

## 6. Serveis

Col·laboradors:

Sr. Lluís Casals Ibáñez

Dr. David Rincón Rivera

Sra. Immaculada Ruiz Vela

Dr. Rafael Vidal Ferré

Dr. Daniel Guasch Murillo

Gener 2022



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Politècnica Superior d'Enginyeria  
de Vilanova i la Geltrú



# 6.1. Característiques del nivell d'aplicació

## Conceptes bàsics

Què s'entén per “nivell d'aplicació”?

- Programes que permeten accedir als **serveis de comunicacions** oferts sobre les xarxes TCP/IP
  - E-mail
  - Transferència de fitxers
  - Terminal remot
  - Transmissió de vídeo en temps real...
- Aplicacions **molt diferents**  $\Rightarrow$  serveis **molt específics**  $\Rightarrow$  requeriments tècnics molt variats.
  - El correu pot patir retards elevats, però una transmissió de vídeo en temps real, no.
  - El vídeo pot admetre pèrdua de paquets, el correu no.

# 6.1. Característiques del nivell d'aplicació

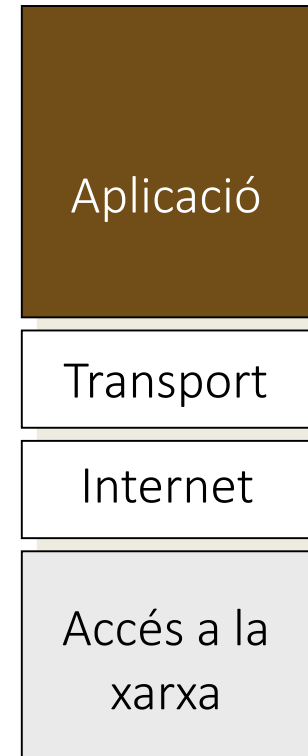
## Conceptes bàsics

Distinció entre **servei** i **aplicació**

- **Servei:** concepte abstracte
- **Aplicació:** peça de software que dona el servei

Aplicacions al model TCP/IP

- És la capa més propera a l'usuari
- Utilitza els serveis proporcionats per la capa de transport
  - A través de la interfície de programació, *sockets*
- S'encarrega d'obrir sessions de treball



## 6.1. Característiques del nivell d'aplicació

### Classificació dels serveis telemàtics

#### Orientats a connexió





- S'estableix un circuit lògic, abans d'iniciar la comunicació
- Concepte semblant al d'una trucada telefònica: trucar abans de parlar
- Exemple: transferència de fitxers
  - Abans de transferir, obres una connexió amb el servidor




#### No orientats a connexió

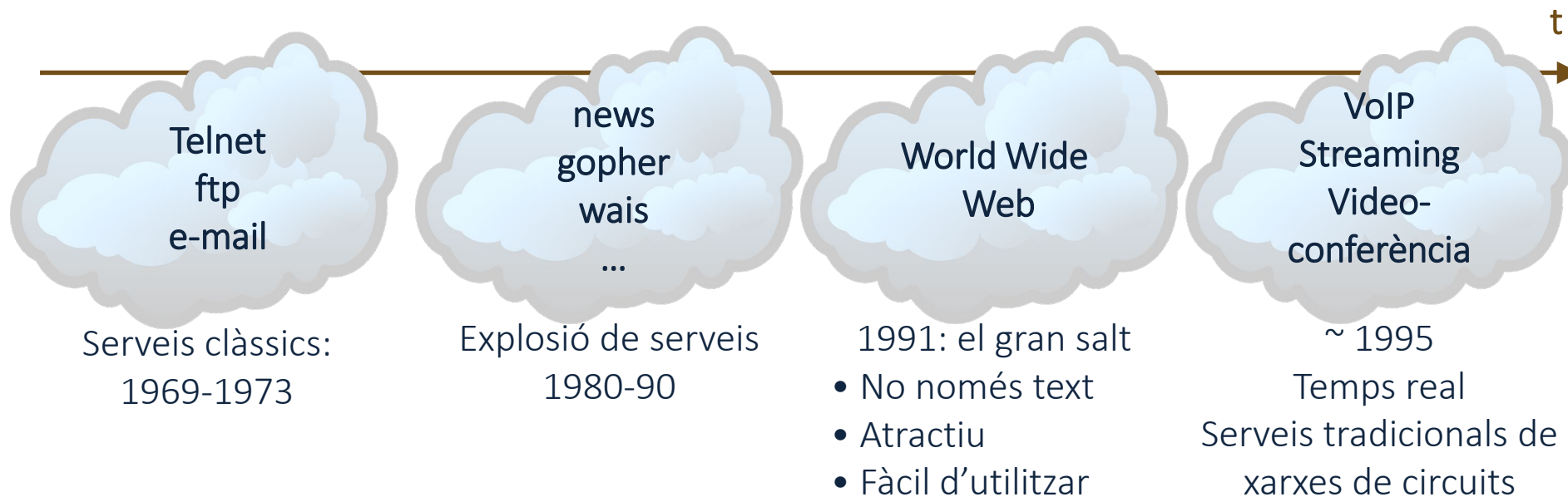
- No cal establir cap circuit previ; només s'envia la informació
- Concepte semblant al d'un telegrama: s'envia sense connexió
- Exemple: correu electrònic

# 6.1. Característiques del nivell d'aplicació

## Evolució dels serveis internet

-  Emulació de terminal: TELNET (Terminal Networking)
-  Transferència de fitxers: FTP (File Transfer Protocol)
-  Correu electrònic: SMTP (Simple Mail Transfer Protocol)
-  World Wide Web: HTTP (HyperText Transfer Protocol)

-  Localització de dominis: DNS (Domain Name System)
-  Administració de xarxa: SNMP (Simple Network Management Protocol)
-  Notícies electròniques: NNTP (News Network Transfer Protocol)



# 6.1. Característiques del nivell d'aplicació

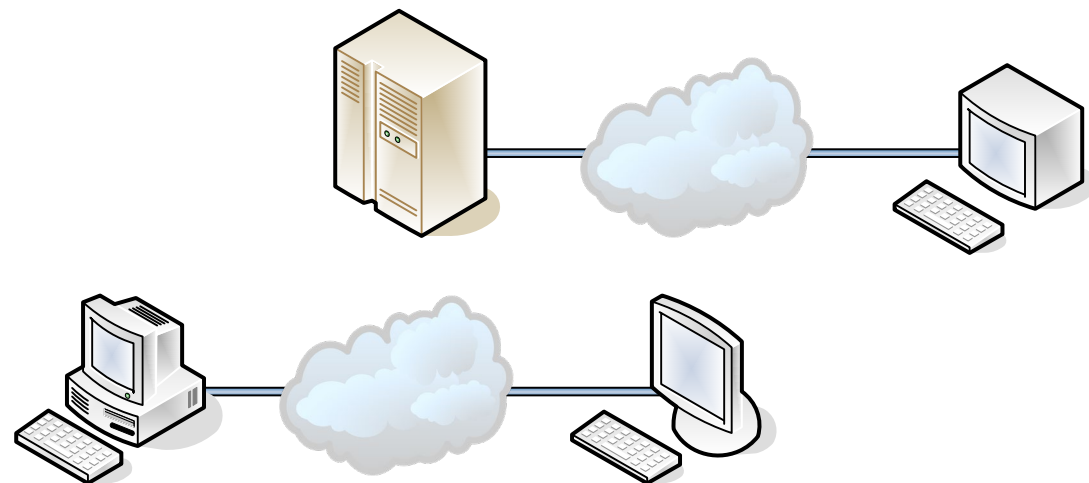
## Arquitectura de les aplicacions

Arquitectura d'una aplicació

- Model que segueixen les aplicacions que ofereixen un cert servei
- Fa referència a com estan construïdes les peces de software i com interactuen entre elles

Dos models clàssics

- Client-servidor
- *Peer-to-peer*



## 6.1. Característiques del nivell d'aplicació

### Arquitectura client - servidor

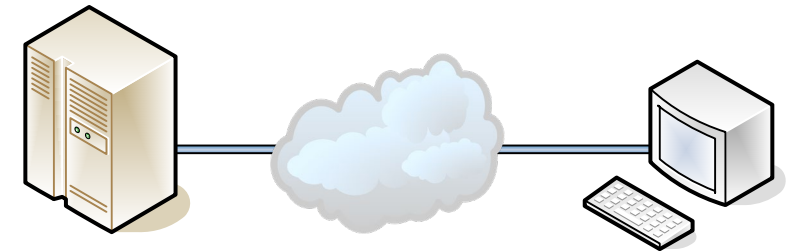
Els serveis TCP/IP solen seguir aquest model

Servidor

- Màquina que es dedica de manera permanent a prestar serveis a la resta de hosts d'Internet
- Corre programes que esperen connexions (*daemons*)
- Pot atendre simultàniament diversos clients

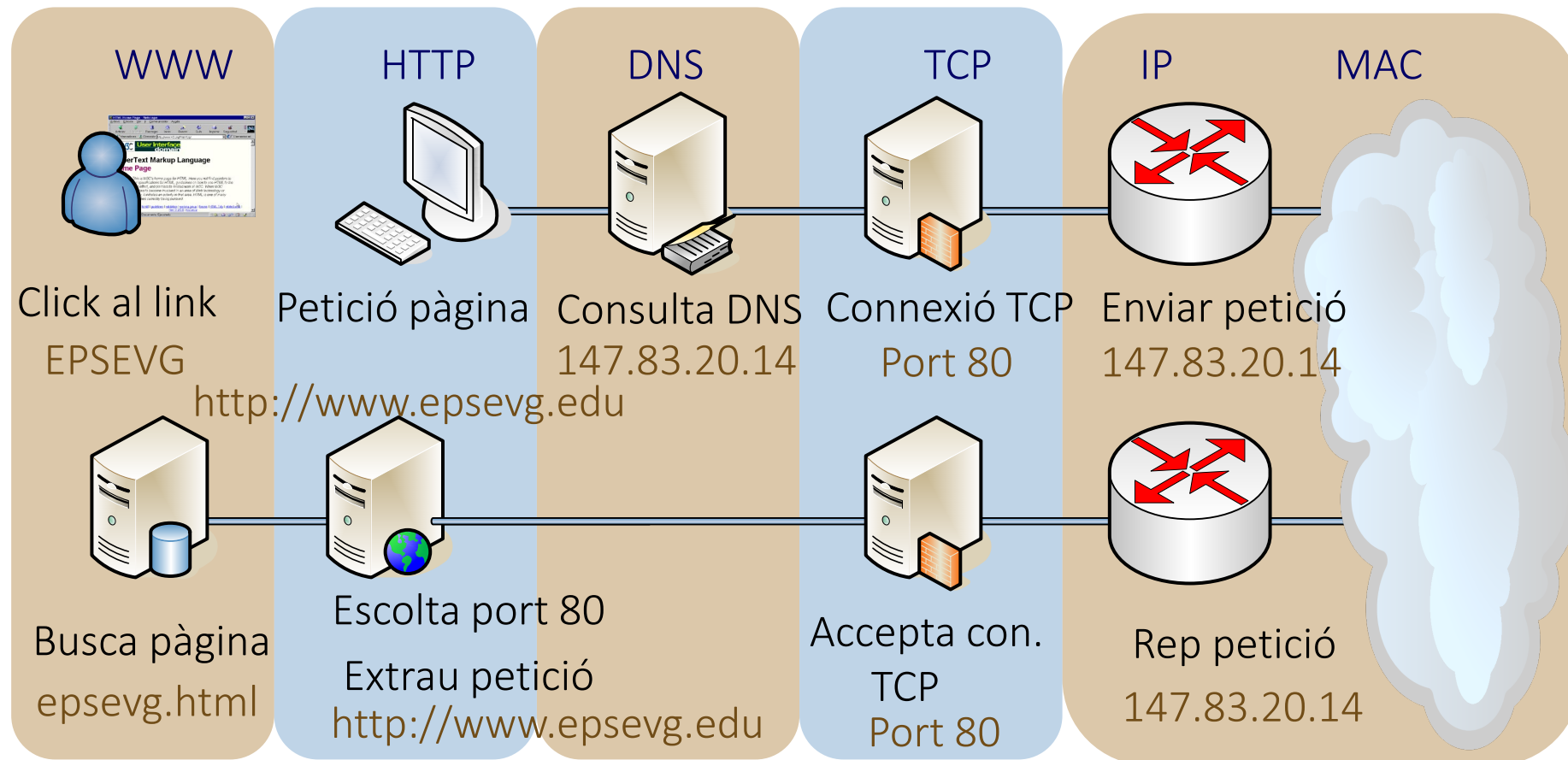
Client

- Màquina d'usuari, “senzilla”
- Es connecta al servidor per demanar un servei
- En principi, només gestiona una connexió per a cada servei



# 6.1. Característiques del nivell d'aplicació

## Exemple d'arquitectura client - servidor





# 6.1. Característiques del nivell d'aplicació

## Arquitectura peer – to - peer

Totes les màquines tenen la mateixa importància

- No es distingeix entre client i servidor
- Totes poden accedir al servei (client) o oferir-lo (servidor)



- Exemple: compartició de fitxers (Napster, Morpheus, etc)

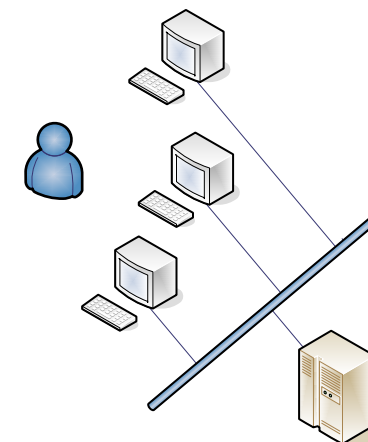
## 6.2. telnet

### Terminal Networking: telnet

Telnet permet a un usuari local obrir una sessió de terminal en una màquina remota

- L'usuari treballa com si fos al costat de la màquina remota.

És l'aplicació més antiga d'Internet: 1969



- Objectiu inicial d'Internet: **compartició** i **accés remot** a supercomputadors.
  - Anècdota: 3 terminals diferents al despatx del cap de l'ARPA

29 OCT 69	2100	LOADED OP. PROGRAM	CSK
		FOR BEN BARKER	
		BBV	
	22:30	Talked to SRI	CSK
		Host to Host	
		Left imp. program	CSK
		running after sending	
		a host dead message	
		to imp.	

## 6.2. telnet

### Característiques bàsiques

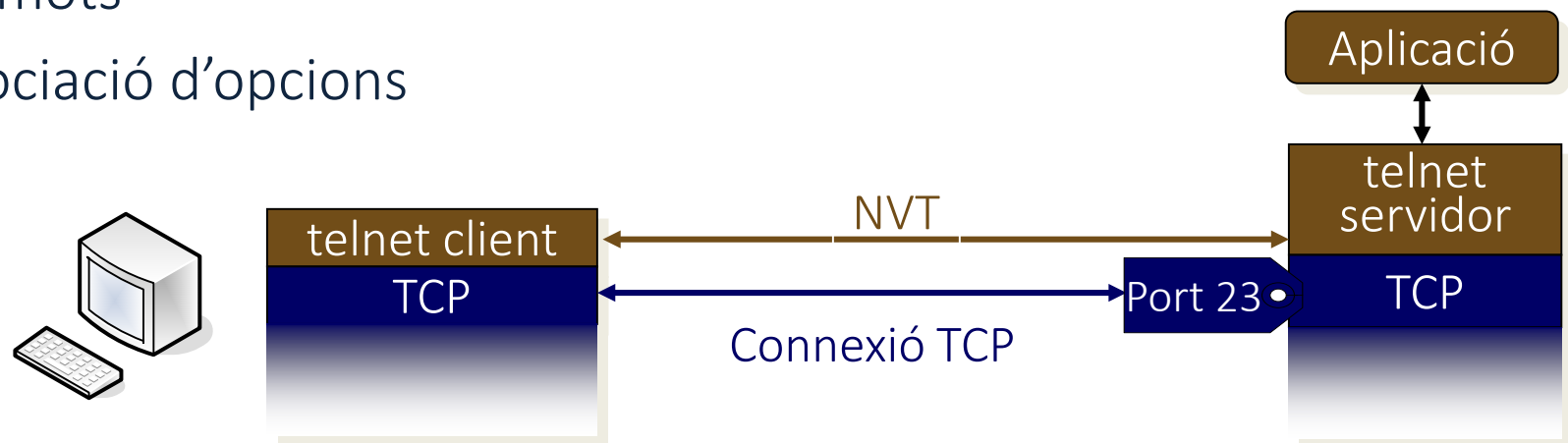
Definit al RFC 854

Telnet fa servir el servei de transport TCP

- Dues comunicacions simètriques pel port 23

Serveis bàsics:

- Defineix un terminal virtual de xarxa (*Network Virtual Terminal*, NVT) que proporciona una interfície estàndard als sistemes remots
- Permet la negociació d'opcions



## 6.2. telnet

### NVT, Network Virtual Terminal

Terminal Virtual de Xarxa

- Simula una pantalla i un teclat
- Dissenyat com a protocol half duplex i amb intercanvi línia a línia
  - Només un host transmet simultàniament. Després d'enviar una línia, el client espera a rebre dades del servidor. El servidor envia les dades i després un *go ahead*, indicant que el client pot transmetre
- Posteriorment opció caràcter-a-caràcter

Caràcters ASCII de 7 bits farcits a 8 amb un 0 inicial

- Cada línia acaba amb una combinació CR i LF (ASCII)
- Els caràcters que comencen per 1 són ordres

Normalment només s'utilitza durant un període de temps curt, per negociar les opcions d'algun emulador de terminal.

- Tipus de terminals: ASCII, IBM 3270, **vt100**

NVT també s'utilitza a FTP, SMTP, Finger...

## 6.2. telnet

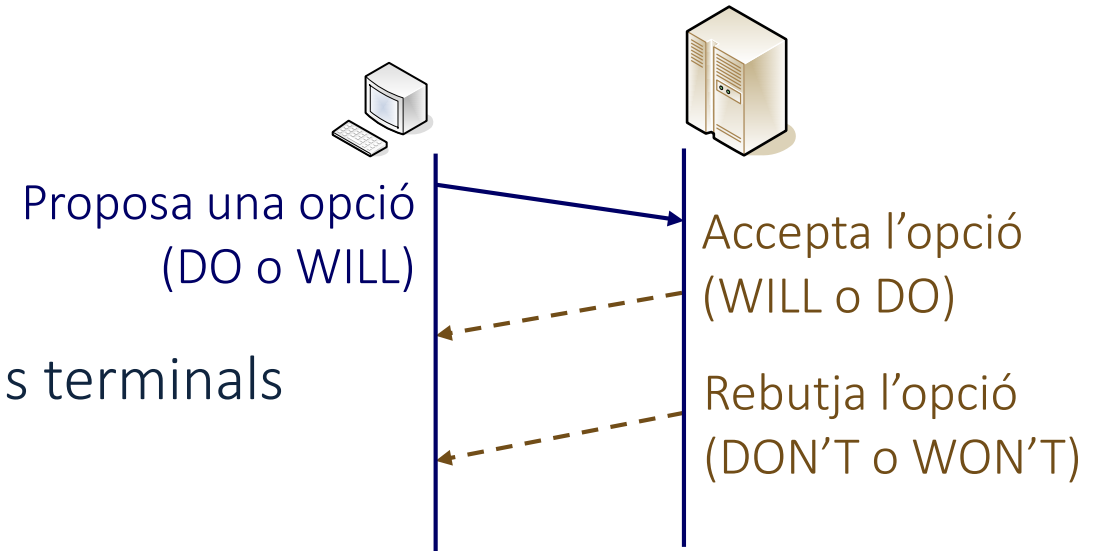
### Establiment de la comunicació

Autenticació:

- login + password

Negociació inicial d'opcions

- Per anar més enllà del NVT, adaptant-se als terminals
- Opcions:
  - Eco local o remot
  - Consultar l'estat de l'extrem oposat (status)
  - 7 / 8 bits per caràcter
  - Intercanvi d'informació sobre el terminal
    - Tipus, velocitat, CR o CR + LF, ...

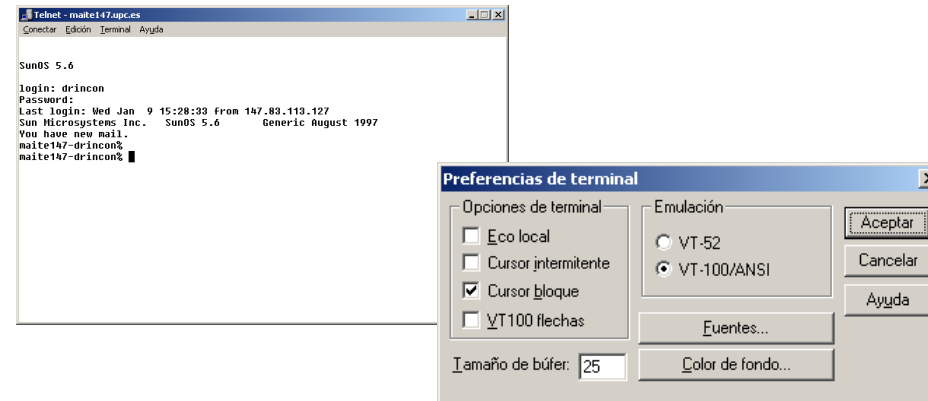


- Control de flux
- Edició mode línia o caràcter
- Encriptació
- Autenticació

## 6.2. telnet

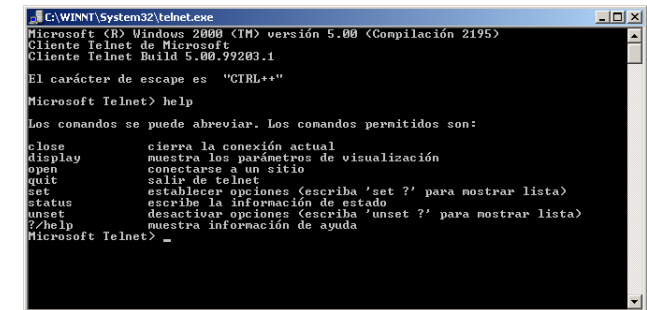
### Client Telnet

Integrat al sistema operatiu (Unix, Windows...)



Sintaxi: telnet[host][port]

- Podem fer telnet a altres ports que no són el 23
- Possibilitat de demanar help



## 6.2. telnet

### Client Telnet

#### Ordres bàsiques

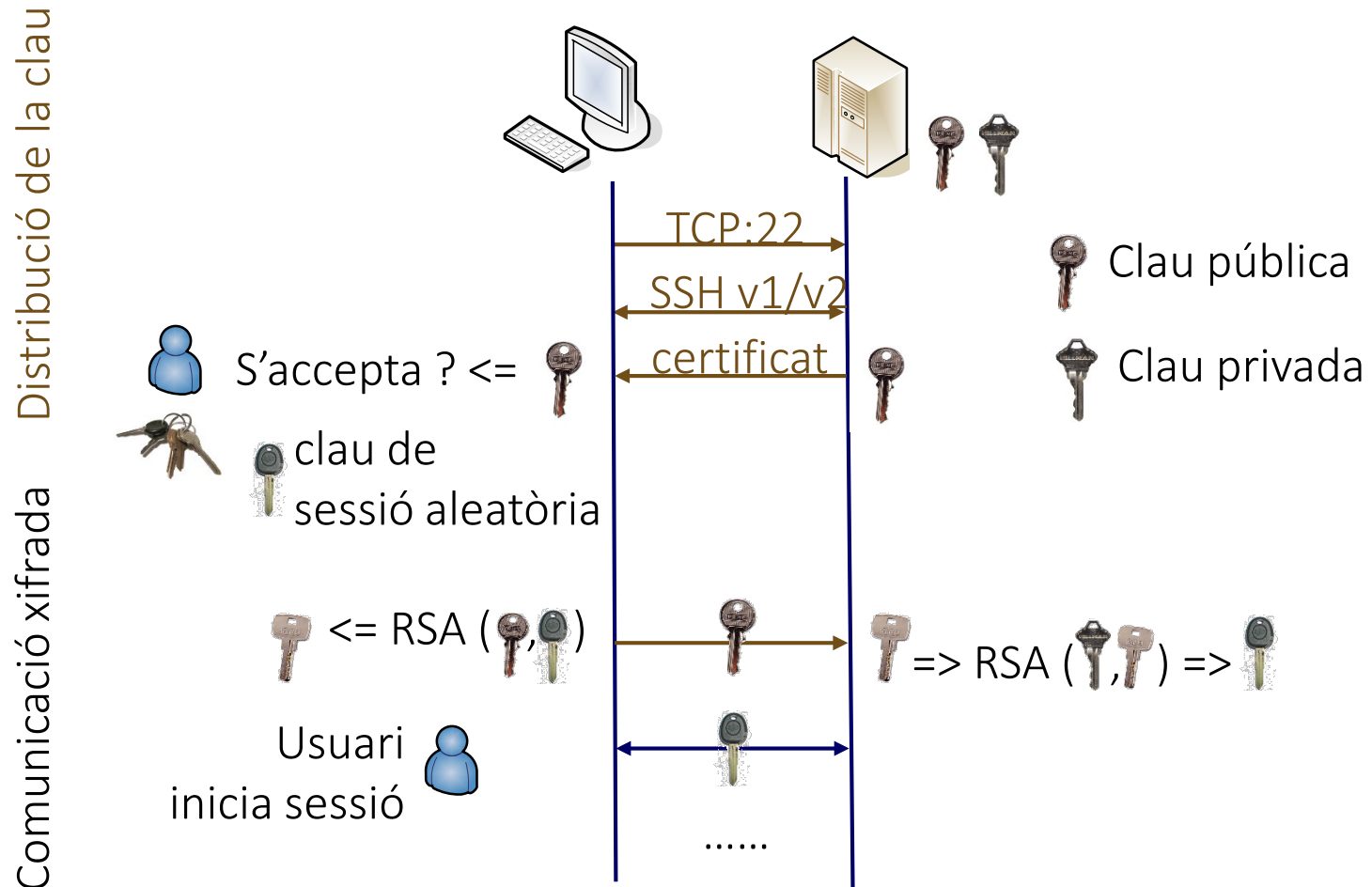
close	tanca la connexió en curs.
logout	fa que l'usuari es desconnecti i tanca la connexió.
open	obre una connexió a un host.
mode	indica si la transmissió es fa caràcter a caràcter ( <i>character</i> ) o s'envia una línia de caràcters ( <i>line</i> ) quan es generi un EOL ( <i>end of line</i> ).
quit	surt del programa.
set	activa paràmetres d'operació (echo, escape, erase, kill, eof, quit)
status	visualitza la informació de l'estat de la connexió, el mode, l'eco i el caràcter d'escape.

Caràcter de control (CTRL], CTRL++): permet sortir al mode d'execució d'ordres.

Es pot tornar al terminal amb return.

## 6.2. telnet

SSH (Secure Shell) millora i securitza el telnet





## 6.3. FTP

### Transferència de fitxers: FTP

Una de les aplicacions bàsiques d'Internet

FTP (*File Transfer Protocol*)

- Ús compartit de fitxers entre sistemes remots
- Possibilitat d'enviar, rebre, esborrar, i gestionar fitxers i directoris

Diversos protocols

- FTP: *File Transfer Protocol*. RFC 959
- SFTP: *Simple File Transfer Protocol*.
- TFTP: *Trivial File Transfer Protocol*. RFC 1350

## 6.3. FTP

### Característiques del FTP

Suposa que es disposa d'un servei fiable extrem a extrem (TCP)

– Dues connexions TCP

- Control pel port 21: sessió NVT
- Dades pel port 20

– Connexió de control:

- Diàleg de comandes (client) i codis de resposta (servidor)
- Comença en el moment en què el client es connecta al servidor



– Connexió de dades sols s'obre per a copiar fitxers o fer llistats

- El client reserva un port x i li indica al servidor (comanda PORT) que a continuació s'hi connecta




## 6.3. FTP

### Característiques del FTP

- Accés interactiu: persona o màquina
  - Codis de resposta de 3 dígits per possibilitar control remot
  - El primer indica el tipus d'operació
    - 2xx : èxit
    - 1xx : l'acció ha començat
    - 3xx : un punt intermig ha estat aconseguit amb èxit
    - 4xx : error transitori
    - 5xx : error permanent

## 6.3. FTP

### Característiques del FTP

- Format de la informació
  - ASCII: per fitxers de text ASCII
    - S'interpreten els caràcters especials com canvi de línia
  - Binari: per altres fitxers
    - Es tracta tot el fitxer com un flux continu de bits
  - És molt important utilitzar el mode correcte !!
    - Exemple: un fitxer Word NO és un fitxer de text
- Assignació de noms
  - fitxer local  fitxer remot
- Autenticació:
  - login + password

## 6.3. FTP

### Exemple de sessió FTP amb Windows 2000

```

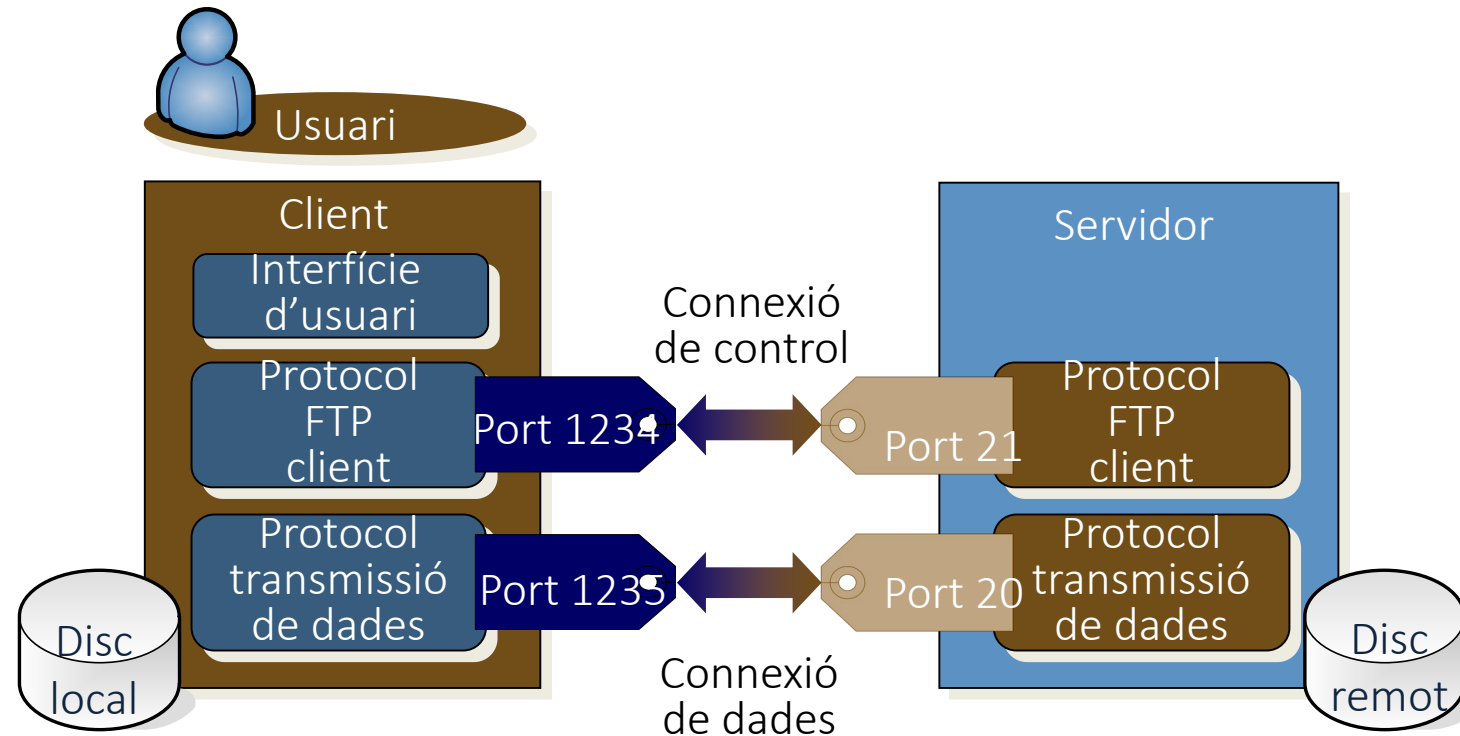
C:\WINNT\System32\ftp.exe
ftp> open maite147.upc.es
Conectado a maite147.upc.es.
220 maite147 FTP server (SunOS 5.6) ready.
Usuario (maite147.upc.es:(none)): drincon
331 Password required for drincon.
Contrase#a:
530 Login incorrect.
Error al iniciar la sesión.
ftp> user drincon
331 Password required for drincon.
Contrase#a:
230 User drincon logged in.
ftp> bin
200 Type set to I.
ftp> hash
Impresión de marcas "hash" Activo  ftp: (2048 bytes/marca "hash") .
ftp> mget *.txt
200 Type set to I.
mget amics.txt? y
200 PORT command successful.
150 Binary data connection for amics.txt (147.83.113.127,2484) (1803 bytes).

226 Binary Transfer complete.
ftp: 1803 bytes recibidos en 0,20Segundos 9,02KB/s.
mget amsterdam.txt?
    
```

## 6.3. FTP

### Model del FTP

- Daemon FTP llançat per inetd
- Per a cada transferència de dades es crea una nova connexió
- El protocol de la connexió de dades és una particularització del telnet



## 6.3. FTP

### Ordres d'usuari en el client FTP

Funcions de control:

- Identificar el tipus de transferència: **ascii, binary**
- Confirmar cada fitxer transferit: **prompt**
- Sortir: **quit, bye**
- Ajut: **help**
- Obrir / tancar una connexió: **open, close**

Funcions de transferència de dades:

- Copiar fitxers entre hosts: **get, put, mget, mput**
- Afegir un fitxer local a un fitxer remot: **append**

Funcions de control sobre fitxers:

- Llistar un directori: **ls, dir**
- Imprimir el directori actual: **pwd**
- Canviar de directori: **cd, lcd**
- Esborrar / renombrar un fitxer: **delete, rename**
- Mostrar l'estat de la connexió: **status**

## 6.3. FTP

### Exemple de dialog de control FTP

Obrim una sessió

Client indica connexió exitosa

Missatge servidor

Login i Password

Canvi directori

Copiem un fitxer

Nova connexió per copiar fitxer

```

Interfaz de comandos
ftp> open watmBS2.upc.es
Connected to watmBS2.upc.es.
220 watmBS2.mat.upc FTP server (Version wu-2.4.2-academ[BETA-18](1) Mon Aug 3 19
:17:20 EDT 1998) ready.
User (watmBS2.upc.es:(none)): rvidal
---> USER rvidal
331 Password required for rvidal.
Password:
---> PASS xxxzzff
230 User rvidal logged in.
ftp> cd watm
---> CWD watm
250 CWD command successful.
ftp> get PUC/client.c
---> PORT 147,83,40,101,4,180
200 PORT command successful.
---> RETR PUC/client.c
150 Opening ASCII mode data connection for PUC/client.c (4573 bytes).
226 Transfer complete.
4747 bytes received in 0,16 seconds (29,67 Kbytes/sec)
ftp> quit
---> QUIT
221 Goodbye.
C:\>

```



## 6.3. FTP

### Exemple d'interfícies FTP

The image shows the WS\_FTP95 interface with two main panes: 'Local System' and 'Remote System'. The 'Local System' pane shows a directory listing for 'C:\Archivos de programa\WS\_FTP' with files like 'complete.wav', 'connect.wav', etc. The 'Remote System' pane shows a directory listing for '/usr/local/httpd/htdocs/grup\_de\_mobils' with folders like '.tkdesk', 'backup', 'DIB', etc. Below the panes is a status bar with a log window showing a successful transfer of a file. A 'Propiedades de Session' dialog box is open in the foreground, showing fields for Profile Name, Host Name/Address, Host Type, User ID, Password, Account, and Comment. Arrows point from text labels to specific parts of the interface.

**Fitxers remots** (Remote files) - points to the Remote System directory listing.

**Configuració de la sessió** (Session configuration) - points to the 'Propiedades de Session' dialog box.

**Fitxers remots** (Remote files) - points to the Local System directory listing.

**Diàleg** (Dialog) - points to the status bar/log window.

**Format transport** (Transport format) - points to the 'ASCII' and 'Binary' radio buttons.

## 6.3. FTP

### FTP anònim

- Alguns servidors permeten l'accés anònim
- Identificació per login anonymous, password = e-mail

```

C:\TMP>ftp ftp.upc.es
Connected to diable.upc.es.
220-    O O O
220-    O O O                Servei d'FTP de la UPC
220-    O O O
220-    U P C
220-
220 diable.upc.es FTP server () ready.
User (diable.upc.es:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-----
230-
230-    Benvingut al servei d'FTP anonim de la UPC.
230-    Bienvenido al servicio de FTP anonimo de la UPC.
230-    Welcome to the anonymous FTP service on the UPC.
230-
230-    T'has connectat des de 'maite30.upc.es'.
230-
230-    Si tens problemes amb el sistema pots consultar a
230-                ftpmanager@upc.es
230-
230-    Usuaris connectats: 2 de 15 permesos
230-
230 User ftp logged in.  Access restrictions apply.
ftp>

```

## 6.3. FTP

### TFTP: Trivial File Transfer Protocol

FTP és difícil d'implementar i ofereix més del que es vol utilitzar en determinades ocasions.

En determinades ocasions és millor utilitzar TFTP

- Protocol petit i fàcil d'utilitzar
- TFTP sols permet la transferència de fitxers
- TFTP no exigeix autenticació
  - Possible forat de seguretat !!
- Servidor tipus *nowait* (1 sola transferència a la vegada)
- La majoria d'errors provoquen un acabament de la connexió:
  - No es pot satisfer la petició:
    - No es troba el fitxer, Accés denegat, No existeix l'usuari
  - Paquet construït erròniament
  - Pèrdua del dispositiu durant la conversa

## 6.3. FTP

### TFTP: Trivial File Transfer Protocol

Utilitza el UDP com mecanisme de transport. **No és fiable**

- Mida màxima: 512 bytes
- Es confirma cada paquet

Utilitzat per a estacions sense disc (RARP, BOOTP)

Típic en descàrrega d'imatges en equips (routers, etc)

Diferents modes de transferència

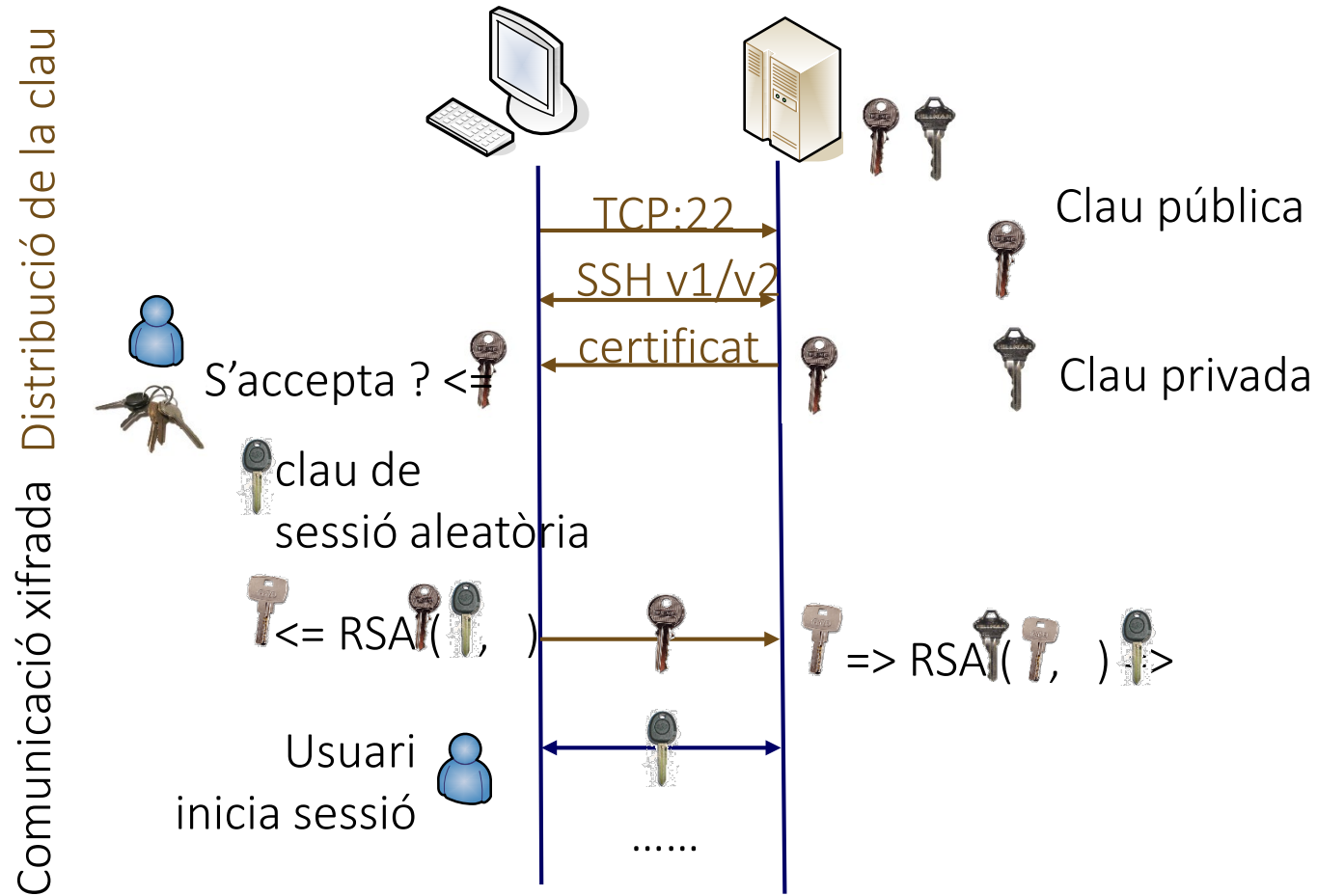
- netascii: text
- octet: format binari
- mail: dades enviades a un usuari en lloc de a un fitxer

Interfície d'usuari: *tftp host*

- put, get, status, ascii, binari

# 6.3. FTP

sftp (Secure ftp) millora i securitza el ftp



## 6.4. Correu electrònic

### Conceptes bàsics sobre correu electrònic

*E-mail*: probablement el servei “estrella” de la xarxa

Evolució dels primers sistemes de comunicació entre usuaris d'un mateix ordinador

- Ordre **mail** de Unix
- Permet enviar missatges de text que queden en la bústia (*mailbox*) de l'usuari receptor

1971: Ray Tomlinson escriu el primer protocol de *transferència* de correu entre hosts Unix

- Adreça = nom\_usuari + nom\_host
- Separació per @

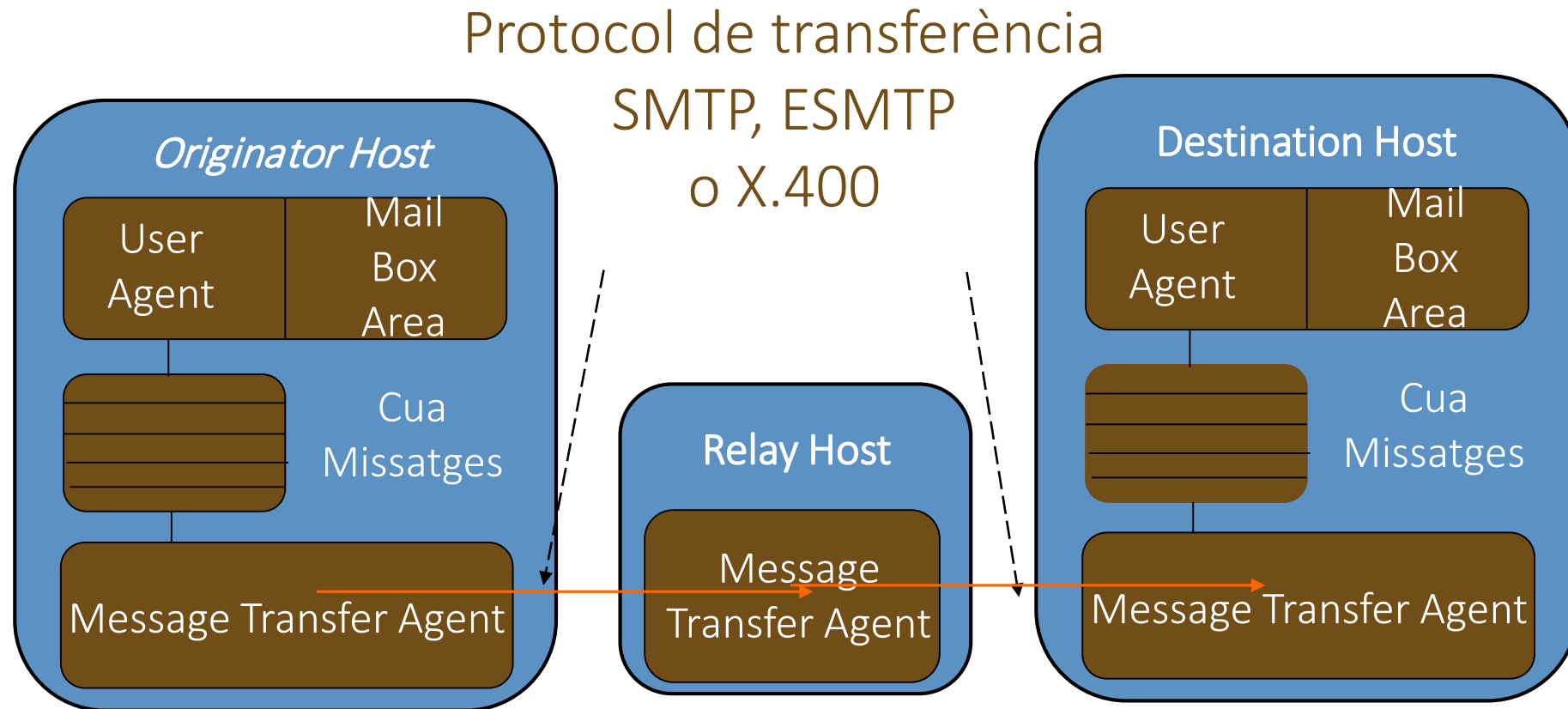
## 6.4. Correu electrònic

### Elements d'un sistema de correu electrònic

- **User Agent (UA) o client e-mail:** que ens permet consultar, contestar i editar el correu
- **Missatge:** Dues parts, sobre i contingut, amb un format determinat
- **Bústia o *mailbox*:** on es guarden el missatges rebuts o per enviar
- ***Message Transfer Agent (MTA)*:** elements encarregats de transportar el correu per la xarxa
- ***Store-and-forward*:** tècnica de transferència de missatges. El missatge va passant per MTAs que el van reenviant fins que arriba a la bústia de destí

## 6.4. Correu electrònic

### Elements d'un sistema de correu electrònic





## 6.4. Correu electrònic

### Protocols de correu electrònic

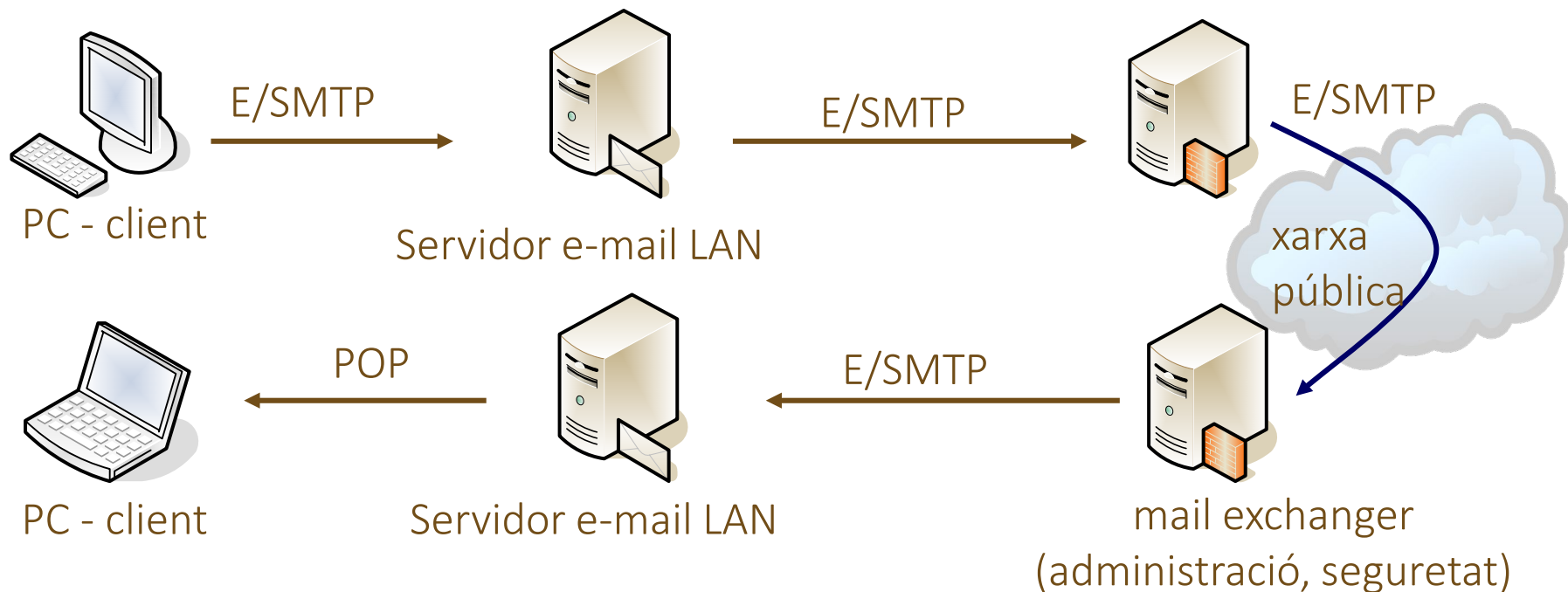
- Standard for the Format of ARPA Internet Messages (RFC 822)
  - Descriu el format dels missatges
- **SMTP** (Simple Mail Transfer Protocol):
  - Servidor clàssic de transferència de correu (RFC 821)
  - Transferència de fitxers senzills (7 bits/caràcter) utilitzant NVT en una sessió telnet
- **ESMTP** (Extended SMTP):
  - Transferència de dades més complexes (8 bits per caràcter)
- **MIME** (Multipurpose Internet Mail Extensions):
  - Extensions multimèdia pels *attachments*
- **POP3** (Post Office Protocol):
  - Accés a la bústia des d'un ordinador personal (no un host)
- **IMAP4** (Internet Access Protocol):
  - Evolució del POP

## 6.4. Correu electrònic

### Model Store & Forward

Avantatges del model Store and Forward

- Utilització del correu electrònic amb seguretat
- Estalvi enviant e-mail en bloc a hores determinades
- Traducció de formats



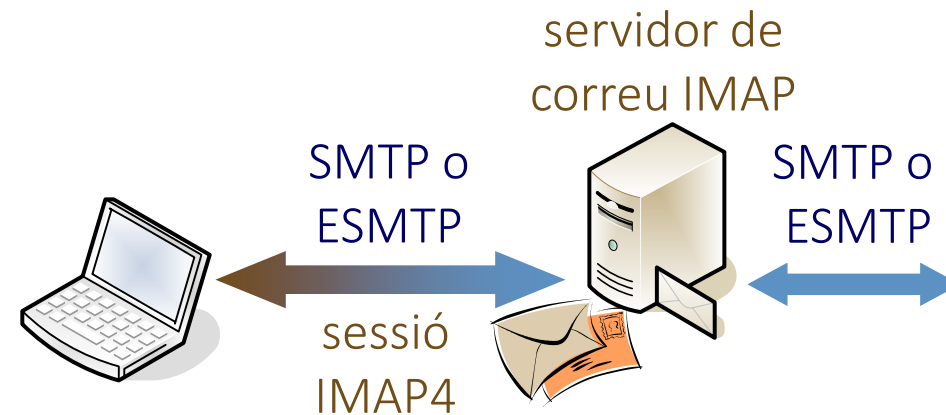
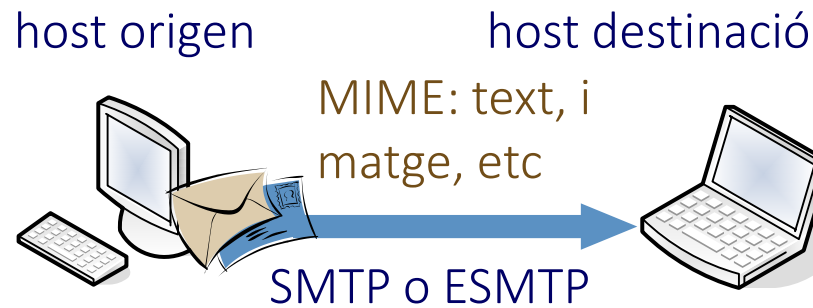
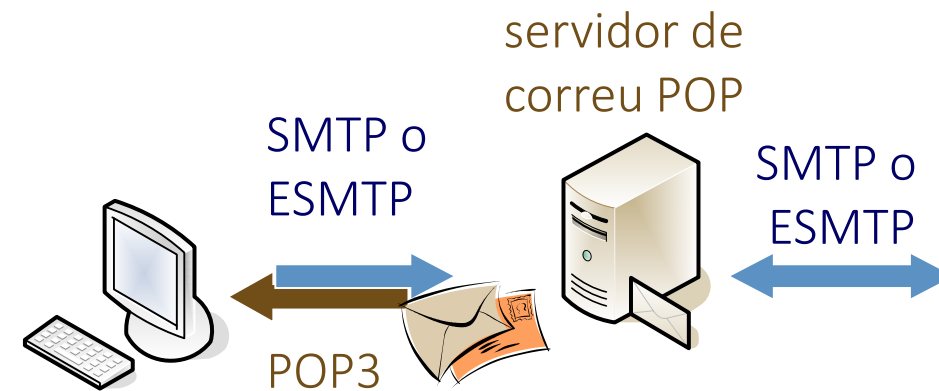
# 6.4. Correu electrònic

## Transferències de correus

*Transferència host-host*



*Transferència host-terminal*



## 6.4. Correu electrònic

### Main relay – servidors de correu a la UPC

```
C:\>nslookup
Servidor predeterminado:  dns1.red.retevision.es
Address:  62.81.16.129

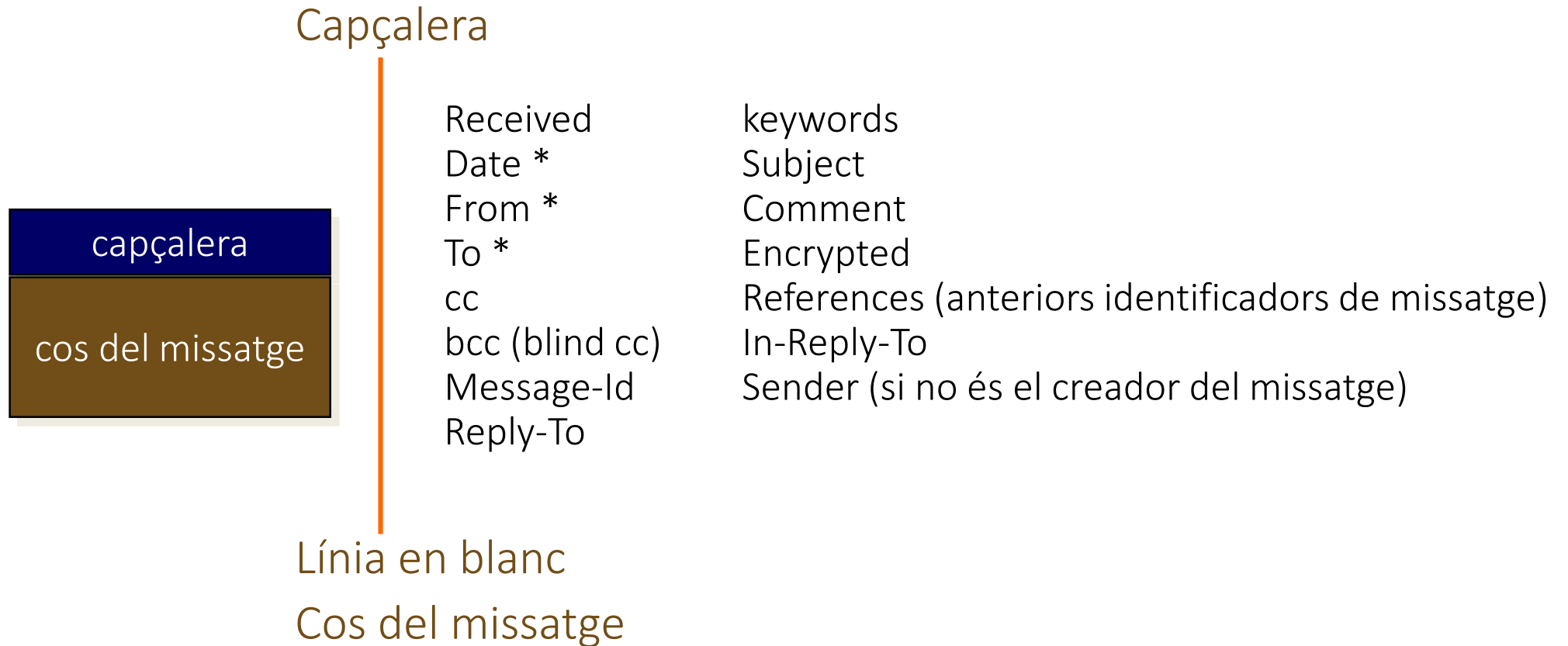
> set type=mx
> upc.es
Servidor:  dns1.red.retevision.es
Address:  62.81.16.129

Respuesta no autoritativa:
upc.es  MX preference = 10, mail exchanger = dukas.upc.es
upc.es  MX preference = 20, mail exchanger = moneo.upc.es
upc.es  MX preference = 30, mail exchanger = mail.rediris.es

upc.es  nameserver = euler.upc.es
upc.es  nameserver = backus.upc.es
dukas.upc.es  internet address = 147.83.2.62
moneo.upc.es  internet address = 147.83.2.91
mail.rediris.es internet address = 130.206.1.11
euler.upc.es  internet address = 147.83.2.10
backus.upc.es internet address = 147.83.2.3
> exit
```

## 6.4. Correu electrònic

### Parts d'un correu electrònic



## 6.4. Correu electrònic

### Exemple d'una capçalera de correu

**Received:** from diable.upc.es [147.83.98.7]by mat.upc.es (8.7.6/8.7.3) with ESMTP id VAA13564 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:55:56 GMT

**Received:** from mail.tiip.edu[192.208.46.30] by diable.upc.es (8.8.6/8.8.6) with ESMTP id VAA01230 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:56:36 +0100 (MET)

**Received:** by mail.tiip.edu with SMTP (Microsoft Exchange Server Internet Mail Connector Version 4.0.996.35) id <01BCE480.75FDDED0@mail.tiip.edu>; Wed, 29 Oct 1997 15:36:44 -0500

**Date:** Wed, 29 Oct 1997 15:36:39 +0200 (MET DST)

**From:** David Rincon <drincon@tiip.edu>

**To:** Rafael Vidal <rvidal@mat.upc.es>

**Subject:** Re: Microsoft i IPv6

**In-Reply-To:** <393BD954.82FF42E7@mat.upc.es>

**Message-ID:** <Pine.GSO.4.10.10006051848230.20185-100000@tn-nit.tiip.edu>

**MIME-Version:** 1.0

**Content-Transfer-Encoding:** QUOTED-PRINTABLE

**Content-Type:** TEXT/PLAIN; charset=ISO-8859-1

**Status:** U

**X-UIDL:** 726e36d36e1699eef6b8bb936c02c013

## 6.4. Correu electrònic

### SMTP: Simple Mail Transfer Protocol

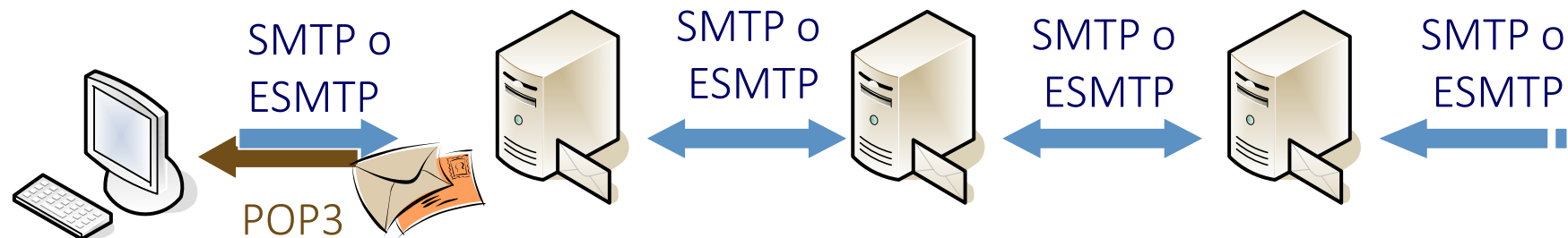
- RFC 821
- Defineix un mecanisme de transferència de correu entre 2 màquines
  - Màquina origen
    - Pot ser un client o bé un altre host (relay de correu)
  - Màquina destinació
    - Ha de ser un servidor; no pot ser un client
  - Per a cada transferència, s'estableix una sessió
    - Una sessió pot constar de diversos correus
    - Protocol TCP, port 25

## 6.4. Correu electrònic

### Registre de temps, identificador i pas en SMTP

Els correus tenen un registre de hosts intermedis i *timestamps* (temps de pas)

- Cada vegada que un missatge passa per un MTA s'hi afegeix una marca temporal del seu pas, *timestamp*
- El correu conté tot el camí recorregut
- L'identificador del missatge el posa el primer MTA que travessa el missatge i fa **únic** el missatge





## 6.4. Correu electrònic

### Registre de temps, identificador i pas en SMTP

**Received:** from diable.upc.es [147.83.98.7]by mat.upc.es (8.7.6/8.7.3) with ESMTP id VAA13564 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:55:56 GMT

**Received:** from mail.tiip.edu[192.208.46.30] by diable.upc.es (8.8.6/8.8.6) with ESMTP id VAA01230 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:56:36 +0100 (MET)

**Received:** by mail.tiip.edu with SMTP (Microsoft Exchange Server Internet Mail Connector Version 4.0.996.35) id <01BCE480.75FDDED0@mail.tiip.edu>; Wed, 29 Oct 1997 15:36:44 -0500

**Date:** Wed, 29 Oct 1997 15:36:39 +0200 (MET DST)

**From:** David Rincon <drincon@tiip.edu>

**To:** Rafael Vidal <rvidal@mat.upc.es>

**Subject:** Re: Microsoft i IPv6

**In-Reply-To:** <393BD954.82FF42E7@mat.upc.es>

**Message-ID:** <Pine.GSO.4.10.10006051848230.20185-100000@tn-nit.tiip.edu>

**MIME-Version:** 1.0

**Content-Transfer-Encoding:** QUOTED-PRINTABLE

**Content-Type:** TEXT/PLAIN; charset=ISO-8859-1

**Status:** U

**X-UIDL:** 726e36d36e1699eef6b8bb936c02c013

## 6.4. Correu electrònic

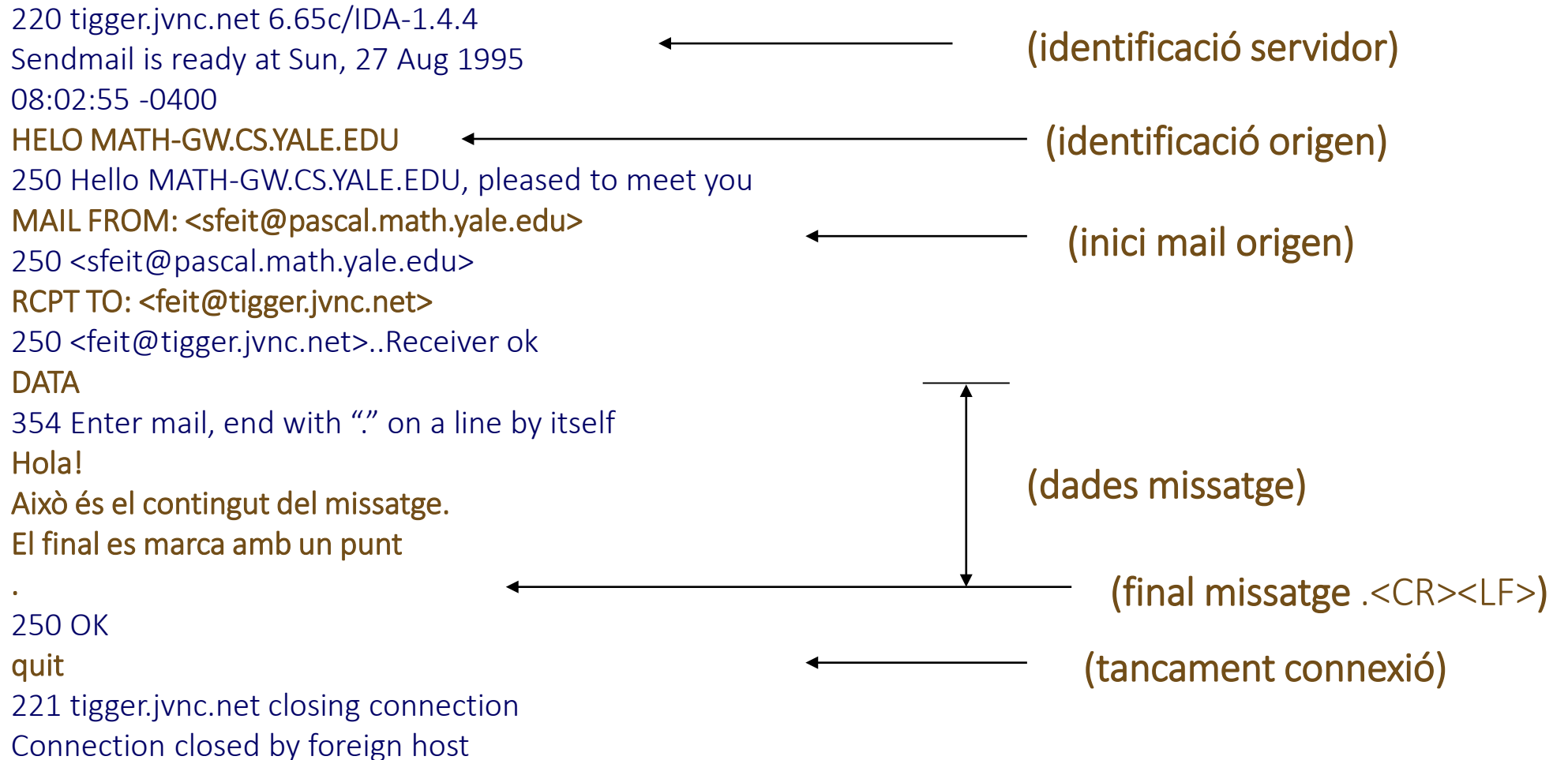
### Ordres i seqüència del protocol SMTP

#### Ordres del protocol

- HELO, MAIL, RCPT, DATA, QUIT, TURN
- Seqüència del protocol
  - Origen envia un paquet inicial, receptor envia identificació de host
  - Origen envia identificació de host (HELO)
    - Clients i servidors ESMTP es reconeixen per EHLO en comptes de HELO
  - Origen envia ID de l'usuari que genera el missatge (MAIL FROM:)
  - Origen identifica els receptors (RCPT TO:)
  - Origen transmet les dades de capçalera i text (MAIL, DATA)
  - Origen transmet ".<CR><LF>" (final de correu)
  - Es repeteix el procés per altres missatges o es finalitza (QUIT)
  - Possibilitat d'intercanviar papers (TURN)

## 6.4. Correu electrònic

### Exemple d'una sessió SMTP



## 6.4. Correu electrònic

### Extensions del protocol SMTP

SMTP i RFC 822 es van dissenyar per missatges textuais

- Com afegim altres continguts (vídeos, imatges) ?
- Con transportar caràcters per sobre d'ASCII 127 ?
- Solució: Modificar protocol transport i format dels missatges
  - *ESMTP, Extended SMTP, RFC 1425 / 1869*
    - Clients i servidors ESMTP es reconeixen per EHLO en comptes de HELO
  - *MIME, Multipurpose Internet Mail Extension, RFC 1521*
    - Definició d'extensions que suporten formats multimèdia
    - Poden transferir-se eficientement amb ESMTP (i no tant amb SMTP)
    - Es delimita cada part del missatge amb una marca
    - Facilita la crida a aplicacions de reproducció

## 6.4. Correu electrònic

### MIME: Multipurpose Internet Mail Extension

- Tipus de dades suportades:
  - text
    - plain, richtext (format bàsic), enriched
  - multipart
    - Mixed (diversos formats processats seqüencialment), parallel, digest...
  - application
    - octet-stream (arbitrari), postscript, ...
  - Imatge
    - BMP, JPEG, PIF, GIF...
  - video
    - mpeg, quicktime, avi, ...
  - audio
    - basic, mpeg,

## 6.4. Correu electrònic

### Exemple de MIME

...

MIME-Version: 1.0

Content-Type: MULTIPART/MIXED;

BOUNDARY="-559023410-2110444415-1011024789=:19819"

X-Virus-Scanned: by AMaViS perl-11

X-Mozilla-Status2: 00000000

Això és el cos del correu. Hi ha accents.

[ Part 2.2, "" Text/PLAIN (Name: "amics.txt") 41 lines. ]

[ Not Shown. Use the "V" command to view or save this part. ]

[ Part 2.3, "" Application/POSTSCRIPT 1.9MB. ]

[ Not Shown. Use the "V" command to view or save this part. ]

[ Part 2.4, "" Application/OCTET-STREAM (Name: "a.mpg") 939KB. ]

[ Cannot display this part. Press "V" then "S" to save in a file. ]

## 6.4. Correu electrònic

### Post Office Protocol – POP3

Protocol més estès per a la descàrrega del correu

- Definit a RFC 1939
  - Anirà sent substituït per l'IMAP4, RFC 2060
- Utilitza TCP, port 110
- Disposa de mecanismes d'autenticació
  - Ordres USER i PASS, més comú: Identificador i clau es passen en clar
  - Ordre APOP, opcional: Més sofisticat, no s'exposa la clau
  - Ordre AUTH, extensió (RFC 1734): Utilitza alguna de les opcions d'autenticació criptogràfica d'IMAP4

Servidor suporta múltiples connexions, cadascuna amb permisos de R/W de la seva bústia

## 6.4. Correu electrònic

### Post Office Protocol – POP3

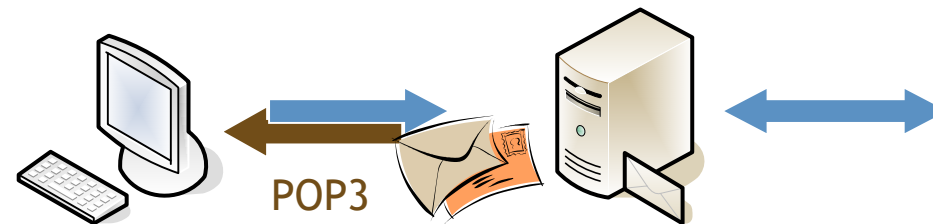
Ordres bàsiques:

- Autenticació

USER name, PASS string, APOP name digest, AUTH

- Diàleg

STAT, LIST [msg], RETR msg, DELE msg, NOOP, RSET, QUIT, TOP msg n, UIDL [msg]





## 6.4. Correu electrònic

### Exemple de diàleg POP3

```

C:                                     (Inicia connexió TCP pel port 110)
S:                                     (accepta connexió TCP)
S: +OK POP3 server ready
C: USER nom_usuari
S: +OK
C: PASS contrasenya
S: +OK user authenticated
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S:                                     (contingut del missatge 1, capçaleres, línia nul·la i cos)
S: .
C: DELE 1
S: +OK message 1 deleted
C:
C: RETR 2
S: +OK 200 octets
S:                                     (contingut del missatge 2, capçaleres, línia nul·la i cos)
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK POP3 server signing off
S:
C:                                     (es tanca el final del servidor de la connexió TCP)
                                     (es tanca el final del client de la connexió TCP)

```

## 6.4. Correu electrònic

### Accés d'una bústia des de múltiples clients amb POP3

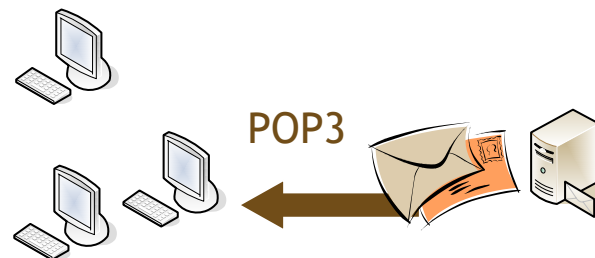
Alguns clients permeten deixar el correu al servidor

- Servidor amb suport ordre UIDL : retorna llista missatges amb amb identificador únic per bústia i totes les sessions

Per saber si un missatge s'ha llegit o no s'ha de tenir un registre a la bústia local

- POP3 no permet fer-ho al servidor!

Cal que el client esborri els missatges del servidor a partir d'un cert temps (7- 30 dies per exemple)



## 6.4. Correu electrònic

### Exemple de diàleg POP3 amb APOP i UIDL

```

C:                                     (Inicia connexió TCP pel port 110)
S:                                     (accepta connexió TCP)
S: +OK POP3 server ready <1234.697170952@guys.com>
C: APOP asmith c9dcb935ce1d21d02afdebcdbd9cb1d5a
S: +OK user authenticated
C: UIDL
S: +OK 2 message (320 octets)
S: 1 XXX001                           (+ de 30 dies)
S: 2 XXX079                           (missatge nou)
S: .
C: RETR 2
S: +OK 200 octets
S:                                     (contingut missatge 2, capçaleres, línia nul.la i cos)
S: .
C: DELE 1                             (client esborra automàticament)
S: +OK message 1 deleted
C: QUIT
S: +OK POP3 server signing off
S:                                     (es tanca el final del servidor de la connexió TCP)
C:                                     (es tanca el final del client de la connexió TCP)
    
```

## 6.4. Correu electrònic

### IMAP: Internet Mail Access Protocol

#### Problemes del POP

- Els correus es “baixen” al client (i s’esborren del servidor)
- Si s’accedeix a la mateixa bústia des de diferents terminals, els missatges es reparteixen
- És difícil d’integrar amb noves interfícies com WWW

#### Solució: IMAP

- 1986, Stanford University
- No ha estat fins ara (1995-2000) que ha tingut èxit
- Versió actual: IMAP4rev1 (1994) descrit a RFC 2060
- Permet *webmail*

## 6.4. Correu electrònic

### IMAP permet webmail

**WebMail**

INBOX Bústia: INBOX | 0 nous, 0 recents IMP, Version 2.2.5

41 - 60 de 60 Missatges

Seleccionar tots | Treure selecció | Eliminar | Recuperar | Ocultar esborrats | Buidar paperera

	Data	De	Assumpte	Mida
<input type="checkbox"/>	2002-05-20	Lluís Casals	practica 1 d'alba sancho, intercampus.	2 kb
<input type="checkbox"/>	2002-05-21	Anna Cutillas/ICE/UPC	Programa d' acció "Utilització de les enquestes pe...	3 kb
<input type="checkbox"/>	2002-05-23	Sergi Martinez-Oliver/UPC	proves del circuit de diagrama d'ull	3 kb
<input type="checkbox"/>	2002-05-27	JOSE A MEMBRIVE/EUPBL/UPC	generadores	28 kb
<input type="checkbox"/>	2002-05-27	Jordi Mataix/MAT/UPC	complement mobiliari	33 kb
<input type="checkbox"/>	2002-05-31	EPSC Difusio	[Idea 4.01] Activitat Acadèmica curs 2001/2002	18 kb

## 6.4. Correu electrònic

### IMAP VS POP

#### POP

Un sol servidor, un sol inbox



Un sol terminal: còpia local dels missatges



Connexió mínima: baixar-se els missatges i pujar els nous



#### IMAP

Es pot treballar amb diferents bústies i servidors



Missatges queden al servidor (accés desde diferents terminals)



Connexió contínua, dependència de la connexió



## 6.5. Notícies

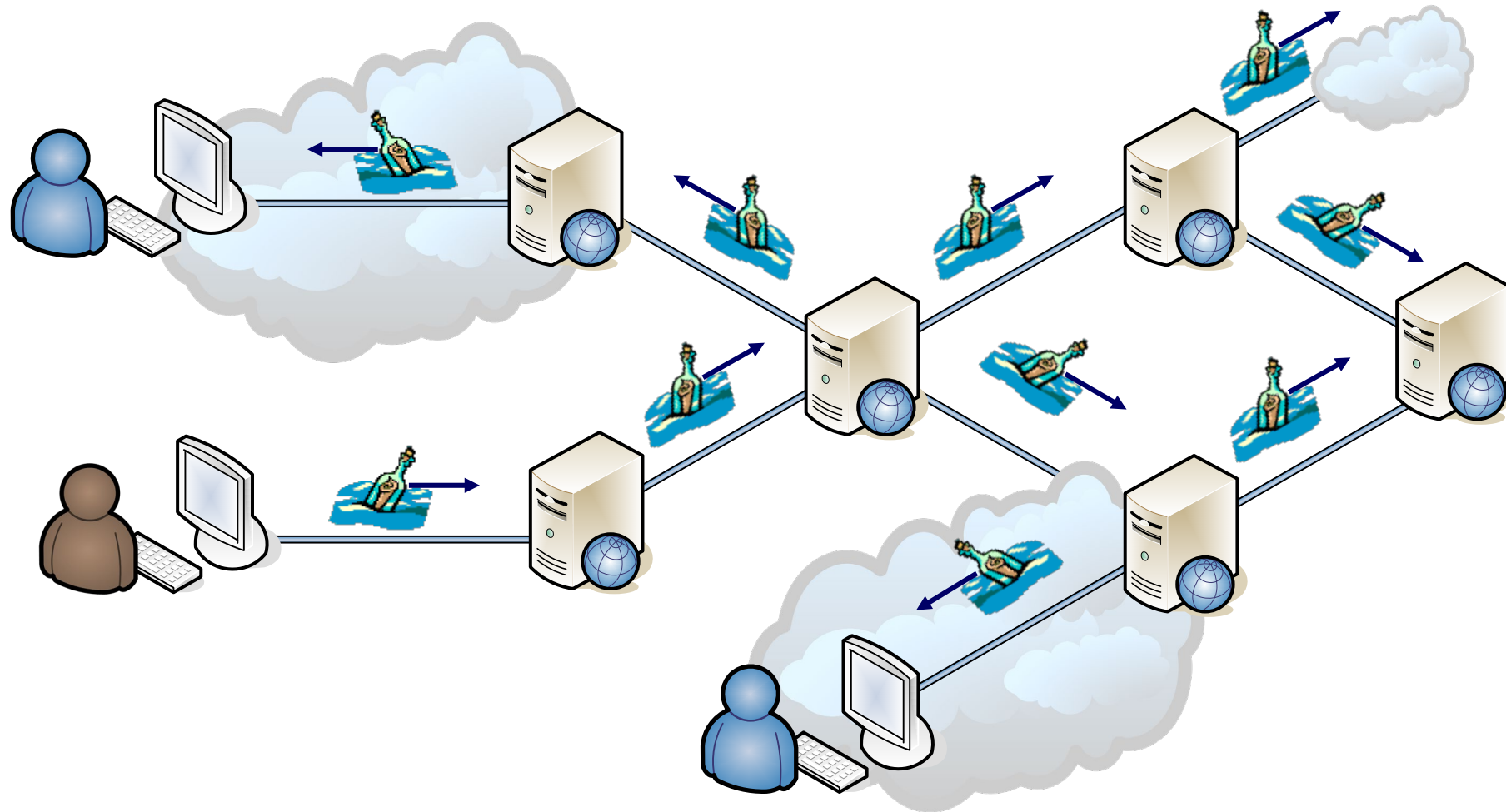
### New

Fòrums de discussió monogràfics accessibles per e-mail

- Protocol NNTP (*News Network Transfer Protocol*)
- Exemple del caràcter distribuït d'Internet
  - Cap node central.
  - Els servidors de news distribueixen els missatges mitjançant un procés de “dispersió” cap als servidors veïns
- Organitzades en jerarquies.
  - Globals:
    - alt (temes alternatius), comp (temes relacionats amb ordinadors), soc (societat), news (gestió dels grups), sci (ciència), etc.
  - Locals:
    - rediris, es, upc, ieee, etc.
- Els noms dels grups indiquen quin és el tema d'interès  
 comp.os.ms-windows.networking.tcp-ip, upc.tertulia, rediris.anuncios.congresos...

## 6.5. Notícies

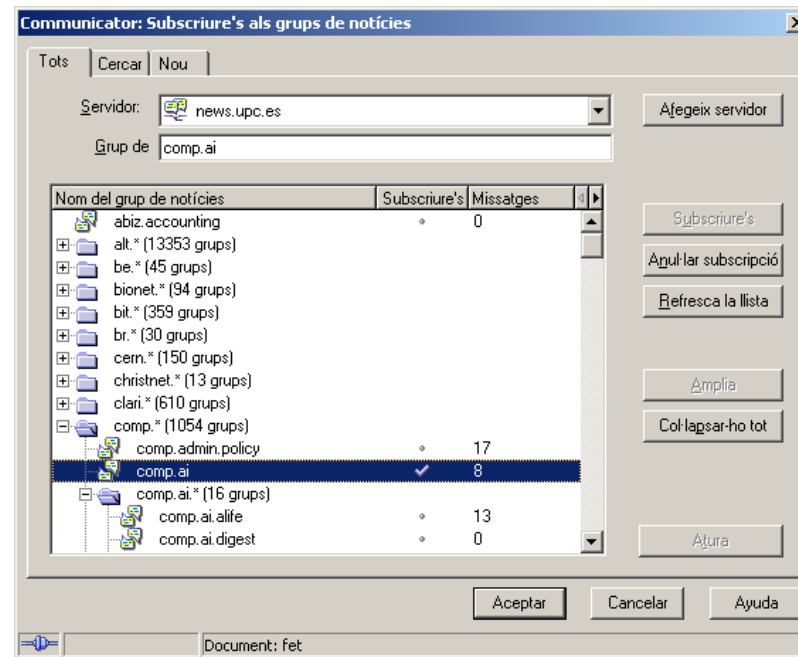
### Esquema de funcionament de les News





## 6.5. Notícies

### Exemple de client de News - Netscape



Pràcticament no s'utilitza  
 RSS l'ha substituït (evolució XML i www)

## 6.6. Conclusions parcials

### Conceptes bàsics

Cal distingir entre servei i aplicació

- Servei: concepte abstracte
- Aplicació: peça de software que dóna el servei

Hi ha serveis i aplicacions orientat i no orientats a connexió

Hi ha 2 architectures bàsiques:

- Client - Servidor
- Peer – to – Peer

Serveis bàsics en xarxes TCP/IP

- DNS: obté la correlació IP – Nom
- Telnet: permet controlar remotament un equip
- FTP: permet transmetre fitxers entre equips
- Cooreu electrònic: permet enviar missatges entre usuaris (POP/IMAP)
- News: permet crear forums de discussió sobre temes

## 6. Serveis

---

### 6.7. Tecnologies web

## 6.7.1. Tecnologies Web

### World Wide Web (WWW)

1989: Tim Berners-Lee el crea al CERN com a sistema d'informació global compartida

- Objectiu: Superar dificultats d'utilització sistemes existents (ftp, archie, gopher...)

Després, desenvolupat pel W3C (*World Wide Web Consortium*)

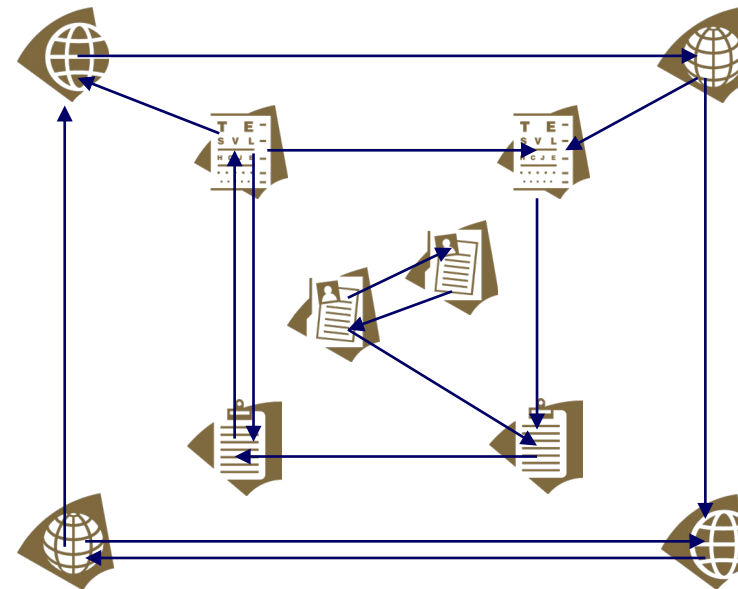
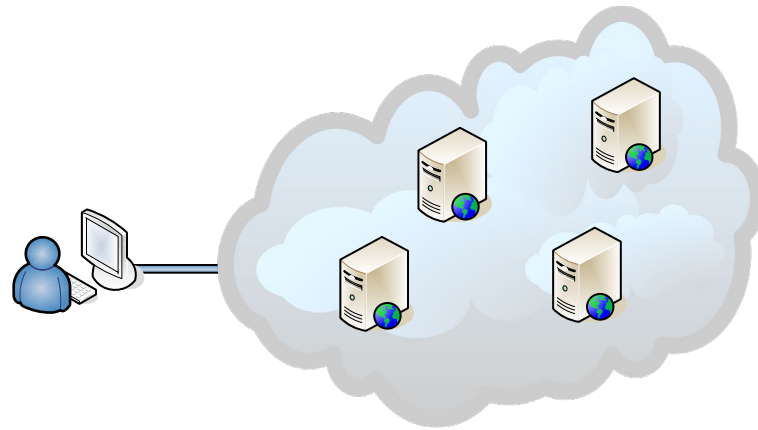
<b>Domini</b>	<b>Activitat</b>	<b>Àrees compreses</b>
<b>Interfície d'usuari</b> (interacció home-màquina)	Millorar la percepció de l'usuari de la informació	HTML, gràfics 3D, internacionalització, fulles d'estils, accés mòbil ...
<b>Tecnologia i societat</b> (interacció home-home)	Permetre aplicacions orientades a la societat.	Iniciativa per a la signatura digital, pagaments electrònics, PICS, seguretat i col.laboració, comerç electrònic ...
<b>Arquitectura</b> (interacció màquina-màquina)	Permetre noves aplicacions distribuïdes.	protocols (HTTP), HTTP-NG, multimèdia sincronitzat, XML, Televisió i Web, caracterització de la Web

## 6.7.1. Tecnologies Web

### Filosofia de treball

Teranyina formada per documents enllaçats sense necessitar saber on són.

- Només necessitem saber el seu identificador
- Enllaços: hipertext



## 6.7.1. Technologies Web

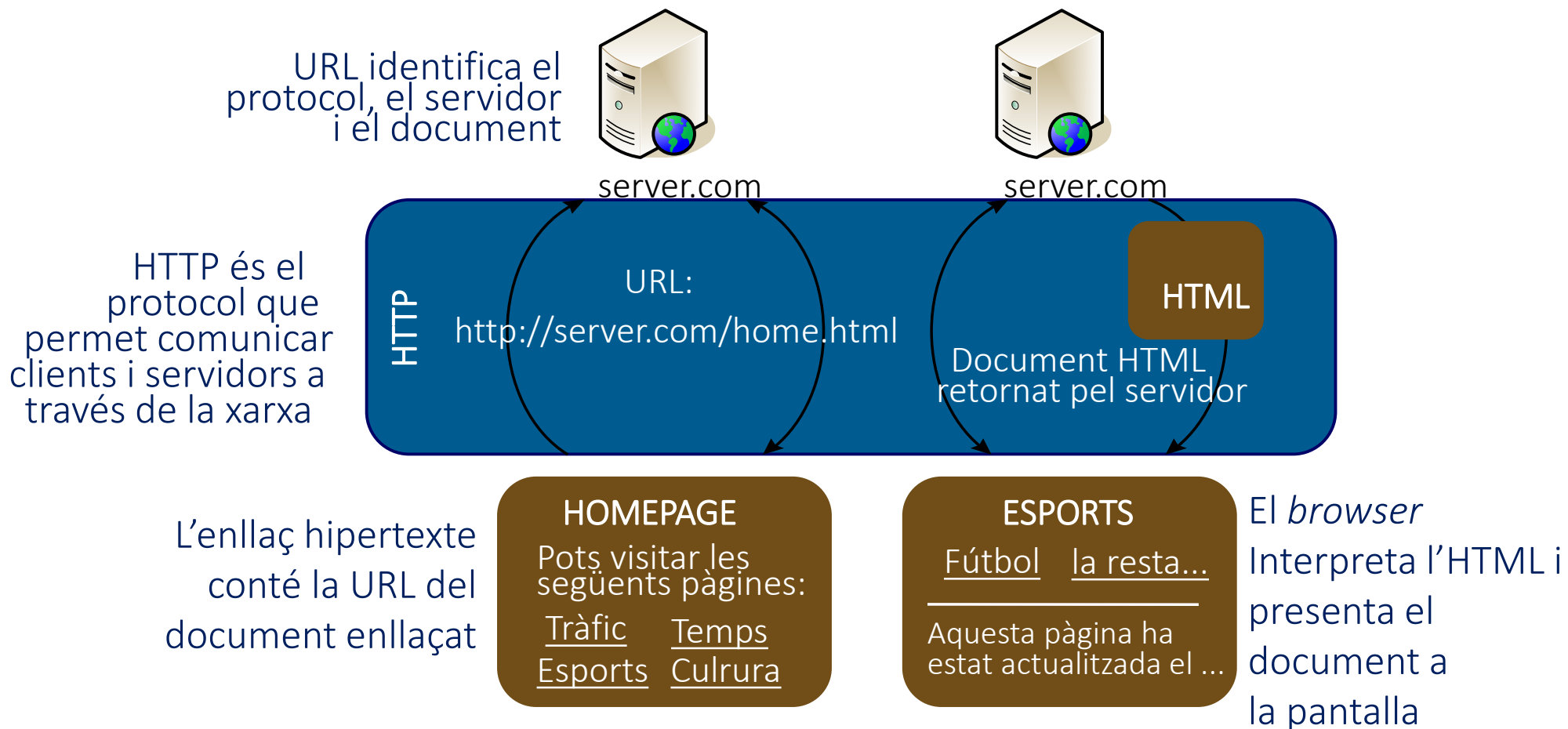
### Famílies de protocols

#### Protocols del servei WWW

- **HTML:** *HyperText Markup Language*
  - Format representació de la informació
- **HTTP:** *HyperText Transfer Protocol*
  - Protocol transport de la informació
- **URL:** *Uniform Resource Locator*
  - Identificació de la informació

# 6.7.1. Tecnologies Web

## Funcionament bàsic



## 6.7.2. HTML

# HyperText Markup Language (HTML)

Neix amb la WWW

Apte per a representar tot tipus de documents

- Ha suposat una veritable revolució a Internet

Estandardització: W3C

- Estàndard actual: HTML 4.0.1 (desembre 1999)

Està evolucionant cap al XML (*eXtended Markup Language*)

- W3C actualment recomana XHTML 1.0 (gener 2000)
- XHTML: *the eXtensible HyperText Markup Language*



## 6.7.2. HTML

### HTML és un llenguatge SGML

És un **llenguatge** (*Language*)

Utilitza **etiquetes** per a donar el format al document (*Markup*)

El format es dóna **basant-se en el contingut** no en l'aparença del document (*Generalized*)

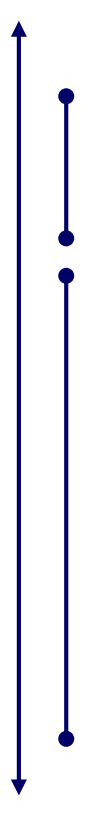
- Flexibilitat en la presentació: s'adapta a les possibilitats de la plataforma

Àmpliament **acceptat** i en el cas de l'HTML, **no és propietari** (*Standard*)

- Informació necessària per desenvolupar és pública

## 6.7.2. HTML

### Exemple de codi HTML



```

<HTML>
<HEAD>
<TITLE> Exemple</TITLE>
</HEAD>
<BODY>
<H1> Llista amb un enllaç al tercer element</H1>
<UL>
<LI> Primer element llista
<LI> Segon element de la llista
<LI> <A HREF="http://www.upc.es"> Tercer element</A>
</UL>
</BODY>
</HTML>

```

## 6.7.3. URL

### Estructura genèrica de les URL

- Definida a la RFC 1738
- Extensió del concepte de nom de fitxer

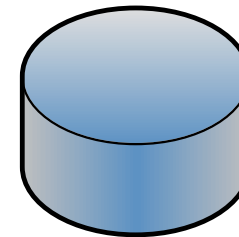
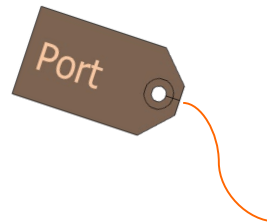
`http://www.host.edu:1234/path/subdir/fixer.ext`

Servei

Servidor



Port

Nom del fitxer i detalls del recurs



## 6.7.3. URL

### Accés a diferents recursos mitjançant les URL

-  HTTP: <http://www.catacrac.es>
  -  FTP: <ftp://soc.alegria.com/>
  -  News (NNTP): [news: rec.motorcycles.harley](news:rec.motorcycles.harley)
  -  e-mail (SMTP): <mailto:rvidal@mat.upc.es>
  -  Fitxers remots: <file://maite168/c:/docs/capitol3.doc>
  -  Fitxers locals: <file://c:/docs/capitol3.doc>
- URLs parcials: Si dintre del document  
<http://www.upc.es/HomePage.html>  
 vull apuntar a  
<http://www.upc.es/departaments.html>,  
 n'hi ha prou amb [departaments.html](http://www.upc.es/departaments.html)

## 6.7.4. HTTP

# HyperText Transfer Protocol (HTTP)

HTTP 1.1 estàndard, RFC 2616

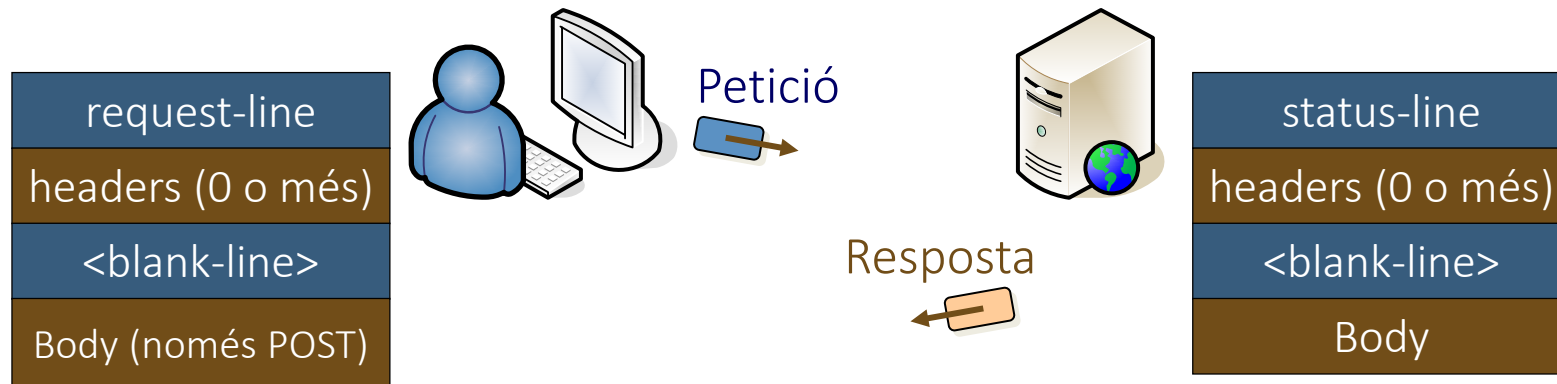
- S'ha definit un sistema d'autenticació, RFC 2617
- Utilitza TCP, típicament el port 80
- Enviament de peticions i respostes en format text
- Diàleg sessió molt senzill

Connexió	Establiment de la connexió per part del client al servidor (TCP/IP, port 80)
Petició	El client envia al servidor el missatge de petició
Resposta	El servidor envia el missatge de resposta al client
Tancament	El servidor indica la fi tancant la connexió

La majoria del tràfic d'Internet correspon a HTTP !!

## 6.7.4. HTTP

### Format de les peticions i respostes HTTP



On tenim que

- Request-line: *request request URI HTTP version*
- Status-line: *HTTP version 3 digit response code human readable response phrase*
- Request: GET, HEAD, POST, ...

## 6.7.4. HTTP

### Descripció de les ordres HTTP (request)

#### GET

- Petició que retorna la informació que és identificada per *request-URI* (URL)

#### HEAD

- Similar a GET. Només retorna els *headers* del servidor, però no el contingut (*body*) del document especificat.
- S'utilitza per testejar un enllaç d'hipertext, la seva accessibilitat i recent modificació.

#### PUT

- Demana al client que accepti i guardi un recurs amb la *request-URI* que demana el client.

#### DELETE

- A l'inrevés de l'anterior, per a esborrar un recurs

## 6.7.4. HTTP

### Descripció de les ordres HTTP (request)

#### POST

- Petició utilitzada per a per enviar correu electrònic, *news*, o formularis que poden ser omplerts per un usuari interactiu.
- És l'única petició que envia un contingut (*body*) amb la petició.
- Necessita un camp pel *header Content-Length* per especificar la longitud del contingut.

#### OPTIONS

- Per a preguntar al servidor per les capacitats d'un recurs determinat o del servidor en general

#### TRACE

- Utilitzat per *debugging* a nivell d'aplicació



## 6.7.4. HTTP

### Codis numèrics de resposta (3 digit response code)

Response	Description
1yz	Informational. Not currently used.
200	Success. OK, request succeeded.
201	OK, new resource created (POST command).
202	Request accepted but processing not completed.
204	OK, but no content to return
301	Redirection; further action need be taken by user agent. Requested resource has been assigned a new permanent URL.
302	Requested resource resides temporaly under a different URL.
304	Document has not been modified (conditional GET).
400	Client error. Bad request.
401	Unauthorized; request requires user authentication.
403	Forbidden for unspecified reason.
404	Not found.
500	Server error. Internal server error.
501	Not implemented.
502	Bad gateway; invalid response from gateway or upstream server.
503	Service temporaly unavaible.

## 6.7.4. HTTP

### Capçaleres HTTP 1.0 (headers)

Nom del Header	Request	Response	Body
Allow			*
Authorization	*		
Content-Encoding			*
Content-Length			*
Content-Type			*
Date	*	*	
Expires			*
From	*		
If-Modified-Since	*		*
Last-Modified			
Location		*	
MIME-Version	*	*	
Pragma	*	*	
Referer	*		
Server		*	
User-Agent	*		
WWW-Authenticate		*	

## 6.7.4. HTTP



Petició



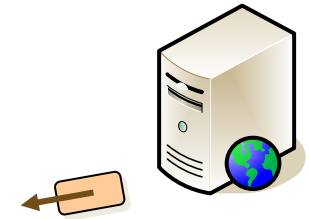
```
GET /foto.gif HTTP/1.0
From: telrvf@maite.upc.es
(línia en blanc)
```

Exemple d'accés telnet a un servidor HTTP

- telnet www.upc.es 80
- GET / HTTP/1.0 (intro dos cops)

## Exemple de transacció HTTP

Resposta



```
HTTP/1.0 200 OK
Date: Sat, 7 Feb 97 17:49:25 GMT
Server: NCSA/1.3
MIME version: 1.0
Content – type: imatge – gif
Last – modified: Mon, 14 Nov 96 12:04:22 GMT
Content – length: 22700
(linea en blanc)
```

(s'envien els 22700 bytes de foto.gif en format binari)  
 (finalment, el servidor taca la connexió TCP)

## 6.7.4. HTTP

### Deficiències HTTP 1.0 solucionades per HTTP 1.1

#### Múltiples connexions TCP/IP

- TCP/IP està pensat per mantenir fluxos de bits durant un període extens de temps.
- Utilitzant una connexió TCP/IP per a cada missatge HTTP (les peticions ocupen molts pocs bits i les respostes en general també), HTTP interactua molt malament amb el disseny de TCP/IP

#### Caché

- HTTP 1.0 simplement permet l'ús del caché, però no descriu com el caché interactua amb els clients o els servidors.

#### Noms dels Hosts

- HTTP 1.0 no permet que una mateixa adreça IP sigui utilitzada per a servidors diferents que són en una mateixa màquina (host)
- Obliga a que tinguin entrades diferents al DNS (*Domain Name System*) ⇒ gran demanda d'adreces IP

## 6.7.4. HTTP

### Altres millores aportades per HTTP 1.1

Negociació del contingut

- Un servidor pot tenir diferents representacions d'un recurs  
⇒ enviar al client la més adequada

Autenticació sense enviar el password en clar

- Utilitza tècnica de secret compartit (MD5)

Descàrrega d'un determinat rang de bytes

- Es pot demanar l'enviament d'una part d'un recurs

## 6.7.4. HTTP

### Evolució futura del protocol HTTP

HTTP 1.1 estandarditzat el Juny de 1999.

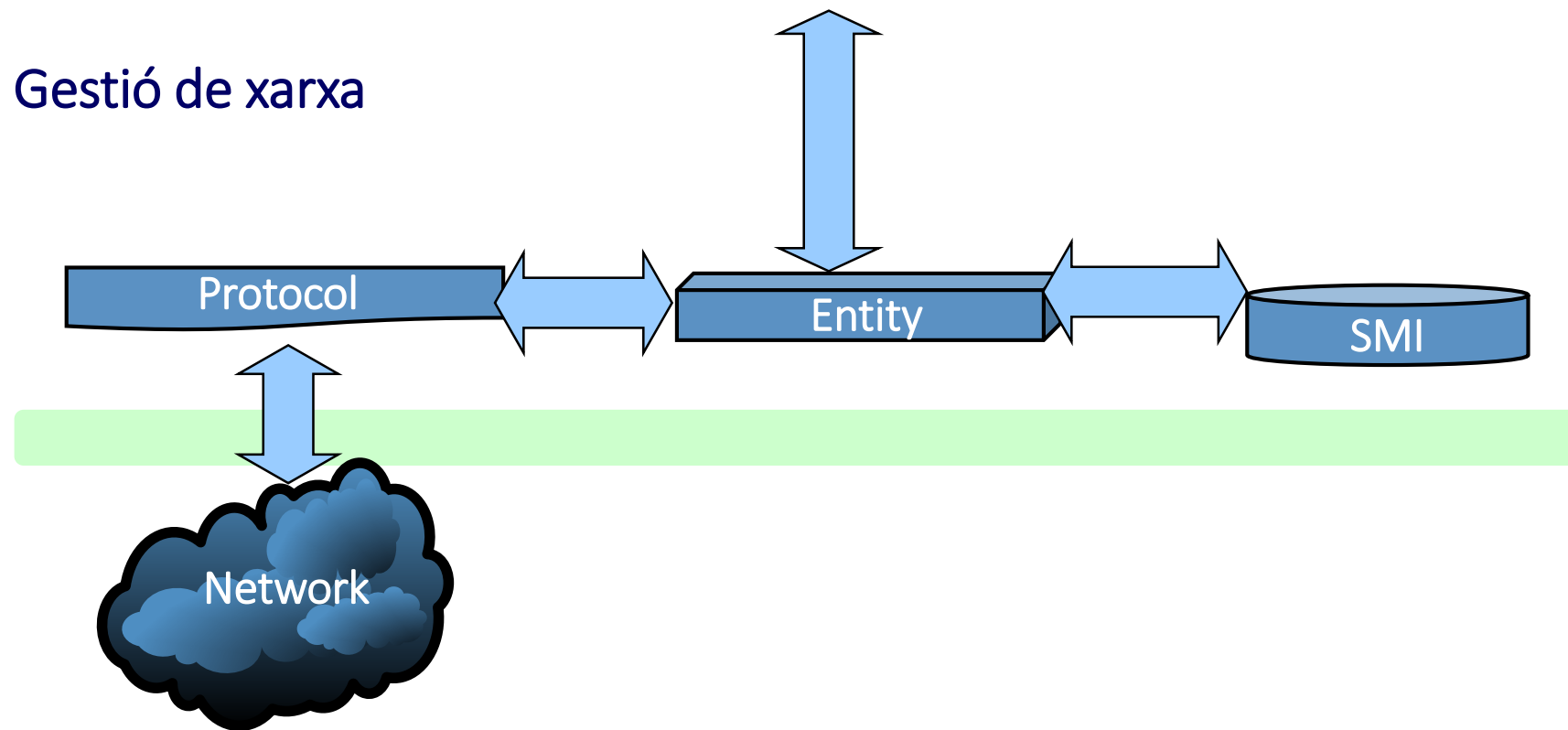
- Ja és àmpliament suportat

HTTP-NG: fer un nou HTTP més simple, modular, amb estructura de capes

- HTTP 1.1 més aviat complicat i voluminós
- MUX: multiplexar informació de diferents peticions en una connexió TCP
- Presentació simultània

## 6.8.1. SNMP

SNMP proporciona un entorn de gestió de xarxes

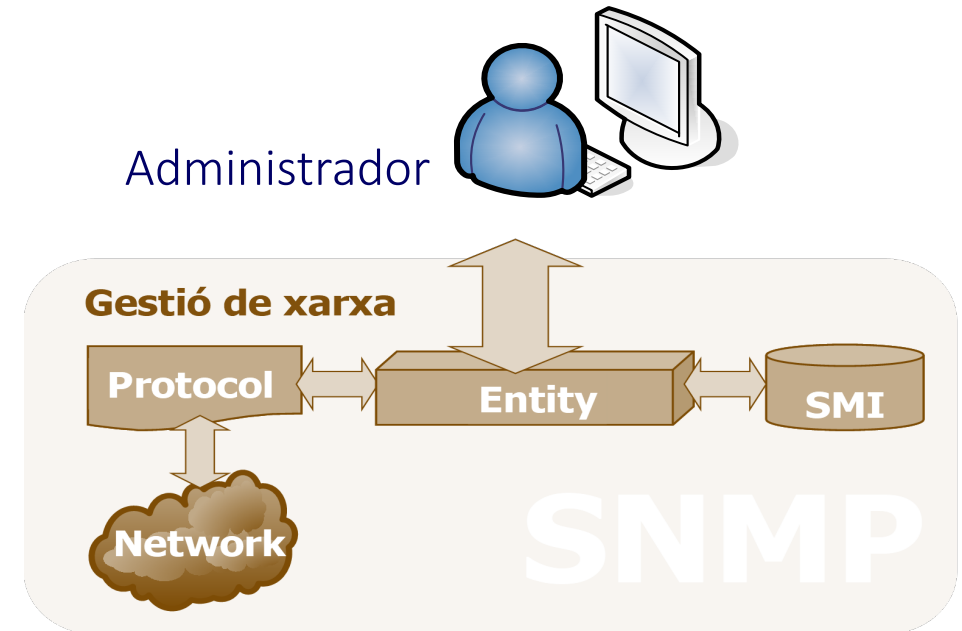


## 6.8.1. SNMP

### Filosofia de treball

Cal que es pugui aplicar...

- A la major escala possible
- Amb la major diversitat d'implementació possible
- Amb la major amplitud de capes de protocols possible
- Amb la major diversitat d'administració que es pugui obtenir





## 6.8.1. SNMP

### Tasques a realitzar

- **Gestió de configuració:** Configuració dels paràmetres de la xarxa gestionada. Gestió centralitzada.
- **Gestió de fallades:** Detecció i, si és possible, correcció de fallides en la xarxa.
- **Gestió d'estadístiques:** Estadístiques sobre la utilització de la xarxa. Permeten definir polítiques d'accés, tarificació, planificació, etc.
- **Gestió de l'eficiència:** Anàlisi de les estadístiques de utilització. Permet descobrir colls d'ampolla.
- **Gestió de la seguretat:** Controlar l'accés als dispositius gestionats, generar alarmes quan es detectin intrusos.

## 6.8.1. SNMP

### Model d'arquitectura SNMP

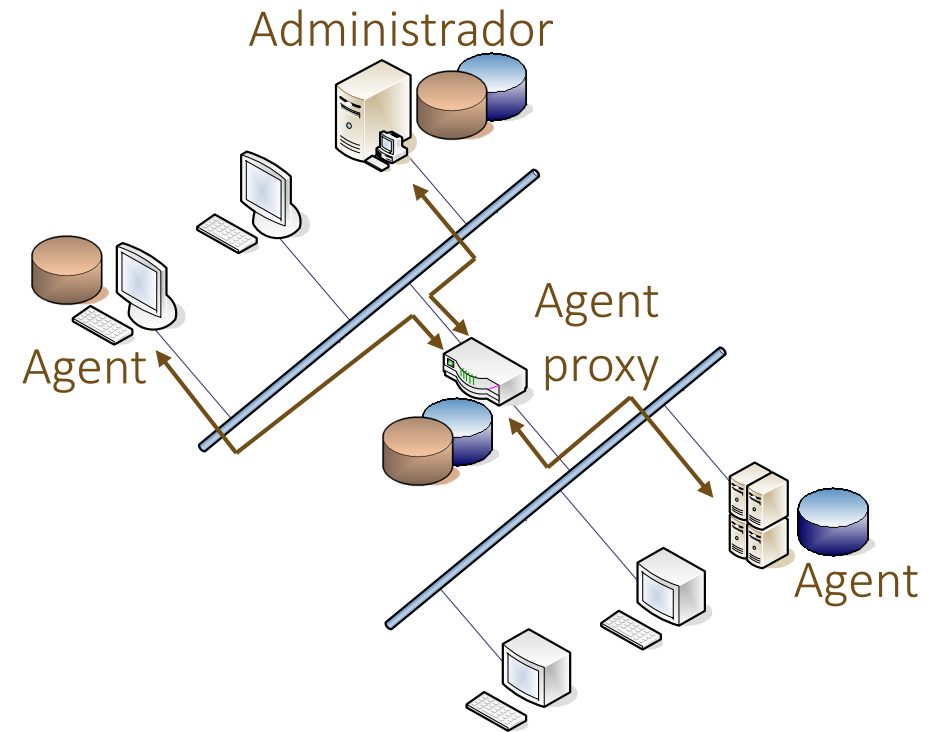
- Base de dades amb informació sobre:
  - ✓ Configuració
  - ✓ Estat
  - ✓ Errors
  - ✓ Rendiment
- Entitats administradores i administrades:
  - ✓ Agents
  - ✓ Agents proxy (SNMP v:2.0)
  - ✓ Administradors
  - ✓ Bases d'informació d'administració (MIB)
- Protocol de comunicació
  - ✓ Accés a la capa de transport
  - ✓ Missatges



## 6.8.1. SNMP

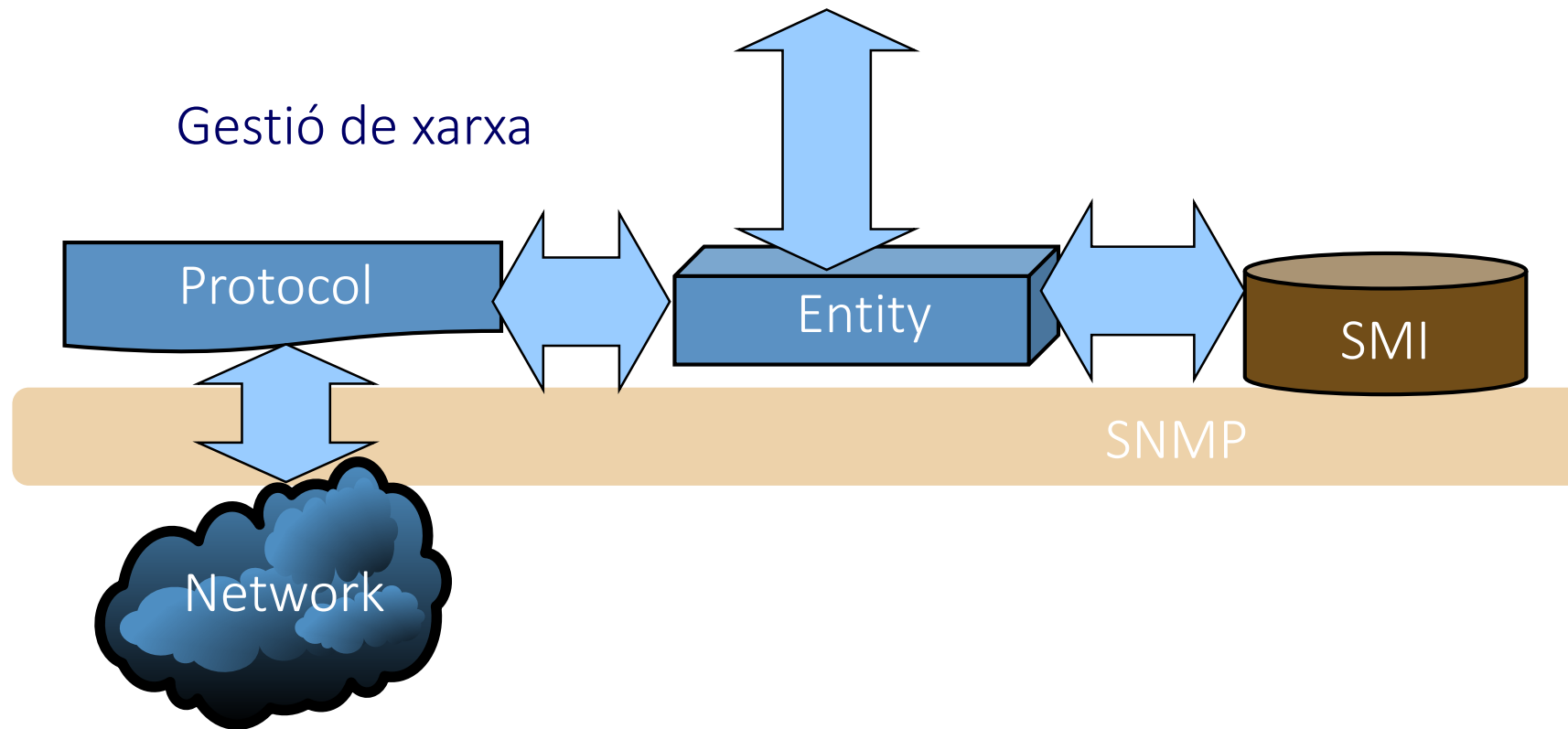
### Arquitectura SNMP

- L'administrador executa aplicacions, genera comandes SNMP, interroga als agents i presenta les dades a l'usuari mitjançant una interfície gràfica
- Les dades s'emmagatzemen en les MIB.
- El protocol SNMP viatja sobre UDP (ports 161 i 162).



## 6.8.2. SMI

SMI (Structure of Management Information) proporciona el marc general de treball de les MIB



## 6.8.2. SMI

Les característiques de la SMI venen recollides en múltiples RFCs

*RFC 1155:* Structure and Identification of Management Information for TCP/IP-base Interfaces

*RFC 1213:* Management Information Base for Network Management of TCP/IP-base Interfaces: MIB-II

*RFC 1643:* Definition of Managed Objects for the Ethernet-like Interface Types

*RFC 2021:* Remote Network Monitoring Management Information Base 2

.....

## 6.8.2. SMI

### Punts clau de l'estructura de la informació de gestió

Estableix la metodologia per crear l'estructura de les MIB, l'*arbre SMI*

Defineix com es creen els objectes de les MIB, tant la sintaxi com el valor, *ASN.1*

Defineix la metodologia per codificar els valors dels objectes de les MIB, *BER*



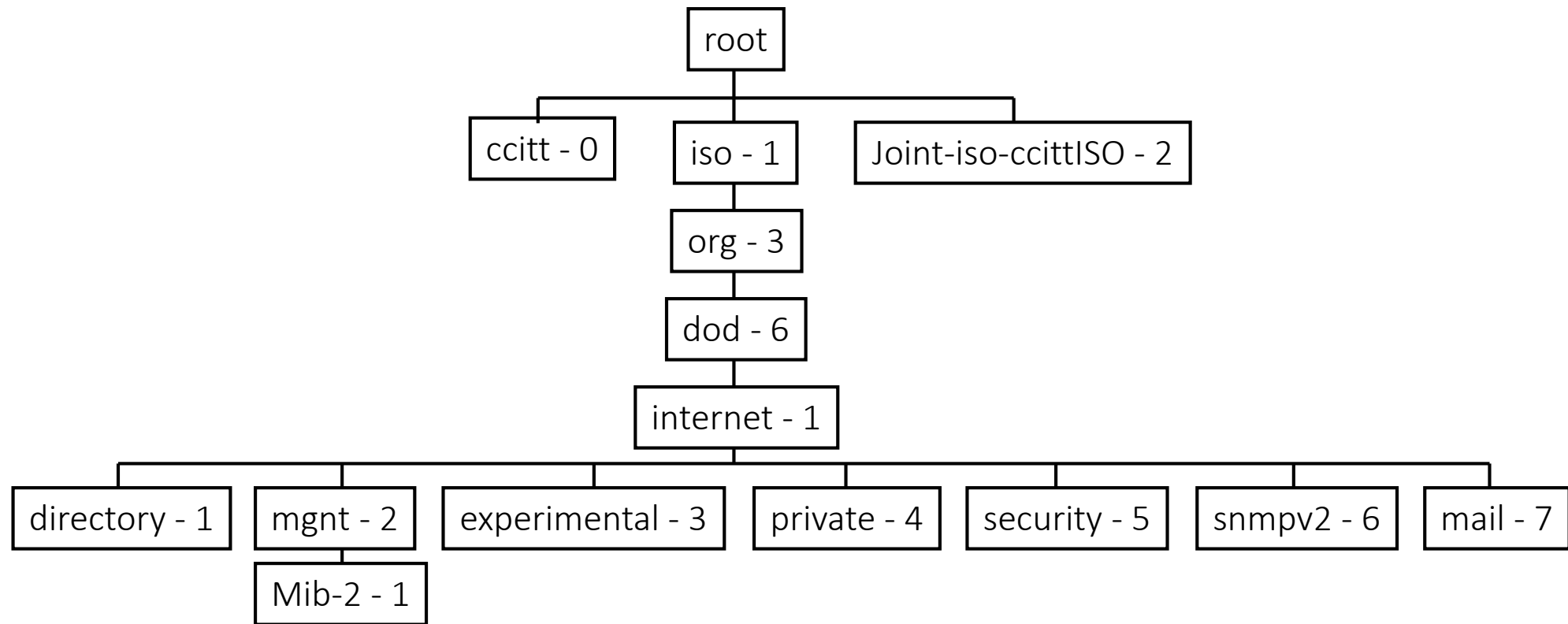
## 6.8.2. SMI

### Les MIBs contenen els objectes administrables

- Contenen la descripció lògica de les dades d'administració de la xarxa (RFC 1155).
- Els conjunts de variables relacionades s'agrupen en **Mòduls MIB** (recollits en RFCs).
- La descripció d'una variable específica:
  - Una definició de què és la variable.
  - Una descripció de com es mesura el seu valor.
  - Un nom per a quan es llegeixi o actualitzi el valor de la variable a la base de dades.

## 6.8.3. Arbre SMI

Les variables de la MIB s'ordenen seguint una estructura en arbre





## 6.8.3. Arbre SMI

### Noms identificadors dels objectes

- Al final s'inclou el número del sistema administrat
- Tota variable es pot identificar de dues maneres:
  - *Identificador d'objecte*: es comença des de l'arrel unint el números dels nodes
  - *Nom de l'objecte*: es comença des de l'arrel unint els noms dels nodes



Identificador d'objecte: *1.3.6.1.6.3*

Nom: *iso.org.dod.internet.snmpv2.xxxx*

## 6.8.3. Arbre SMI

### Ordenació en les taules MIB

- Es comença per l'esquerra
- Es compara fins que es troba el primer valor diferent
- L'element amb el número major en aquesta posició és l'element més gran
- Altrament, l'identificador major és el de valor més gran

Exemple:

.....

*1.3.6.1.6.2*

*1.3.6.1.6.3*

*1.3.6.1.6.4*

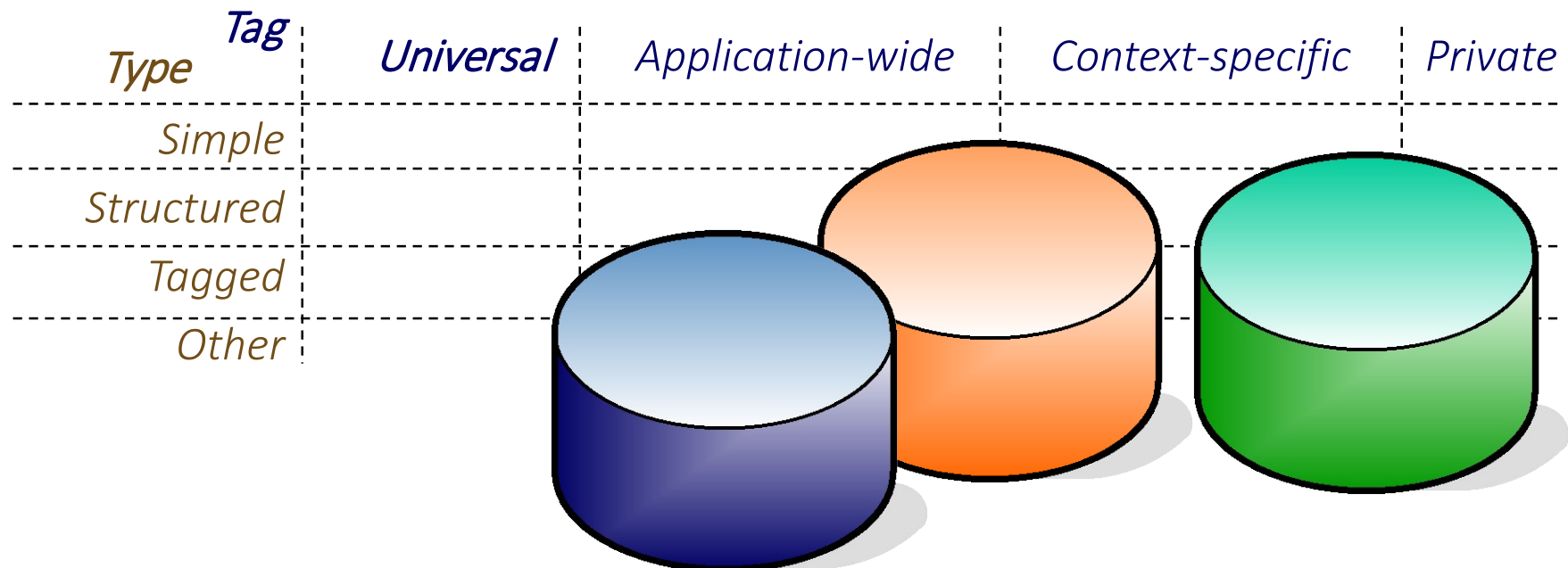
*1.3.6.1.6.4.1*

.....

## 6.8.4. Codificació ASN.1

Les dades es classifiquen amb tipus i etiquetes

S'agrupen per la seva naturalesa (*type, ex: integer*) i per la seva utilització (*tag, ex: router UPC*)



## 6.8.4. Codificació ASN.1

La classe de dades UNIVERSAL forma la base de la resta de variables

### *Tipus bàsics*

- 1 - BOOLEAN
- 2 - INTEGER
- 3 - BIT STRiNG
- 4 - OCTET STRinG
- 9 - REAL
- 10 - ENUMERATED

### *Tipus d'objectes*

- 6 - OBJECT IDENTIFIER
- 7 - Object descriptor

### *Reservats*

- 19-5 , 28-...

### *Tipus de cadenes de caràcters*

- 18 - NumericString
- 19 - PrintableString
- 27 - GeneralString, ...

### *Tipus estructurats*

- 16 – SEQUENCE, SEQUENCE OF
- 17 – SET, SET OF

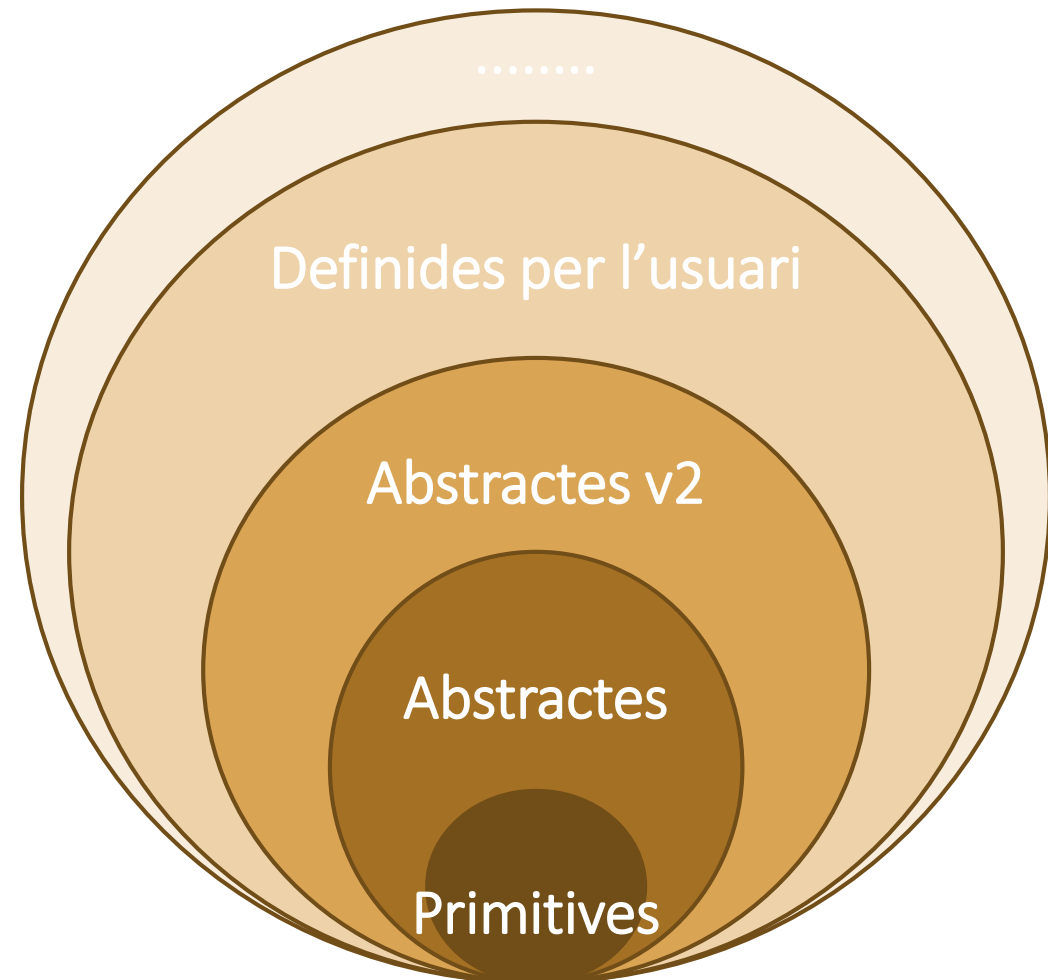
### *Tipus varitas*

- 5 - NULL
- 8 - EXTERNAL
- 23 – UTCTime...

## 6.8.4. Codificació ASN.1

### Encapsulament de dades

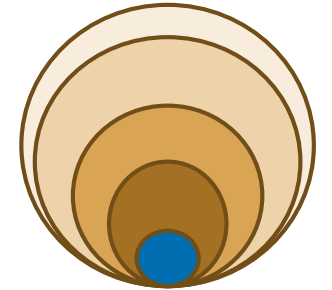
Els tipus de dades es van formant a partir de les combinacions de les dades ja definides en les MIBs



## 6.8.4. Codificació ASN.1

### Primitive data

Són els tipus bàsics de dades

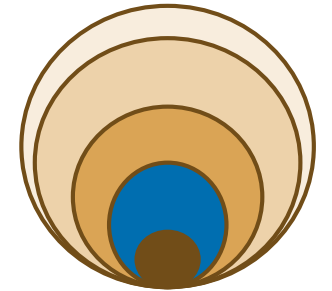


- ***integer***. 32 bits en complement a dos. S'utilitza per a enumeracions
- ***Octet string***: cadena de 0 o més octets. S'utilitza per a representar texts.
- ***Object identifier***: seqüència d'enters. S'utilitza per a identificar els objectes en les MIBs.
- ***null***: en blanc
- ***Sequence, sequence-of***: s'utilitza per a construir taules

## 6.8.4. Codificació ASN.1

### Abstract data (SNMP v1)

Afegeixen el primer grau d'abstracció

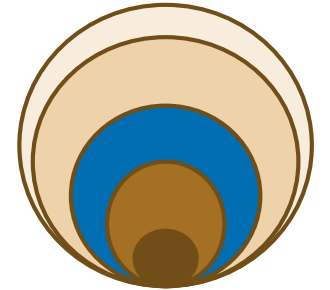


- *NetworkAdress*: adreça de xarxa, depén del protocol
- *IpAddressCounter*: 4-byte OCTET STRING
- *Counter*: 32 bits sense signe amb overflow
- *Gauge*: 32 bits sense signe amb saturació
- *TimeTicks*: comptador de temps de 32bits (centèsimes de segon). Com a màxim 497 dies
- *Opaque*: qualsevol dada en format oCTET STRING

## 6.8.4. Codificació ASN.1

### Abstract data (SNMP v2)

Són un refinament de les anteriors

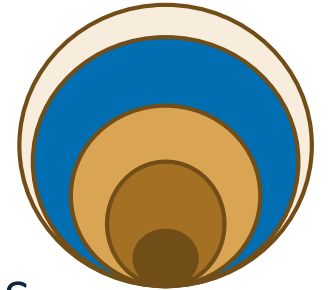


- *Integer32*: igual que INTEGER
- *Counter32*: igual que Counter
- *Counter64*: igual que Counter però amb 64 bits
- *Gauge32*: igual que Gauge
- *Unsigned32*: a la pràctica igual que Gauge



## 6.8.4. Codificació ASN.1

### Textual conventions



Les textual conventions amplien les descripcions dels tipus de objectes administrats continguts en les MIBs

- *Definir nous tipus*
- *Representar els tipus existents*
- Representar els valors dels tipus
- Codificar els valors dels tipus existents

## 6.8.5. Regles BER

Les Basic Encoding Rules ,BER, especifiquen com codificar qualsevol valor definit amb ASN.1

Un valor es pot codificar de diverses maneres

Utilitzen el tipus OCTET STRING

Segueixen l'estructura: tipus – longitud – valor

Existeixen tres mètodes

- Primitiva, definite-length encoding
- Construïdes, definite-length encoding
- Construïdes, indefinite-length encoding

## 6.8.5. Regles BER

### Format de la codificació BER

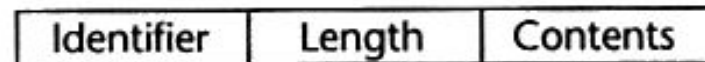
Els objectes es codifiquen amb quatre camps

Identificador: referència quin objecte administrat es refereix

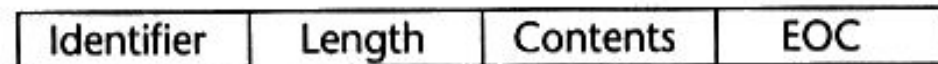
Longitud: indica el número de bytes que utilitza

Contingut: valor de l'objecte

Final de contingut: marca el final del valor en els casos de longitud variable



definite-length encoding



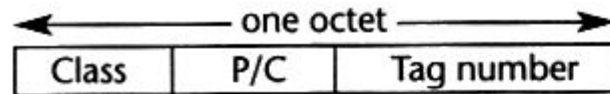
indefinite-length encoding

EOC = 0000<sub>16</sub>

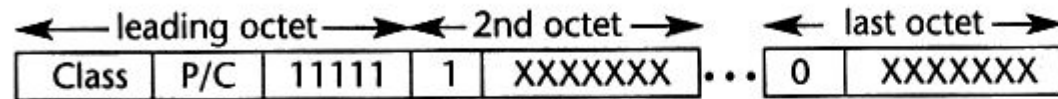
## 6.8.5. Regles BER

### Format del camp identificador de les BER

Tags <31



Tags ≥31



Class:

- 00 = Universal
- 01 = Application
- 10 = Context specific
- 11 = Private

P/C = primitive  
encoding

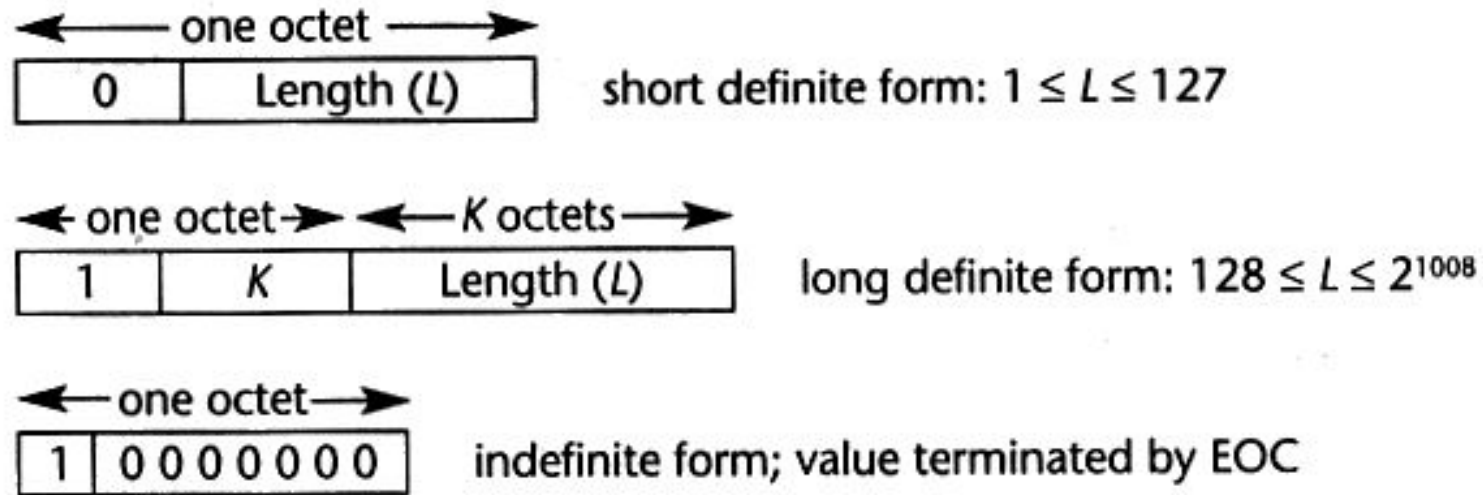
P/C = constructed  
encoding

Tag number:

- 1 = Boolean type
- 2 = Integer type
- 3 = Bitstring type
- 4 = Octetstring type
- 5 = Null type
- 6 = Object identifier type
- 9 = Real type
- 10 = Enumerated type
- 16 = Sequence and sequence-of types
- 17 = Set and set-of types
- 18–22, 25–27 = Character string types
- 23–24 = Time types
- >30: XX...X = Tag number

## 6.8.5. Regles BER

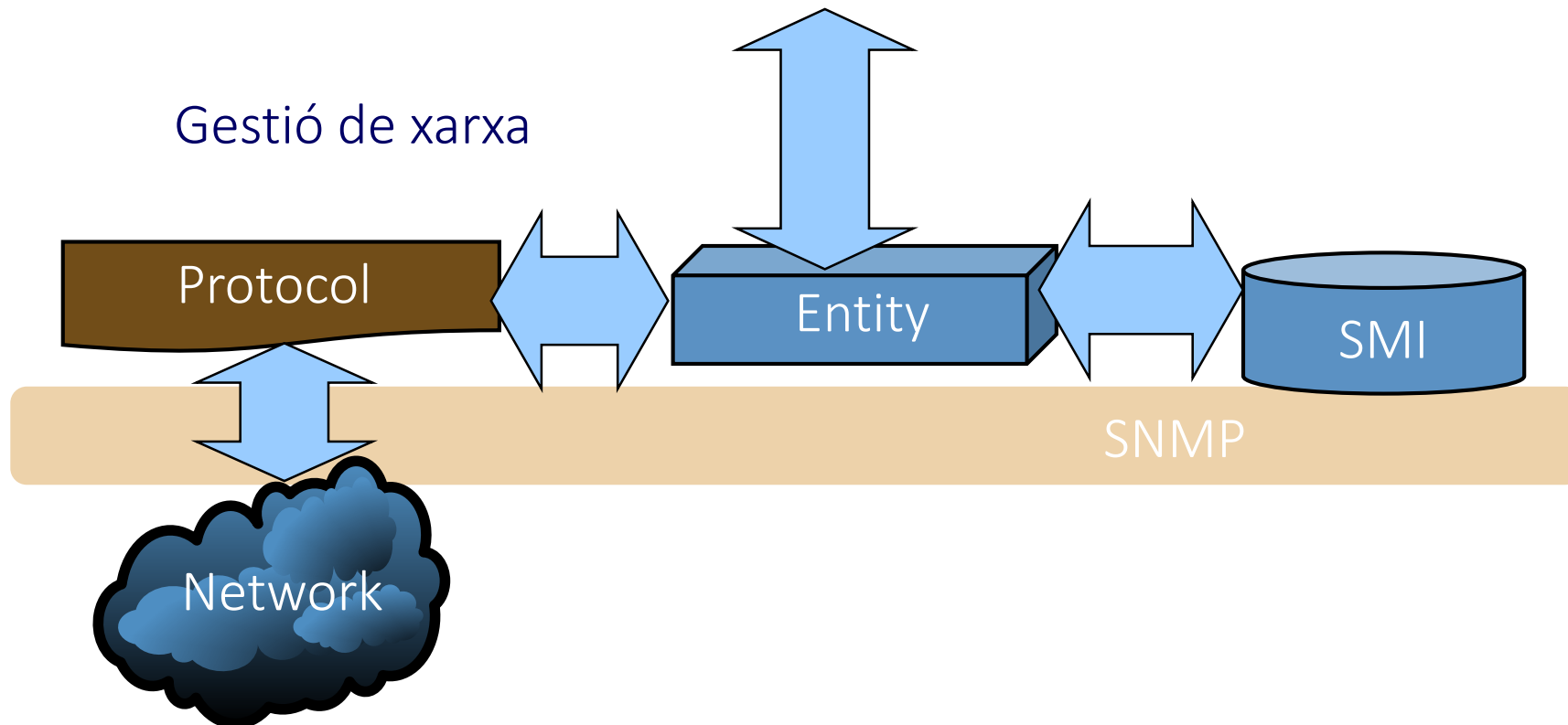
### Format del camp longitud de les BER



El camp EOC és format per dos bytes a zero

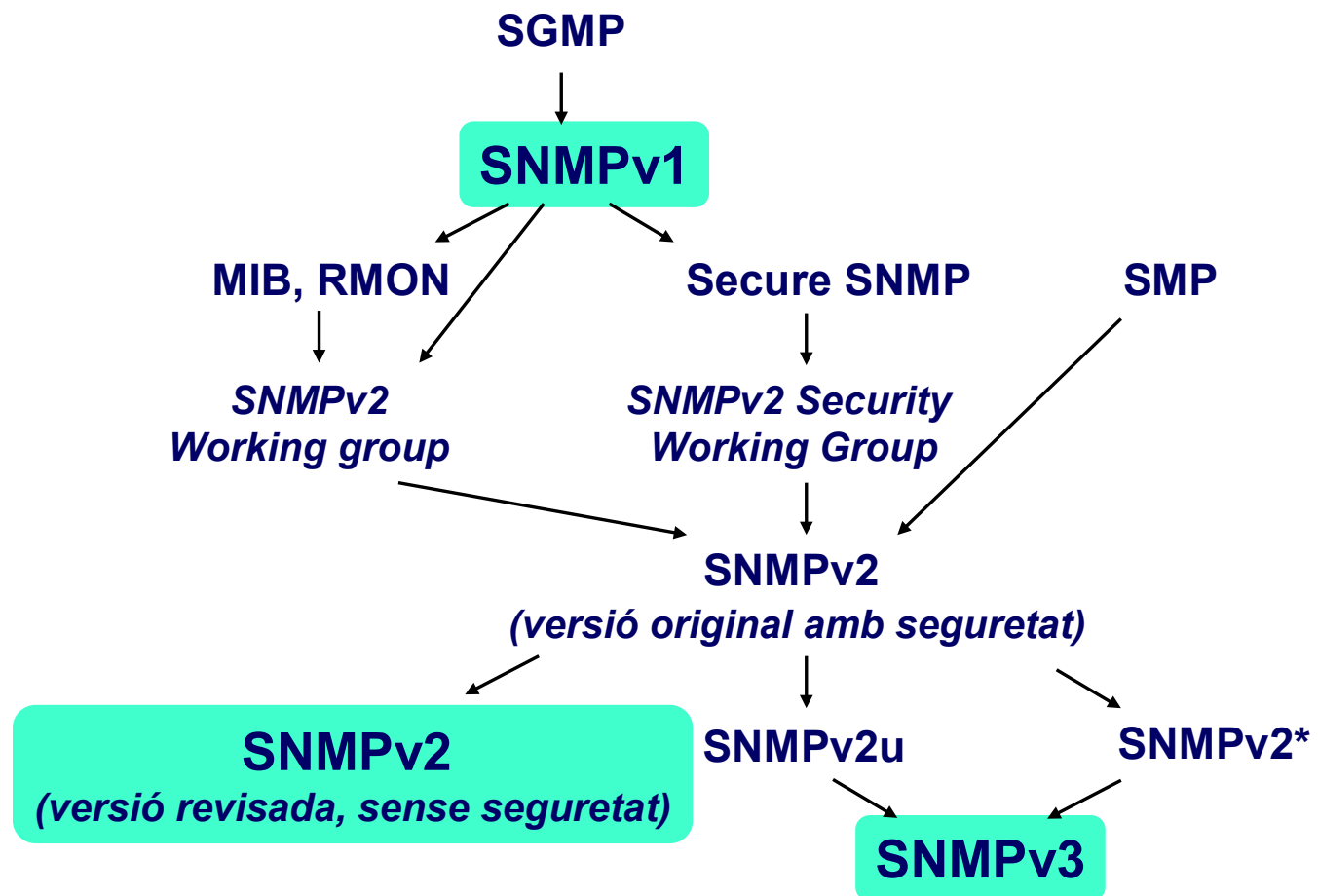
## 6.8.6. Protocol SNMP

Mitjançant el Protocol de gestió de xarxa les entitats es comuniquen per la xarxa



# 6.8.6. Protocol SNMP

## Evolució del protocol SNMP



## 6.8.7. Protocol SNMP v1

La versió 1 de SNMP ve recollida en 5 RFCs

*RFC 1155:* Structure and Identification of Management Information for TCP/IP-base Interfaces

*RFC 1157:* A Simple Network Management Protocol (SNMP)

*RFC 1212:* Concise MIB definitions

*RFC 1213:* Management Information Base for Network Management of TCP/IP-base Interfaces: MIB-II

*RFC 1643:* Definition of Managed Objects for the Ethernet-like Interface Types



## 6.8.7. Protocol SNMP v1

SNMP necessita serveis a nivell de transport per enviar els missatges

Protocols de transport suportats

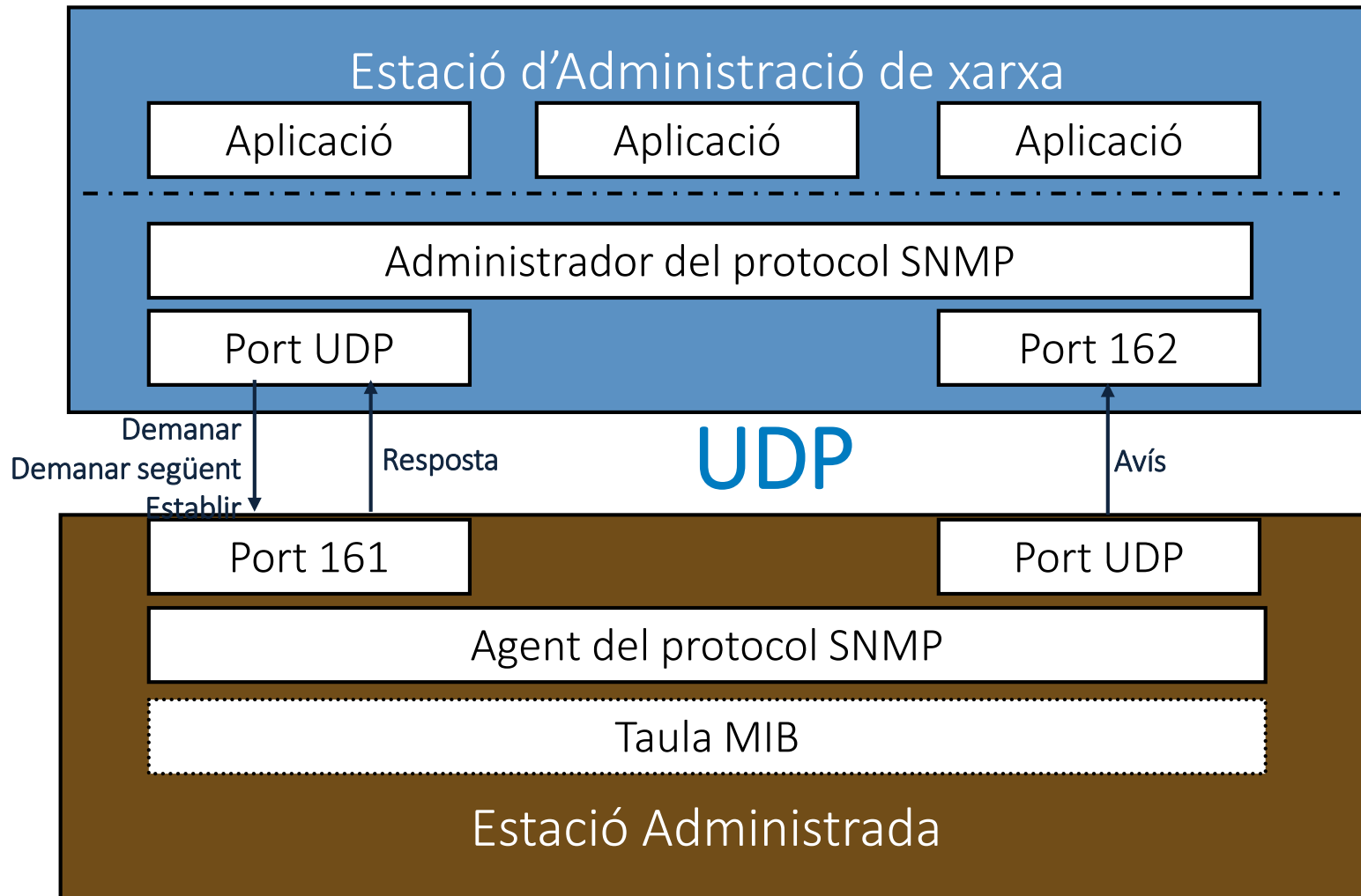
- **UDP**: User Datagram protocol de l'arquitectura TCP/IP
- **CLTS**: ConnectionLess Transport Services de l'arquitectura OSI

Està pensat per córrer sobre protocols de transport orientats a connexió

- L'administrador pot mantenir la connexió oberta per anticipar-se a l'agent.
- L'agent pot tancar una connexió si li fan falta recursos

# 6.8.7. Protocol SNMP v1

## Accés a la capa de transport



## 6.8.7. Protocol SNMP v1

### UDP no garantitza el lliurament dels missatges

Si es perd un missatge...

GetRequest o GetNextRequest ⇒ es **repeteixen** els missatges

SetRequest ⇒ primer es **comprova** si l'operació s'ha realitzat amb un GetRequest i si cal es **repeteix** el missatge SetRequest

Trap ⇒ **no es pot saber** ⇒ periòdicament l'administrador cal que contacti amb els agents (**polling**)

## 6.8.7. Protocol SNMP v1

Al tractar-se d'un sistema distribuït cal agrupar els equips que pertanyen a l'entorn de gestió: les comunitats SNMP

El concepte de comunitat permet oferir serveis de:

*Autenticació*: l'agent pot limitar l'accés a les MIB només a comunitats autoritzades

*Polítiques d'accés*: l'agent pot assignar diferents privilegis a cada administrador

*Proxy*: un agent pot actuar com a proxy dels agents d'una altre comunitat

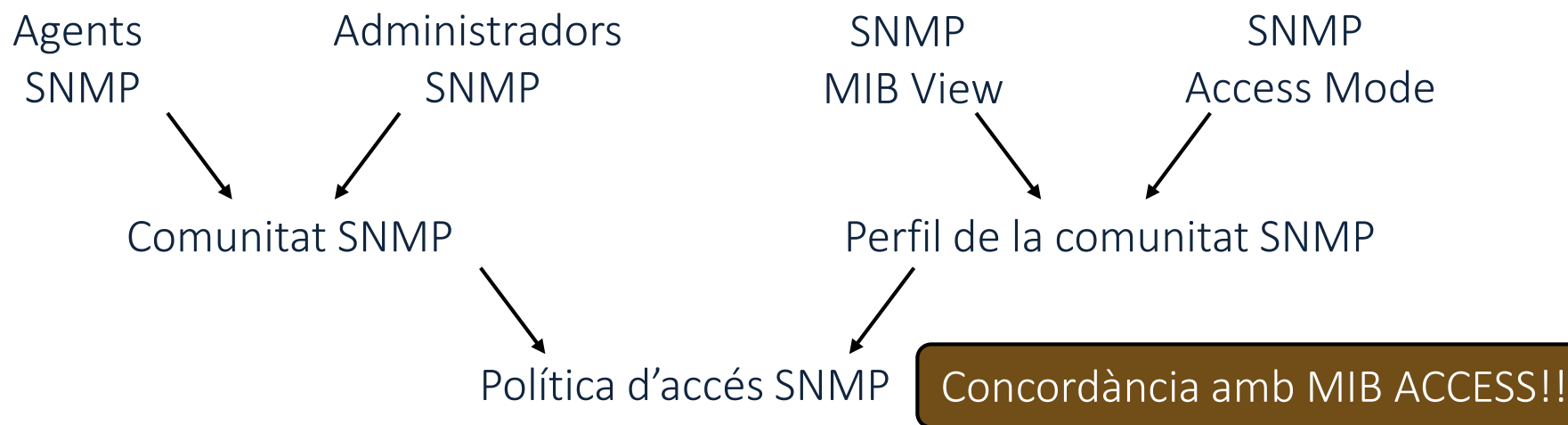
## 6.8.7. Protocol SNMP v1

Mitjançant polítiques d'accés es gestionen els privilegis sobre els objectes de les comunitats

Es porta a la pràctica mitjançant dos conceptes:

**Views:** seleccions d'objectes de les MIBs

Access Mode: privilegis (READ-ONLY, READ-WRITE)



## 6.8.7. Protocol SNMP v1

### Seqüència de transmissió dels missatges SNMP



1

Es construeix la *PDU* mitjançant l'*ASN.1*

2

Es crea el *cos del missatge*, junt amb dades de *seguretat*, i es processa amb els mecanismes d'autenticació i xifrat

3

S'afegeixen els camps de la *capçalera*

4

Es codifiquem amb les *BER* i es passen a la capa de transport *UDP*



## 6.8.7. Protocol SNMP v1

### Seqüència de recepció dels missatges SNMP



1

La capa de transport passa el missatge i es descodifica amb les BER

2

S'analitza la capçalera

3

Amb les dades de seguretat s'autentica i llavors es descripta el cos del missatge

4

Mitjançant l'ASN.1 s'extrauen els objectes



## 6.8.7. Protocol SNMP v1

### Tipus de missatges SNMP v1

El SNMP especifica 5 tipus de missatges:

**Get Request:** el gestor demana el valor de variables de la MIB de l'agent

**Get next Request:** no fa falta el nom de la variable

**Set Request:** el gestor demana canvis en valors de variables de la MIB de l'agent

**Get Response:** respostes de l'agent al gestor a les comandes anteriors. Conté el missatge original més la resposta



## 6.8.7. Protocol SNMP v1

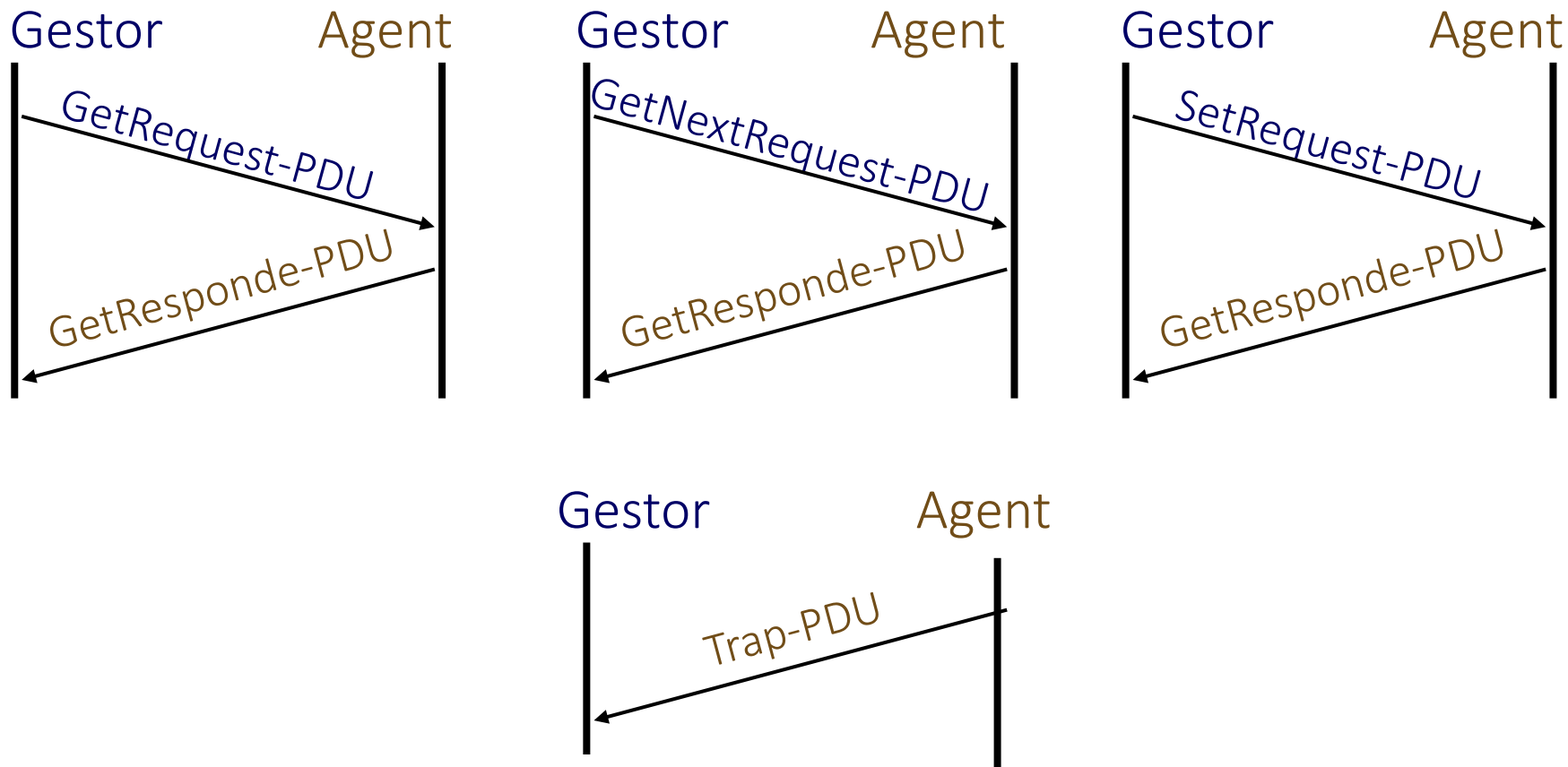
### Tipus de missatges SNMP v1

**trap:** respostes no sol·licitades. Alarmes:

- **coldStart:** l'agent s'ha inicialitzat per si sol
- **warmStart:** l'agent s'ha reiniciat per si sol
- **linkDown:** la interfície s'ha desactivat
- **linkUp:** la interfície s'ha activat
- **authenticationFailure:** missatge rebut d'un agent que no pertany a la comunitat
- **egpNeighborLoss:** un EGP s'ha desactivat (s'envia la seva adreça IP)
- **enterpriseSpecific:** codi d'alarma específic

# 6.8.7. Protocol SNMP v1

## Diagrama de flux dels missatges



## 6.8.7. Protocol SNMP v1

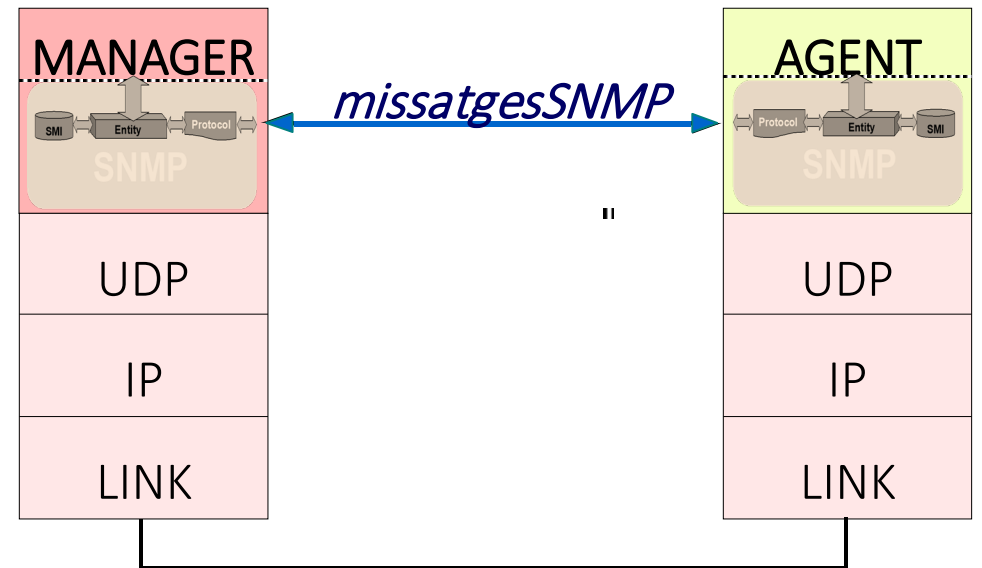
### Camps dels missatges SMNP v1

*Versió del protocol:* 0 per la v1, 1 per la v2 i 3 per la v3

*Nom de la comunitat:* s'utilitza com a contrasenya

*Identificador de missatge:* indica el tipus de missatge

*Identificador de sol·licitud:* s'utilitza per a relacionar sol·licituds i respostes



## 6.8.7. Protocol SNMP v1

### Camps dels missatges SMNP v1

*Estat de l'error.* 0 per a les peticions. A les respostes un valor diferent de 0 indica un error

- noError (0): no hi ha error
- tooBig (1): el resultat no hi cap en el PDU del response
- noSuchName (2): l'objecte no existeix
- badValue (3): valor de l'objecte incorrecte (resposta a un set)
- readOnly (4): objecte només de lectura (resposta a un set)
- genError (5): Causa d'error desconeguda

*Índex d'error.* 0 a les peticions. A les respostes indica la variable que ha causat els problemes

## 6.8.7. Protocol SNMP v1

### Camps dels missatges SMNP v1

*Identificador d'objecte:* tipus de l'objecte que ha generat l'error (sysObjectID)

*Adreça de l'agent:* adreça de l'objecte que ha generat l'alarma

*Identificador d'alarma:* indicador del tipus d'alarma

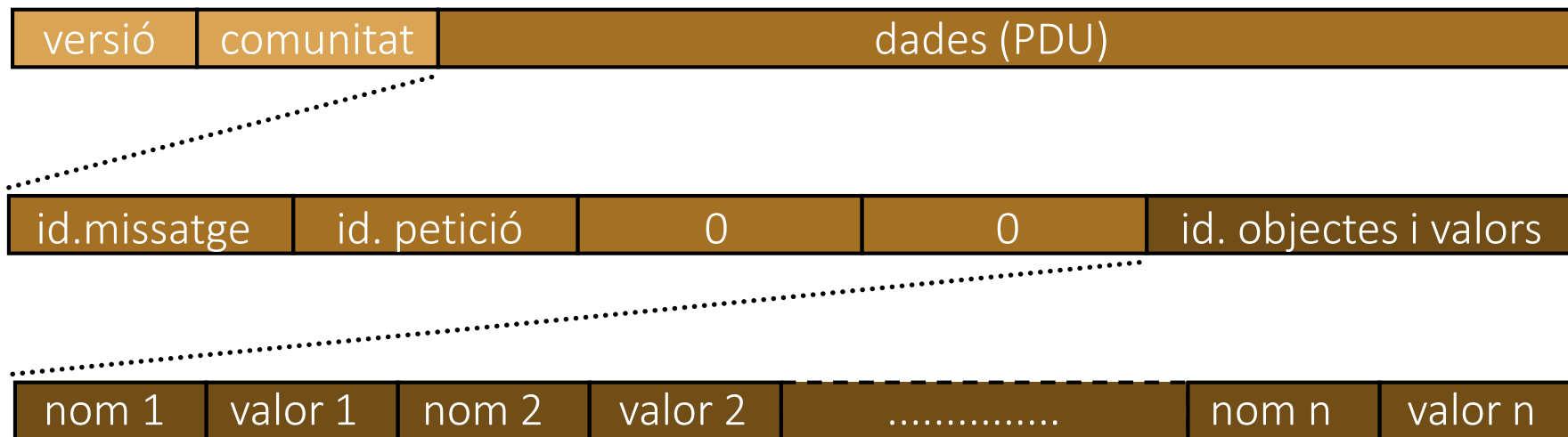
*Identificador del codi d'alarma:* codi de l'alarma

*Marca de temps:* temps qe ha passat entre la darrera reinicialització del dispositiu i l'alarma actual

*Id d'objectes i valors:* variables i valors demanats/resposts. N'hi ha tants com variables es demanin/responguin

## 6.8.7. Protocol SNMP v1

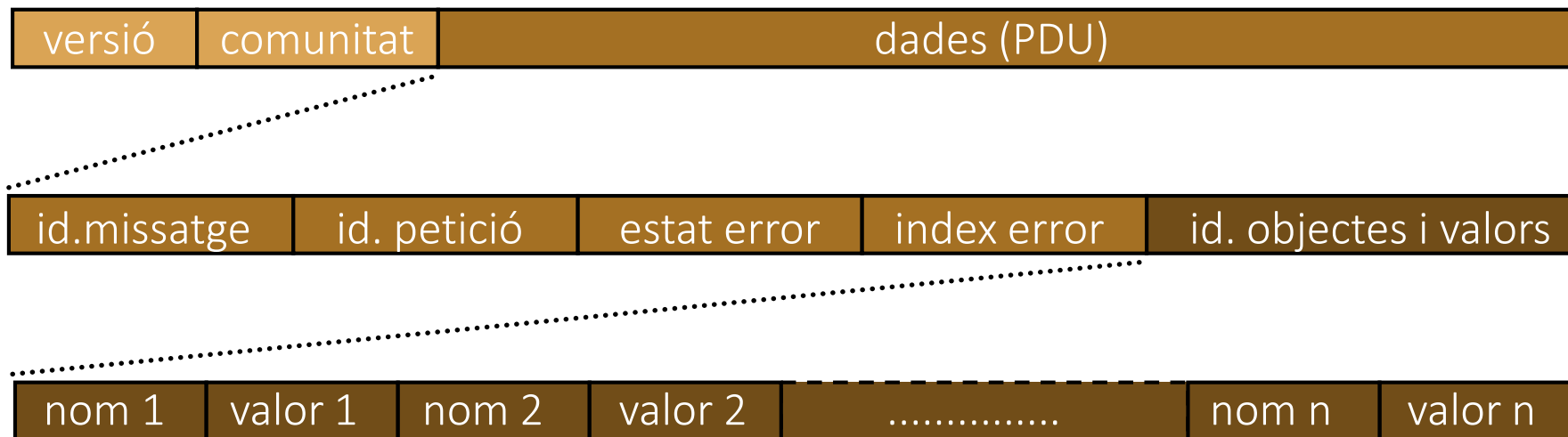
### Missatges Get Request, Get Next Request i Set Request



*PDU= Protocol Data Unit*

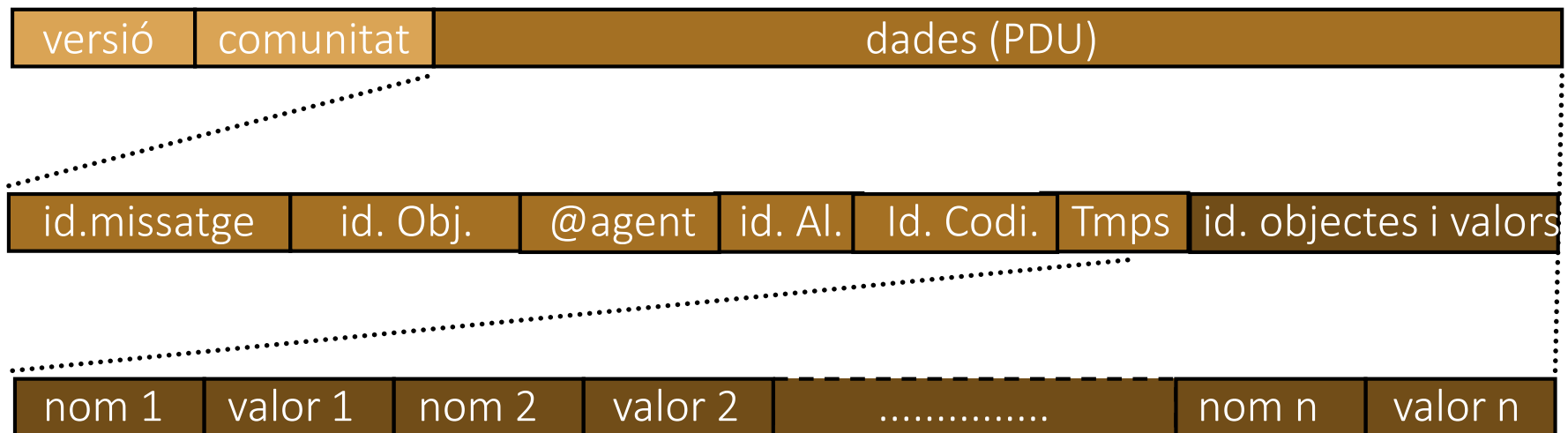
## 6.8.7. Protocol SNMP v1

### Missatges Get Response



## 6.8.7. Protocol SNMP v1

### Missatges Trap





## 6.8.8. Protocol SNMP v2

La versió 2 de SNMP ve recollida en 8 RFCs

*RFC 1901:* Introduction to Community-based SNMPv2

*RFC 1902:* Structure of Management Information for SNMPv2

*RFC 1903:* Textual Conventions for SNMPv2

*RFC 1904:* Conformance Statements for SNMPv2

*RFC 1905:* Protocol Operations for SNMPv2

*RFC 1906:* Transport Mappings for SNMPv2

*RFC 1907:* Management Information Base for SNMPv2

*RFC 1908:* Coexistence between Version 1 and Version 2 of Internet-Standard Network Management Framework

## 6.8.8. Protocol SNMP v2

La versió 2 incorpora millores respecte SNMP v1 en 4 grans blocs

***Àmbit d'aplicació:*** s'agafa el concepte de SMP framework del SMP (Simple Management Protocol)

***Mida, velocitat i eficiència:*** s'incorporen nous missatges

***Seguretat i privacitat:*** s'incorporen funcions de segretat del Secure SNMP a la versió original, que es treuen a la versió revisada

***Desenvolupament i compatibilitat:*** basant-se en l'SMP s'augmenta la compatibilitat amb noves arquitectures de comunicacions OSI

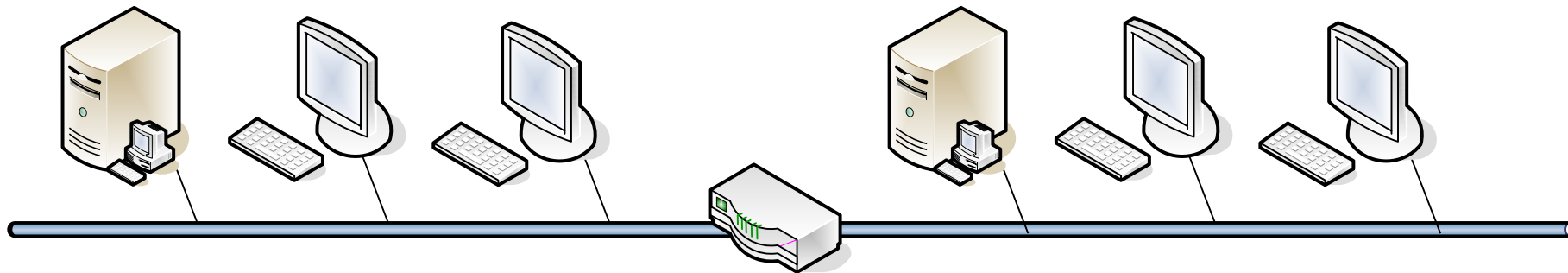
## 6.8.8. Protocol SNMP v2

En l'àmbit d'aplicació es proposen 3 tipus d'accés a les MIBs

***Manager-Agent request-response:*** Igual que en la v1, l'administrador pot obtenir i modificar les MIBs de l'agent

***Manager-Manager request-response:*** nou àmbit en que dos administradors poden compartir MIBs d'un agent

***Agent-Manager unconfirmed:*** Igual que en la v1, l'agent pot enviar alertes a l'administrador



## 6.8.8. Protocol SNMP v2

En l'àmbit de desenvolupament s'augmenten les architectures de transport

SNMPv2 pot utilitzar els serveis de transport dels protocols:

*UDP*: User Datagram Protocol

*CLNS*: OSI ConnectionLess\_Mode Network Service

*CONS*: OSI Connection-Oriented Network Service

*IPX*: Novell Internetwork Packet Exchange

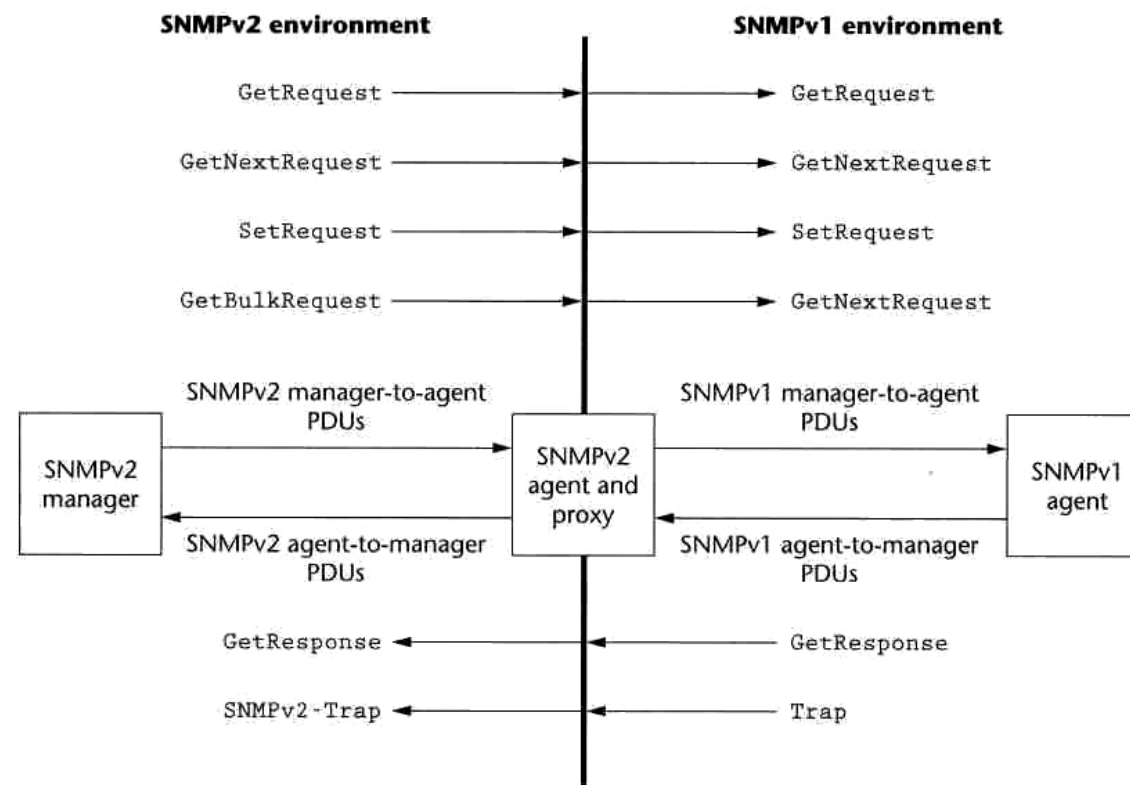
*DDP (Appletalk)*: Apple Network

## 6.8.8. Protocol SNMP v2

En l'àmbit de compatibilitat es defineix com poden coexistir la versió v1 i v2

Existeixen 2 alternatives:

- Incloure un *Proxy* que enllaci les dues *comunitats*
- *Entitas* amb *dos* *lectors* de missatges



## 6.8.8. Protocol SNMP v2

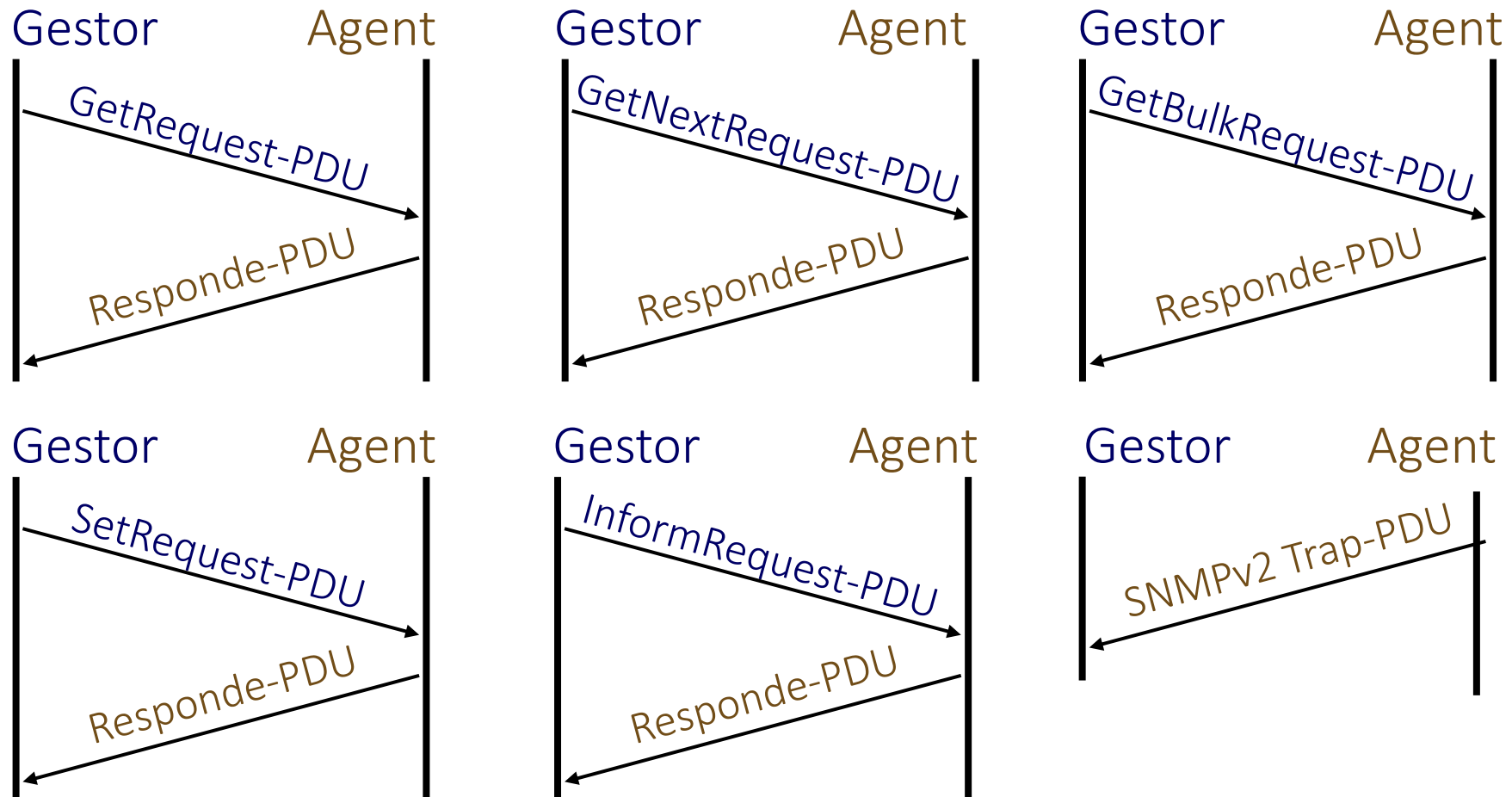
### Tipus de missatges SNMP v2

El SNMP especifica 7 tipus de missatges:

- **Get Request:** es demana el valor de variables de la MIB
- **Get Next Request:** no fa falta el nom de la variable
- **Get Next Bulk:** com *get next request*, però permet demanar més d'una variable alhora i indicar repeticions
- **Set Request:** es demanen canvis en variables de la MIB
- **Response:** respostes a les comandes anteriors
- **Inform Request:** informacions entre gestors
- **Trap:** alarmes, segueix la macro *NOTIFICATION-TYPE*
- **Report:** sense definir

## 6.8.8. Protocol SNMP v2

### Diagrama de flux dels missatges



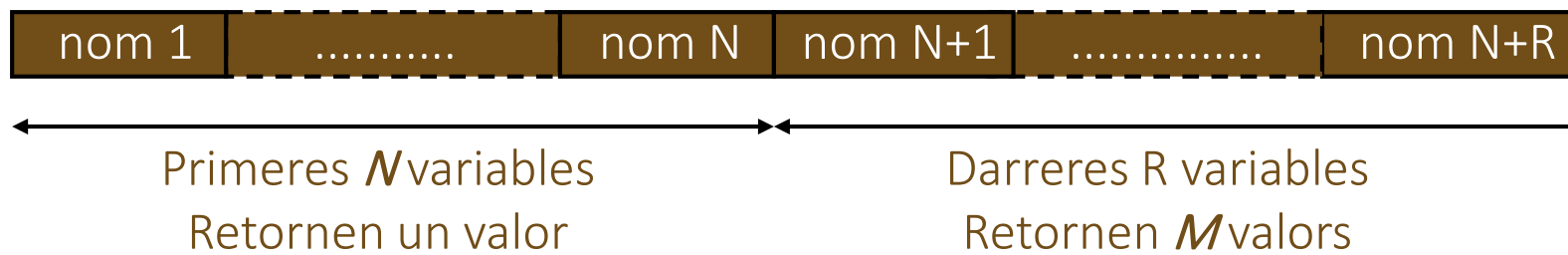
## 6.8.8. Protocol SNMP v2

### Camps dels missatges SMNP v2

Als camps descrits en la versió 1 cal afegir els dos següents:

*No repetir*: número de variables sense repetir ( $N$ )

*Max iter*: número màxim de repeticions ( $M$ )





## 6.8.8. Protocol SNMP v2

### Camps dels missatges SMNP v2

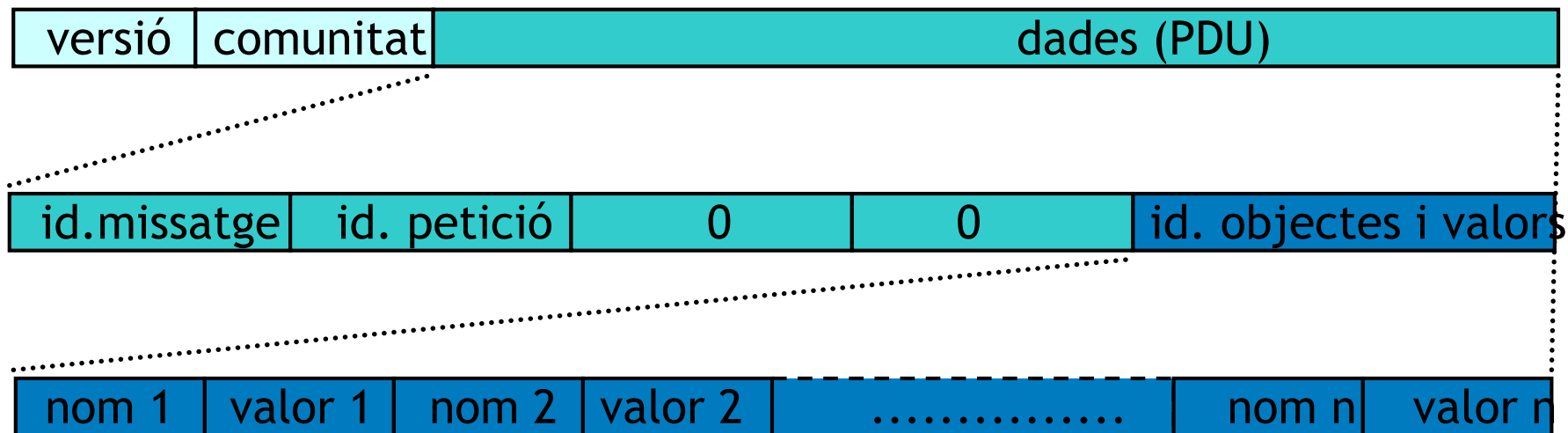
El Camp error  
pot prendre els  
següents valors:

	(1)	(2)	(3)		(1)	(2)	(3)
noError(0)	X	X	X	wrongValue(10)	x		
tooBig(1)	X	X	X	noCreation(11)	X		
noSuchName(2)				InconsistentValue(12)	X		
badValue(3)				resourceUnavailable(13)	X		
readOnly(4)				commitFailed(14)	x		
genError(5)	X	X	X	undoFailed(15)	X		
noAccess(6)		X		authorizationError(16)	X		
wrongType(7)		X		notWritable(17)	X		
wrongLength(8)		X		inconsistentName (18)	X		
wrongEncoding(9)		X					

*(1) GetRequest, GetNextRequest, GetBulkRequest (2) SetRequest (3) InformRequest*

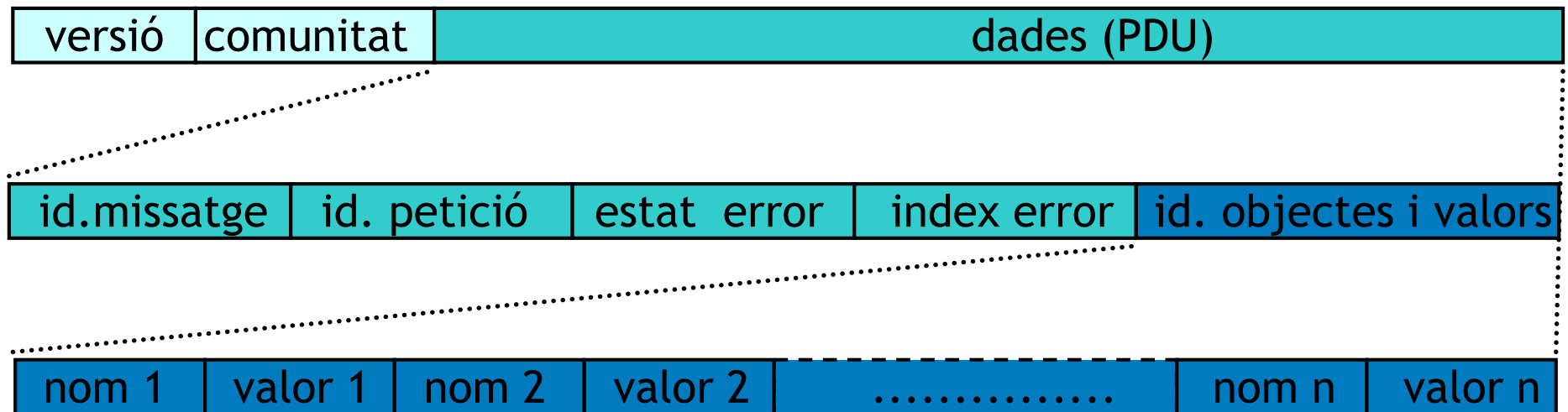
## 6.8.8. Protocol SNMP v2

Missatges Get Request, Get Next Request, Set Request, Trap i Inform Request



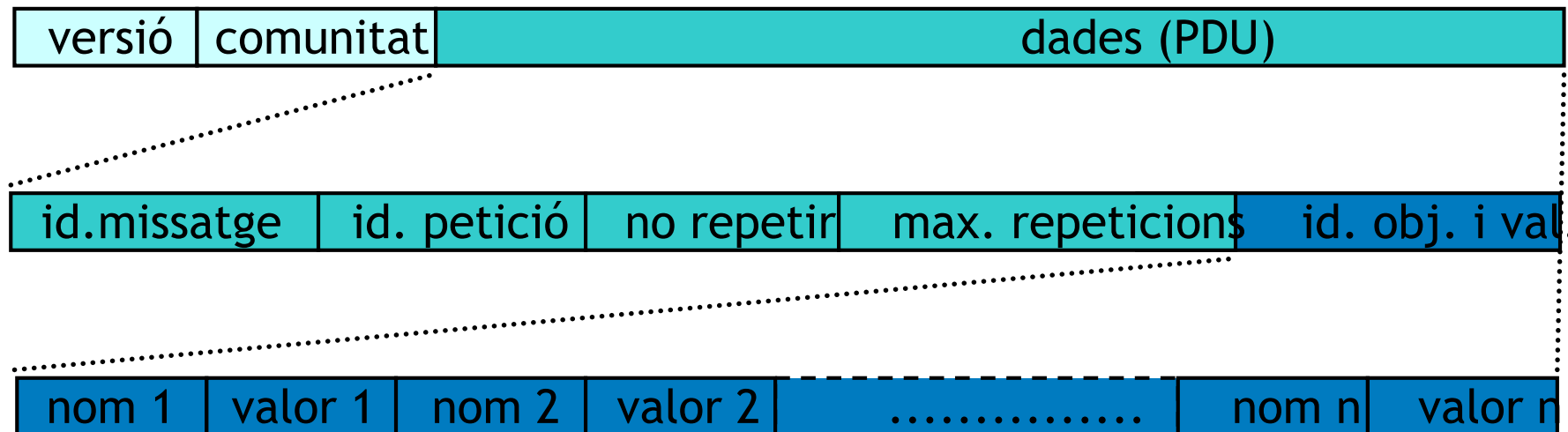
## 6.8.8. Protocol SNMP v2

### Missatges Response



## 6.8.8. Protocol SNMP v2

### Missatges Get Bulk Request



## 6.8.9. Protocol SNMP v3

La versió 3 de SNMP ve recollida en 6 RFCs

*RFC 2271:* An Architecture for Describing SNMP Management Frameworks

*RFC 2272:* Message Processing and Dispatching for SNMP

*RFC 2273:* SNMPv3 Applications

*RFC 2274:* Use-based Security Model for SNMPv3

*RFC 2275:* View Access Control Model (VACM ) for SNMP

*Internet Draft:* Introduction to Version 3 of the Internet Network Management Framework

## 6.8.9. Protocol SNMP v3

La versió 3 incorpora millores en 3 grans blocs

### *Seguretat:*

- USM: User-Based Security Model
- VACM View-Based Access Model

### *Nova estructura dels missatges*

- Hi han els mateixos missatges que en SNMPv2
- S'aplica encapsulament per aplicar habilitats de seguretat

### *S'amplien les MIBs*

- Es defineixen nous objectes i estructures

## 6.8.9. Protocol SNMP v3

L'User-based Security Model proporciona els serveis d'autenticació i xifrat

Protegeix de:

- Suplantació d'identitat
- Modificacions de la informació
- Alteració del flux de missatges (reordenació, duplicat,...)
- Revelat d'informació

No protegeix de:

- Denegació del servei
- Anàlisi del tràfic de missatges

## 6.8.9. Protocol SNMP v3

### El View-Based Access Control Model proporciona serveis de control d'accés a les mIBs (RFC2275)

Té dues funcions principals

- Determinar quin tipus d'accés té una entitat remota a un objecte local
- Determinar la política de control d'accés per l'agent i l'habilita per la configuració remota

Es basa en 5 elements

- Grups: agrupacions d'associacions <securityModel,securityName>
- Nivell de seguretat: els drets sobre un grup dependran del Nivell de seguretat del missatge
- Context: subconjunt d'objectes d'una MIB local
- Vistes de les MIBs: subarbre de la SMI local
- Política d'accés: aplica prioritàriament un joc de privilegis d'accés



## 6.8.9. Protocol SNMP v3

### Camps dels missatges SMNP v3

Als camps descrits en les versions 1 i 2 cal afegir :

**Versió:** indica la versió de SNMP per la versió 3 val 3.

**Identificador de missatge:** s'utilitza per a relacionar sol·licituds i respostes entre dues entitats  $[0, 2^{31}-1]$

**Tamany max.** indica el nº màxim de bytes que l'emissor suporta en els missatges  $[484, 2^{31}-1]$

**Atributs:** octet string on els tres bits menys significatius signifiquen:

- **Report:** a '1' força l'enviament d'un PDU Report a l'emissor
- **Privat:** a '1' el missatge s'ha encriptat
- **Autor:** a '1' el missatge porta autenticació (P=0 i A=1 no permès)

## 6.8.9. Protocol SNMP v3

Paràmetres seguretat: conté els paràmetres de seguretat USM

- *Id motor autenticació*: paràmetre *snmpEngineID* del motor d'autenticació involucrat en la comunicació.
- *Id boot autenticació*: paràmetre *snmpEngineBoots* del motor d'autenticació involucrat en la comunicació.
- *Id time autenticació*: paràmetre *snmpEngineTime* del motor d'autenticació involucrat en la comunicació.
- *Nom usuari*: indica de qui procedeix el missatge.
- *Paràmetres d'autenticació*: codi d'autenticació de missatge HMAC
- *Paràmetre de privacitat*: valor inicial per l'algoritme DES CBC

## 6.8.9. Protocol SNMP v3

### Camps dels missatges SMNP v3

*Model seguretat:* indica els mètodes de seguretats aplicats [0, 2<sup>31</sup>-1]. Estan reservats 1(SNMPv1), 2(SNMPv2), 3(USM)

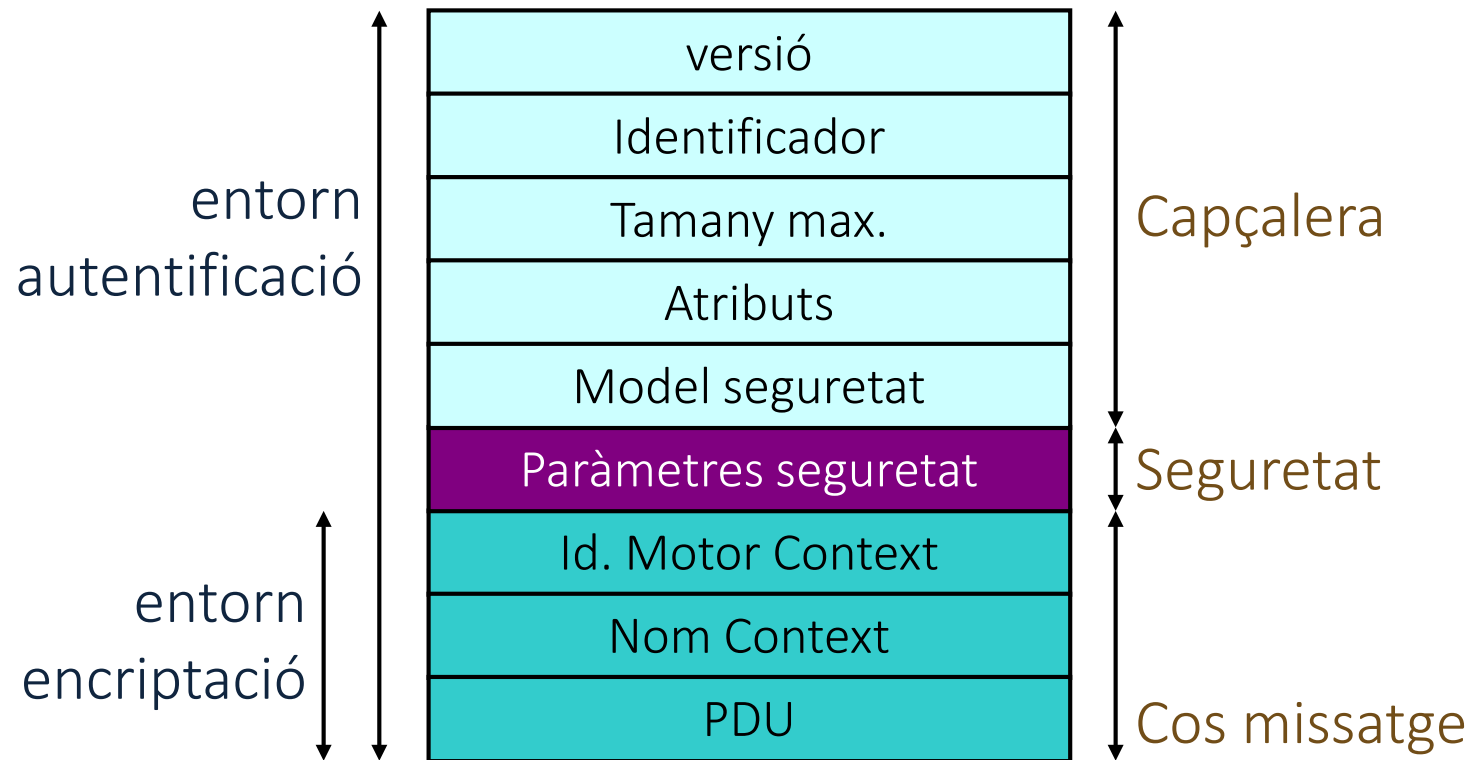
*Id. Motor Context:* identificador de l'entitat SNMP que el genera

*Nom Context:* Identificador del context

*PDU:* dades SNMP versió 2!

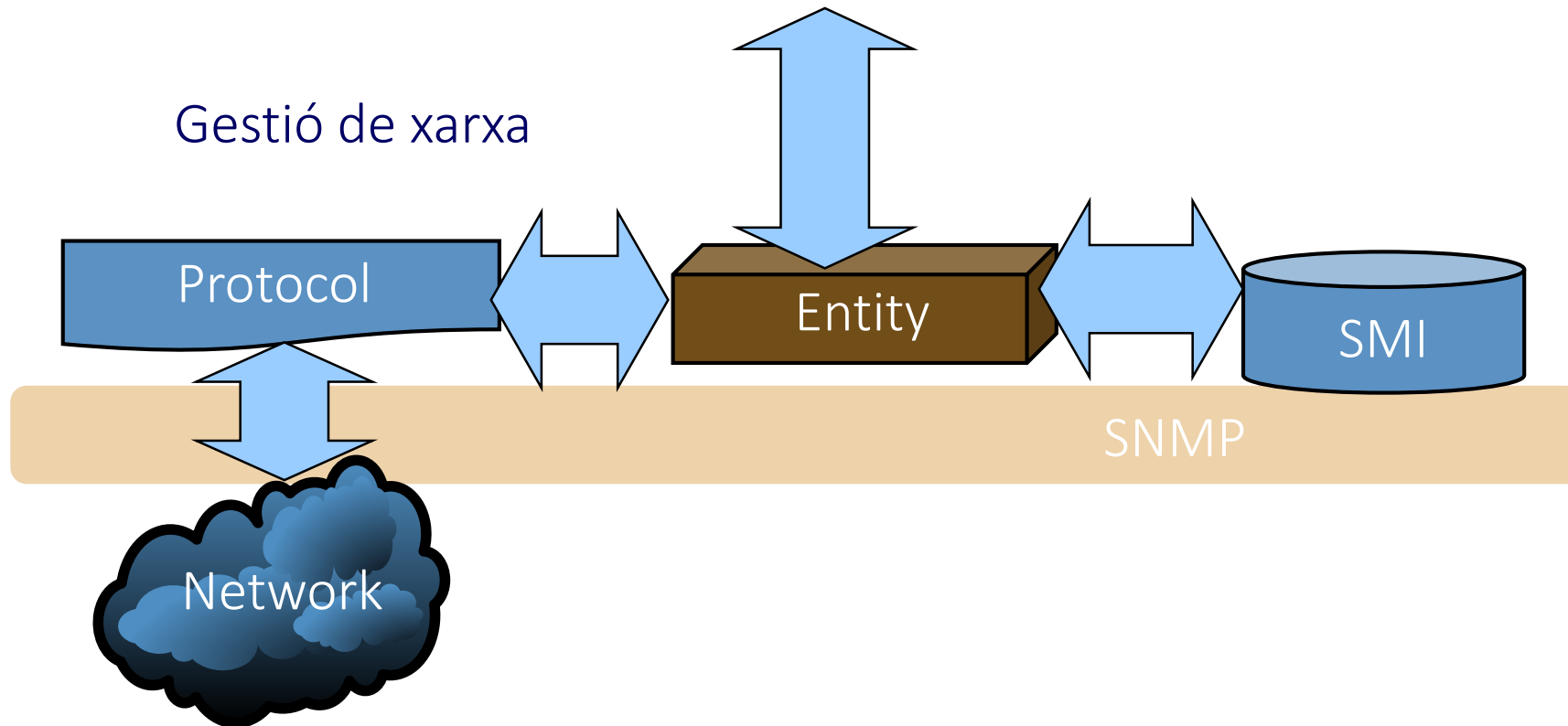
## 6.8.9. Protocol SNMP v3

### Missatges SNMPv3



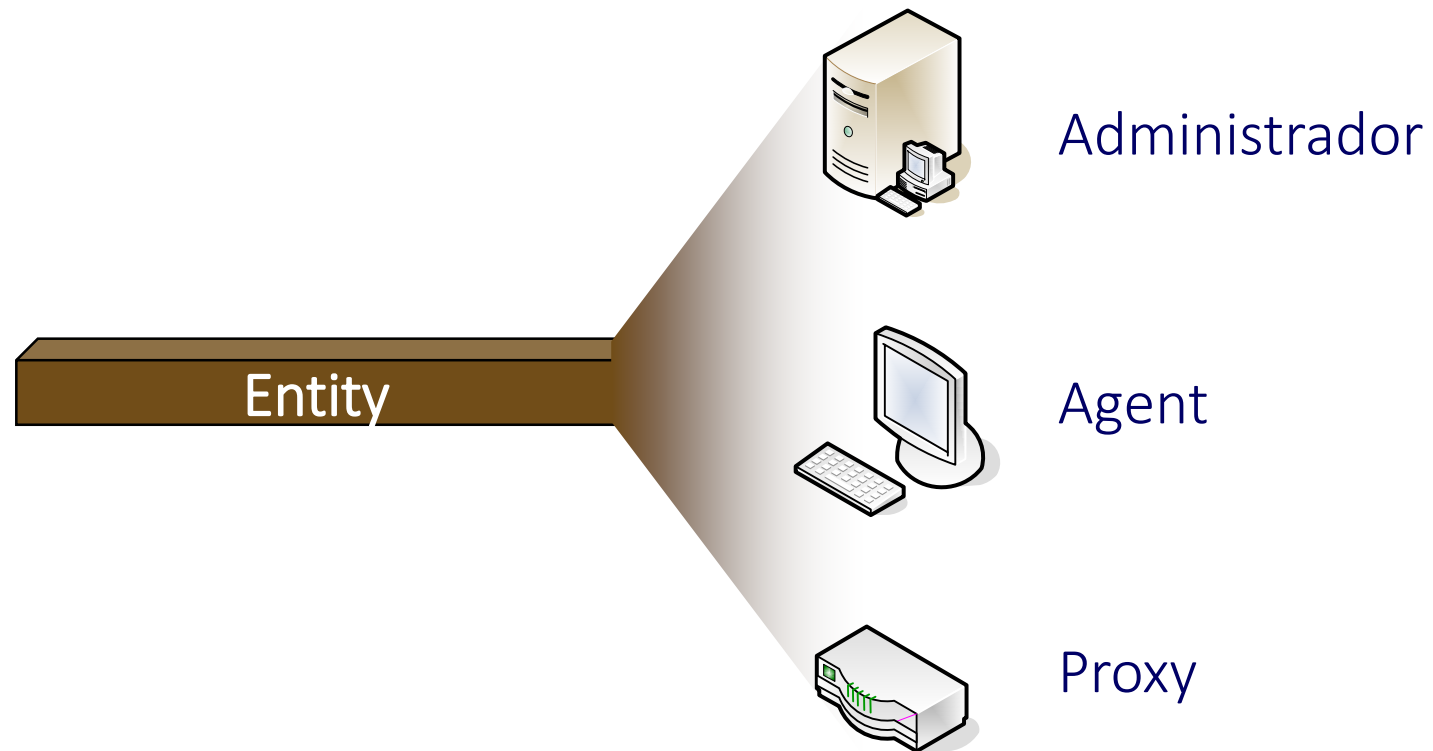
## 6.8.10. Entitats SNMP

El SNMP proporciona un entorn de gestió de xarxes



## 6.8.10. Entitats SNMP

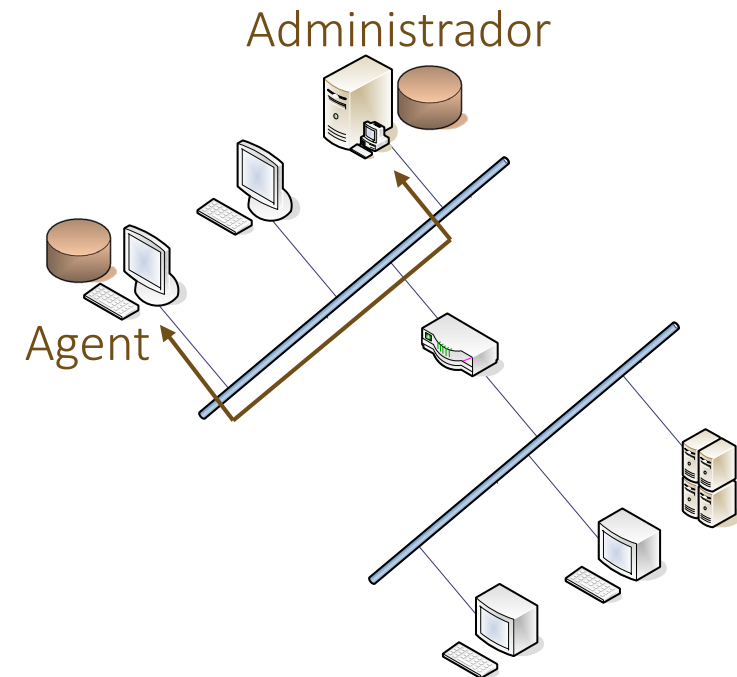
En l'arquitectura SNMP es defineixen 3 tipus d'entitats



## 6.8.10. Entitats SNMP

### Agents

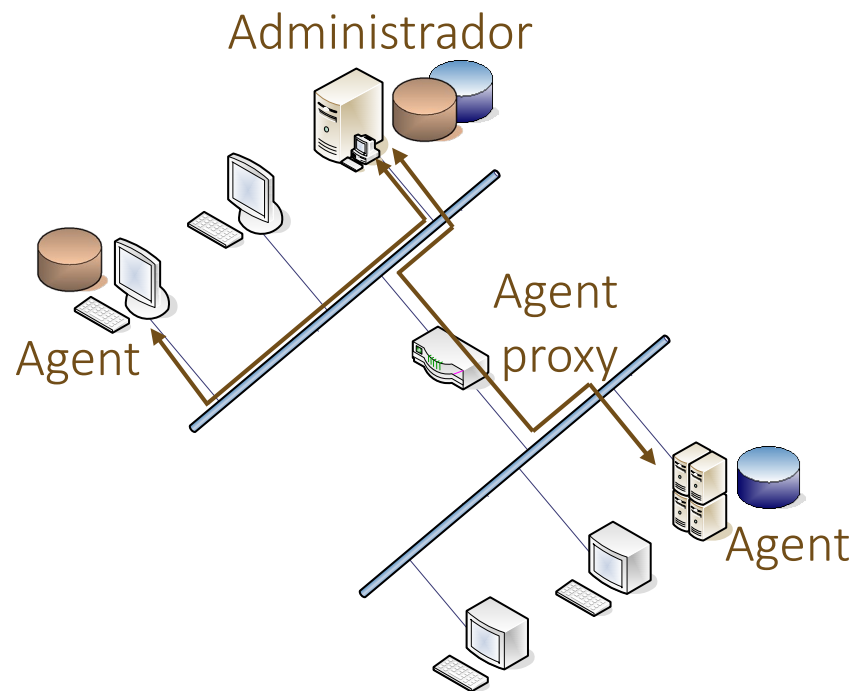
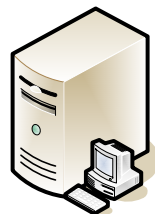
- Software instal·lat en tots els sistemes administrats
- Tasques
  - ✓ Respondre a les peticions dels administradors
  - ✓ Realitzar actualitzacions
  - ✓ Informar de problemes



## 6.8.10. Entitats SNMP

### Administradors

- Software instal·lat en tots els sistemes que gestionen la xarxa
- Tasques
  - ✓ Enviar i rebre missatges SNMP als sistemes administrats
  - ✓ Gestionar la xarxa

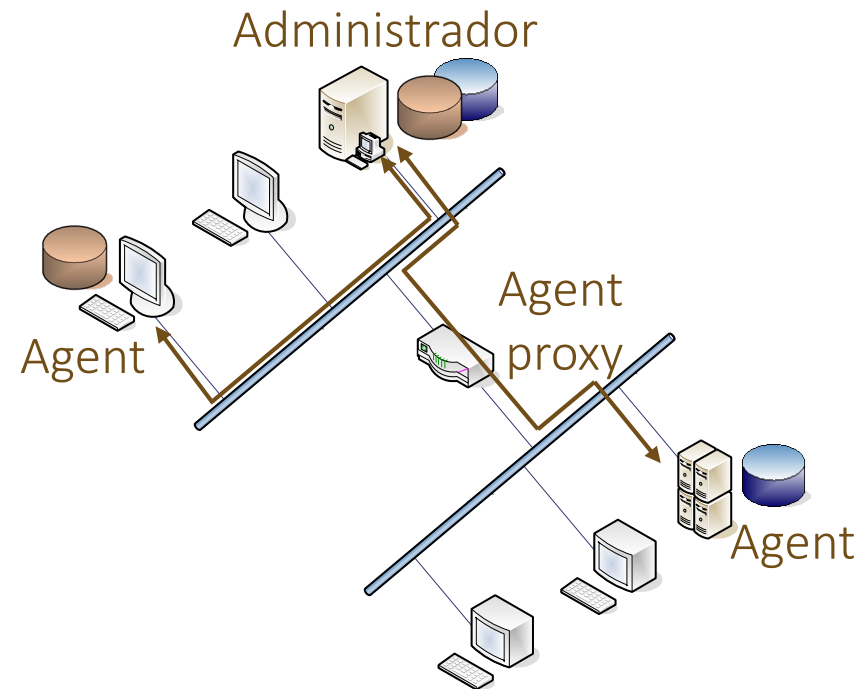
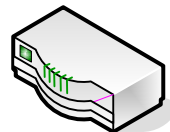




## 6.8.10. Entitats SNMP

### Agents proxy

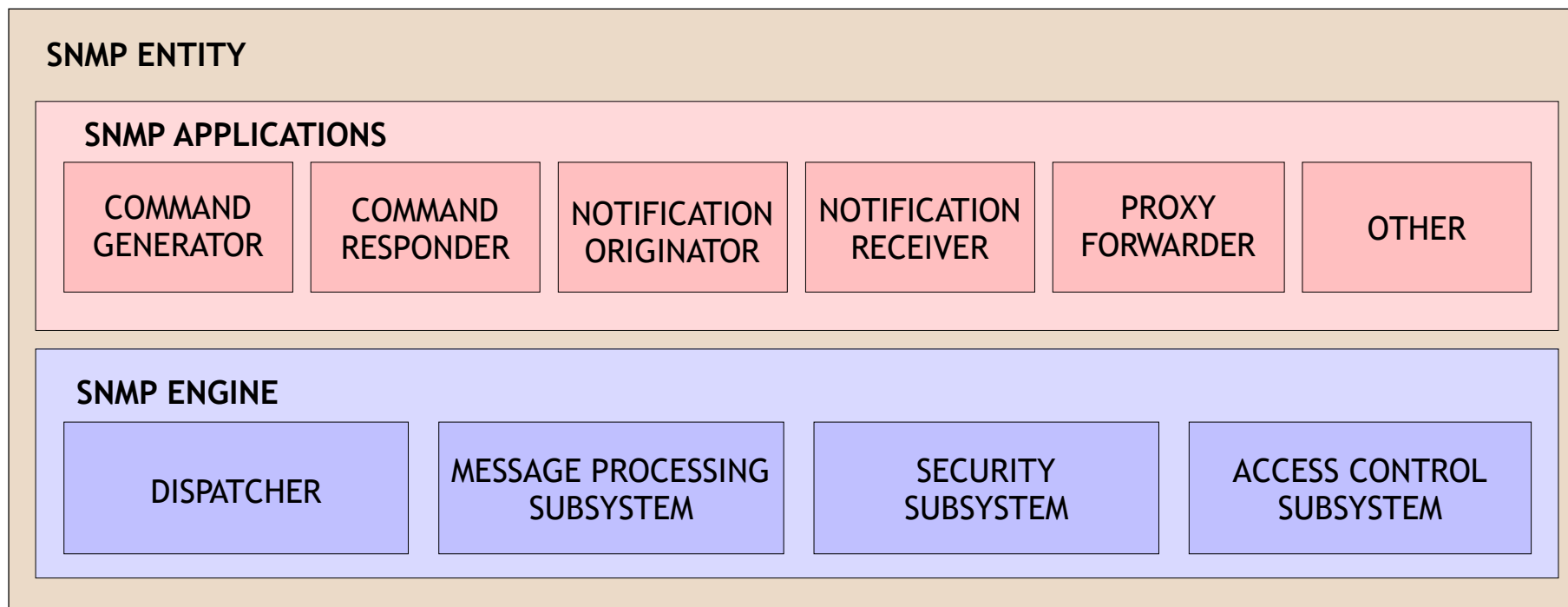
- Software instal·lat en un equip de gestió de xarxa
- Tasques
  - ✓ Redirecciona els missatges SNMP però no interpreta els objectes
  - ✓ Tradueix entre diferents versions SNMP (v2)
  - ✓ Tradueix entre SNMP i altres protocols de xarxa (v2)



## 6.8.10. Entitats SNMP

### Esquema general d'una entitat SNMP

Basat en  
SNMPv3



## 6.8.10. Entitats SNMP

### Esquema del motor de l'entitat

Basat en  
SNMPv3

*Dispatcher*: gestor dels missatges, és l'encarregat de distribuir els missatges tant internament com per la xarxa

*Message Processing Subsystem*: és l'encarregat d'interpretar els missatges

*Security Subsystem*: és el responsable de proporcionar seguretat en el flux de missatges

*Access Control Subsystem*: és el responsable d'otorgar els privilegis d'accés als objectes

## 6.8.10. Entitats SNMP

### Esquema de les aplicacions de l'entitat

Basat en  
SNMPv3

*Command Generator*: crea els missatges *request* i interpreta els *response*

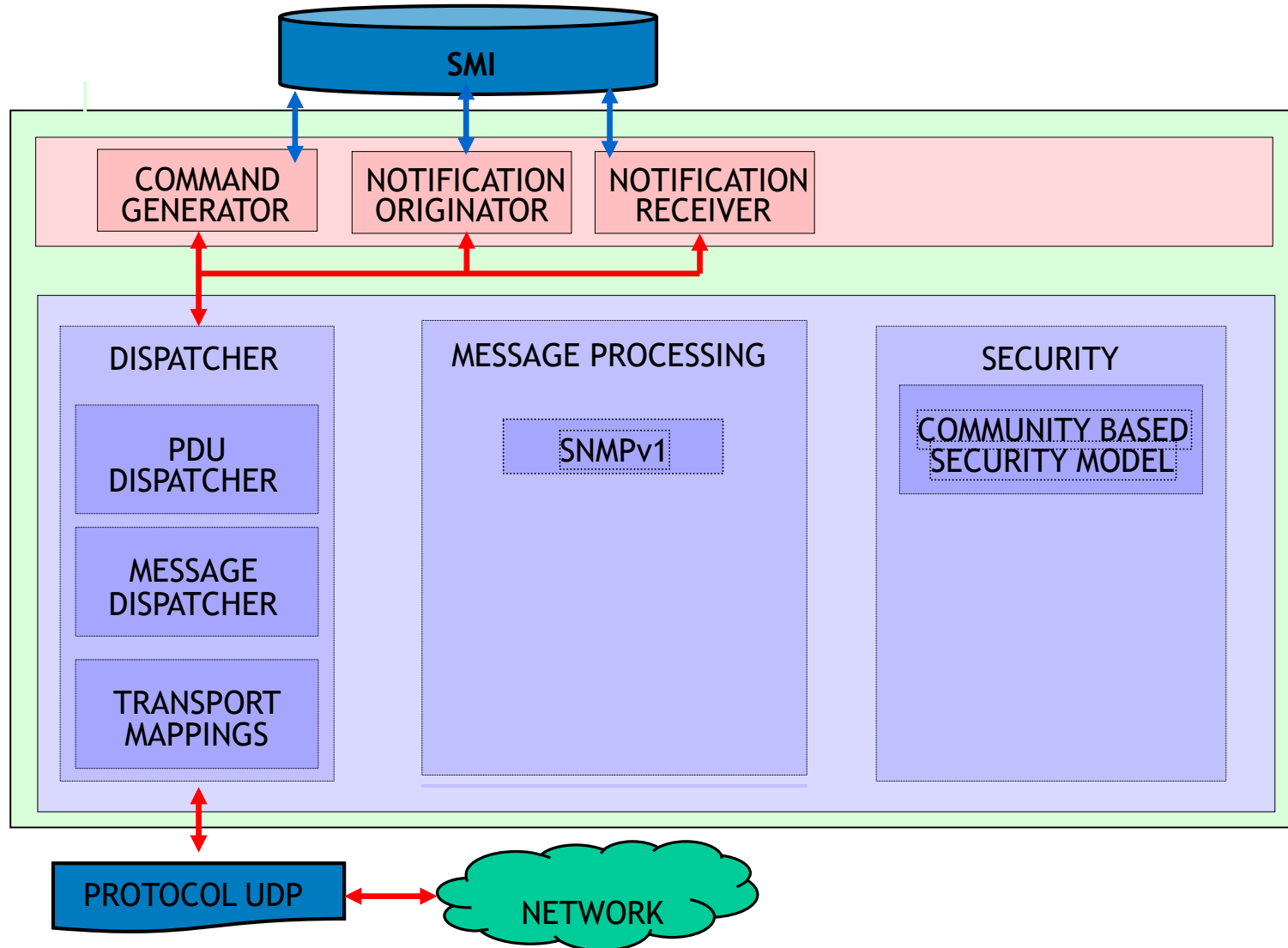
*Command Responder*: interpreta els missatges *request* i genera els *response*

*Notification Originator*: monitoritza la xarxa i genera els missatges *inform* i *trap*

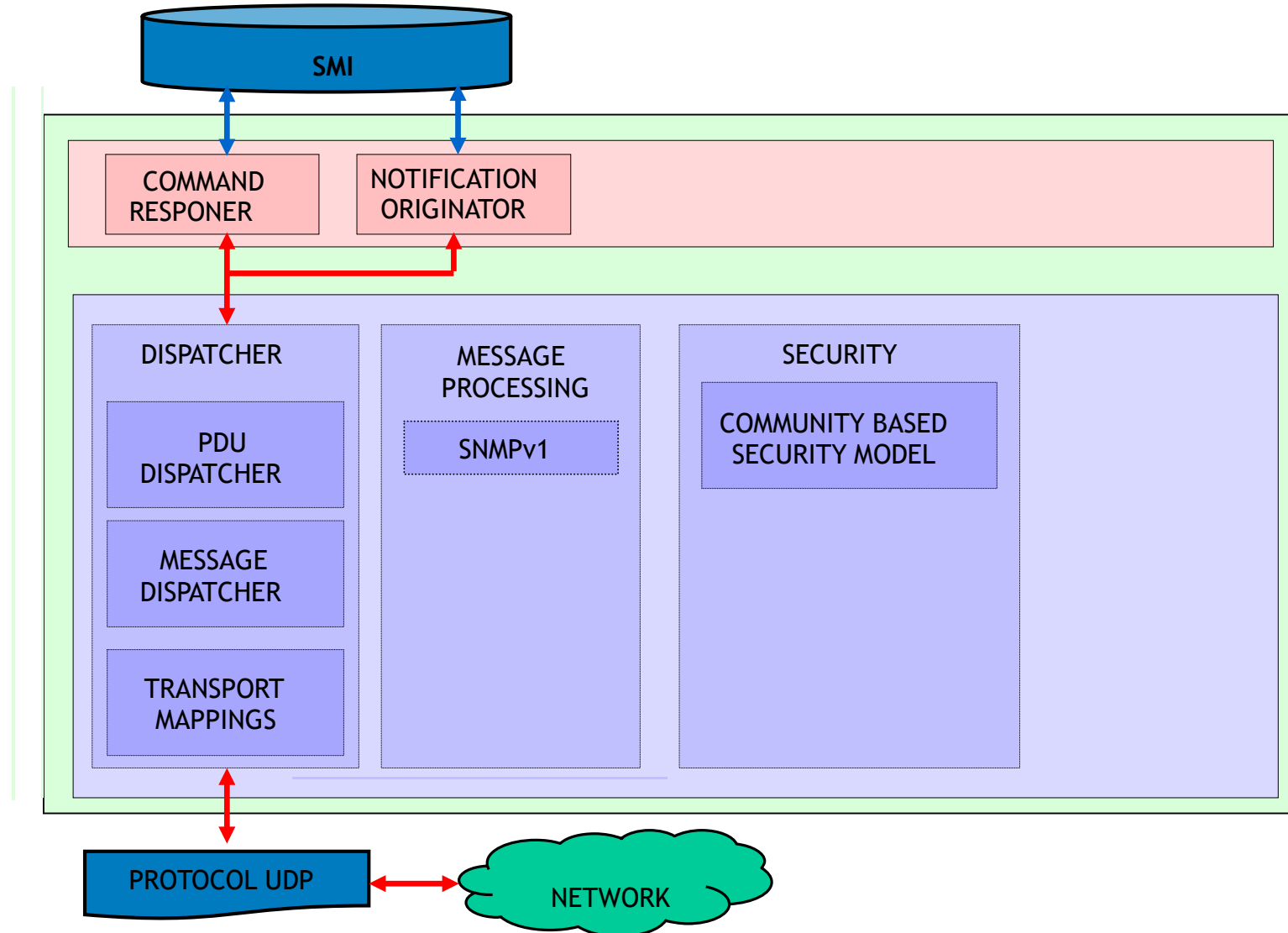
*Notification Receiver*: interpreta els missatges *inform* i *trap*. En el cas del *inform* genera el *response* corresponent

*Proxy Forwarder*: redireccions els missatges SNMP

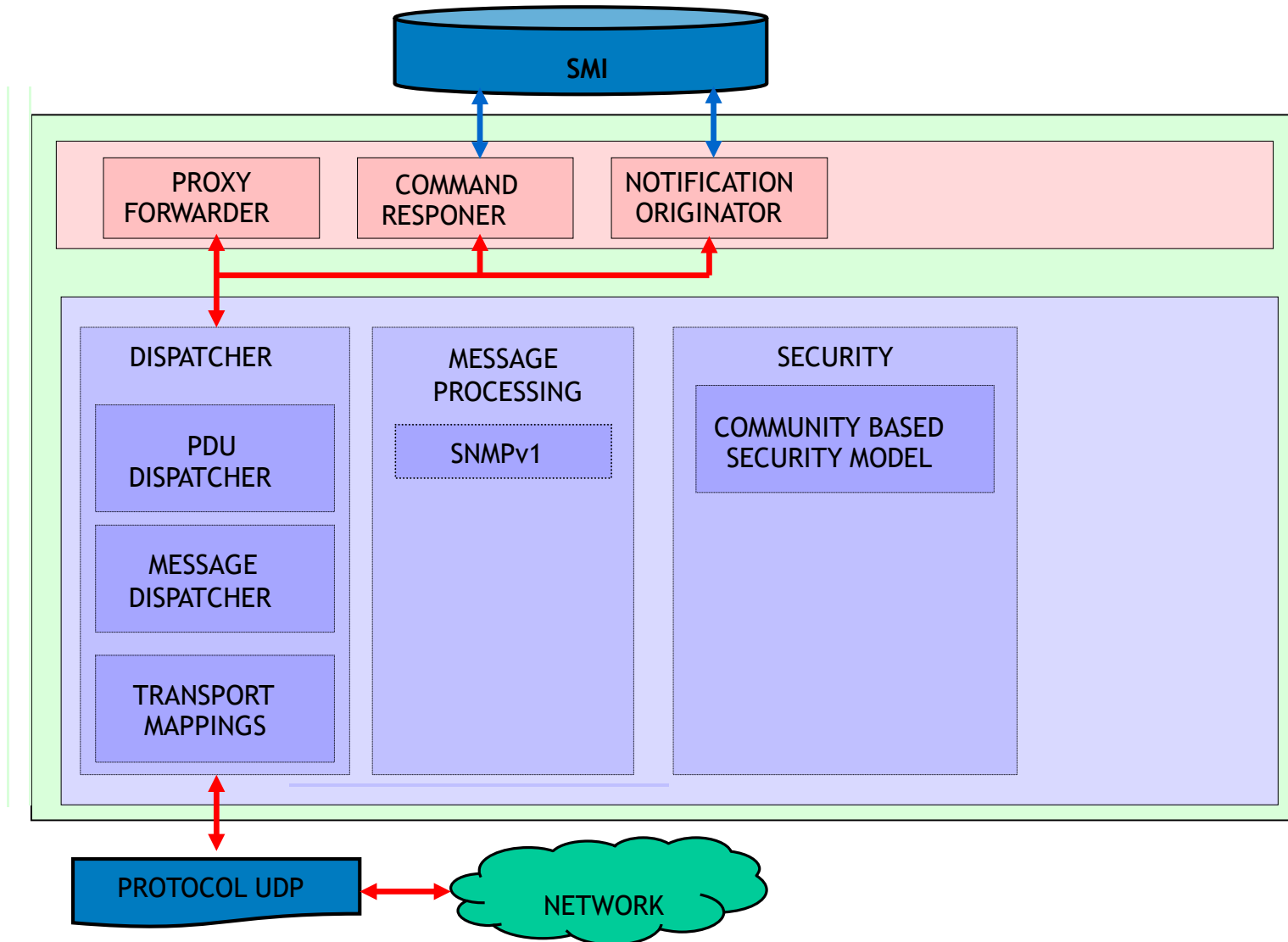
# Entitat administrador



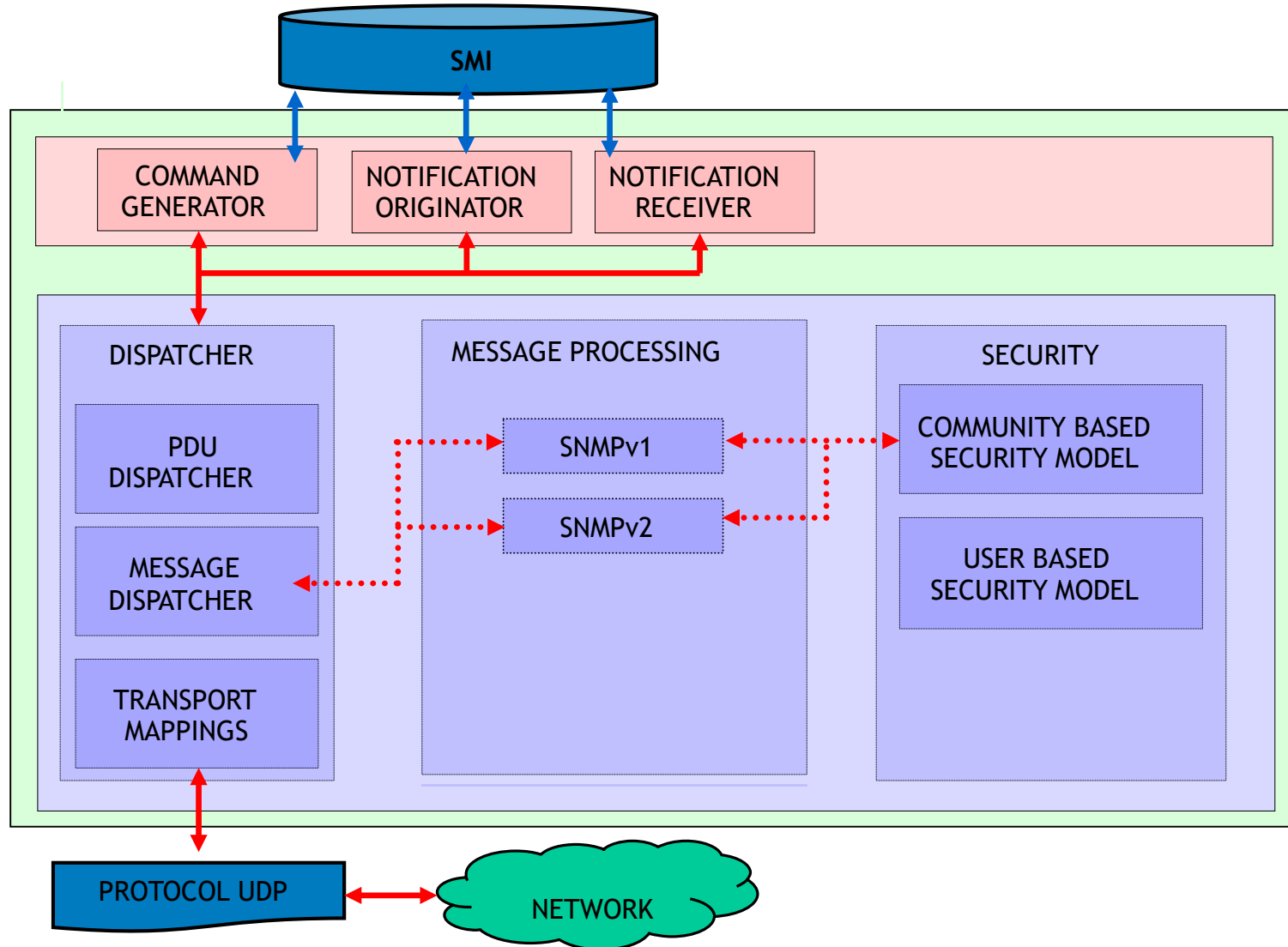
# Entitat agent



# Entitat proxy

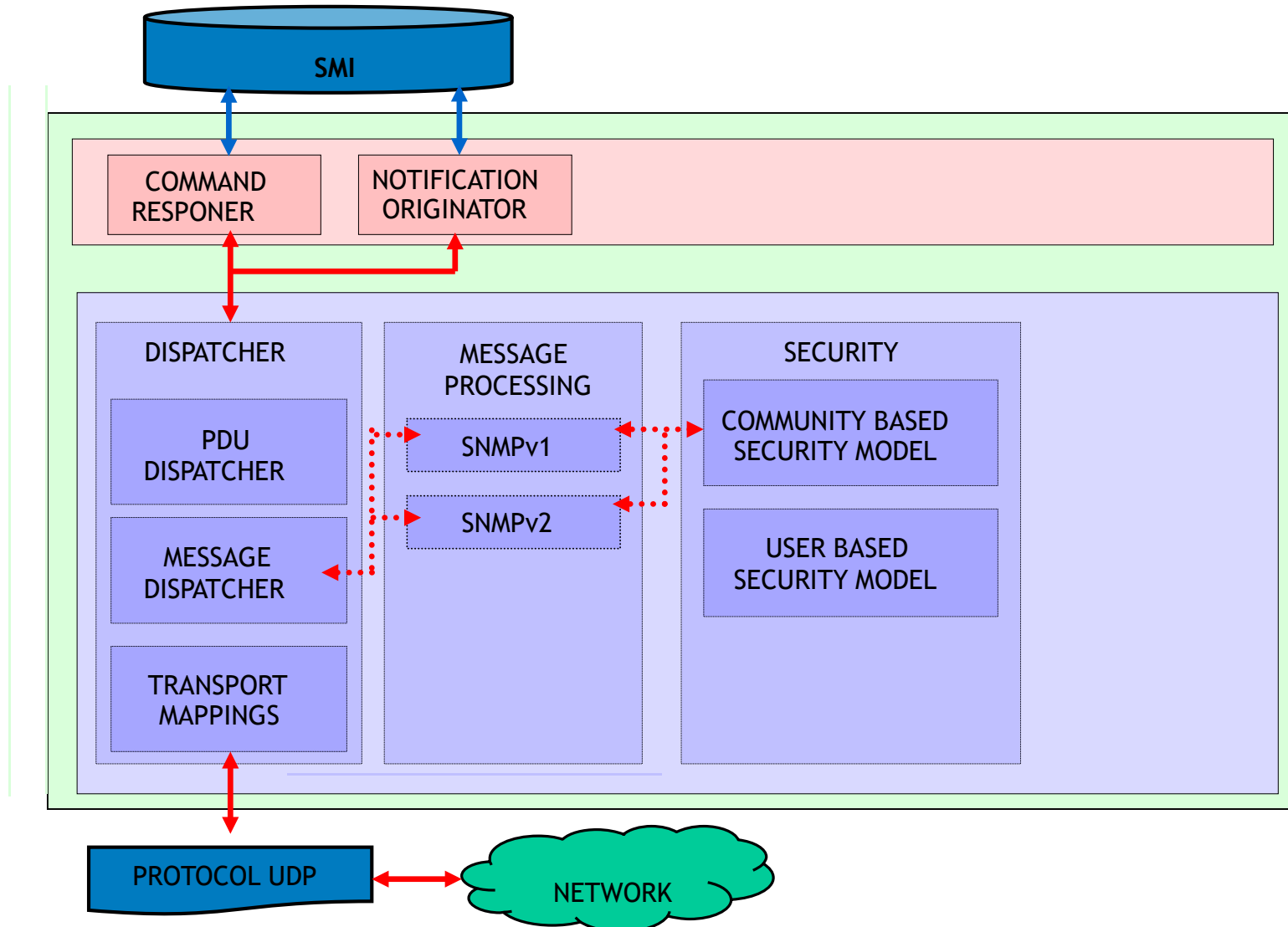


# Entitat administrador

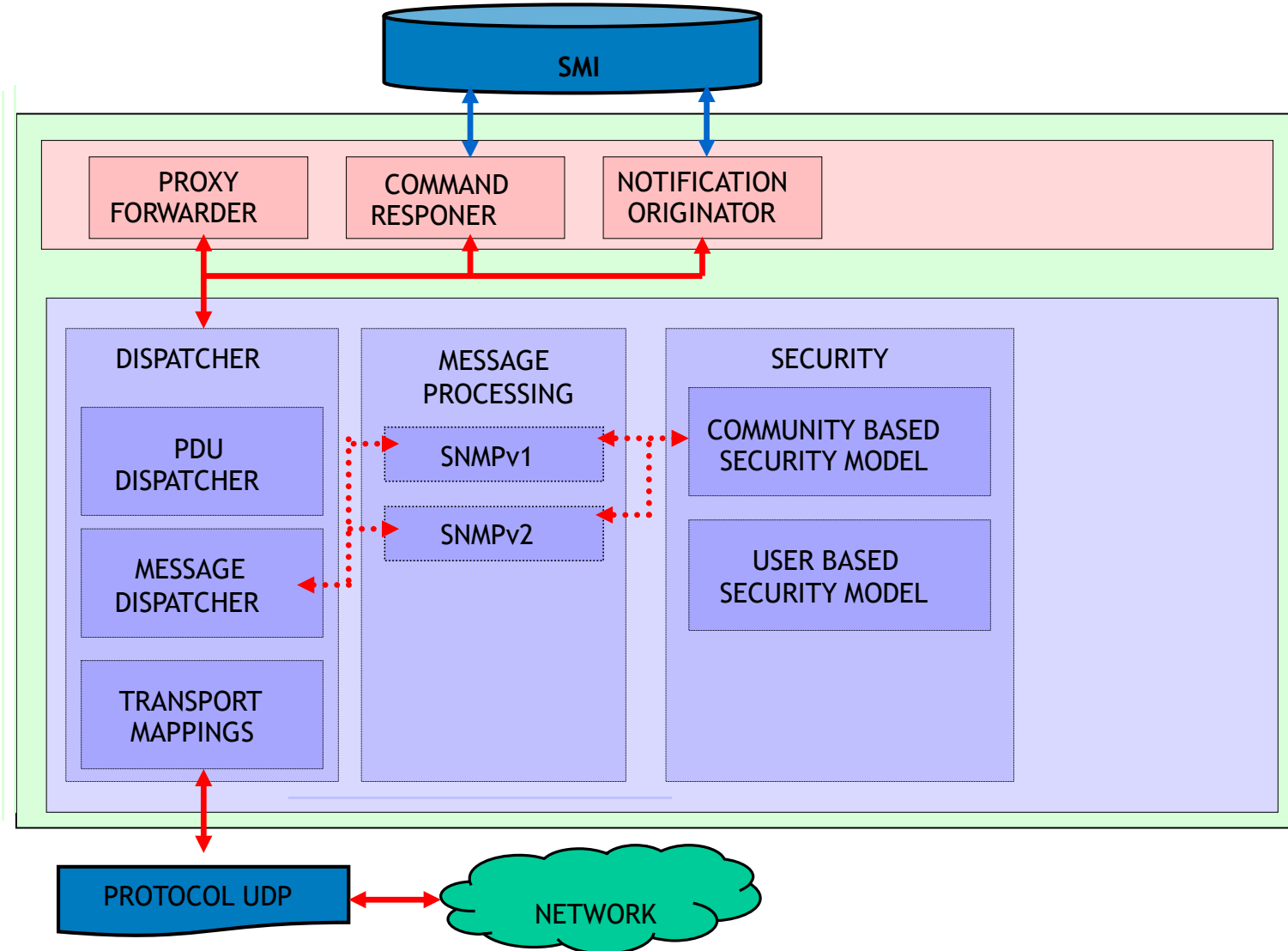




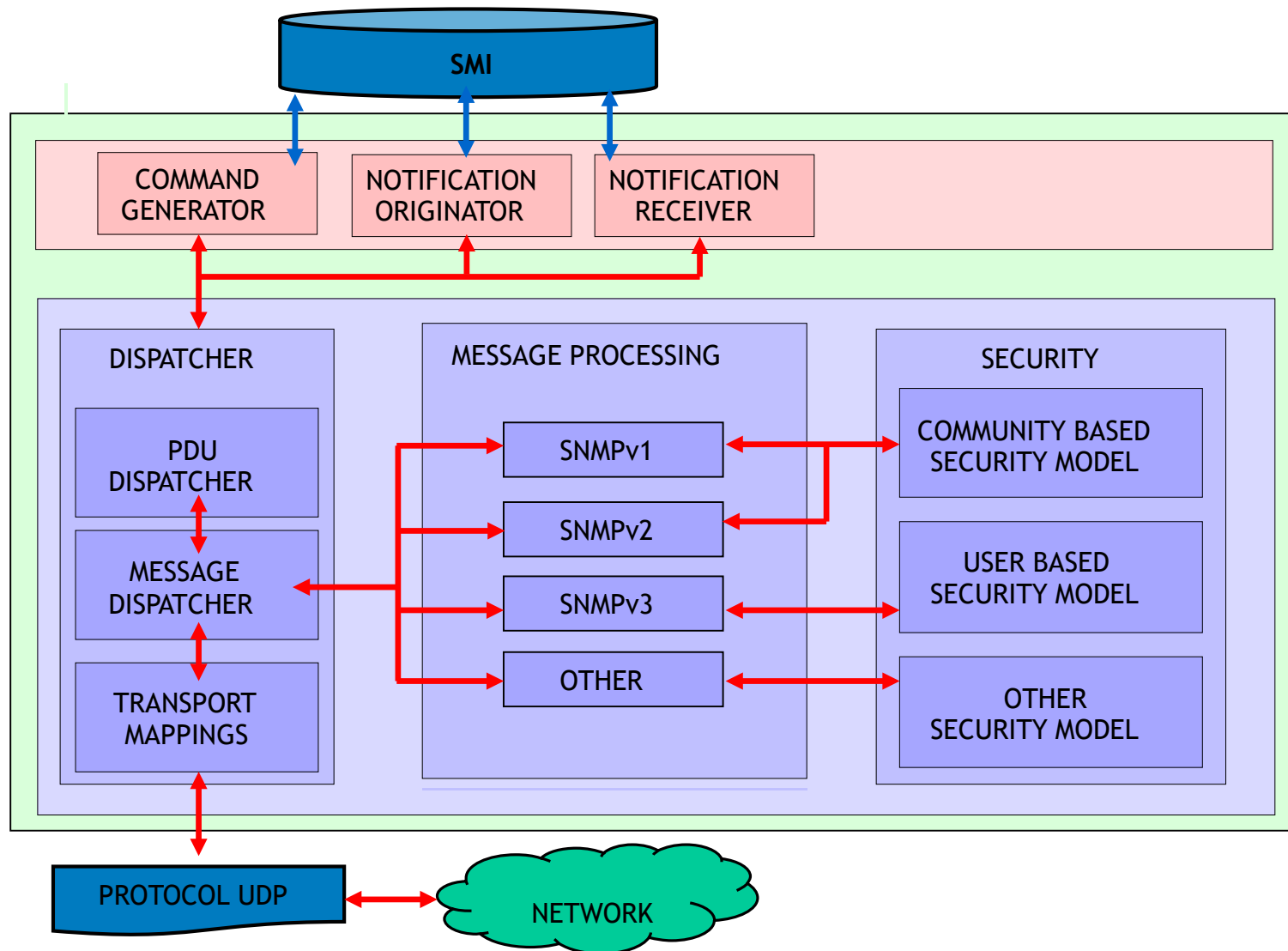
# Entitat agent



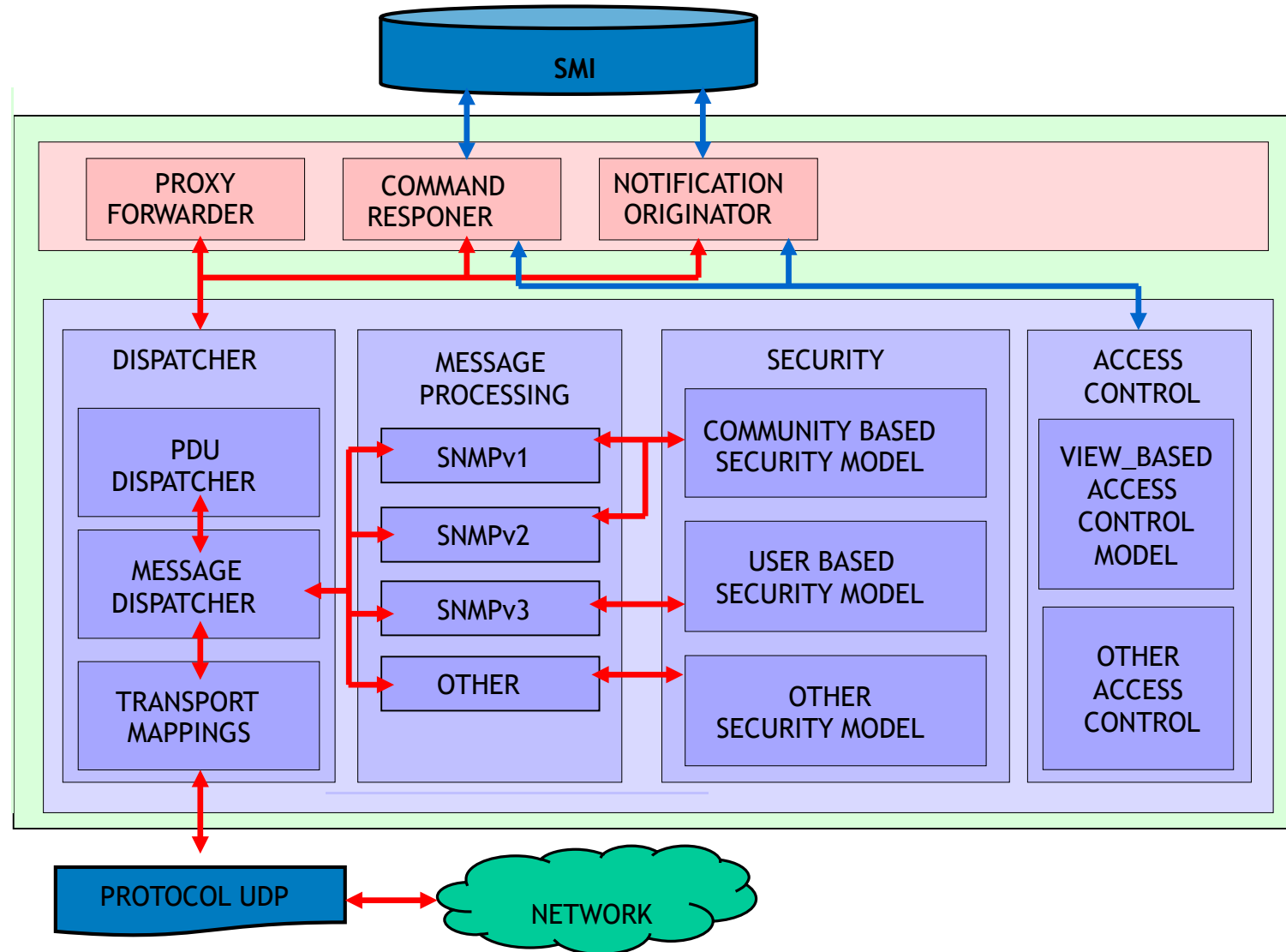
# Entitat proxy



# Entitat administradora



# Entitat agent





UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH



Aquest treball es publica amb una llicència Creative Commons  
Reconeixement – No Comercial 4.0 Internacional (CC BY-NC 4.0)