors en rebre el datagrama a la capa de xarxa extrauran el segment de transport i el passaran a la capa de transport del receptor. La capa de transport processarà el segment i el passarà a l'aplicació corresponent.

En aquest mòdul estudiarem amb detall els protocols i les funcionalitats de la capa de transport de dades, fent èmfasi en els protocols de transport de dades d'Internet, el User Datagram Protocol (UDP) i el Transmission Control Protocol (TCP), que ofereixen serveis diferents a les aplicacions que els invoquen.

La jerarquia TCP/IP



1.2. Serveis oferts per la capa de transport

La capa de transport ofereix les funcionalitats següents:

- Garanteix la transmissió sense errors, extrem a extrem, independentment del tipus de xarxa.
- Controla la transmissió extrem a extrem.
- És responsable del control de flux de les dades i control de congestió de la xarxa.
- És responsable d'establir, mantenir i finalitzar les connexions entre dos amfitrions o un amfitrió i un servidor en una xarxa.
- Assegura que les dades arribin sense pèrdues, sense errors i sense ser duplicades.
- Ordena els paquets que arriben.

- S'encarrega de fragmentar els missatges i recompondre'ls en la destinació quan és necessari.

- Permet la multiplexació de diverses connexions de transport sobre una mateixa connexió de xarxa.

1.3. Relació entre la capa de transport i la capa de xarxa

La capa de transport s'ubica just per sobre de la capa de xarxa en la pila de protocols. Mentre que la capa de transport s'encarrega de proveir de comunicació lògica entre processos que s'executen en amfitrions diferents, la capa de xarxa proveeix de comunicació lògica entre amfitrions. Aquesta diferència és substancial, ja que la capa de transport ha de permetre que múltiples processos es comuniquin de manera lògica fent ús d'una única connexió lògica proveïda per la capa de xarxa. Aquest concepte s'anomena **multiplexació** i **desmultiplexació** de la capa de transport.

A més, la capa de xarxa no dóna garanties de lliurament de la informació, no garanteix el lliurament dels segments ni garanteix la integritat de la informació continguda en el segment. Per aquesta raó la capa de xarxa és **no confiable**. La missió, doncs, de la capa de transport, és proveir de transmissió de dades fiable i permetre la comunicació procés a procés en una xarxa creada comunicant els amfitrions.

Per a introduir els protocols del nivell de transport, ens centrem en la jerarquia de protocols Transmission Control Protocol/Internet Protocol (TCP/IP). En aquesta jerarquia es defineixen dos protocols de transport: l'UDP i el TCP.

L'UDP és no orientat a la connexió, fet que vol dir que no implementa fases d'establiment de la connexió, enviament de dades i acabament de la connexió; el TCP sí que és orientat a la connexió.

En el cas de la jerarquia TCP/IP, es defineixen dues adreces que relacionen el nivell de transport amb els nivells superior i inferior:

- L'**adreça IP** és l'adreça que identifica un subsistema dins una xarxa.
- El **port** identifica l'aplicació que requereix la comunicació.

Per a identificar les diferents aplicacions, els protocols TCP/IP marquen cada paquet (o unitat d'informació) amb un identificador de 16 bits anomenat **port**.

- **Ports coneguts**. Són regulats per la Internet Assigned Numbers Authority (IANA). Ocupen el rang inferior a 1024 i són utilitzats per a accedir a serveis oferts per servidors.

- **Ports efímers**. Són assignats de manera dinàmica pels clients dins d'un rang específic per sobre de 1023. Identifica al procés del client només mentre dura la connexió.

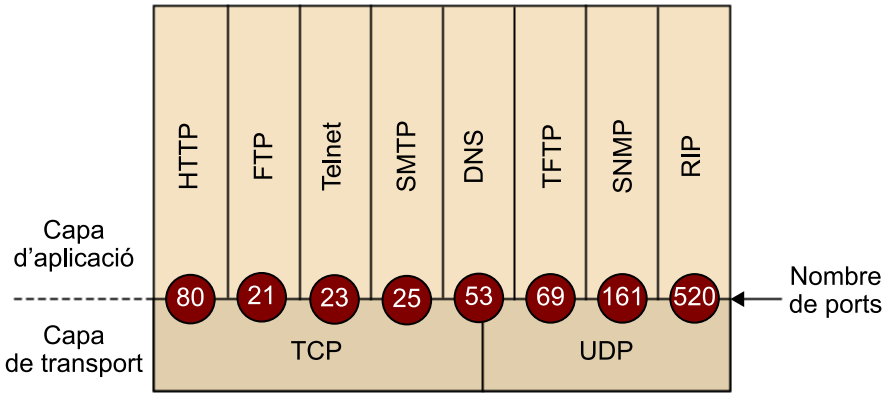
0-255 (IANA)	Aplicacions públiques
255-1023(IANA)	Assignats a empreses amb aplicacions comercials
> 1023	No estan registrats (efímers)

Ports d'aplicacions públiques establerts per la IANA

Decimal	Paraula clau	Descripció
0		Reservat
1-4		No assignat
5	RJE	Entrada remota de tasques
7	ECHO	Eco
9	DISCARD	Descartar
11	USERS	Usuaris actius
13	DAYTIME	De dia
15	NETSTAT	Qui està connectat o NETSTAT
17	QUOTE	Cita del dia
19	CHARGEN	Generador de caràcters
20	FTP-DATA	Protocol de transferència d'arxius (dades)
21	FTP	Protocol de transferència d'arxius
23	TELNET	Connexió de terminal
25	SMTP	Protocol SMTP (Simple Mail Transfer Protocol)
37	TIME	Hora
39	RLP	Protocol d'ubicació de recursos
42	NAMESERVER	Servidor de nom d'amfitrió
43	NICNAME	Qui és
53	DOMAIN	Servidor de denominació
67	BOOTPS	Servidor de protocol d'arrencada
68	BOOTPC	Client del protocol d'arrencada
69	TFTP	Protocol trivial de transferència d'arxius

Decimal	Paraula clau	Descripció
75		Qualsevol servei privat de connexió telefònica
77		Qualsevol servei RJE privat
79	FINGER	Finger
80	HTTP	Protocol de transferència d'hipertext
95	SUPDUP	Protocol SUPDUP
101	HOSTNAME	Servidor de nom d'amfitrió NIC
102	ISO-TSAP	ISO-TSAP
113	AUTH	Servei d'autenticació
117	UUCP-PATH	Servei de ruta UUCP
123	NTP	Protocol de temps de xarxa
137	NetBIOS	Servei de noms
139	NetBIOS	Servei de datagrames
143	IMAP	Interim Mail Access Protocol
150	NetBIOS	Servei de sessió
156	SQL	Servidor SQL
161	SNMP	Simple Network Management Protocol
179	BGP	Border Gateway Protocol
190	GACP	Gateway Access Control Protocol
194	IRC	Internet Relay Chat
197	DLS	Servei de localització de directoris
224-241		No assignat
242-255		No assignat

Ports usats per alguns dels protocols de nivell d'aplicació



Activitat

Coneixes aplicacions comercials o de programari lliure que facin servir un port determinat? Quins ports fan servir l'Skype? I l'Emule? El BitTorrent? L'MSN?

1.4. Transport no orientat a la connexió: UDP

El User Datagram Protocol (UDP) és un protocol no orientat a la connexió, de manera que no proporciona cap tipus de control d'errors ni de flux, tot i que utilitza mecanismes de detecció d'errors. En cas de detectar un error, l'UDP no lliura el datagrama a l'aplicació, sinó que el descarta.

Cal recordar que per sota l'UDP està fent servir l'IP, que també és un protocol no orientat a la connexió, i que UDP bàsicament el que ofereix en comparació d'IP és la possibilitat de multiplexar connexions de processos sobre una mateixa connexió de xarxa.

La simplicitat de l'UDP fa que sigui ideal per a aplicacions que requereixen pocs retards (per exemple, aplicacions en temps real o el DNS). L'UDP també és ideal per als sistemes que no poden implementar un sistema tan complex com el TCP. UDP és útil en aplicacions en què no importi la pèrdua de paquets, com telefonia o videoconferència sobre TCP/IP, monitoratge de senyals, etc. Aquestes aplicacions no toleren retards molt variables. Si un datagrama arriba més tard de l'instant que hauria de ser rebut, llavors es descarta perquè és inservible per a l'aplicació. Això es tradueix en un soroll en el so o imatge, que no impedeix la comunicació. La taula següent mostra algunes de les aplicacions més comunes que fan ús de l'UDP.

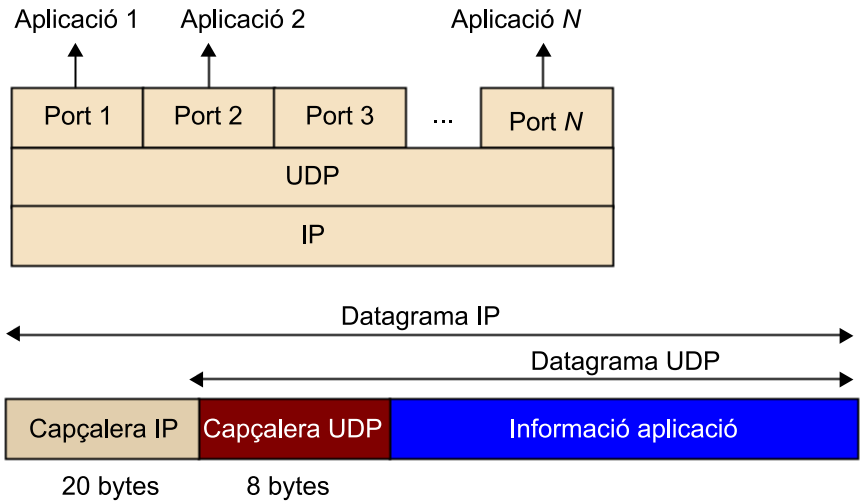
Aplicacions	Ports
TFTP	69
SNMP	161
DHCP-BOOTP	67/68
DNS	53

1.4.1. Encapçalament UDP

La unitat d'encapçalament de l'UDP és el datagrama UDP.

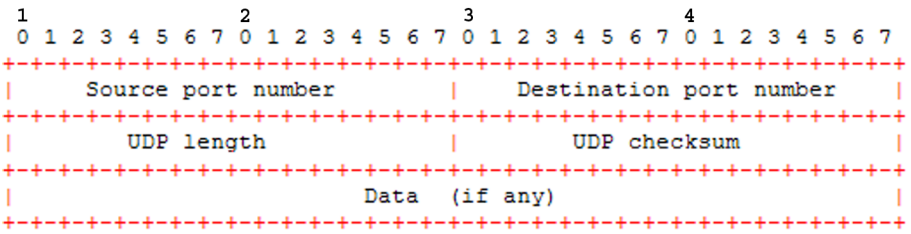
- Cada escriptura per part de l'aplicació provoca la creació d'un datagrama UDP. No hi ha segmentació.

- Cada datagrama UDP creat és encapsulat dintre d'un datagrama IP (nivell 3) per a transmetre'l.



1.4.2. Capçalera UDP

La capçalera del datagrama UDP està formada per 8 bytes. Té 4 camps.



- **Ports (font i destinació)**: permeten identificar els processos que es comuniquen.

- **UDP length**: longitud total del datagrama UDP (*payload* UDP + 8). És un camp redundant, ja que IP duu la longitud també. Com que la longitud màxima d'un datagrama IP és de 65.335 bytes (16 bits de longitud de datagrama), llavors:
  - Longitud màxima d'un datagrama UDP: 65.335 – 20 bytes = 65.315 bytes.
  - Longitud mínima d'un datagrama UDP: 8 bytes.
  - Longitud de dades d'un datagrama UDP: 65.315 – 8 bytes = 65.307 bytes.
  - La grandària d'un datagrama UDP depèn del sistema operatiu. Gairebé totes les API[1] limiten la longitud dels datagrames UDP (també per a TCP) a la màxima longitud de les memòries intermèdies de lectura i escriptura definides pel sistema operatiu (anomenades en el sistema *read* i *write*). Se solen usar 8.192 bytes com a grandària màxima del datagrama UDP (FreeBSD).

[1] Sigla d'*application programming interface*.

- **UDP checksum**: detector d'errors, l'objectiu del qual consisteix a detectar que ningú no ha modificat el datagrama UDP i que arriba a la seva destinació correctament. Si la suma de verificació UDP és errònia, es descarta el datagrama UDP i no es genera cap tipus de missatge cap a l'origen. La suma s'aplica conjuntament a una pseudocapçalera més la capçalera UDP més el camp de dades. Aquesta pseudocapçalera té 3 camps de la capçalera del paquet IP que s'envia. Noteu que la suma de verificació del datagrama IP només cobreix la capçalera IP.

Suma de verificació

La suma de verificació es calcula fent el complement a 1 de la suma de totes les paraules de 16 bits que conformen el datagrama UDP.

Exemple

Suposem que tenim el datagrama (descompost en 3 paraules de 16 bits):

0110011001100000

0101010101010101

1000111100001100

La suma de les dues primeres paraules de 16 bits és:

```
 0110011001100000
 0101010101010101
------------------
 1011101110110101
```

Afegint la tercera paraula obtenim:

```
 1011101110110101
 1000111100001100
------------------
 0100101011000010
```

Ara fem el complement a 1 (canviant 0 per 1) i obtenim:

1011010100111101

Aquesta darrera paraula s'afegeix també al datagrama UDP. En la recepció es fa la suma de totes les paraules de 16 bits, que ha de donar 1111111111111111.

1.5. Transport orientat a la connexió: TCP

El Transmission Control Protocol (TCP) és un protocol de nivell de transport que s'usa en Internet per a la transmissió fiable d'informació. Potser és el protocol més complex i important de la pila de protocols d'Internet.

Com hem vist, l'UDP no garanteix el lliurament de la informació que li proporciona una aplicació. Tampoc no reordena la informació en cas que arribi en un ordre diferent de l'ordre en què s'ha transmès. Hi ha aplicacions que no poden tolerar aquestes limitacions. Per a superar-les, el nivell de transport proporciona TCP.

El TCP dóna fiabilitat a l'aplicació, és a dir, garanteix el lliurament de tota la informació en el mateix ordre en què ha estat transmesa per l'aplicació d'origen. Per a aconseguir aquesta fiabilitat, el TCP proporciona un servei orientat a la connexió amb un control de flux i errors.

TCP és un protocol ARQ[2], extrem a extrem, orientat a la connexió (fases d'establiment de la connexió, tramesa de dades i tancament de la connexió) i bidireccional (dúplex). La unitat de dades TCP és el *segment TCP*. TCP, a diferència d'UDP, intenta generar segments de mida òptima, que s'anomena *maximum segment size* (MSS), generalment la major possible, per a minimitzar el sobrecost (*overhead* en anglès) de les capçaleres, però que no produeixi fragmentació a escala d'IP. Igual que UDP, TCP utilitza multiplexatge mitjançant l'ús de ports, tal com hem vist amb anterioritat.

(2)*ARQ* és la sigla d'*Automatic Repeat-reQuest Protocol*; en català, 'protocol de repetició de petició automàtic'.

Proporciona fiabilitat mitjançant els controls següents:

- **Control d'errors**. TCP és semblant al Go-Back-N però no descarta segments posteriors quan arriben fora de seqüència. Permet les retransmissions selectives.

- **Control de flux** (finestra advertida). Serveix per a adaptar la velocitat entre l'emissor i el receptor. Vigila que l'emissor no enviï els segments més ràpidament de la velocitat amb què els pot processar el receptor, de manera que es puguin perdre paquets per saturació de la memòria intermèdia de recepció.

- **Control de la congestió** (finestra de congestió). Per a adaptar la velocitat de l'emissor als encaminadors intermedis de la xarxa i evitar així que se'n col·lapsin les memòries intermèdies i es puguin perdre paquets.

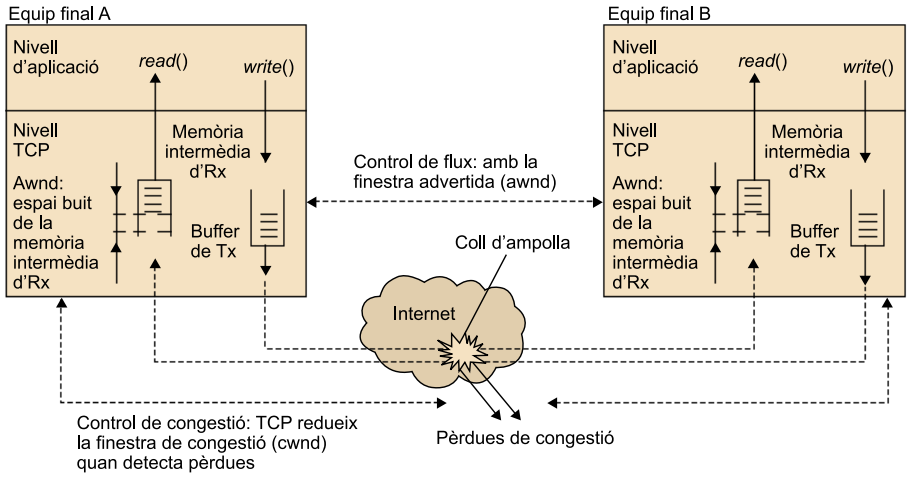
En transmissió, TCP s'encarrega de fragmentar les dades del nivell d'aplicació, i els assigna un número de seqüència abans d'enviar els fragments al nivell d'IP.

En recepció, com els segments poden arribar fora d'ordre, TCP els ha de reordenar abans de passar-los als nivells superiors.

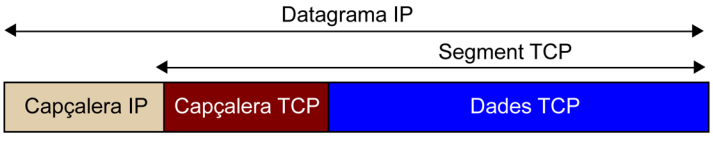
1.5.1. Funcionament bàsic de TCP

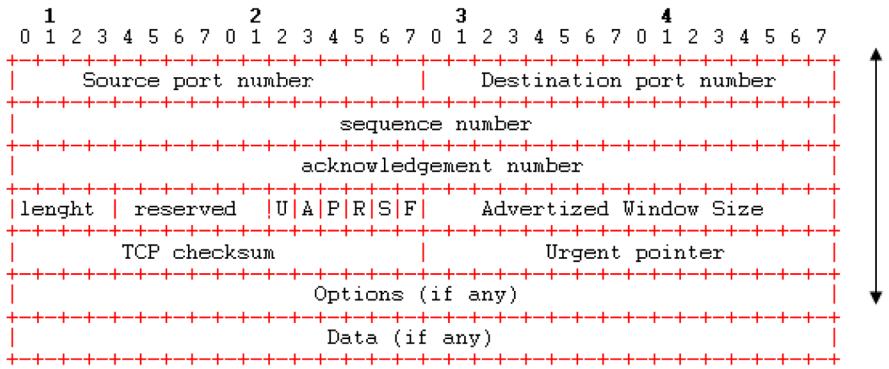
En cada extrem, TCP manté una memòria intermèdia de transmissió i una de recepció. L'aplicació utilitza les crides al sistema operatiu *read()* i *write()* per a llegir i escriure en aquestes memòries intermèdies. Quan l'aplicació escriu la informació que s'ha d'enviar a l'emissor, TCP la desa en una memòria intermèdia de transmissió. Quan la memòria intermèdia de Tx és plena, la crida *write()* queda bloquejada fins que hi torni a haver espai.

Cada vegada que arriben segments de dades al receptor, l'API *read*() els passa al nivell superior i s'envien les confirmacions corresponents. Les dades, segons són confirmades pel receptor, s'esborren de la memòria intermèdia de transmissió i queda espai lliure perquè l'aplicació continuï escrivint.



1.5.2. Capçalera TCP





La capçalera TCP està formada pels elements següents:

- **Source port / destination port**: ports origen i destinació que identifiquen les
aplicacions.

- **Sequence number**: número de seqüència del segment. Identifica el primer byte dins d'aquest segment de la seqüència de bytes enviats fins aquell moment.

- **Initial seq. number (ISN)**: primer número de seqüència escollit pel protocol TCP.

- **Seq. Number**: serà un número a partir d'ISN. Per tant, per a saber quants bytes hem enviat cal fer "*seq. number* – ISN".

- **Acknowledgment number**: conté el pròxim número de seqüència que el transmissor de l'ACK espera rebre, és a dir, és "*seq. number* + 1" de l'últim byte rebut correctament.
  – TCP és dúplex: cada extrem manté un *seq. number* i un *ACK number*.

- **Header length**: longitud total de la capçalera del segment TCP en paraules de 32 bits (igual que el camp *header length* de la capçalera IP).
  – Mida mínima de la capçalera TCP = 20 bytes (*header length* = 5).

  – Mida màxima de la capçalera TCP = 60 bytes (*header length* = 15).

- **Reserved**: bits reservats per a possibles ampliacions del protocol. Se'n posen 0.

- **Flags**: hi ha 6 indicadors (bits) a la capçalera. Són vàlides si estan a 1.

- **Urgent (URG)**: indica que s'utilitza el camp *urgent pointer* de la capçalera TCP.

- **Acknowledgement (ACK)**: indica que s'utilitza el camp *acknowledgement number*. Vàlid per a tots menys per al SYN inicial.

- **Push (PSH)**: indica que el receptor ha de passar les dades de la memòria intermèdia de recepció als nivells superiors tan ràpid com sigui possible. Operació més ràpida sense omplir la memòria intermèdia. L'activació d'aquest indicador depèn de la implementació. Les implementacions derivades de BSD l'activen quan la memòria intermèdia de Tx es queda buida.

- **Reset (RST)**: s'activa quan es vol avortar la connexió. Un exemple és quan es rep un segment d'un client dirigit a un port en què no hi ha cap servidor escoltant. En aquest cas, TCP contesta amb un segment amb la indicació de *reset* activada.

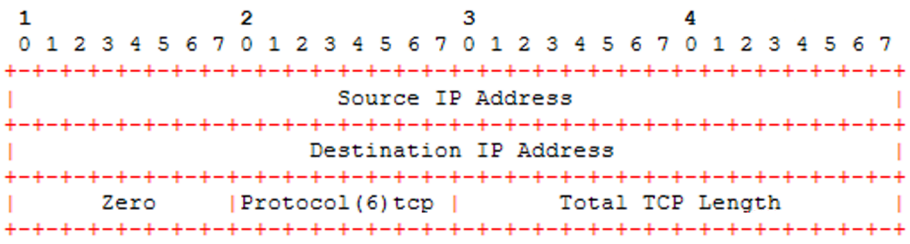
- **Synchronize (SYN)**: s'utilitza a l'establiment de la connexió, durant la sincronització dels números de seqüència a l'inici de la connexió.

- **Finalize (FINAL)**: s'utilitza en l'acabament de la connexió.

- **Advertised window size**: mida de la finestra advertida pel receptor al transmissor (*sliding window*) per al control de flux. La màxima finestra és de 65.535 bytes.

- **TCP checksum**: s'utilitza per a detectar errors en el segment TCP, i per a tenir més certesa que el segment no ha arribat a una destinació equivocada.

- Igual que en UDP, la suma de verificació TCP s'aplica conjuntament a una pseudocapçalera + la capçalera TCP + el camp de dades TCP. Aquesta pseudocapçalera té 3 camps de la capçalera IP i la mida del segment TCP (capçalera + *payload*).
  - La pseudocapçalera només és per a calcular la suma de verificació: no es transmet en el segment TCP, ni s'inclou en la longitud del paquet IP.
  - La mida del segment TCP no es posa en la capçalera TCP, només es té en compte en el càlcul de la suma de verificació.
  - A diferència d'UDP, en TCP el càlcul de la suma de verificació és obligatori.



- **Urgent pointer**: camp vàlid si la indicació URG = 1. Implementa un mecanisme per a indicar dades urgents (és a dir, que s'han d'atendre al més aviat possible). És un punter al *seq. number* que indica la part de dades urgents dins del camp de dades. Les dades urgents aniran del primer byte del segment al byte indicat per l'*urgent pointer*. Aquesta indicació s'utilitza poques vegades. Un exemple és quan es tecleja un control+C (interrupció) des de l'aplicació Telnet.

- **Options**: TCP permet afegir opcions a la capçalera, però a diferència d'IP, les opcions de TCP se solen utilitzar. En podem destacar:
  - **Maximum segment size**: s'utilitza durant l'establiment de la connexió per a suggerir el valor de l'MSS en l'altre extrem MSS = MTU xarxa directament connectada – capçalera IP – capçalera TCP (sense opcions). Si la xarxa és Ethernet (MTU 1500), llavors MSS = 1460.
  - **Window scale factor**: s'utilitza durant l'establiment de la connexió per a indicar que el valor de la finestra advertida s'ha de multiplicar per aquest factor d'escala. Això permet advertir finestres majors que 216.

- **Timestamp**: s'utilitza en el càlcul de l'RTT.

- **SACK**: permet que TCP faci retransmissió selectiva (*selective ack*). TCP utilitza el camp *ack* per a indicar fins on s'ha rebut correctament. Amb l'opció SACK el receptor pot indicar blocs de segments que s'han rebut correctament més enllà del segment confirmat per l'*ack*. D'aquesta manera, l'emissor pot escollir millor els segments que s'han de retransmetre.

- **Padding**: bytes de farcit afegits perquè la capçalera tingui un múltiple de 32 bits.

Activitat

Assumim que un extrem client TCP (emissor) ha triat el 28.325 com a número de seqüència inicial (ISN), mentre que l'extrem receptor TCP (servidor) ha triat com a ISN el 12.555. Què indica un segment client (emissor) TCP amb número de seqüència 29.201, número ACK 12.655 i finestra 1.024?

Solució

El número de seqüència indica que el client ja ha transmès des del byte 28.325 fins al byte 29.200 (875 bytes en total) i que en aquest segment transmetrà a partir del byte 29.201. El número ACK indicarà al receptor que l'emissor ha rebut correctament fins al byte 12.654 i que espera rebre a partir del 12.655. La finestra indica al receptor que el client només pot acceptar 1.024 bytes abans de confirmar-los. Per tant, el servidor TCP actualitzarà la seva finestra de transmissió a 1.024.

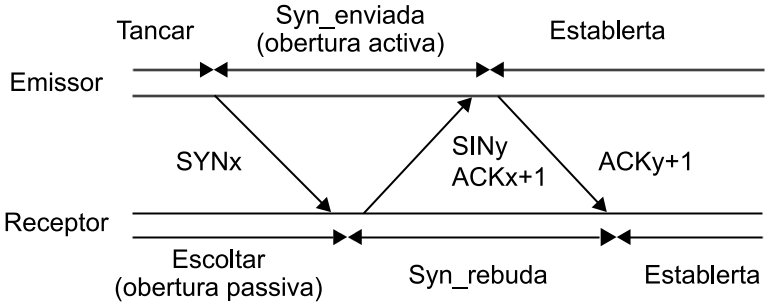
1.5.3. Establiment de la connexió

TCP és un protocol orientat a la connexió, tal com hem dit amb anterioritat. Això implica que en tot procés de comunicació hi haurà una fase d'establiment de la connexió en què emissor i receptor se sincronitzen per tal de poder intercanviar dades. En el TCP s'usa l'algoritme 3-Way Handshake, que consisteix en l'intercanvi de tres segments que no porten dades (només és la capçalera TCP):

1) L'emissor envia un segment SYN, amb petició de connexió.

2) El receptor (servidor) torna un segment SYN + ACK com a resposta.

3) El client respon amb un segment ACK reconeixent SYN + ACK. Un cop establerta la connexió es passa a la fase d'enviament de dades.



De la mateixa manera que l'establiment de la connexió, es fa necessari un procés d'acabament de la connexió. El tancament de la connexió pot ser per diverses causes:

- El client o el servidor tanquen la connexió (*LLS close()*).
- Per alguna raó s'envia un *reset* de la connexió (indicació activa RST).
- Tancament a causa d'una interrupció, un control∧D o un control∧C, etc.

L'acabament també es fa per mitjà de l'enviament de segments TCP. El primer segment de final el pot enviar tant el client com el servidor. El tancament normal és a causa d'un *close()* del client, fet que provoca l'intercanvi de 3 o 4 segments TCP.

2. El nivell de xarxa

La capa de xarxa s'encarrega de proporcionar connectivitat i d'oferir mecanismes per a la selecció del millor camí entre dos punts separats de la xarxa, i permet la interconnexió d'equips que poden estar ubicats en xarxes geogràficament separades entre si, tot garantint la connectivitat extrem a extrem, independentment de la tecnologia d'enllaç de dades utilitzada i del camí que segueixi la informació en els punts intermedis.

Els principals avantatges que ens proporciona aquesta capa són, per una banda, independència de la tecnologia de xarxa (envers capes inferiors), i per l'altra, un sistema d'abstracció que permet utilitzar una gran diversitat d'aplicacions i protocols de transport (envers capes superiors), com per exemple TCP o UDP.

Bàsicament la capa de xarxa, especialment a Internet, està composta per tres grans blocs:

1) el protocol, que descriu la manera d'enviar informació,

2) el protocol d'encaminament, que decideix per on han d'anar els datagrames per a arribar a la seva destinació,

3) la capa de xarxa també identifica el mecanisme per a informar de qualsevol error que s'hagi produït en l'enviament de la informació.

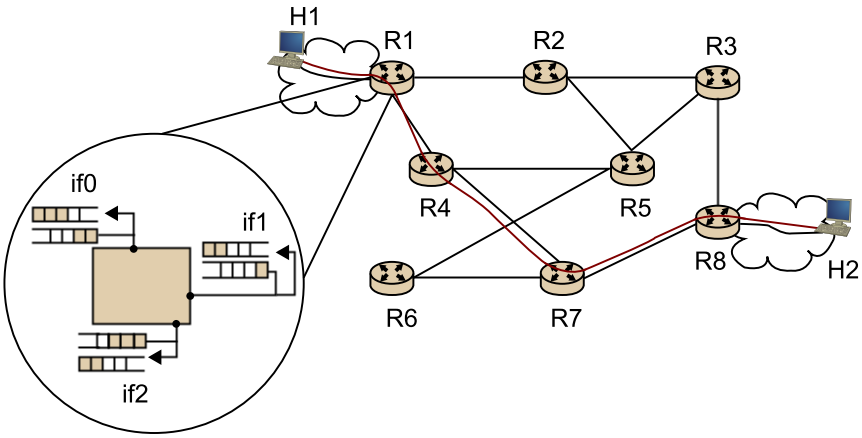
2.1. Funcionalitats bàsiques: encaminament

Una xarxa està composta bàsicament per dos tipus d'entitats, els clients (també coneguts amb el nom d'amfitrions, o bé equips finals) i els encaminadors. Els clients són els equips de xarxa encarregats de la comunicació, són l'origen i el final. Normalment són servidors d'informació o bé equips d'usuaris finals que accedeixen als servidors. Per la seva part, els encaminadors, tot i que en segons quins casos també poden ser equips finals, es limiten a enviar la informació que reben per una interfície d'entrada a la corresponent de sortida que porti els datagrames cap a la destinació. Per a poder saber cap a on va la informació els encaminadors s'ajuden del que es coneix com a *taules d'encaminament*.

La capa de xarxa necessita que tant els encaminadors com els equips finals tinguin un identificador únic. Aquest identificador permet que qualsevol altre equip de la xarxa el pugui localitzar i enviar-li informació. En particular, en una xarxa com Internet aquests identificadors es coneixen com a *adreces* (adreces IP).

Exemple de xarxa amb encaminadors i equips finals

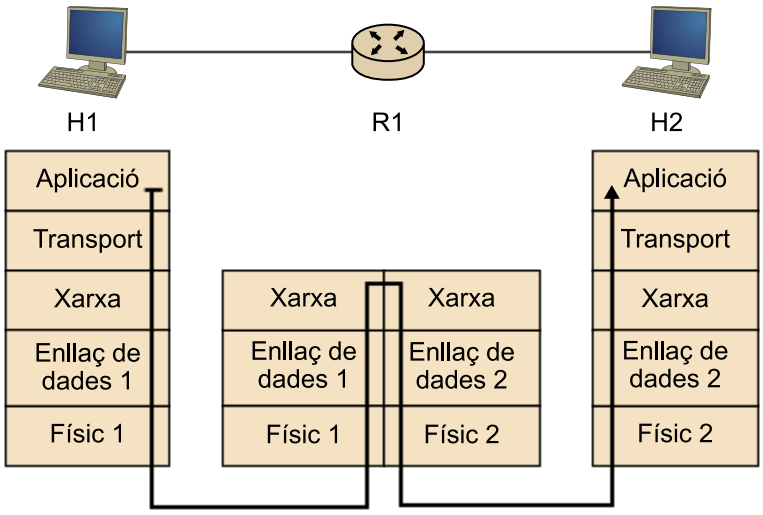
La figura següent mostra una xarxa amb vuit encaminadors i dos equips finals. A la figura també es pot observar una simplificació de com funciona un encaminador internament. Per simplicitat en comptes d'indicar les adreces dels diversos equips ho hem identificat per una banda amb una *R* (de *router*, encaminador) i un número que identifica els diferents encaminadors, i per l'altra amb una *H* (de *host*, amfitrió) i un número per a identificar els diferents equips finals.



Els encaminadors estan compostos per una sèrie d'interfícies d'entrada i sortida, que són les encarregades de rebre els datagrames dels equips veïns; aquestes interfícies estan controlades per unes cues (o *buffers*), que emmagatzemen els paquets (d'entrada o de sortida) per a poder-los enviar quan sigui possible, o el que és el mateix, quan l'encaminador tingui recursos per a atendre les cues d'entrada, o bé quan la xarxa tingui recursos (amplada de banda disponible) per a les cues de sortida. Internament l'encaminador disposa d'una lògica per a decidir què ha de fer amb els datagrames que arriben. Aquesta decisió normalment implica enviar el datagrama per una altra interfície que el portarà més a prop de la destinació.

Així el datagrama va saltant pels encaminadors fins a arribar a la destinació. Cada equip de xarxa pel qual passa el datagrama es coneix com a *salt* o *hop*. Cal notar que els encaminadors treballen a escala de xarxa, cosa que vol dir que no interpreten els camps presents en els nivells superiors, tal com mostra la figura següent.

Capes usades per a l'encaminament en protocols de xarxa



Quan un equip envia un datagrama cap a una destinació, aquest datagrama inicialment va dirigit a l'encaminador associat a la xarxa de l'equip. Aquest encaminador mirarà la destinació del datagrama, i l'enviarà per la interfície que el porti cap a la destinació, depenent d'una taula d'encaminament. L'encaminador següent farà el mateix fins que el datagrama arribi a la seva destinació final. La llista d'encaminadors que segueix un datagrama es coneix com el *camí* o *path* del datagrama. Cal notar que aquest *path* serà diferent depenent de l'origen i la destinació del datagrama.

Com a mostra es pot veure a la figura de l'exemple anterior que el camí que segueixen els datagrames per a anar des d'H1 fins a H2 és H1-R1-R4-R7-R8-H2, i fa un total de 5 salts per a arribar a la destinació.

Hem dit que l'equip envia el datagrama a l'encaminador de la seva xarxa, i això implica que s'ha de tenir coneixement *a priori* de com cal arribar a aquest encaminador per a poder-li enviar el datagrama. La seqüència específica d'accions que fa l'equip es poden veure a la figura següent; més detalladament:

1) Es crea el datagrama amb les adreces de xarxa origen i destinació apropiades.

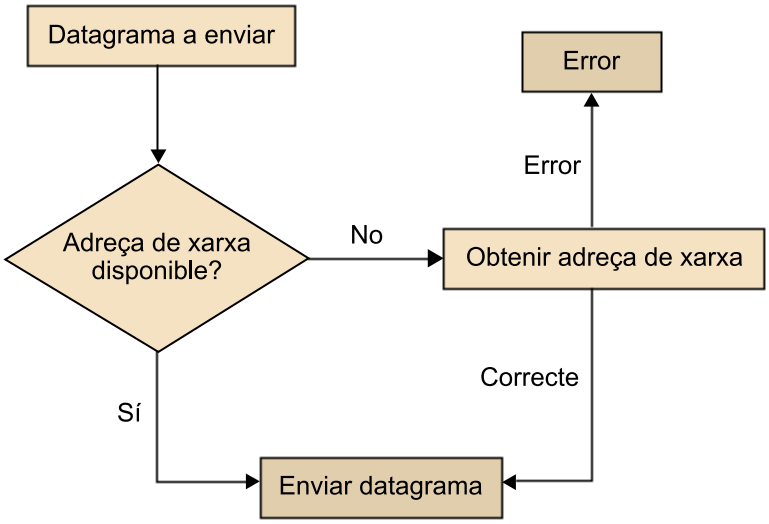
2) Es busca l'adreça de xarxa de l'encaminador *–next hop–* (o de l'equip final si està directament connectat a l'encaminador *–last hop–*).

3) Si no es disposa de l'adreça de xarxa sortim amb error, ja que no sabem quin és el salt següent per a enviar el datagrama.

4) Si hem pogut aconseguir l'adreça enviem el datagrama (amb les adreces de xarxa origen i destinació originals) al salt següent del *path* o a l'equip final si ens trobem al *last hop*.

5) Cal repetir des del pas 2 fins que el datagrama arribi a la seva destinació.

Diagrama de blocs simplificat de l'enviament d'un datagrama a un equip de xarxa



El datagrama no segueix qualsevol *path*, sinó que els encaminadors disposen d'una taula d'encaminament (*forwarding table* o *routing table*) que indica per quina interfície s'han d'enviar els datagrames depenent de la seva destinació. Per a omplir aquestes taules és necessari utilitzar uns algorismes d'encaminament.

2.2. Serveis de xarxa

El servei de xarxa defineix les característiques que ha de tenir el transport punt a punt de les dades en la capa de xarxa. Així es defineixen característiques com la fiabilitat enviant la informació, ordre d'arribada dels paquets, els llindars de retard en fer arribar la informació a la destinació, la informació de congestió a la xarxa, etc., entre els diferents emissors i receptors dins de la xarxa.

Actualment hi ha dos models de serveis de xarxa clarament diferenciats: el **model de circuit virtual** i el **model de datagrama**. A continuació es descriuen tots dos, fent èmfasi en el model de datagrama, atès que és l'utilitzat pel nivell de xarxa proposat per Internet, i per tant el més rellevant actualment.

2.2.1. Model de xarxa en mode de circuits virtuals

Un circuit virtual és un camí que es preconfigura entre dos punts de la xarxa de manera que els nodes intermedis saben *a priori* la direcció a la qual s'ha d'enviar la informació pertanyent a cada circuit. Aquest paradigma permet accelerar enormement l'enviament de paquets entre dos punts, ja que el processament intermedi és mínim; aquesta prereserva a sobre permet garantir una sèrie de recursos de xarxa per al trànsit que passa pel circuit. Per això aquest model de xarxa es va pensar per a serveis en temps real (multimèdia).

En qualsevol circuit virtual es poden distingir tres fases clarament separades:

1) Establiment del circuit virtual: aquesta fase s'inicia a la capa de xarxa de l'emissor, utilitzant l'adreça del receptor. L'emissor envia un datagrama de creació de circuit que provoca que cada node intermedi reservi els recursos demanats de manera iterativa fins a arribar a la destinació. Cada un dels nodes intermedis haurà d'actualitzar el seu estat per a acomodar el nou circuit, o denegar-ne la creació en cas que no quedin més recursos disponibles (normalment amplada de banda). Si l'establiment del circuit pot arribar fins al destinatari s'avisa a l'emissor indicant que la connexió ha estat satisfactòria i que es pot començar a enviar informació.

2) Transferència de dades: en el cas que s'hagi pogut establir el circuit virtual es poden començar a enviar dades entre els dos punts.

3) Desconnexió del circuit virtual: aquesta desconnexió pot ser iniciada tant per l'emissor com pel receptor, i s'avisa seqüencialment per mitjà de la capa de xarxa a tots els nodes intermedis fins a arribar a l'altre extrem. Aquesta desconnexió permet alliberar els recursos ocupats pel circuit.

Activitat

Quines diferències creieu que hi pot haver entre l'inici d'un circuit virtual a la capa de xarxa i l'establiment d'una connexió a la capa de transport? (per exemple, el *three-way-handshaking*).

Solució

L'establiment de la connexió de la capa de transport involucra únicament dos sistemes finals. Els dos extrems acorden la comunicació i determinen els paràmetres de connexió, mentre que els nodes intermedis de la xarxa no hi intervenen. En contraposició, l'establiment d'un circuit virtual a la capa de xarxa obliga a involucrar tots els nodes intermedis.

El principal inconvenient que té la utilització de circuits virtuals és que els nodes intermedis han de mantenir les reserves de recursos demanades independentment que s'estiguin utilitzant o no, amb el problema potencial d'infrautilitzar la xarxa.

2.2.2. Model de xarxa en mode datagrama

Si enviar informació a través d'un circuit virtual implica prèviament establir un camí i reservar recursos, en una xarxa en mode datagrama (també anomenat *commutació de paquets*), el paquet s'envia directament a la xarxa amb una adreça origen i una adreça destinació. Aleshores és feina de la xarxa (per mitjà de les taules d'encaminament de cada encaminador) fer arribar el paquet a la seva destinació.

Com es pot comprovar, en aquest tipus de comunicació no hi ha ni reserva de recursos ni camí preestablert entre els extrems de la comunicació. Per tant, a un encaminador poden arribar datagrames de diferents destinacions a la vegada, i els datagrames poden seguir camins diferents per a arribar a la destinació (depenent dels algorismes d'encaminament), cosa que provoca l'efecte col·lateral que els paquets poden arribar fora d'ordre (el paquet número 2 arriba abans que el número 1).

Les xarxes en mode datagrama actualment són les més usades, principalment perquè el protocol de xarxa d'Internet (IP) l'utilitza. Tot i que hem vist que el model de datagrama fa una utilització dels recursos més eficient, això té un cost associat. Amb aquest tipus de xarxes es complica moltíssim la priorització del trànsit, ja que mai no se sap *a priori* quant trànsit es rebrà, i el que és més greu, no se sap quina prioritat s'ha de donar a cada un dels fluxos de dades presents a la xarxa; tant és així que Internet es basa en el paradigma conegut com a *best effort*, que implica que la xarxa no ens dóna cap garantia de qualitat i que "ho farà el millor que pugui" per a fer arribar el datagrama a la seva destinació.

Activitat

Quin dels dos models de xarxa vistos considereu que fa un ús dels recursos més eficient?

Solució

El fet que un circuit virtual obliga a fer una prereserva de recursos implica que s'ha de conèixer prèviament el model i el patró de trànsit que segueix l'aplicació; com això molts cops no és possible *a priori*, s'acostuma a fer el que es coneix com a *overprovisioning* (reservar més recursos dels que es consideren necessaris), fet que inequívocament porta a un sistema menys eficient en termes de recursos.

Per la seva banda, utilitzar el mode datagrama no implica cap prereserva, i per això la xarxa sempre enviarà tan ràpid com pugui la informació, sempre que hi hagi recursos disponibles.

2.2.3. Servei de xarxa orientat i no orientat a la connexió

Anàlogament als protocols de transport que hem vist anteriorment, en el nivell de xarxa també podem tenir protocols que siguin orientats a connexió i d'altres que no ho siguin. La principal diferència entre les dues alternatives és que el servei orientat a connexió desa l'estat de la connexió, o el que és el mateix, té coneixement de totes les connexions establertes, mentre que en el

cas del servei no orientat a connexió no es té constància de les connexions existents. Un exemple clar de servei de xarxa orientat a la connexió és el model de circuits virtuals vist anteriorment.

Cal notar que el disseny d'un protocol de xarxa no orientat a connexió no exclou que en nivells superiors (transport) es pugui definir un protocol orientat a connexió. L'exemple més indicatiu d'això és la pila de protocols TCP/IP, en què TCP és orientat a la connexió, mentre que IP no ho és. Tant és així que l'arquitectura actual d'Internet només proporciona el model de servei de datagrama, que no garanteix l'ordre dels paquets, el retard en l'enviament, ni l'arribada del datagrama.

2.3. Adreçament a Internet: el protocol IP

El protocol de capa de xarxa per excel·lència és l'Internet Protocol (IP). IP és un protocol que basa l'intercanvi d'informació amb el model no orientat a connexió. IP és el protocol utilitzat a Internet per a identificar els nodes de la xarxa, i també s'utilitza per a enviar la informació d'una manera estàndard i independent de la tecnologia de xarxa utilitzada. Una altra característica molt important és que IP no implementa mecanismes que garanteixin la integritat de les dades que s'envien per la xarxa (això es fa a la capa de transport); només es verifica que no hi hagi errors de transmissió a la capçalera.

Tots els protocols de xarxa requereixen algun mecanisme per a identificar els nodes de la xarxa, i aquesta identificació en el protocol IP es fa per mitjà del que es coneix com a *adreça IP*; actualment hi ha dues versions diferents del protocol IP: IPv4 i IPv6. IPv4 és el protocol més utilitzat actualment a Internet, però atès el gran creixement que ha sofert la xarxa, se n'ha proposat una extensió, IPv6, més actual i que algun dia es preveu que substitueixi IPv4.

Vegeu també

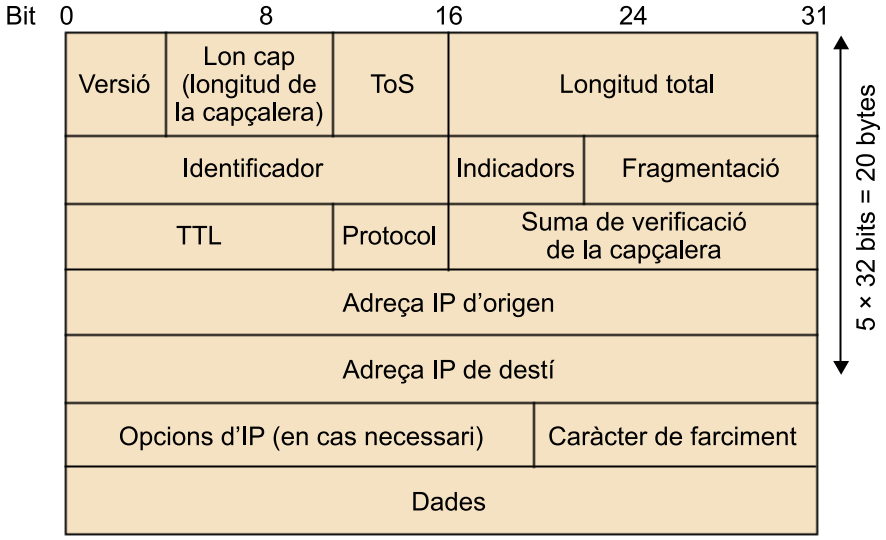
En el subapartat "Adreçament a Internet: el protocol IP" es detalla com funcionen els protocols IPv4 i IPv6 i quins avantatges i inconvenients tenen.

2.3.1. IPv4

IPv4 va ser proposat el 1981 (document RFC-791) i actualment encara és el protocol de xarxa per excel·lència. IPv4 defineix el format que s'ha d'utilitzar per a enviar informació entre dos punts distants de la xarxa; el protocol proporciona mecanismes que determinen com es divideix l'adreçament d'una manera escalable en una xarxa tan gran com Internet.

La capçalera IP

IPv4 defineix quina informació de control i quin format han de tenir els paquets que s'envien a la xarxa. Per això, i igual com ocorre amb els protocols de transport vistos anteriorment, és necessari definir una capçalera que serveixi per a poder identificar els paquets. La capçalera d'IPv4 es pot veure a la figura següent.



En aquesta figura podem veure els elements següents:

Versió (4 bits): indica quin protocol de xarxa utilitza aquest datagrama. Per a IPv4 està fixat a `0x04`.

- **Hdr. len (4 bits)**: la capçalera IP pot tenir una mida variable a causa del camp d'opcions. Aquesta mida indica en quin punt comencen les dades del protocol de transport. En particular aquest camp indica el valor en funció de la quantitat de paraules de 4 octets que té la capçalera; així un valor de `0x05` vol dir una capçalera de 20 octets, el valor usat en la majoria dels casos com a mida per defecte quan no hi ha opcions.

- **Type of service (TOS) (8 bits)**: aquest camp permet distingir entre diferents tipus de datagrames IP; inicialment es van definir paràmetres en funció de: retard baix, taxa de transferència alta o fiabilitat. Així, depenent del tipus de trànsit que contingui el paquet, per exemple trànsit interactiu, es pot voler un retard baix, o en el cas que el trànsit sigui de baixa prioritat es pot voler cost mínim. La llista dels diferents tipus de servei es pot trobar en el document RFC-1349. En la realitat, els encaminadors normalment ignoren aquest camp i utilitzen la tècnica *best effort* per a encaminar els paquets.

- **Total length (16 bits)**: indica la mida total del datagrama en octets, i això inclou la capçalera i el camp de dades. Els 16 bits indiquen una mida mà-

xima del datagrama de 65.535 octets, tot i que en general la mida màxima utilitzada és de 1.500 octets.

- **Identifier (16 bits), flags (3 bits) i fragmentation (13 bits)**: aquests camps fan referència al que es coneix com a *fragmentació IP*.

Vegeu també

La fragmentació serà explicada en el subapartat "Fragmentació IP".

- **TTL (8 bits)**: inicialment aquest camp feia referència al temps de vida del datagrama en mil·lisegons. Però en la pràctica conté el màxim nombre d'encaminadors que pot travessar el paquet fins que arribi a la destinació. En cada salt, un encaminador decreix en 1 el valor d'aquest camp, i quan el TTL arriba a 0 el paquet és descartat. Amb aquesta tècnica es permeten descartar datagrames en el cas que hi hagi algun bucle provocat per algun problema amb el sistema d'encaminament, i així evitar tenir paquets a la xarxa més temps del necessari. D'aquest camp es pot derivar que el "diàmetre" màxim possible d'Internet és de 255 salts. Tot i que actualment no acostuma a superar els 30.

- **Protocol (8 bits)**: aquest camp indica el protocol present en la capa de transport, que serà capaç d'interpretar-lo. Normalment aquest camp pot ser `0x06` per a TCP o `0x11` per a UDP. La llista completa es pot trobar en els documents RFC-1700 i RFC-3232. Amb aquest enllaç entre la capa de xarxa i la de transport, permet tenir diversos protocols de transport i poder-los distingir fàcilment, i passa el control al corresponent de manera eficient.

- **Header checksum (16 bits)**: permet detectar algun tipus d'error de transmissió en la capçalera. És important notar que no es comprova la integritat de la capa de transport i superiors. Recordem que IP no garanteix la recepció de les dades. La suma de verificació es calcula tractant cada dos octets de la capçalera com enters i sumant-los utilitzant aritmètica de complement a 1, ignorant per a la suma el camp mateix que conté la suma de verificació. La integritat es comprova comparant la suma amb l'emmagatzemada en la capçalera. En el cas d'error el paquet es descarta. Un petit inconvenient d'aquesta suma de verificació és que cada encaminador l'ha de recalcular per a cada paquet, atès que el camp TTL (i potser algunes opcions) canvien a cada salt.

- **Adreça origen (32 bits)**: indica l'adreça origen del paquet.

Vegeu també

Es pot trobar més informació sobre l'adreçament en el subapartat "Adreçament IPv4".

- **Adreça destinació (32 bits)**: on va dirigit el paquet.

- **IP options**: aquest camp és el que fa que la capçalera IP pugui ser variable en mida. Les opcions, que normalment no s'utilitzen, permeten ampliar les funcionalitats de la capçalera IP. Tot i no fer-se servir gairebé mai, el fet de comprovar-ne l'existència a cada encaminador fa baixar molt el rendiment del protocol IPv4. Per això durant el disseny de la versió 6 del protocol es va canviar la manera d'implementar aquestes opcions.

- **Padding**: per motius d'eficiència les dades han de començar en una posició múltiple de 4 octets; per tant, en el cas que algunes opcions introdueixin una desalineació, el caràcter de farciment, que normalment són tot zeros, alinea a la paraula del camp següent.

- **Data (payload)**: les dades del datagrama que es passaran al nivell de transport, o sigui, la informació que realment es vol transmetre.

Fragmentació IP

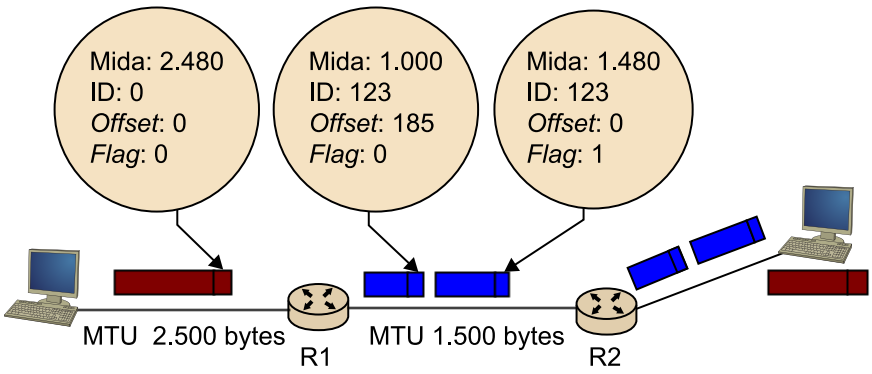
Un dels punts més crítics a l'hora de dissenyar el protocol IP va ser la necessitat d'introduir la fragmentació. La fragmentació IP és necessària perquè no totes les xarxes, ni tots els protocols d'enllaç de dades, poden transportar paquets de mida arbitrària. En general, la mida màxima estarà delimitada depenent de la tecnologia de xarxa utilitzada. Per tant, a causa de la diversitat de tecnologies que coexisteixen a Internet actualment, ens podem trobar casos en què la mida màxima de trama permesa[3] sigui menor en algun encaminador dins del camí per seguir dels datagrames, i això força IP a dividir la trama en fragments més menuts que puguin ser transmesos. Un exemple pot ser Ethernet, que permet trames de mida màxima de 1.500 bytes, mentre que una tecnologia com Asynchronous Transfer Mode (ATM) en general té el màxim a 9.180 bytes. Cal notar que quan es fragmenta un datagrama IP, cada fragment ha de ser autocontingut, i ha de poder ser acoblat en la destinació final (fer-ho els encaminadors intermedis representaria una pèrdua de rendiment considerable), per la qual cosa només dividir el datagrama no és suficient, i és necessari fer-hi algun tipus de procés.

[3] En anglès, *maximum transfer unit* (MTU).

Quan s'ha de dividir un datagrama IP, primer es replica la capçalera IP per a cada fragment, i tot seguit s'actualitzen els camps de la capçalera: *identification*, *flag* i *fragmentation offset*. Així tots els fragments pertanyents al mateix datagrama tindran el mateix identificador, cada fragment contindrà el desplaçament i finalment la indicació, que serà 1 si hi ha més paquets o 0 si és el darrer. Per restriccions amb la implementació i per reduir el nombre de bits que es fan servir per a emmagatzemar aquest desplaçament es va decidir fer-ho amb múltiples de 8 bytes, i així un desplaçament de 64 bytes –o sigui, que el fragment IP conté des del byte 65 del datagrama original– es representarà amb un 8 en el camp *fragmentation offset* (ja que 8 × 8 = 64).

Exemple de fragmentació

A la figura següent es pot veure un cas en què un equip envia un paquet de mida MTU = 2.500 bytes. Això vol dir que el paquet tindrà 2.480 bytes d'informació útil i 20 de capçalera. A l'hora de fragmentar, es generen dos paquets diferents, un de 1.480 + 20 i un altre de 1.000 + 20. Com es pot veure, la mida útil no canvia, però pel fet de tenir dos paquets diferents estem replicant la capçalera. El valor de l'identificador està determinat per un comptador intern en l'encaminador que fragmenta, i el desplaçament (*offset*) per al primer fragment és 0, i la indicació, 1, i això indica que encara hi ha més fragments; per al segon fragment el desplaçament conté un 185, ja que s'especifica amb grups de 8 bytes, i un 0 com a indicació, ja que es tracta del darrer fragment del datagrama original. Quan el datagrama arribi a la seva destinació final serà acoblat i passat als nivells superiors de manera transparent.



Adreçament IPv4

Els protocols de xarxa necessiten disposar d'una adreça única que permeti identificar tots els nodes de la xarxa. En el cas d'IPv4, tal com es pot deduir de la capçalera IPv4, la màxima quantitat d'adreces disponibles és molt gran: 2^{32} (4.294.967.296). Per a simplificar l'escriptura es divideixen els 32 bits en 4 blocs de 8 bits cada un; a més, en comptes d'utilitzar la representació binària, que és poc llegible, en la pràctica una adreça IP s'escriu en notació decimal separada per punts: una adreça estarà formada per 4 blocs de nombres entre 0 i $2^8 - 1$ (255).

Representació binària i decimal d'una adreça IP

10001111 00101101 00000001 00010111
143 · 45 · 1 · 23

A més, tenir un nombre tan gran d'adreces representa un enorme problema de gestió, i per això es va proposar un sistema d'assignació d'adreces jeràrquic. A Internet les adreces IP estan compostes per dues parts: la part de xarxa i la part de l'equip, que s'utilitza per a poder estructurar les adreces i organitzar-les per zones administratives.

La part de xarxa està formada pels bits superiors de l'adreça IP i indica a quina xarxa pertanyen un conjunt d'equips, o el que és el mateix, qui és l'encaminador de sortida del conjunt d'equips. Per contra, la part de l'equip són els bits inferiors de l'adreça IP, i identifiquen l'equip dins de la seva xarxa.

Inicialment aquesta divisió amb xarxes es va fer per mitjà de classes; concretament es van definir 5 classes diferents (A, B, C, D i E), tal com mostra la taula següent:

Mancança d'adreces IP

El repartiment de classes A és un dels causants de la forta mancança d'adreces IP actualment.

- Les **adreces de classe A** són les destinades a empreses molt grans, com per exemple IBM o grans operadores americanes com AT&T WorldNet Services, i proporcionen accés a 2^{24} (16.777.216) equips per xarxa, en què 8 bits estan destinats a identificar la xarxa i la resta, fins als 32, s'utilitzen per als equips finals. D'adreces de classe A n'hi ha un total de 2^7 (255).

- Les **adreces de classe B** són les que es donen a grans entitats, universitats i alguns proveïdors d'Internet. Permeten repartir 2^{16} (65.536) equips per xarxa, i hi ha un total de 2^{14} (16.384) adreces de classe B.

- En el cas de les **adreces de classe C**, són les destinades a mitjanes empreses amb forta presència a Internet, i en aquest cas es disposa de 2^8 (256) adreces, amb un total de 2^{21} (2.097.152) adreces de tipus C per repartir.

- Pel que fa a les **adreces de classe D**, es consideren un tipus de classe especial, anomenades *classes de multidestinació*, que serveixen per a enviar trànsit anomenat *punt multipunt*.

- Finalment les **adreces de classe E** estan reservades per a un ús futur.

Divisió de xarxes per classes

Classe	Bits inicials	Bits xarxa	Rang xarxa
A	0	7 (+1)	1.0.0.0-127.0.0.0
B	1 0	14 (+2)	128.0.0.0-191.255.0.0
C	1 1 0	21 (+3)	192.0.0.0-223.255.255.0
D	1 1 1 0	-	224.0.0.0-239.0.0.0
E	1 1 1 1 0	-	240.0.0.0-255.0.0.0

Adreces de propòsit específic

A part de la divisió en xarxes també es van destinar una sèrie d'adreces de propòsit específic per a casos especials:

- **Adreces d'amfitrió**. Indiquen un equip dins de la xarxa actual i tenen la forma 0.**host**, en què *host* és la part de l'equip de la xarxa actual, o sigui que la part de l'adreça de xarxa és tot 0.

- **Adreces de xarxa**. Fan referència a la xarxa però no als equips que hi ha a dins; les adreces de xarxa són de la forma **xarxa**.0 per a adreces de classe C, **xarxa**.0.0 per a la classe B i **xarxa**.0.0.0 per la classe A, o el que és el mateix, que l'adreça de l'equip de xarxa està tota a 0. Hi ha un cas especial, que és l'adreça 0.0.0.0 que indica "aquest amfitrió" d'"aquesta xarxa", tot i que no sempre s'implementa en els sistemes operatius actuals.

- **Adreces de difusió**. Indiquen tots els equips d'una xarxa concreta. L'adreça es representa amb **xarxa**.255 per a adreces de classe C. Anàlogament al cas de les adreces de xarxa les de classe B seran **xarxa**.255.255 i les de classe A **xarxa**.255.255.255, o sigui que l'adreça de l'equip de xarxa és tot 1. Sempre que es rebi un datagrama a l'adreça de difusió tots els equips hi han de respondre.

Les adreces de difusió tenen una adreça especial a la seva vegada, que és la 255.255.255.255, la que fa referència a tota la xarxa (Internet). Aleshores si algú enviés un datagrama a l'adreça 255.255.255.255 tota la Internet hauria de respondre. Com això provocaria greus problemes d'escalabilitat i excés de trànsit, no hi ha cap encaminador que reenviï trànsit de difusió per les seves interfícies. El trànsit de difusió sempre es quedarà a la xarxa que l'ha emès.

Activitat

Donada l'adreça IP 120.1.32.54, indiqueu quina és l'adreça de xarxa, l'adreça de l'amfitrió i l'adreça de difusió de la xarxa.

Solució

L'adreça 120.1.32.54 forma part de les adreces de classe A; per tant, l'adreça de xarxa serà la 120.0.0.0, la de l'amfitrió la 0.1.32.54 i la de difusió seria la 120.255.255.255.

- **Adreces de *loopback***. Són des de la 127.0.0.0 fins a la 127.0.0.255, i són les que utilitzen internament els equips. Quan un equip arrenca, automàticament crea una interfície virtual (interfície de *loopback*) per a ús intern del sistema operatiu); normalment només s'utilitza la 127.0.0.1.

- **Adreces privades**. Són les utilitzades per xarxes locals internes que no surten a Internet. La llista amb tots els rangs privats es pot trobar a la taula següent. Com es pot observar, es poden configurar internament diversos rangs d'adreces privades, i la seva utilitat és evitar col·lisions en assignacions d'adreces en configuracions internes amb altres nodes d'altres xarxes de l'exterior. Una altra funcionalitat és permetre assignar més adreces a les nostres xarxes que no pas IP públiques assignades pels operadors.

Llista de rangs d'adreces privades

Classe	Rang xarxa	Nombre de subxarxes
A	10.0.0.0-10.255.255.255	1
B	172.16.0.0-172.31.255.255	16
C	192.168.0.0-192.168.255.255	255

Network address translation

El problema principal de les adreces privades és que no poden accedir a Internet directament, els encaminadors mai no enviaran a Internet el trànsit originat en adreces privades

o amb destinació a aquestes, ja que el salt següent no sabria com encaminar-lo. Per tal d'evitar aquesta limitació, i permetre la transferència de dades entre adreces privades i públiques, els encaminadors inclouen una tècnica anomenada *network address translation* (NAT).

Network address translation (NAT)

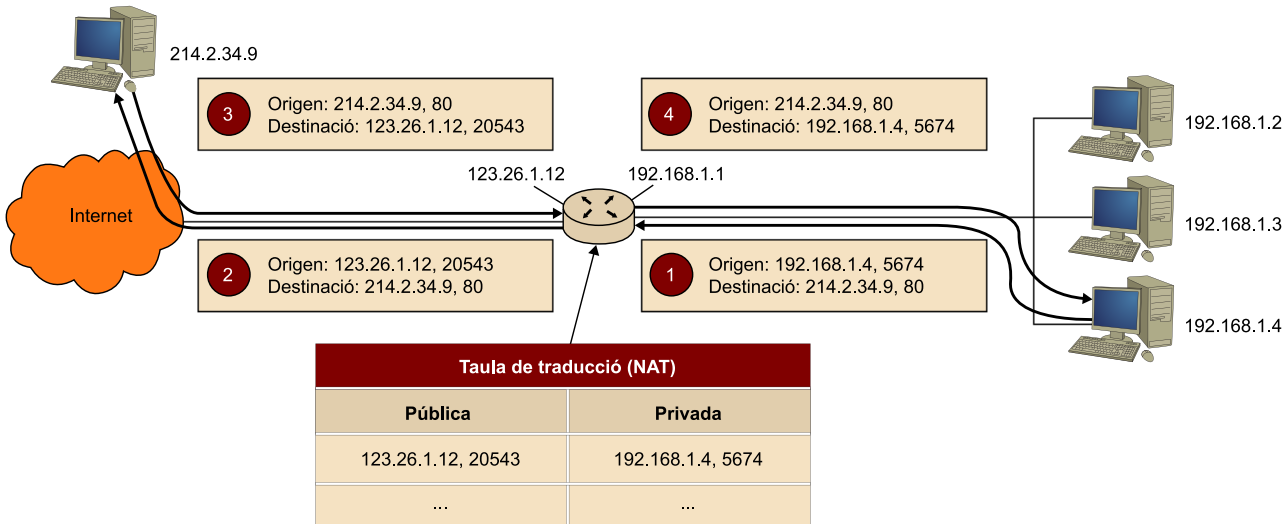
Referència bibliogràfica

NAT està definit en detall en els documents RFC-2663 i RFC-3022.

Pel que es pot deduir del que s'ha vist fins ara, cada equip d'una xarxa IPv4 ha de disposar d'una IP pública per a poder accedir a la xarxa. Un dels problemes principals que es troben quan es demanen IP a les operadores és que generalment l'usuari (o l'empresa) té més equips que no IP assignades. Un exemple d'això és un usuari amb connexió ADSL, que rep una sola IP per part de la companyia telefònica, mentre que molts cops l'usuari disposa de diversos equips, com ara el PC de sobretaula, el portàtil, la PDA, etc. Per a permetre que tots els equips es puguin connectar a la xarxa al mateix temps hi ha dues opcions: demanar més IP (solució difícil i cara) o bé utilitzar adreces IP privades, i configurar l'encaminador perquè faci la conversió des de l'adreça IP privada a la IP pública disponible. Això es pot aconseguir mitjançant el que es coneix com a *NAT*.

NAT és una taula de traducció que s'utilitza de la manera següent. Si, per exemple, un client amb adreça privada vol establir una connexió amb un equip que té una adreça IP pública (punt 1 de la figura següent) –per exemple un servidor–, el client enviarà el paquet cap a l'encaminador de la seva xarxa. Aquest encaminador tindrà configurada una taula de traducció, en què transformarà la IP origen del datagrama a una IP pública que tingui reservada a tal efecte. Per a completar la traducció, l'encaminador mapparà el port origen (de la capa de transport) a un port origen nou assignat per l'encaminador. El punt 2 de la figura en mostra un exemple, en què l'encaminador transforma la IP origen (192.168.1.4) i el port origen (5.674) de l'equip en la IP pública de l'encaminador (123.26.1.12) i un port assignat dinàmicament (20.543 en l'exemple). L'estació destinació veu un datagrama com si hagués estat enviat per l'encaminador, al qual respon de la manera habitual usant TCP/IP. Finalment l'encaminador en rebre la resposta mira la taula de traducció i desfà el canvi per a enviar el paquet final a l'estació origen. Si l'entrada no hagués estat a la taula l'encaminador hauria assumit que el paquet anava realment dirigit a ell.

Exemple de xarxa amb NAT



Aquest mecanisme és molt útil per a estalviar l'ús d'adreces públiques, tot i tenir una sèrie d'inconvenients que no el fan usable en segons quins entorns. Primer hi ha protocols d'aplicació (per exemple FTP), que incrusten la IP del client dins del datagrama; aquesta IP és usada pel servidor per a establir una nova connexió (per exemple, el cas de l'FTP actiu), i com el client incrusta la IP privada això impedeix que es pugui establir la connexió.

Un altre problema important és que totes les connexions s'han d'iniciar des de l'equip amb IP privada, ja que l'encaminador ha d'establir l'entrada a la taula de traducció abans de poder enviar informació cap a l'equip amb IP privada, cosa que implica que normalment no es poden tenir servidors amb IP privades. Cal dir, però, que això es pot solucionar amb una tècnica anomenada *port address translation* (PAT), en què l'encaminador té configurat de manera estàtica un mapatge pel qual quan arriba un datagrama a un port concret, automàticament reenvia el paquet cap a l'equip amb IP privada que estigui configurat. En segons quins entorns el PAT es coneix també com a *destination NAT* (DNAT) o fins i tot com a *port forwarding*, però la idea de fons és la mateixa.

Classless inter-domain routing

Un cop definides les diferents classes de xarxes es va veure que aquesta solució era clarament insuficient, ja que igualment forçava a les operadores grans i mitjanes (amb classes A i B) a gestionar des d'un sol equip un nombre d'adreces massa gran, i per això es va proposar el *classless inter-domain routing* (CIDR).

El CIDR proposa un mecanisme més flexible per a poder subdividir les nostres xarxes. El CIDR no substitueix la divisió per classes, que continuen essent les unitats bàsiques d'assignació d'adreces; per contra el CIDR ens permet dividir les adreces assignades en subxarxes més petites i manejables.

Així amb CIDR la separació entre l'equip i la xarxa s'aconsegueix gràcies a una màscara. Aquesta màscara té la forma d'una adreça IP, que emmascara els bits d'una adreça normal per a poder distingir l'equip i la xarxa de manera senzilla.

Exemple de màscara

Una màscara de 255.255.255.0 permet separar l'adreça de xarxa amb la de l'equip final fent AND amb l'adreça IP; així:

143	. 45	. 1	. 23
255	. 255	. 255	. 0
143	. 45	. 1	. 0

D'on es pot extreure l'adreça de la subxarxa (els uns de la màscara – 143.45.1) i l'adreça de l'equip (els zeros de la màscara – 23). En aquest cas la xarxa constarà de 2^8 IP vàlides, com una classe C, de les quals $2^8 - 2$ seran assignables a equips; cal recordar que les adreces especials de xarxa i de difusió no són assignables (143.45.1.0 i 143.45.1.255, respectivament). La representació d'aquesta subxarxa es fa amb la nomenclatura següent: 143.45.1.23/255.255.255.0.

Com es pot observar això ens dóna un nivell més fi de divisió que ens simplificarà molt la gestió interna de xarxes; si una entitat disposa d'una classe B (146.43.0.0), internament l'entitat pot decidir subdividir les 65.536 adreces en diverses subxarxes, per exemple amb 256 subxarxes de 256 IP cada una: de la 146.43.0.0/255.255.255.0 a la 146.43.255.0/255.255.255.0. Noteu que els valors de 255 i 0 per a l'adreça de xarxa són correctes, i no representen adreces de difusió i de xarxa, respectivament. Això ho podem saber gràcies a la màscara.

Una restricció no escrita però generalment adoptada a l'hora de definir les màscares és que tots els uns de la màscara han de ser consecutius.

Màscares correctes i incorrectes

Màscares com 255.145.0.0 es consideren invàlides, ja que traduïdes a format binari seria:

11111111 10010001 00000000 00000000

Mentre que altres com 255.255.128.0 són totalment correctes, ja que en format binari resulta:

11111111 11111111 11111110 00000000

Aquí tots els uns són consecutius, tot i no estar alineats en l'octet.

Aquesta restricció dels uns consecutius ens permet simplificar la representació de la màscara a un format més compacte, i així una altra forma d'indicar la separació entre la xarxa i l'equip és per mitjà d'un format que indica quants bits representen la xarxa; per exemple, 143.45.1.23/24 indica que la màquina 143.45.1.23 pertany a la xarxa 143.45.1.0/255.255.255.0, o el que és el mateix, que té 24 bits per a l'adreça de xarxa i 8 per a la dels equips.

D'altra banda, gràcies a la classificació per subxarxes els encaminadors tenen la feina més fàcil, ja que per a poder decidir la ruta que ha de prendre qualsevol datagrama n'hi ha prou de mirar la xarxa de destinació, i no és necessari comprovar tota l'adreça IP. L'adreça IP sencera, idealment, només la mirarà l'últim encaminador de la cadena, o sigui, el que estigui dins de la mateixa subxarxa a la qual pertanyi aquella destinació. Per a implementar aquest mecanisme els encaminadors basen la decisió d'encaminament amb una política anomenada *longest prefix match*, que significa que de totes les rutes possibles, sempre s'agafa la que té més bits coincidents amb la destinació del paquet.

Activitat

Indiqueu de les subxarxes i IP següents quantes IP assignables pot contenir la xarxa, i expliqueu-ne el significat.

Adreça	IP assignables	Explicació
147.83.32.0/24		
1.23.167.23/32		
1.23.167.0/32		
147.83.32.0/16		

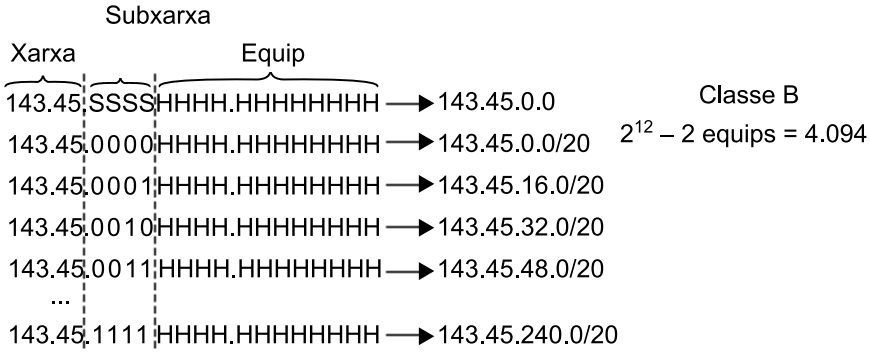
Solució

Adreça	IP assignables	Explicació
147.83.32.0/24	254	És una adreça de xarxa en què tenim 8 bits per als equips; sabent que l'adreça de difusió i la de xarxa no són assignables, acabem amb el total de $2^8 - 2$ adreces per assignar.
1.23.167.23/32	1	Adreça amb una subxarxa amb només un equip. No és útil en un cas real però és correcta.
1.23.167.0/32	1	Com no hi ha part d'equip, tot és de xarxa, i el fet que l'últim octet sigui 0 no fa que la IP sigui una adreça de xarxa genèrica sinó una d'específica, com el cas anterior.
147.83.32.0/16	1	Fa referència a l'equip 32.0 de la xarxa de classe B 147.83.0.0. Cal notar que no és una adreça de xarxa, ja que no tots els bits de fora de la màscara són 0, i així, es tracta com una adreça d'equip.

Activitat

De l'adreça de classe B 143.45.0.0/16, indiqueu quines subxarxes /20 es poden crear i quants equips contenen cada una.

Solució

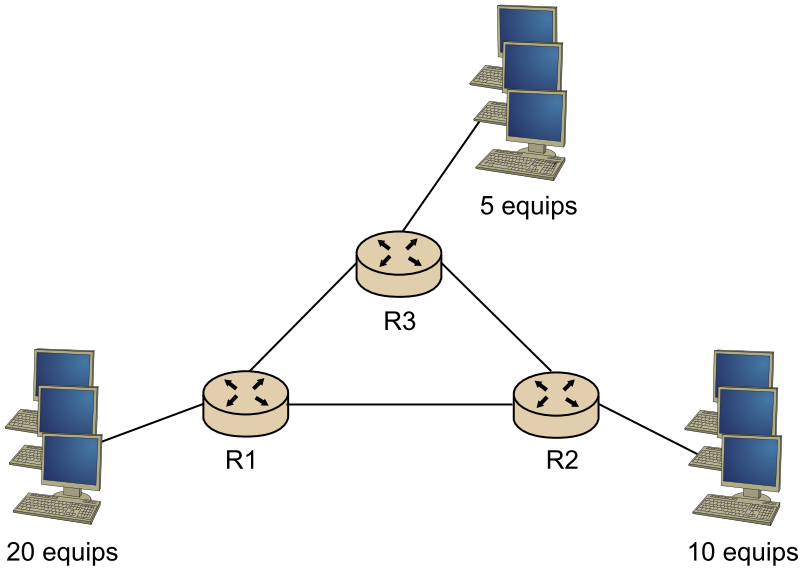


El CIDR generalment s'utilitza en conjunció amb la màscara de subxarxa de mida variable[4], tècnica que té per objectiu optimitzar la utilització de les adreces IP per mitjà d'una assignació intel·ligent de les màscares de xarxa. Aquesta assignació ara es farà tenint en compte el nombre de màquines de cada subxarxa, i s'assignaran màscares de mida ajustada a les necessitats particulars de cada una. VLSM es pot veure com la creació de subxarxes de les subxarxes.

[4]En anglès, *variable length subnet mask* (VLSM).

Activitat

1) Donada la xarxa de la figura, ens proporcionen el rang d'adreces 147.83.85.0/24. Es demana que s'assignin rangs d'adreces a totes les subxarxes i als enllaços entre els encaminadors.



2) Quin problema té l'assignació d'adreces feta a l'exercici anterior?

Solució

1) L'assignació d'adreces es pot fer seguint la política següent:

- Els 20 equips més l'encaminador necessiten un total de 5 bits (2^5 = 32).
- Els 10 + 1 equips en tenen suficient amb 4 bits (2^4 = 16).
- Els 5 + 1 equips en necessiten 3 (2^3 = 8), amb la qual cosa aquesta subxarxa no podrà créixer més.

- Finalment els enllaços punt a punt necessitaran 2 bits, ja que necessitem espai per a les adreces de difusió i de xarxa.

En resum, ens faran falta un prefix /27, un /28, un /29 i 3 prefixos /30, respectivament. D'aquesta manera una assignació possible seria:

- Els 20 equips poden usar la 147.83.85.0/27, en què el darrer byte seria 000XXXXX, amb adreça de xarxa 147.83.85.0 i adreça de difusió 147.83.85.31.
- Els 10 equips disposaran de la 147.83.85.32/28, en què el darrer byte serà 0010XXXX, amb adreça de xarxa 147.83.85.32 i adreça de difusió 147.83.85.47.
- En el cas dels 5 equips utilitzarem la subxarxa 147.83.85.48/29, en què el darrer byte serà 00110XXX, amb adreça de xarxa 147.83.85.48 i adreça de difusió 147.83.85.55.
- Finalment els tres prefixos /30 (dels enllaços entre els encaminadors) es poden dividir amb el darrer byte 001110XX, 001111XX i 010000XX, respectivament. O sigui, 147.83.85.56/30, 147.83.85.60/30 i 147.83.85.64/30, amb adreces de xarxa 147.83.85.56, 147.83.85.60 i 147.83.85.64, i amb adreces de difusió 147.83.85.59, 147.83.85.63 i 147.83.85.67.

2) El problema d'aquesta assignació està determinat pel fet que s'ha ajustat massa el nombre de bits per a cada subxarxa, i en el cas que creixi el nombre d'equips (especialment en la subxarxa de 5 equips) ens tocaria redimensionar la xarxa de nou, amb el cost que això representa.

Tipus de datagrames IP

IPv4 especifica tres tipus de trànsit clarament diferenciats dins de la xarxa: unidestinació, difusió i multidestinació.

El **trànsit unidestinació** és el més comú; la comunicació està formada per dos interlocutors que s'intercanvien informació, i sovint aquestes connexions són des d'un client cap a un servidor, que a la vegada pot tenir connexions unidestinació cap a altres clients.

El **trànsit de difusió** es basa a enviar la informació a tots els equips presents en una subxarxa. Com ja hem vist anteriorment, això es pot aconseguir enviant un paquet a una adreça que sigui l'adreça de xarxa i tot uns a l'adreça de l'equip final; per exemple, per a la xarxa 126.76.31.0/24, l'adreça de difusió seria 126.76.31.255.

Requisits del trànsit de difusió

Cal notar però, que enviar trànsit de difusió normalment requereix algun privilegi en la xarxa (ser administrador); a més, els encaminadors en general no propaguen aquest tipus de trànsit per a evitar problemes de seguretat, com per exemple atacs del tipus *denial of service* (DoS).

Finalment, el cas del **trànsit multidestinació** es basa en el paradigma d'enviar informació des d'un sol origen cap a moltes destinacions a la vegada; la base del trànsit multidestinació és que l'emissor no ha de tenir necessàriament coneixement de qui seran els seus receptors (en contra de la política d'unidestinació, que requereix conèixer els interlocutors). Això s'aconsegueix per mitjà del que es coneix com a *grups de multidestinació*. Com s'ha vist anteriorment, la Internet Assigned Numbers Authority (IANA) ha reservat les adreces de tipus D a multidestinació, i són les que van del rang 224.0.0.0 al 239.0.0.0. D'aquest grup d'adreces n'hi ha unes quantes de reservades a grups multidestinació coneguts com a permanents.

Així, si una estació concreta està interessada a rebre un contingut multidestinació el que farà serà subscriure's al servei mitjançant el protocol Internet Group Management Protocol (IGMP), que especifica el format del paquet que s'ha de generar per a poder registrar-se en un grup i poder rebre'n el contingut. IGMP suporta dos tipus de paquets: els de pregunta i els de resposta. Normalment els de pregunta són uns paquets dirigits a tots els equips, i els que tenen sessions multidestinació actives responen, i així els encaminadors (que han de tenir suport de multidestinació) poden construir el que es coneix com l'arbre multidestinació, per a encaminar els paquets cap a les seves destinacions.

L'avantatge principal de la multidestinació és que amb la informació que s'envia, en comptes de replicar-se des de l'origen un cop per a cada destinació, es forma un arbre, de manera que es minimitza el nombre de còpies.

Activitat

Un servidor de xat té en un moment donat un total de 80 clients connectats arreu del món. Indiqueu quin nombre i de quin tipus són les connexions que té obertes aquest servidor.

Solució

Atès que el xat és un protocol que utilitza TCP/IP, i que els clients, tot i parlar entre ells, passen sempre pel sevidor, és tracta del típic escenari amb 80 connexions unidestinació entre els 80 clients i el servidor.

Activitat

Un administrador de la xarxa 147.83.0.0/16 vol enviar un paquet de difusió a la subxarxa 147.83.20.0/24. Indiqueu quina adreça de destinació tindria el paquet, quants paquets es generarien i a quantes màquines com a màxim podria arribar.

Solució

Atès que la subxarxa a la qual es vol enviar la difusió té 8 bits, això implica que es generaria un sol paquet amb adreça destinació 147.83.20.255, i que el rebrien com a màxim 255 – 2 = 253 estacions, ja que l'adreça 147.83.20.0 i la 147.83.20.255 estan reservades per a l'adreça de xarxa i la de difusió, respectivament.

El futur d'IPv4

Quan es va dissenyar IPv4 es creia que el seu gran nombre d'adreces IP (2^{32}) seria suficient per a poder aguantar el gran creixement que s'esperava d'una xarxa com Internet. Cal recordar que Internet va entrar en funcionament el 1969 amb el nom d'ARPANET, un projecte subvencionat pel Departament de Defensa dels Estats Units. Això va provocar que, quan Internet es va desplegar al cap d'uns anys a la xarxa comercial, el repartiment d'adreces no es va fer de manera equitativa; les grans empreses nord-americanes es van poder adjudicar una gran quantitat d'adreces de classe A, i van deixar països com la Xina i d'altres que s'han desenvolupament posteriorment amb moltes menys adreces de les necessàries. Com a referència, els EUA tenen vora 1.500 milions

d'adreces assignades, mentre que la Xina, amb una població molt més nombrosa, només en disposa aproximadament de 200 milions. Per tenir una idea, Espanya en té assignades actualment vora 22 milions.

Amb aquest paradigma, molt aviat es veu que amb l'actual política per al repartiment d'adreces, en molt poc temps no quedaran adreces IPv4 disponibles per assignar, fet que implicarà inevitablement que Internet no podrà créixer més. Per tal de minimitzar aquest problema es va dissenyar el NAT, com ja hem vist, que permet utilitzar adreces privades per a accedir a la xarxa amb una sola IP pública. Actualment, països com la Xina o l'Índia estan fent un ús intensiu del NAT per la manca d'adreces disponibles.

Com aquesta solució no és escalable i comporta una sèrie molt important de problemes als proveïdors de serveis, es va arribar a la conclusió que els 32 bits d'adreçament del protocol IPv4 eren insuficients, i per això es va dissenyar el protocol Ipv6, com veurem a continuació.

Implantació del protocol IPv6

Per motius econòmics IPv6 encara no ha estat implantat, i si la demanda d'adreces IPv4 segueix al ritme actual, es preveu que la IANA assignarà el darrer rang d'adreces IPv4 a mitjan 2011, i que les autoritats regionals esgotaran les que tenen pendents per assignar el 2012, fet que de ben segur forçarà molts països a adoptar prematurament IPv6. En aquest sentit, països com el Japó, la Xina, l'Índia i alguns d'Amèrica del Sud ja han adoptat el protocol i utilitzen algunes tècniques que permeten la interoperabilitat dels dos protocols.

2.3.2. IPv6

La mancança d'adreces IPv4 va incentivar el disseny d'un nou protocol de xarxa, IPv6. Actualment IPv6 està totalment desenvolupat, tot i que encara no és possible utilitzar-lo dins la xarxa comercial, ja que els operadors encara no han preparat els seus equips i tampoc no han fet el repartiment d'adreces als seus usuaris. Això i la dificultat d'implantar progressivament aquesta nova versió en substitució de l'anterior és el que n'està retardant la incorporació a escala comercial.

Aquest subapartat descriu breument aquest protocol, se'n remarquen les diferències amb la versió anterior, i les novetats que incorpora. Per acabar el subapartat es descriuen els principals problemes que hi ha per a la migració d'IPv4 a IPv6.

Motivació

Inicialment, a l'hora de dissenyar el protocol es va pensar que no era necessari crear un protocol sencer, ja que només fer una adaptació d'IPv4 hauria de ser suficient. Però ben aviat es va veure que per a poder gaudir de bones optimitzacions en comparació de la versió anterior caldrien bastants més canvis. Així es va optar per un disseny que té poc en comú amb la versió anterior.

El motiu principal que va portar a plantejar-se una nova versió del protocol era el limitat rang d'adreces que permet IPv4, que encara que pugui semblar molt elevat, es va veure que seria clarament insuficient per la demanda del mercat en un futur. Sobretot, l'aparició els darrers anys d'una gran quantitat de dispositius mòbils que volen formar part de la gran xarxa que és Internet, ha provocat ràpidament que els 32 bits d'adreçament IPv4 siguin insuficients; tant és així, que si tots aquests dispositius es volguessin connectar de manera

simultània a la xarxa probablement els operadors tindrien problemes per la manca d'adreces IPv4. Per veure aquest problema només cal pensar en quants telèfons mòbils hi ha actualment només a l'Estat espanyol –vora uns 44 milions–, i el nombre d'IP que hi ha assignades actualment al país és de vora 22 milions. Això sense considerar els usuaris que es connecten des de les seves llars. Es podria pensar que el problema es podria minimitzar amb la utilització de NAT, però a la llarga això pot representar un greu problema de rendiment en els encaminadors per haver de mantenir les taules de traducció d'adreces de milions de connexions a la vegada. A sobre, com més va hi ha més petites i mitjanes empreses que volen oferir als seus clients una sèrie de serveis que necessiten una connexió permanent a la xarxa, amb la despesa conseqüent d'adreces i la impossibilitat d'utilitzar el NAT massivament. IPv6 soluciona aquest problema proposant un camp d'adreces de 128 bits.

Capçalera IPv6

La capçalera IPv6 té una longitud fixa de 40 octets (tal com es pot veure a la figura següent), i consta dels camps següents:

- **Version (4 bits)**: indica la versió del protocol que conté el paquet. Aquest camp té el mateix significat que el de la versió IPv4, però ara amb el valor `0x06`.

- **Traffic class (8 bits)**: aquest camp classifica un paquet dins d'un tipus de trànsit determinat; conceptualment és l'equivalent al *type of service* (TOS) d'IPv4.

- **Flow label (20 bits)**: serveix per a etiquetar un conjunt de paquets que tinguin les mateixes característiques, servirà per a poder oferir qualitat de servei.

- **Payload length (16 bits)**: longitud del *payload* del paquet, o sigui, el paquet sense la capçalera IP. La mida està representada en octets.

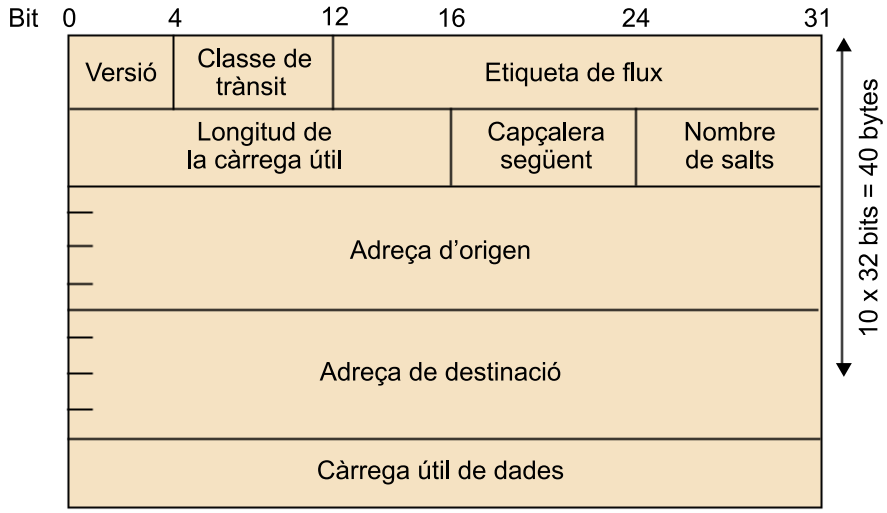
- **Next header (8 bits)**: aquest camp és una gran innovació d'IPv6 respecte a IPv4, atès que permet tenir una capçalera bàsica de mida fixa. Aquest camp indica la posició en què es pot trobar la capçalera següent, i així s'estalvia temps de procés als encaminadors intermedis, en no haver-hi opcions.

- **Hop limit (8 bits)**: aquest camp és equivalent al TTL d'IPv4 però directament compta *hops* i no temps.

- **Source address (128 bits)**: adreça de l'amfitrió que ha originat el paquet.

- **Destination address (128 bits)**: adreça de l'amfitrió al qual va destinat el paquet.

Capçalera IPv6



De les dades de la capçalera, es pot observar que la diferència més directa que hi ha entre tots protocols és la longitud de les adreces IP, ja que IPv4 té 32 bits i IPv6 passa a tenir-ne 128. Aquest augment amb l'espai d'adreçament permet que el rang d'adreces de la xarxa passi de 2^{32} a 2^{128} adreces possibles.

Com que ara tenim molts més bits per l'adreça, la forma d'especificar adreces IPv6 es fa amb la **notació dels dos punts**. Una adreça IPv6 es representa amb blocs de 16 bits representats en hexadecimal i separats pel símbol ":". Per exemple: 2001:0DB8:0000:0000:0319:8A2E:0370:7348. Una simplificació d'aquesta notació es pot aplicar en el cas que una adreça tingui molts 0 consecutius; la forma abreviada de representar-la és utilitzant "::", i així, la forma compacta de representar l'adreça anterior seria 2001:0DB8::0319:8A2E:0370:7348.

Una altra diferència notable entre IPv4 i IPv6 és la jerarquització de les adreces. L'assignació d'adreces IPv4 es va fer, al seu temps, d'una manera molt anàrquica, atès que no s'esperava que el creixement d'Internet fos tan espectacular. Actualment cada corporació o cada operadora de telefonia té rangs d'adreces molt dispersos i mal dimensionats, que fan extremadament difícil la gestió de les adreces disponibles, l'assignació de les noves i l'encaminament global. Per això IPv6 el que ha fet ha estat jerarquitzar d'una manera més intel·ligent el repartiment de les seves adreces, de manera que cada país, operador o ISP disposa d'un rang concret amb un nombre d'adreces proporcional a la seva utilització possible de la xarxa. Independentment de la millora d'aquesta jerarquia quant a la localització geogràfica, el fet de separar d'aquesta manera les adreces permet assignar-ne de noves d'una manera molt més senzilla que fins ara.

De manera semblant a IPv4, es pot identificar quin tipus d'adreça és només amb el prefix de l'adreça IPv6, com indica la taula següent.

Assignació d'adreces

Prefix	Espai d'assignació
0000::/8	Reservat. Les adreces de *loopback* i les adreces amb integració d'IPv4 surten d'aquest prefix
0100::/8	Reservat
0200::/7	Reservat
0400::/6	Reservat
0800::/5	Reservat
1000::/4	Reservat
2000::/3	Adreça unidestinació global. D'aquí surt el rang d'adreces que es repartiran als usuaris. Hi ha 2^{125} adreces disponibles.
4000::/3	Reservat
6000::/3	Reservat
8000::/3	Reservat
A000::/3	Reservat
C000::/3	Reservat
E000::/4	Reservat
F000::/5	Reservat
F800::/6	Reservat
FC00::/7	Adreça unidestinació local única
FE00::/9	Reservat
FE80::/10	Adreça d'enllaç local unidestinació (*link-local*)
FEC0::/10	Reservat
FF00::/8	Adreces multidestinació

Un fet molt interessant que es va considerar per a fer aquesta assignació d'adreces és que dóna la possibilitat de representar adreces de diverses tecnologies incrustades dins la nova versió del protocol. D'aquesta manera es poden representar adreces IPv4, i fins i tot adreces de maquinari de l'enllaç de dades (com per exemple Ethernet).

El gran avantatge d'inserir altres tipus d'adreces directament a la IPv6 és que tenen un prefix assignat; així, per exemple, per a tenir una adreça Ethernet d'un equip dins d'una IPv6 el prefix LAN és FE80::/10. Per tant, si l'adreça de la targeta Ethernet és 00:90:F5:0C:0F:ED aleshores l'adreça IPv6 queda: FE80::0090:F50C:0FED. Com es pot observar el procés també es pot fer a la inversa, i quan arriba un paquet amb el prefix de xarxa FE80 ja es pot suposar

que es tracta d'una adreça local (*link-local*) i se'n pot extreure l'adreça de maquinari fàcilment. A més, amb aquest mecanisme qualsevol interfície de xarxa es pot configurar de manera automàtica i autònoma.

Cal remarcar que Ipv6, a part de tenir adreces unidestinació, de difusió i de multidestinació com IPv4, afegeix suport per un quart tipus, que són les adreces de servei.

Les adreces de servei són una gran innovació d'IPv6, sobretot perquè aprofiten les adreces unidestinació ja existents. Així una adreça unidestinació esdevé de servei des del moment que una mateixa IPv6 s'assigna a més d'una interfície (incloent-hi equips diferents). La idea al darrere d'aquesta implementació és que respongui les peticions a un servei concret l'estació més propera.

Imaginem dos servidors web amb la mateixa Ipv6, com per exemple 2001:0DB8::0319:8A2E:0370:7348; quan la xarxa rebi un paquet dirigit a aquesta IPv6, l'enviarà a les dues estacions, i la primera que respongui serà la que estigui més a prop de l'equip que fa la petició. Actualment, per complexitats en la implementació aquest tipus d'adreces només s'utilitzen per a encaminadors. Així, una subxarxa pot tenir més d'un encaminador per a sortir a Internet usant el mateix prefix, i cada equip fa servir el que està més proper a l'estació, i s'aconsegueix de manera senzilla un sistema de balanceig de càrrega.

Una altra innovació rellevant que incorpora IPv6 és la utilització molt més intensiva del trànsit multidestinació dins de les xarxes locals; tant és així que els equips, per defecte, escolten adreces multidestinació amb el prefix FF02::1:FF00:0000/104, per tal d'evitar la generació de trànsit de difusió que afecti tots els equips de la subxarxa, que no sempre és desitjable.

Altres millores menors que introdueix aquest protocol són:

- **Mecanisme d'opcions ampliat**: les opcions formen part d'una capçalera col·locada entre la capçalera IP pròpiament dita i la capçalera de la capa de transport. Aquesta manera de posar les opcions permet una gestió més simple de les capçaleres per part dels dispositius que han de tractar el paquet fins que no arriba a la seva destinació, i permet un sistema més simple i flexible.

- **Adreces d'autoconfiguració**: l'assignació dinàmica d'adreces ha estat substancialment millorada respecte al seu predecessor. Un dels motius principals d'això és el fet que es pot afegir l'adreça maquinari dins l'adreça IPv6, i així només amb un prefix donat pel dispositiu d'encaminament més proper i amb l'adreça de maquinari es garanteix una adreça única a escala mundial, sempre que s'utilitzi el prefix assignat per l'operador a l'hora de generar l'adreça.

- **Facilitat per a l'assignació de recursos**: el que amb IPv4 era el *type of service* ara s'anomena *traffic class*, tot i que ara es té la possibilitat de marcar

fluxos individuals, cosa que dóna molta més flexibilitat a l'hora de marcar trànsit prioritari.

- **Capacitats de seguretat**: com que la seguretat, avui dia, és un tema molt important IPv6 inclou característiques d'autenticació i privacitat. Per defecte IPv6 inclou funcionalitats natives per a la creació de xarxes privades virtuals (VPN, en la sigla en anglès) per mitjà d'IPsec (RFC-4301), protocol de xifratge de les dades en temps real que amb IPv4 era opcional.

Activitat

Tenim un PC amb una targeta Ethernet amb MAC 34:27:A4:6F:AE:53. L'operador li proporciona el prefix 2001:0A54:0039::/48. Indiqueu l'adreça *link-local* i l'adreça d'autoconfiguració d'aquest equip.

Solució

L'adreça *link-local* estarà determinada pel prefix *link-local*, i així l'adreça serà FE80::3427:A46F:AE53. Mentre que l'adreça d'autoconfiguració surt del prefix i de la MAC; per tant serà 2001:0A54:0039::3427:A46F:AE53.

Problemes de la migració a IPv6

Un dels motius principals pels quals encara és treballa amb IPv4 és la dificultat que comporta la migració al nou protocol. La incompatibilitat de les adreces i de les capçaleres de tots dos protocols fan que l'actualització a la nova versió no sigui fàcil. També cal tenir en compte que les aplicacions existents només suporten el sistema d'adreces d'IPv4, i per a acceptar les noves adreces s'ha de canviar el codi de l'aplicació, i també totes les crides al sistema d'accés a la xarxa.

A part del nivell d'aplicació, hi ha un altre problema molt greu. Atesa la gran diversitat de xarxes que formen Internet, hi ha funcionant equips molt diversos, i no tots aquests equips de comunicacions tenen suport per al nou protocol; per tant, s'ha d'actualitzar el sistema operatiu dels encaminadors de la xarxa, amb la despesa econòmica i de temps conseqüent que això representa, cosa que moltes de les empreses no estan disposades a invertir (especialment les grans corporacions nord-americanes, que són les que tenen adreces suficients).

Finalment, a causa de la gran utilització que té actualment Internet, és un gran problema haver de parar totes les xarxes per a fer la migració. Així el problema que es troba és l'enorme despesa econòmica per a les empreses que controlen totes les seves transaccions per mitjà de la xarxa, fet que força a fer la migració de manera progressiva, transparent per als usuaris i sense deixar d'oferir els serveis disponibles en cap moment.

Mecanismes per a assistir la transició

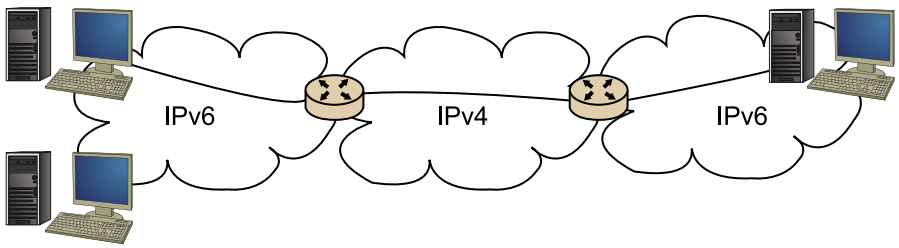
Una migració entre dos protocols quan un s'està utilitzant massivament és extremadament complexa. Com hem vist, les raons normalment són econòmiques, ja que la gran majoria de les aplicacions actualment només suporten IPv4, i migrar-les a la nova versió no sempre és senzill (per exemple, aplicacions bancàries). D'altra banda, tot l'equipament maquinari que forma el tronc de la xarxa, si bé està preparat per a suportar Ipv6, no sempre té la configuració correcta, ni l'assignació d'adreces feta. Amb tot, s'espera que hi hagi una fase de coexistència dels dos protocols. De totes maneres, tot i la coexistència hi ha escenaris que obliguen a dissenyar un pla de migració controlat. Així els diferents mecanismes de transició es poden dividir en dos grans grups: mecanismes bàsics i mecanismes per la interconnexió d'illes.

De mecanismes bàsics se'n poden distingir dos: el conegut com a *dual-stack*, en què els equips utilitzen simultàniament els dos protocols i es connecten amb el que més s'ajusti a les necessitats del moment, i el de tunelització, en què dos equips amb *dual-stack* creen un túnel IPv4 entre ells, i per dins del túnel es comuniquen amb IPv6.

Cal notar que un túnel és aquell mecanisme pel qual s'encapsulen dos protocols de xarxa dins d'un mateix datagrama; així, hi ha dues capçaleres de xarxa consecutives del nivell de xarxa. IPv4 suporta el mecanisme de túnel per mitjà d'un valor especial al camp *protocol*, que es troba a la capçalera.

Pel que fa als mecanismes per a connectar illes, pretenen resoldre el cas en què diverses màquines interconnectades per mitjà d'IPv6 (illa IPv6) es volen connectar amb una altra illa IPv6, però pel camí hi ha una illa IPv4, tal com mostra la figura següent.

Xarxes IPv4 i xarxes IPv6



Aquests mecanismes també estan basats principalment en túnels. Se'n poden distingir els tipus següents:

- **Túnels configurats**: els extrems dels túnels entre les illes es configuren de manera manual entre les dues xarxes. Els extrems han de ser *dual-stack*.

- **Túnels automàtics**: s'utilitzen adreces IPv4 mapades dins d'IPv6 amb el prefix reservat, que és el ::/96, com per exemple, ::195.123.57.93, que l'encaminador converteix a l'adreça IPv4, i a l'altre extrem es torna a con-

vertir a la IPv6. Els extrems no s'adonen del canvi i es poden comunicar amb IPv6 sense problemes.

- **Túnel *broker***: s'utilitza un *broker* (gestor), que indica al client un *script* per a fer el túnel de manera automàtica. El client ha de ser *dual-stack*, ja que la petició al *broker* va amb IPv4, que contesta amb un *script* que permet connectar a un *tunnel-server* (que també és *dual-stack*), que permet connectar-se a la xarxa IPv6.

- **6to4**: s'assigna una adreça IPv4 compatible amb el prefix IPv6 als encaminadors, que fan un túnel. Per exemple, per a la xarxa 2001:d002:0507::/48 l'encaminador tindria l'adreça 208.2.5.7 (que surt de d002:0507). A l'altre extrem es faria l'operació anàloga i s'establiria el túnel.

2.4. Protocols de suport a IP

Tant IPv4 com IPv6 són protocols de xarxa, però tots dos necessiten suport d'altres protocols de la mateixa capa per a poder dur a terme certes funcionalitats que seria complex aconseguir d'una altra manera. Tot i que, ateses les diferències estructurals entre els dos protocols, molts dels serveis són diferents, s'engloben tots en aquest subapartat per simplificar-ne la comprensió, ja que si bé els protocols divergeixen, la seva funcionalitat sovint és molt similar.

2.4.1. Internet Control Message Protocol

IP és un protocol no orientat a connexió que té per objectiu l'enviament d'informació independentment de la tecnologia de nivells inferiors utilitzada. Això proporciona un entorn ideal per a poder comprovar l'estat de la xarxa, o enviar informació de control en cas que hi hagi problemes a la xarxa. Per exemple, quan un datagrama no pot arribar a la seva destinació, o bé si hi ha congestió en un enllaç, o bé si a un paquet se li ha expirat el TTL. Totes aquestes situacions requereixen algun protocol, que treballant a la mateixa capa que IP permeti avisar de manera automàtica de qualsevol d'aquests esdeveniments. Per això es va dissenyar l'Internet Control Message Protocol (ICMP).

L'ICMP és l'encarregat d'enviar missatges de control (i d'error) entre els diferents equips que formen la xarxa.

La taula següent mostra tots els tipus de missatges existents amb ICMP.

Bibliografia

Per a una descripció més completa de tots els tipus de missatges existents amb ICMP es pot consultar el document RFC-792.

Descripció dels diferents missatges ICMP

Missatge	Descripció
Destination unreachable	Indica que no es pot arribar a la destinació. Aquest missatge té un camp de codi que indica si és culpa de la xarxa, de l'equip en particular o bé del port. També distingeix si no es pot arribar a la destinació o bé si la destinació és desconeguda.
Echo request/ Echo reply	Aquests dos missatges són el de petició i el de resposta; quan una estació rep un *echo request*, ha de respondre amb un *echo reply* si està activa. En general és aconsellable que tots els equips responguin aquestes peticions, tot i que per seguretat molts cops es filtren els missatges. L'aplicació per excel·lència que utilitza els *echo request*/*reply* és el `ping`.
Source quench	Aquest paquet serveix per a regular, indica que aquell enllaç està patint congestió. Actualment aquest tipus de paquet no s'utilitza perquè el control es fa normalment en la capa de transport.
Router advertisement	Aquest paquet d'ICMP s'envia a una adreça multidestinació en què tots els encaminadors hi escolten per defecte. Amb aquest missatge és possible descobrir automàticament l'existència de nous encaminadors a la xarxa.
Router discovery	És un paquet complementari amb el de *router advertisement*. Però en aquest cas és un encaminador que acaba d'entrar a una xarxa qui pregunta quins altres encaminadors hi ha a la xarxa.
TTL expired	Quan el TTL d'un paquet arriba a 0 aquest es descarta, i l'encaminador que l'ha descartat genera un paquet TTL *expired* en l'origen del paquet descartat.
IP header bad	En el cas que es detecti un error de suma de verificació en la capçalera d'un paquet IP, aquest es descarta i s'avisa a l'origen amb aquest paquet ICMP.

Els missatges ICMP tenen diverses utilitats, des d'informar errors fins a depurar l'estat de la xarxa. Una de les eines més utilitzades per a poder veure si un equip està connectat a la xarxa és el `ping`. Una altra funcionalitat és possible gràcies al camp TTL de la capçalera IP, que ajuda a fer una altra tasca de depuració mitjançant una eina anomenada `traceroute`; aquesta eina ens permet descobrir els encaminadors intermedis entre l'origen i la destinació dels datagrames. Per a aconseguir saber quins encaminadors travessa un datagrama, el `traceroute` envia paquets IP consecutius amb un TTL d'1, un altre de 2, un altre de 3 i així successivament fins a arribar a la destinació. L'efecte d'això és que el primer encaminador en rebre un paquet amb TTL = 1 el sostreu, i en ser 0, el descarta i envia de tornada un paquet ICMP de *TTL expired*. Ara l'aplicació només cal que miri qui ha enviat aquest paquet (IP origen) per a saber de quin encaminador es tracta. Per descomptat, amb TTL = 2 succeirà el mateix amb el segon encaminador, i després amb el tercer i així successivament fins a la destinació.

2.4.2. Address Resolution Protocol

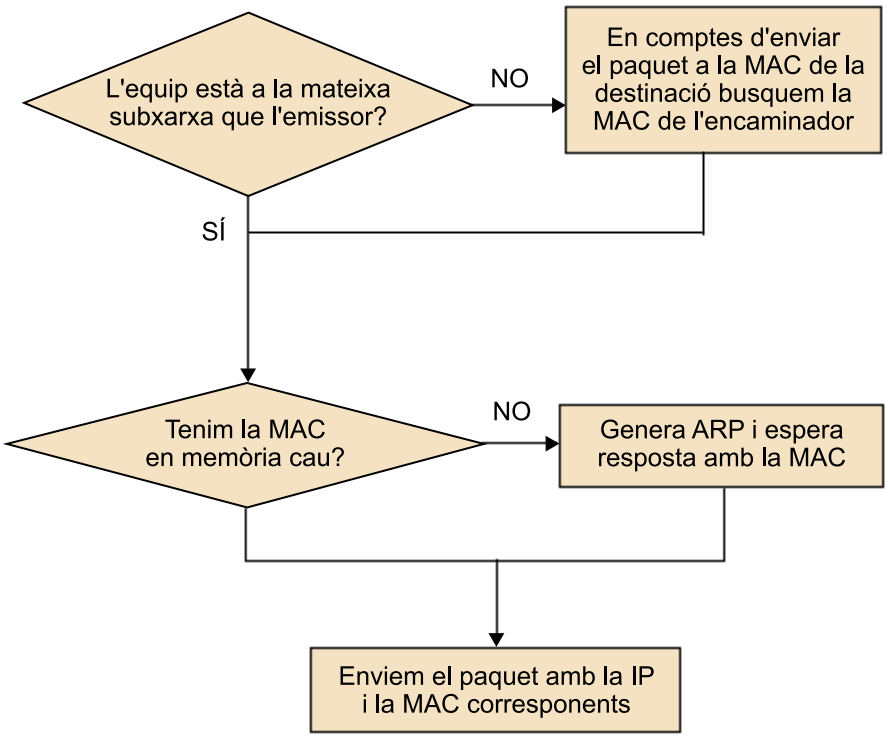
Adreça de nivell d'enllaç

L'adreça de nivell d'enllaç és coneguda com a adreça MAC i està determinada pel dispositiu de xarxa. Cada dispositiu de xarxa té una adreça MAC única.

Per a enviar un datagrama IP a una estació de la mateixa xarxa que l'emissor és necessari descobrir quina adreça del nivell de l'enllaç de dades té aquesta estació, ja que les tecnologies de capes inferiors no entenen què és una adreça IP. Així doncs, perquè IP funcioni necessita interactuar amb les capes inferiors i descobrir de manera automàtica quina és l'adreça d'enllaç de dades a la qual respon un equip per a poder-se intercanviar informació. El protocol Address Resolution Protocol (ARP) va ser dissenyat específicament per a IPv4 i, com veurem més endavant, per a IPv6 tenim el Network Discovery Protocol.

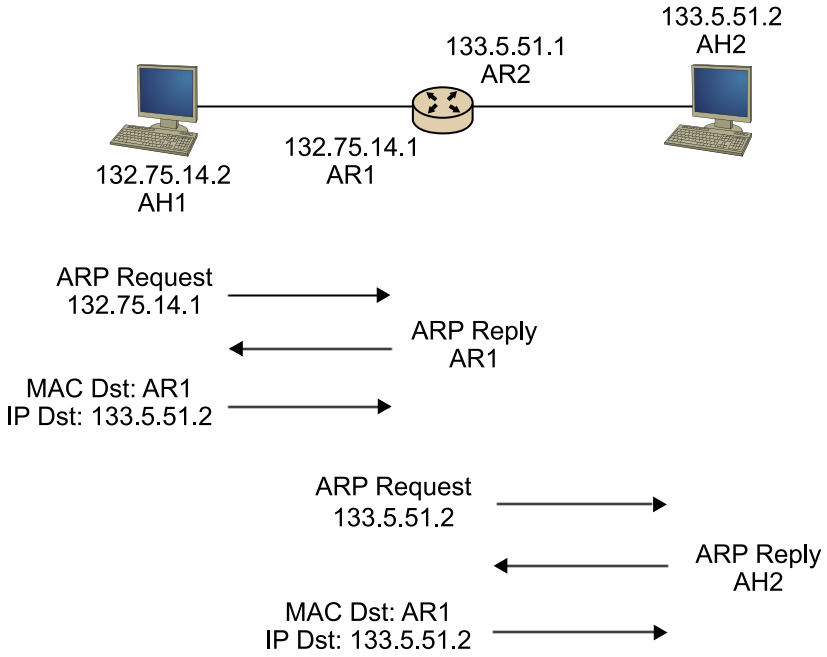
Per a aconseguir el descobriment de l'adreça de maquinari d'un equip a partir de la seva IP ARP se serveix de la funcionalitat que ens proporciona la capa d'enllaç de dades per a enviar paquets de difusió. Aleshores el procés per a aconseguir l'adreça de maquinari (anomenada MAC) i poder enviar el paquet es pot veure al diagrama de la figura següent.

Diagrama de seqüència per a l'enviament d'un paquet IP



El procés d'ARP, il·lustrat a la figura següent amb un exemple, segueix una sèrie de passos per a descobrir l'adreça. Primer s'ha d'aconseguir l'adreça MAC del salt següent, i per això s'envia un paquet ARP a l'adreça de difusió de la nostra xarxa local. Aquest ARP buscarà la IP de destinació o bé la de l'encaminador, depenent de si l'estació forma part de la subxarxa o no. Un cop s'obté l'adreça MAC, es construeix un paquet que té com a adreça MAC de destinació l'obtinguda per ARP i com a IP de destinació l'original (o sigui, en el cas que s'enviï el paquet a l'encaminador aquesta IP serà la de destinació final, no la de l'encaminador).

Exemple de petició per ARP



RARP

Hem vist que l'ARP ens serveix per a poder descobrir quina adreça MAC correspon a una IP, i també hem vist la importància d'aquesta operació. Per completesa ARP té una variant anomenada *RARP* que ens permet fer l'operació inversa, o sigui, des d'una adreça MAC poder esbrinar a quina IP correspon. RARP no és ben bé un protocol de la capa de xarxa, ja que inclou moltes funcionalitats de la capa d'enllaç de dades, però atesa l'estreta relació amb ARP s'acostumen a considerar conjuntament. RARP ja no s'utilitza, ja que hi ha protocols com BOOTP i DHCP que ens ofereixen aquesta i més funcionalitats, com veurem a continuació.

2.4.3. Network Discovery Protocol

ARP és un protocol que va ser dissenyat específicament per a IPv4, i amb els avenços de les xarxes fins avui, ha provat que és insuficient per a segons quins serveis. Per això, amb l'aparició d'IPv6, es va decidir que calia un protocol més complet. Així va aparèixer el Network Discovery Protocol (NDP).

NDP és un protocol que permet descobrir els veïns que hi ha en una xarxa local. El mode d'operació és molt similar al que utilitzàvem amb IPv4: ara s'envien *neighbour solicitations* i es reben *neighbour advertisements*. Però la gran diferència amb ARP és que s'utilitza trànsit multidestinació en comptes de difusió. I què NDP forma part d'un protocol més gran anomenat *ICMPv6*, que és l'extensió a IPv6 del protocol ICMP.

Quan un node IPv6 es dóna d'alta es posa a escoltar un conjunt d'adreces multidestinació, i una és la de *solicited-node*. Per automatitzar aquest procediment, l'adreça multidestinació *solicited-node* d'una estació es construeix de la manera següent: s'agafen els darrers tres octets de l'adreça unidestinació i s'hi afegeix al principi el prefix multidestinació FF02::1:FF00:0000/104. Per exemple, l'adreça multidestinació *solicited-node* per

a 2001:630:1310:FFE1:02C0:4EA5:2161:AB39 seria la FF02::1:FF61:AB39. Aleshores l'estació es posa a escoltar el grup multidestinació per a respondre amb l'adreça de maquinari de l'equip al qual va dirigida la petició.

Això ens dóna la versatilitat que no tots els nodes reben els anuncis, i així, en el cas que no vagin dirigits cap a ells, ja ni els arriben a veure, amb la reducció en l'ús de recursos que això representa.

2.4.4. Dynamic Host Configuration Protocol

El protocol de xarxa següent que veurem no és realment un protocol de xarxa, sinó un protocol d'aplicació. De totes maneres, atès que s'utilitza per a configurar la xarxa, s'explica en aquest subapartat. El Dynamic Host Configuration Protocol (DHCP) va ser inicialment definit en el document RFC-1531. És un protocol que utilitzen els dispositius per a obtenir informació de la configuració dels paràmetres de xarxa d'un equip IPv4 de manera automàtica.

L'administrador de la xarxa configura un prefix de xarxa, juntament amb un subrang d'adreces destinades a autoconfiguració (aquest rang d'adreces s'anomena *pool*). Quan es rep una petició, el protocol comprova si el client està autoritzat, i si ho està se li assigna una IP preconfigurada, que s'obté d'una base de dades a partir de l'adreça de maquinari de l'equip, o bé una a l'atzar del *pool* si l'adreça de maquinari no es troba. Aquesta cessió d'IP està controlada per un temporitzador, i quan aquest temporitzador expira i no s'ha rebut cap notícia del client es torna la IP al *pool* d'adreces lliures. Per a evitar això, el protocol implementa un sistema de *keep-alive*, que va enviant renovacions d'ús de la IP al servidor per a evitar que caduquin. Per a fer totes aquestes tasques DHCP utilitza UDP per a enviar la informació.

Entrant en una mica més de detall, un client quan dóna d'alta una interfície enviarà un *DHCP discovery*, que és un paquet de difusió per a descobrir servidors DHCP. El servidor, quan veu el paquet, comprova la validesa del client (base de dades de MAC) i envia un *DHCP offer* amb la seva IP. Això el client ho respon directament amb un *DHCP request*, el qual finalment el servidor accepta amb un *DHCP acknowledgement*, que conté la duració de la IP i la configuració específica que el client hagi demanat en el *DHCP request*. Per exemple: encaminador per defecte, servidor de DNS, etc.

3. L'enllaç de dades i el control d'accés al medi

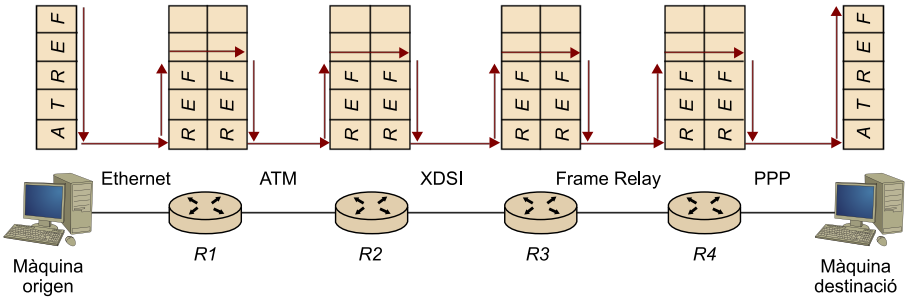
Hem vist que la capa de xarxa proporciona un servei de comunicació entre dues màquines, i estableix diferents rutes o camins entre aquestes. Cada ruta de comunicació està formada per una sèrie d'enllaços, que connecten la màquina origen amb la de destinació utilitzant uns dispositius encaminadors intermedis. Quan un datagrama del nivell de xarxa surt de la màquina origen cap a la màquina destinació, va travessant cada un d'aquests enllaços individuals que conformen el recorregut extrem a extrem.

Es fa necessària una capa lògica addicional situada immediatament sota de la capa de xarxa, que s'ocupi de subministrar-li un transport d'informació fiable entre els diferents enllaços que travessa al llarg d'un recorregut. Aquesta capa rep el nom de *nivell d'enllaç*, i se situa per sobre de la capa física. La capa física no és capaç d'aportar cap dels elements necessaris per a la transmissió efectiva d'informació en un enllaç.

El **nivell d'enllaç** consisteix en dos programes o processos que s'executen en tots dos costats d'un enllaç i es comuniquen entre si. Perquè aquests dos processos es puguin comunicar és necessari establir:

- un **format** per a la informació que s'intercanvien, i
- un conjunt de regles de comportament o protocols necessaris per a la transmissió de dades.

La principal comesa de la capa d'enllaç és la d'aconseguir que la comunicació de dades en un enllaç es faci correctament a través d'un medi físic de transmissió. D'una manera gràfica podem dir que el nivell d'enllaç s'encarrega d'establir i mantenir un pont de comunicació entre dos nodes veïns, perquè per sobre puguin circular els datagrames de nivell superior.



Ruta de comunicació creada entre dues màquines finals, formada per 5 enllaços: 2 enllaços comuniquen les màquines finals amb els encaminadors de la xarxa i 3 enllaços interns intercomuniquen només encaminadors de la xarxa.

De l'observació de la figura anterior, podem destacar dues característiques molt importants del nivell d'enllaç:

- Els enllaços al llarg d'un recorregut de comunicació poden utilitzar diferents protocols de la capa d'enllaç i estar constituïts per tecnologies de base totalment diferents. Un encaminador pot disposar de diferents enllaços i cada un pot utilitzar un protocol de nivell d'enllaç diferent. A la figura podem observar com un datagrama enviat des de la màquina origen és manejat per Ethernet en el primer enllaç, pel protocol ATM en el segon enllaç, i va canviant de tecnologia successivament en cada nou enllaç.

- Una de les funcionalitats bàsiques del nivell d'enllaç consisteix a encapsular/desencapsular els datagrames de la capa de xarxa en PDU[5] d'informació de la capa d'enllaç, anomenades també *trames*. Observem les fletxes de la figura que indiquen el flux que segueix la informació al llarg del recorregut. Quan una trama arriba a l'encaminador des d'un enllaç entrant, la capa d'enllaç desencapsula/extreu el datagrama de la trama rebuda i el lliura a la capa de xarxa. Una vegada la capa de xarxa ha determinat l'enllaç de sortida per on ha d'encaminar el datagrama, l'envia a l'enllaç. Aquí el datagrama és encapsulat segons les normes del protocol d'aquest enllaç i és preparat per a ser enviat.

[5] *PDU* són les sigles de *protocol data unit*.

3.1. Terminologia i definicions

El **node** és la màquina o encaminador.

L'**enllaç** és el canal que connecta dos nodes adjacents al recorregut de la comunicació.

El **protocol de la capa d'enllaç** és la manera de comunicar-se entre els nodes, per a moure un datagrama sobre un enllaç individual. Defineix el format dels paquets (PDU) intercanviats entre els nodes en els extrems de l'enllaç i també les accions preses per aquests nodes quan envien i reben aquests paquets.

La **trama** són les unitats de dades intercanviades per un protocol de la capa d'enllaç. El node transmissor encapsula el datagrama de la capa en una trama de la capa d'enllaç i transmet la trama a l'enllaç, i un node receptor rep la trama i extreu el datagrama.

3.2. Tipus d'enllaços

Bàsicament podem destacar dos tipus d'enllaços:

- **Enllaços de comunicació punt a punt**. Només participen dues entitats o punts. Són enllaços 1 a 1: compostos per un únic node emissor en un extrem de l'enllaç i un únic node receptor en l'altre. Tots dos nodes utilitzen en exclusiva l'enllaç, sense compartir el canal.

  Exemples d'enllaços punt a punt

  Són enllaços punt a punt:

  - Bucle d'abonat local, cable de dos fils telefònic per a accés a Internet.
  - Les xarxes d'àrea local Fast Ethernet.
  - Les xarxes d'àrea local Gigabit Ethernet.
  - PPP, HDLC (en l'àmbit d'enllaç), X.25 en l'àmbit de xarxa i TCP en l'àmbit de transport (en aquest cas és a més extrem a extrem).

- **Enllaços de difusió o canals de multidifusió**. Són enllaços 1 a *N*, en què una sèrie de nodes estan connectats al mateix canal de comunicació. La transmissió feta per un node, la reben tots els nodes connectats a l'enllaç. Es fan necessàries unes polítiques de coordinació (o protocols d'accés al medi) que permetin la compartició de l'únic medi de manera eficient, tractant d'evitar al màxim les col·lisions entre trames.

  Exemples d'enllaços de difusió

  Són enllaços de difusió:

  - Les xarxes d'àrea local Ethernet (semidúplex).
  - Les xarxes d'àrea local sense fil Wi-Fi.
  - Els enllaços amb satèl·lits.
  - Les xarxes d'accés híbrid fibra-cable (HFC).
  - Les xarxes d'àrea local Token Ring.
  - Les xarxes d'àrea local FDDI.

3.3. Tipus de serveis subministrats en la capa de xarxa

El tipus de servei que la capa d'enllaç subministra a la capa de xarxa sol ser algun entre els següents:

- **Servei no orientat a connexió i sense justificant de recepció**. La tramesa es fa sense esperar cap indicació del receptor sobre l'èxit o fracàs de l'operació. Tampoc no s'estableix o allibera una connexió. Aquest tipus de servei és apropiat quan la taxa d'error és molt baixa (xarxes locals o fibra òptica) i es deixa la missió de comprovar la correcció de la transmissió a les capes superiors (nivell de xarxa o de transport). També s'usa el servei no confirmat quan es vol transmetre informació en temps real (típicament, veu o dades) i no es vol sofrir el retard que imposaria un servei més sofisticat a la capa d'enllaç (se suposa que aquest tipus d'informació pot sofrir una petita taxa d'error sense efecte apreciable).

- **Servei no orientat a connexió amb justificant de recepció**. Es produeix un justificant de recepció per a cada trama enviada encara que no hi hagi establiment de connexió. D'aquesta manera l'emissor pot estar segur que ha arribat.

- **Servei orientat a connexió amb justificant de recepció**. És el més segur i sofisticat. L'emissor i el receptor estableixen una connexió explícita per endavant, les trames per enviar s'enumeren, i s'assegura que totes són rebudes correctament en la destinació, i són transmeses seguidament a la capa de xarxa. En el servei orientat a connexió es poden distingir tres fases: establiment de la connexió, tramesa de les dades, i acabament de la connexió. En la primera es disposen els comptadors i memòries temporals necessaris per a la transmissió, en la segona s'envien les dades i en la tercera s'allibera la memòria ocupada amb dades temporals i variables.

3.4. Serveis proporcionats per la capa d'enllaç

El servei bàsic del nivell d'enllaç consisteix a moure correctament un datagrama de nivell de xarxa, des d'un node fins a un altre d'adjacent sobre un enllaç de comunicació fixant el recorregut. Els possibles serveis que pot oferir un protocol de la capa d'enllaç són:

- **Gestió de les trames**. El nivell d'enllaç s'encarrega de l'organització i gestió de les trames.
  - Entramat o composició de la trama.
  - Sincronització en l'àmbit de trama.
  - Transparència de trama.
  - Numeració i seqüenciament (números de seqüència).
  - Direccionament: si hi ha més d'una destinació possible d'un missatge és necessari identificar-la perfectament.

- **Gestió de l'enllaç**. Tot el procés d'inici, manteniment i acabament de la transmissió requereix un considerable esforç de gestió i coordinació.

- **Control de flux**. L'estació emissora i la receptora s'han de posar d'acord en el ritme de transmissió de dades. Si l'estació receptora rep les trames més ràpidament del que és capaç de processar-les, el nivell d'enllaç remot les ha de "frenar" per evitar que se saturi la memòria intermèdia o memòria temporal que emmagatzema les trames pendents.

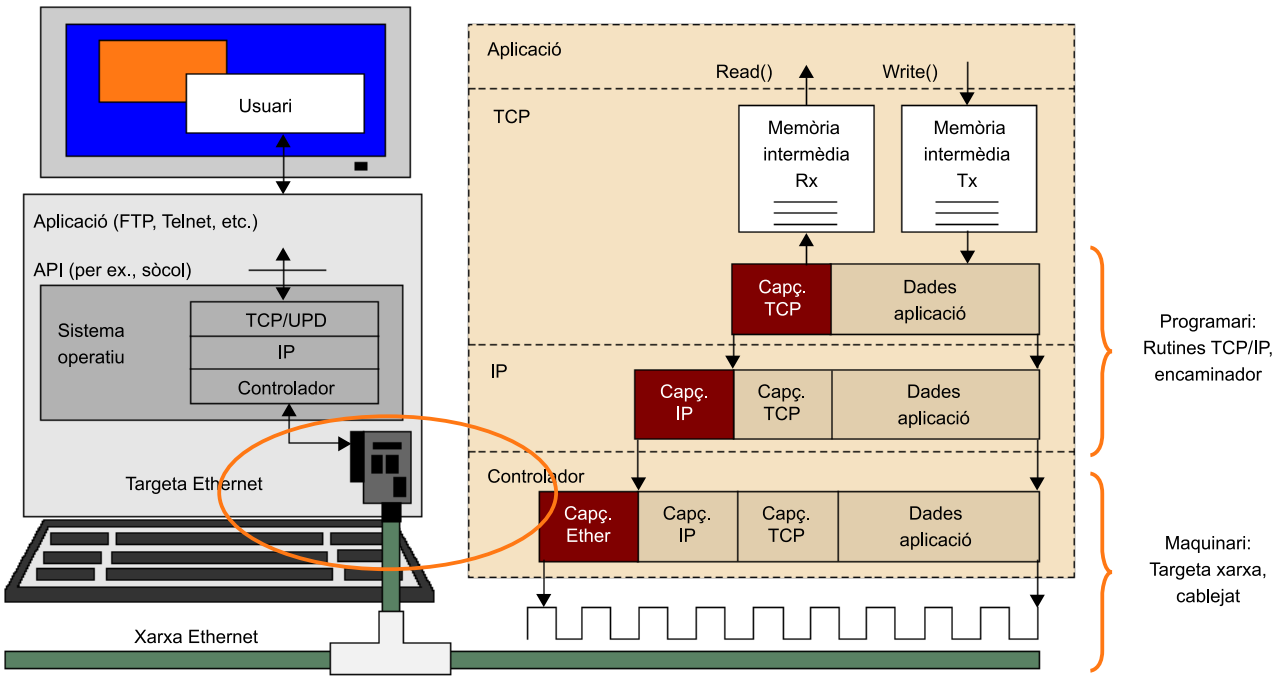
- **Control d'errors**. Es tracta d'una de les funcions bàsiques del nivell d'enllaç. S'assumeix que el medi de transmissió subjacent no és perfecte i introdueix errors de transmissió. És necessari destinar una part dels bits que s'intercanvien a la detecció i a la gestió posterior dels errors, per a controlar que no es produeixin errors de transmissió. En el control d'errors es distingeixen tres conjunts de tècniques:

– Detecció d'errors. Utilització de codis detectors d'errors eficients.
– Correcció d'errors, si s'utilitzen codis correctors adequats.
– Retransmissió de trames errònies (lliurament fiable).

- **Control d'accés al medi**. Aquesta funcionalitat pren rellevància en els enllaços d'accés múltiple o de difusió en què un nombre determinat de nodes comparteixen el mateix medi físic. El model OSI divideix la capa d'enllaç en dues subcapes: l'LLC (*logical link layer*) i la MAC (*medium access control*). La subcapa MAC és l'encarregada d'especificar les regles amb què es transmet una trama sobre l'enllaç. Els protocols MAC coordinen la transmissió de les trames dels nodes, amb l'objectiu d'evitar les col·lisions de trames. En els enllaços punt a punt els protocols d'accés al medi deixen de tenir sentit.

3.5. Adaptadors i dispositius de xarxa

Els nodes o encaminadors es connecten als enllaços per mitjà d'un adaptador, conegut com a targeta d'interfície de xarxa o *network interface card* (NIC). Físicament un adaptador és una placa de maquinari (una targeta PCMCIA o un dispositiu connectat al port USB) que conté tots els elements d'un petit computador: memòria RAM, xip DSP, una interfície de bus amb la màquina, i una interfície d'enllaç. Normalment es troba allotjat a la mateixa capa física que la resta del node, comparteix l'alimentació i els busos amb la resta del node, i és en el fons sota el control del node.



Un adaptador té un cert grau d'autonomia:

- **En recepció**: quan rep una trama, determina si la trama té errors. Si és així la rebutja sense notificar-ho al seu node pare. Si és correcta, desencapsularà el datagrama de la capa de xarxa i interromprà el seu node pare per passar-lo cap a dalt de la pila de protocols.

- **En transmissió**: quan un node passa un datagrama cap a baix a la pila de protocols a un adaptador, delega totalment a l'adaptador la tasca de transmetre el datagrama sobre l'enllaç. L'adaptador encapsula el datagrama en una trama i transmet la trama en l'enllaç de comunicació.

Els components principals d'un adaptador són la interfície del bus i la de l'enllaç. La interfície del bus és responsable de comunicar amb el node pare de l'adaptador. Transfereix dades i informació de control entre l'adaptador i el node pare. La interfície de l'enllaç és responsable d'implementar el protocol de la capa d'enllaç. També inclou el maquinari de transmissió i recepció.

El protocol de la capa d'enllaç està, majoritàriament, implementat en aquest adaptador. Si el protocol de la capa d'enllaç proporciona detecció d'errors, lliurament fiable (numeració i reconeixements) o accés aleatori (a més d'emmarcar i desemmarcar datagrames), llavors aquestes funcionalitats estan implementades completament en els adaptadors.

4. El nivell físic

Per acabar aquest mòdul, estudiarem l'última capa del model OSI[6]. El nivell físic caracteritza el senyal i la seva modulació i descriu els medis de transmissió físics.

(6)*OSI* és la sigla d'*Open Systems Interconnection*; en català, 'interconnexió de sistemes oberts'.

En aquest mòdul no entrarem en els fonaments matemàtics de la modulació i del senyal. Simplement hem de saber que la informació binària es pot transmetre per un medi per mitjà de les variacions d'alguna propietat física; habitualment, el voltatge o la intensitat. Podem representar el valor d'aquesta magnitud física com una funció depenent del temps. Donada aquesta funció en forma d'ona la podem codificar per a transmetre la informació que volem. Aquest és el fonament de la transmissió d'informació a través d'un medi i sobre el qual es basa tota la teoria de la comunicació.

4.1. Medis de transmissió

Hi ha diversos medis de transmissió en funció de la seva amplada de banda, el retard, el cost, la facilitat d'ús, la instal·lació i el manteniment. Els medis es poden classificar en medis guiats (parell de fils, fibra òptica) o no guiats (ones de ràdio, làser a través de l'aire). Els factors següents d'un medi de transmissió determinen la velocitat màxima de transmissió i la distància màxima del medi:

- **Amplada de banda**. En augmentar l'amplada de banda es pot augmentar la velocitat de transmissió.
- **Atenuació**. En ordre decreixent van el parell trenat, el cable coaxial i la fibra òptica.
- **Interferències**. En ordre decreixent van el parell trenat i el cable coaxial.
- **Nombre de receptors**. Atenuen i distorsionen el senyal que representa menys distància.

4.1.1. Parell trenat

Es tracta del sistema més antic i encara actualment molt utilitzat. Consisteix en dos parells de fils de coure o d'acer cobert amb coure, aproximadament d'un mil·límetre de diàmetre cada un, que estan embolicats entre ells en forma d'hèlix, com una molècula d'ADN.

Es pot utilitzar tant en les transmissions digitals com en les analògiques. L'amplada de banda que ofereixen depèn del gruix del cable i de la distància.

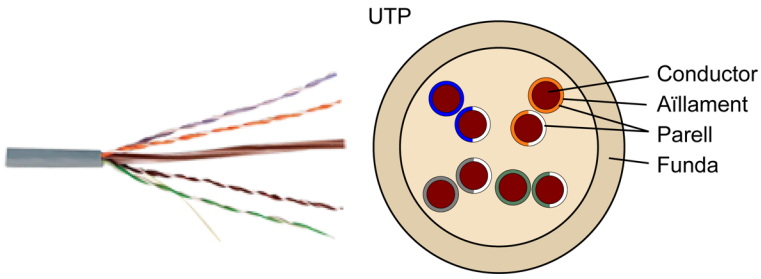
Cost del parell trenat

A causa del seu baix cost i el seu rendiment, és un sistema molt popular. El cost és el més econòmic (més que el coaxial o la fibra òptica) per metre, però té uns costos de connectivitat semblant a altres medis.

És un sistema molt utilitat en els sistemes de telefonia. Generalment, els telèfons dels habitatges estan connectats amb la centraleta telefònica a través de parells trenats (el seu amplada de banda és de 4 kHz). També s'utilitza en les LAN a velocitats de 10, 100 i 1.000 Mbps. És molt utilitzat en les connexions punt a punt, i el seu àmbit geogràfic sol ser d'uns 100 metres (en xarxes Ethernet).

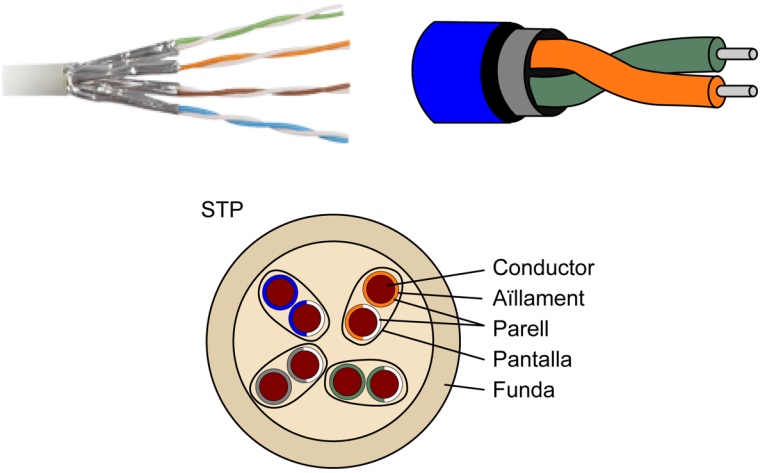
Els primers parells trenats no apantallats s'anomenaren *unshielded twisted pair* (UTP). El parell trenat habitualment s'agrupava en 4 parells trenats més dins una protecció de plàstic. Aquest tipus de cable s'anomenava *de categoria 3*. Després s'introduïren els cables de categoria 5, agrupament amb més densitat per centímetre de voltes al cable, que oferia unes característiques de més qualitat sobre llargues distàncies.

Parell trenat no apantallat (UTP)



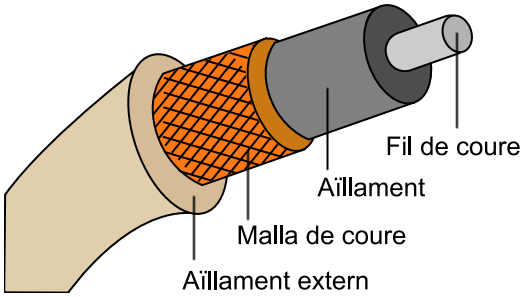
Després vingué l'evolució al tipus de cablatge *shielded twisted pair* (STP), anomenat *parell trenat apantallat*, en què cada parell de fils té una protecció individual.

Parell trenat apantallat (STP)



4.1.2. Cable coaxial de banda base

El cable coaxial té un recobriment superior als parells trenats, i funciona a llargues distàncies i altes velocitats. Aquest cable consisteix en un fil de coure, rodejat per un material aïllant. Per a cables d'1 km de llargària, suporta velocitats d'1 a 2 Gbps. Es poden utilitzar distàncies més llargues però amb velocitats de transmissió més baixes. Aquest tipus de cable s'ha utilitzat per a interconnectar els equips de les centrals telefòniques, per a alguna xarxa d'àrea local o per a la televisió per cable. És més econòmic que la fibra òptica i més car que el parell trenat.

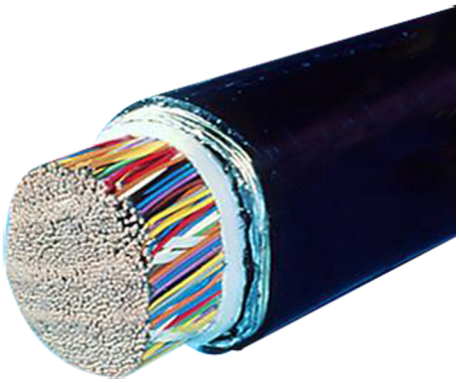


4.1.3. Fibra òptica

Un sistema de comunicacions òptiques està format per tres components: una font de llum, el medi de transmissió i el detector de llum. Convencionalment, un impuls de llum indica un bit amb valor 1, i l'absència de llum indica un bit de valor 0. El medi de transmissió és una fibra òptica molt fina. El detector genera un impuls elèctric quan la llum incideix a sobre. Així doncs, instal·lant una font de llum a l'inici de la fibra òptica i un detector de llum al final de la fibra, tenim un sistema de transmissió unidireccional que accepta senyals elèctrics, els converteix en impulsos de llum que es transmeten i es reconverteixen en un senyal elèctric al final de la fibra.

La matèria primera de la fibra òptica és el vidre; és un material no excessivament car i n'hi ha grans quantitats al nostre planeta. El vidre utilitzat en les fibres òptiques és un vidre transparent. La velocitat de transmissió és molt elevada, i arriba a uns quants Gbps en 30 km de distància.

Fibra òptica



Resum

En aquest mòdul s'han vist en detall les característiques i el funcionament dels diferents nivells de la xarxa. L'objectiu del mòdul ha estat donar una visió general del funcionament intern d'una xarxa de computadors, tant d'àrea local com a gran escala, com és Internet. L'aproximació als nivells de la xarxa s'ha fet des dels nivells més propers a les aplicacions fins als nivells més específics del maquinari. En primer terme s'ha introduït el nivell de transport, que ofereix fiabilitat a la xarxa tot permetent l'accés de diferents aplicacions a un únic medi de transmissió, la xarxa. S'ha vist que el funcionament de la xarxa es regeix sobretot pel protocol TCP, que és orientat a connexió i que dóna garanties de fiabilitat a la xarxa. Per a aplicacions que no requereixen garanties hi ha el protocol UDP, que és usat majoritàriament per aplicacions en temps real. Seguidament s'ha presentat la capa de xarxa. El nivell de xarxa és cabdal per al funcionament d'Internet, ja que ens permet adreçar i identificar els nodes d'una xarxa. En aquest apartat hem vist IPv4 i IPv6. S'ha presentat el model d'adreçament i la problemàtica actual amb el nombre d'adreces a la xarxa.

S'ha presentat la capa d'enllaç que abstrau els nodes de la xarxa del medi físic sobre el qual transmeten la informació. S'ha vist que aquesta capa s'encarrega d'adreçar la informació entre nodes físics adjacents i garantir transmissió fiable entre aquests. Cal destacar la tasca de la subcapa de control d'accés al medi, que permet transmetre informació entre dos nodes independentment de la tecnologia de xarxa utilitzada, ja sigui sense fil o cablejada. Finalment s'ha fet una breu introducció al nivell físic, que s'encarrega dels aspectes de modulació i codificació dels senyals físics, que són dependents directament de la tecnologia i el medi de transmissió. S'han vist els principals medis de transmissió.

Bibliografia

Almquist, P. (1992). "RFC-1349: Type of Service in the Internet Protocol Suite". Consultant (juliol).

Bing, B. (2000). *Broadband Wireless Access*. Dordrecht: Kluwer Academic Publishers.

Bing, B. (2000). *High-Speed Wireless ATM and LAN*. Londres: Artech House.

Croft, B.; Gilmore, J. (1985). "RFC-951: Bootstrap Protocol (BOOTP)". Stanford University & Sun Microsystems (setembre).

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* (núm. 1, pàg. 269-271).

Droms, R. (1993). "RFC-1531: Dynamic Host Configuration Protocol". Lewisburg: Bucknell University (octubre).

Gartner, F. C. (2003). *A Survey of Self-Stabilizing Spanning-Tree Construction Algorithms*, Lausana: Swiss Federal Institute of Technology Tech. [Rep. IC/2003/38, School of Computer and Communication Sciences.]

IANA Reserved Multicast Address List [document en línia]. <http://www.iana.org/assignments/multicast-addresses>. [Data de consulta: 12 d'abril del 2010.]

Kent, S.; Seo, K. (2005). "RFC-4301: Security Architecture for the Internet Protocol". *BBN Technologies* (desembre).

Kurose, J.; Ross, K. (2005). *Computer Networking: a Top-Down Approach Featuring the Internet*. Amherst: Department of Computer Science, University of Massachusetts.

Information Sciences Institute (1981). "RFC-791: Internet Protocol Specification". Marina del Rey: Information Sciences Institute, University of Southern California (setembre).

Intel Corporation (1999). *Preboot Execution Environment (PXE) Specification* (setembre).

Hedrick, C. (1988). "RFC-1058: Routing Information Protocol". Piscataway: Rutgers University (juny).

Malkin, G. (1998). "RFC-2453: RIP Version 2". *Bay Networks* (novembre).

Moy, J. (1994). "RFC-1584: Multicast Extensions to OSPF" (març).

Moy, J. (1998). "RFC-2328: OSPF Version 2". Ascend Communications (abril).

Postel, J. (1981). "RFC-792: Internet Control Message Protocol (ICMP)". ISI (setembre).

Rekhter, Y.; Li, T.; Hares, S. (2006). "RFC-4271: A Border Gateway Protocol 4 (BGP-4)" (gener).

Reynolds, J. (2002). "RFC-3232: Assigned Numbers: RFC 1700 is Replaced by an On-line Database" (gener).

Reynolds, J.; Postel, J. (1994). "RFC-1700: Assigned Numbers" (octubre).

Srisuresh, P.; Egevang, K. (2001). "RFC-3022: Traditional IP Network Address Translator (Traditional NAT)". Intel Corporation (gener).

Srisuresh, P.; Holdrege, M. (1999). "RFC-2663: IP Network Address Translator (NAT) Terminology and Considerations". Murray Hill: Lucent Technologies (agost).

Tanenbaum, A. S. (2003). *Redes de computadores* (4a. ed.). Nova York: Prentice-Hall Professional Technical Reference.