6. Servicios

Colaboradores:
Sr. Lluís Casals Ibáñez
Dr. David Rincón Rivera
Sra. Immaculada Ruiz Vela
Dr. Rafael Vidal Ferré

Dr. Daniel Guasch Murillo

Enero de 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú

6.1. Características del nivel de aplicación

Conceptes bàsics

Qué se entiende por "**nivel de aplicación**"?

- Programas que permiten acceder a los **servicios de comunicaciones** ofrecidos sobre las redes TCP/IP
  - E-mail
  - Transferencia de ficheros
  - Terminal remoto
  - transmisión de vídeo en tiempo real...
- Aplicaciones **muy diferentes** ⇒ servicios **muy específicos** ⇒ requisitos técnicos muy variados.
  - El correo puede sufrir retardos elevados, pero una transmisión de vídeo en tiempo real, no.
  - El vídeo puede admitir perdida de paquetes, el correo no.

6.1. Características del nivel de aplicación

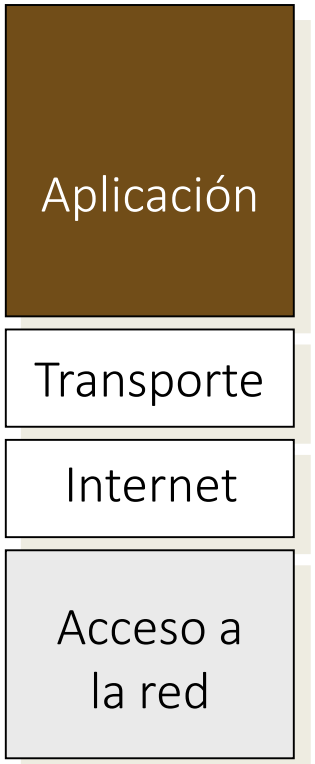
Conceptos básicos

Distinción entre **servicio** y **aplicación**

- **servicio:** concepto abstracto
- **aplicación**: pieza de software que da el servicio

Aplicaciones al modelo TCP/IP

- Es la capa más cercana al usuario
- Utiliza los servicios proporcionados por la capa de transporte
  - A través de la interfaz de programación, *sockets*
- Se encarga de abrir sesiones de trabajo



6.1. Características del nivel de aplicación

Clasificación de los servicios telemáticos

Orientados a conexión

- Se establece un circuito lógico, antes de iniciar la comunicación
- concepto parecido al de una llamada telefónica: llamar antes de hablar
- Ejemplo: transferencia de ficheros
  - Antes de transferir, abres una conexión con el servidor

No orientados a conexión

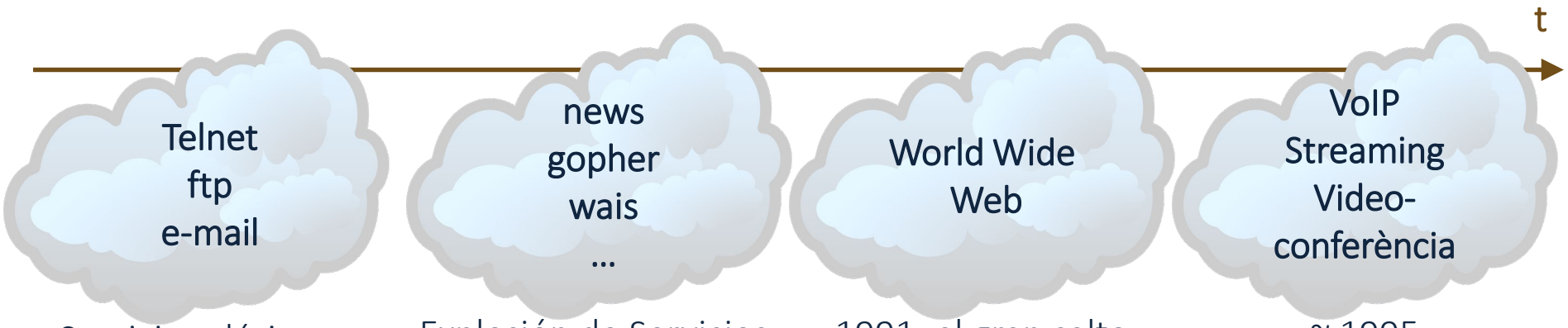
- No es necesario establecer ningún circuito previo; solo se envía la información
- Concepto semejante al de un telegrama: se envía sin conexión
- Ejemplo: correo electrónico

6.1. Características del nivel de aplicación

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)



Servicios clásicos:
1969-1973

Explosión de Servicios:
1980-90

1991: el gran salto
- No solo texto
- Atractivo
- Fácil de utilitzar

~ 1995
Tiempo real
Servicios tradicionales
de redes de circuitos

6.1. Características del nivel de aplicación

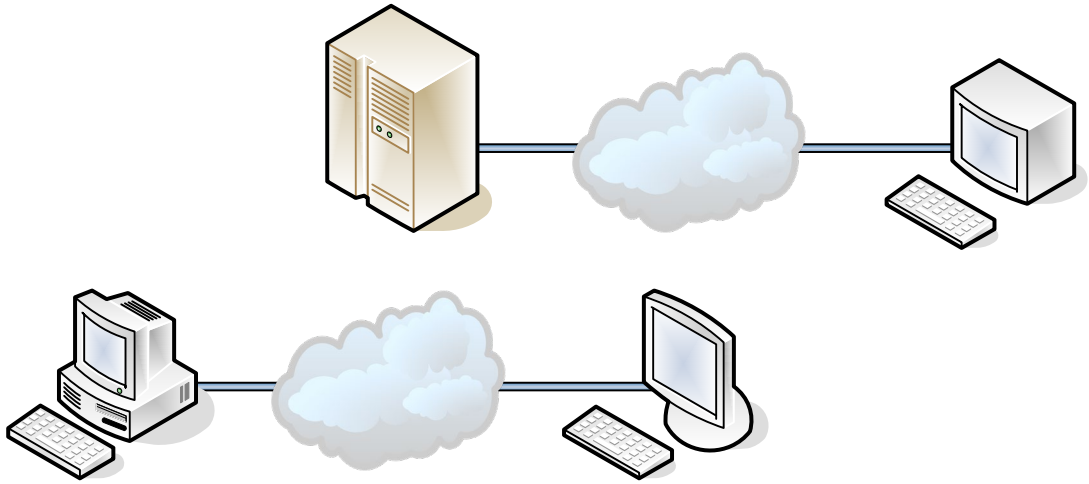
Arquitectura de las aplicaciones

Arquitectura de una aplicación

- modelo que siguen las aplicaciones que ofrecen un cierto servicio
- Hace referencia a como están construidas las piezas de software y como interactúan entre ellas

Dos modelos clásicos

- Cliente-servidor
- *Peer-to-peer*



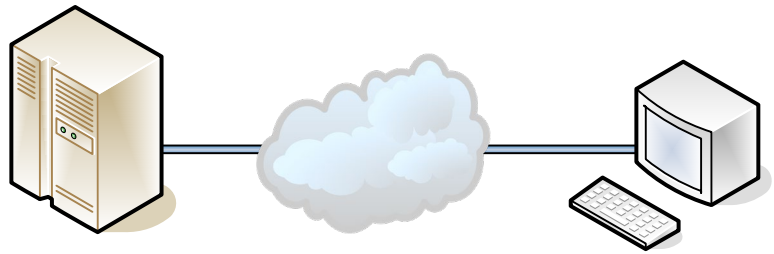
6.1. Características del nivel de aplicación

Arquitectura cliente - servidor

Los servicios TCP/IP suele seguir este modelo

Servidor

- Máquina que se dedica de manera permanente a prestar servicios al resto de hosts de Internet
- Corre programas que esperan conexiones (*daemons*)
- Puede atender simultáneamente varios clientes

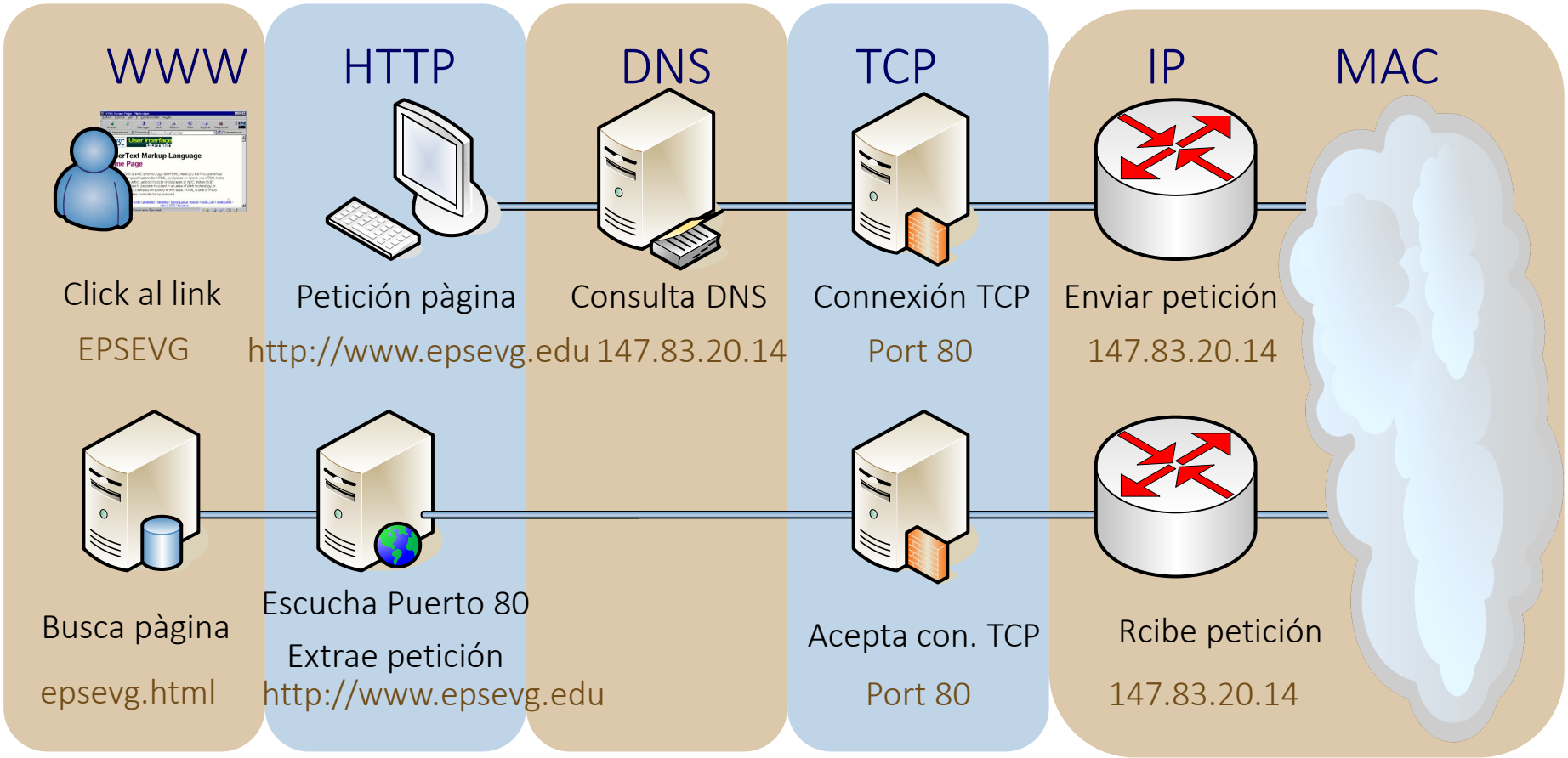


Cliente

- Máquina de usuario, “sencilla”
- Se conecta al servidor para pedir un servicio
- En principio, solo gestiona una conexión para cada servicio

6.1. Características del nivel de aplicación

Ejemplo de arquitectura cliente - servidor



6.1. Características del nivel de aplicación

Arquitectura peer – to - peer

Todas les máquinas tienen la misma importancia

- No se distingue entre cliente y servidor
- Todas pueden acceder al servicio (cliente) o ofrecerlo(servidor)

- Ejemplo: compartición de ficheros (Napster, Morpheus, etc)

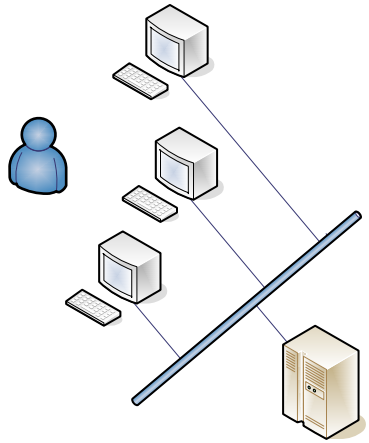
6.2. telnet

Terminal Networking: telnet

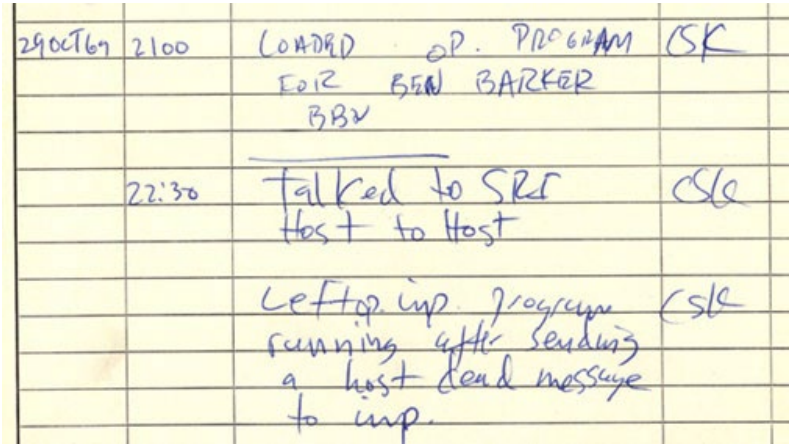
Telnet permite a un usuario local abrir una sesión de terminal en una máquina remota

- El usuario trabaja como si estuviera al lado de la máquina remota.

Es la aplicación más antigua de Internet: 1969



- Objetivo inicial de Internet: **compartición** y **acceso remoto** a supercomputadores.
  - Anécdota: 3 terminales diferentes al despacho del jefe de ARPA



6.2. telnet

Características básicas

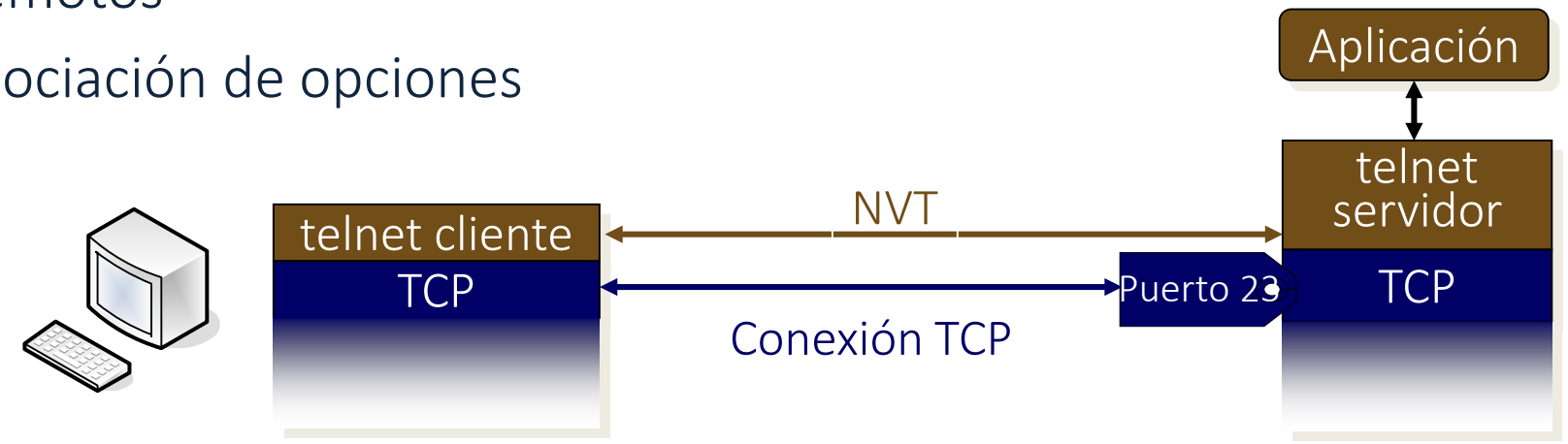
Definido al RFC 854

Telnet usa el servicio de transporte TCP

- Dos comunicaciones simétricas por el puerto 23

Servicios básicos:

- Define un terminal virtual de red (*Network Virtual Terminal*, NVT) que proporciona una interfaz estándar a los sistemas remotos
- Permite la negociación de opciones



6.2. telnet

NVT, Network Virtual Terminal

Terminal Virtual de red

- Simula una pantalla y un teclado
- Diseñado como Protocolo half duplex y con intercambio línea a línea
  - solo un host transmite simultáneamente. Después de enviar una línea, el cliente espera a recibir datos del servidor. El servidor envía los datos y después un *go ahead*, indicando que el cliente puede transmitir
- Posteriormente opción carácter-a-carácter

Caracteres ASCII de 7 bits rellenados a 8 con un 0 inicial

- Cada línea acaba con una combinación CR y LF (ASCII)
- Los caracteres que empiezan por 1 son ordenes

Normalmente solo se utiliza durante un periodo de tiempo corto, para negociar las opciones de algún emulador de terminal.

- Tipo de terminales: ASCII, IBM 3270, **vt100**

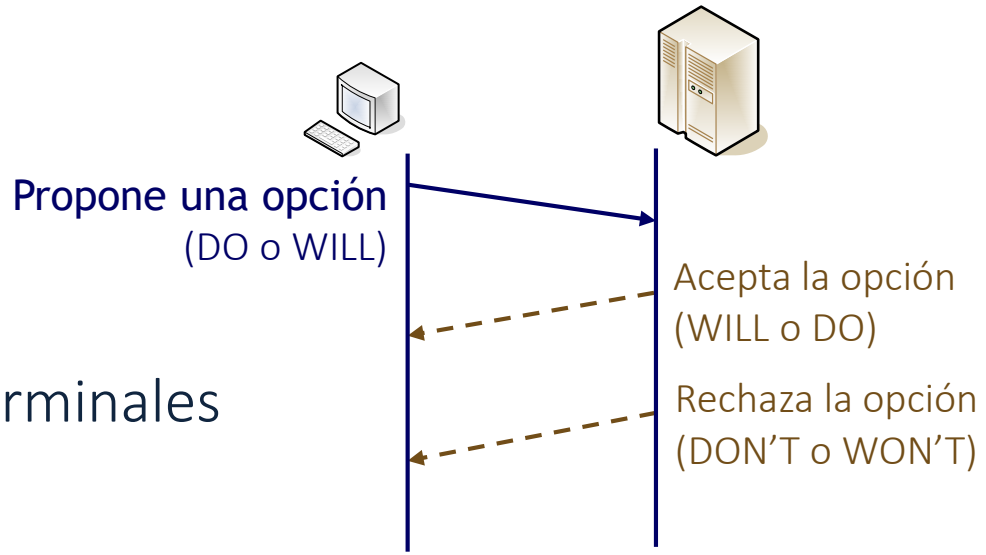
NVT también se utiliza a FTP, SMTP, Finger...

6.2. telnet

Establecimiento de la comunicación

Autentificación:

- login + pasword



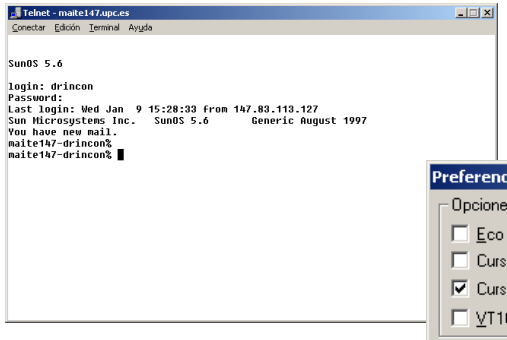
Negociación inicial de opciones

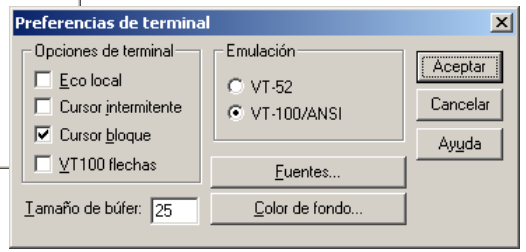
- Para ir más allá del NVT, adaptándose a los terminales
- Opciones:
  - Eco local o remoto
  - Consultar el estado del extremo opuesto (status)
  - 7 / 8 bits por carácter
  - Intercambio de información sobre el terminal
    - tipo, velocidad, CR o CR + LF, ...
    - Control de flujo
    - Edición modo línea o carácter
    - Encriptación
    - Autenticación

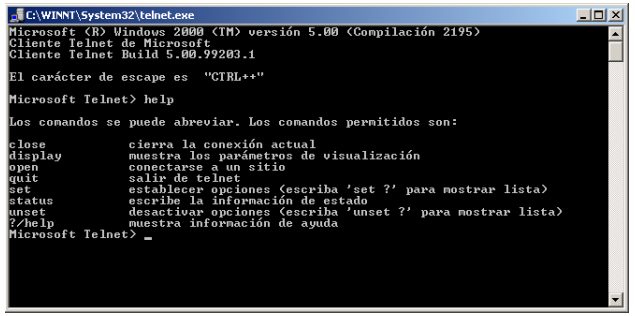
6.2. telnet

Cliente Telnet

Integrado al sistema operativo (Unix, Windows...)







Sintaxis: telnet[host][puerto]

- Podemos hacer telnet a otros puertos que no son el 23
- Posibilidad de pedir help

6.2. telnet

Cliente Telnet

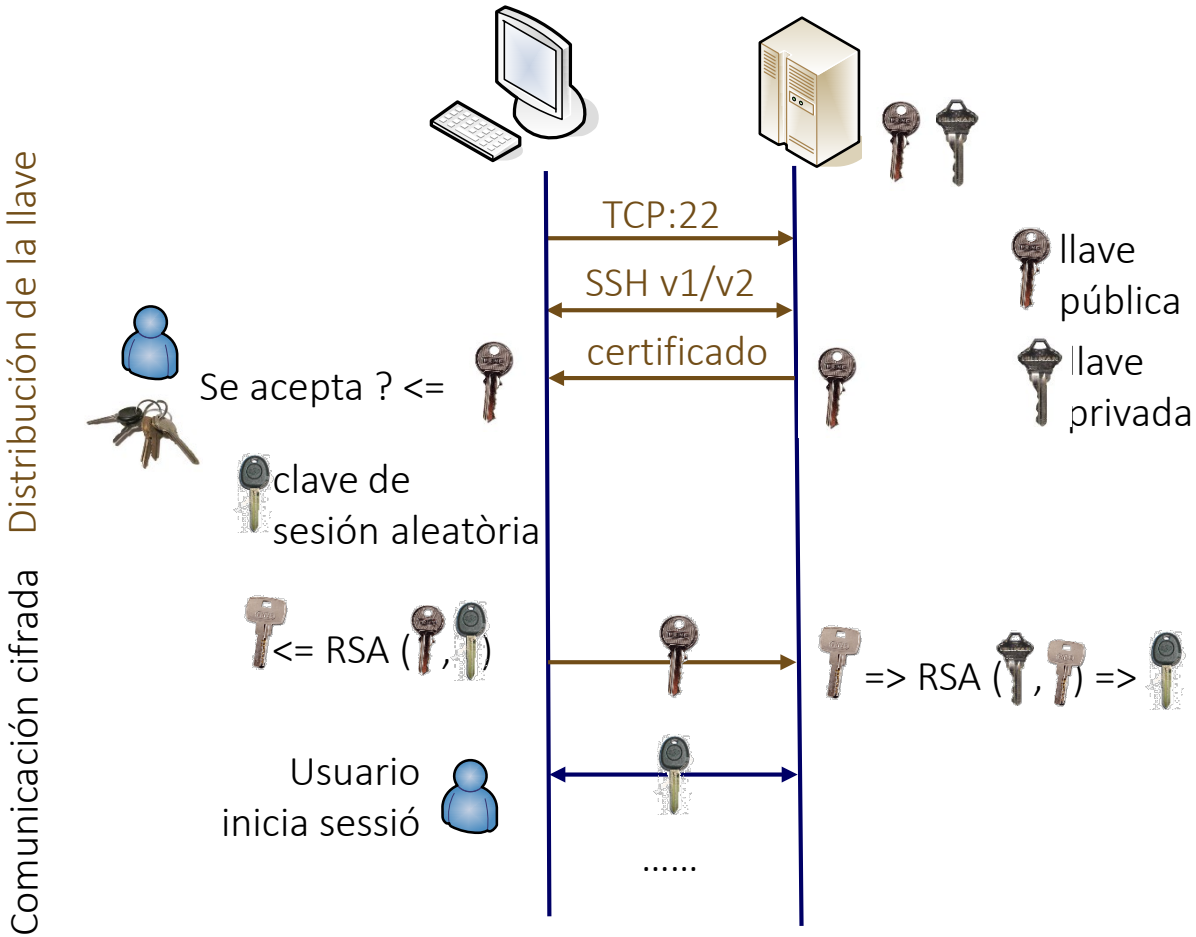
Ordenes básicas

`close`	cierra la conexión en curso.
`logout`	hace que el usuario se desconecte y cierra la conexión.
`open`	abre una conexión a un host.
`modo`	indica si la transmisión se hace carácter a carácter (*character*) o se envía una línea de caracteres (*line*) cuando se genere un EOL (*end of line*).
`quit`	sale del programa.
`set`	activa parámetros de operación (echo, escape, erase, kill, eof, quit)
`status`	visualiza la información del estado de la conexión, el modo, el eco y el carácter de escape.

carácter de control (CTRL], CTRL++): permite salir al modo de ejecución de ordenes. Se puede volver al terminal con return.

6.2. telnet

SSH (Secure Shell) mejora y securiza el telnet



6.3. FTP

Transferencia de ficheros: FTP

Una de les aplicaciones básicas de Internet

FTP (*File Transfer Protocolo*)

- Uso compartido de ficheros entre sistemas remotos
- Posibilidad de enviar, recibir, borrar, y gestionar ficheros y directorios

Diversos Protocolos

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

6.3. FTP

Características del FTP

Supone que se dispone de un servicio fiable extremo a extremo (TCP)

- Dos conexiones TCP
  - Control por el puerto 21: sesión NVT
  - Datos por el puerto 20

- Conexión de control:
  - Diálogo de comandos (cliente) y códigos de respuesta (servidor)
  - Empieza en el momento en que el cliente se conecta al servidor

cliente (y) ⟷ (21) Servidor

- Conexión de datos solo se abre para copiar ficheros o hacer listados
  - El cliente reserva un puerto x y le indica al servidor (comando puerto) que a continuación se conecta

Client (x) ⟷ *(20) Servidor*

Características del FTP

- Acceso interactivo: persona o máquina
  - Códigos de respuesta de 3 dígitos para posibilitar control remoto
  - El primero indica el tipo de operación
    - 2xx : éxito
    - 1xx : la acción ha empezado
    - 3xx : un punto intermedio ha sido logrado con éxito
    - 4xx : error transitorio
    - 5xx : error permanente

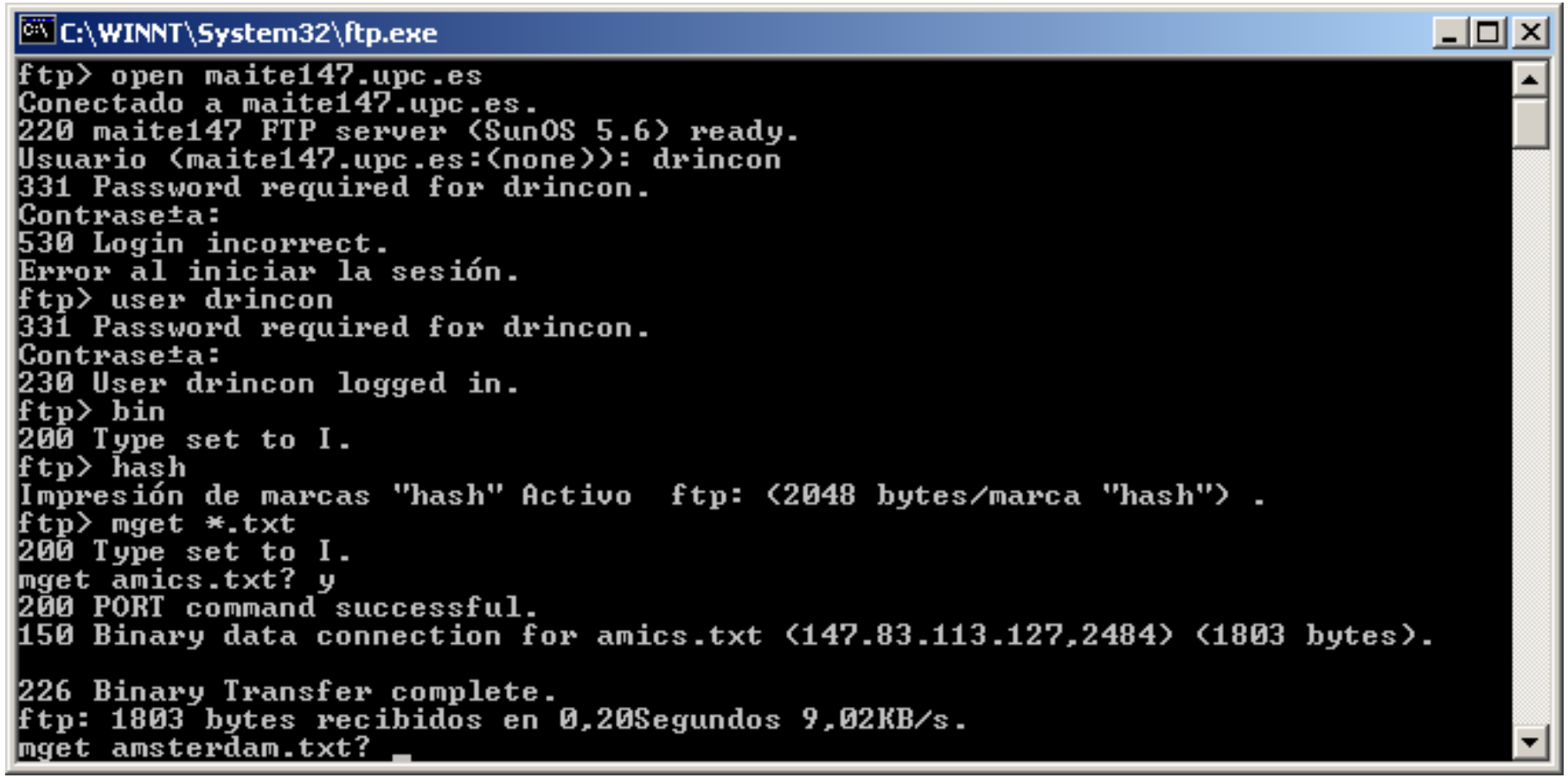
6.3. FTP

Características del FTP

- Formato de la información
  - ASCII: para ficheros de texto ASCII
    - Se interpretan los caracteres especiales como cambio de línea
  - Binario: para otros ficheros
    - Se trata todo el fichero como un flujo continuo de bits
  - Es muy importante utilizar el modo correcto !!
    - Ejemplo: un fichero Word NO es un fichero de texto
- Asignación de nombres
  - fichero local → fichero remoto
- Autentificación:
  - login + pasword

6.3. FTP

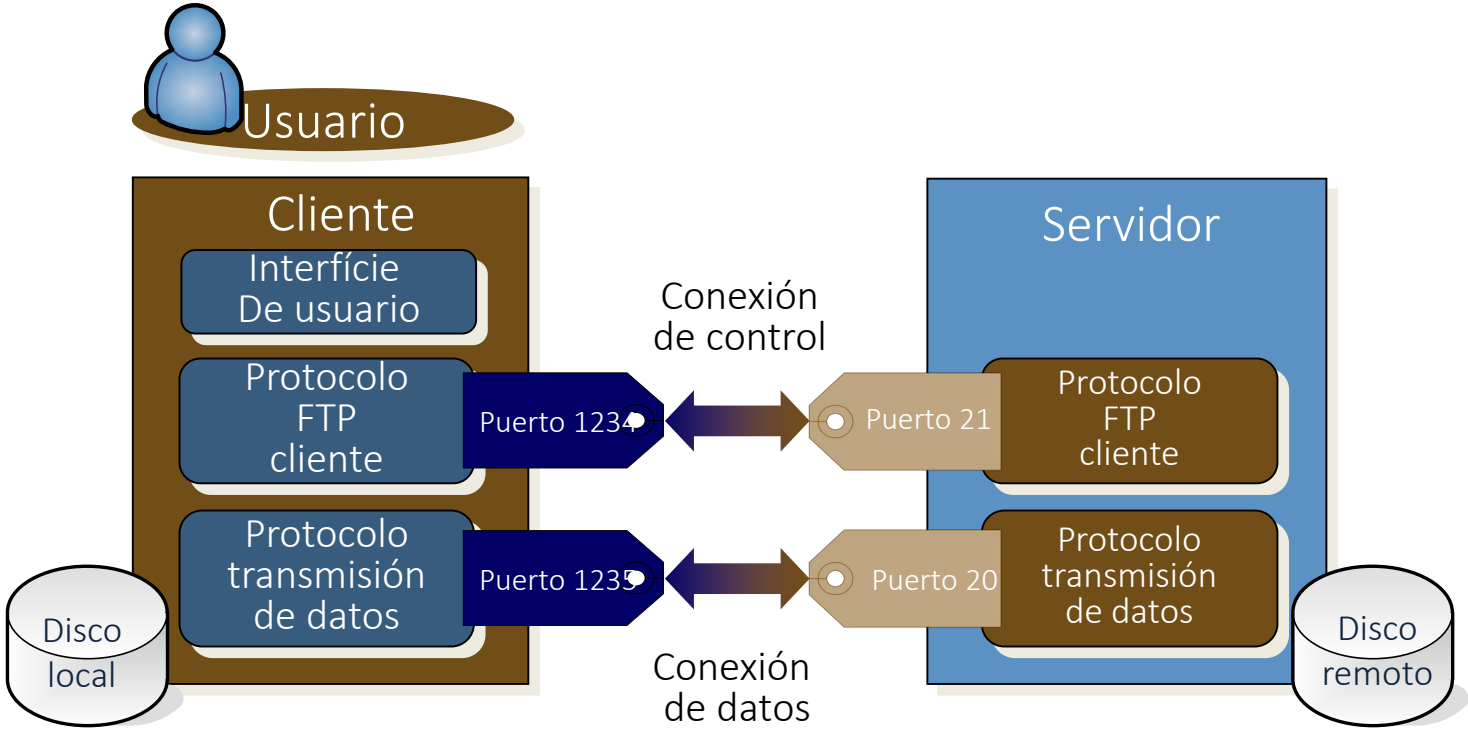
Ejemplo de sesión FTP con Windows 2000



6.3. FTP

Modelo del FTP

- Daemon FTP lanzado por inetd
- Para cada transferencia de datos se crea una nueva conexión
- El protocolo de la conexión de datos es una particularización del telnet



6.3. FTP

Ordenes de usuario en el cliente FTP

Funciones de transferencia de datos:

- Copiar ficheros entre hosts: **get, put, mget, mput**
- Añadir un fichero local a un fichero remoto: **append**

Funciones de control:

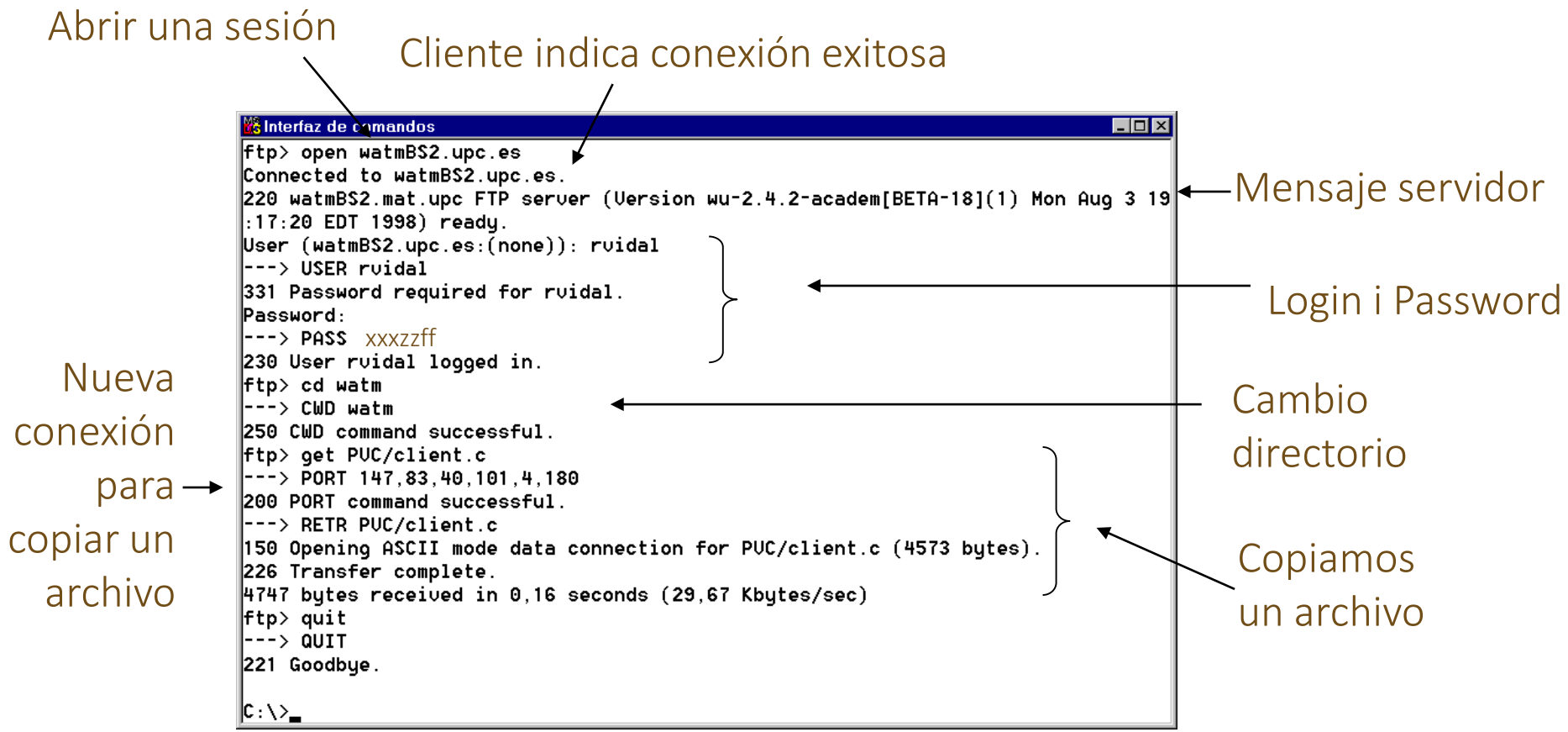
Identificar el tipo de transferencia: **ascii, binary**
Confirmar cada fichero transferido: **prompt**
Salir: **quit, bye**
Ayuda: **help**
Abrir / cerrar una conexión: **open, close**

Funciones de control sobre ficheros:

- Listar un directorio: **ls, dir**
- Imprimir el directorio actual: **pwd**
- cambiar de directorio: **cd, lcd**
- Borrar / renombrar un fichero: **delete, rename**
- Mostrar el estado de la conexión: **status**

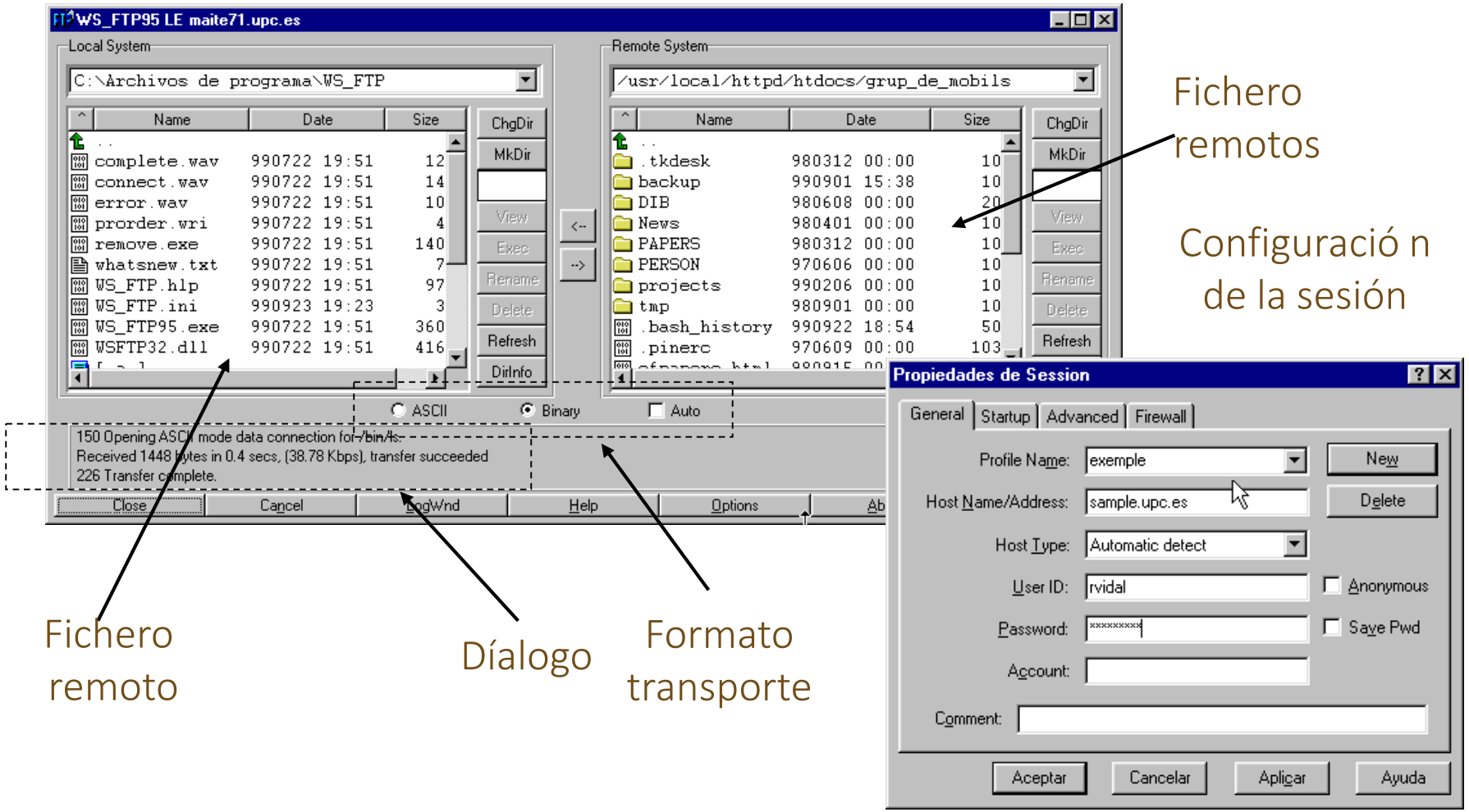
6.3. FTP

Ejemplo de dialogo de control FTP



6.3. FTP

Ejemplo de interfaces FTP



6.3. FTP

FTP anónimo

- Algunos servidores permiten el acceso anónimo
- Identificación por login `anonymous`, pasword = 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 Protocolo

FTP es difícil de implementar y ofrece más de lo que se quiere utilizar en determinadas ocasiones.

En determinadas ocasiones es mejor utilizar TFTP

- Protocolo pequeño y fácil de utilizar
- TFTP solo permite la transferencia de ficheros
- TFTP no exige autentificación
  - Posible agujero de seguridad !!
- Servidor tipo *nowait* (1 solo transferencia a la vez)
- La mayoría de errores provocan un final de la conexión:
  - No se puede satisfacer la petición:
    - No se encuentra el fichero, acceso denegado, No existe el usuario
  - Paquete construido erróneamente
  - Perdida del dispositivo durante la conversación

6.3. FTP

TFTP: Trivial File Transfer Protocolo

Utiliza el UDP como mecanismo de transporte. **No es fiable**

- Tamaño máximo: 512 bytes
- Se confirma cada paquete

Utilizado para estaciones sin disco (RARP, BOOTP)

Típico en descarga de imágenes en equipos (routers, etc.)

Diferentes modos de transferencia

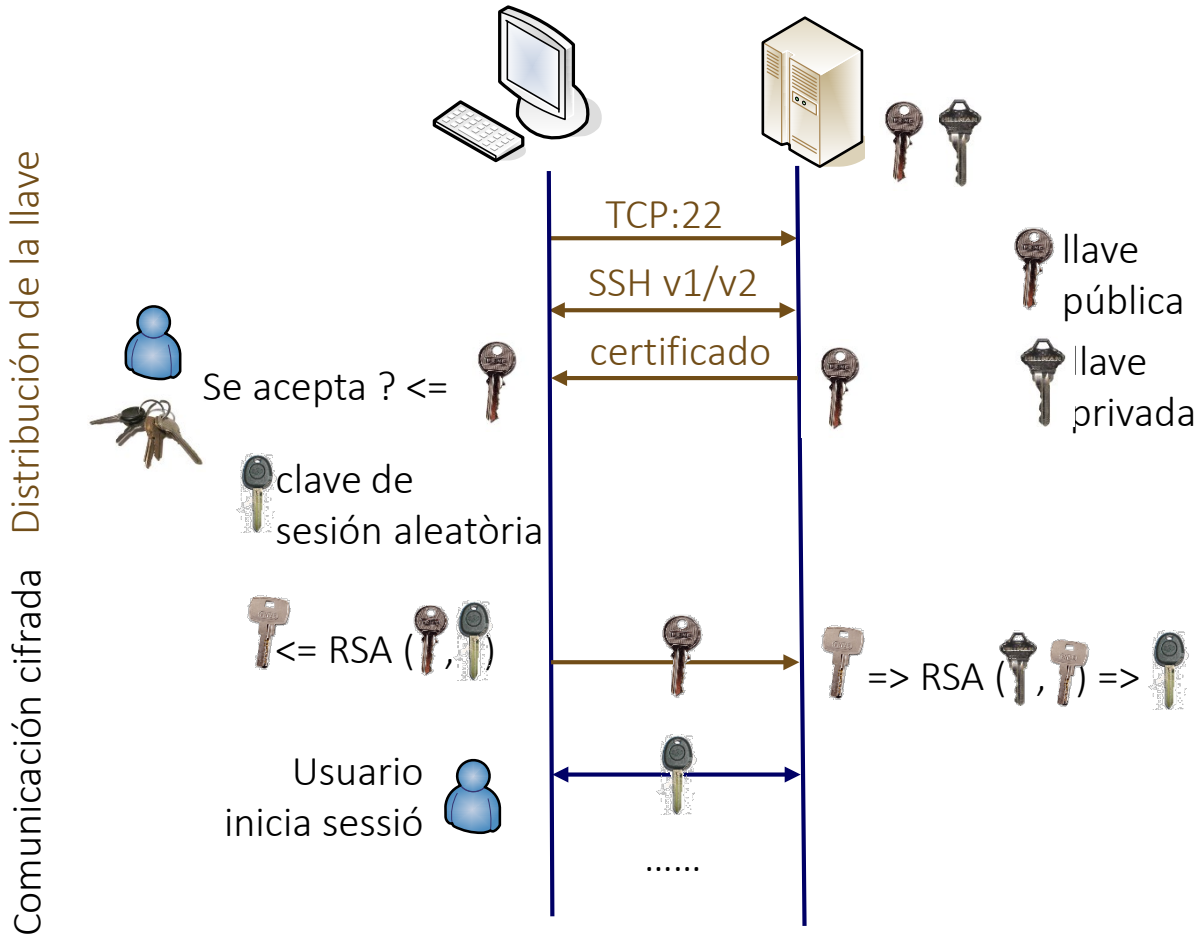
- netascii: texto
- octet: formato binario
- mail: datos enviados a un usuario en lugar de a un fichero

Interfaz de usuario: *tftp host*

- put, get, sdadus, ascii, binario

6.3. FTP

sftp (Secure ftp) mejora y securiza el ftp



6.4. Correo electrónico

Conceptos básicos sobre correo electrónico

E-mail: probablemente el servicio “estrella” de la red

Evolución de los primeros sistemas de comunicación entre usuarios de un mismo ordenador

- Orden **`mail`** de Unix
- Permite enviar mensajes de texto que quedan en el buzón (*mailbox*) del usuario receptor

1971: Ray Tomlinson escribe el primer protocolo de *transferencia* de correo entre hosts Unix

- Dirección = nombre_usuario + nombre_host
- Separación por @

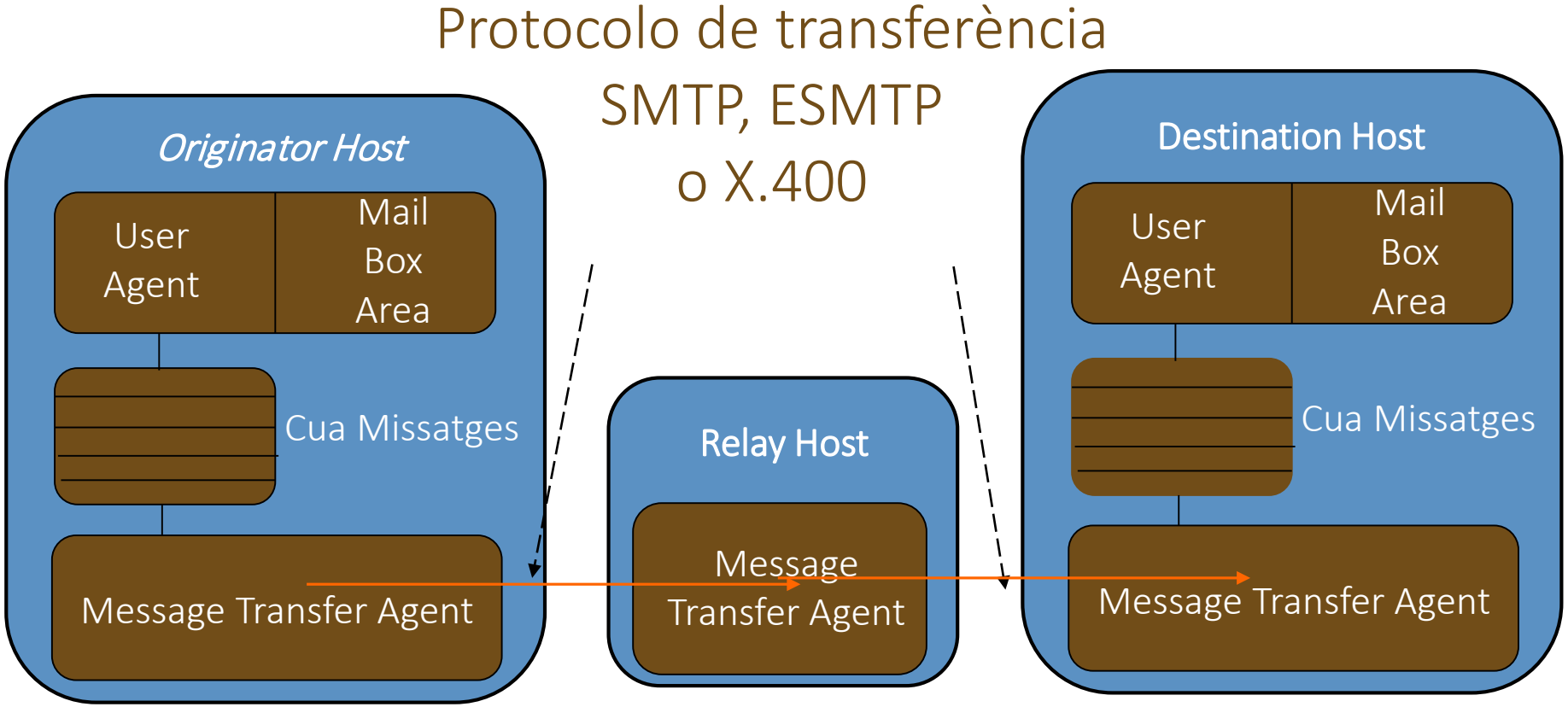
6.4. Correo electrónico

Elementos de un sistema de correo electrónico

- User Agent (UA) o cliente e-mail: que nos permite consultar, contestar y editar el correo
- Mensaje: dos partes, sobre y contenido, con un formato determinado
- Buzón mailbox: donde se guardan los mensajes recibidos o para enviar
- Mesage Transfer Agent (MTA): elementos encargados de transportar el correo por la red
- Store-and-forward : técnica de transferencia de mensajes. El mensaje va pasando por MTAs que lo van reenviando hasta que llega al buzón de destino

6.4. Correo electrónico

Elementos de un sistema de correo electrónico



6.4. Correo electrónico

Protocolos de correo electrónico

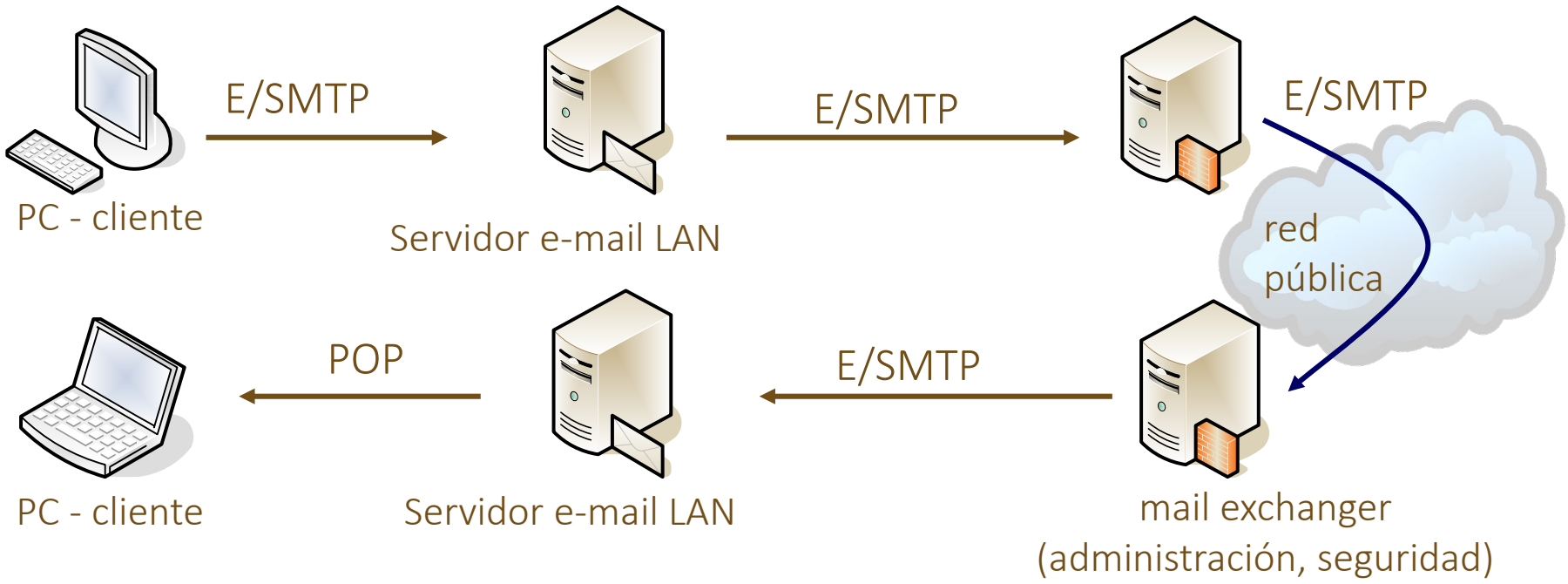
- Standard for the format of ARPA Internet Messages (RFC 822)
  - Describe el formato de los mensajes
- **SMTP** (Simple Mail Transfer Protocol):
  - Servidor clásico de transferencia de correo (RFC 821)
  - Transferencia de ficheros sencillos (7 bits/carácter) utilizando NVT en una sesión telnet
- **ESMTP** (Extended SMTP):
  - Transferencia de datos más complejos (8 bits per carácter)
- **MIME** (Multipurpose Internet Mail Extensions):
  - Extensiones multimedia para los *attachments*
- **POP3** (Post Office Protocolo):
  - Acceso al buzón des de un ordenador personal (no un host)
- **IMAP4** (Internet Access Protocolo):
  - Evolución del POP

6.4. Correo electrónico

Modelo Store & Forward

Ventajas del modelo Store and Forward

- Utilización del correo electrónico con seguridad
- Ahorro enviando e-mail en bloque a horas determinadas
- Traducción de formatos

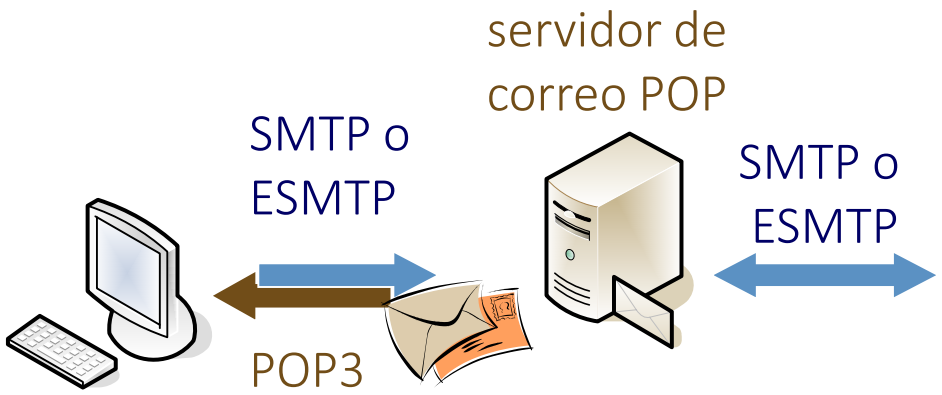


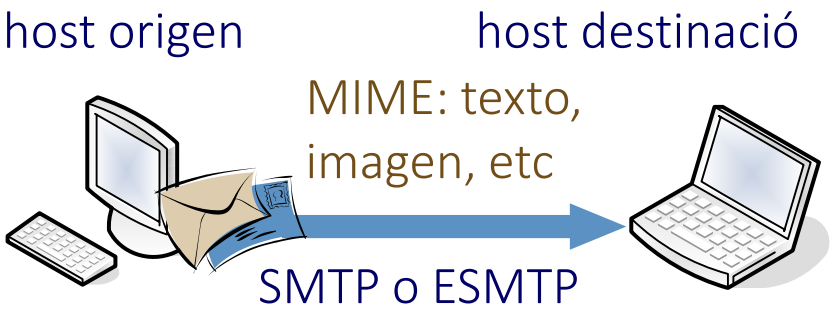
6.4. Correo electrónico

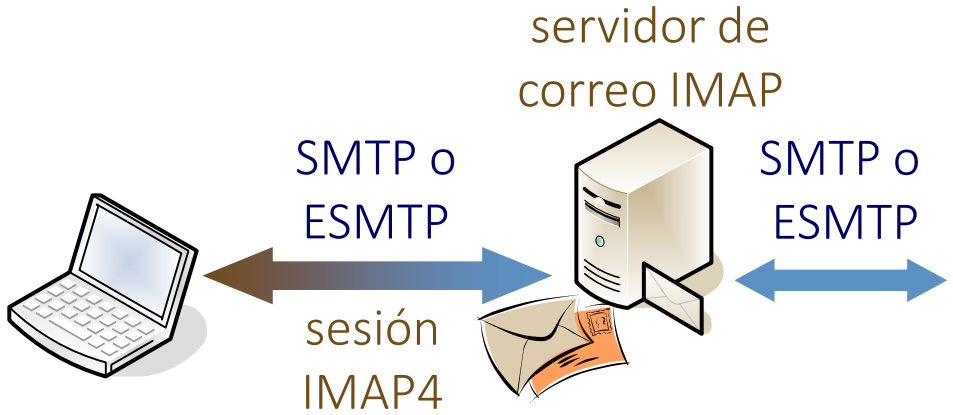
Transferencias de correos

Transferència host-host

Transferència host-terminal







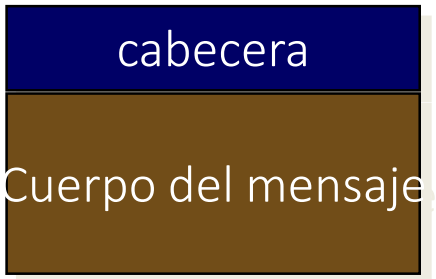
6.4. Correo electrónico

Main relay – servidores de correo en 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. Correo electrónico

Partes de un correo electronico



Cabecera

Received	keywords
Date *	Subject
From *	Comment
To *	Encrypted
cc	References (anteriores identificadores de mensaje)
bcc (blind cc)	In-Reply-To
Message-Id	Sender (si no es el creador del mensaje)
Reply-To	

Linea en blanco

Cuerpo del mensaje

6.4. Correo electrónico

Ejemplo de una cabecera de correo

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. Correo electrónico

SMTP: Simple Mail Transfer Protocol

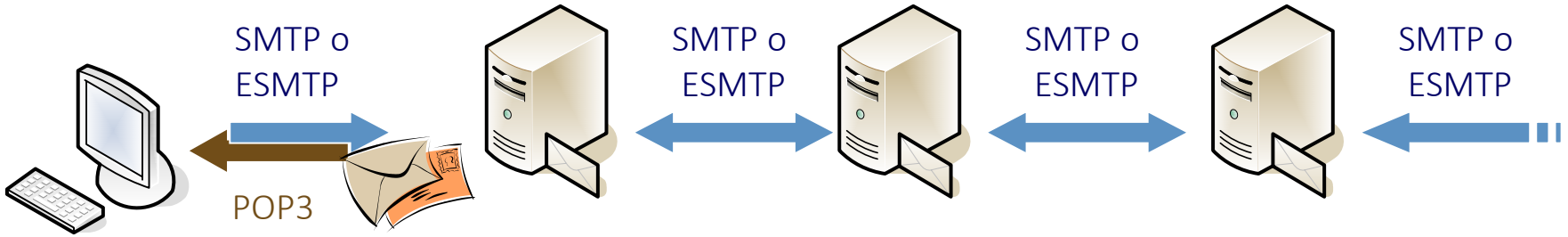
- RFC 821
- Define un mecanismo de transferencia de correo entre 2 máquinas
  - máquina origen
    - puede ser un cliente o bien otro host (relay de correo)
  - máquina destinación
    - Debe ser un servidor; no puede ser un cliente
  - para cada transferencia, se establece una sesión
    - Una sesión puede constar de diversos correos
    - Protocolo TCP, puerto 25

6.4. Correo electrónico

Registro de tiempo, identificador y paso en SMTP

Los correos tienen un registro de hosts intermedios y *timestamps* (tiempo de paso)

- Cada vez que un mensaje pasa por un MTA se añade una marca temporal de su paso, ***timestamp***
- El correo contiene todo el camino recorrido
- El identificador del mensaje lo pone el primer MTA que atraviesa el mensaje y hace **único** el mensaje



6.4. Correo electrónico

Registro de tiempo, identificador y paso 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. Correo electrónico

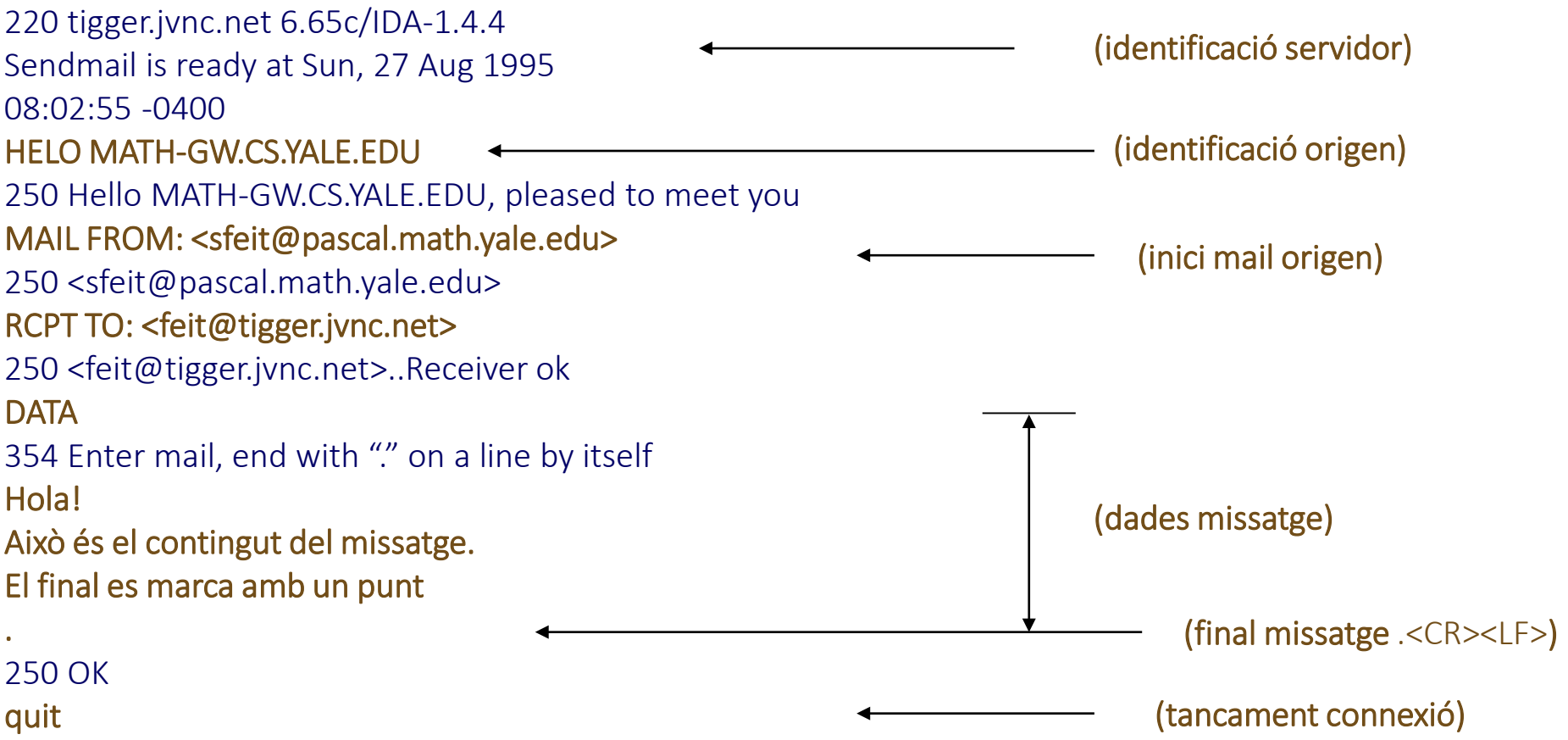
Ordenes y secuencia del Protocolo SMTP

Ordenes del protocolo

- HELO, MAIL, RCPT, DATA, QUIT, TURN
- Secuencia del protocolo
  - Origen envía un paquete inicial, receptor envía identificación de host
  - Origen envía identificación de host (HELO)
    - clientes y servidores ESMTP se reconocen por EHLO en vez de HELO
  - Origen envía ID del usuario que genera el mensaje (MAIL FROM:)
  - Origen identifica los receptores (RCPT TO:)
  - Origen transmite los datos de cabecera y texto (MAIL, DATA)
  - Origen transmite ".<CR><LF>" (final de correo)
  - Se repite el proceso para otros mensajes o se finaliza (QUIT)
  - Posibilidad de intercambiar papeles (TURN)

6.4. Correo electrónico

Ejemplo de una sesión SMTP



6.4. Correo electrónico

Extensiones del protocolo SMTP

SMTP y RFC 822 se diseñaron para mensajes textuales

- Como añadimos otros contenidos (vídeos, imágenes) ?
- Como transportar caracteres por encima de ASCII 127 ?
- solución: Modificar protocolo transporte y formato de los mensajes
  - ESMTP, Extended SMTP, RFC 1425 / 1869
    - Clientes y servidores ESMTP se reconocen por EHLO en vez de HELO
  - MIME, *Multipurpose Internet Mail Extension,* RFC 1521
    - Definición de extensiones que soportan formatos multimedia
    - pueden transferirse eficientemente con ESMTP (y no tanto con SMTP)
    - Se delimita cada parte del mensaje con una marca
    - Facilita la llamada a aplicaciones de reproducción

6.4. Correo electrónico

MIME: Multipurpose Internet Mail Extension

- Tipo de datos soportados:
  - texto
    - plain, richtext (formato básico), enriched
  - multipart
    - Mixed (diversos formatos procesados secuencialmente), parallel, digest...
  - application
    - octet-stream (arbitrario), postscript, ...
  - Imagen
    - BMP, JPEG, PIF, GIF...
  - video
    - mpeg, quicktime, avi, ...
  - audio
    - basic, mpeg,

6.4. Correo electrónico

Ejemplo 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. Correo electrónico

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'autentificació
  - 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'autentificació criptogràfica d'IMAP4

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

6.4. Correo electrónico

Post Office Protocol – POP3

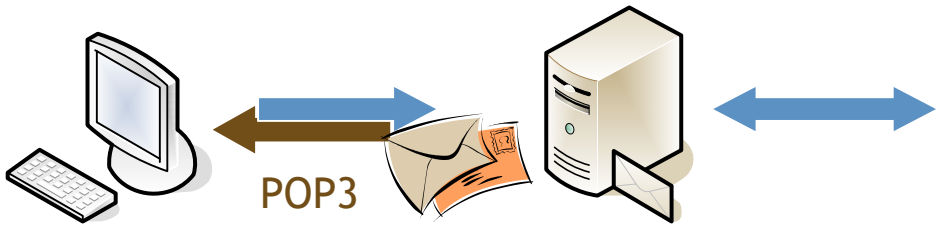
Ordres bàsiques:

- Autentificació

  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. Correo electrónico

Ejemplo de dialogo 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:	(es tanca el final del servidor de la connexió TCP)
C:	(es tanca el final del client de la connexió TCP)

6.4. Correo electrónico

Acceso de un buzón des de múltiples clientes con POP3

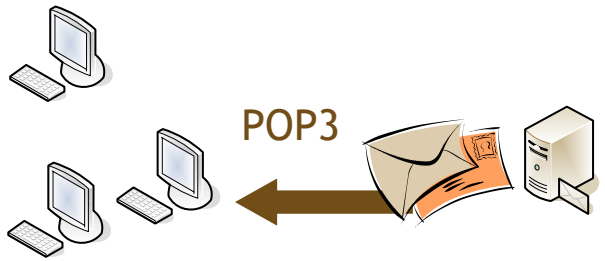
Alguns clients permeten deixar el correu al servidor

- 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. Correo electrónico

Ejemplo de dialogo POP3 con APOP y UIDL

```
C:                                                  (Inicia connexió TCP pel port 110)
S:                                                  (accepta connexió TCP)
S: +OK POP3 server ready <1234.697170952@guys.com>
C: APOP asmith c9dcb935ce1d21d02afdebcbd9cb1d5a
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. Correo electrónico

IMAP: Internet Mail Access Protocol

Problemes del POP

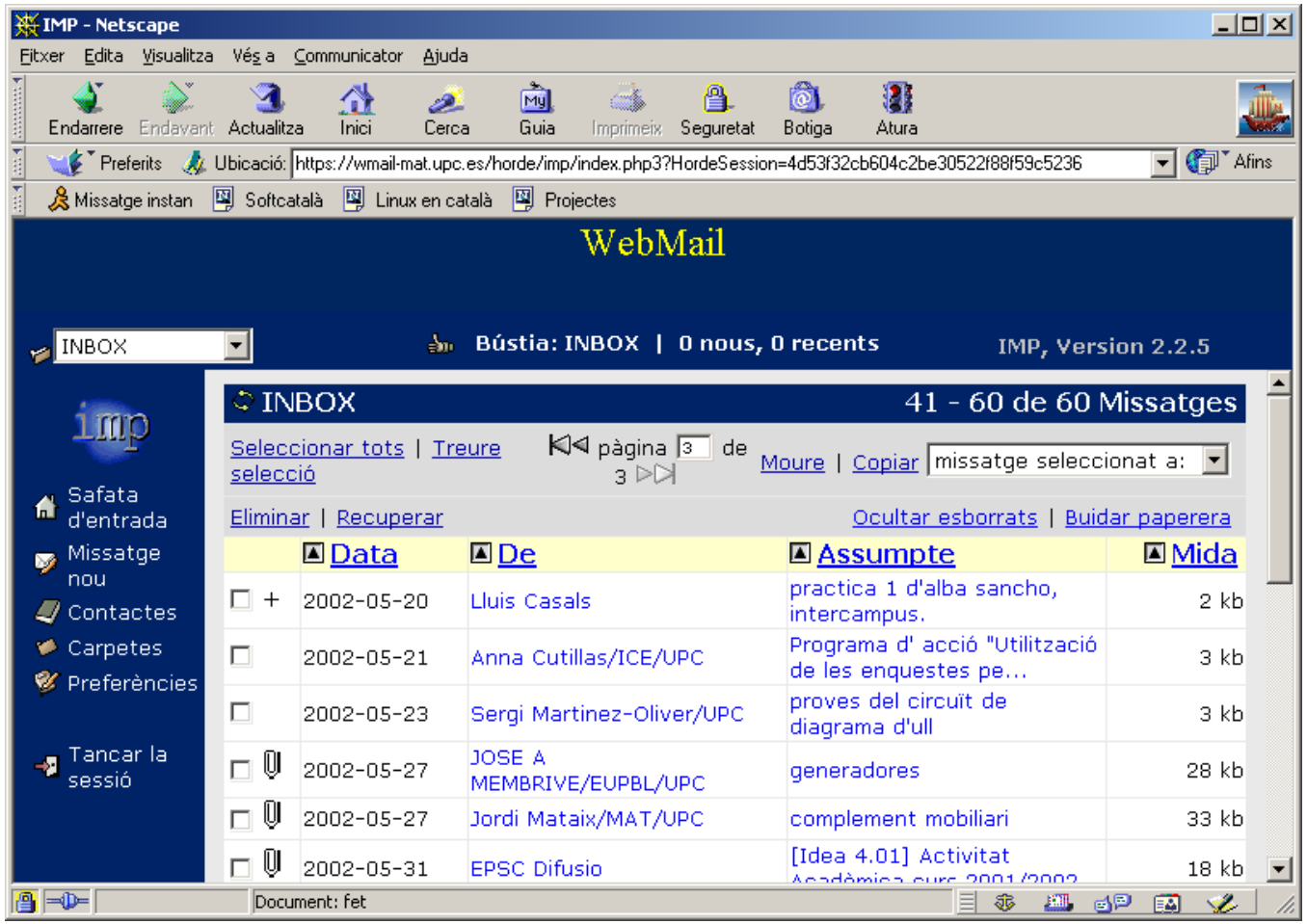
- Els correus es “baixen” al client (i s’esborren del servidor)
- Si s’accedeix a la mateixa bústia des de diferents terminals, els missatges es reparteixen
- És difícil d’integrar amb noves interfícies com WWW

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. Correo electrónico

IMAP permite webmail



6.4. Correo electrónico

IMAP VS POP

POP		IMAP	
Un sol servidor, un sol inbox		Es pot treballar amb diferents bústies i servidors	
Un sol terminal: còpia local dels missatges		Missatges queden al servidor (accés desde diferents terminals)	
Connexió mínima: baixar-se els missatges i pujar els nous		Connexió contínua, dependència de la connexió	

6.5. Noticias

News

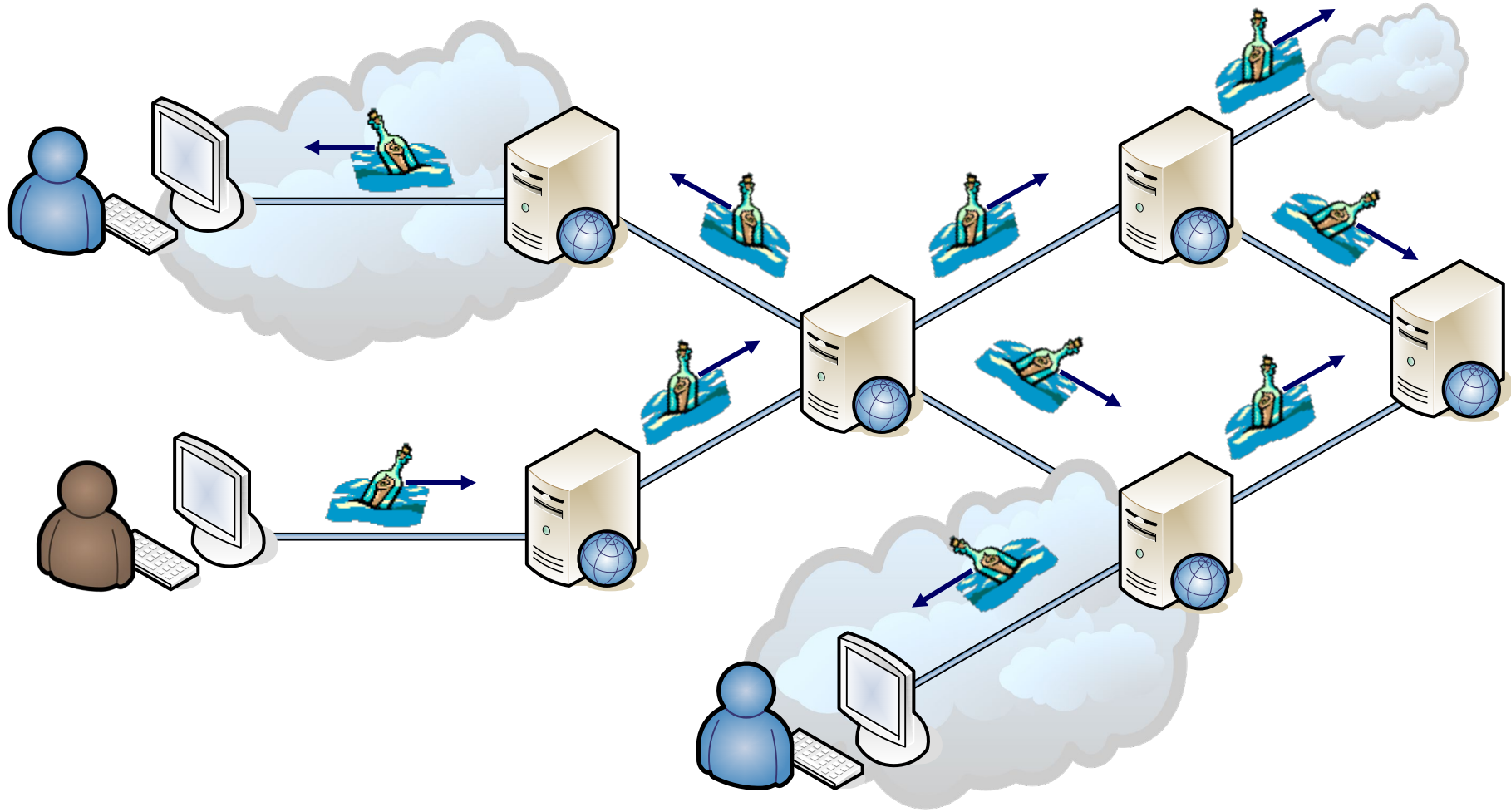
Fòrums de discussió monogràfics accessibles per e-mail

- 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...

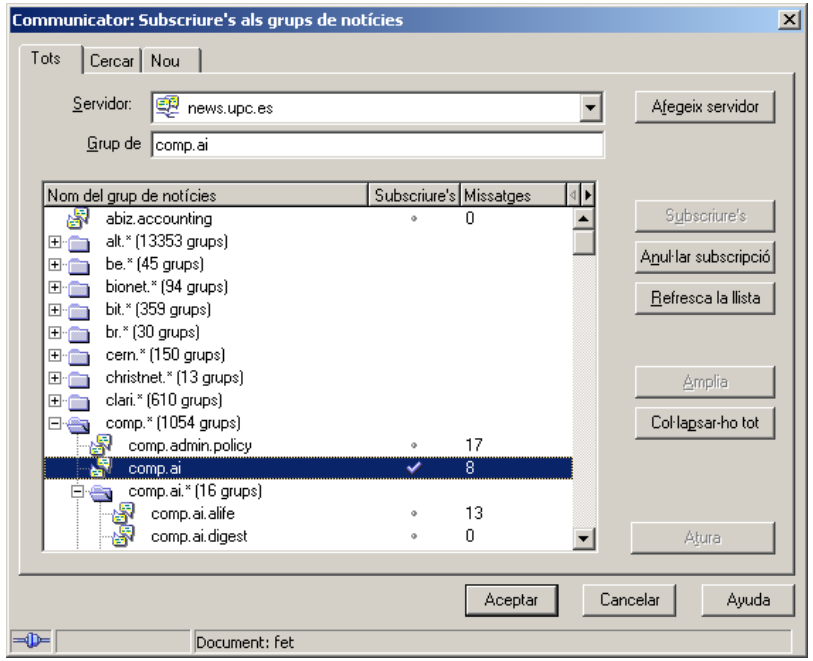
6.5. Noticias

Esquema de funcionamiento de las News



6.5. Noticias

Ejemplo de cliente de News - Netscape



Pràcticament no s'utilitza

RSS l'ha substituit (evolució XML i www)

6.6. Conclusiones parciales

Conceptos básicos

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 arquitectures 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

6.7. Tecnologias web

6.7.1. Tecnologías 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*)

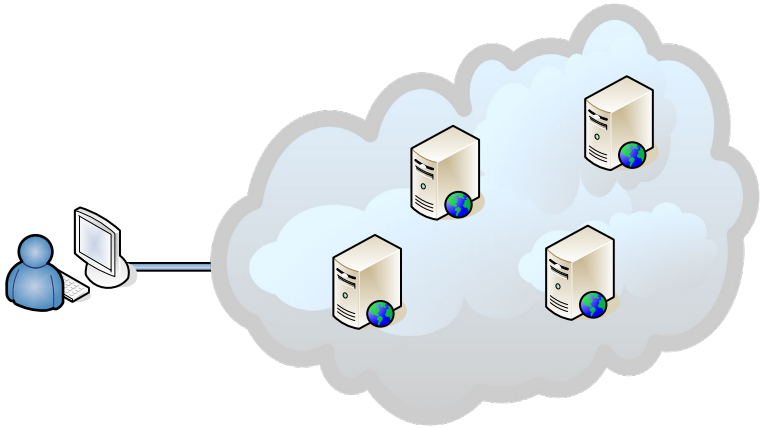
Domini	Activitat	Àrees compreses
Interfície d'usuari (interacció home-màquina)	Millorar la percepció de l'usuari de la informació	HTML, gràfics 3D, internacionalització, fulles d'estils, accés mòbil ...
Tecnologia i societat (interacció home-home)	Permetre aplicacions orientades a la societat.	Iniciativa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

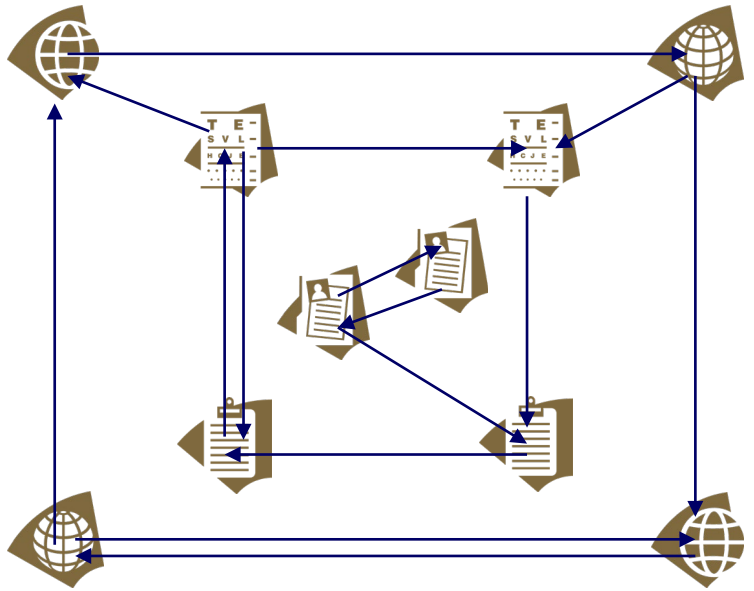
6.7.1. Tecnologías Web

Filosofía de trabajo

Teranyina formada per documents enllaçats sense necessitar saber on són.

- Només necessitem saber el seu identificador
- Enllaços: hipertext





6.7.1. Tecnologies Web

Families de protocolos

Protocols del servei WWW

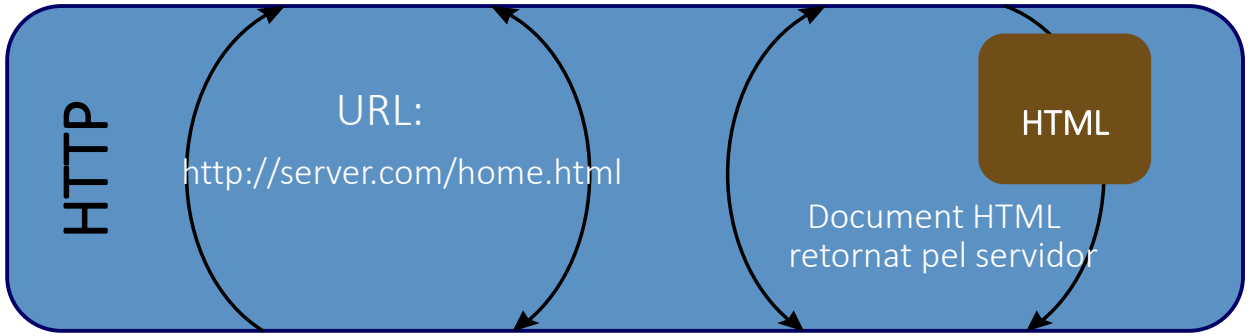
- **HTML:** *HyperText Markup Language*
  - Format representació de la informació
- **HTTP:** *HyperText Transfer Protocol*
  - Protocol transport de la informació
- **URL:** *Uniform Resource Locator*
  - Identificació de la informació

6.7.1. Tecnologías Web

Funcionamiento básico

URL identifica el protocol, el servidor i el document

HTTP és el protocol que permet comunicar clients i servidors a través de la xarxa



L'enllaç hipertexte conté la URL del document enllaçat

HOMEPAGE

Pots visitar les següents pàgines:

- Tràfic
- Temps
- Esports
- Culrura

ESPORTS

* Fútbol
* La resta...

Aquesta pàgina ha estat actualitzada el ...

El *browser* Interpreta l'HTML i presenta el document a la pantalla

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 es un lenguaje 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

Ejemplo de código 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>
```

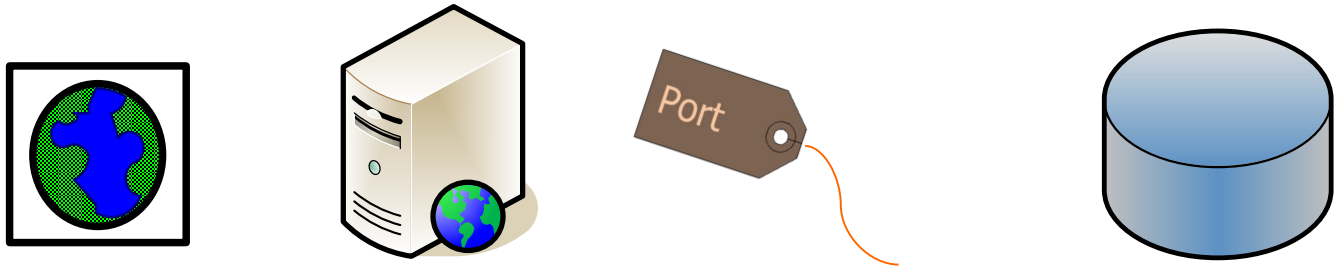
6.7.3. URL

Estructura genérica de las 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

Acceso a diferentes recursos mediante las URL


HTTP: http://www.catacrac.es

FTP: ftp://soc.alegria.com/


News (NNTP): 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

6.7.4. HTTP

HyperText Transfer Protocol (HTTP)

HTTP 1.1 estàndard, RFC 2616

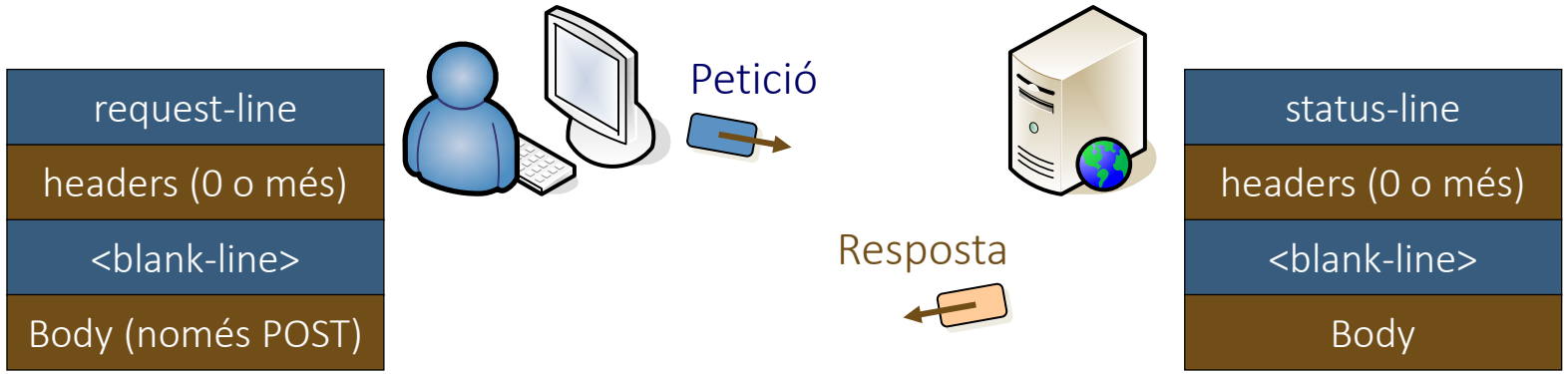
- S'ha definit un sistema d'autenticació, RFC 2617
- Utilitza TCP, típicament el port 80
- Enviament de peticions i respostes en format text
- Diàleg sessió molt senzill

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

Formato de las peticiones y respuestas HTTP



On tenim que

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

6.7.4. HTTP

Descripción de las ordenes 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

Descripción de las ordenes 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

Códigos numéricos de respuesta (3 dígitos response code)

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

6.7.4. HTTP

Cabeceras 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

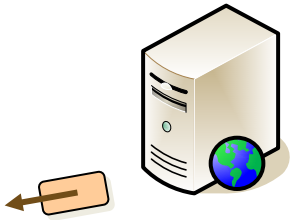
Ejemplo de transacción HTTP

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)



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

Deficiencias HTTP 1.0 solucionadas por 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

Otras mejoras aportadas por HTTP 1.1

Negociación del contenido

- Un servidor puede tener diferentes representaciones de un recurso ⇒ enviar al cliente la mas adecuada

Autenticación sin enviar el password en claro

- Utiliza técnica de secreto compartido (MD5)

Descarga de un determinado rango de bytes

- Se puede pedí el envío de una parte de un recurso

6.7.4. HTTP

Evolución futura del protocolo HTTP

HTTP 1.1 estandarizado en Junio de 1999.

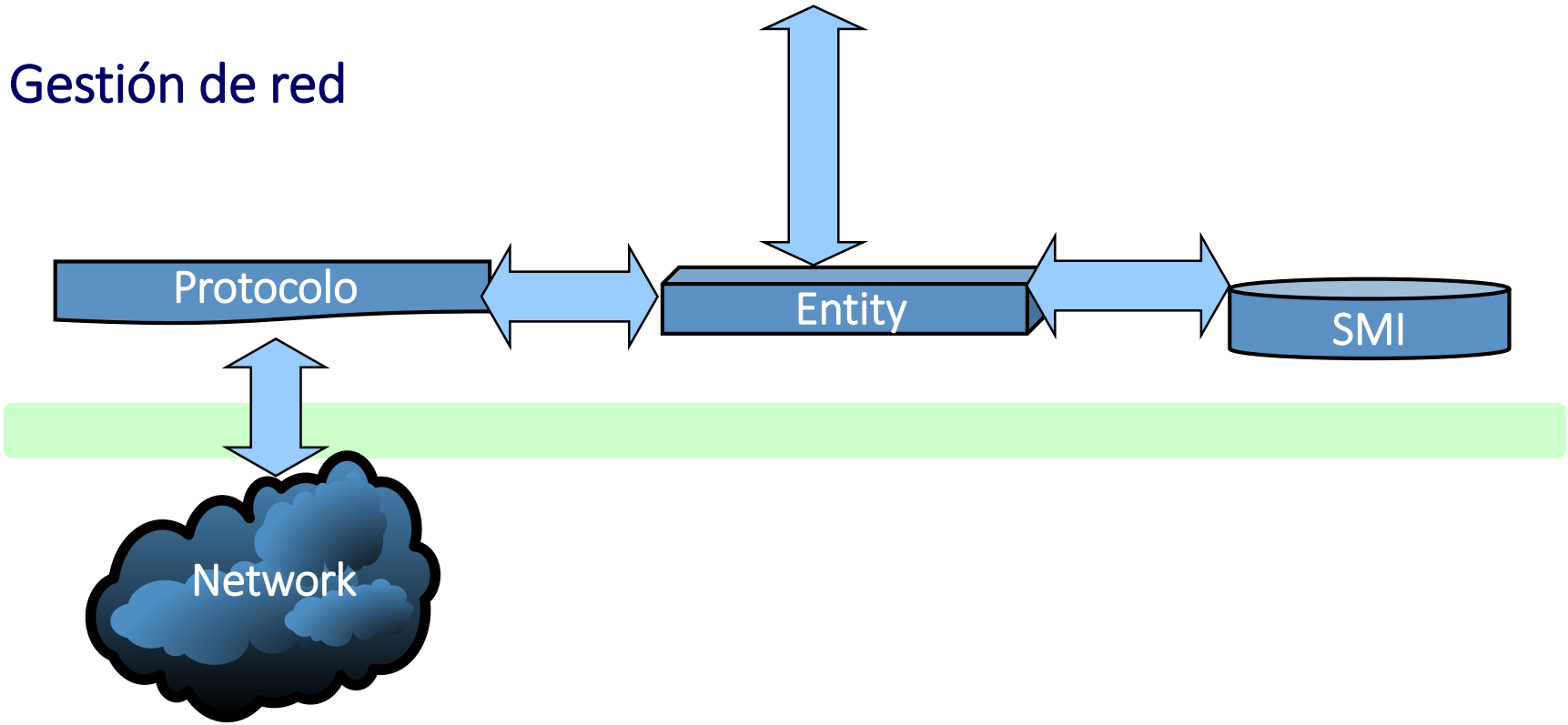
- Ya es ampliamente soportado

HTTP-NG: hacer un nuevo HTTP mas simple, modular, con estructura de capas

- HTTP 1.1 ms bien complicado y voluminoso
- MUX: multiplexar información de diferentes peticiones en una conexión TCP
- Presentación simultanea

6.8.1. SNMP

SNMP proporciona un entorno de gestión de redes

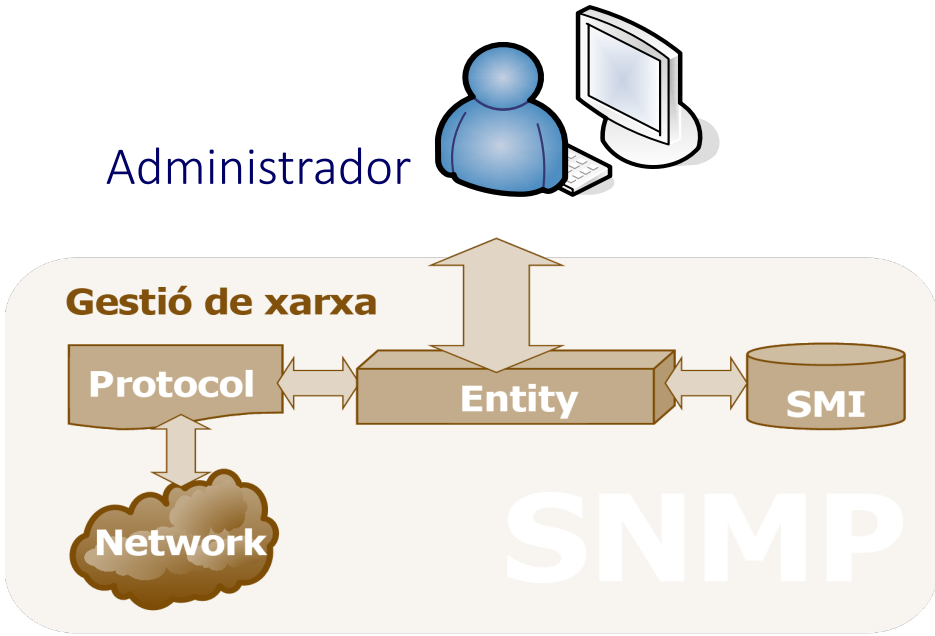


6.8.1. SNMP

Filosofía de trabajo

Debe poderse aplicar...

- A la mayor escala posible
- Con la mayor diversidad de implementación posible
- Con la mayor amplitud de capas de protocolos posible
- Con la mayor diversidad de administración que se pueda obtener



6.8.1. SNMP

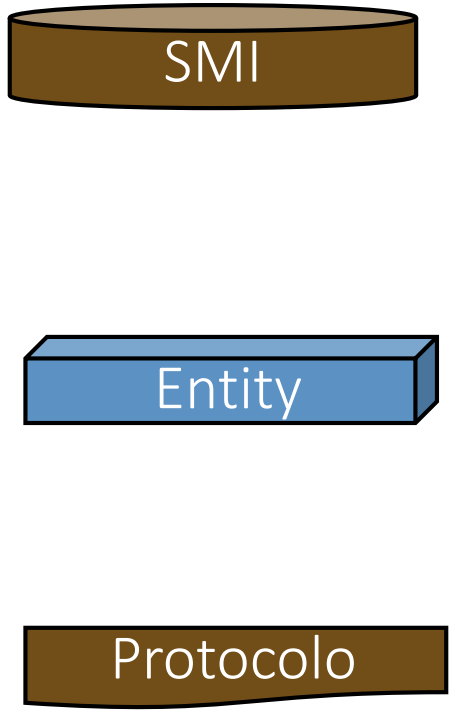
Tareas a realizar

- **Gestión de Configuración**: Configuración de los parámetros de la red gestionada. gestión centralizada.
- **Gestión de fallos**: Detección y, si es posible, corrección de fallos en la red.
- **Gestión de estadísticas**: Estadísticas sobre la utilización de la red. permiten definir políticas de acceso, tarificación, planificación, etc.
- **Gestión de la eficiencia**: Análisis de las estadísticas de utilización. permite descubrir cuellos de botella.
- **Gestión de la seguridad**: Controlar el acceso a los dispositivos gestionados, generar alarmas cuando se detecten intrusos.

6.8.1. SNMP

Modelo de arquitectura SNMP

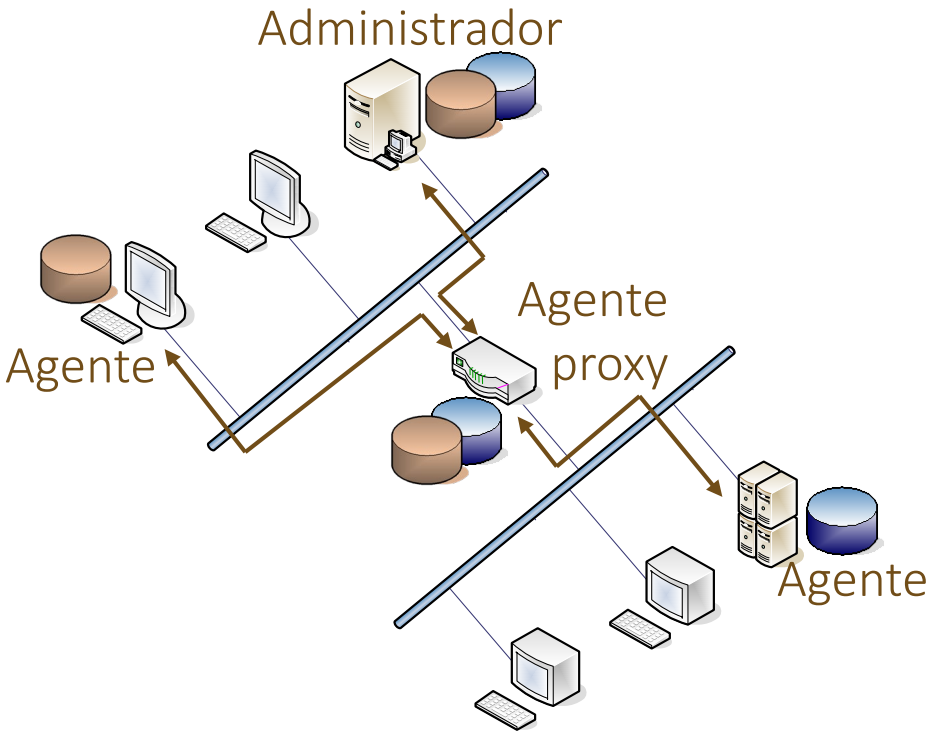
- Base de datos con información sobre:
  - ✓ Configuración
  - ✓ Estado
  - ✓ Errores
  - ✓ Rendimiento
- Entidades administradoras y administradas:
  - ✓ agentes
  - ✓ agentes proxy (SNMP v:2.0)
  - ✓ Administradores
  - ✓ Bases de información de administración (MIB)
- Protocolo de comunicación
  - ✓ Acceso a la capa de transporte
  - ✓ Mensajes



6.8.1. SNMP

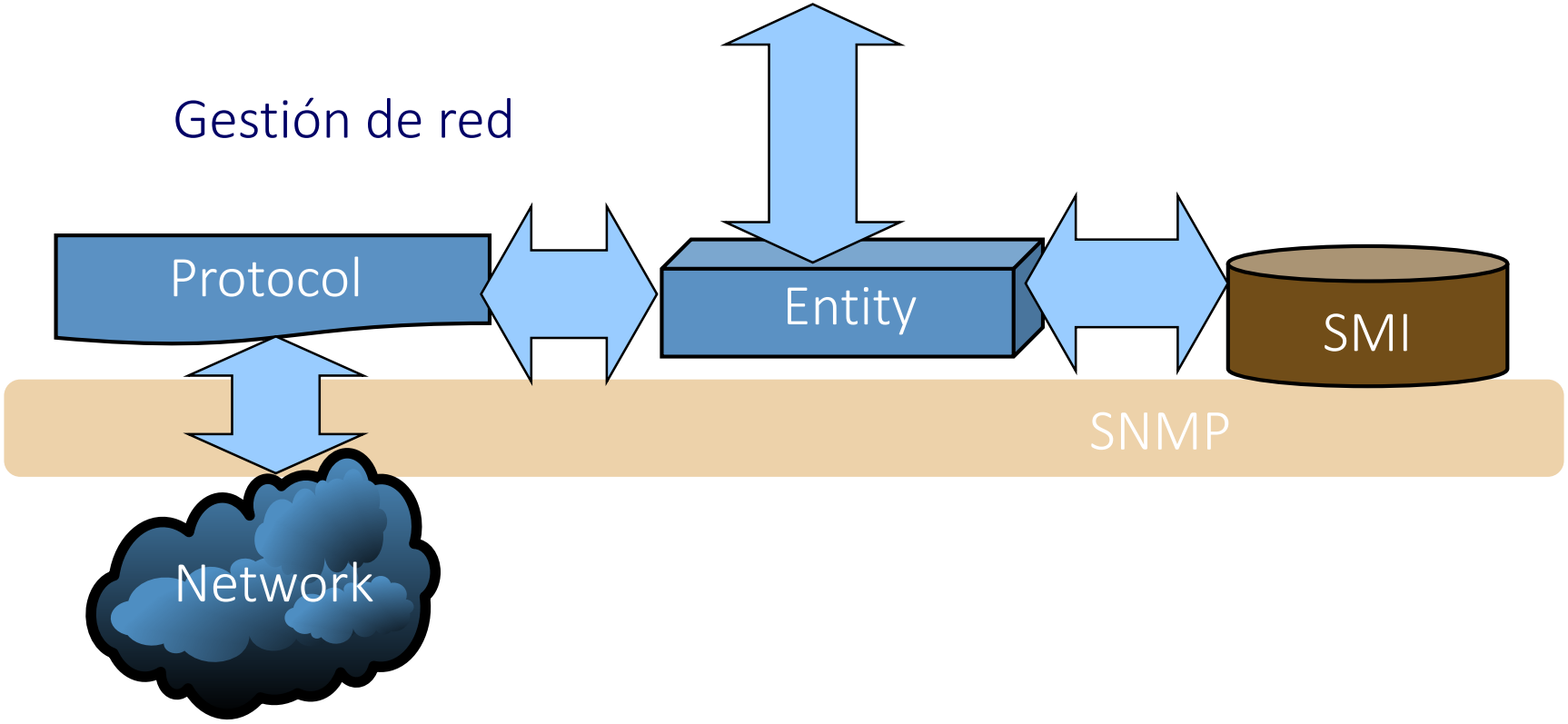
Arquitectura SNMP

- El administrador ejecuta aplicaciones, genera comandos SNMP, interroga a los agentes y presenta los datos al usuario mediante una interfaz gráfica
- Los datos se almacenan en las MIB.
- El protocolo SNMP viaja sobre UDP (puertos 161 y 162).



6.8.2. SMI

SMI (Structure of Management Information) proporciona el marco general de Trabajo de las MIB



6.8.2. SMI

Las características de la SMI vienen recogidas en múltiples RFCs

RFC 1155: Structure and Identification of Management Informationfor TCP/IP-base Interfaces

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

RFC 1643: Definition of Managed Objects for the Ethernet-likeInterface Types

RFC 2021: Remote Network Monitoring Management Information Base 2

.....

6.8.2. SMI

Puntos clave de la estructura de la información de gestión

Establece la metodología para crear la estructura de las MIB, el árbol SMI

Define como se crean los objetos de las MIB, tanto la sintaxis como el valor, ASN.1

Define la metodología para codificar los valores de los objetos de las MIB, BER

6.8.2. SMI

Las MIBs contienen los objetos administrables

- Contienen la descripción lógica de los datos de administración de la red (RFC 1155).
- Los conjuntos de variables relacionadas se agrupan en **Módulos MIB** (recogidos en RFCs).
- La descripción de una variable especifica:

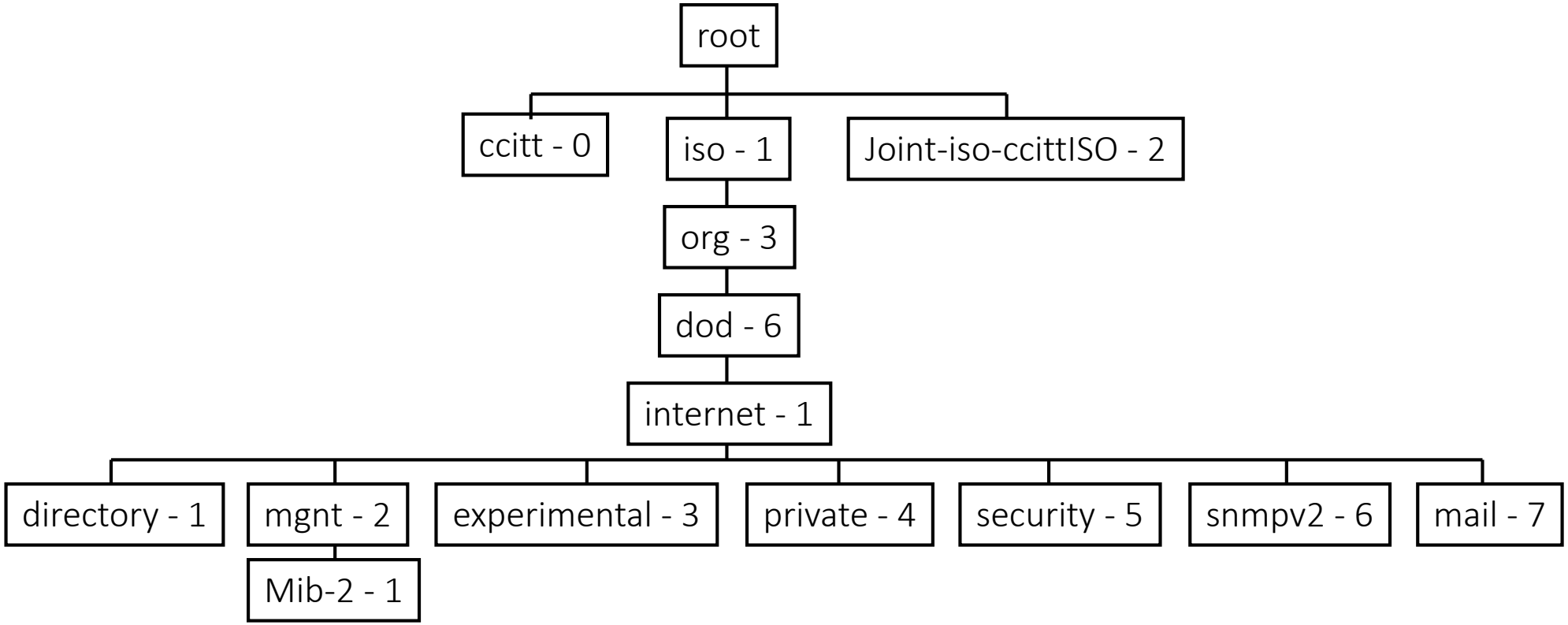
  Una definición de qué es la variable.

  Una descripción de como se mide su valor.

  Un nombre para cuando se lea o actualice el valor de la variable a la base de datos.

6.8.3. Arbol SMI

Las variables de la MIB se ordenan siguiendo una estructura en árbol

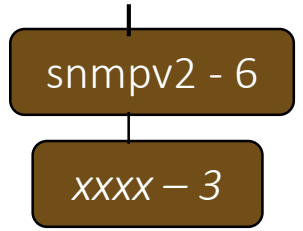


6.8.3. Arbol SMI

Nombres identificadores de los objetos

- Al final se incluye el número del sistema administrado
- Toda variable se puede identificar de do maneras:
- Identificador de objeto: e empieza des de la raíz uniendo los números de los nodos
- Nombre del objeto: se empieza des de la raíz uniendo los nombres de los nodos

Ejemplo:



Identificador d'objecte: *1.3.6.1.6.3*

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

6.8.3. Arbol SMI

Ordenación en les tablas MIB

- Se empieza por la izquierda
- Se compara hasta que se encuentra el primer valor diferente
- El elemento con el número mayor en esta posición es el elemento más grande
- De otro modo, el identificador mayor es el de valor más grande

Ejemplo:

.....

1.3.6.1.6.2

1.3.6.1.6.3

1.3.6.1.6.4

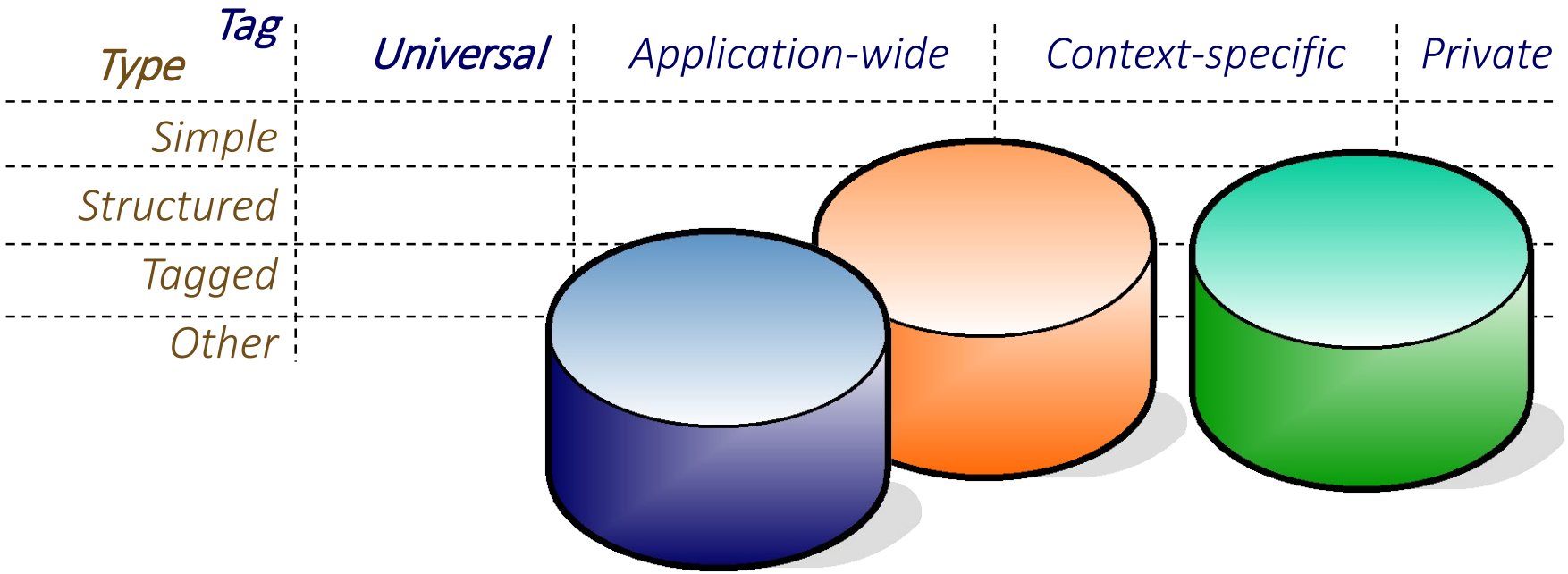
1.3.6.1.6.4.1

.....

6.8.4. Codificación ASN.1

Los datos se clasifican con tipo y etiquetas

Se agrupan por su naturaleza (type, ej.: integer) y por su utilización (tag, ex: router UPC)



6.8.4. Codificación ASN.1

La clase de datos UNIVERSAL forma la base del resto de variables

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

Tipo de objetos
6 - OBJECT IDENTIFIER
7 - Objeto descriptor

Reservados
19-5 , 28-...

Tipo de cadenas de caracteres
18 - NumericString
19 - PrintableString
27 - GeneralString, ...

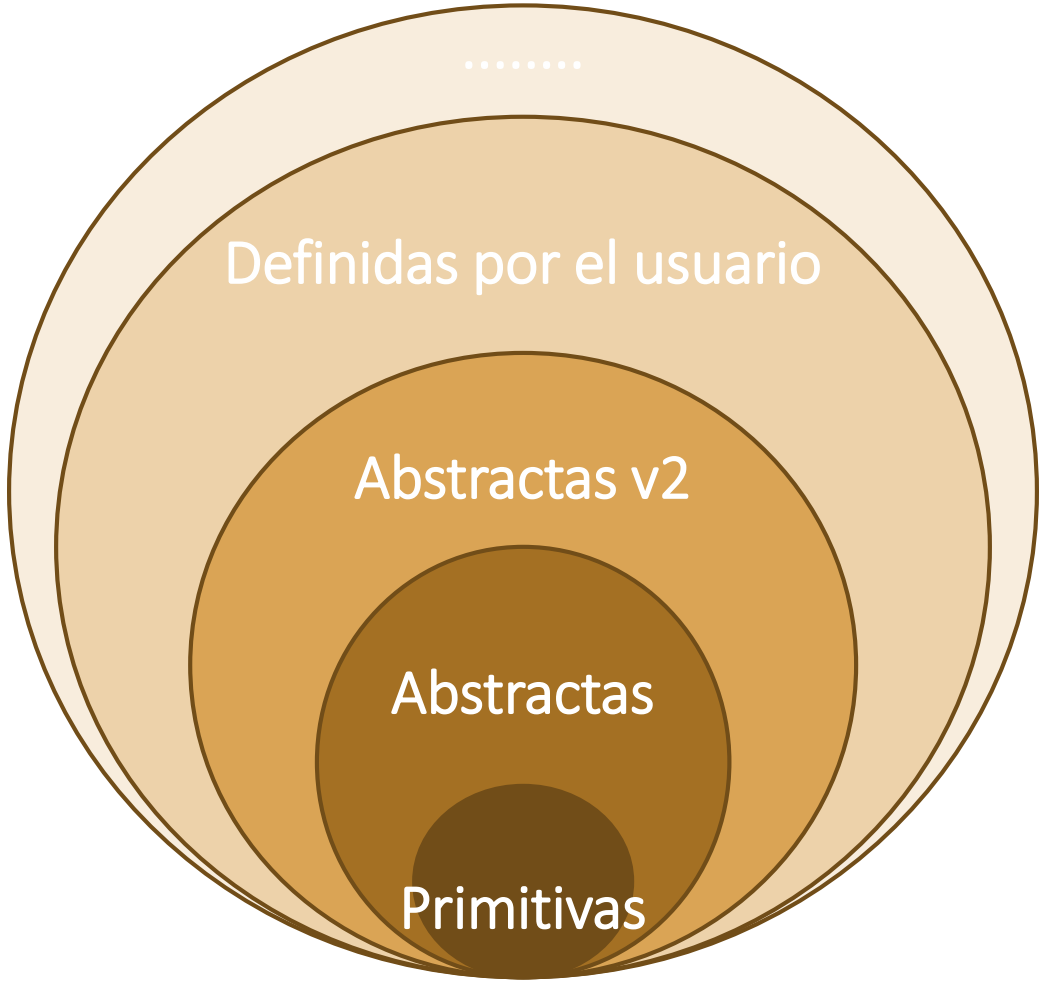
Tipos estructurados
16 – SEQUENCE, SEQUENCE OF
17 – SET, SET OF

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

6.8.4. Codificación ASN.1

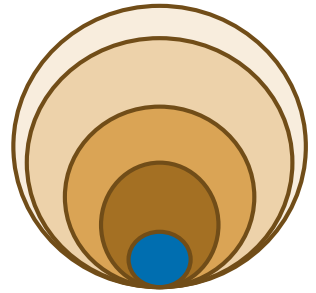
Encapsulamiento de datos

Los tipos de datos se van formando a partir de las combinaciones de los datos ya definidos en las MIBs



6.8.4. Codificación ASN.1

Primitive data

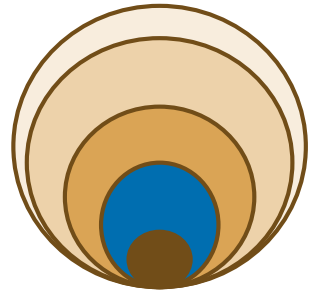


Son los tipos básicos de datos

- *Integer*: 32 bits en complemento a dos. Se utiliza para enumeraciones
- *Octet string*: cadena de 0 o mas octetos. Se utiliza para representar textos.
- *Object identifier*: secuencia de integers. Se utiliza para identificar los objetos en las MIBs.
- *Null*: en blanco
- *Sequence, sequence-of*: se utiliza para construir tablas

6.8.4. Codificación ASN.1

Abstract data (SNMP v1)

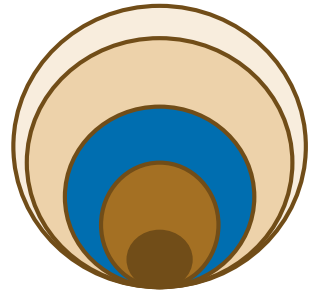


Añaden el primer grado de abstracción

- ***NetworkAdres***: dirección de red, depende del protocolo
- ***IpAddresCounter***: 4-byte OCTET STRING
- ***Counter***: 32 bits sin signo con overflow
- ***Gauge***: 32 bits sin signo con saturación
- ***TimeTicks***: contador de tiempo de 32bits (centésimas de segundo). Como máximo 497 días
- ***Opaque***: cualquier dato en formato oCTET STRING

6.8.4. Codificación ASN.1

Abstract data (SNMP v2)



Son un refinamiento de las anteriores

- *Integer32:* igual que INTEGER
- *Counter32*: igual que Counter
- *Counter64*: igual que Counter pero con 64 bits
- *Gauge32*: igual que Gauge
- *Unsigned32*: a la práctica igual que Gauge

6.8.4. Codificación ASN.1

Textual conventions

Las textual conventions amplían las descripciones de los tipos de objetos administrados contenidos en les MIBs

- *Definir nuevos tipos*
- *Representar los tipos existentes*
- *Representar los valores de los tipos*
- *Codificar los valores de los tipos existentes*

6.8.5. Reglas BER

Las Basic Encoding Rules ,BER, especifican como codificar cualquier valor definido con ASN.1

Un valor se puede codificar de diversas maneras

Utilizan el tipo OCTET STRING

Siguen la estructura: tipo – longitud – valor

Existen tres métodos

- Primitiva, definite-length encoding
- Construidas, definite-length encoding
- Construidas, indefinite-length encoding

6.8.5. Reglas BER

Formato de la codificación BER

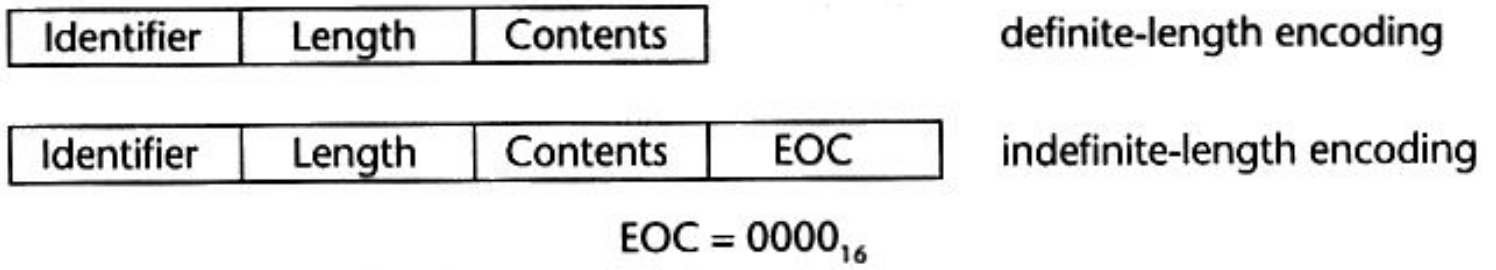
Los objetos e codifican con cuatro campos

Identificador: referencia que objeto administrado se refiere

Longitud: indica el número de bytes que utiliza

Contenido: valor del objeto

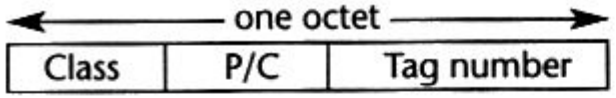
Final de contenido: marca el final del valor en los casos de longitud variable



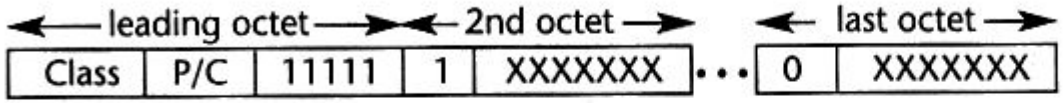
6.8.5. Reglas BER

Formato del campo identificador de las BER

Tags <31



Tags ≥31



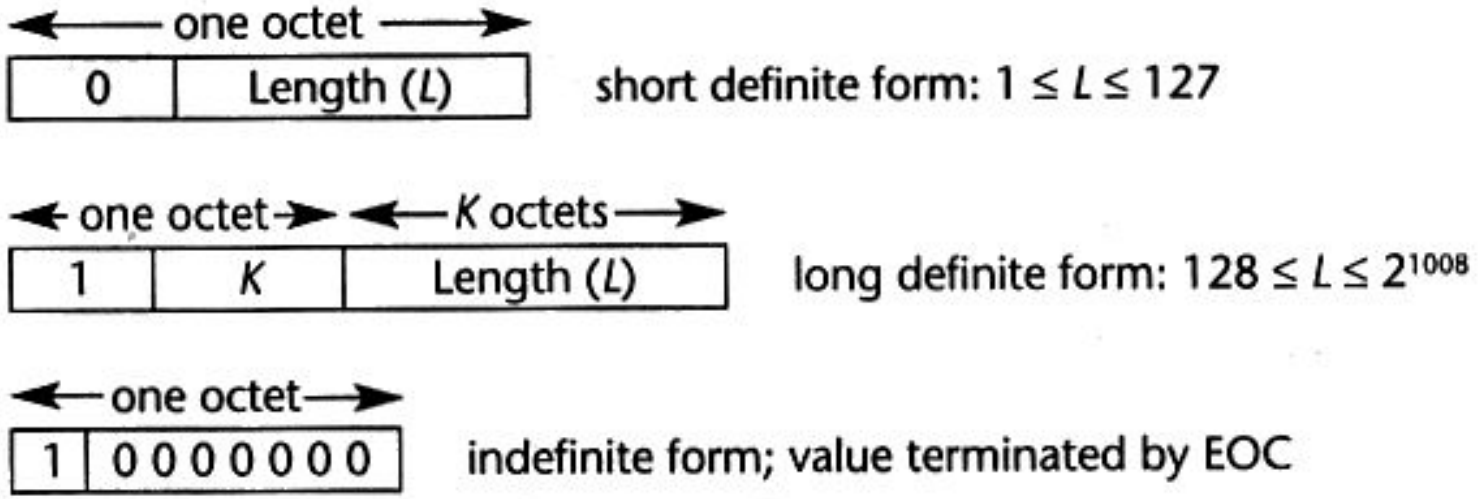
Class:
00 = Universal
01 = Application
10 = Context specific
11 = Private

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. Reglas BER

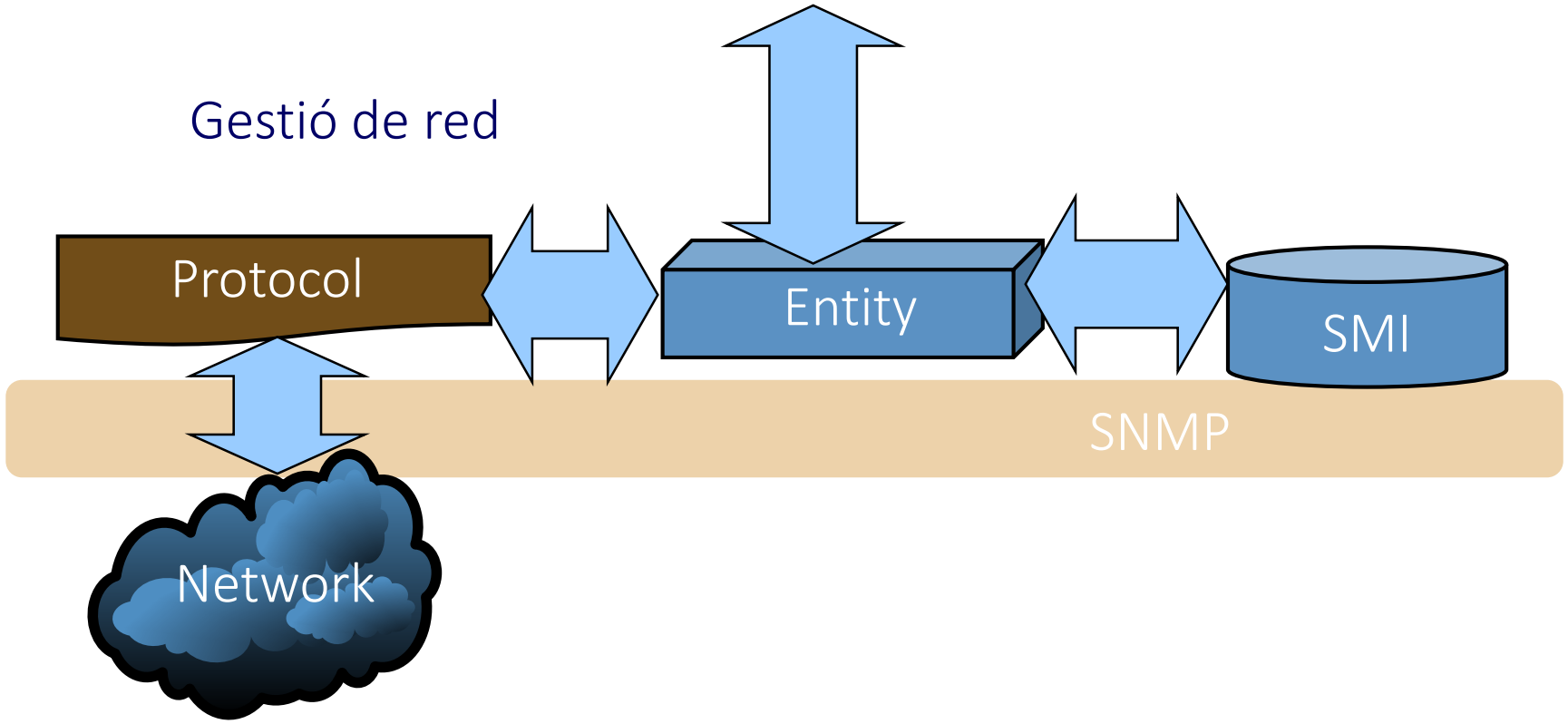
Formato del campo longitud de las BER



El campo EOC es formado por dos bytes a cero

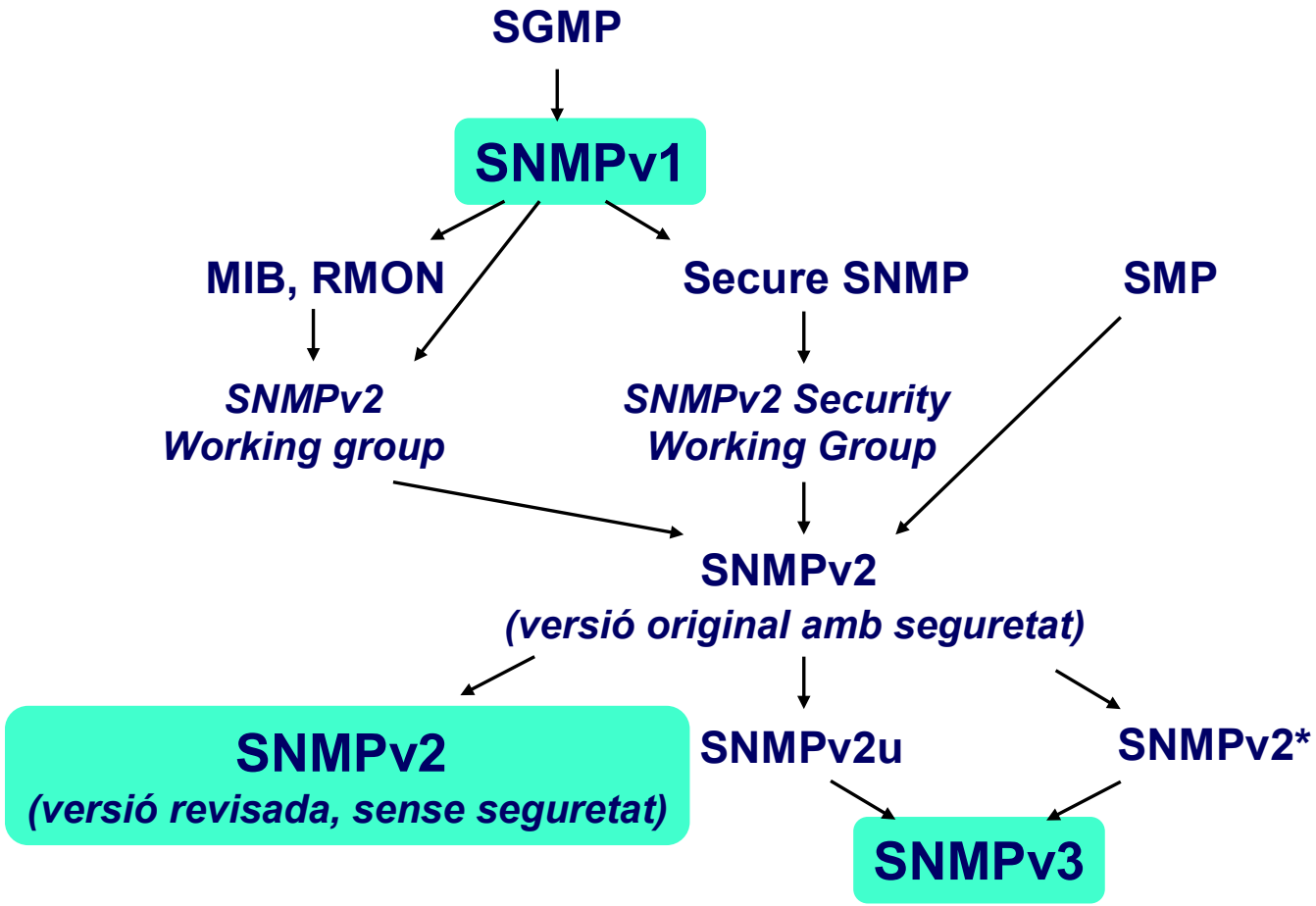
6.8.6. Protocolo SNMP

Mediante el protocolo de gestión de red las entidades se comunican por la red



6.8.6. Protocolo SNMP

Evolución del protocolo SNMP



6.8.7. Protocolo SNMP v1

La versión 1 de SNMP ve recogida en 5 RFCs

RFC 1155: Structure and Identification of Management Informationfor TCP/IP-base Interfaces

RFC 1157: A Simple Network Management Protocol (SNMP)

RFC 1212: Concise MIB definitions

RFC 1213: Management Information Base for Netware Management ofTCP/IP-base Interfaces: MIB-II

RFC 1643: Definition of Managed Objects for the Ethernet-likeInterface Types

6.8.7. Protocolo SNMP v1

SNMP necesita servicios a nivel de transporte para enviar los mensajes

Protocolos de transporte soportados

UDP: User Datagram protocol de la arquitectura TCP/IP

CLTS: ConnectionLes transport Services de la arquitectura OSI

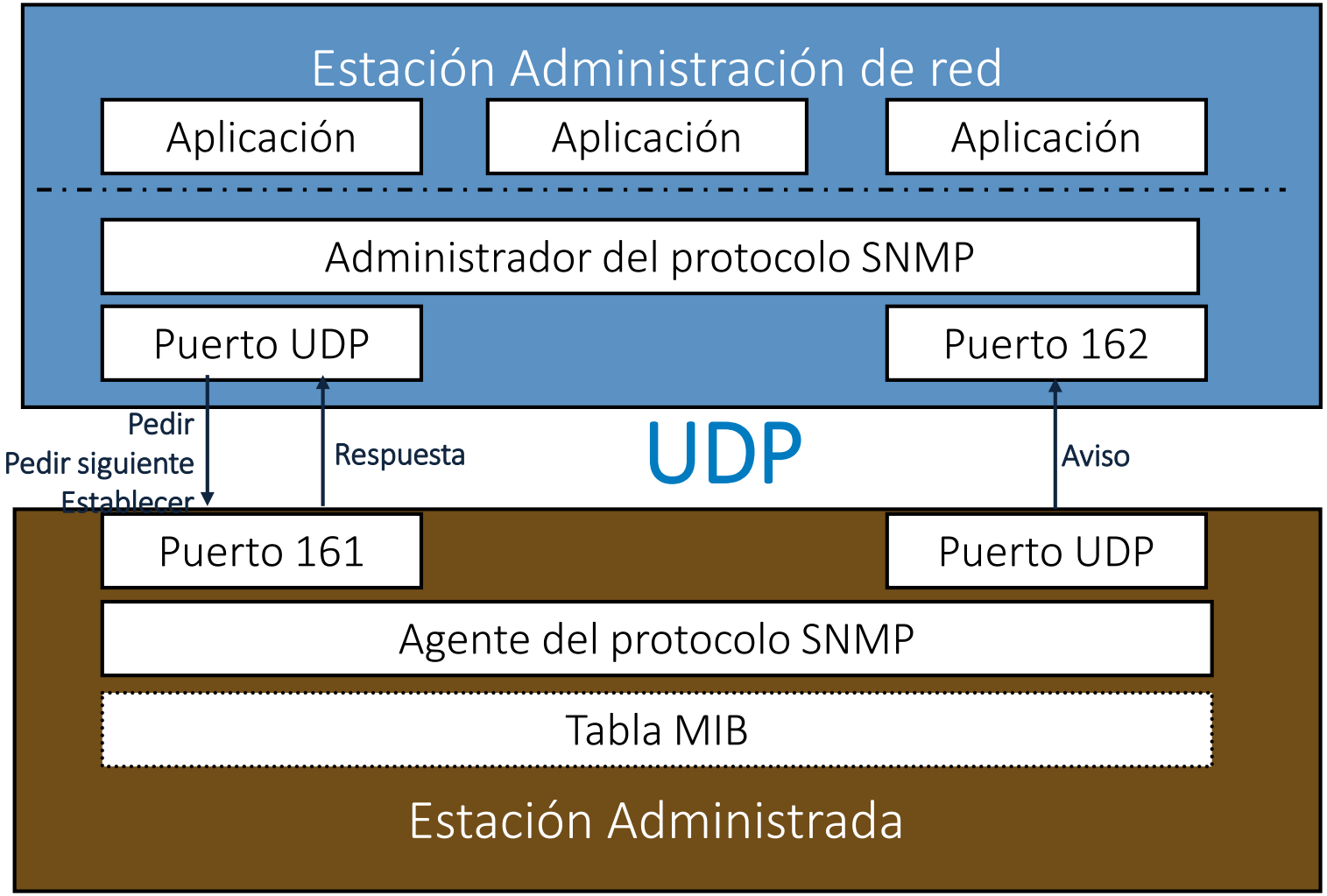
Está pensado para correr sobre protocolos de transporte orientados a conexión

El administrador puede mantener la conexión abierta para anticiparse al agente.

El agente puede cierra una conexión si le hacen falta recursos

6.8.7. Protocolo SNMP v1

Acceso a la capa de transporte



6.8.7. Protocolo SNMP v1

UDP no garantiza la entrega de los mensajes

Si se pierde un mensaje...

GetRequest o GetNextRequest ⇨ se **repiten** los mensajes

SetRequest ⇨ primero se **comprueba** si la operación se ha realizado con un GetRequest y si es necesario se **repite** el mensaje SetRequest

Trap ⇨**no se puede saber** ⇨ periódicamente el administrador debe contactar con los agentes (**polling**)

6.8.7. Protocolo SNMP v1

Al tratarse de un sistema distribuido hay que agrupar los equipos que pertenecen al entorno de gestión: las comunidades SNMP

El concepto de comunidad permite ofrecer servicios de:

Autenticación: el agente puede limitar el acceso a las MIB nombres a comunidades autorizadas

Políticas de acceso: el agente puede asignar diferentes privilegios a cada administrador

Proxy: un agente puede actuar como proxy de los agentes de una otra comunidad

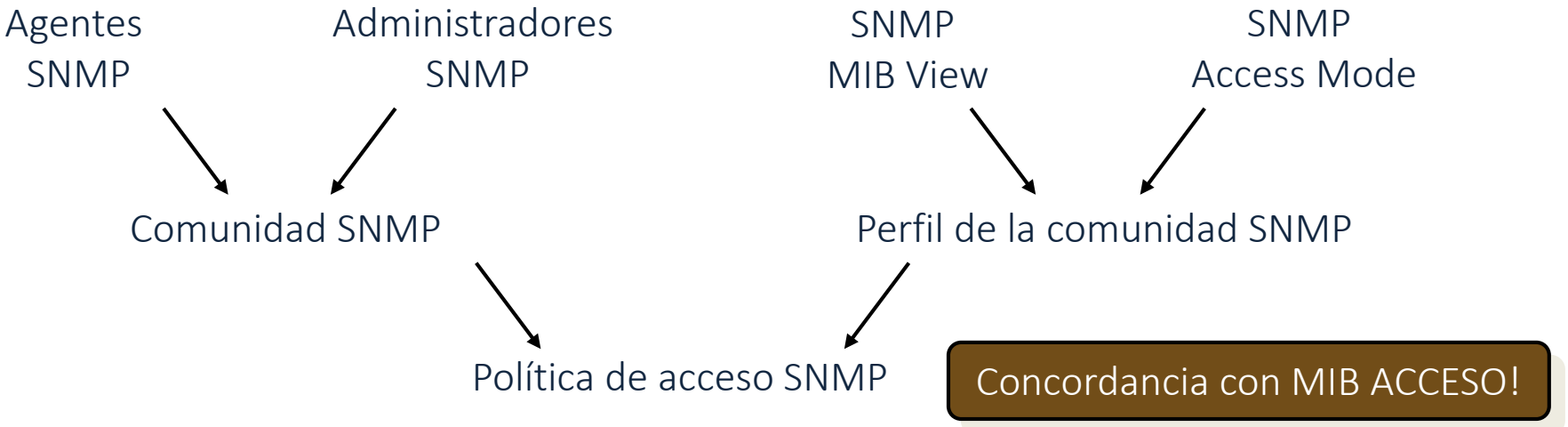
6.8.7. Protocolo SNMP v1

mediante políticas de acceso se gestionan los privilegios sobre los objetos de las comunidades

Se lleva a la práctica mediante dos conceptos:

Views: selecciones de objetos de las MIBs

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



6.8.7. Protocolo SNMP v1

Secuencia de transmisión de los mensajes SNMP

1. Se construye la PDU mediante l'ASN.1
2. Se crea el cuerpo del mensaje, junto con datos de seguridad, y se procesa con los mecanismos de autentificación i cifrado
3. Se añaden los campos de la cabecera
4. Se codifican con las BER t se pasan a la capa de transporte UDP

6.8.7. Protocolo SNMP v1

Secuencia de recepción de los mensajes SNMP

1. La capa de transporte pasa el mensaje y es descodifica con las BER
2. Se analiza la cabecera
3. Con los datos de seguridad se autentica y entonces se desencripta el cuerpo del mensaje
4. Mediante la ASN.1 se extraen los objetos

6.8.7. Protocolo SNMP v1

Tipo de mensajes SNMP v1

El SNMP especifica 5 tipos de mensajes:

Get Request: el gestor pide el valor de variables de la MIB del agente

Get next Request: no hace falta el nombre de la variable

Set Request: el gestor pide cambios en valores de variables de la MIB del agente

Get Response: respuestas del agente al gestor a los comandos anteriores. Contiene el mensaje original mas la respuesta

6.8.7. Protocolo SNMP v1

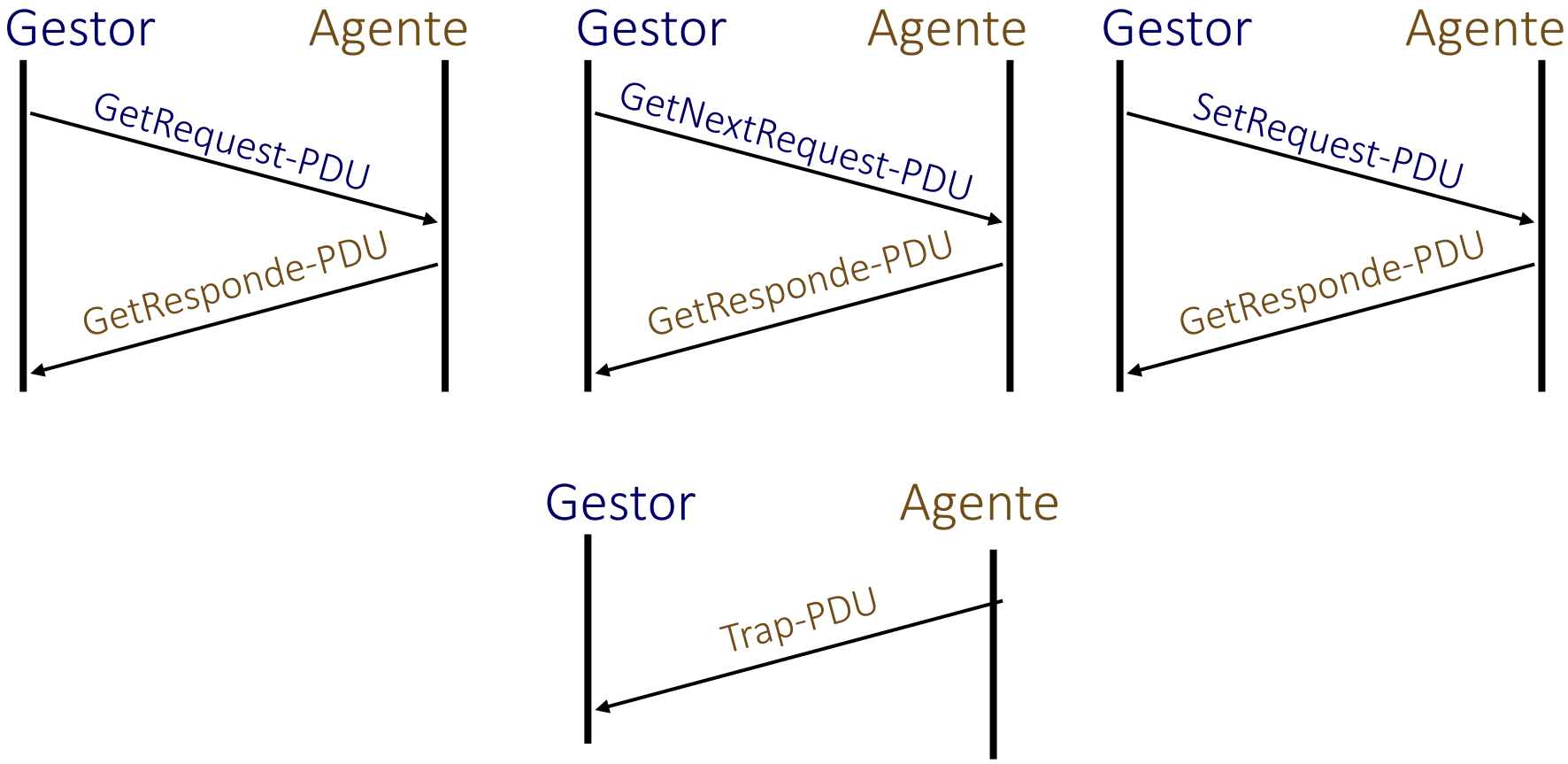
Tipo de mensajes SNMP v1

Trap: respuestas no solicitadas. Alarmas:

- **coldStart:** el agente se ha inicializado por si solo
- **warmStart:** el agente se ha reiniciado por si solo
- **linkDown:** la interfaz se ha desactivado
- **linkUp:** la interfaz se ha activado
- **authentificationFailure:** mensaje recibido de un agente que no pertenece a la comunidad
- **egpNeighborLos:** un EGP se ha desactivado (se envía su dirección IP)
- **enterpriseSpecific:** código de alarma específico

6.8.7. Protocolo SNMP v1

Diagrama de flujo de los mensajes



6.8.7. Protocolo SNMP v1

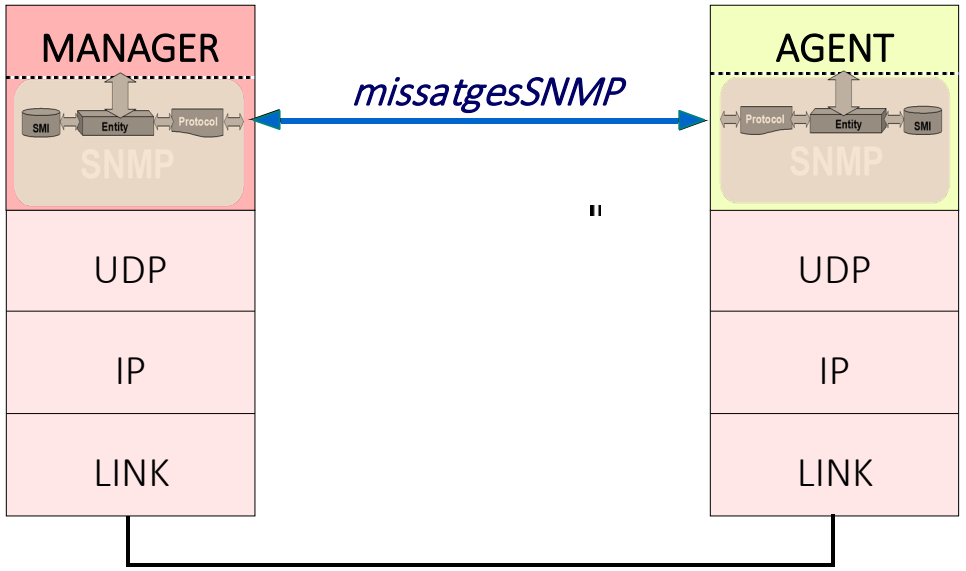
Campos de los mensajes SMNP v1

Versión del protocolo: 0 por la v1, 1 por la v2 i 3 por la v3

Nombre de la comunidad: se utiliza como contraseña

Identificador de mensaje: indica el tipo de mensaje

Identificador de solicitud: se utiliza para relacionar solicitudes y respuestas



6.8.7. Protocolo SNMP v1

Campos de los mensajes SMNP v1

Estado del error: 0 para las peticiones. A las respuestas un valor diferente de 0 indica un error

- noError (0): no hay error
- tooBig (1): el resultado no cabe en el PDU del response
- noSuchName (2): el objeto no existe
- badValue (3): valor del objeto incorrecto (respuesta a un set)
- readOnly (4): objeto nombres de lectura (respuesta a un set)
- genError (5): Causa de error desconocida

Índice de error: 0 a las peticiones. A les respuestas indica la variable que ha causado los problemas

6.8.7. Protocolo SNMP v1

Campos de los mensajes SMNP v1

Identificador de objeto: tipo del objeto que ha generado el error (sysObjectID)

Dirección del agente: dirección del objeto que ha generado la alarma

Identificador de alarma: indicador del tipo de alarma

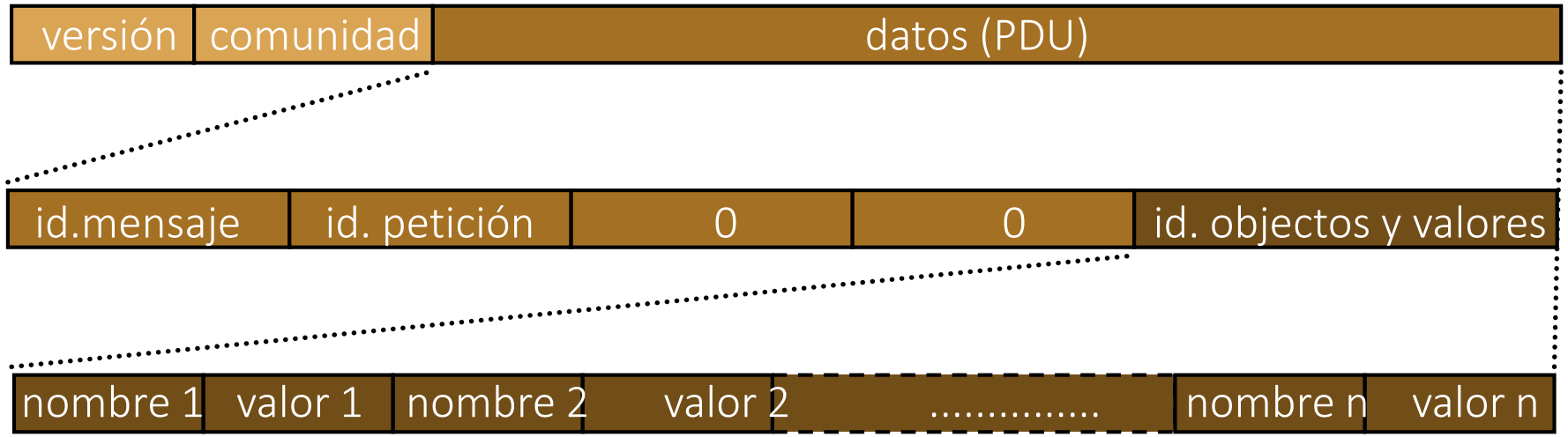
Identificador del código de alarma: código de la alarma

Marca de tiempo: tiempo que ha pasado entre la ultima reinicialización del dispositivo i la alarma actual

Id de objetos y valores: variables y valores pedidos/respondidos. Hay tantos como variables se pidan/respondan

6.8.7. Protocolo SNMP v1

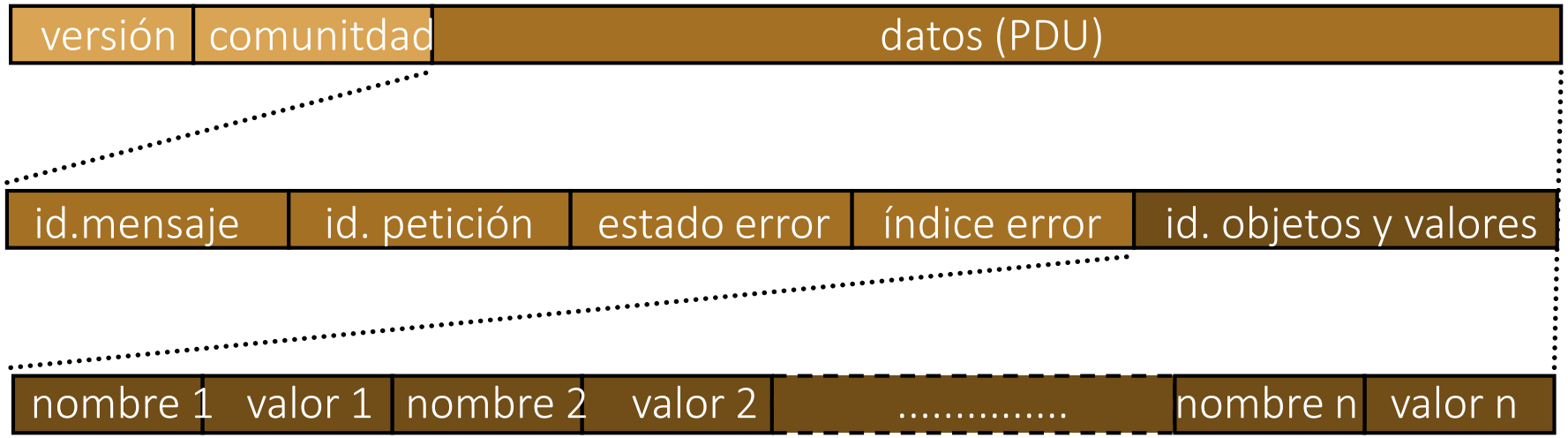
Missatges Get Request, Get Next Request i Set Request



PDU= Protocol Data Unit

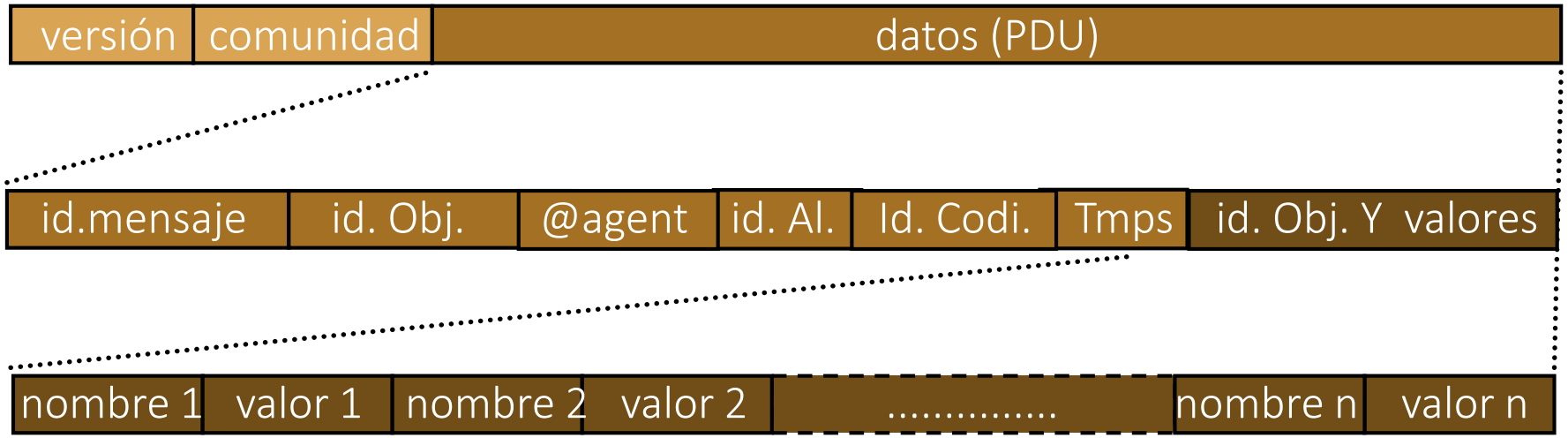
6.8.7. Protocolo SNMP v1

Missatges Get Response



6.8.7. Protocolo SNMP v1

Missatges Trap



6.8.8. Protocolo SNMP v2

La versión 2 de SNMP viene recogida en 8 RFCs

RFC 1901: Introduction to Community-based SNMPv2

RFC 1902: Structure of Mangement 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. Protocolo SNMP v2

La versión 2 incorpora mejoras respecto SNMP v1 en 4 grandes bloques

Ámbito de aplicación: se agarra al concepto de SMP framework del SMP (Simple Management Protocol)

Tamaño, velocidad y eficiencia: se incorporan nuevos mensajes

Seguridad i privacidad: se incorporan funciones de seguridad del Secure SNMP a la versión original, que se sacan a la versión revisada

Desarrollo y compatibilidad: basándose en l SMP se aumenta la compatibilidad con nuevas arquitecturas de comunicaciones OSI

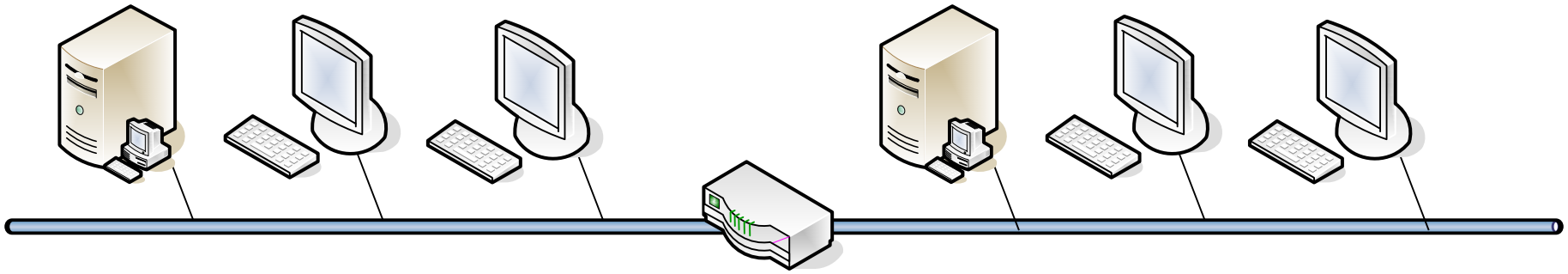
6.8.8. Protocolo SNMP v2

En el ámbito de aplicación se proponen 3 tipos de acceso a las MIBs

Manager-agent request-response: Igual que en la v1, el administrador puede obtener y modificar las MIBs del agente

Manager-Manager request-response: nuevo ámbito en que dos administradores pueden compartir MIBs de un agente

Agent-Manager unconfirmed: Igual que en la v1, el agente puede enviar alertas al administrador



6.8.8. Protocolo SNMP v2

En el ámbito de desarrollo se aumentan las arquitecturas de transporte

SNMPv2 puede utilizar los servicios de transporte de los protocolos:

UDP: User Datagram Protocol

CLNS: OSI ConnectionLes_Mode Network Service

CONS: OSI Connection-Oriented Network Service

IPX: Novell Internetwork Packet Exchange

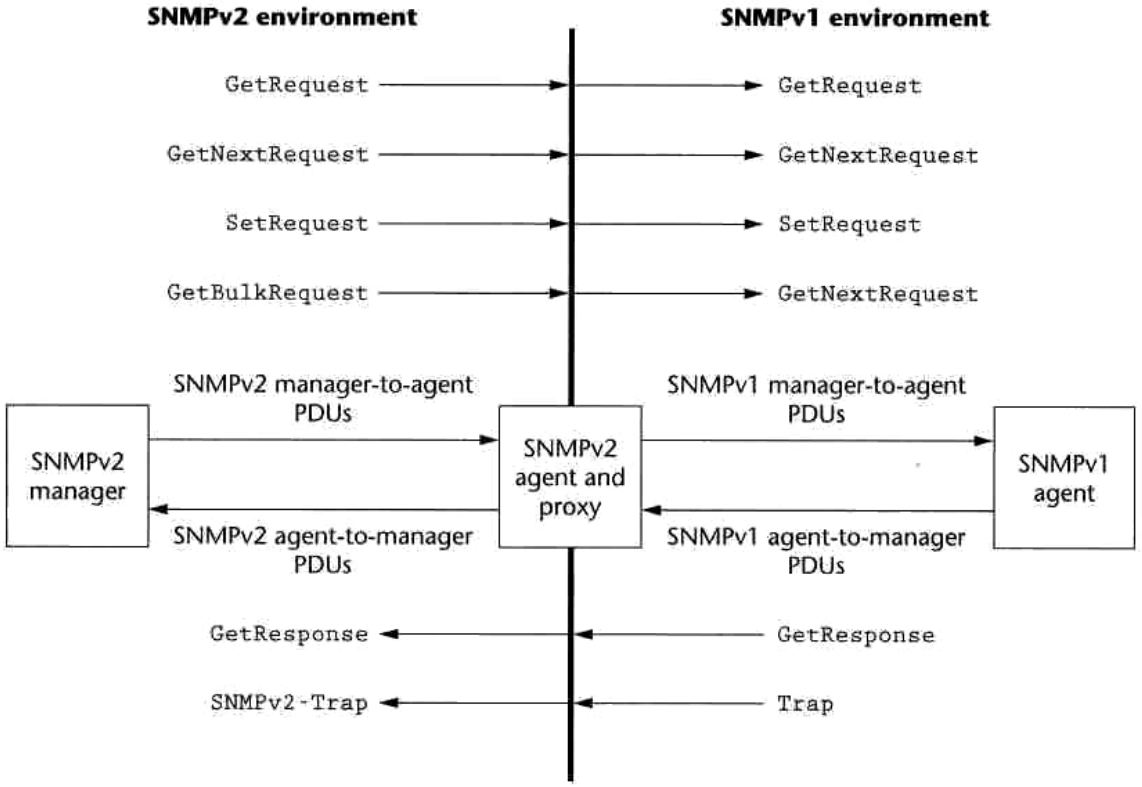
DDP (Appletalk): Apple Network

6.8.8. Protocolo SNMP v2

En el ámbito de compatibilidad se define como pueden coexistir la versión v1 y v2

Existen 2 alternativas:

- Incluir un *Proxy* que enlace las dos *comunidades*
- *Entidades* con *dos lectores* de mensajes



6.8.8. Protocolo SNMP v2

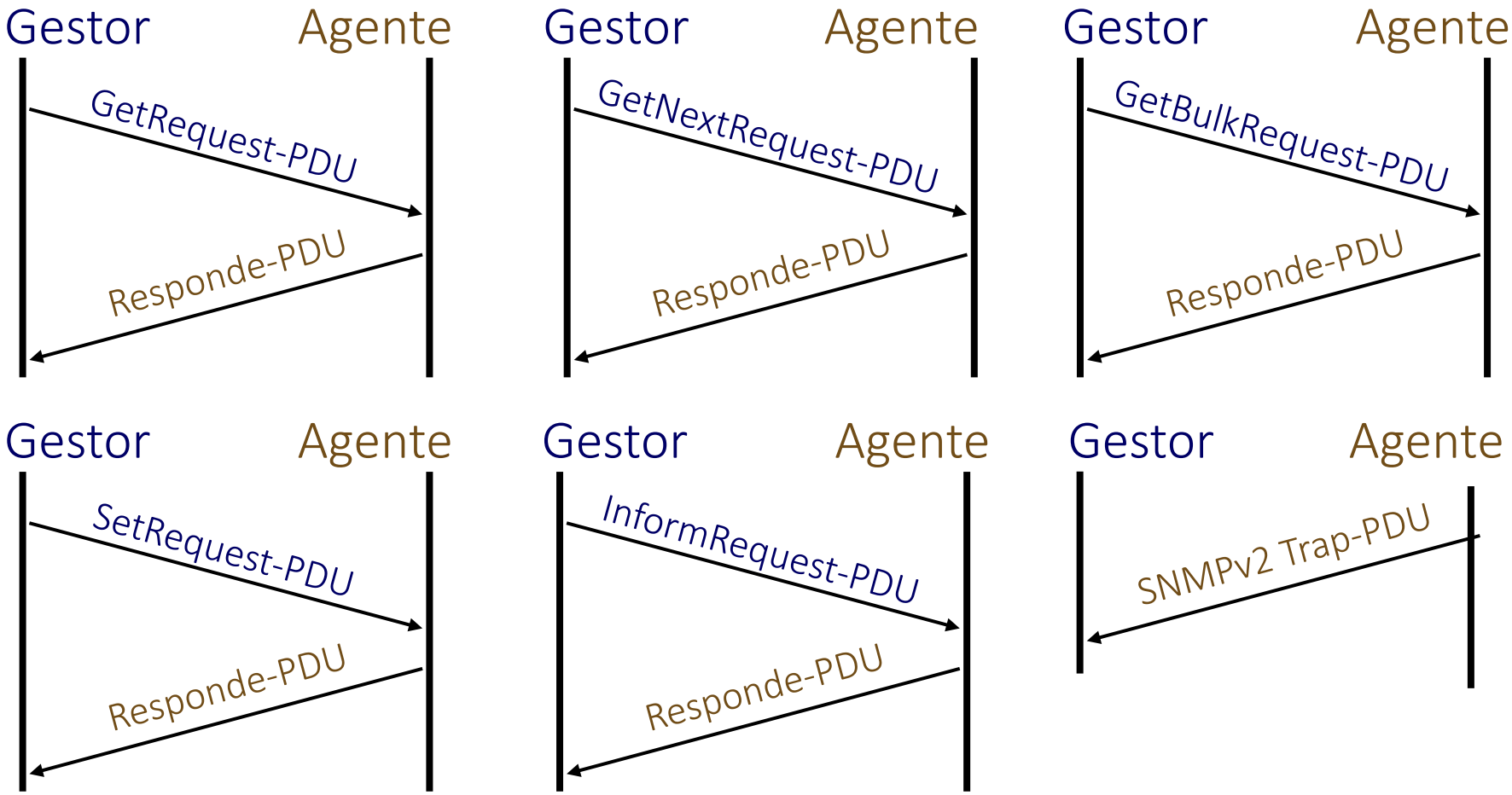
Tipo de mensajes SNMP v2

El SNMP especifica 7 tipos de mensajes:

- **Get Request:** se pide el valor de variables de la MIB
- **Get Next Request:** no hace falta el nombre de la variable
- **Get Next Bulk**: como *get next request*, pero permite pedir más de una variable a la ve y indicar repeticiones
- **Set Request:** se piden cambios en variables de la MIB
- **Response:** respuestas a los comandos anteriores
- **Inform Request**: informaciones entre gestores
- **Trap:** alarmas, sigue la macro *NOTIFICATION-TYPE*
- **Report**: sin definir

6.8.8. Protocolo SNMP v2

Diagrama de flujo de los mensajes



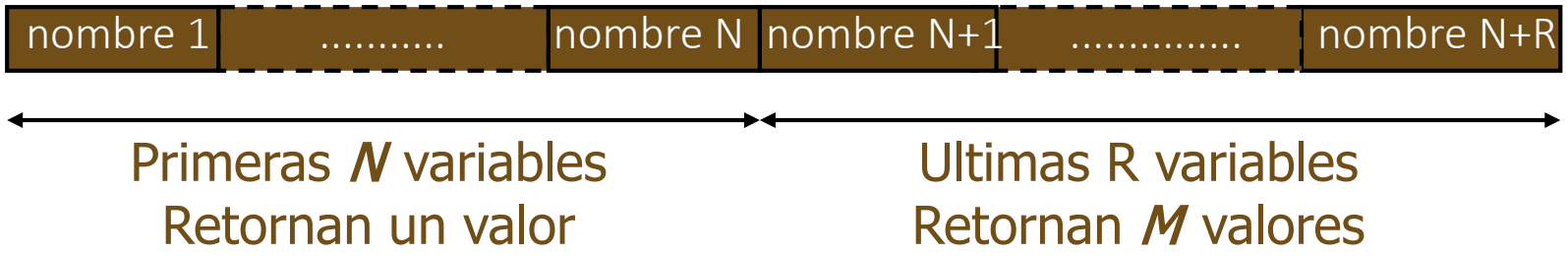
6.8.8. Protocolo SNMP v2

Campos de los mensajes SMNP v2

A los campos descritos en la versión 1 hay que añadir los dos siguientes:

No repetir: número de variables sin repetir (N)

Max iter: número máximo de repeticiones (M)



6.8.8. Protocolo SNMP v2

Campos de los mensajes SMNP v2

El campo error puede tomar los siguientes valores:

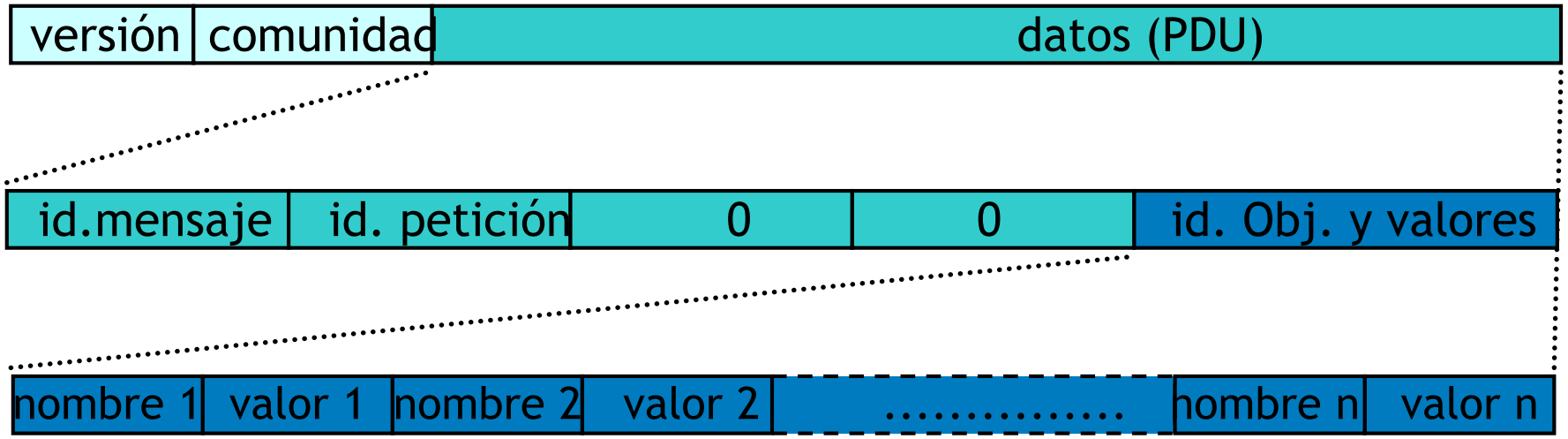
	(1)	(2)	(3)
noError(0)	X	X	X
tooBig(1)	X	X	X
noSuchName(2)			
badValue(3)			
readOnly(4)			
genError(5)	X	X	X
noAccess(6)		X	
wrongType(7)		X	
wrongLength(8)		X	
wrongEncoding(9)		X	

	(1)	(2)	(3)
wrongValue(10)		x	
noCreation(11)		X	
InconsistentValue(l2)		X	
resourceUnavailable(13)		X	
commitFailed(14)		x	
undoFailed(15)		X	
authorizationError(16)	X		
notWritable(17)		X	
inconsistentName (18)		X	

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

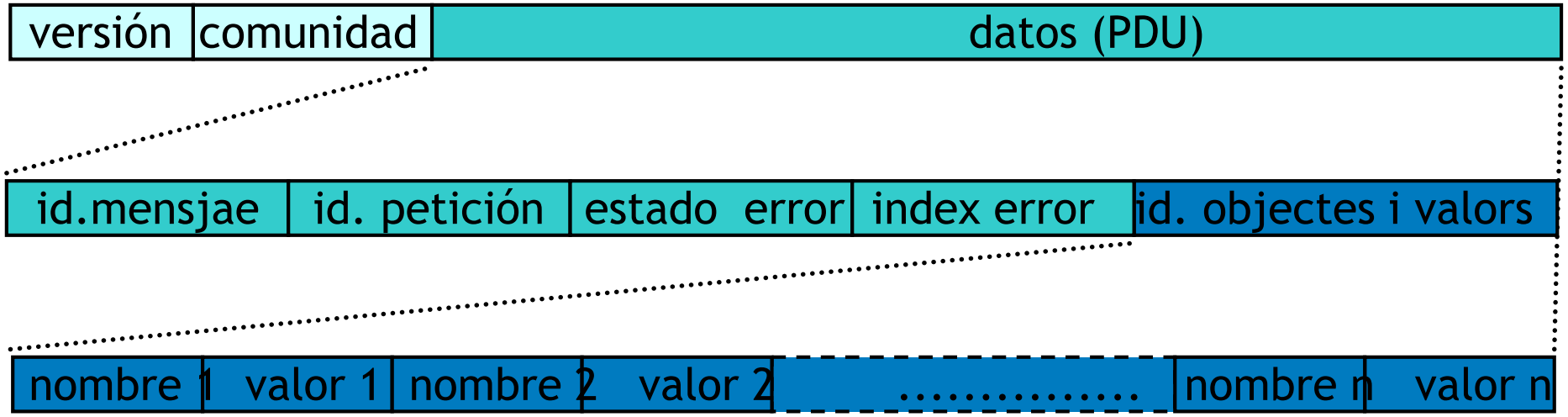
6.8.8. Protocolo SNMP v2

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



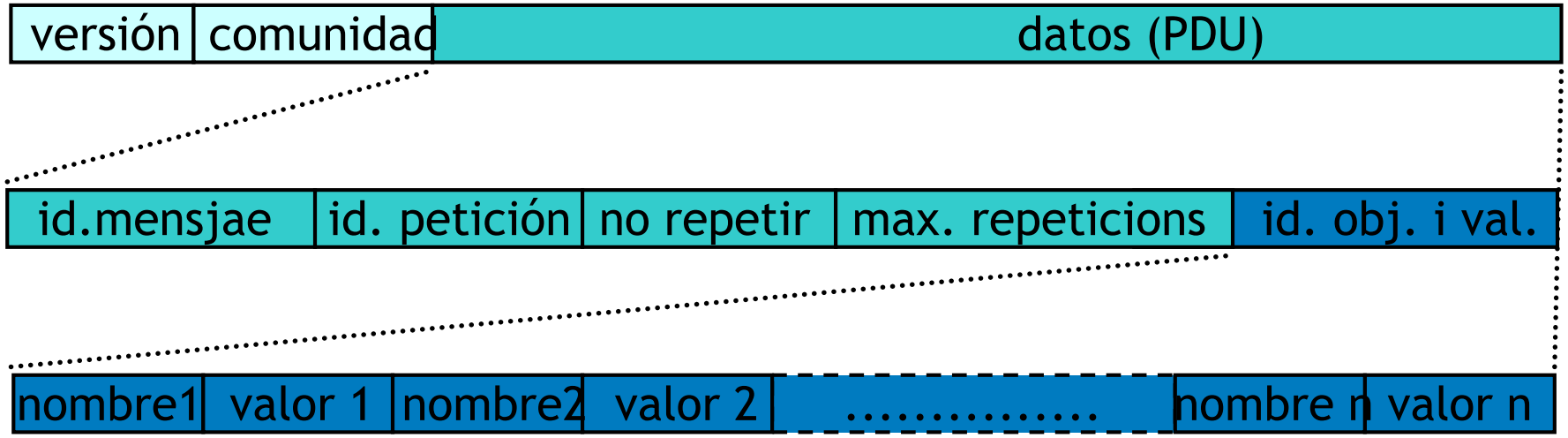
6.8.8. Protocolo SNMP v2

Missatges Response



6.8.8. Protocolo SNMP v2

Missatges Get Bulk Request



6.8.9. Protocolo SNMP v3

La versión 3 de SNMP viene recogida 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 Acces Control Model (VACM) for SNMP

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

6.8.9. Protocolo SNMP v3

La versión 3 incorpora mejoras en 3 grandes bloques

Seguridad:

- USM: User-Based Security Modelo
- VACM View-Based acceso Modelo

Nueva estructura de los mensajes

- Hay los mismos mensajes que en SNMPv2
- Se aplica encapsulamiento para aplicar habilidades de seguridad

Se amplían les MIBs

- Se definen nuevos objetos y estructuras

6.8.9. Protocolo SNMP v3

El User-based Security Model proporciona los servicios de autenticación i cifrado

Protege de:

- Suplantación de identidad
- Modificaciones de la información
- Alteración del flujo de mensajes (reordenación, duplicado,...)
- Revelado de información

No protege de:

- Denegación del servicio
- Análisis del tráfico de mensajes

6.8.9. Protocolo SNMP v3

El View-Based Access Control Model proporciona servicios de control de acceso a las mIBs (RFC2275)

Tiene dos funciones principales

- Determinar que tipo de acceso tiene una entidad remota a un objeto local
- Determinar la política de control de acceso por el agente y la habilita para la configuración remota

Se basa en 5 elementos

- Grupos: agrupaciones de asociaciones <securityModelo,securityName>
- Nivel de seguridad: los derechos sobre un grupo dependerán del nivel de seguridad del mensaje
- Contexto: subconjunto de objetos de una MIB local
- Vistas de las MIBs: subárbol de la SMI local
- Política de acceso: aplica prioritariamente un juego de privilegios de acceso

6.8.9. Protocolo SNMP v3

Camps dels missatges SMNP v3

A los campos descritos en las versiones 1 y 2 hay que añadir:

Versión: indica la versión de SNMP para la versión 3 vale 3.

Identificador de mensaje: se utiliza para relacionar solicitudes y respuestas entre dos entidades [0, 2^{31}-1]

Tamaño máx. indica el nº máximo de bytes que el emisor soporta en los mensajes [484, 2^{31}-1]

Atributos: octeto string donde los tres bits menos significativos significan:

- *Report:* a '1' fuerza el envío de un PDU report al emisor
- *Private:* a '1' el mensaje se ha encriptado
- *Autor:* a '1' el mensaje lleva autenticación (P=0 i A=1 no permitido)

6.8.9. Protocolo SNMP v3

Campos de los mensajes SMNP v3

Parámetros seguridad: contiene los parámetros de seguridad USM

- ***Id motor autentificación***: parámetro *snmpEngineID* del motor de autentificación involucrado en la comunicación.
- ***Id boot autentificación***: parámetro *snmpEngineBoots* del motor de autentificación involucrado en la comunicación.
- ***Id time autentificación***: parámetro *snmpEngineTime* del motor de autentificación involucrado en la comunicación.
- ***Nombre usuario***: indica de quien procede el mensaje.
- ***Parámetros de autentificación***: código de autentificación de mensaje HMAC
- ***Parámetro de privacidad:*** valor inicial para el algoritmo DES CBC

6.8.9. Protocolo SNMP v3

Campos de los mensajes SMNP v3

Modelo seguridad: indica los métodos de seguridad aplicados [0, 2^{31}-1]. Están reservados 1(SNMPv1), 2(SNMPv2), 3(USM)

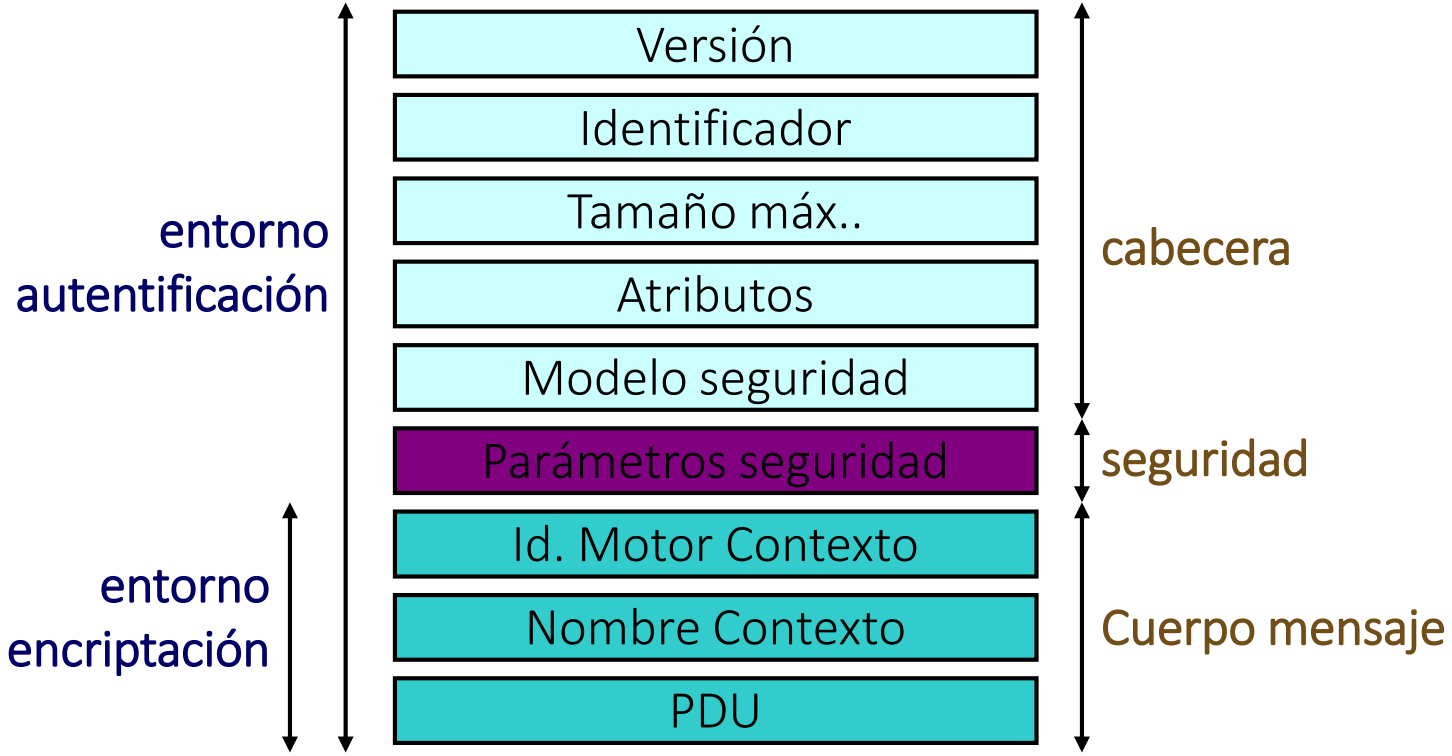
Id. Motor Contexto: identificador de la entidad SNMP que lo genera

Nombre Contexto: Identificador del contexto

PDU: datos SNMP versión 2!

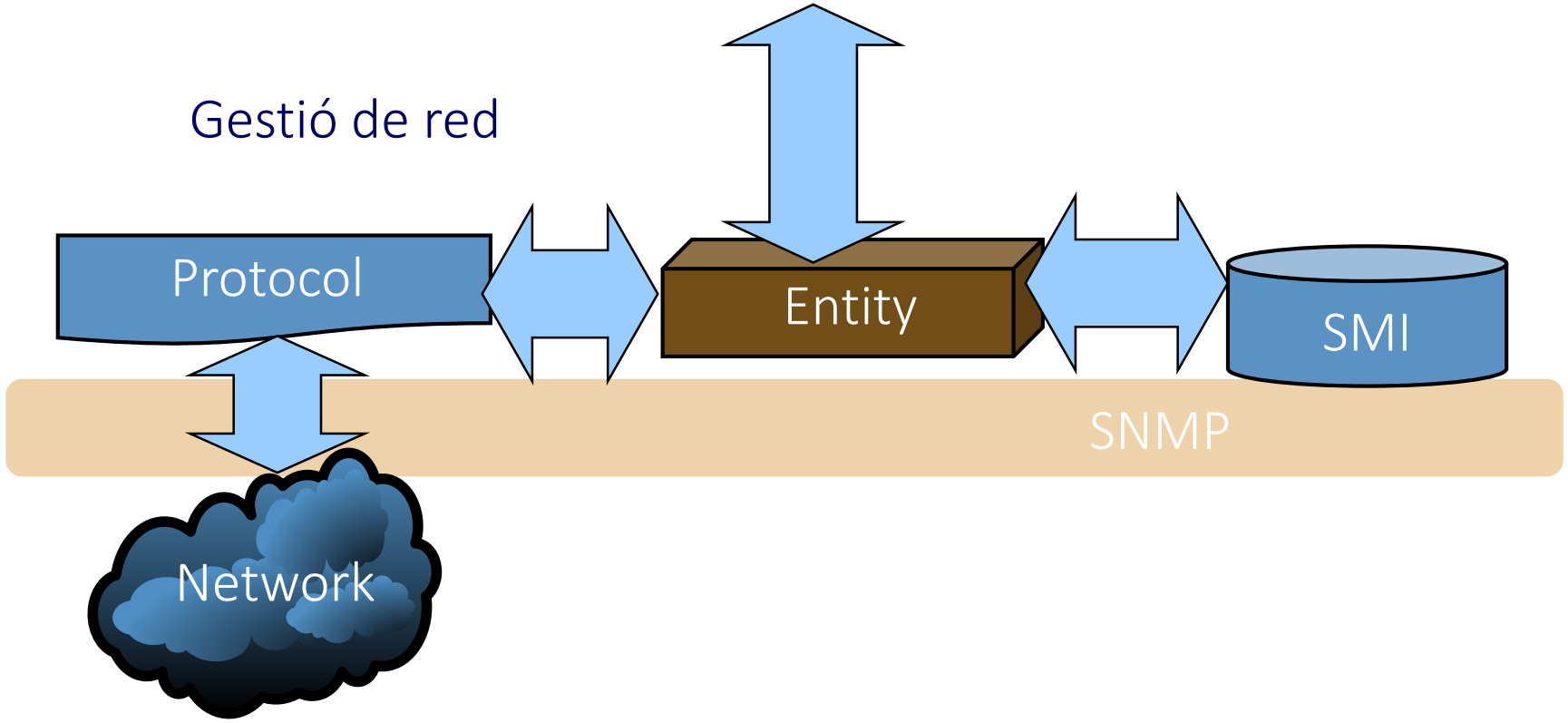
6.8.9. Protocolo SNMP v3

Mensajes SNMPv3



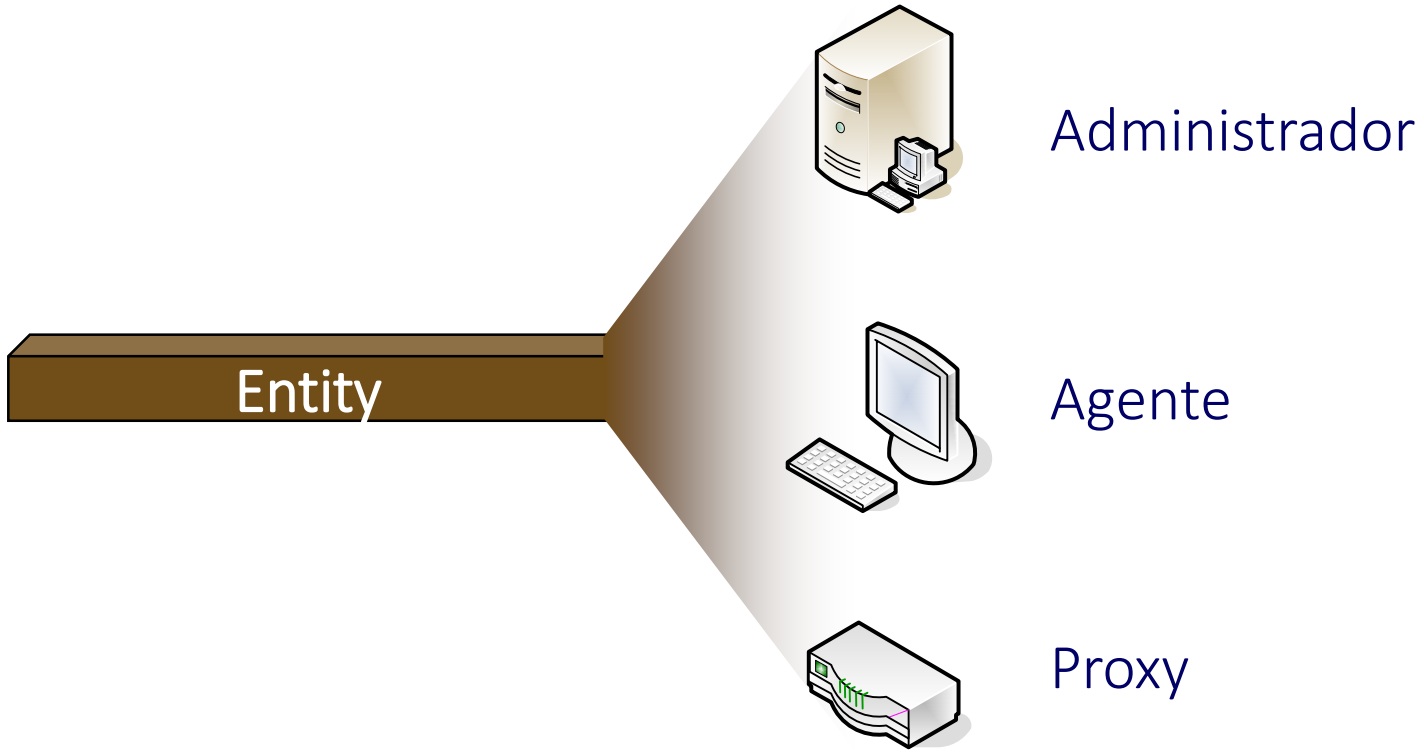
6.8.10. Entidades SNMP

El SNMP proporciona un entorno de gestión de redes



6.8.10. Entidades SNMP

En la arquitectura SNMP se definen 3 tipos de entidades

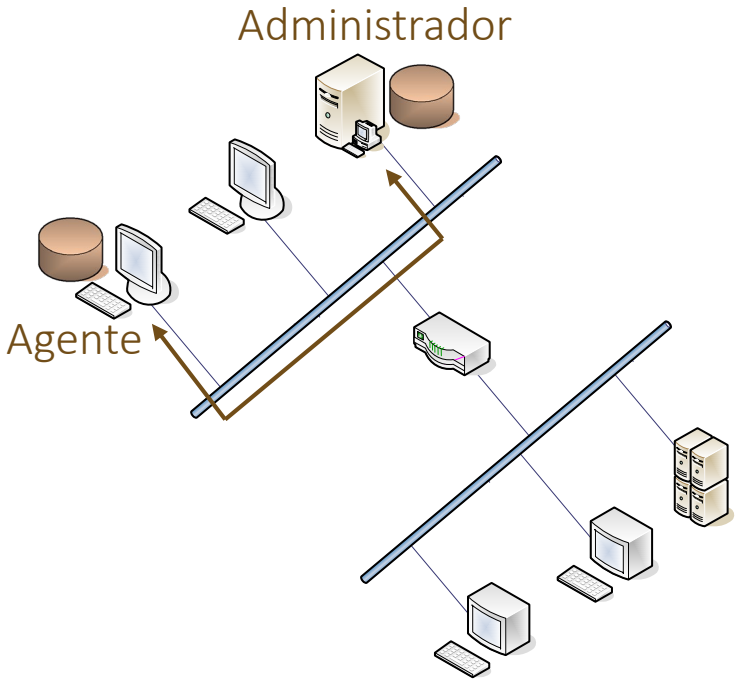


6.8.10. Entidades SNMP

Agentes

- Software instalado en todos los sistemas administrados

- Tareas
  - ✓ Responder a las peticiones de los administradores
  - ✓ Realizar actualizaciones
  - ✓ Informar de problemas

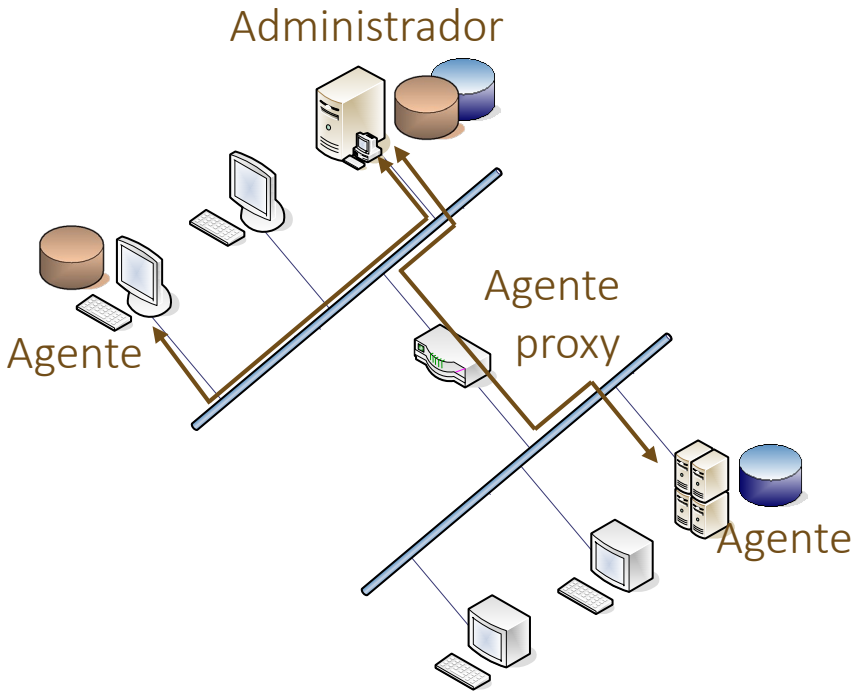


6.8.10. Entidades SNMP

Administradores

- Software instalado en todos los sistemas que gestionan la red

- Tareas
  - ✓ Enviar y recibir mensajes SNMP a los sistemas administrados
  - ✓ Gestionar la red

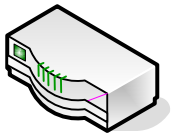


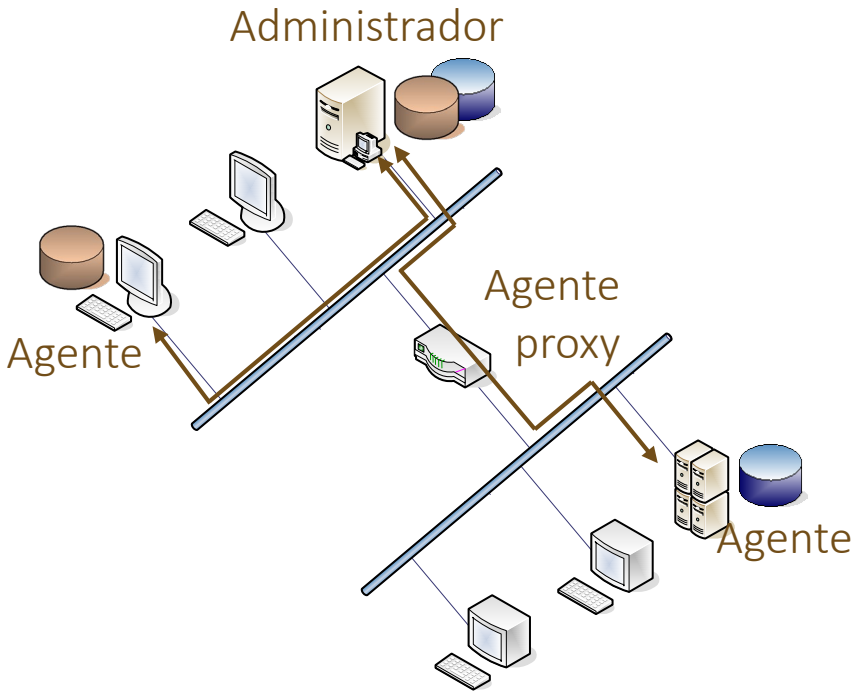
6.8.10. Entidades SNMP

Agentes proxy

- Software instalado en un equipo de gestión de red

- Tareas
  - ✓ Redirecciona los mensajes SNMP pero no interpreta los objetos
  - ✓Traduce entre diferentes versiones SNMP (v2)
  - ✓Traduce entre SNMP y otros protocolos de red (v2)

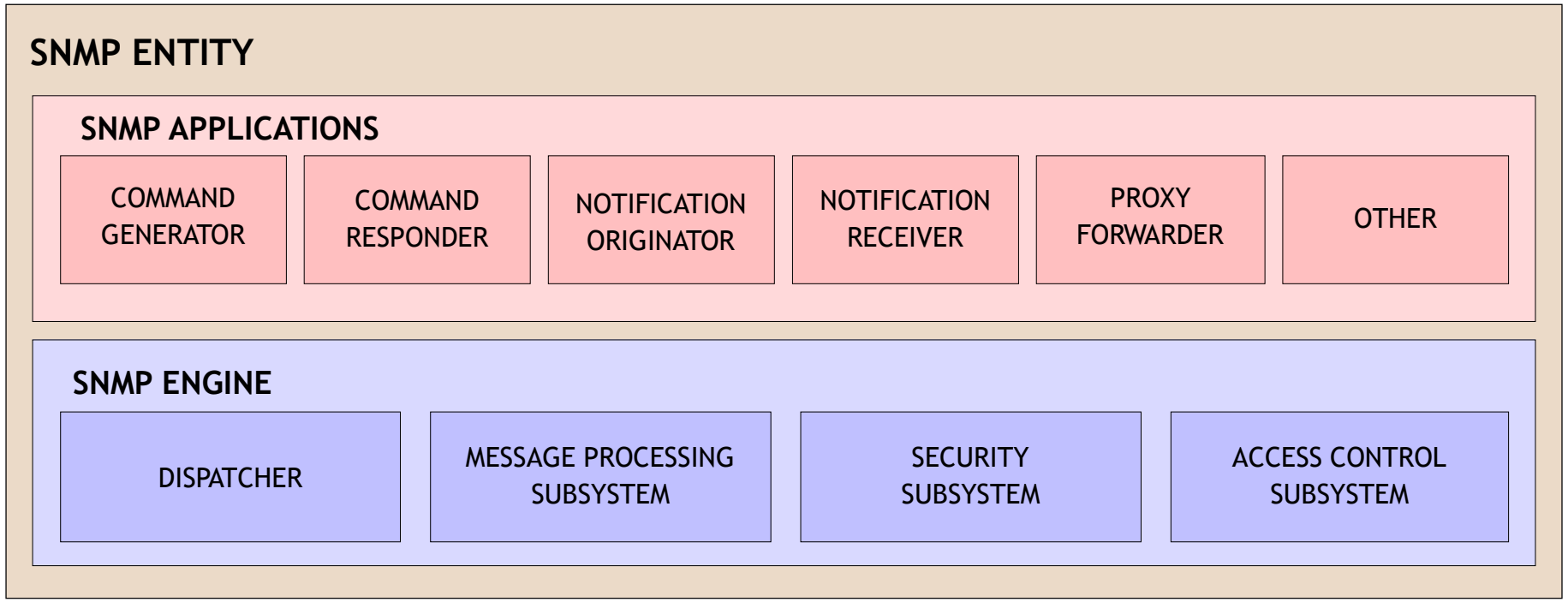




6.8.10. Entidades SNMP

Esquema general de una entidad SNMP

Basado en SNMPv3



6.8.10. Entidades SNMP

Esquema del motor de la entidad

Basado en SNMPv3

Dispatcher: gestor de los mensajes, es el encargado de distribuir los mensajes tanto internamente como por la red

Message Processing Subsystem: es el encargado de interpretar los mensajes

Security Subsystem: es el responsable de proporcionar seguridad en el flujo de mensajes

acceso Control Subsystem: es el responsable de otorgar los privilegios de acceso a los objetos

6.8.10. Entidades SNMP

Esquema de les aplicaciones de la entidad

Basado en SNMPv3

Command Generator: crea los mensajes *request* y interpreta los *response*

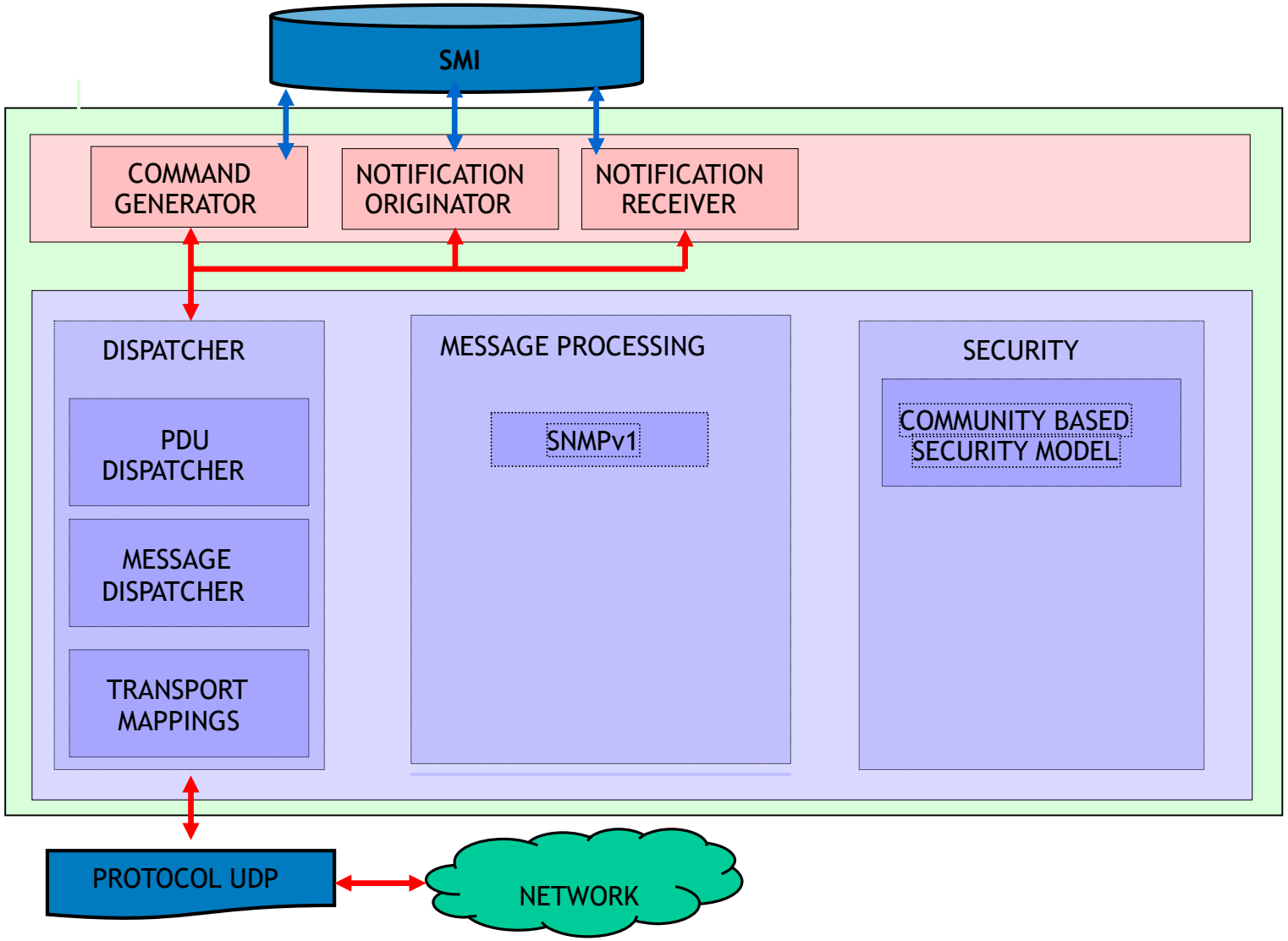
Command Responder: interpreta los mensajes *request* y genera los *response*

Notification Originator: monitoriza la red y genera los mensajes *inform* y *trap*

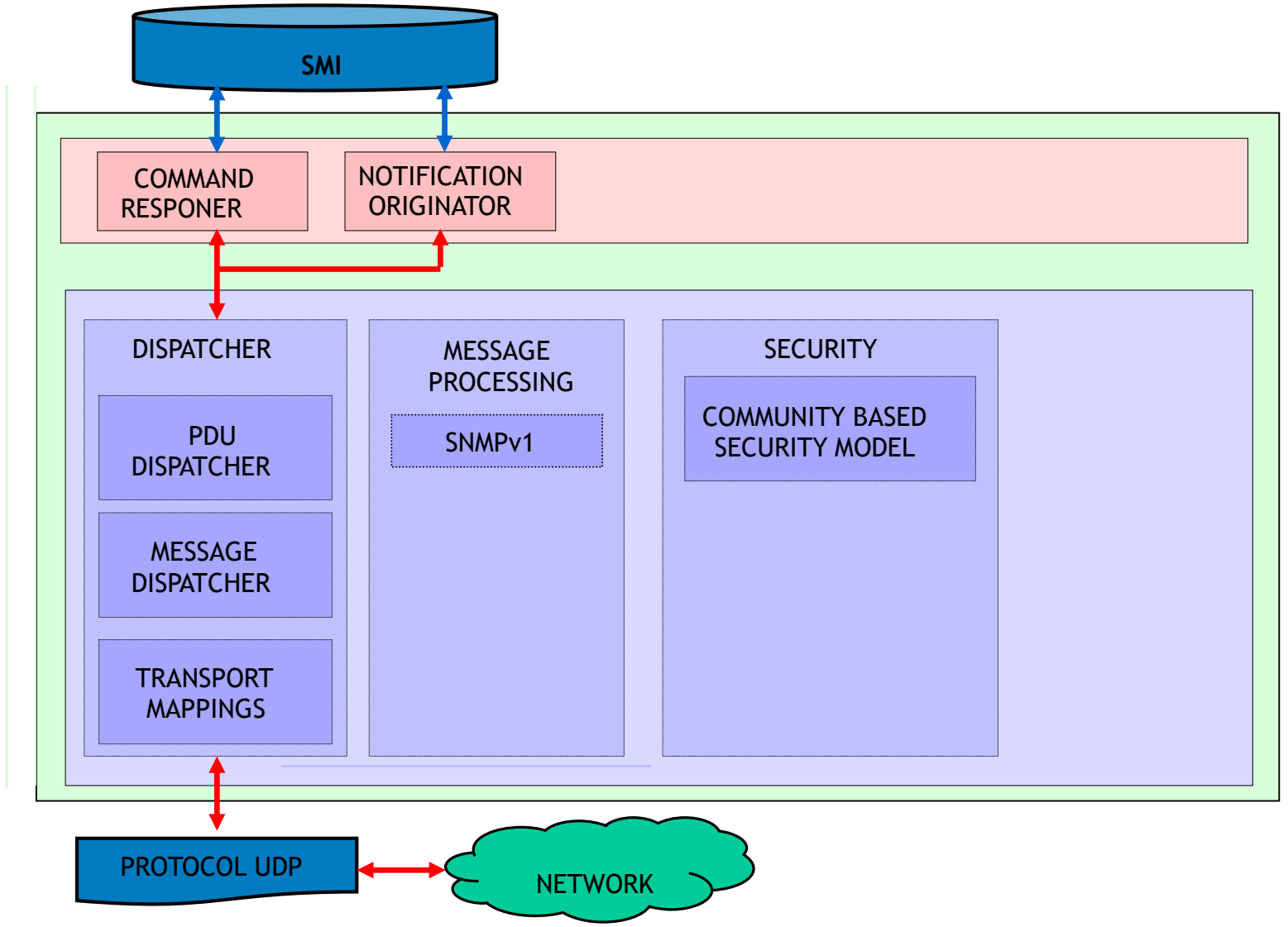
Notification Receiver: interpreta los mensajes *inform* y *trap*. En el caso de los informes genera el response correspondiente

Proxy Forwarder: redirecciones los mensajes SNMP

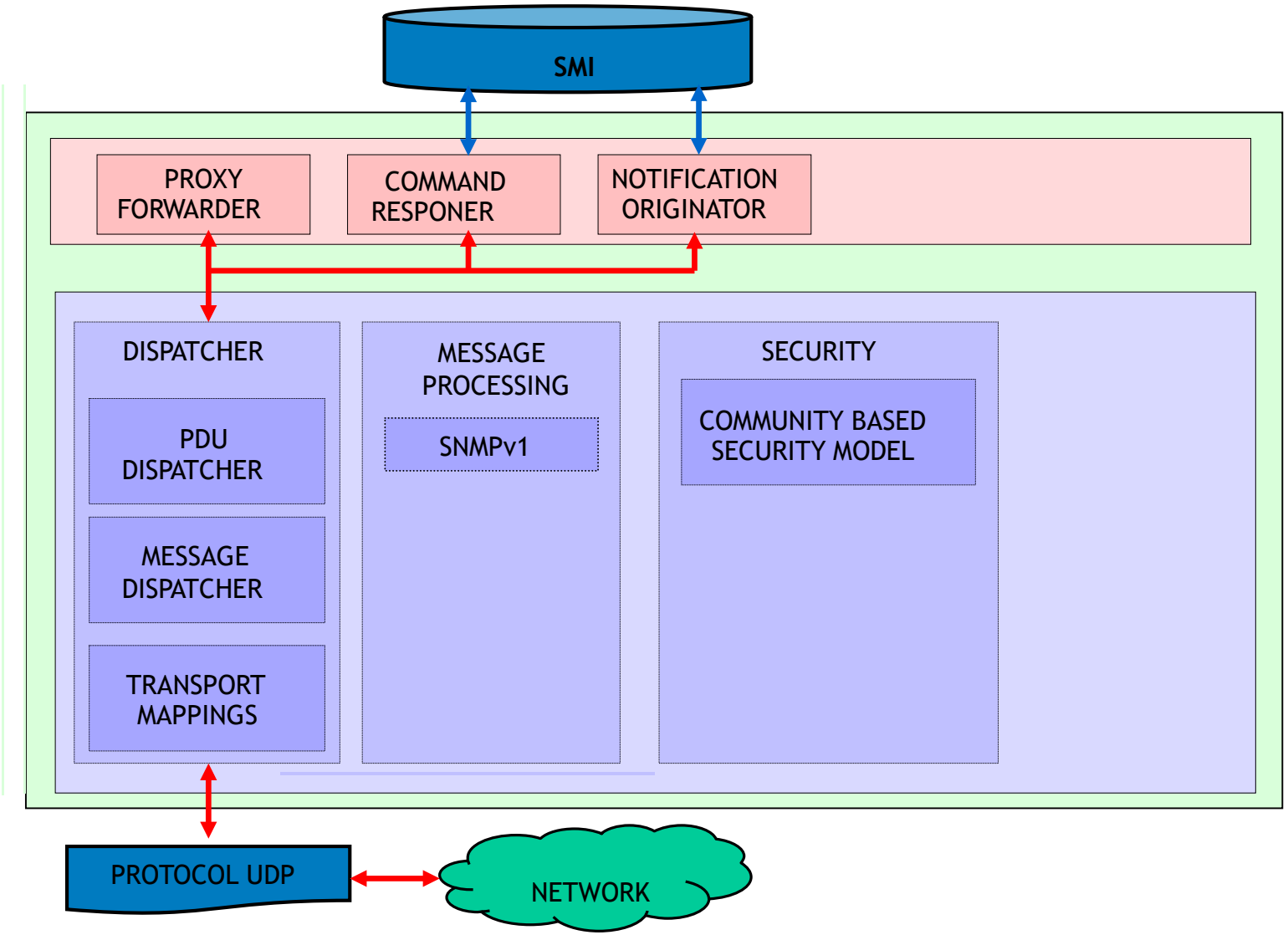
Entidad administrador



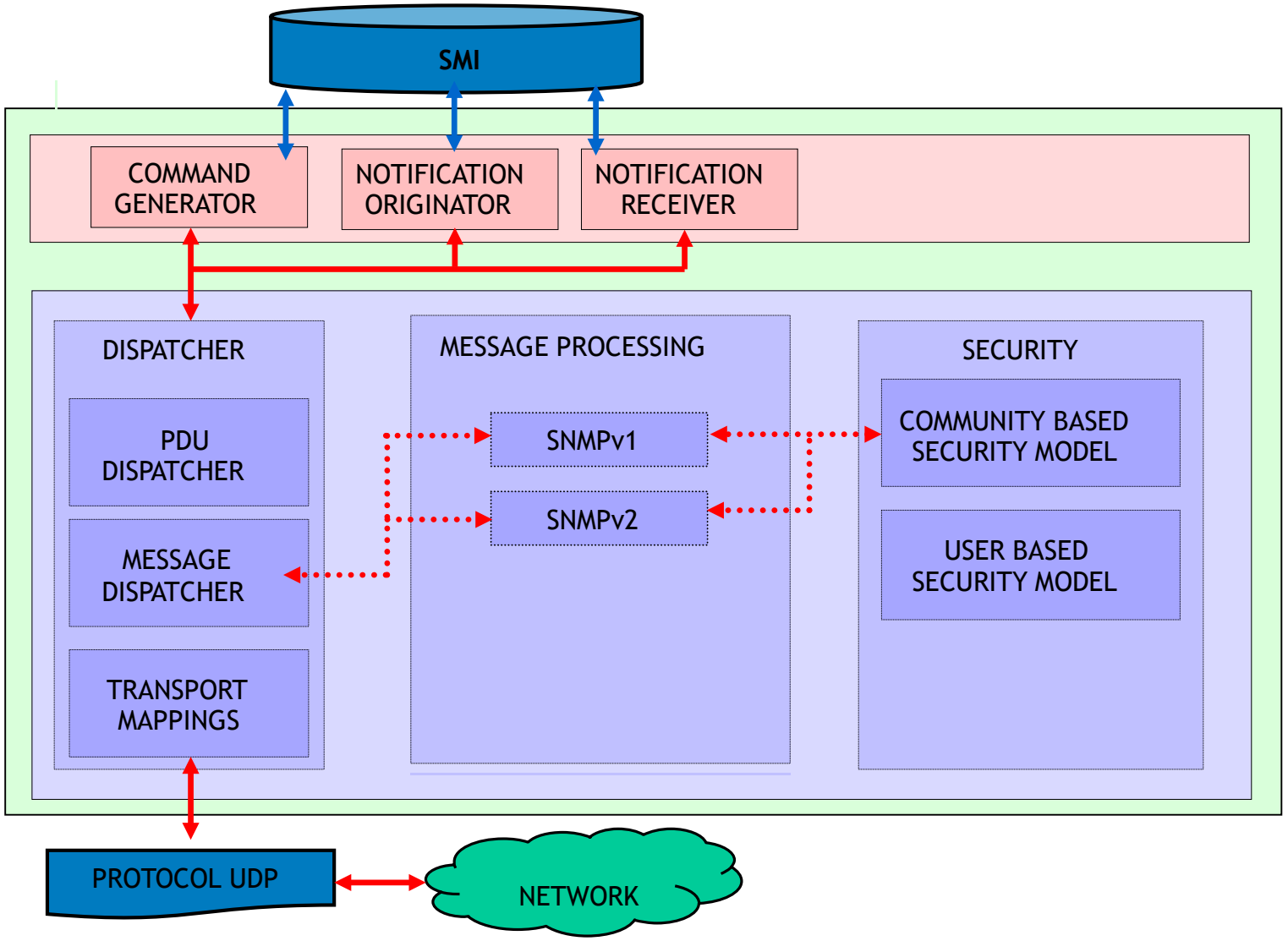
Entidad agente



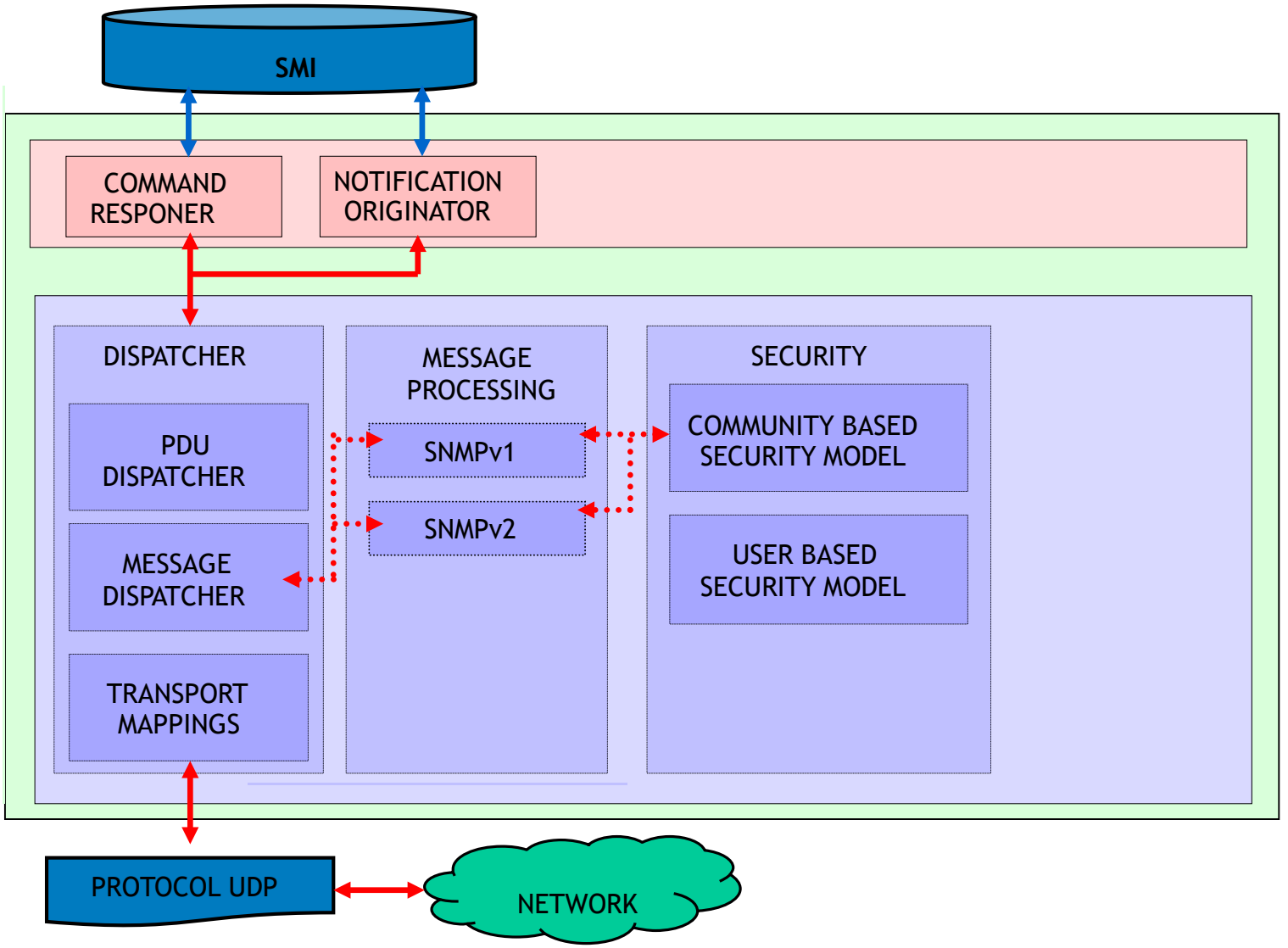
Entidad proxy



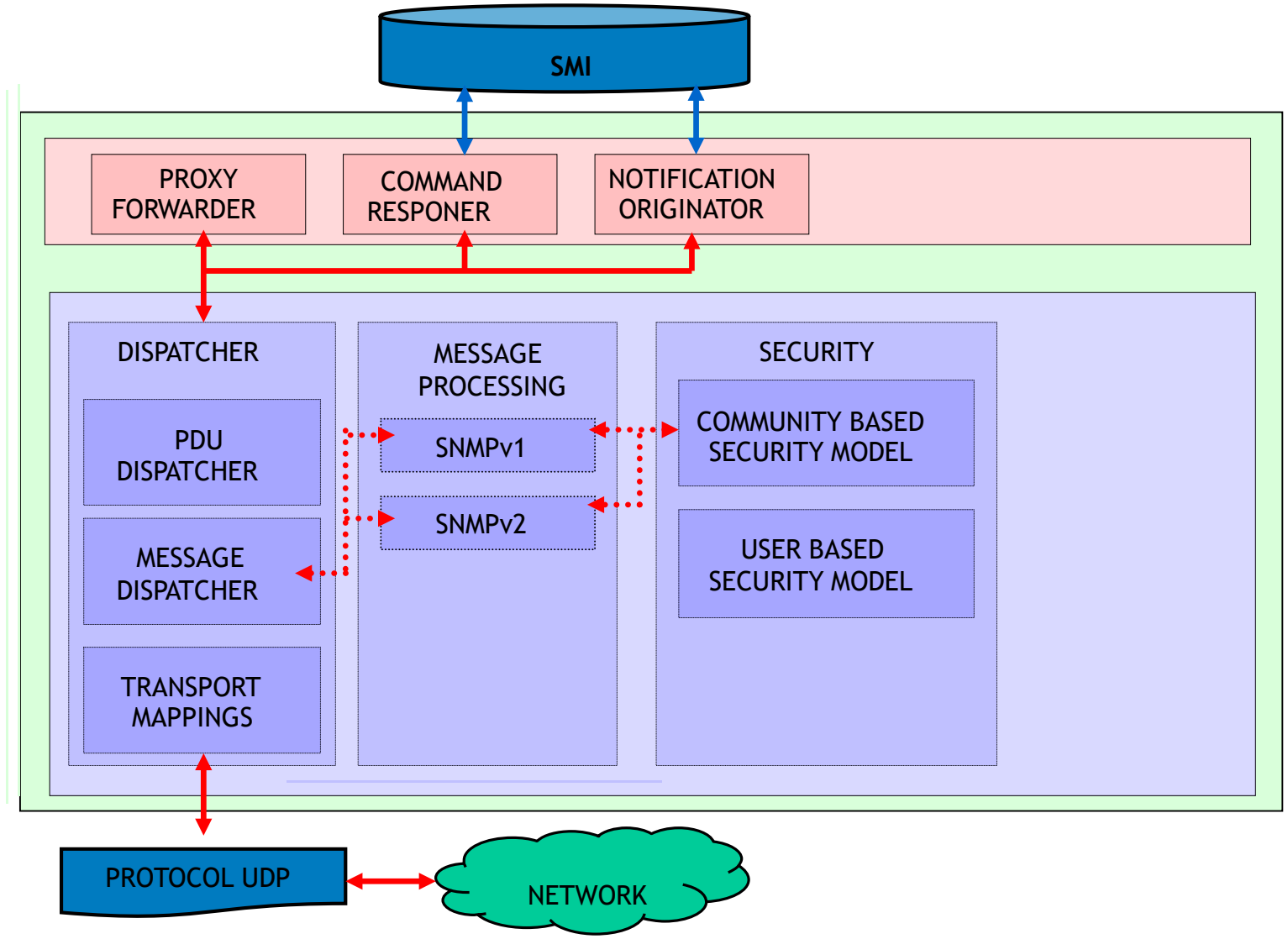
Entidad administrador



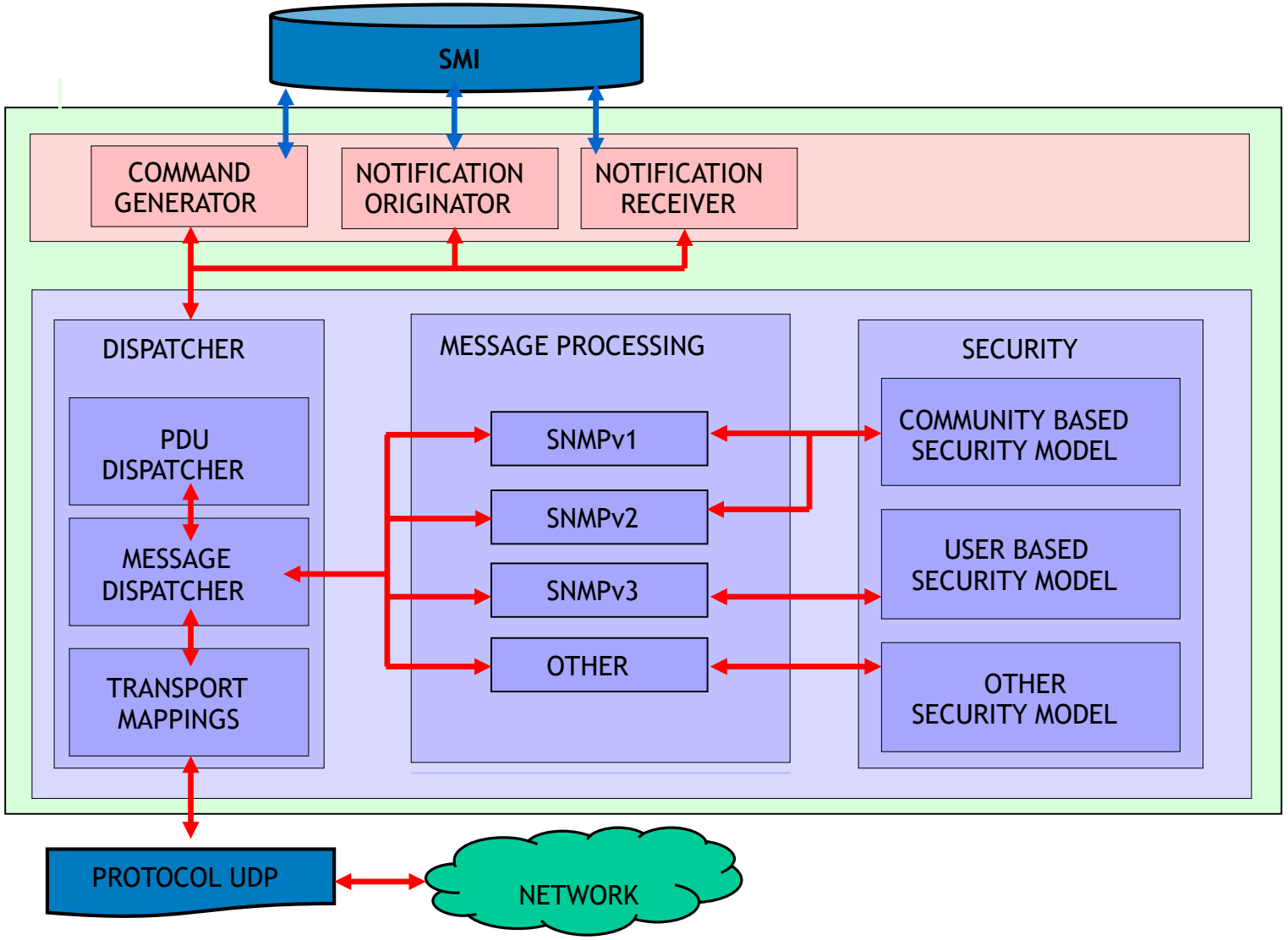
Entidad agente



Entidad proxy



Entidad administradora



Entidad agente