Escuela Complutense
Latinoamericana

Administración Avanzada y Redes TCP/IP en UNIX

Módulo 3:

Administración de redes TCP/IP en el entorno UNIX

Profesores:

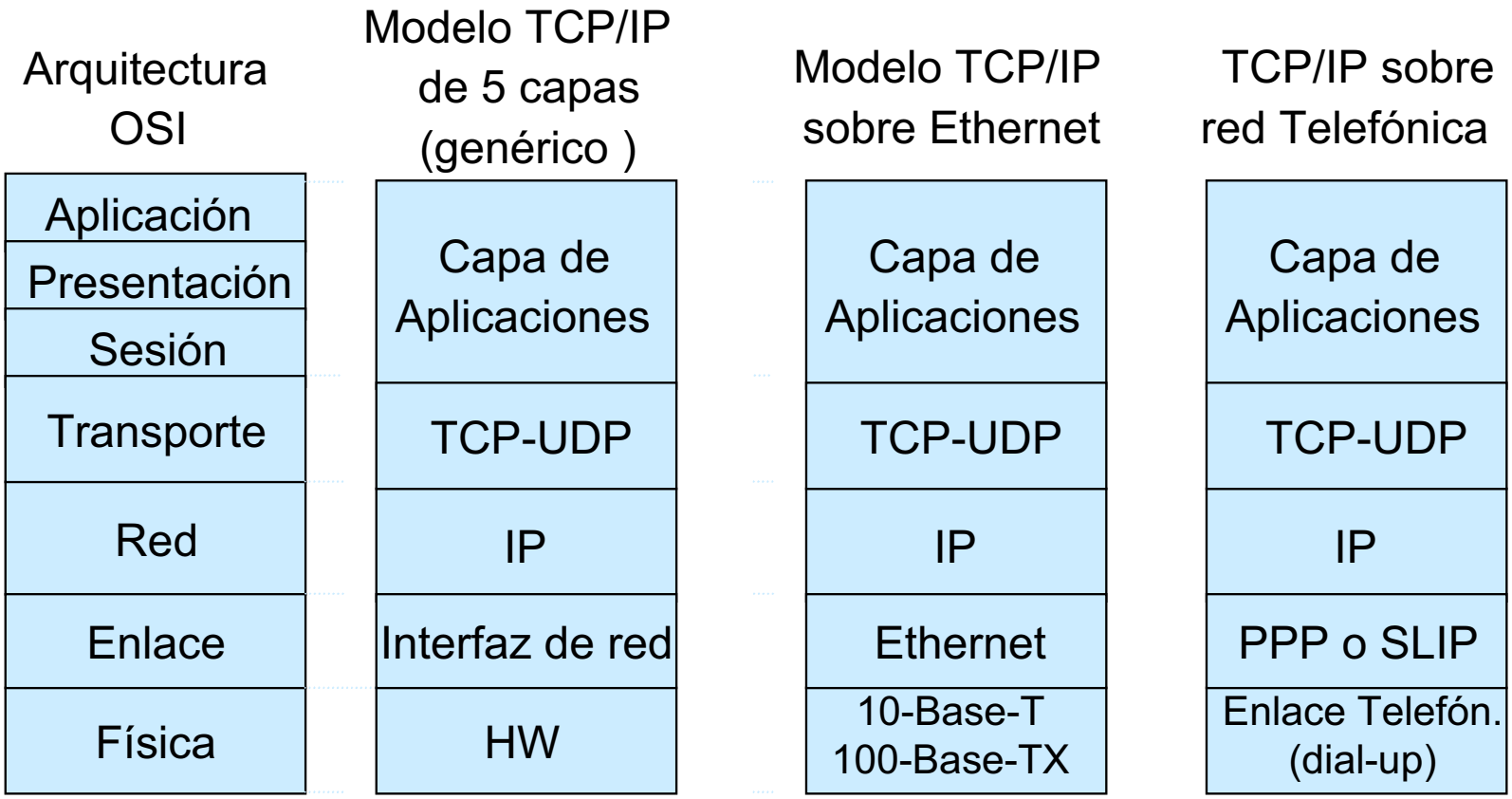
Dr. Rafael Moreno Vozmediano (UCM)

Dr. Juan Carlos Fabero Jiménez (UCM)

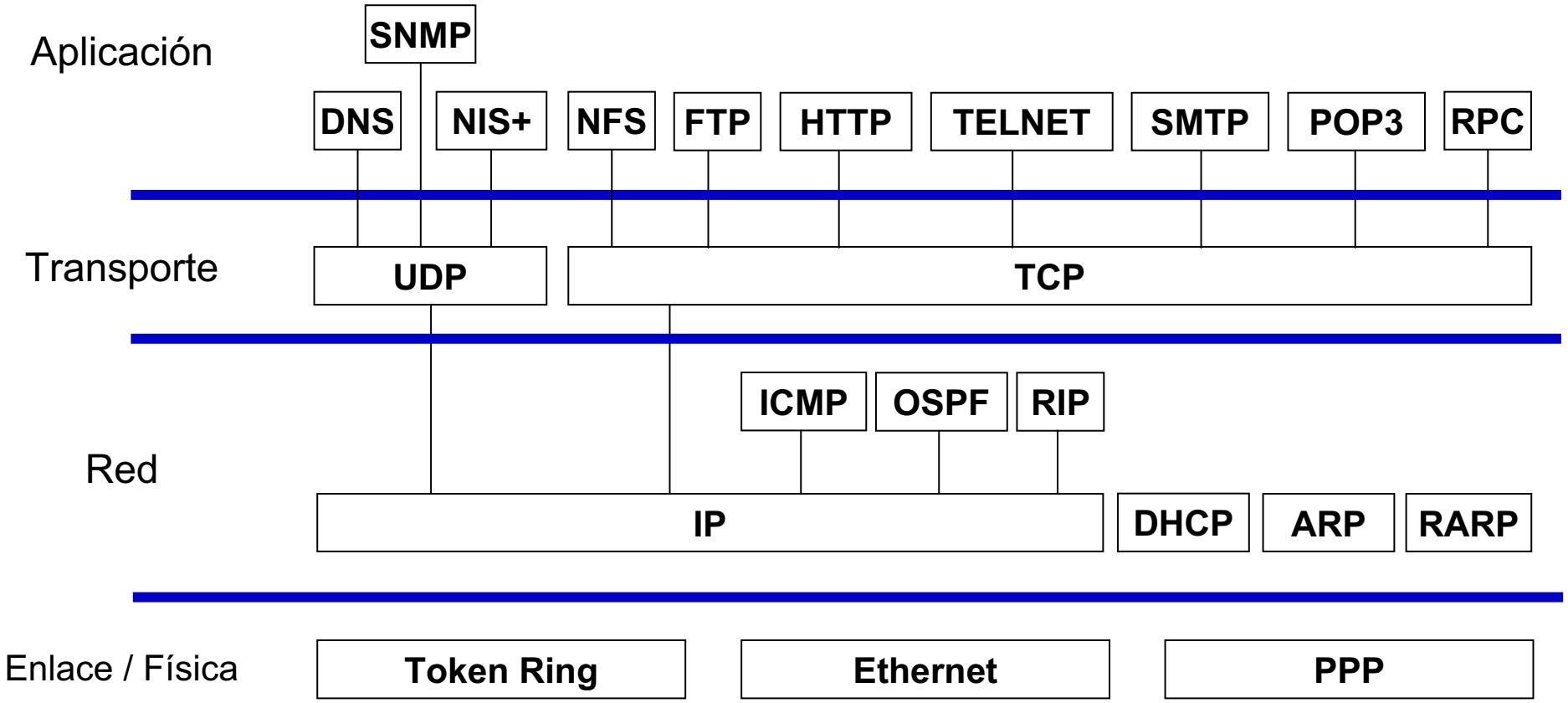
Contenidos

- **Tema 13. Redes de área local y arquitectura de protocolos TCP/IP**
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

Arquitecturas de redes: OSI vs. TCP/IP



Protocolos de la arquitectura TCP/IP



Contenidos

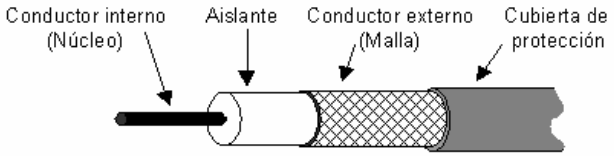
- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- **Tema 14. La interfaz de red Ethernet**
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

Redes LAN Ethernet

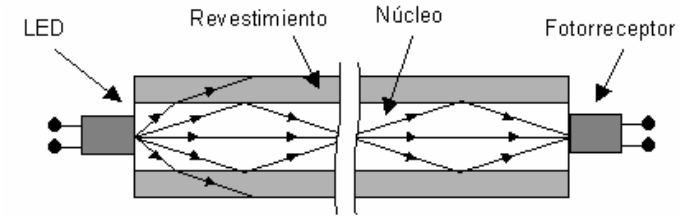
- Características generales
  - Desarrollado originalmente por Xerox
  - Estandarizado por el IEEE 802.3
  - Método de acceso al medio: CSMA/CD
- Distintos medios físicos

Par Trenzado

Cable Coaxial

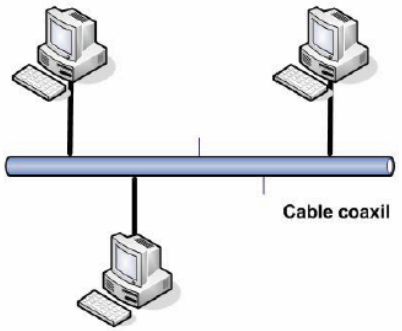


Fibra óptica

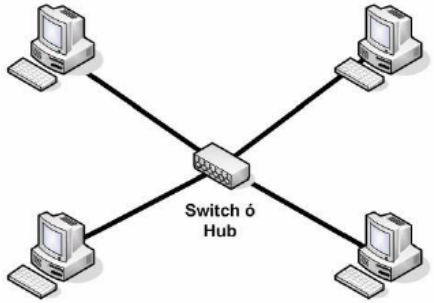


- Distintas Topologías

Bus



Estrella (HUB o SWITCH)



- Evolución de redes Ethernet
  - **Ethernet** → 10 Mbps
  - **Fast Ethernet** → 100 Mbps
  - **Gigabit Ethernet** → 1Gbps

Rafael Moreno (UCM) y Juan Carlos Fabero (UCM)

Protocolo CSMA/CD en Ethernet

- Cuando una estación quiere transmitir, debe realizar las siguientes acciones:
  1. La estación escucha el canal
  2. Si el canal está libre
     → transmite inmediatamente
  3. Si el canal está ocupado
     → Se queda escuchando a la espera de que quede libre
     → Cuando queda libre transmite inmediatamente.
  4. Durante la transmisión
     → Sigue escuchando el canal.
     → Si información escuchada ≠ información transmitida ➔ **COLISIÓN**
  5. En caso de colisión
     → Interrumpir inmediatamente transmisión
     → Envíar señal de invalidación a todas las estaciones
     → Iniciar mecanismo de contienda para retransmitir

Protocolo CSMA/CD (cont.)

Mecanismo contienda (en caso de colisión)

- Generar un número entero aleatorio *m*
- Esperar ranuras temporales antes de intentar retransmitir la trama

1ª colisión: m $\in$ [0,1]

2ª colisión: m [0,1,2,3]

...

n-ésima colisión: m [0,1,...,2n-1]

...

10ª colisión: m [0,1,...,1023]

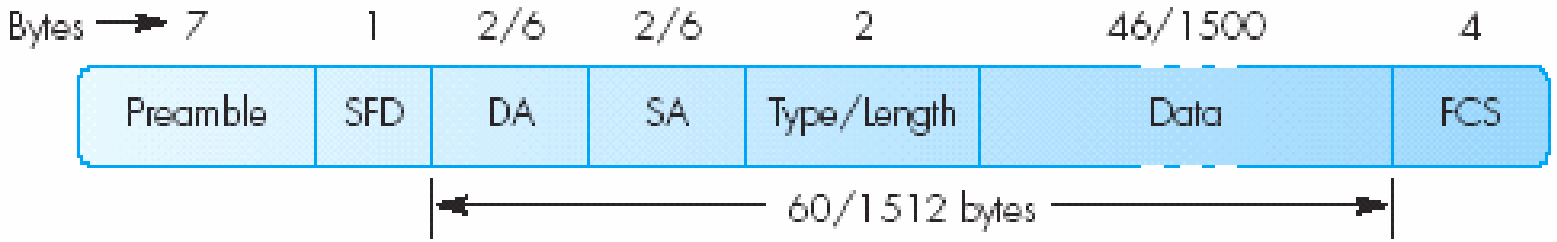
11ª colisión: m [0,1,...,1023]

...

16ª colisión: m [0,1,...,1023]

17ª colisión: ERROR

Formato de trama Ethernet



SFD = Delimitador de inicio de trama
FCS = Secuencia de comprobación de trama

DA = Dirección MAC destino
SA = Dirección MAC fuente

Velocidad de Transmisión →	10 Mbps (Manchester encoded)
Ranura Temporal →	512 bit times
Espacio intertrama →	9.6 microseconds
Límite de intentos →	16
Límite de espera →	10
Tamaño de señal de interferencia →	32 bits
Tamaño máximo de trama (incluyendo FCS) →	1518 bytes
Tamaño mínimo de trama (incluyendo FCS) →	512 bits

Implementaciones físicas de redes Ethernet

Ethernet

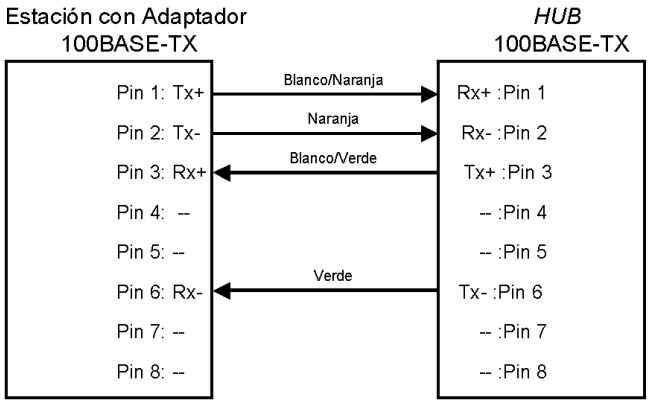
	10BASE5	10BASE2	10BASE-T
Medio Transm.	Coaxial grueso	Coaxial delgado	UTP cat. 3
Topología	Bus	Bus	Estrella con HUB
Modo de transmisión	Half-duplex	Half-duplex	Half-duplex
Velocidad	10 Mbps	10 Mbps	10 Mbps
Longitud máx.	500	185	100

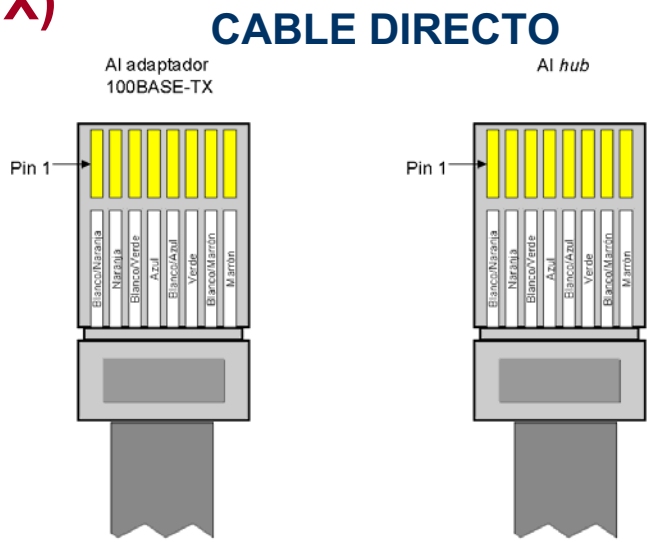
Fast Ethernet

	100BASE-TX	100BASE-FX	100BASE-T4
Medio Transm.	2 pares UTP Cat 5	2 Fibras ópticas multimodo	4 pares UTP Cat 3
Topología	Estrella (HUB o Switch)	Estrella (HUB o Switch)	Estrella (HUB)
Modo de transmisión	HUB ➔ Half-duplex SWITCH ➔ Full-duplex	HUB ➔ Half-duplex SWITCH ➔ Full-duplex	Half-duplex
Velocidad	100 Mbps	100 Mbps	100 Mbps
Longitud máx.	100 m	Hasta 2000 m	100 m

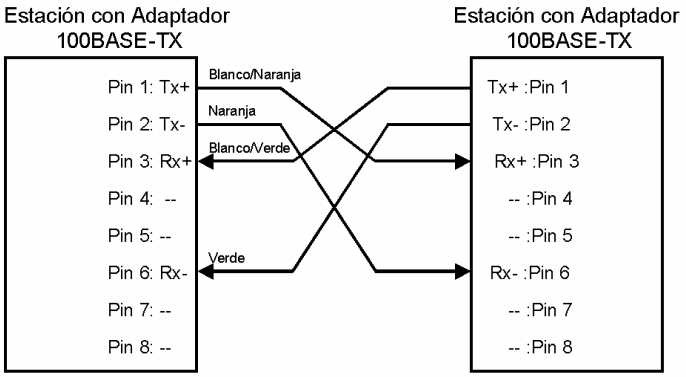
Cableado Ethernet (10BaseT y 100BaseTX)

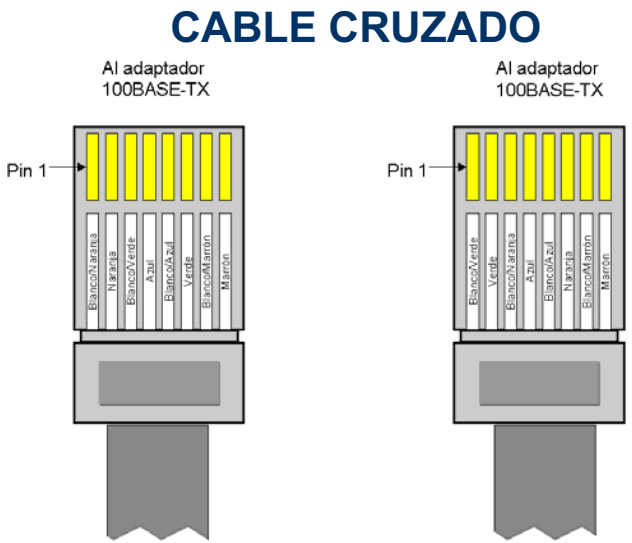
Conexión Computador – Hub





Conexión Computador-Computador





- PRACTICAS (1)
  - Uso de la orden **`ifconfig`** para configurar la interfaz de red
    - Configuración de la interfaz de red eth0

      `ifconfig eth0 192.168.1.<puesto> up`
  - Uso de la orden **`ifconfig`** para visualizar la configuración de la red
    - Uso de las siguientes órdenes

      `ifconfig -a`

      `ifconfig lo`

      `ifconfig eth0`
    - Interpretar salida de de ***ifconfig***

```
eth0      Link encap:Ethernet  HWaddr 00:C0:26:A0:9B:FF
          inet addr:147.96.80.200  Bcast:147.96.80.255
                Mask:255.255.255.0
          inet6 addr: fe80::2c0:26ff:fea0:9bff/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4928981 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6992 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:675432448 (644.1 Mb)  TX bytes:918369 (896.8 Kb)
          Interrupt:9 Base address:0xe800
```

PRACTICAS (2)

Interpretar la salida de la orden *ifconfig*

- Datos básicos

HWaddr	Dirección MAC/Hardware/Ethernet/Física
inet addr	Dirección IPv4
Bcast	Dirección de broadcast (difusión)
Mask	Máscara de red
inet6 addr	Dirección IPv6
MTU	Unidad máxima de transferncia (Ethernet)

- Estadísticas

RX packets	Paquetes recibidos
TX packets	Paquetes enviados
Errors	Paquetes erróneos (CRC erróneo)
Dropped	Paquetes descartados por falta de memoria (buffer lleno)
Overruns	Paquetes perdidos por no poder atender la interrupción (flujo de paquetes superior al soportado)
Frame	Paquetes con formato de trama inválido
Carrier	Paquetes no enviados por falta de señal portadora (cable desconectado)
Collisions	Nº de colisiones sufridas

- PRACTICAS (3)
  - Cambiar la configuración de Ethernet con la orden *ifconfig*
    - Cambiar la MTU

      ```
      ifconfig eth0 down
      ifconfig eth0 mtu 1000
      ifconfig eth0 up
      ```

    - Cambiar la dirección Ethernet

      ```
      ifconfig eth0 down
      ifconfig eth0 hw ether 00:11:22:33:44:55
      ifconfig eth0 up
      ```

  - Visualización del tráfico con ETHEREAL
    - Ejecutar como superusuario (root): `ethereal &`

      Si no funciona entorno gráfico ejecutar las órdenes: `DISPLAY=:0.0; export DISPLAY`
    - Para generar tráfico usar `ping <destino>`
  - Comprobar las estadísticas de colisión
    - Para generar carga de tráfico elevada usar la siguiente orden

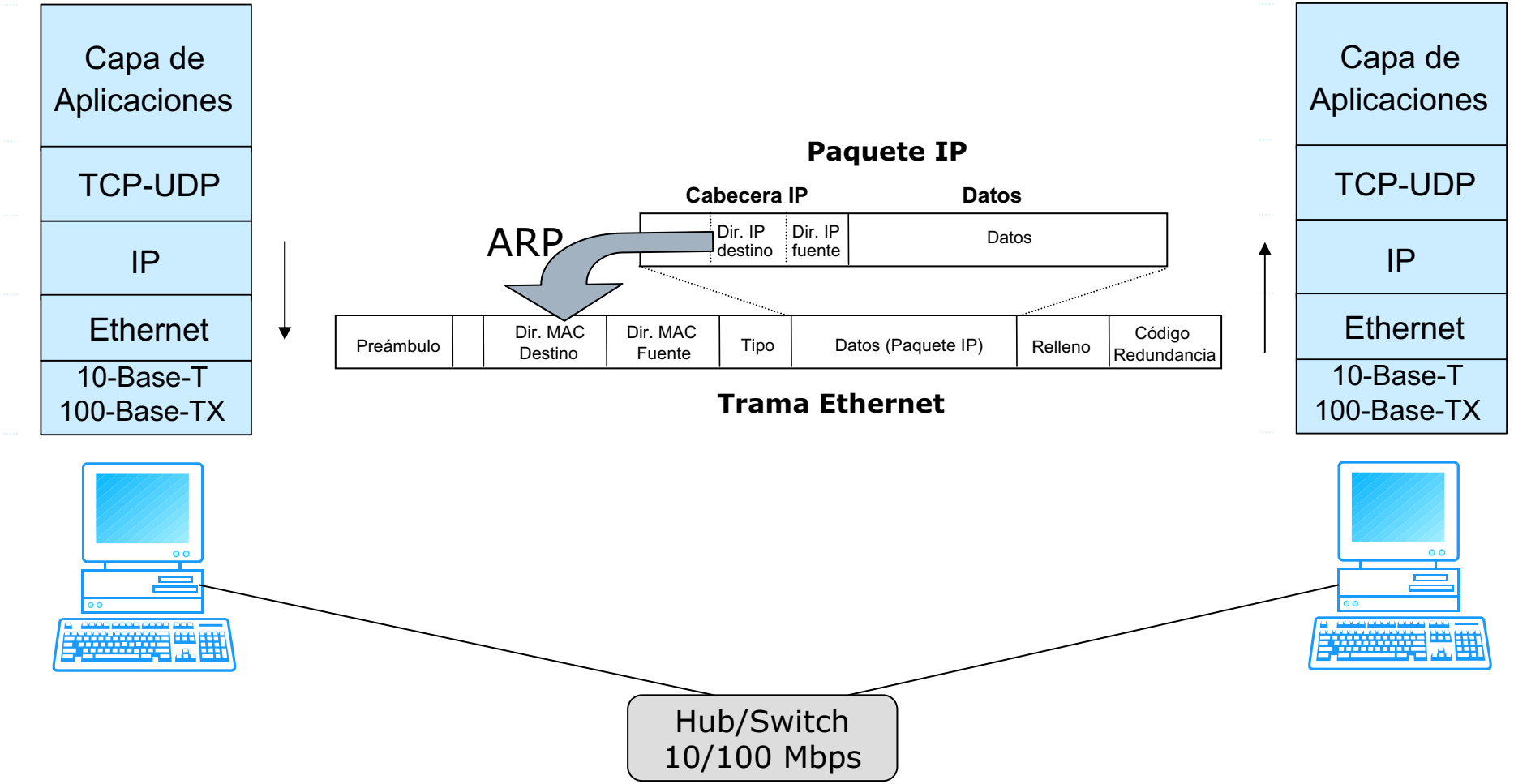
      `ping -f -b <dir_broadcast>`

Contenidos

- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- **Tema 15. Protocolo de resolución de direcciones (ARP)**
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

ARP: Address Resolution Protocol

- Traducción: Dirección IP → Dirección MAC



Tema 15. Protocolo de resolución de direcciones (ARP)

El protocolo ARP (Address Resolution Protocol)

La tabla ARP

- Mantiene las direcciones IP de las últimas máquinas con las que nos hemos comunicado y las direcciones Ethernet asociadas
- Ver la tabla arp: orden **`arp -a`**
- Ejemplo de tabla ARP

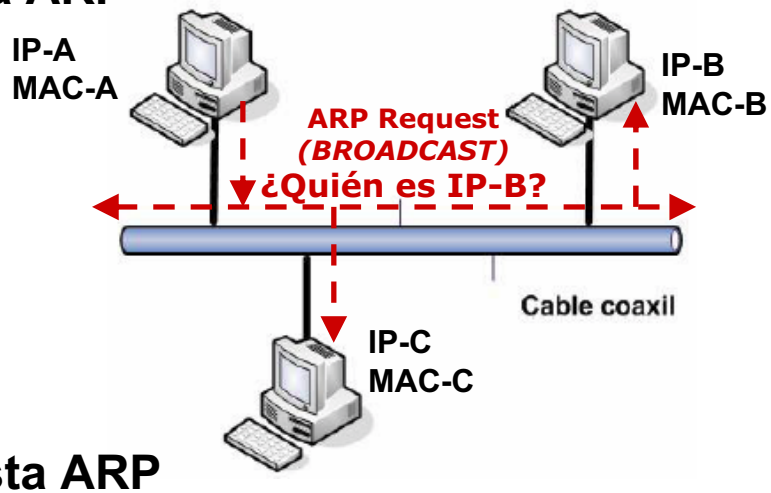
```
Net to Media Table
Device   IP Address               Mask      Flags   Phys Addr
------ -------------------- --------------- ----- ---------------
le0    147.96.48.203        255.255.255.255       00:00:b4:c3:c8:f4
le0    147.96.37.196        255.255.255.255       00:a0:24:57:78:3e
le0    147.96.48.217        255.255.255.255       00:20:18:2f:1d:60
```

Funcionamiento de ARP

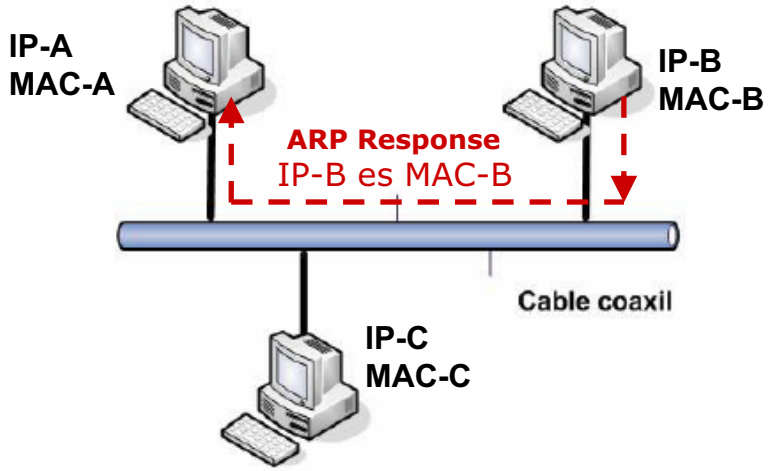
- Si Host A quiere enviar un paquete a Host B
  1. Host A consulta su tabla ARP para ver si la dirección MAC de Host B está contenida en dicha tabla
  2. Si la dirección MAC de Host B no está en la tabla entonces envía un mensaje ***broadcast*** preguntando por la dirección MAC de Host B → **ARP Request**
  3. Host B responde a Host A informándole de su dirección IP → **ARP Response**

Esquema de Funcionamiento de ARP

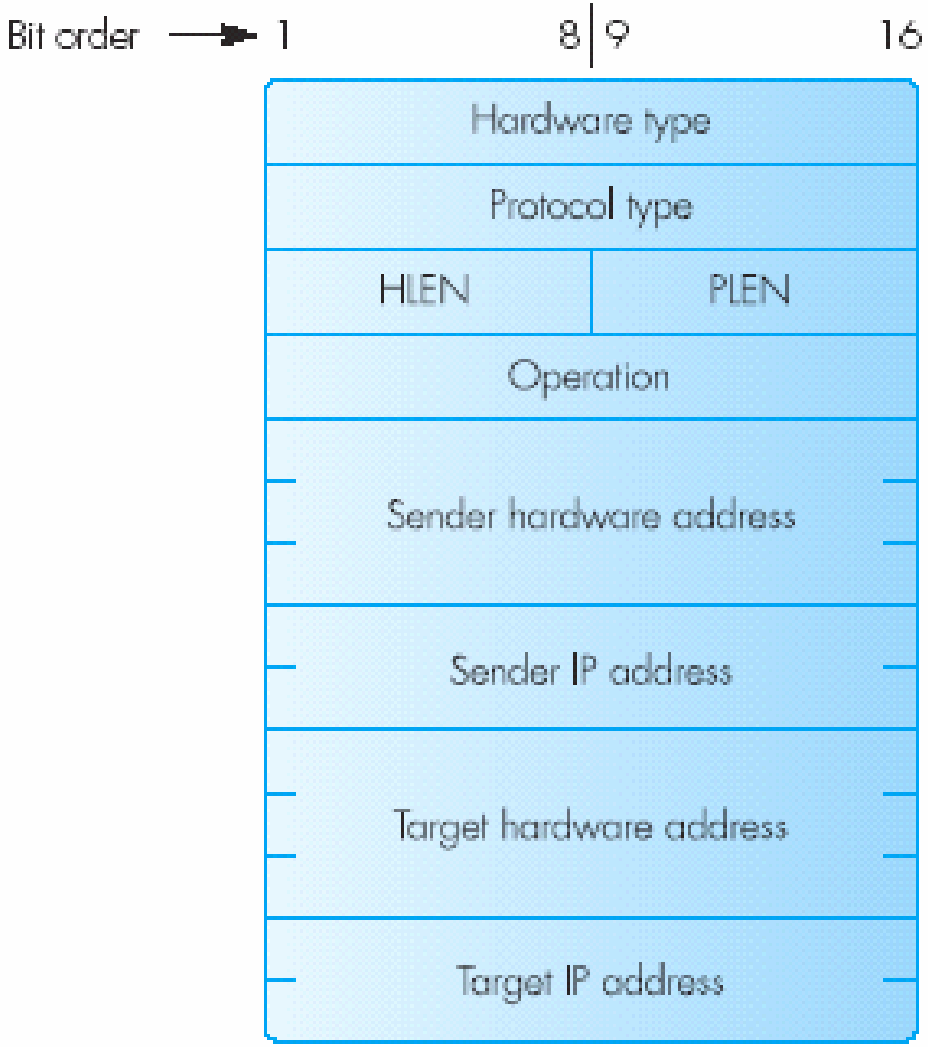
- Host A quiere enviar paquete a Host B
  - **Pregunta ARP**



  - **Respuesta ARP**



Formato de paquetes ARP



HLEN = Hardware address length
PLEN = IP address length
Operation = 1 ARP request
= 2 ARP response
= 3 RARP request
= 4 RARP response

Ejemplos de paquetes ARP (i)

Pregunta ARP

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Destination = ff:ff:ff:ff:ff:ff, (broadcast)
ETHER:  Source      = 8:0:20:88:c9:ee, Sun
ETHER:  Ethertype = 0806 (ARP)
ETHER:
ARP:  ----- ARP/RARP Frame -----
ARP:
ARP:  Hardware type = 1
ARP:  Protocol type = 0800 (IP)
ARP:  Length of hardware address = 6 bytes
ARP:  Length of protocol address = 4 bytes
ARP:  Opcode 1 (ARP Request)
ARP:  Sender's hardware address = 8:0:20:88:c9:ee
ARP:  Sender's protocol address = 147.96.21.31
ARP:  Target hardware address = ?
ARP:  Target protocol address = 147.96.21.120
```

Ejemplos de paquetes ARP (ii)

Respuesta ARP

```
ETHER:  ----- Ether Header -----
ETHER:
ETHER:  Destination = 8:0:20:88:c9:ee, Sun
ETHER:  Source      = 0:3:ba:d:e7:e,
ETHER:  Ethertype = 0806 (ARP)
ETHER:
ARP:  ----- ARP/RARP Frame -----
ARP:
ARP:  Hardware type = 1
ARP:  Protocol type = 0800 (IP)
ARP:  Length of hardware address = 6 bytes
ARP:  Length of protocol address = 4 bytes
ARP:  Opcode 2 (ARP Reply)
ARP:  Sender's hardware address = 0:3:ba:d:e7:e
ARP:  Sender's protocol address = 147.96.21.120
ARP:  Target hardware address = 8:0:20:88:c9:ee
ARP:  Target protocol address = 147.96.21.31
```

Tema 15. Protocolo de resolución de direcciones (ARP)

- **PRACTICAS**
  - **Visualizar paquetes ARP**

    Usar `ethereal`
  - **Visualizar la tabla ARP**

    Orden `arp -a`
  - **Manipular de forma manual la tabla ARP**
    - Borrar una entrada de la tabla ARP

      `arp -d <dir_IP>`
    - Añadir una entrada permanente a la tabla ARP (ver flag TEMP)

      `arp -s <dir_IP> <dir_ETHER>`
    - Añadir una entrada temporal a la tabla ARP (ver flag TEMP)

      `arp -s <dir_IP> <dir_ETHER> temp`
  - **ARP "poisoning"**
    - Añadir a la tabla ARP una entrada con dirección ethernet errónea

      `arp -s <dir_IP> <dir_ETHER_ERRONEA>`
    - Comprobar posteriormente que la dirección IP es innacesible

Contenidos

- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- **Tema 16. Configuración de IPv4. Redes y subredes**
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

Tema 16. Configuración de IPv4. Redes y subredes

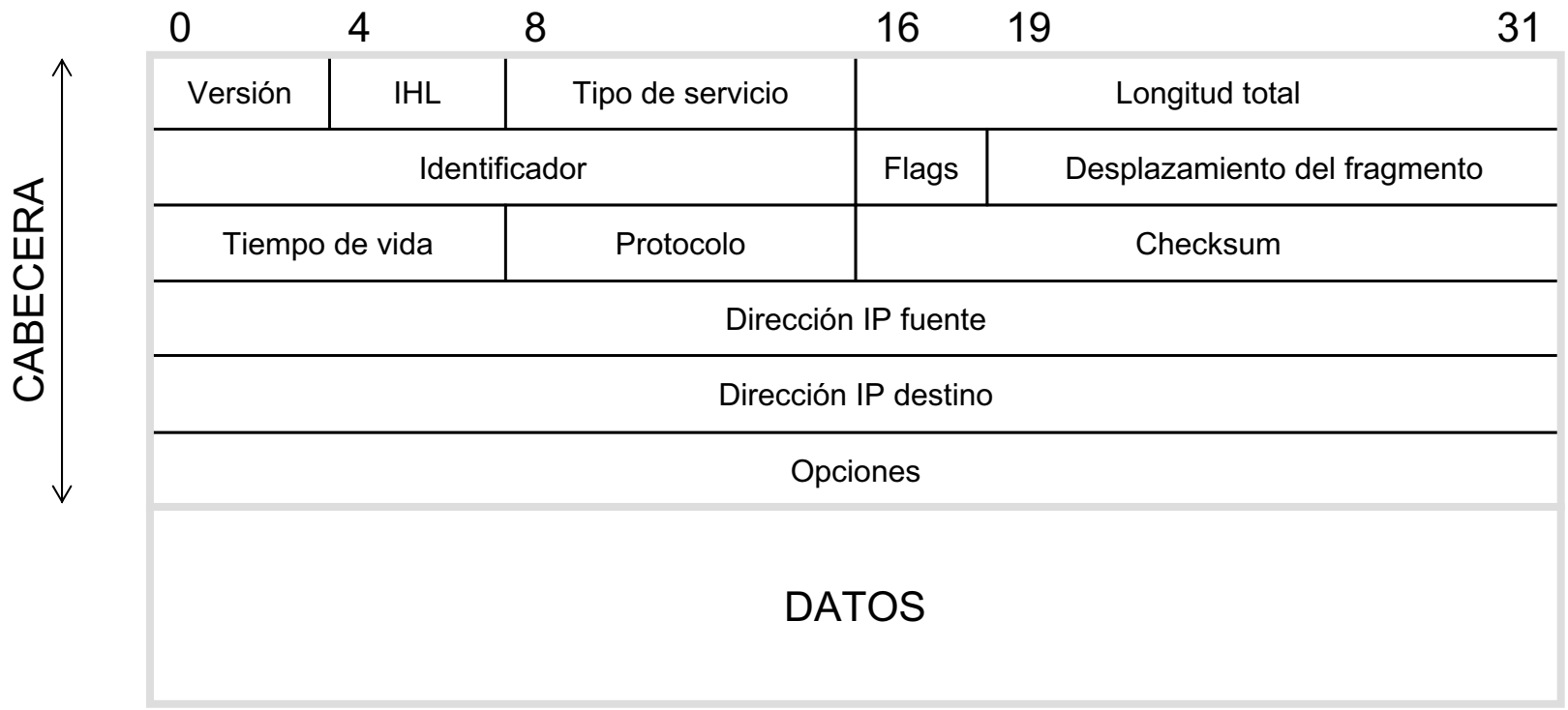
Protocolo de red de Internet

- Proporciona un servicio básico de entrega de paquetes
  - Sobre el que se construyen las redes TCP/IP.
- Protocolo **no orientado a conexión** (no fiable)
  - No realiza detección ni recuperación de paquetes perdidos o erróneos
  - No garantiza que los paquetes lleguen en orden
  - No garantiza la detección de paquetes duplicados

Funciones básicas del protocolo IP

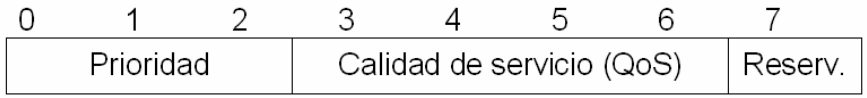
- Direccionamiento
  - Esquema global de direccionamiento
- Fragmentación y reensamblaje de paquetes
  - División del paquetes en fragmentos de un tamaño aceptable por la red
- Encaminamiento de datagramas
  - Encaminado de paquetes atendiendo a información de tabla de rutas
  - La construcción de tablas de rutas puede ser
    - Manual (routing estático)
    - Mediante algún protocolo de routing diñámico: RIP, OSPF, BGP, etc.

Formato del paquete IP (datagrama)



Campos de la cabecera IP (i)

- Versión
  - Valor=4 (IPv4)
- IHL
  - Longitud de la cabecera, en palabras de 32 bits.
- Tipo de servicio



  - Prioridad
    - Especifica la prioridad del datagrama (hasta 8 niveles).
    - Un paquete de alta prioridad debe ser reexpedido por un router antes que un paquete de baja prioridad (aunque este llegase antes)
  - QoS: Puede tomar los siguientes valores
    - 1000 → Minimizar retardo
    - 0100 → Maximizar rendimiento (velocidad de transmisión)
    - 0010 → Maximizar fiabilidad (seguridad en la entrega)
    - 0001 → Minimizar coste monetario
    - 0000 → Servicio normal

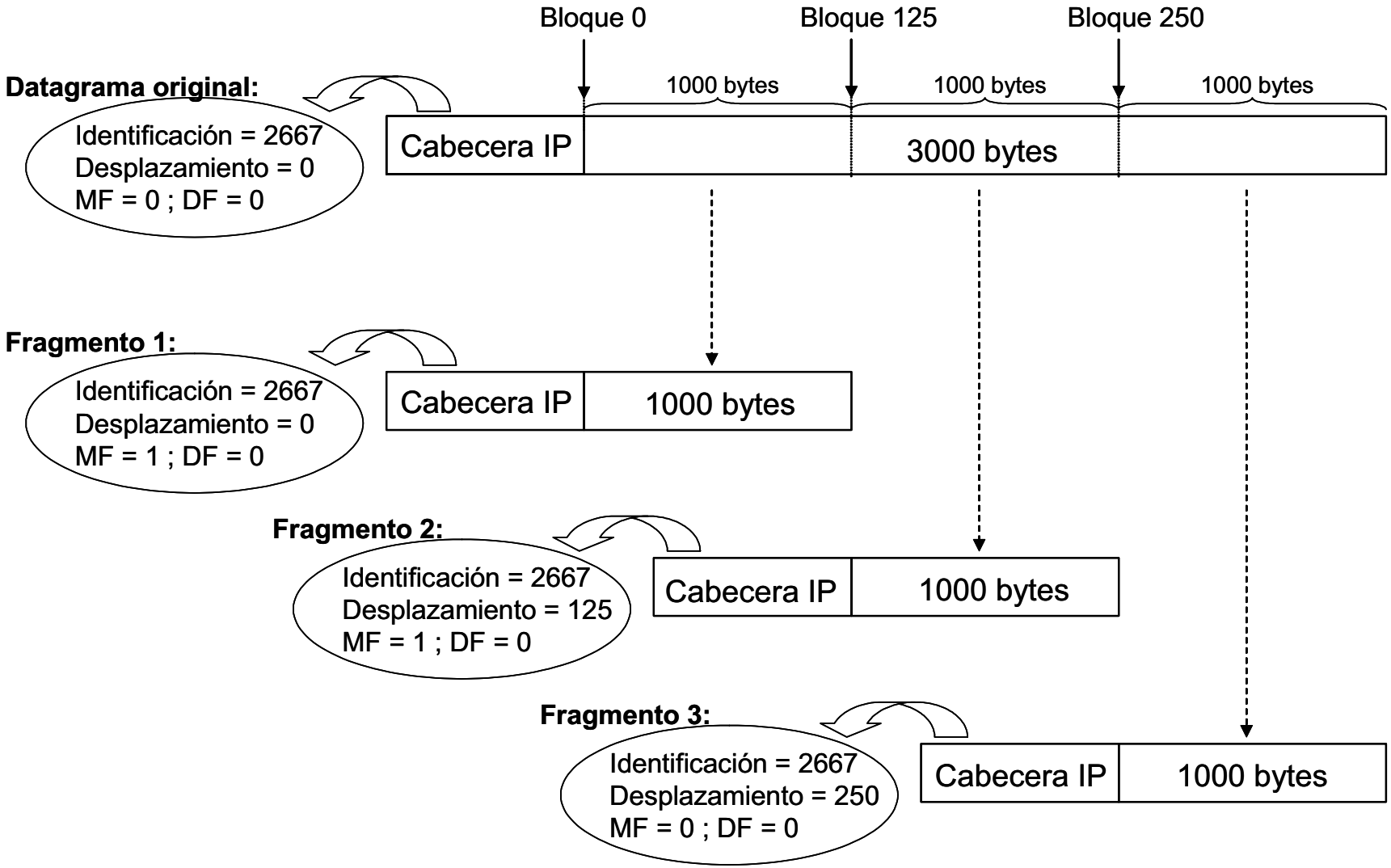
Campos de la cabecera IP (ii)

- Longitud total
  - Longitud del datagrama (cabecera + datos) medida en bytes.
- Identificador
  - Número de 16 bits que identifica al datagrama
- Flags
  - MF (More Fragments): si está a 1 indica que no es el último fragmento
  - DF (Don´t Fragment): si es 1 prohíbe la fragmentación
- Desplazamiento del fragmento
  - Nº secuencia del fragmento (unidades = 8 bytes)
- Tiempo de vida (TTL, *Time To Tive*)
  - Nº encaminadores que puede atravesar el paquete
  - Cuando TTL=0 el paquete debe ser descartado

Campos de la cabecera IP (iii)

- Protocolo
  - Protocolo de la capa superior al que deben entregarse los datos
  - Ejemplos
    - 1: Internet Control Message Protocol (ICMP)
    - 2: Internet Group Management Protocol (IGMP)
    - 6: Transmission Control Protocol (TCP)
    - 8: Exterior Gateway Protocol (EGP)
    - 17: User Datagram Protocol (UDP)
    - 41: IP Version 6 (IPv6)
    - 89: Open Shortest Path First (OSPF)
- Checksum
  - Suma de control de la cabecera
- Direcciones IP origen y destino
  - Identifican al host emisor y receptor del paquete
- Opciones
  - Campo opcional, con opciones especiales
  - Ejemplos: encaminamiento de origen, sello de ruta, sello de tiempo, etc.

Ejemplo de fragmentación



Tema 16. Configuración de IPv4. Redes y subredes

Direcciones IPv4

- Las direcciones IP constan de 4 bytes (32 bits)
- Para expresarlas se utiliza la “notación de punto”
  - Ejemplo: 128.2.7.9 = 10000000 . 00000010 . 00000111 . 00001001

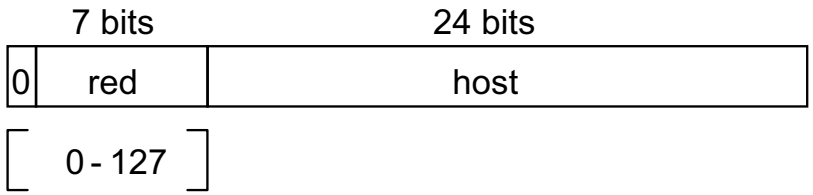
Tipos de direcciones IPv4

- Unicast
  - Un único host
- Multicast
  - Un grupo de hosts
- Broadcast
  - Todos los hosts dentro de mi red local

Tema 16. Configuración de IPv4. Redes y subredes

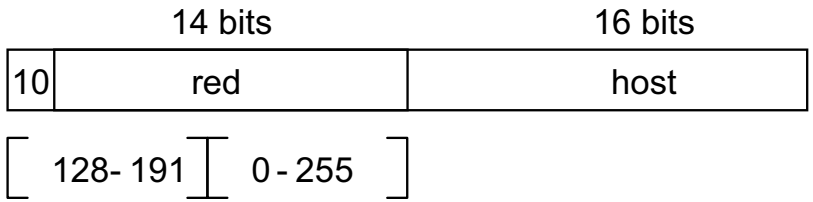
Formato y clases de direcciones IPv4

Clase A



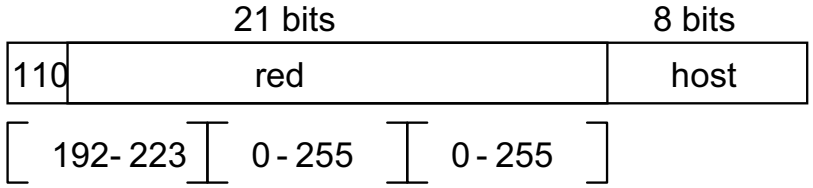
2^7 = 128 redes
2^{24} = 16.777.216 hosts
Ejemplo: ***26***.56.120.9

Clase B



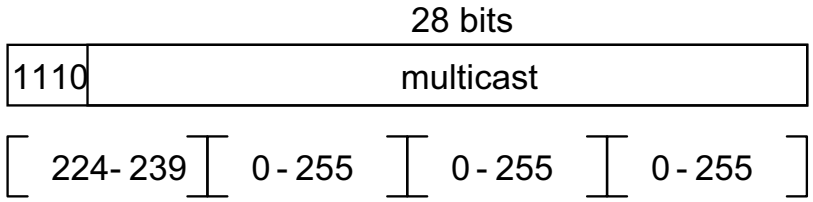
2^{14} = 16.384 redes
2^{16} = 65.536 hosts
Ejemplo: ***147.96***.50.110

Clase C



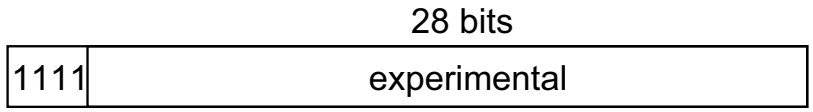
2^{21} = 2.097.152 redes
2^8 = 256 hosts
Ejemplo: ***217.6.95***.44

Clase D



Ejemplo: ***224.0.0.1***

Clase E



Direcciones IP especiales (i)

Direcciones reservadas para redes privadas

- Existen un conjunto de direcciones reservadas para uso privado.
  - Se pueden asignar a redes aisladas de Internet
  - Se pueden asignar redes conectadas a través de un router que hace traducción de direcciones de red (NAT)
- Estas direcciones son las siguientes:
  - 10.0.0.0 – 10.255.255.255 → 1 red privada de clase A
  - 172.16.0.0 – 172.31.255.255 → 16 redes privadas de clase B
  - 192.168.0.0 – 192.168.255.255 → 256 redes privadas de clase C

Direcciones de loopback (127.x.y.z)

- Direcciones de bucle interno (loopback)
- Casi todas las máquinas tienen como dirección de loopback la **127.0.0.1**

Direcciones broadcast (terminadas en 255)

- La dirección broadcast se utilizan para enviar un paquete a todas las máquinas de la red local
- Ejemplos:
  - Red clase A: 27.255.255.255
  - Red Clase B: 142.88.255.255
  - Red Clase C: 199.67.239.255
  - Dirección broadcast universal: 255.255.255.255

Direcciones IP especiales (ii)

Direcciones de red

- Son direcciones terminadas en 0
- Ejemplos:
  - Red clase A: 27.0.0.0
  - Red Clase B: 142.88.0.0
  - Red Clase C: 199.67.239.0
- Nunca se utilizan como dirección destino ni se asignan a un host concreto
  - Se utilizan únicamente en las tablas de encaminamiento

- Ejemplo de tabla de rutas en Linux (orden **`netstat -nr`**)

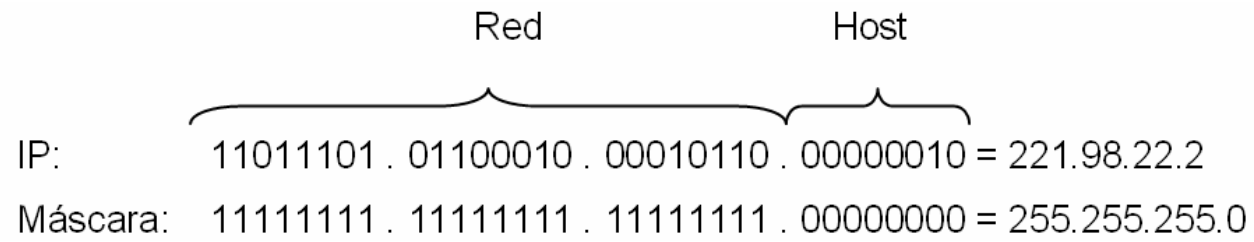
```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         eth0
192.168.2.0     0.0.0.0         255.255.255.0   U         eth1
0.0.0.0         192.168.1.1     255.255.255.0   UG        eth0
```

Tema 16. Configuración de IPv4. Redes y subredes

Máscaras de red

La máscara de red indica:

- Qué bits de la dirección IP identifican a la red (bits de la máscara a 1)
- Qué bits identifican al host dentro de la red (bits de la máscara a 0)
- Ejemplo
  - Dirección de clase C: 221.98.22.2
  - Máscara: 255.255.255.0



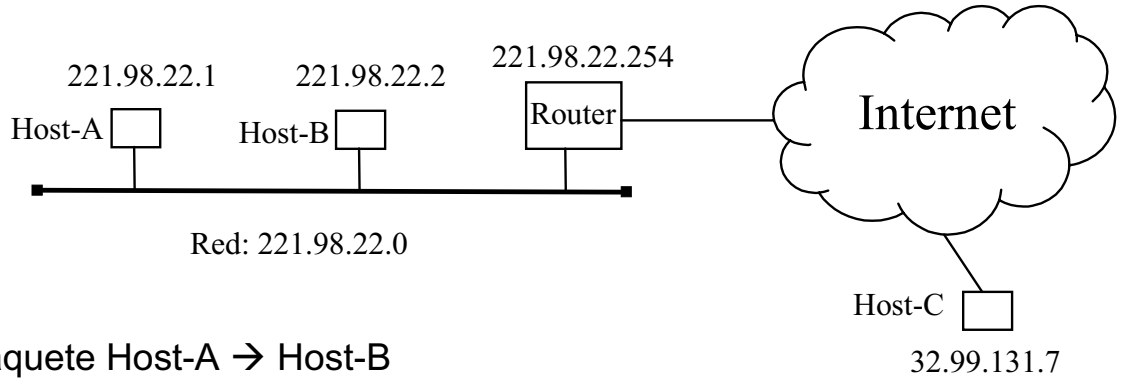
  - Notación alternativa: 221.98.22.2/24

Máscaras típicas de red

- Red de la clase A: 255.0.0.0 (equivalente a /8)
- Red de la clase B: 255.255.0.0 (equivalente a /16)
- Red de la clase C: 255.255.255.0 (equivalente a /24)

Máscaras de red y tablas de encaminamiento

Ejemplo



- Enviar paquete Host-A → Host-B
  - Host-A envía paquete directamente a través de su red local
- Enviar paquete Host-A → Host-C
  - Host-A envía el paquete al router y este se encargará de encaminarlo hasta su destino
- Para saber como tiene que tratar el paquete, el Host-A tiene que realizar las siguientes operaciones:
  - Aplicar la máscara de red a la dirección destino
    - convierte la dirección del host destino en una dirección de red
  - Consultar la tabla de encaminamiento
    - decide a quien debe entregar el paquete (host destino o router)
- La máscara de red es en nuestro caso el siguiente valor:
  - 255.255.255.0

Máscaras de red y tablas de encaminamiento

Ejemplo (cont)

- Aplicación de la máscara: realizar Y-lógica bit a bit entre la dirección destino y la máscara:

Dirección del Host-B

```
221.98.22.2      = 11011101 . 01100010 . 00010110 . 00000010
255.255.255.0    = 11111111 . 11111111 . 11111111 . 00000000
221.98.22.0      = 11011101 . 01100010 . 00010110 . 00000000
```

Dirección del Host-C

```
32.99.131.7      = 00100000 . 01100010 . 10000011 . 00000111
255.255.255.0    = 11111111 . 11111111 . 11111111 . 00000000
32.99.131.0      = 00100000 . 01100010 . 10000011 . 00000000
```

- El Host-A consultar su tabla de encaminamiento (orden **netstat -nr**)

```
Destination           Gateway
-------------------- -------------
221.98.22.0          0.0.0.0
0.0.0.0 (default)    221.98.22.254
127.0.0.1            127.0.0.1
```

  - Si la dirección de Gateway asociada es el propio remitente (0.0.0.0), el paquete se debe enviar directamente a través de la red local
  - Si la dirección de Gateway asociada es la de un router, se usa ese router para enviar el paquete a su destino
  - Si la dirección no aparece en la tabla se usa el **Default Router** para enviar el paquete a su destino

EJERCICIOS (1)

- Visualizar paquetes IP con **ethereal**

- Configurar manualmente una dirección IP de clase B y la máscara de red de clase B
  - Asignar la siguiente dirección/máscara:

```
ifconfig eth0 172.16.<fila>.<puesto> netmask 255.255.0.0
          broadcast 172.16.255.255
```

  - Visualizar el tráfico con Etherreal
  - Hacer ping a otras máquina de la misma o distintas filas (observar que todas las máquinas se ven)
  - Visualizar tabla de rutas

```
netstat -nr
```

```
Kernel IP routing table
Destination     Gateway   Genmask         Flags   Iface
172.16.0.0      0.0.0.0   255.255.0.0      U       eth0
```

Tema 16. Configuración de IPv4. Redes y subredes

EJERCICIOS (2)

- Configurar manualmente una dirección IP de clase C y la máscara de red de clase C
  - Asignar la siguiente dirección/máscara:

```
ifconfig eth0 192.168.<fila>.<puesto> netmask 255.255.255.0 \
          broadcast 192.168.<fila>.255
```

  - Visualizar el tráfico con Etherreal
  - Hacer ping a otras máquina de la misma o distintas filas (observar que todas las máquinas de distintas filas no se ven)
  - Visualizar tabla de rutas

    netstat –nr (ejemplo de tabla de rutas de una máquina en la **fila 5**)

```
Kernel IP routing table
Destination     Gateway   Genmask         Flags   Iface
192.168.5.0     0.0.0.0   255.255.255.0   U        eth0
```

Tema 16. Configuración de IPv4. Redes y subredes

- **EJERCICIOS (3)**
  - Configuración de un router (sólo máquina del profesor)
    - Activar interfaces virtuales (una por cada fila/subred)

```
ifconfig eth0 192.168.1.100 netmask 255.255.255.0 up
ifconfig eth0:1 192.168.2.100 netmask 255.255.255.0 up
. . . .
ifconfig eth0:4 192.168.5.100 netmask 255.255.255.0 up
```

    - Activar forwarding

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

  - Configuración del router predeterminado (default) en todos los sistemas
    - Añadir router predeterminado a la tabla de rutas:

```
route add default gw 10.10.<fila>.100
```

    - Visualizar tabla de rutas (ejemplo de máquina en la fila 5)

```
netstat -nr
```

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Iface
192.168.5.0     0.0.0.0         255.255.255.0   U         eth0
0.0.0.0         192.168.5.100   255.255.255.0   UG        eth0
```

EJERCICIOS (4)

Configuración permanente de la red

- Archivo de configuración de la red (ejemplo de máquina 192.168.5.1)

  /etc/network/interfaces

```
auto lo eth0
   iface eth0 inet static
   address 192.168.5.1
   netmask 255.255.255.0
   network 192.168.5.0
   broadcast 192.168.5.255
   gateway 192.168.5.100
```

Tema 16. Configuración de IPv4. Redes y subredes

Organización de redes en subredes

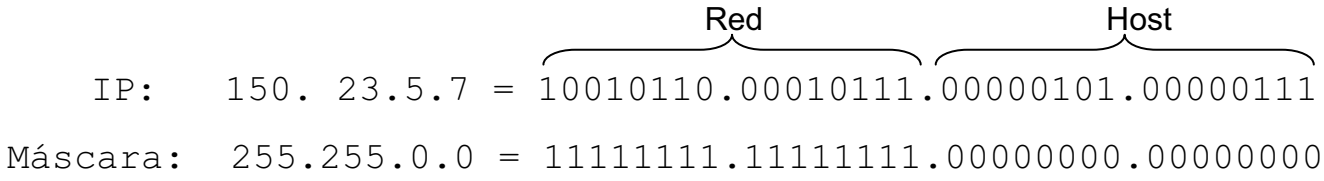
Ventajas de las subredes

- Permite aislar el tráfico entre las distintas subredes
  - Se reduce el tráfico global
- Permite limitar y proteger el acceso a las distintas subredes
  - La comunicación entre éstas se realiza mediante un router
- Permite organizar la red en áreas o departamentos
  - Se asigna a cada departamento un subconjunto de direcciones IP
  - La gestión de las direcciones IP se puede delegar en el propio área o departamento
    - Se descentraliza la tarea de asignación de direcciones
    - Se facilita la tarea del administrador de la red

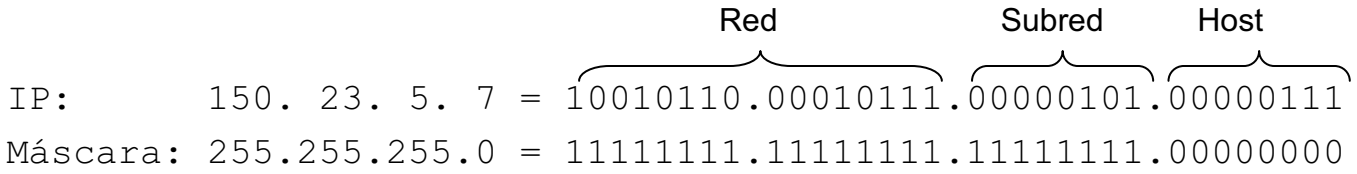
Organización de redes en subredes

Ejemplo 1

- Supongamos la red de la clase B: 150.23.0.0
  - Tenemos 16 bits para identificar a host (2^{16} hosts)



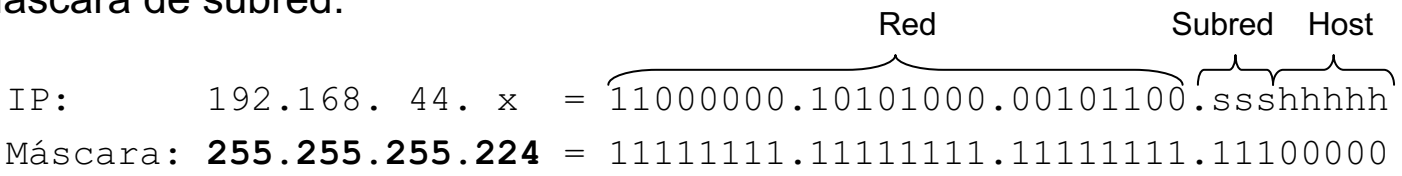
- Esta red se puede dividir, por ejemplo, en 256 subredes con 256 hosts cada una
  - Usamos 8 bits para identificar a la subred (2^8 = 256 subredes)
  - Usamos 8 bits para identificar a host (2^8 = 256 hosts)
- Nos queda la siguiente organización:
  - Subred 0: 150.23.0.0 (Dpto. de administración)
  - Subred 1: 150.23.1.0 (Dpto. de RRHH)
  - .....
  - Subred 255: 150.23.255.0 (Dpto. comercial)
- Por tanto la máscara de subred adecuada es la siguiente:



Organización de redes en subredes

Ejemplo 2

- Supongamos la red de la clase C: 192.168.44.0
- Queremos dividir la red en 8 subredes
  - 3 bits para identificar la subred (2^3 = 8 subredes)
  - 5 bits para identificar el host (2^5 = 32 hosts por subred)
- Máscara de subred:



- Organización resultante:

Subred **192.168.44.0**
- hosts: de **192.168.44.1** al **192.168.44.30**
- broadcast : **192.168.44.31**

Subred **192.168.44.32**
- hosts: de **192.168.44.33** al **192.168.44.62**
- broadcast : **192.168.44.63**

Subred **192.168.44.64**
- hosts: de **192.168.65.** al **192.168.44.94**
- broadcast: **192.168.44.95**

Subred **192.168.44.96**
- hosts: de **192.168.44.97** al **192.168.44.126**
- broadcast: **192.168.44.127**

Subred **192.168.44.128**
- hosts: de **192.168.44.129** al **192.168.44.158**
- broadcast: **192.168.44.159**

Subred **192.168.44.160**
- hosts: de **192.168.44.161** al **192.168.44.190**
- broadcast: **192.168.44.191**

Subred **192.168.44.192**
- hosts: de **192.168.44.193** al **192.168.44.222**
- broadcast: **192.168.44.223**

Subred **192.168.44.224**
- hosts: de **192.168.44.225** al **192.168.44.254**
- broadcast: **192.168.44.255**

Tema 16. Configuración de IPv4. Redes y subredes

- **PRACTICAS**
  - Configurar las máquinas del aula en subredes según el ejemplo anterior
    - Fila 1 (**subred 192.168.44.33**)

```
ifconfig eth0 192.168.44.33 netmask 255.255.255.224 broadcast 192.168.44.63
ifconfig eth0 192.168.44.34 netmask 255.255.255.224 broadcast 192.168.44.63
etc.
```

    - Fila 2 (**subred 192.168.44.64**)

```
ifconfig eth0 192.168.44.65 netmask 255.255.255.224 broadcast 192.168.44.95
ifconfig eth0 192.168.44.66 netmask 255.255.255.224 broadcast 192.168.44.95
etc.
```

    - Fila 3 (**subred 192.168.44.96**)

```
ifconfig eth0 192.168.44.97 netmask 255.255.255.224 broadcast 192.168.44.127
ifconfig eth0 192.168.44.98 netmask 255.255.255.224 broadcast 192.168.44.127
etc.
```

    - Fila 4 (**subred 192.168.44.128**)

```
ifconfig eth0 192.168.44.129 netmask 255.255.255.224 broadcast 192.168.44.159
ifconfig eth0 192.168.44.130 netmask 255.255.255.224 broadcast 192.168.44.159
etc.
```

    - Fila 5 (**subred 192.168.44.160**)

```
ifconfig eth0 192.168.44.161 netmask 255.255.255.224 broadcast 192.168.44.191
ifconfig eth0 192.168.44.162 netmask 255.255.255.224 broadcast 192.168.44.191
etc.
```

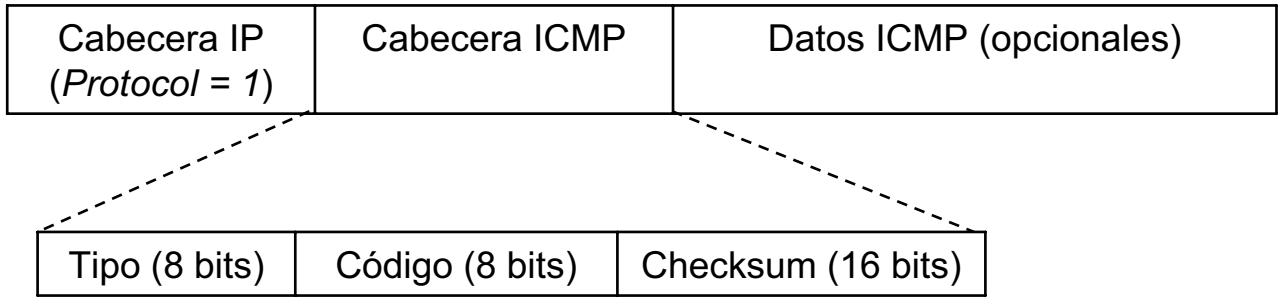
  - Configurar la máquina del profesor como router entre las distintas subredes y añadir este como router predeterminado en todas las máquinas

Protocolo ICMP (Internet Control Message Protocol)

Mensajes ICMP

- Permiten conocer si una máquina es alcanzable (echo request & replay)
- Permiten a los routers informar de mensajes de error en la entrega de paquetes (unreachable destination)
- Permiten redirigir los paquetes (redirect)
- Permiten evitar la saturación de un router (source quench)

Formato paquetes ICMP



Tipos de paquetes ICMP

	Tipo	Significado
	0	Echo Reply
Mensajes Informativos	5	Redirect
	8	Echo Request
	9	Router Solicitation
	10	Router Advertisement
Mensajes de error	3	Destination Unreachable
	4	Source Quench
	11	Time Exceeded
	12	Parameter Problem

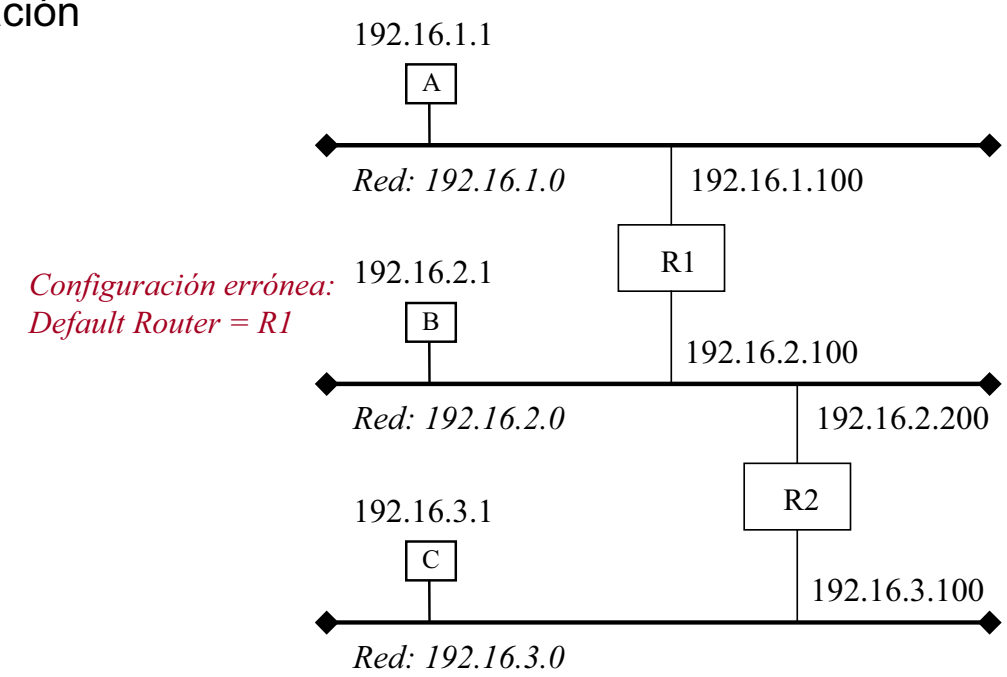
Paquetes ICMP Echo Request y Echo Replay

- Tipo
  - Echo Request: valor 8
  - Echo Reply: valor 0
- Código = 0
- Identificador/Secuencia
  - Permite establecer la correspondencia el Echo Request y el Echo Reply
  - El mensaje Echo Reply contiene siempre el mismo identificador que su correspondiente Echo Request
- Datos
  - Este campo contiene un número de bytes, generados aleatoriamente
- **PRACTICA**
  - Visualizar con **Ethereal** paquetes ICMP echo request / echo reply

Tipo (0/8)	Código (0)	Checksum
Identificador		Nº de secuencia
Datos		

Paquetes ICMP Redirect

- Los mensajes de redirección los envía un router cuando un host no está eligiendo la ruta adecuada hacia un determinado destino
- **PRACTICA**
  - Visualizar con **ethereal** los paquetes **ICMP Redirect** que se generan en la siguiente situación



  - Supongamos que el host B tiene configurado como Default Router a R1
    - Para ir de host B a host C, usará la ruta a través de R1 (ruta incorrecta)
    - R1 devolverá un mensaje ICMP Redirect al host B la ruta correcta al host C

Paquetes ICMP Destination Unreachable

- Subtipos

Código	Significado
0	Network unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and DF set
5	Source route failed
6	Destinantion network unknown
7	Destinantion host unknown

Paquetes ICMP Time Exceeded

- Los envía un router al host origen cuando descarta el paquete por haber agotado su tiempo de vida (**TTL de tránsito**)
- **PRACTICA**
  - Visualizar con **ethereal** los paquetes **ICMP Redirect** que se genera la orden `traceroute`

Contenidos

- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- **Tema 17. Configuración de routers y protocolos de routing**
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

Tema 17. Configuración de routers y protocolos de routing

- **Técnicas de encaminamiento**
  - Encaminamiento estático
    - Consiste en construir manualmente las tablas de encaminamiento, basándonos en el conocimiento de cómo está constituida la red
    - El encaminamiento estático se suele emplear en las tablas de rutas de los hosts que no actúan como routers, ya que suelen ser tablas cortas
  - Encaminamiento dinámico
    - Para la configuración de routers, el encaminamiento estático resulta poco adecuado, por varias razones:
      - Necesidad conocer en detalle la estructura de la red
      - Las tablas de encaminamiento pueden tener un gran número de entradas
      - La estructura de la red puede cambiar, por tanto es necesario actualizar continuamente las tablas

Tema 17. Configuración de routers y protocolos de routing

Interpretación de tablas de rutas

Ejemplo:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         eth0
192.168.2.0     192.168.1.1     255.255.255.0   U         eth1
0.0.0.0         192.168.1.2     255.255.255.0   UG        eth0
```

Campos principales de la tabla de rutas

- **Destination:** Red o host destino
- **Gateway:** El host o router que debe entrega o reexpide el paquete
- **Flags:** Indican el estado de la ruta

  U → La interfaz está activada

  H → El campo Destination representa un Host y no una red

  G →El host de entrega es un router (un camino indirecto)

  D → La ruta es consecuencia de una redirección ICMP

Tema 17. Configuración de routers y protocolos de routing

Encaminamiento estático

- Órdenes para construcción manual de tablas de rutas
  - Añadir una ruta a una nueva red

    Ejemplo: `route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.100`

  - Para eliminar una ruta a una red

    Ejemplo: `route delete -net 192.168.2.0 netmask 255.255.255.0`
  - Añadir una ruta a un nuevo host

    Ejemplo: `route add -host 192.168.3.1 gw 192.168.1.200`
  - Para eliminar una ruta a un host

    Ejemplo: `route delete -host 192.168.3.1`
- Configuración de un router
  - Habilitar dos o más interfaces de red
  - Activar la variable forwarding

    ```
    echo 1 > /proc/sys/net/ipv4/ip_forward
    ```

Tema 17. Configuración de routers y protocolos de routing

PRACTICA

- Uno de los puestos de cada fila debe actuar como encaminador.
  - Dispondrá de 2 direcciones, una perteneciente a la red de su propia fila y otra de la red de la fila siguiente.
  - La última fila se enlaza con la primera para formar un anillo.
- En el puesto que actúa de encaminador:

```
ifconfig eth0 192.168.<fila>.<puesto> netmask 255.255.255.0
ifconfig eth0:1 192.168.<fila_siguiente>.<puesto>
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Ahora todos los puestos deben añadir manualmente las rutas a todas las demás redes.
  - Cada red tiene dos encaminadores. Se debe utilizar la ruta más corta.
  - Cada puesto puede, entonces, elegir a cualquiera de los dos como encaminador para alcanzar un destino determinado.
- Para añadir una ruta:

```
route add -net <destino> netmask 255.255.255.0 gw <encaminador>
```

- Comprobar que funciona haciendo ping y traceroute a distintos destinos.

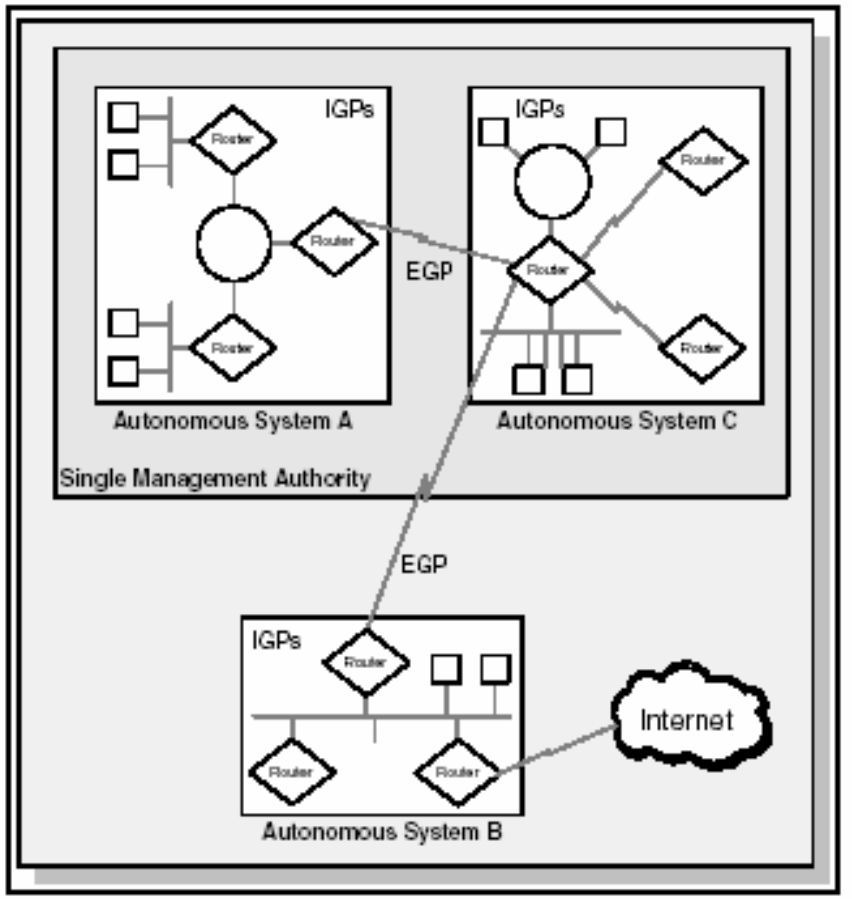
Encaminamiento dinámico (1)

Objetivo general del encaminamiento

- Encontrar el "**camino más corto**" desde el origen al destino a través de routers y redes intermedias
- Necesidad de definir una **métrica de encaminamiento**
  - **Número de saltos:** El camino más corto es aquel que tiene que atravesar un menor número de redes intermedias
  - **Distancia geográfica hasta el destino:** El camino más corto es aquel por el cual el paquete tiene que recorrer un menor número de kilómetros
  - **Retardo promedio:** El camino más corto es el más rápido, es decir el que traviesa las redes más rápidas y menos congestionadas
  - **Función de varias métricas**

Encaminamiento dinámico (2)

- Organización jerárquica de Internet: **SISTEMAS AUTÓNOMOS**
  - **Protocolos Internos (IGP)**
    - Lo utilizan los routers internos, para el encaminamiento dentro de un sistema autónomo
      - **RIP:** Routing Information Protocol
      - **OSFP:** Open Shortest Path First
      - **IGRP:** Internal Gateway Routing Protocol (de CISCO)
  - **Protocolos Externos (EGP)**
    - Lo utilizan los routers frontera, para el encaminamiento entre distintos sistemas autónomo
      - **EGP:** External Gateway Protocol (actualmente en desuso)
      - **BGP:** Border Gateway Protocol



Rafael Moreno (UCM) y Juan Carlos Fabero (UCM)

RIP: Routing Information Protocol (1)

Características generales

- Protocolo de interior usado en redes de tamaño reducido, ampliamente utilizado al incluirse en la distribución BSD de UNIX
- Basado en el protocolo de routing XNS de Xerox
- RIP es un protocolo de vector de distancia

Funcionamiento:

- El protocolo RIP se basa en el intercambio periódico de las tablas de rutas (o vectores de distancia) entre routers vecinos
  - Estas tablas incluyen las redes que son alcanzables por cada router y a que distancia se encuentran
- A partir de esta información los routers construyen o actualizan sus tablas de rutas
- La información se difunde mediante broadcast. Los paquetes RIP se envían usando el protocolo UDP
- La métrica utilizada en el protocolo RIP es el número de saltos

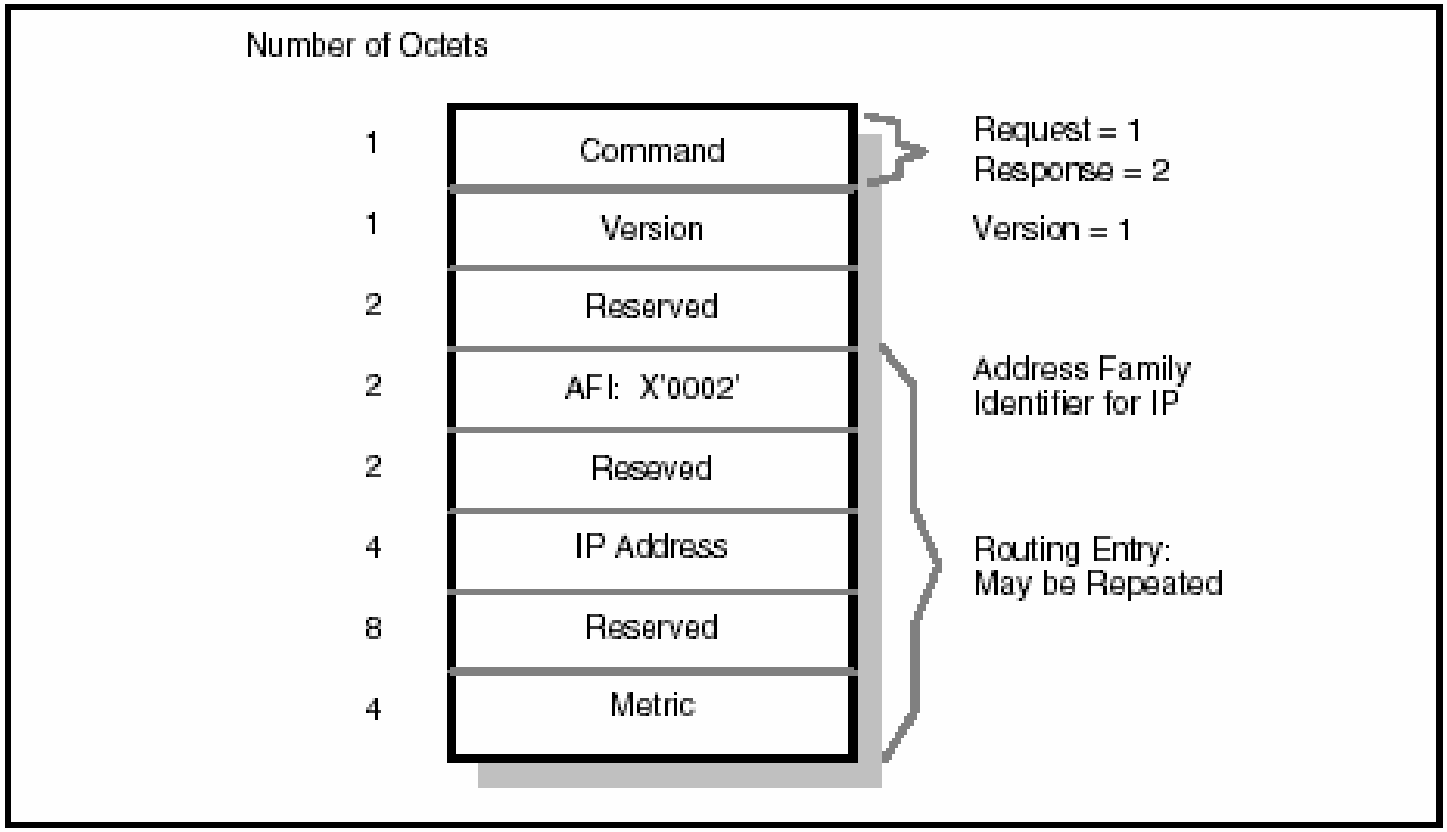
RIP: Routing Information Protocol (2)

Tipos de mensajes RIP

- Mensajes de solicitud (**REQUEST**)
  - Cuando un router nuevo se conecta, envía un mensaje REQUEST para solicitar información de encaminamiento al resto de routers
- Mensajes de respuesta (**RESPONSE**)
  - Los mensajes **RESPONSE** se utilizan para enviar la información sobre las redes alcanzables por un router y las distancias a las mismas
  - Estos mensajes se envían en las siguientes circunstancias:
    - Periódicamente (cada 30 segundos), los routers difunden mediante broadcast sus tablas de rutas a la red
    - Como respuesta a un mensaje de REQUEST, los routers envían un paquete de respuesta dirigido al router solicitante
    - En caso de que estén habilitada el mecanismo de “actualización forzada”, los routers anuncian las tablas de vectores de distancia siempre que se detecta alguna modificación

RIP: Routing Information Protocol (3)

- Formato de los mensajes RIP



Ejemplo de funcionamiento de RIP (1)

- Supongamos la siguiente red

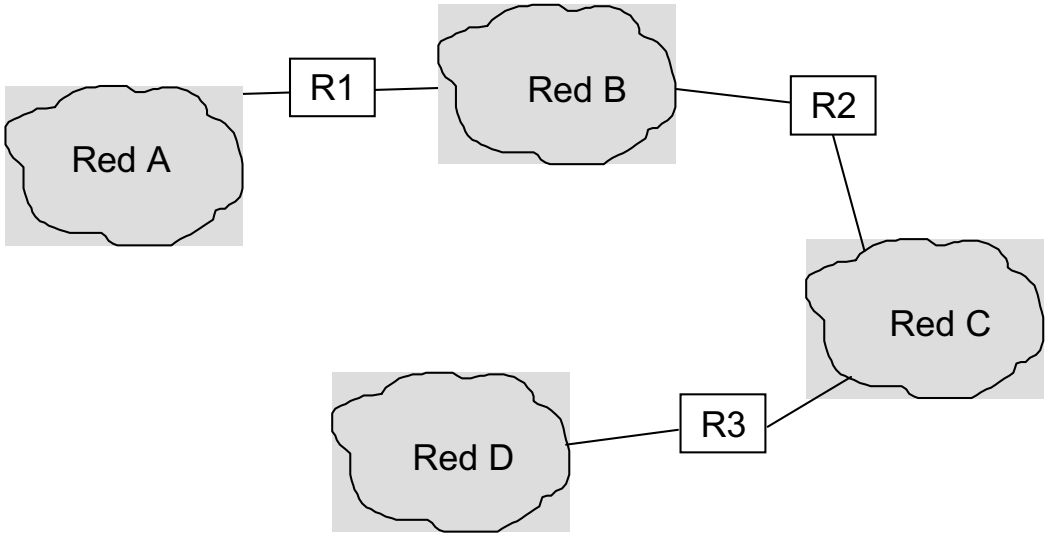


Tabla de rutas de R1

Destino	Router	Metric
Red A	Direct	1
Red B	Direct	1
Red C	R2	2
Red D	R2	3

Tabla de rutas de R2

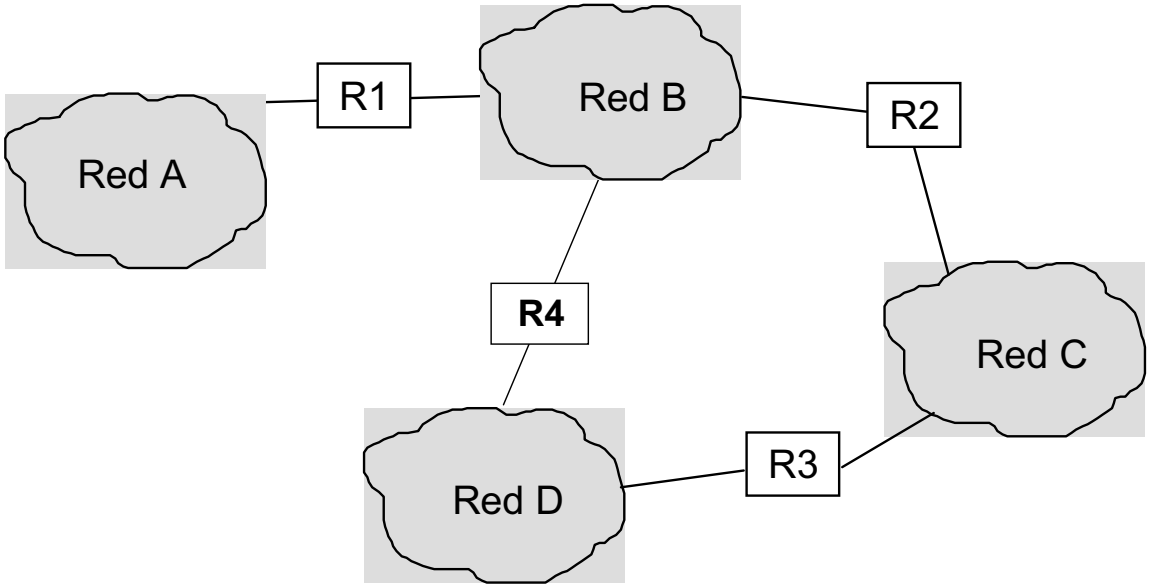
Destino	Router	Metric
Red A	R1	2
Red B	Direct	1
Red C	Direct	1
Red D	R3	2

Tabla de rutas de R3

Destino	Router	Metric
Red A	R2	3
Red B	R2	2
Red C	Direct	1
Red D	Direct	1

Ejemplo de funcionamiento de RIP (2)

- Supongamos que el router R4 se conecta por primera vez



- Inicialmente su tabla de rutas sólo contiene información de las redes a las que está conectado:

Tabla de rutas de R4 (inicial)

Destino	Router	Metric
Red B	Direct	1
Red D	Direct	1

Ejemplo de funcionamiento de RIP (3)

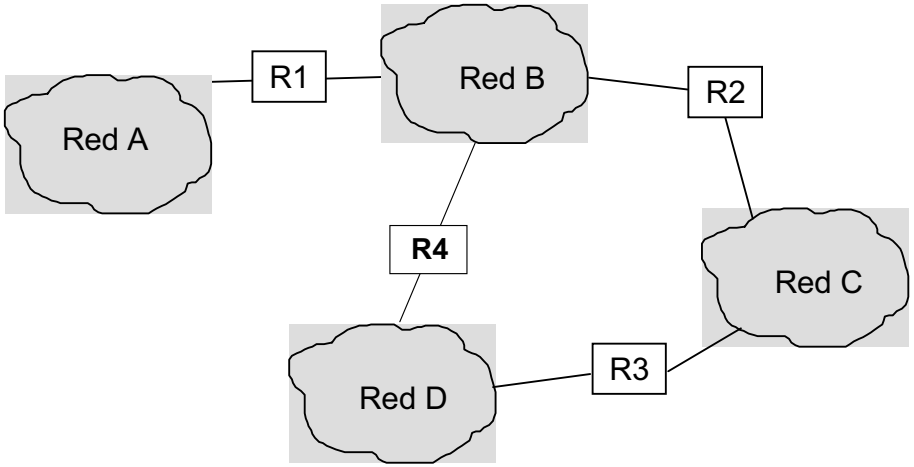
- El router R4 envía un mensaje RIP de tipo REQUEST solicitando información de rutas a sus routers vecinos

  Los routers vecinos le envían la siguiente información

RESPONSE de R1
- Red A - Distancia 1
- Red B - Distancia 1
- Red C - Distancia 2
- Red D - Distancia 3

RESPONSE de R2
- Red A - Distancia 2
- Red B - Distancia 1
- Red C - Distancia 1
- Red D - Distancia 2

RESPONSE de R3
- Red A - Distancia 3
- Red B - Distancia 2
- Red C - Distancia 1
- Red D - Distancia 1



- A partir de esta información R4 construye sus tablas de encaminamiento:

Tabla de rutas de R4

Destino	Router	Metric
Red A	R1	2
Red B	R4	1
Red C	R4	1
Red D	R2 ó R3	2

Ejemplo de funcionamiento de RIP (4)

- Transcurridos 30 segundos el router R4 transmite su tabla al resto de routers, los cuales modificarán sus propias tablas para adaptarlas a la nueva situación.

Tabla de rutas de R1

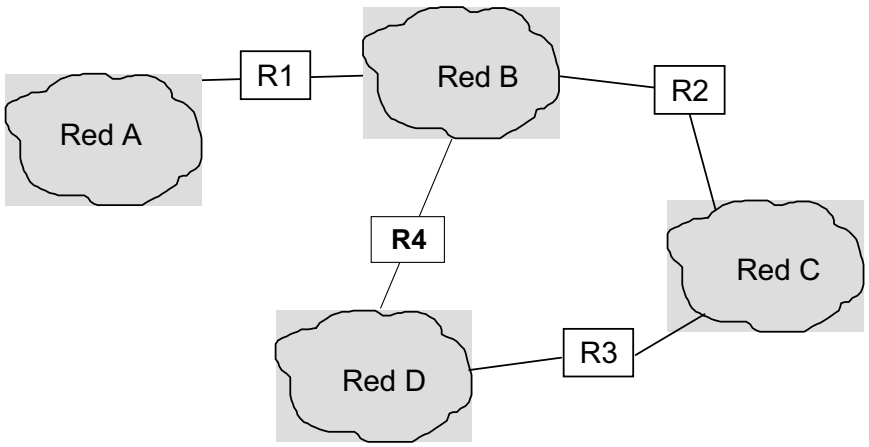
Destino	Router	Metric
Red A	Direct	1
Red B	Direct	1
Red C	R2	2
Red D	**R4**	**2**

Tabla de rutas de R2

Destino	Router	Metric
Red A	R1	2
Red B	Direct	1
Red C	Direct	1
Red D	R3	2

Tabla de rutas de R3

Destino	Router	Metric
Red A	R2	3
Red B	R2	2
Red C	Direct	1
Red D	Direct	1



Contenidos

- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- **Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios**
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, NAT
- Tema 20. Seguridad de la red

- **El protocolo TCP (Transmission Control Protocol)**
  - Protocolo de transporte "orientado a conexión"
  - Garantiza un servicio extremo a extremo fiable
    - Detectar/retransmitir segmentos de datos perdidos o erróneos
    - Detectar y descartar segmentos duplicados
    - Ordenar los segmentos en el destino y pasarlos de forma ordenada a la capa de aplicación
  - Utilizado en aplicaciones en las que la seguridad en la entrega es más importante que la rapidez
    - FTP, HTTP, Telnet, SMTP, etc.
- **El protocolo UDP (User Datagram Protocol)**
  - Protocolo de transporte "sin conexión"
    - NO garantiza un servicio extremo a extremo fiable
    - No controla la pérdida de paquetes, los errores o la duplicidad
  - Utilizado en aplicaciones en las que la rapidez en la entrega es más importante que la seguridad
    - DNS, SMNP, RIP, RTP, etc.

El protocolo TCP: Conceptos generales

Unidad de transferencia

- Segmento TCP

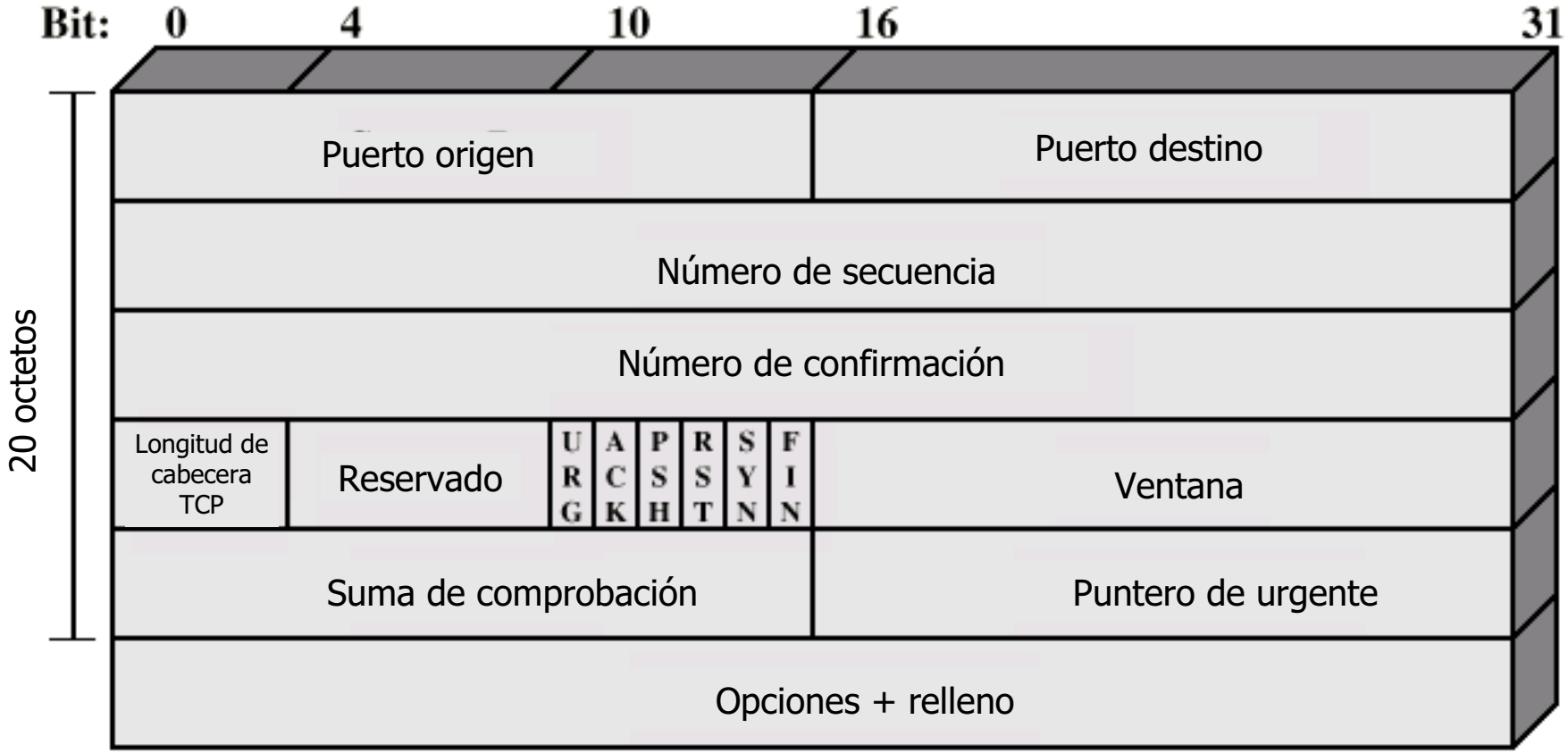
Fases en una transmisión mediante TCP

- Establecimiento de conexión
- Transferencia de datos
- Cierre de conexión

Mecanismos de control de errores en TCP

- Numeración de segmentos
  - Cada segmento lleva un número de secuencia de 32 bits
  - Indica la posición que ocupa el primer byte del segmento dentro del mensaje original
- Confirmaciones superpuestas del receptor
  - Cuando el receptor recibe un segmento de datos correcto y sin errores, **envía una confirmación** al emisor
- Retransmisión de segmentos
  - Si transcurrido un tiempo desde que se envió el segmento, el emisor no recibe confirmación, entonces **retransmite de nuevo el segmento**

Formato del segmento TCP



Campos de la cabecera del segmento TCP

- **Puerto origen y destino:**
  - Identifican los extremos de la conexión
- **Nº de secuencia:**
  - Indica la posición del primer byte del segmento con respecto al mensaje original
- **Nº de confirmación:**
  - Para enviar confirmaciones superpuestas en sentido contrario. Indica el nº de secuencia del siguiente byte que se espera recibir
- **Longitud de la cabecera:**
  - medida en palabras de 32 bits
- **Flag URG y puntero urgente:**
  - Si URG=1, el segmento transporta datos urgentes a partir del nº de byte especificado en el campo **puntero urgente**
- **Flag ACK:**
  - Si ACK=1, el segmento transporta una confirmación válida en el campo de superposición

Campos de la cabecera del segmento TCP

- **Flag PUSH:**
  - Si PUSH=1, indica que los datos deben ser pasado inmediatamente a la aplicación.
  - Si PUSH=0, los datos se pueden almacenar en un buffer de recepción y éstos se pasan a la aplicación cuando el buffer se llena.
- **Flag RST:**
  - Flag utilizado para abortar una conexión
- **Flag SYN:**
  - Flag utilizado en el establecimiento de la conexión.
  - Significa que los extremos deben sincronizar los números de secuencia iniciales de la transmisión
- **Flag FIN:**
  - Flag utilizado en la finalización de la conexión
- **Código de redundancia:**
  - Checksum de todas las palabras de 16 bits de la parte de datos
- **Ventana:**
  - Permite negociar dinámicamente el tamaño de la ventana de transmisión
- **Opciones:**
  - Permite negociar parámetros adicionales de la conexión, por ejemplo el tamaño máximo del segmento (MSS)

- **El modelo cliente-servidor**
  - **El cliente:**
    - Es la aplicación que solicita la conexión con la máquina remota
  - **El servidor:**
    - Es la aplicación que recibe y acepta la solicitud de conexión del cliente
- **Parámetros en una conexión TCP**
  - **Dir. IP del cliente:**
    - Identifica a la máquina cliente (la que solicita la conexión)
  - **Puerto del cliente:**
    - Identifica al proceso cliente dentro de la máquina cliente
  - **Dir. IP del servidor:**
    - Identifica a la máquina servidora (la que acepta la conexión)
  - **Puerto del servidor:**
    - Identifica al proceso servidor dentro de la máquina servidora

Asignación de números de puerto

El puerto del cliente

- Cuando un proceso cliente solicita una conexión TCP
  - El SO de la máquina cliente le asigna un número de puerto libre
  - Este número de puerto es siempre un número mayor que 1023

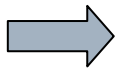
El puerto del servidor

- Cuando un proceso cliente solicita una conexión TCP con un servidor debe conocer de antemano su número de puerto
- Para ello existen dos soluciones:
  - Que el servidor utilice un "puerto bien conocido" (well-known ports)
    - FTP: 21
    - SSH: 22
    - TELNET: 23
    - SMTP: 25
    - HTTP: 80
    - POP3: 110
    - NNTP(News): 119
  - Usar el servicio de llamada a procedimiento remoto
    - **RPC** (Remote Procedure Call)

PRACTICAS (1)

Ver lista de puertos bien conocidos

- Ver contenidos del archivo **/etc/services**



```
. . .
ftp      21/tcp
ssh      22/tcp
telnet   23/tcp
smtp     25/tcp
. . .
```

Ver lista de servicios básicos de red controlados por proceso inetd

- Ver contenidos del archivo de configuración **/etc/inetd.conf**

```
name    type    proto  flags    user   pathname              args          .
ftp     stream  tcp    nowait   root   /usr/sbin/in.ftpd     in.ftpd
tftp    dgram   udp    wait     root   /usr/sbin/in.tftpd    in.tftpd /boot
```

- **Name:** puerto/nombre asociado al servicio
- **Type:** tipo de datos intercambiados
  - stream → datos secuenciados (protocolo TCP) y
  - dgram → datos no secuenciados (protocolo UDP)
- **Proto:** protocolo usado por el servicio: TCP o UDP
- **Flags:** indica si se cierra el servicio al finalizar la conexión (nowait) o se mantiene un tiempo (wait)
- **User:** usuario propietario del servicio
- **Pathname:** ruta completa al ejecutable
- **Args:** llamada al ejecutable (nombre + opciones + argumentos)

PRACTICAS (2)

Abrir servicio telnet

- Editar archivo de configuración `/etc/inetd.conf` y añadir la siguiente línea

```
telnet stream tcp nowait root /usr/sbin/in.telnetd /usr/sbin/in.telnetd -z nossl
```

- Ejecutar las siguientes órdenes para crear certificado de seguridad SSL (necesario para esta versión particular del servicio telnet)

```
cd /usr/lib/ssl/certs
openssl req -new -x509 -nodes -out telnetd.pem -keyout telnetd.pem
```

- Reiniciar proceso inetd:

```
/etc/init.d/inetd    restart
```

PRACTICAS (3)

Modificar el número de puerto de un servicio

- Ejemplo: modificar el puerto del servicio telnet → **nuevo puerto = 1000**
  - Editar y modificar el archivo **`/etc/services`**

```
. . .
ftp     21/tcp
ssh     22/tcp
telnet  23/tcp
smtp    25/tcp
. . .
```

```
. . .
ftp     21/tcp
ssh     22/tcp
telnet  1000/tcp
smtp    25/tcp
. . .
```

  - Reiniciar proceso inetd:

    `/etc/init.d/inetd restart`

PRACTICAS (4)

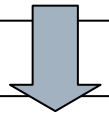
Abrir/cerrar servicios de red controlados por procesos propios

- Existen servicios controlados por sus propios procesos y sus propios scripts de arranque/parada. Se enumeran algunos ejemplos:
- **Servidor de Shell Seguro SSH**
  - Proceso: `sshd`
  - Script: `/etc/init.d/ssh`
- **Servidor de Correo Electrónico SMail**
  - Proceso: `smail`
  - Script: `/etc/init.d/smail`
- **Servidor Web Apache**
  - Proceso: `apache`
  - Script: `/etc/init.d/apache`
- **Servidor DNS (Domain Name Service)**
  - Proceso: `named`
  - Script: `/etc/init.d/bind9`
- **Servidor DHCP (versión 3)**
  - Proceso: `dhcpd3`
  - Script: `/etc/init.d/dhcp3-server`

Protección de puertos con TCP-Wrappers (1)

- TCP Wrappers es una herramienta de dominio público que permite:
  - Controlar el acceso a los servicios de red particularizando las acciones que se permite realizar para diferentes clientes
  - Registrar la ejecución de procesos asociados a servicios de red
- ¿Cómo convertir un servicio normal a TCP Wrapper?
  - Es necesario modificar el archivo **`/etc/inetd.conf`** para aquellos servicios que vayan a usar TCP Wrapper
  - Ejemplo: para modificar el servicio **telnet**

```
telnet stream tcp nowait root /usr/sbin/in.telnetd /usr/sbin/in.telnetd
```



```
telnet stream tcp nowait root    /usr/sbin/tcpd      /usr/sbin/in.telnetd
```

Protección de puertos con TCP-Wrappers (2)

- El acceso a servicios controlados por el TCP Wrapper se configuran mediante el archivo **`/etc/hosts.allow`**
  - Contiene la lista de sistemas remotos y usuarios que pueden utilizar cada servicio
- Ejemplo: archivo **`/etc/hosts.allow`**

```
sshd :  ALL : ALLOW
in.telnetd : .ucm.es, 147.96.80. :ALLOW
in.ftpd : .ucm.es, 147.96.80. :ALLOW
ALL : ALL@ALL : DENY
```

Practica (1)

Ejemplo: proteger el servicio *telnet* con TCP-wrappers

- Editar archivo de configuración `/etc/inetd.conf`

```
telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd -z nossl
```

- Editar archivo de configuración `/etc/hosts.allow` y añadir la siguiente entrada

```
in.telnetd : 192.168.1. : ALLOW
```

- Reiniciar proceso inetd:

```
/etc/init.d/inetd  restart
```

Practica (2)

Ejemplo: proteger el servicio *FTP* con TCP-wrappers

- Por defecto, el servicio FTP ya viene configurado con TCP-Wrapper

```
ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
```

- Editar archivo de configuración **/etc/hosts.allow** y añadir la siguiente entrada

```
in.ftpd : 192.168.1. : ALLOW
```

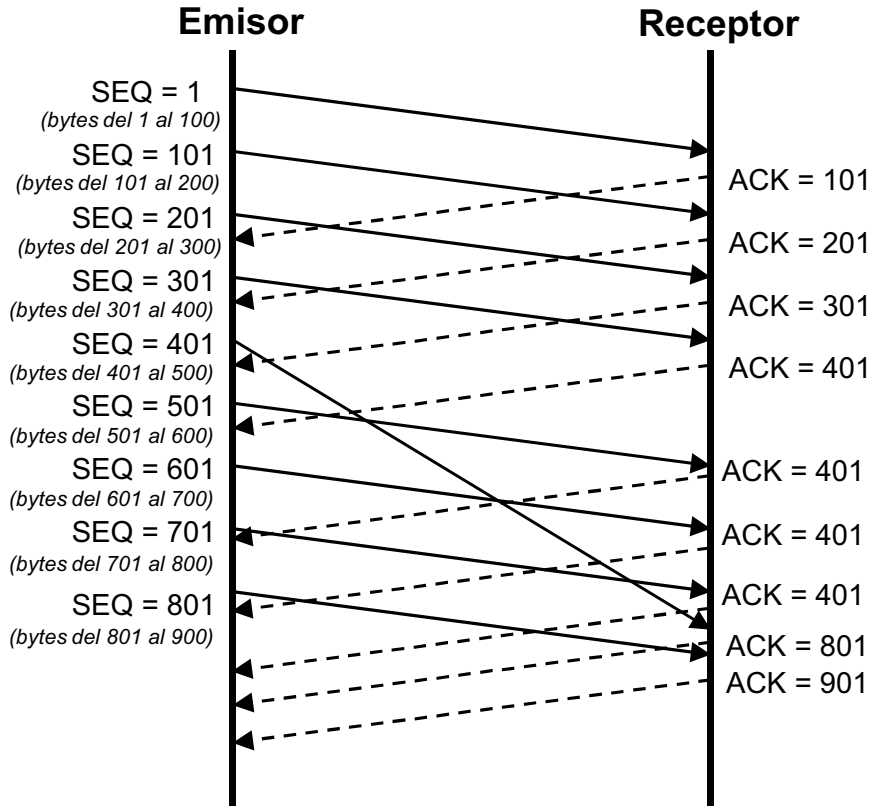
- Para permitir abrir una sesión FTP como usuario **root**
  - Editar el archivo **/etc/ftpusers** (lista de usuarios FTP restringidos)
  - Eliminar al usuario **root** de esta lista

Mecanismo de transmisión de datos en TCP (i)

Se utiliza un mecanismo similar al método de la ventana deslizante

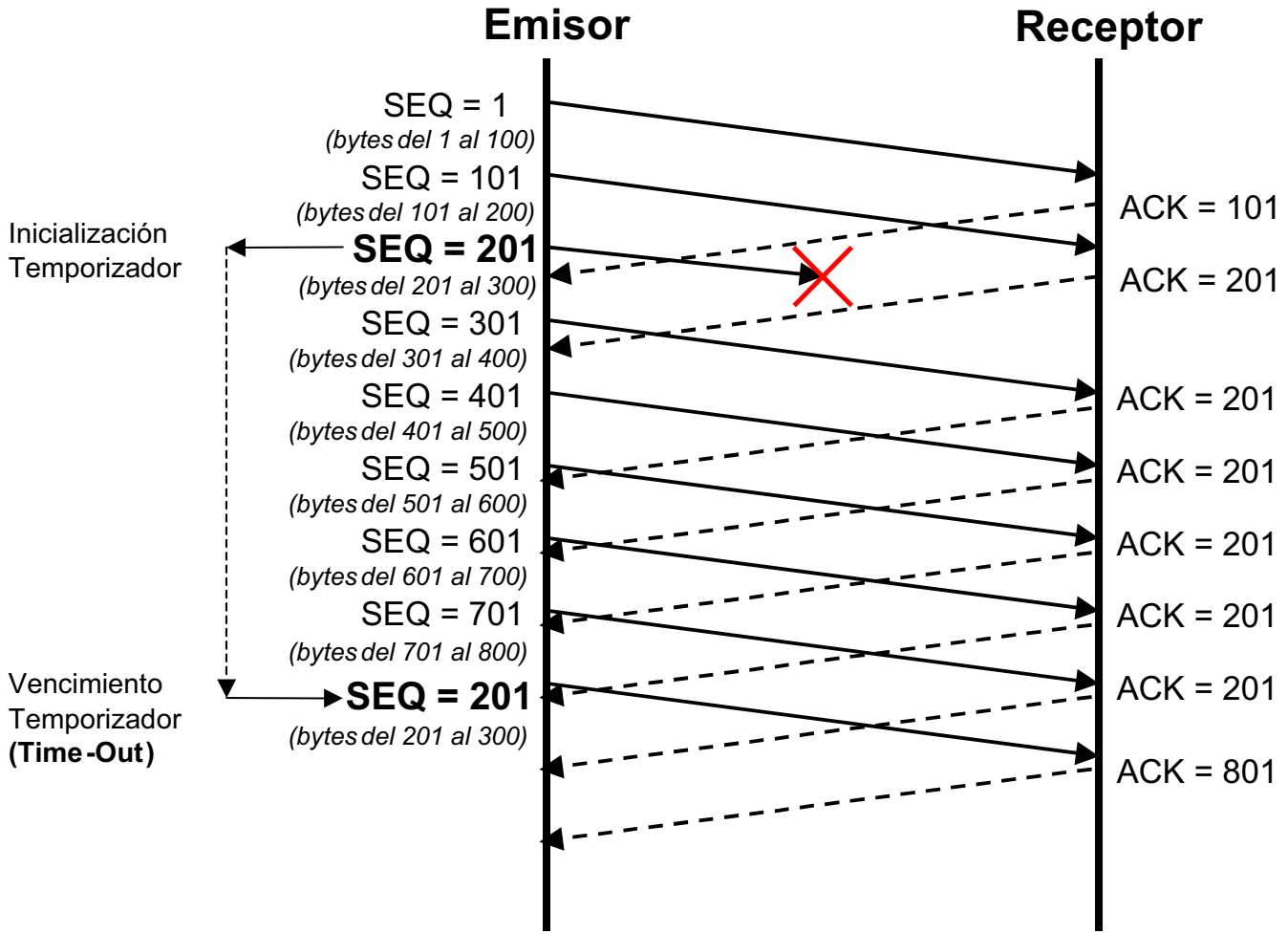
- El emisor envía todos los segmentos numerados
  - El campo **nº de secuencia** indica la posición del primer byte del segmento
- El receptor confirma los segmentos recibidos correctamente y en secuencia
  - El **campo nº de confirmación** contiene el identificador del siguiente byte que se espera recibir

Ejemplo 1

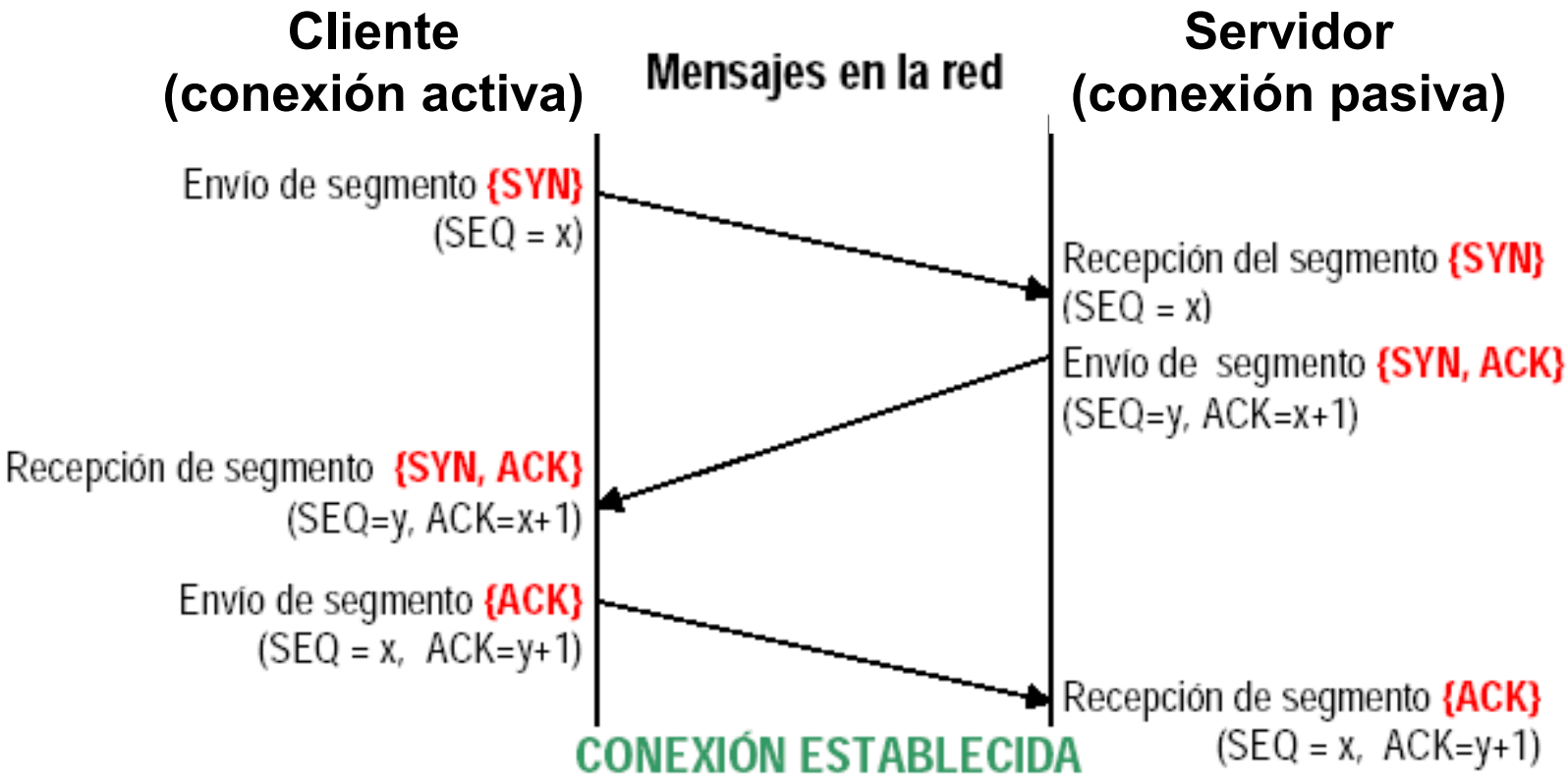


Mecanismo de transmisión de datos en TCP (ii)

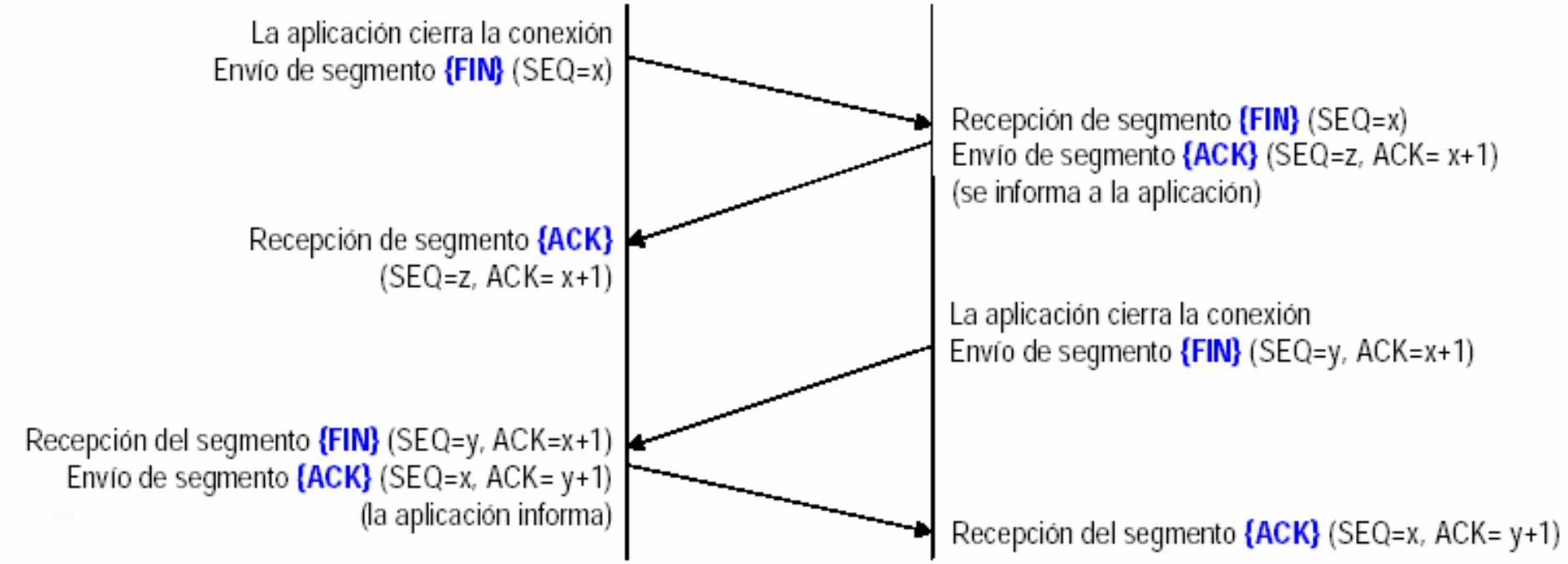
Ejemplo 2



Mecanismo de establecimiento de conexión TCP



Mecanismo de desconexión en TCP



Mecanismo de desconexión abrupta (aborto) en TCP

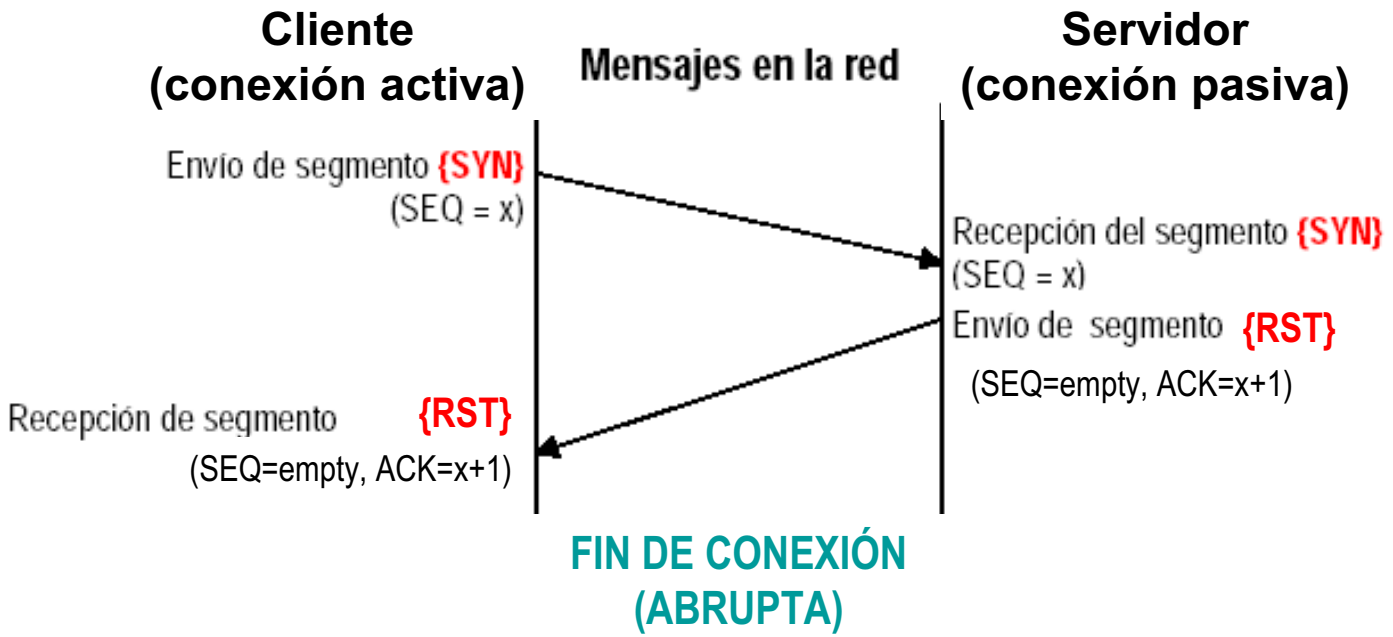
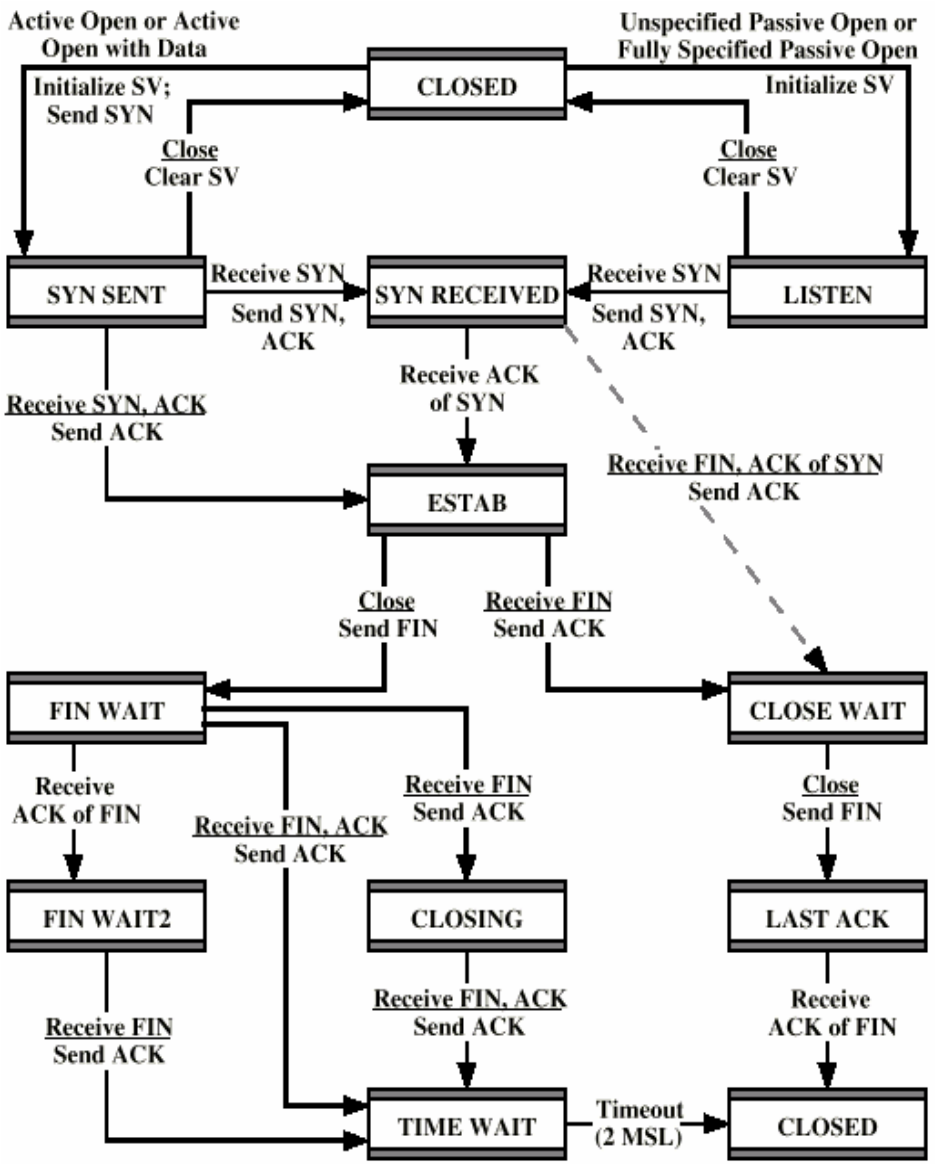
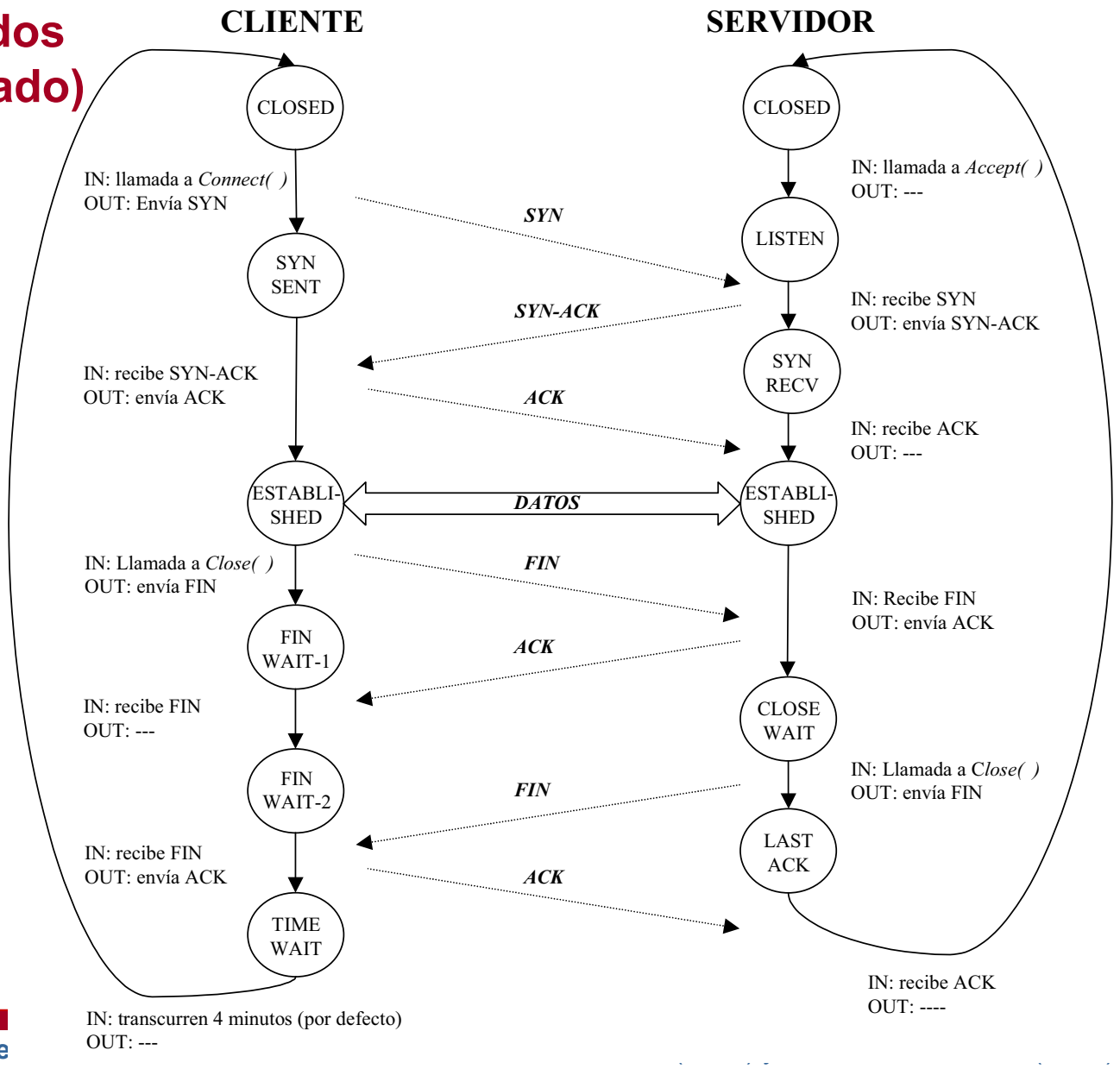


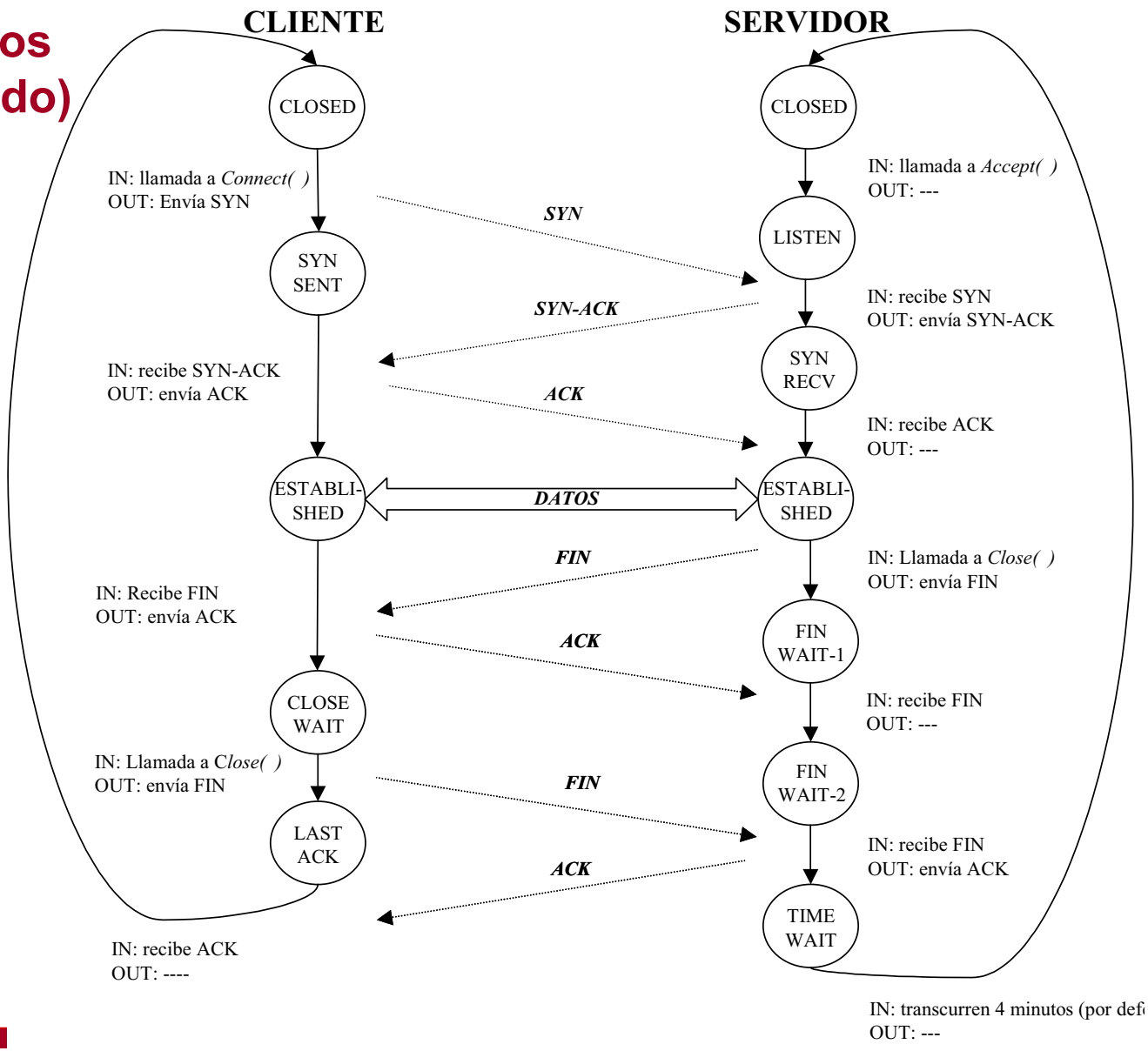
Diagrama de estados de TCP



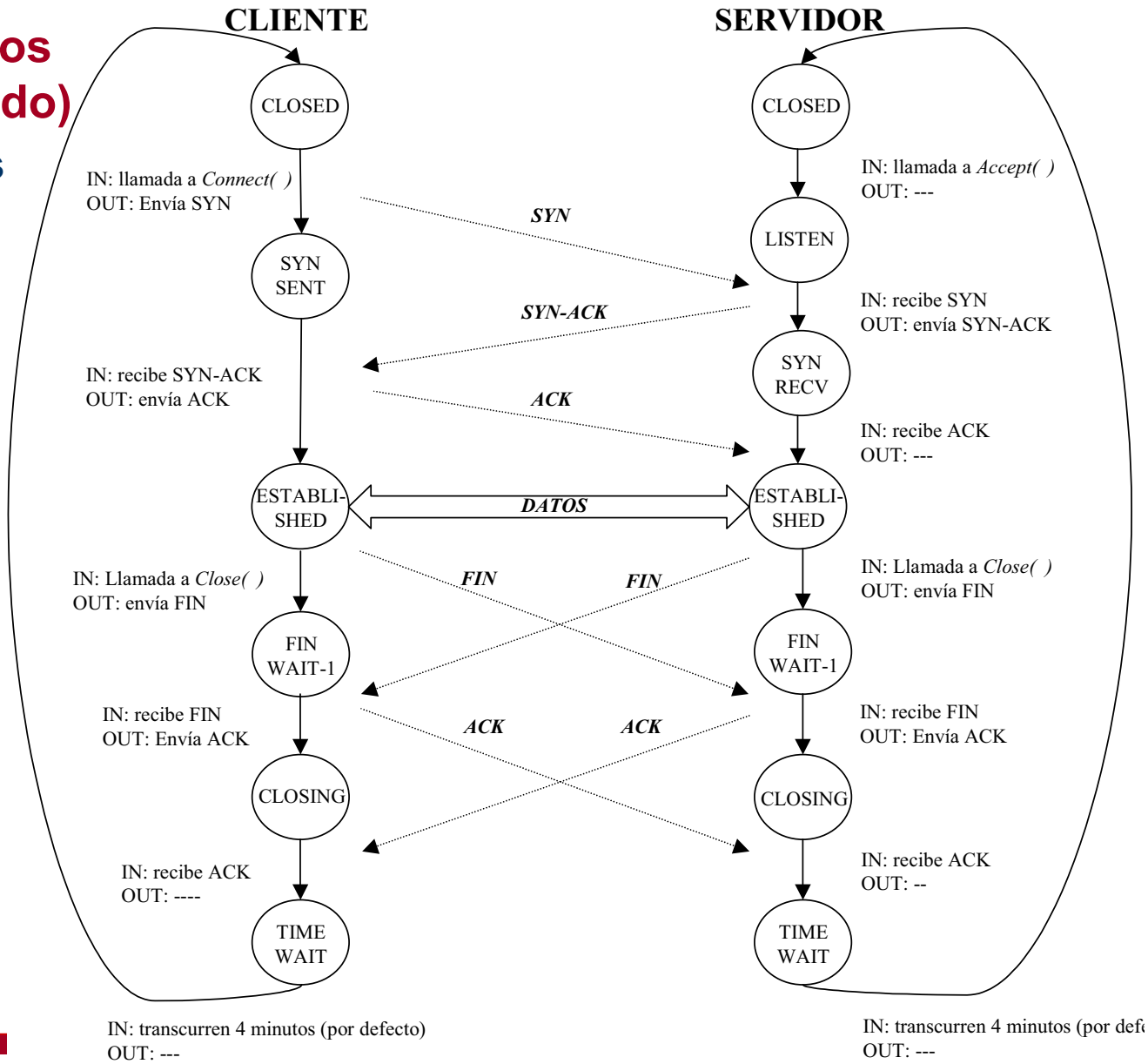
- **Diagrama de estados de TCP (simplificado)**
  - **Cierra conexión el cliente**



- **Diagrama de estados de TCP (simplificado)**
  - **Cierra conexión el servidor**



- **Diagrama de estados de TCP (simplificado)**
  - **Ambos extremos cierran conexión**



PRACTICAS

- Visualizar con **ethereal** los segmentos de establecimiento de conexión de conexión TCP
  a) Para el caso de un **servicio abierto**
     - Ver los segmentos **SYN, SYN-ACK, ACK**
     - Ver los números de secuencia iniciales del cliente y servidor
  b) Para el caso de un **servicio cerrado**
     - Ver los segmentos **SYN, RST**

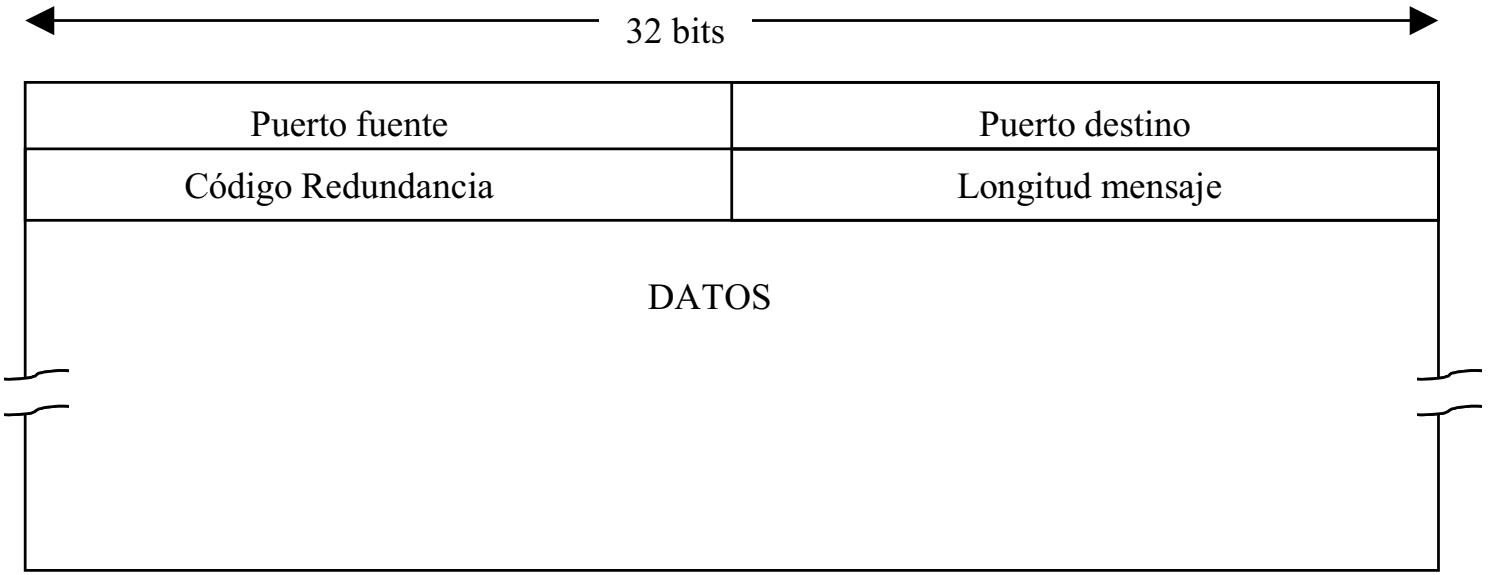
- Ver lista de puertos TCP abiertos y sus correspondientes estados
  - Orden **netstat -an**

El protocolo UDP

Conceptos generales

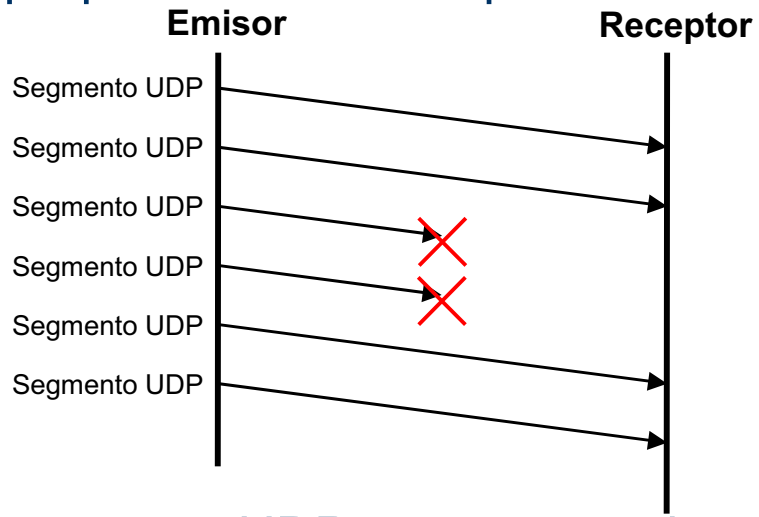
- El protocolo UDP es un protocolo sin conexión y no fiable
  - UDP divide los mensajes en segmentos, pero éstos NO se numeran
  - El receptor NO envía confirmación de la recepción de los mismos
- UDP no garantiza recuperación de paquetes perdidos o erróneos ni evita la duplicidad de paquete

Formato del paquete UDP

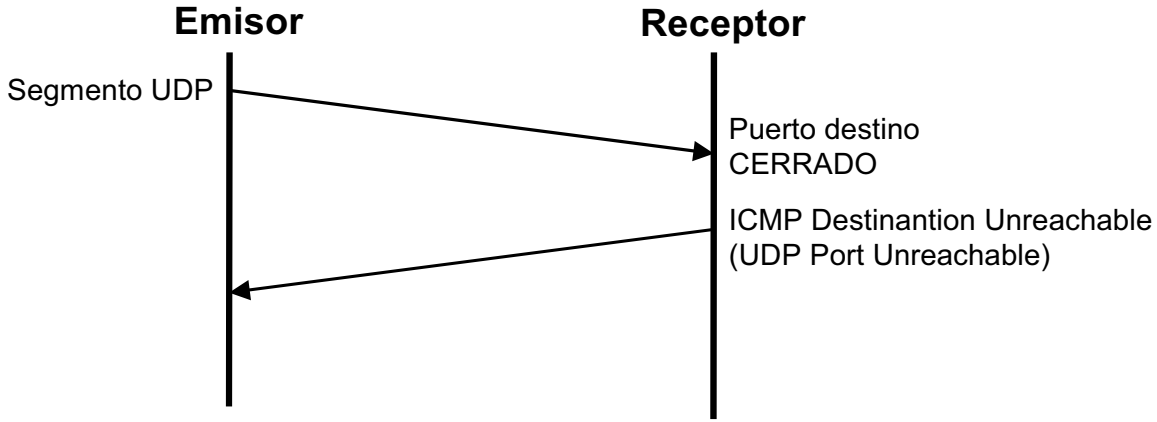


Mecanismo de transmisión de datos en UDP

- Envío de paquetes UDP a un puerto de servicio **ABIERTO**



- Envío de paquetes UDP a un puerto de servicio **CERRADO**



Asignación de puertos UDP

- Utiliza un esquema similar a TCP
  - **Puerto Cliente**
    - Lo asigna el S.O.
  - **Puerto del Servidor**
    - Puertos bien conocidos
    - Aplicaciones RPC

Contenidos

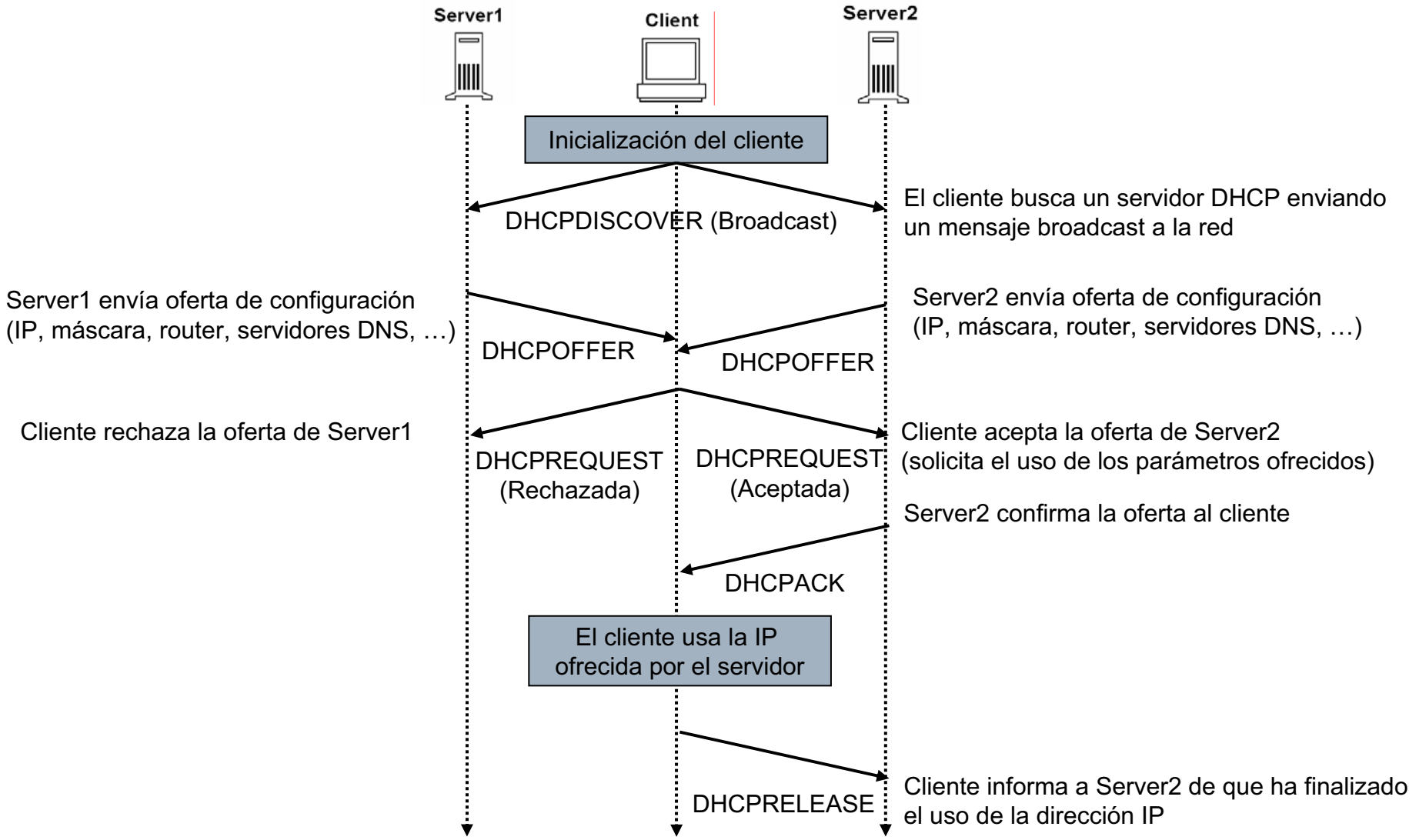
- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- **Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, …..**
- Tema 20. Seguridad de la red

DHCP (Dynamic Host Configuration Protocol)

¿Qué es DHCP?

- Configuración automática de los parámetros de la red
  - Dirección, máscara, router predeterminado, servidores DNS, etc.
- Clientes DHCP
  - No disponen de una configuración de red fija
  - Cuando arranca el sistema busca un servidor DHCP que le proporcione la información de configuración de red necesaria
- Servidor DHCP
  - Proporciona los parámetros de configuración de la red a los clientes que lo solicitan
- Ámbitos de aplicación
  - Entornos móviles (redes inalámbricas, hoteles, congresos, etc.)
  - Acceso telefónico o ADSL a través de ISP

Funcionamiento del protocolo DHCP



Mensajes del protocolo DHCP

- DHCPDISCOVER:
  - Mensaje broadcast del cliente para localizar a los servidores DHCP activos
- DHCPOFFER:
  - Respuesta del servidor, con una oferta de parámetros de configuración conforme a la situación del cliente
- DHCPREQUEST:
  - Mensaje del cliente con dos posibles respuestas
    - Oferta aceptada y solicitud del uso los parámetros ofertados
    - Oferta rechazada
- DHCPACK
  - Mensaje de confirmación y cierre desde el servidor hacia el cliente indicando los parámetros definitivos
- DHCPRELEASE
  - Mensaje del cliente para informar al servidor de que ha finalizado el uso de la dirección IP

PRACTICA

Configuración del servidor DHCP (versión 3)

- Crear archivo de configuración: **/etc/dhcp3/dhcp.conf**

```
subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.201 192.168.1.220;
   default-lease-time 86400;
   max-lease-time 86400;
   option routers 192.168.1.1;
   option broadcast-address 192.168.1.255;
   option subnet-mask 255.255.255.0;
   option domain-name-servers 192.168.1.100;
}
```

- Arrancar proceso servidor: **dhcpd3**

Tema 19. Conceptos avanzados de redes: DHCP

- **Elementos del archivo de configuración `/etc/dhcp3/dhcp.conf`**
  - subnet 192.168.1.0 netmask 255.255.255.0
    - Subred a la que se da servicio DHCP
  - range 192.168.1.201 192.168.1.220
    - Rango de direcciones IP ofrecidas dinámicamente a los clientes DHCP
  - default-lease-time 86400
    - Tiempo por defecto de alquiler de la dirección IP (en segundos)
  - max-lease-time 86400
    - Tiempo máximo de alquiler de la dirección IP (en segundos)
  - option routers 192.168.1.1
    - Router predeterminado anunciado a los clientes DHCP
  - option broadcast-address 192.168.1.255
    - Dirección de broadcast anunciada a los clientes DHCP
  - option subnet-mask 255.255.255.0
    - Mácara de subred anunciada a los clientes DHCP
  - option domain-name-servers 192.168.1.100
    - Lista de servidores DNS anunciada a los clientes DHCP

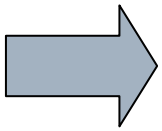
PRACTICA

Configuración del cliente DHCP en Linux

- Modificar el archivo de configuración de la red
  /etc/network/interfaces

Cliente con IP fija

```
auto eth0
iface eth0 inet static
        address 192.168.1.40
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.0.255
        gateway 192.168.1.1
```



Cliente con IP dinámica (DHCP)

```
auto eth0
iface eth0 inet dhcp
```

- Reiniciar la interfaz de red eth0 con las siguientes órdenes:

```
ifdown eth0
ifup eth0
```

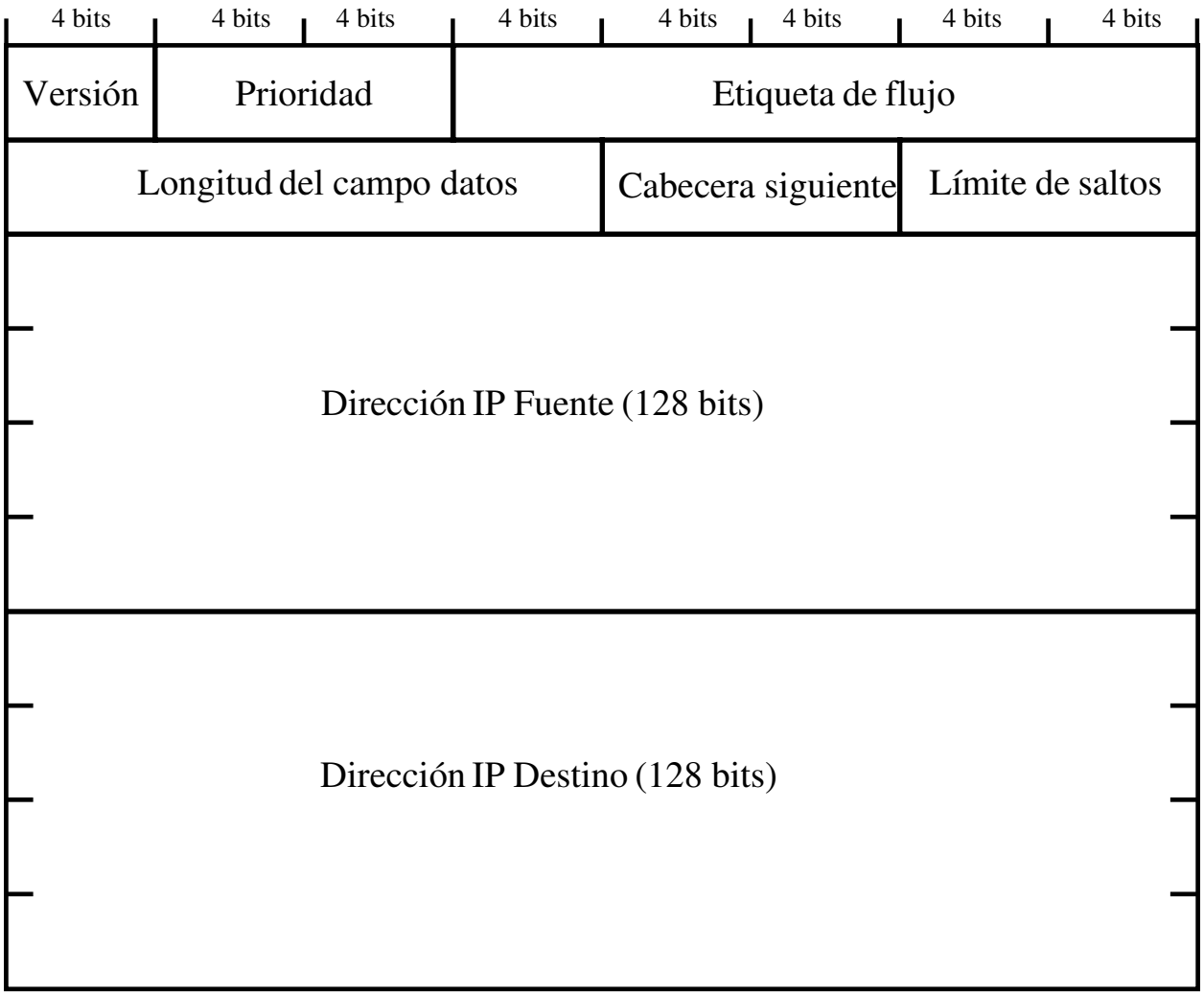
IPv6

- **Limitaciones de IPv4:**
  - Direcciones de 32 bits:
    - Inconveniente: pocas direcciones
    - Solución: Utilizar intranets con direcciones privadas y NAT
  - Organización en clases:
    - Inconveniente: se desperdician muchas direcciones
    - Solución: CIDR
  - Formato de datagrama:
    - Longitud de cabecera variable
    - Fragmentación en los encaminadores
  - Seguridad:
    - No tiene prevista ninguna opción de seguridad. Solución: IPSec.

IPv6

- **La solución: IPv6**
  - Direcciones: 128 bits (más de $3.4 * 10^{38}$ direcciones)
  - Formato simplificado de cabecera (mayor velocidad de procesamiento)
  - Autoconfiguración de los interfaces de red (*plug-and-play*)
  - Soporte para tráfico en tiempo real (VoIP, Vídeo bajo demanda...)
  - Opciones de seguridad

Formato de la cabecera IP



Formato de cabecera IP

- *Versión*: distingue entre las versiones 4 y 6.
- *Prioridad*: clase de tráfico. 0-7: tráfico normal; 8-15: tráfico en tiempo real.
- *Etiqueta de flujo*: 0 para tráfico normal; permite a los encaminadores distinguir entre datagramas que pertenecen al mismo flujo de datos.
- *Longitud de los datos*: incluye la longitud de las cabeceras adicionales.
- *Siguiente cabecera*: indica el tipo de la siguiente cabecera de extensión (si la hay), o el protocolo de capa superior en el campo de datos.
- *Límite de saltos*: similar al TTL de IPv4.
- *Direcciones origen y destino*: 128 bits cada una.

Tipos de direcciones.

- IPv6 soporta los siguientes tipos de direcciones:
  - **Unicast**: se refieren a una sola máquina en Internet. Un datagrama dirigido a una dirección unicast se entrega sólo a la máquina con esa dirección.
  - **Multicast**: identifican a un grupo de máquinas. Un datagrama dirigido a una dirección multicast se entrega a todas las máquinas que tienen esa dirección.
  - **Anycast**: identifican a un grupo de máquinas. Un datagrama dirigido a una dirección anycast se entrega sólo a la máquina más cercana con esa dirección.
  - No hay direcciones de **broadcast**.

Representación de las direcciones

- Las direcciones se representan por 8 grupos de 16 bits cada uno expresados con caracteres hexadecimales. Los grupos están separados por el carácter ":".
  - Ejemplos:
    - FEC0:BAC8:934F:0234:5678:12AB:CF23:0987
    - 2001:0000:0000:0000:0001:0000:0000:0056
- Los 0 al comienzo de un campo se pueden omitir:
    - FEC0:BAC8:934F:234:5678:12AB:CF230987
    - 2001:0:0:0:1:0:0:56
- Varios grupos :0: contiguos se pueden resumir por ::
    - 2001::1:0:0:56
  - Sólo puede aparecer una vez ::
    - 2001::1::56 -> es ambigua

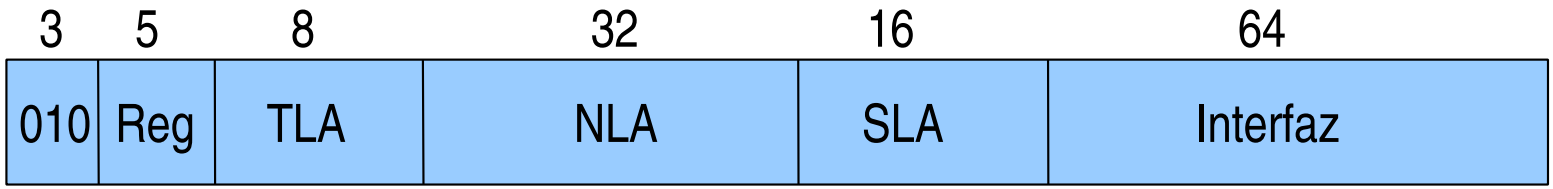
Estructura de las direcciones

- IPv4 sólo permite un nivel jerárquico: netid y hostid.
- Para facilitar las tareas de encaminamiento, IPv6 permite más niveles de jerarquía.
  - Es lo que se conoce como *agregado de direcciones*.
- Cada dirección IPv6 comienza por un prefijo que indica qué tipo de dirección es:

Tipo de dirección	FP (binario)	FP (hexadecimal)
Reservada	0000 0000	00
Unicast Global Agregable	001	2 -o 3
Unicast Enlace Local	1111 1110 10	FE8
Unicast Sitio Local	1111 1110 11	FEC
Multicast	1111 1111	FF

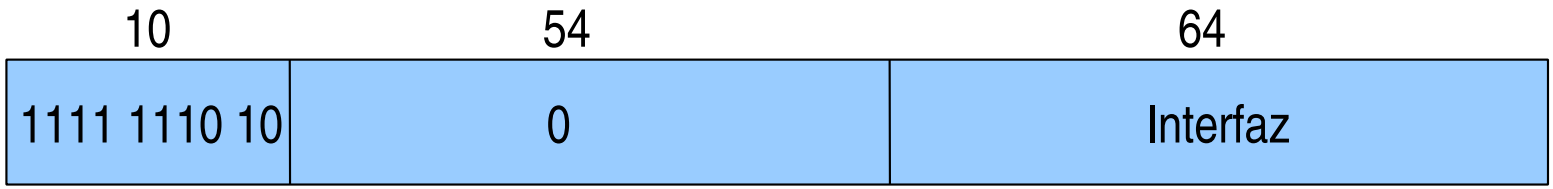
Direcciones Unicast Globales Agregables

- Son las que utiliza una máquina conectada a Internet.
- Permiten la autoconfiguración:
  - Una máquina puede obtener el prefijo de red desde el encaminador de su red.
  - El identificador de interfaz se puede construir a partir de la dirección MAC, o bien se puede fijar en la configuración de la máquina.



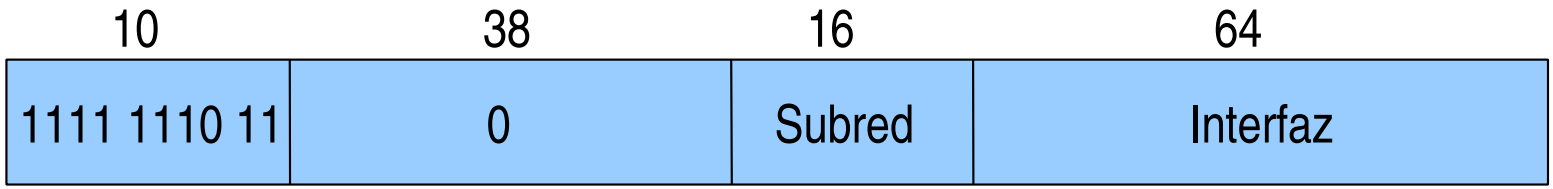
Direcciones Unicast de Enlace Local

- Son direcciones privadas que pueden utilizarse en intranets no jerárquicas (planas).
- Nunca se encaminan hacia el exterior.
- Permiten realizar las funciones de "*descubrimiento de vecino*".
- En GNU/Linux todo interfaz de red se autoconfigura con una dirección unicast de enlace local.



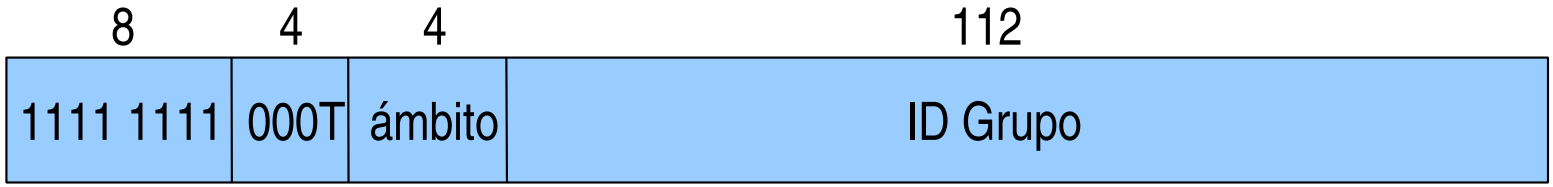
Direcciones Unicast de Sitio Local

- Son direcciones privadas que pueden utilizarse en intranets jerárquicas.
- Nunca se encaminan hacia el exterior.



Direcciones Multicast

- Son direcciones asignadas a grupos de máquinas.
- Se caracterizan por su **ámbito**:
  - Nodo local (0001) (0x1)
  - Enlace local (0010) (0x2)
  - Sitio local (0101) (0x5)
  - Global (1110) (0xE)
- Indicador T: permanente (T=0); temporal (T=1).



Direcciones Multicast

- El protocolo de descubrimiento de vecino tiene direcciones reservadas:
  - Desde FF02::1:FF00:0 hasta FF02::1:FFFF:FFFF
  - Ejemplo:
    - Se necesita averiguar la dirección MAC asociada con la dirección IPv6 2001::1:800:200E:8C6C
    - Se envía un mensaje ICMP Neigbour Discovery a la dirección FF02::1:FF0E:8C6C
- Direcciones multicast de los encaminadores:
  - FF01::2 -> encaminadores del nodo local.
  - FF02::2 -> encaminadores del enlace local.
  - FF05::2 -> encaminadores del sitio local.
  - FF02::9 -> encaminadores RIP del enlace local.
- Direcciones multicast de los computadores:
  - FF01::1 -> computador del nodo local (todos los interfaces del nodo local).
  - FF02::1 -> computadores del enlace local.

Configuración IPv6 en GNU/Linux

- El kernel ya incorpora soporte para IPv6.
- Comprobar la configuración:

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0F:B0:A5:00:6E
          inet addr:147.96.80.26  Bcast:147.96.81.255  Mask:255.255.254.0
          inet6 addr: fe80::20f:b0ff:fea5:6e/64 Scope:Link
          inet6 addr: fec0::50:10/96 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Configuración en GNU/Linux

Desde la línea de órdenes:

```
# ifconfig eth0 add fec0::50:10/96
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0F:B0:A5:00:6E
          inet addr:147.96.80.26  Bcast:147.96.81.255  Mask:255.255.254.0
          inet6 addr: fe80::20f:b0ff:fea5:6e/64 Scope:Link
          inet6 addr: fec0::50:10/96 Scope:Site
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1396375 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1626 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:171897115 (163.9 MiB)  TX bytes:316161 (308.7 KiB)
          Interrupt:225 Base address:0x5000
```

Configuración en GNU/Linux

- En el archivo /etc/network/interfaces :

```
auto eth0
iface eth0 inet static
  address 147.96.80.26
  netmask 255.255.254.0
  network 147.96.80.0
  broadcast 147.96.81.255
  gateway 147.96.80.1
  dns-servers 127.0.0.1
  up /sbin/ifconfig eth0 add fec0::50:10/96
```

Anuncio de prefijos en un encaminador Linux

- Se puede configurar una máquina Linux para que actúe de encaminador y anuncie el prefijo de red.
- Cada vez que arranca una máquina, solicita el prefijo de red desde el encaminador y autoconfigura su interfaz.
- Activar el demonio **zebra** (del paquete **quagga**):
  - En /etc/quagga/daemons añadir la entrada: zebra=yes
  - En /etc/quagga/zebra.conf:

```
! Zebra configuration saved from vty
!
hostname zebra
password clave1
enable password clave2
log file /var/log/zebra/zebra.log
!
interface eth0
        no ipv6 nd suppress-ra
        ipv6 nd prefix fec0:1::/64
```

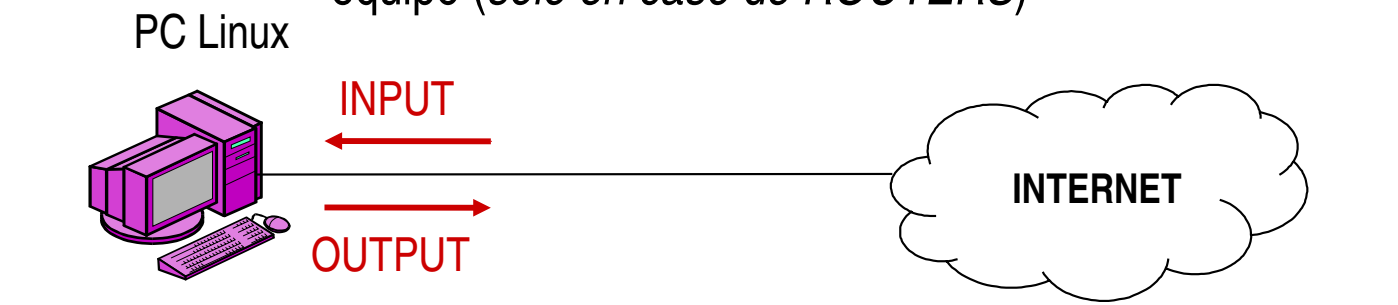
Firewalls: filtrado de paquetes

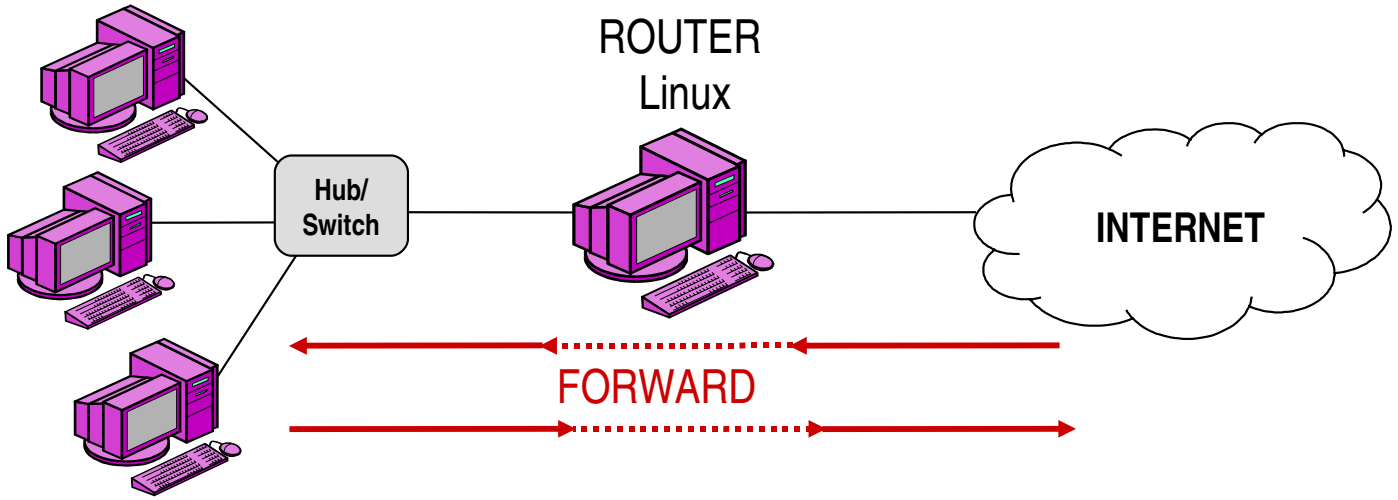
- **Firewalls: conceptos básicos**
  - Funciones principales de un firewall
    - Filtrado de paquetes
      - El firewall protege un equipo o una red controlando y limitando los paquetes de entrada y de salida
    - Registro de actividad (LOG)
      - El firewall puede registrar las acciones realizadas con los paquetes que entran o salen del equipo o de la red
    - Traducción de direcciones de red (NAT)
      - La mayoría de firewalls permiten realizar operaciones de NAT
  - Tipos de firewalls
    - Firewalls personales
      - Aplicación software para proteger un único equipo de forma individual
    - Firewalls de red
      - Permiten filtrar el tráfico de entrada o salida de una red completa
      - Pueden ser de dos tipos:
        - Router dedicado con posibilidad de filtrado (Cisco, 3Com, etc.)
        - Computador con varias tarjetas y un software adecuado (Linux+IPTables, Solaris+SunScreen, NT+Firewall-1, etc.)

Firewalls: filtrado de paquetes

Firewalls personales: IPtables de Linux (i)

- Cadenas de reglas de filtrado
  - Cadena INPUT → Para paquetes que entran al equipo
  - Cadena OUTPUT → Para paquetes que salen del equipo
  - Cadena FORWARD → Para paquetes que pasan a través del equipo (*sólo en caso de ROUTERS*)





Firewalls: filtrado de paquetes

Firewalls personales: IPtables de Linux (ii)

Principales criterios de filtrado de IPTables

Opción/Ejemplo	Significado
-A INPUT	Añade regla a cadena de entrada
-A OUTPUT	Añade regla a cadena de salida
-A FORWARD	Añade regla a la cadena forward (sólo en caso de routers)
-s 192.168.1.1	Filtrado por dirección IP origen
-d 140.10.15.1	Filtrado por dirección IP destino
-p tcp	Filtrado de paquetes TCP
-p udp	Filtrado de paquetes UDP
-p icmp	Filtrado de paquetes ICMP
--sport 3000	Filtrado por nº de puerto origen (sólo para TCP o UDP)
--dport 80	Filtrado por nº de puerto destino (sólo para TCP o UDP)
--icmp_type 8	Filtrado por código del paquete ICMP (sólo para ICMP)
-i eth0	Filtrado por interfaz de red de entrada
-o eth1	Filtrado por interfaz de red de salida

Firewalls: filtrado de paquetes

Firewalls personales: IPtables de Linux (iii)

Filtrado por estado de la conexión

Opción	Significado
-m state --state NEW	Filtrado de paquetes correspondientes a conexiones nuevas (el primer paquete visto en una conexión)
-m state --state ESTABLISHED	Filtrado de paquetes correspondientes a conexiones ya establecidas
-m state --state RELATED	Filtrado de paquetes relacionados con otras conexiones existentes (Ej. conexión de datos FTP, o paquetes ICMP)
-m state --state INVALID	Filtrado de paquetes que no pertenecen a ninguno de los estados anteriores

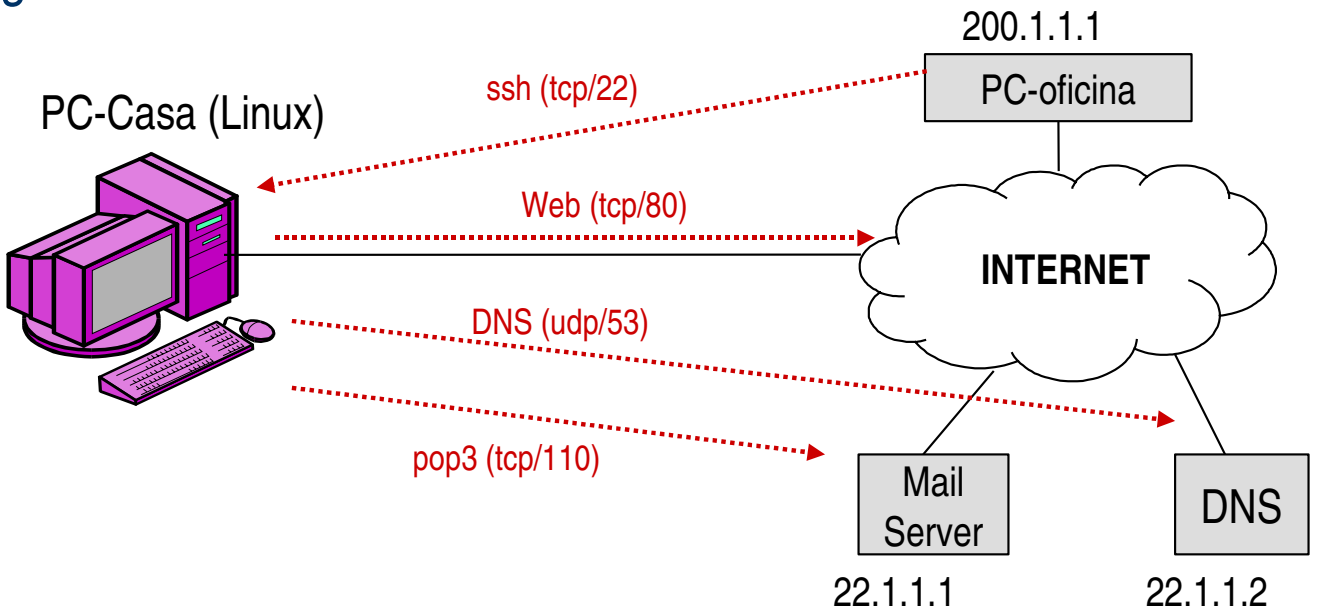
Acciones

Opción	Significado
-j ACCEPT	El paquete es aceptado
-j DROP	El paquete es descartado

Firewalls: filtrado de paquetes

Firewalls personales: IPtables de Linux (iv)

Ejemplo de reglas de filtrado



- Conexiones entrantes permitidas
  - Servicio SSH desde PC-oficina (200.1.1.1)
- Conexiones salientes permitidas
  - Servicio web a cualquier destino
  - Servicio pop3 a servidor de correo (22.1.1.1)
  - Servicio DNS a servidor DNS (22.1.1.2)
- Resto conexiones: RECHAZADAS

Firewalls: filtrado de paquetes

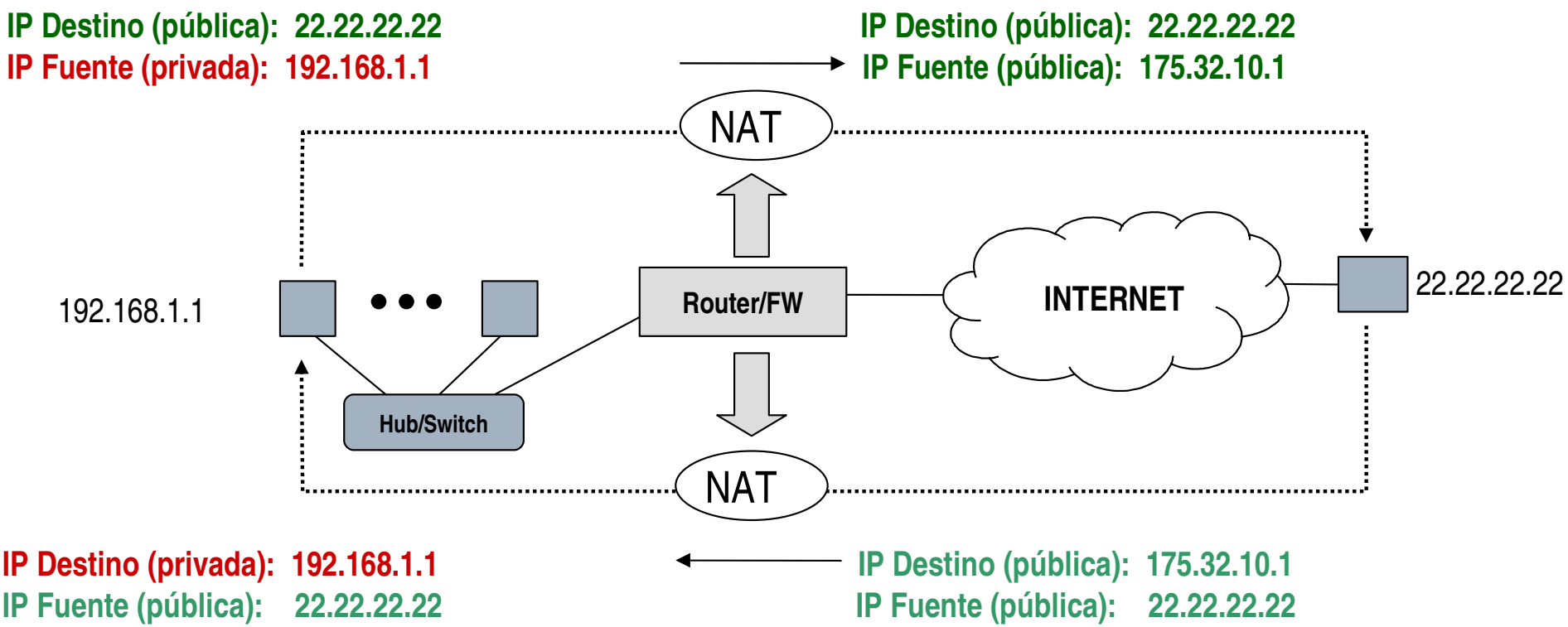
Firewalls personales: IPtables de Linux (v)

Ejemplo de reglas de filtrado (cont.)

```
# Establecemos política por defecto para cadenas INPUT, OUPUT y FORWARD
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Dejamos entrar o salir cualquier paquete correspondiente a
# conexiones establecidas o relacionadas
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Permitimos conexiones entrantes SSH (tcp/22) desde pc-oficina
iptables -A INPUT -s 200.1.1.1 -p tcp --sport 22 -m state   \
     --state NEW -j ACCEPT
# Permitimos conexiones web salientes (tcp/80) a cualquier destino
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
# Permitimos conexiones pop3 salientes (tcp/110) con servidor de correo
iptables -A OUTPUT -d 22.1.1.1 -p tcp --dport 110 -m state   \
      --state NEW -j ACCEPT
# Permitimos conexiones DNS salientes (udp/53) con servidor DNS
iptables -A OUTPUT -d 22.1.1.2 -p udp --dport 53 -m state   \
      --state NEW -j ACCEPT
```

NAT: Traducción de Direcciones de Red

- **NAT (Network Address Translation)**
  - Redes privadas
    - Necesidad de traducir IP privadas a IP públicas



NAT: Traducción de Direcciones de Red

- **Tipos de NAT (i)**
  - NAT estático
    - N direcciones privadas ⟷ N direcciones públicas
    - Asignación fija
    - Ejemplo (N=7)

Tabla de traducción NAT (estática)

IP Privada	IP Pública
192.168.1.1	175.20.12.1
192.168.1.2	175.20.12.2
192.168.1.3	175.20.12.3
192.168.1.4	175.20.12.4
192.168.1.5	175.20.12.5
192.168.1.6	175.20.12.6
192.168.1.7	175.20.12.7

NAT: Traducción de Direcciones de Red

Tipos de NAT (ii)

- NAT dinámico
  - N direcciones privadas ⟷ M direcciones públicas (M<N)
  - Asignación dinámica
    - Sólo puede darse salida a Internet a M máquinas simultáneamente
  - Ejemplo (N=7; M=3)

Tabla de traducción NAT (dinámica)

IP Privada	IP Pública
192.168.1.3	175.20.12.1
192.168.1.7	175.20.12.2
192.168.1.5	175.20.12.3

192.168.1.1
192.168.1.2
192.168.1.4
192.168.1.6

} Sin posibilidad de acceso a Internet hasta que se libere una IP pública

NAT: Traducción de Direcciones de Red

Tipos de NAT (iii)

- NAPT (Network Address/Port Translation) o Masquerading
  - N direcciones privadas ⟷ 1 dirección pública
  - Funcionamiento
    - La única IP pública disponible es la IP pública del Router/FW
    - El nº puerto cliente de la máquina origen se traduce a un puerto libre del Router/FW
  - Ejemplo:

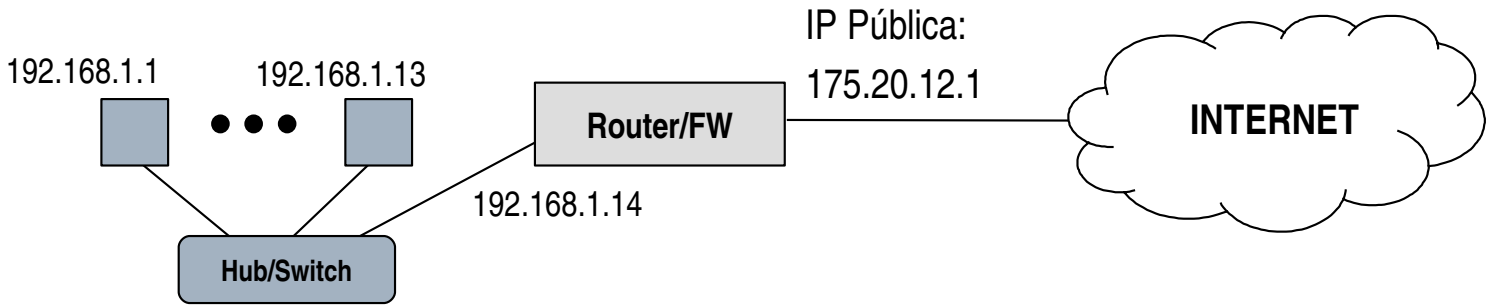


Tabla de traducción NAPT

IP_Privada:Port_Cliente	IP_Pública:Port_Cliente_FW
192.168.1.1:3289 →	175.20.12.1:10001
192.168.1.7:4256 →	175.20.12.1:10002
192.168.1.5:3882 →	175.20.12.1:10003

NAT: Traducción de Direcciones de Red

Tipos de NAT (iv)

- Port Forwarding o Virtual Servers
  - N direcciones privadas ⟷ 1 dirección pública
  - Permite tener servidores en la red privada "visibles" desde Internet
    - Desde Internet, todos los servidores se ven con una misma IP pública (la IP del Router/FW)
    - El Router/FW debe redireccionar los paquetes al servidor real de la red interna
- Ejemplo:

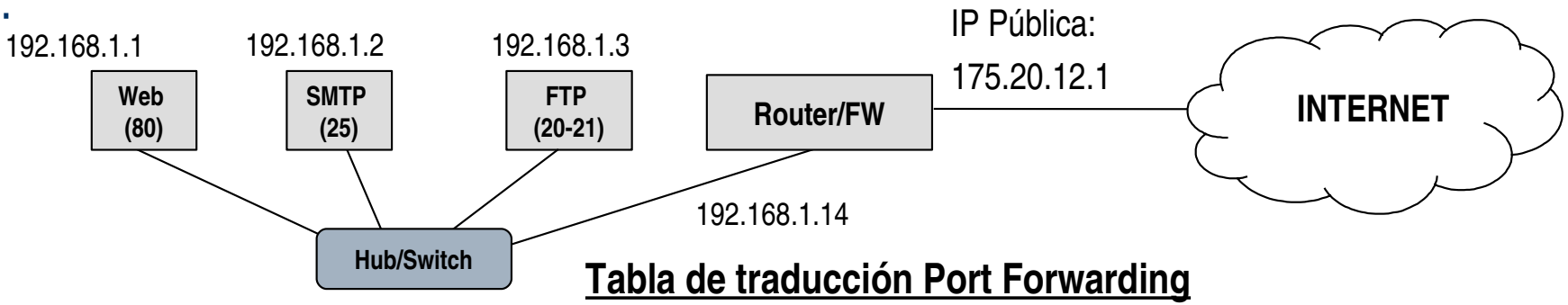
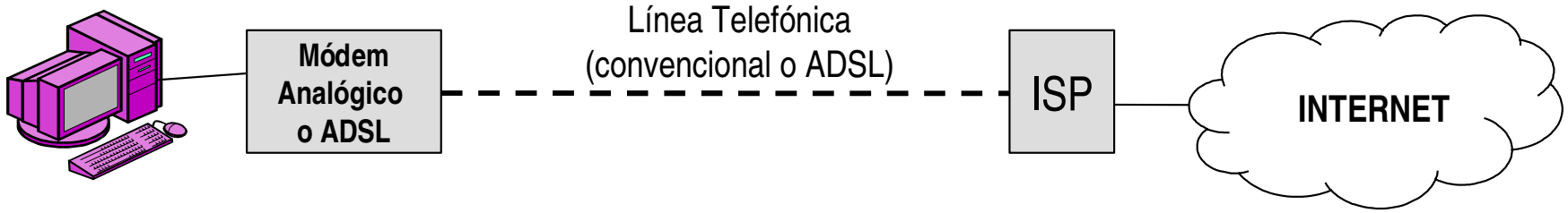


Tabla de traducción Port Forwarding

IP_Privada:Port_Servidor	IP_Pública:Port_Servidor
192.168.1.1:80 ←	← 175.20.12.1:80
192.168.1.2:25 ←	← 175.20.12.1:25
192.168.1.3:20 ←	← 175.20.12.1:20
192.168.1.3:21 ←	← 175.20.12.1:21

NAT: Traducción de Direcciones de Red

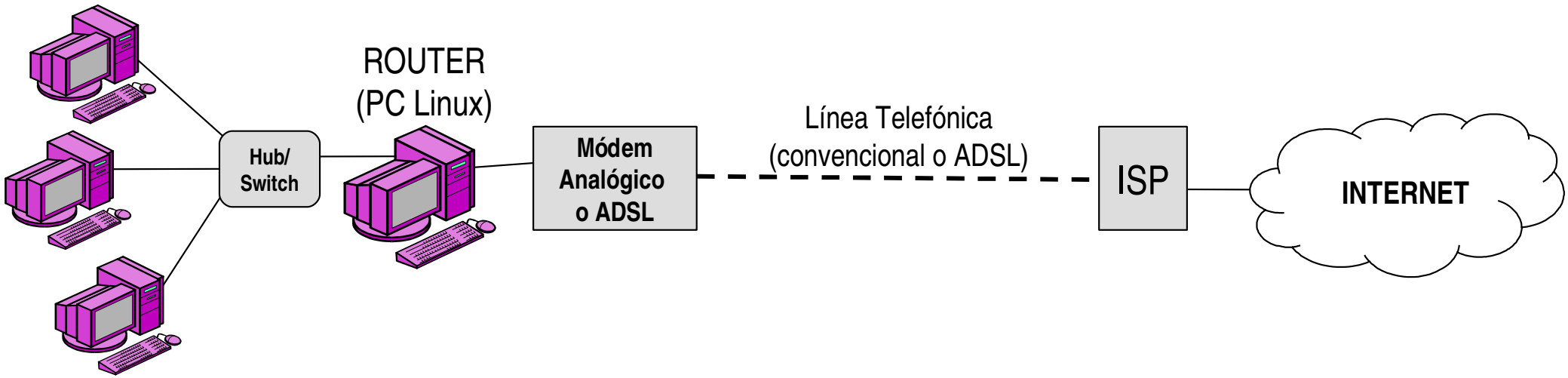
- **Configuraciones de redes de acceso telefónico y ADSL**
  - Configuración 1: módem convencional o módem ADSL
    - Un solo PC conectado a Internet (monopuesto)
      - No necesario NAT



  - Configuración 2: Router ADSL
    - Varios PCs conectados a Internet a través del router (multipuesto)
      - El propio router ADSL puede hacer NAPT y Port Forwarding (según configuración)

NAT: Traducción de Direcciones de Red

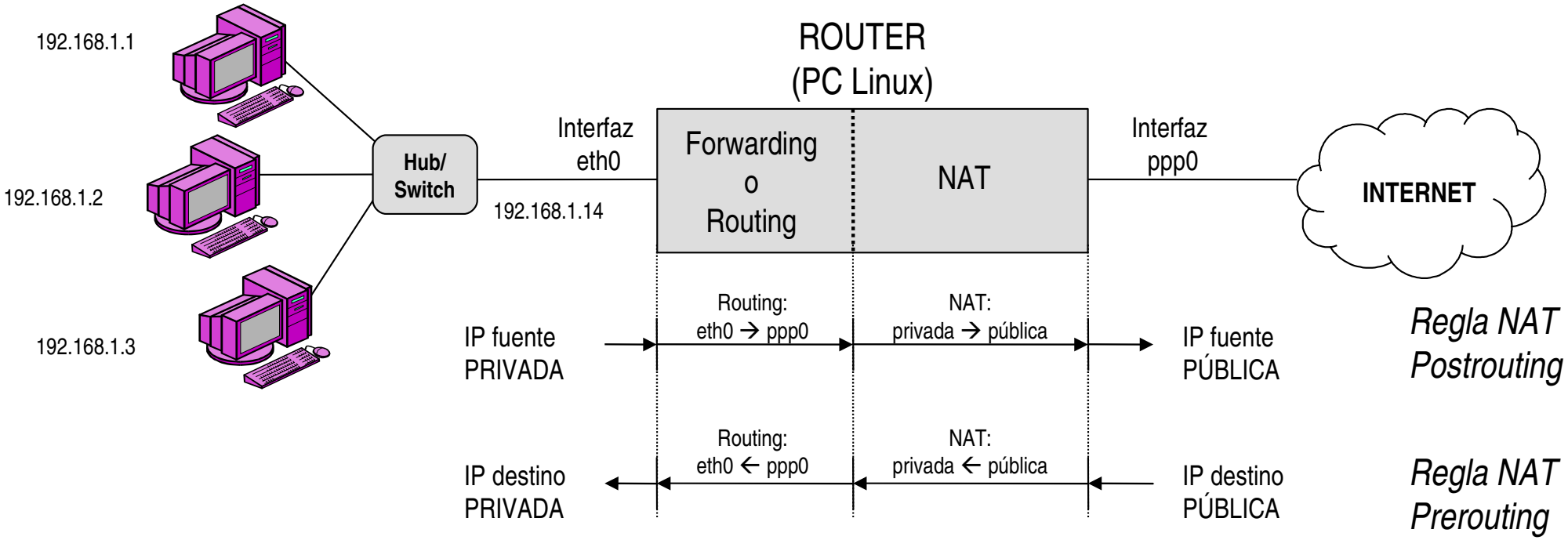
- **Configuraciones de redes de acceso telefónico y ADSL**
  - Configuración 3: módem convencional o ADSL + Router Linux
    - El PC conectado a Internet a través de módem se configura como un Router
      - Preferible S.O. Linux en el router
      - Implementación de NAPT y Port Forwarding mediante IPTables



NAT: Traducción de Direcciones de Red

Configuración de NAT en Linux con IPTables (i)

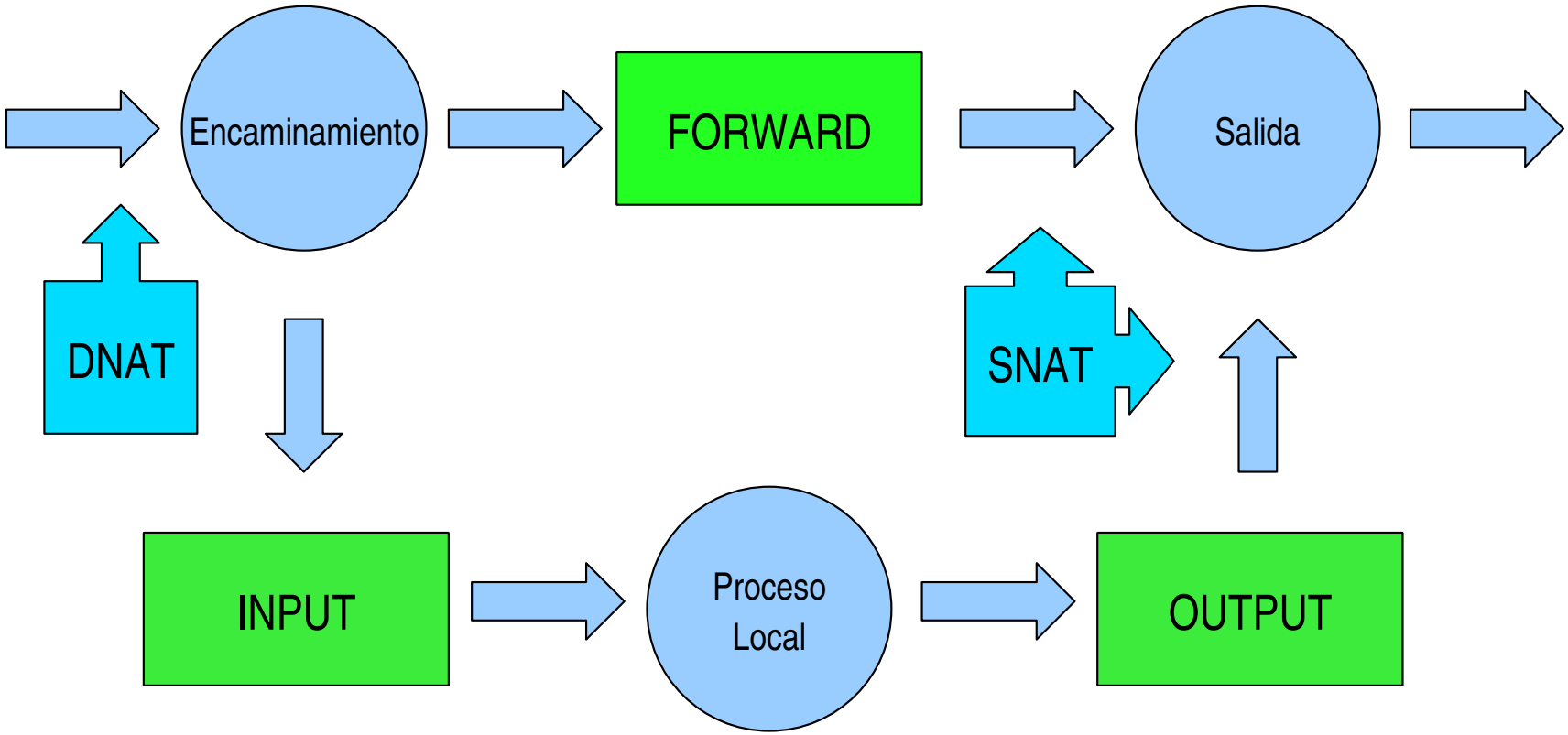
- Tipos de reglas NAT
  - POSTROUTING
    - Se usa para reglas donde se traduce la IP fuente: SNAT (Source NAT)
    - Ejemplo: NAPT o Masquerading
  - PREROUTING
    - Se usa para reglas donde se traduce la IP destino: DNAT (Destination NAT)
    - Ejemplo: Port Forwarding



NAT: Traducción de Direcciones de Red

Configuración de NAT en Linux con IPTables (ii)

- Aplicación de las reglas:



NAT: Traducción de Direcciones de Red

Reglas de NAT en Linux con IPTables (i)

- Regla MASQUERADE: NAPT con IP pública dinámica
  - Cuando la IP pública es dinámica se usa la operación denominada “MASQUERADE”
    - La regla MASQUERADE traduce cualquier IP privada a la IP pública del encaminador.
    - Esta IP pública puede cambiar de una conexión a otra, al ser dinámica.

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

- Regla SNAT: NAPT con IP pública fija
  - Cuando la IP pública es fija se usa la operación SNAT (Source NAT)
    - Se supone que la IP pública fija del router es la 175.20.12.1

```
iptables -t nat -A POSTROUTING -o ppp0 \
         -j SNAT --to 175.20.12.1
```

NAT: Traducción de Direcciones de Red

Reglas de NAT en Linux con IPTables (ii)

Regla DNAT: Port Forwarding

- Para realizar "port forwarding" se usa la operación DNAT (Destination NAT)
  - Se supone que en la red privada tenemos tres servidores:
    - Servidor Web (puerto 80) → IP privada: 192.168.1.1
    - Servidor SMTP (puerto 25) → IP privada: 192.168.1.2
    - Servidor FTP (puertos 20 y 21) → IP privada: 192.168.1.3

```
iptables -t nat -A PREROUTING -d 175.20.12.1 -p tcp --dport 80 \
     -j DNAT --to 192.168.1.1:80
iptables -t nat -A PREROUTING -d 175.20.12.1 -p tcp --dport 25 \
     -j DNAT --to 192.168.1.2:25
iptables -t nat -A PREROUTING -d 175.20.12.1 -p tcp --dport 20 \
     -j DNAT --to 192.168.1.3:20
iptables -t nat -A PREROUTING -d 175.20.12.1 -p tcp --dport 21 \
     -j DNAT --to 192.168.1.3:21
```

NAT: Traducción de Direcciones de Red

Reglas de NAT en Linux con IPTables (iii)

Campos de la regla

Opción/Ejemplo	Significado
-t nat	Tipo de regla: reglas de NAT
-t filter	Tipo de regla: reglas de filtrado (opción por defecto)
-A POSTROUTING	Añade regla a cadena de POSTROUTING
-A PREROUTING	Añade regla a cadena de PREROUTING
-i eth0	Filtrado por interfaz de red de entrada (eth0)
-o ppp0	Filtrado por interfaz de red de salida (ppp0)

Acciones

-j MASQUERADE	Realizar operación MASQUERADE
-j SNAT --to 175.20.12.1	Traducir dirección origen a 175.20.12.1
-j DNAT --to 192.168.1.1:80	Traducir dirección destino y puerto destino a 192.168.1.1:80

Configuración de un Router ADSL: 3Com Wireless 11g

- **Router ADSL 3Com Officeconnect Wireless 11g**
  - Router/Firewall ADSL con conexión Fast Ethernet y Wireless
    - 4 puertos Fast Ethernet
    - Soporte Wi-Fi para conexión de usuarios wireless

Configuración de un Router ADSL: 3Com Wireless 11g

- **Administración remota a través de navegado web**
  - Conectar al menos un PC al Router
    - Asignación de direcciones IP
      - La interfaz privada del router está configurada con la IP 192.168.1.1
      - El router asignará una dirección IP al PC mediante DHCP
    - Arrancar el navegador web e introducir la siguiente dirección

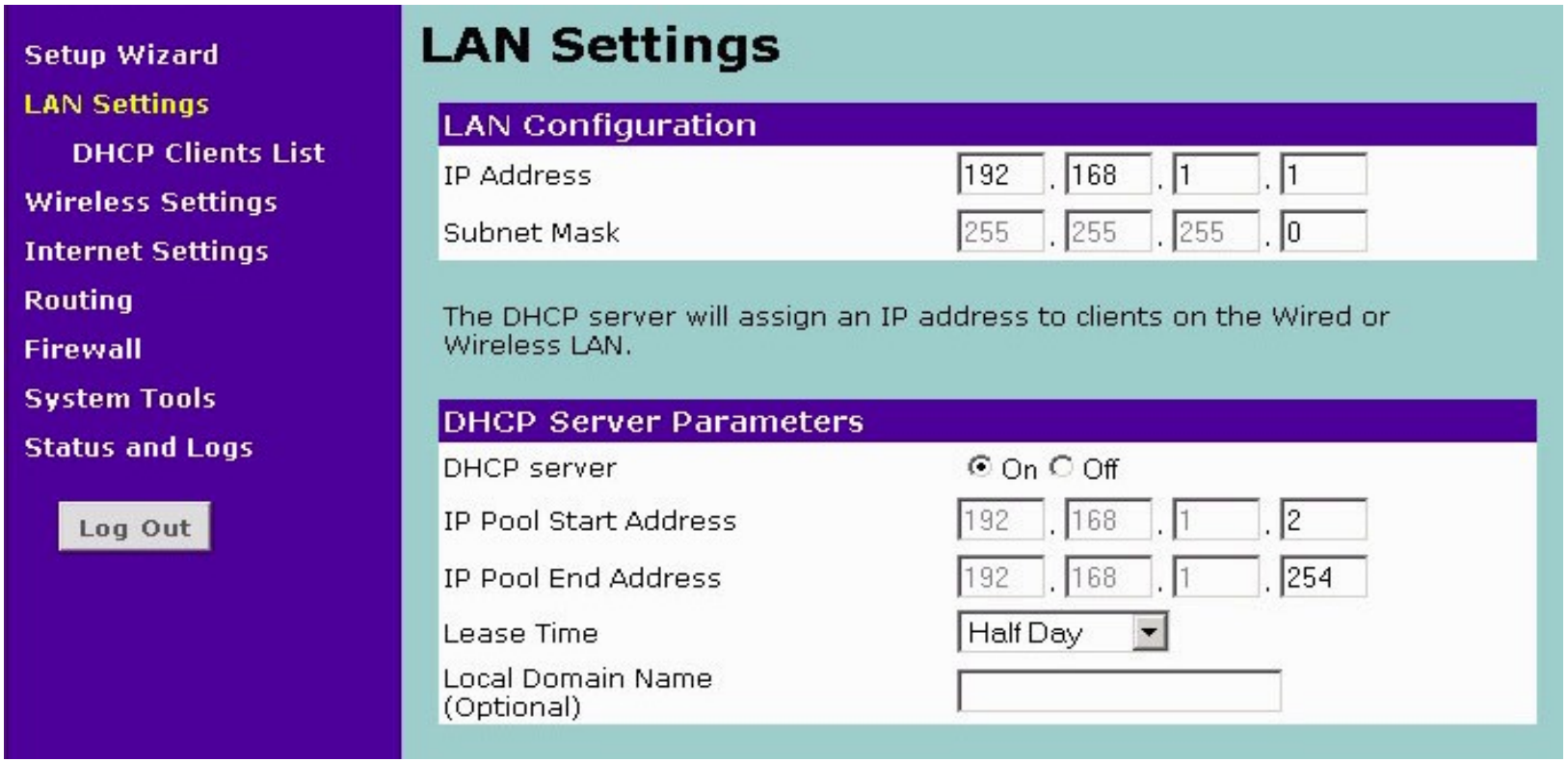
    - Se abrirá la ventana de configuración del router

Configuración de la conexión ADSL

- Seleccionar “Setup Wizard”
  - Seleccionar tipo de conexión (*Información proporcionada por el ISP*)
  - Introducir parámetros de la conexión (caso de PPPoE y PPPoA) (*Información proporcionada por el ISP*)
    - *VPI = ATM Virtual Path Identifier ; VCI = ATM Virtual Channel Identifier*

Configuración de IP del router y servicio DCHP

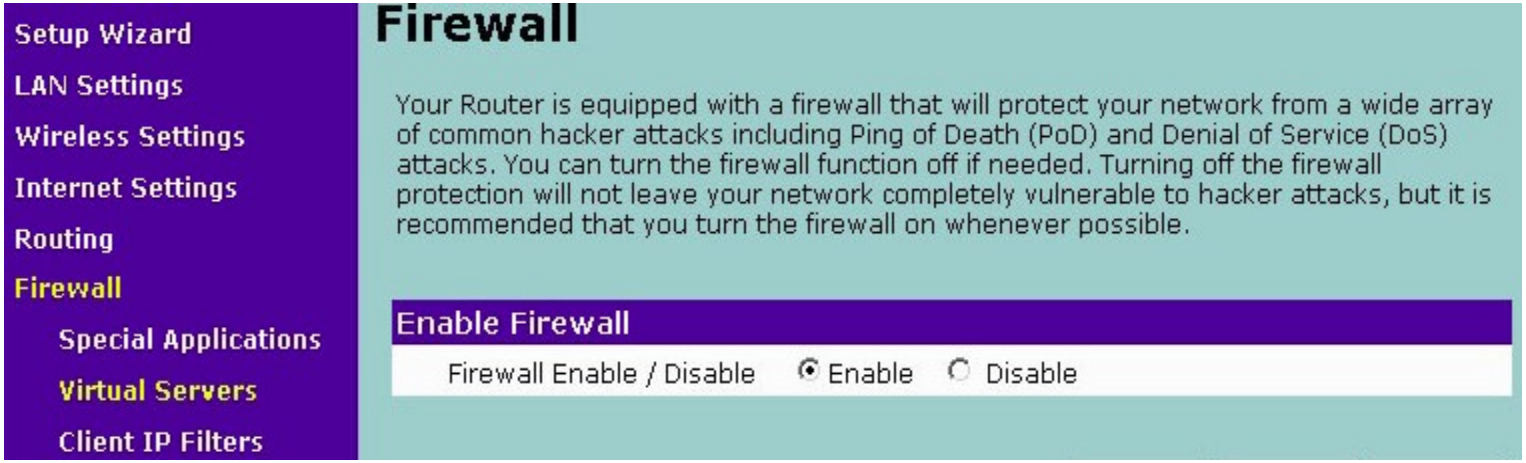
- Seleccionar “LAN Settings”
  - Introducir la IP privada del router y la máscara de red
    - Valores por defecto: IP = 192.168.1.1; Máscara = 255.255.255.0
  - Configurar el router como servidor DHCP
    - Indicar conjunto de direcciones a asignar dinámicamente
      - Valores por defecto: desde 192.168.1.2 hasta 192.168.1.254



Configuración de un Router ADSL: 3Com Wireless 11g

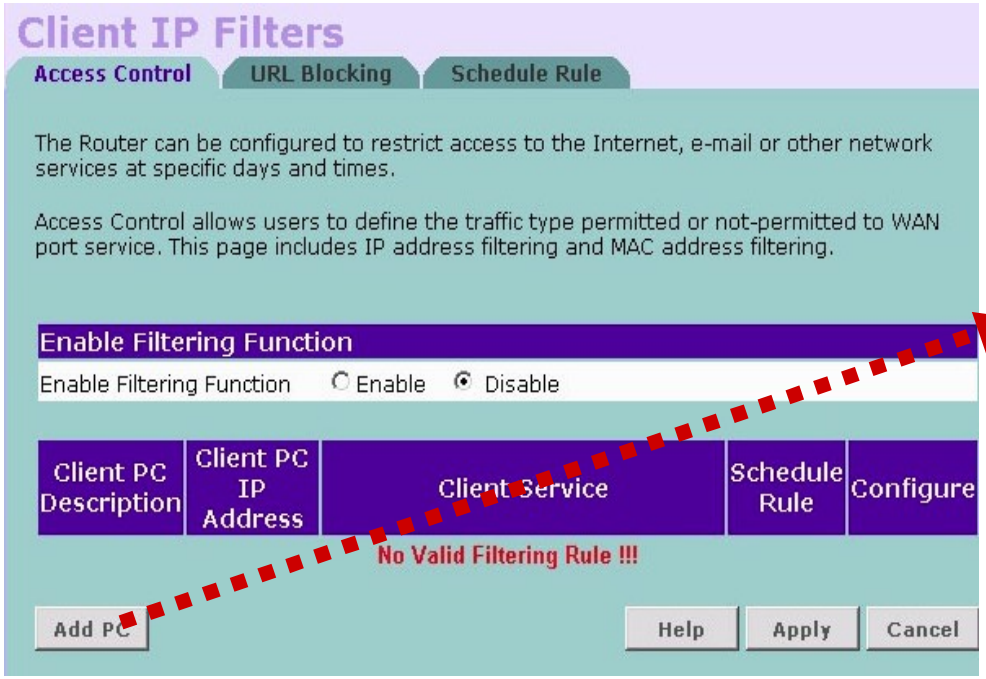
- **Visualizar lista de clientes DCHP**
  - Seleccionar "LAN Settings" → "DHCP Client List"

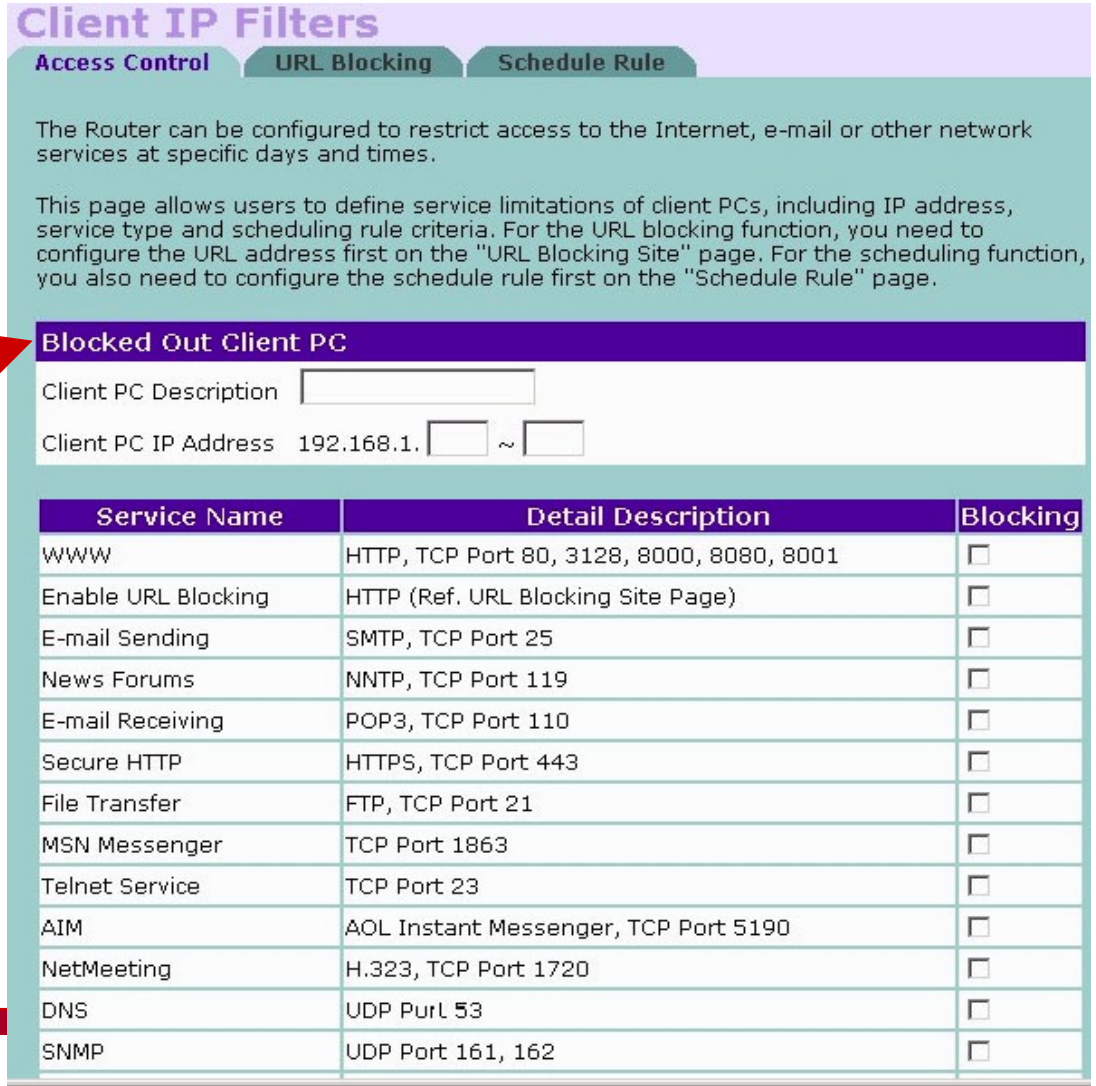
- **Activación del firewall**
  - Seleccionar "Firewall" para activar el firewall



Configuración de un Router ADSL: 3Com Wireless 11g

- **Filtrado de servicios de Internet para clientes**
  - Seleccionar "Firewall" → "Client IP Filters"
    - Permite limitar el acceso de los clientes a servicios de Internet
    - Para añadir nuevas reglas de filtrado, usar el botón "Add PC"

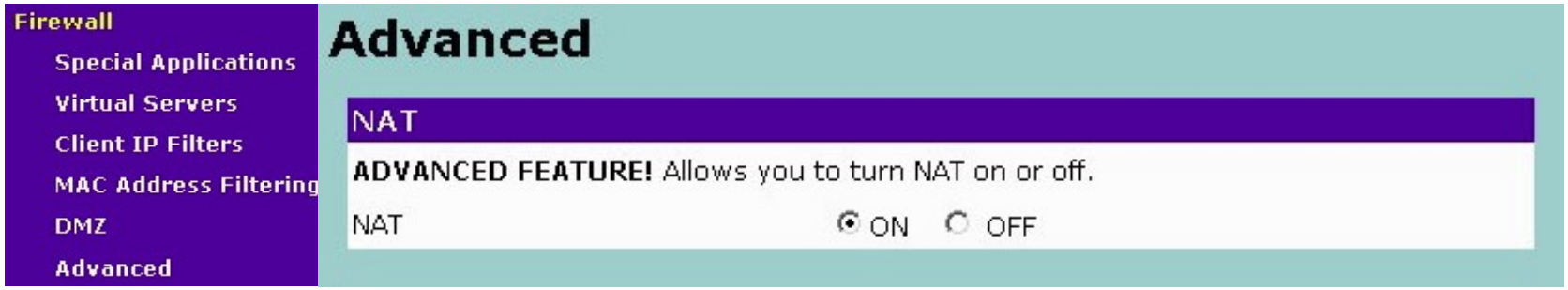




Activación de NAT

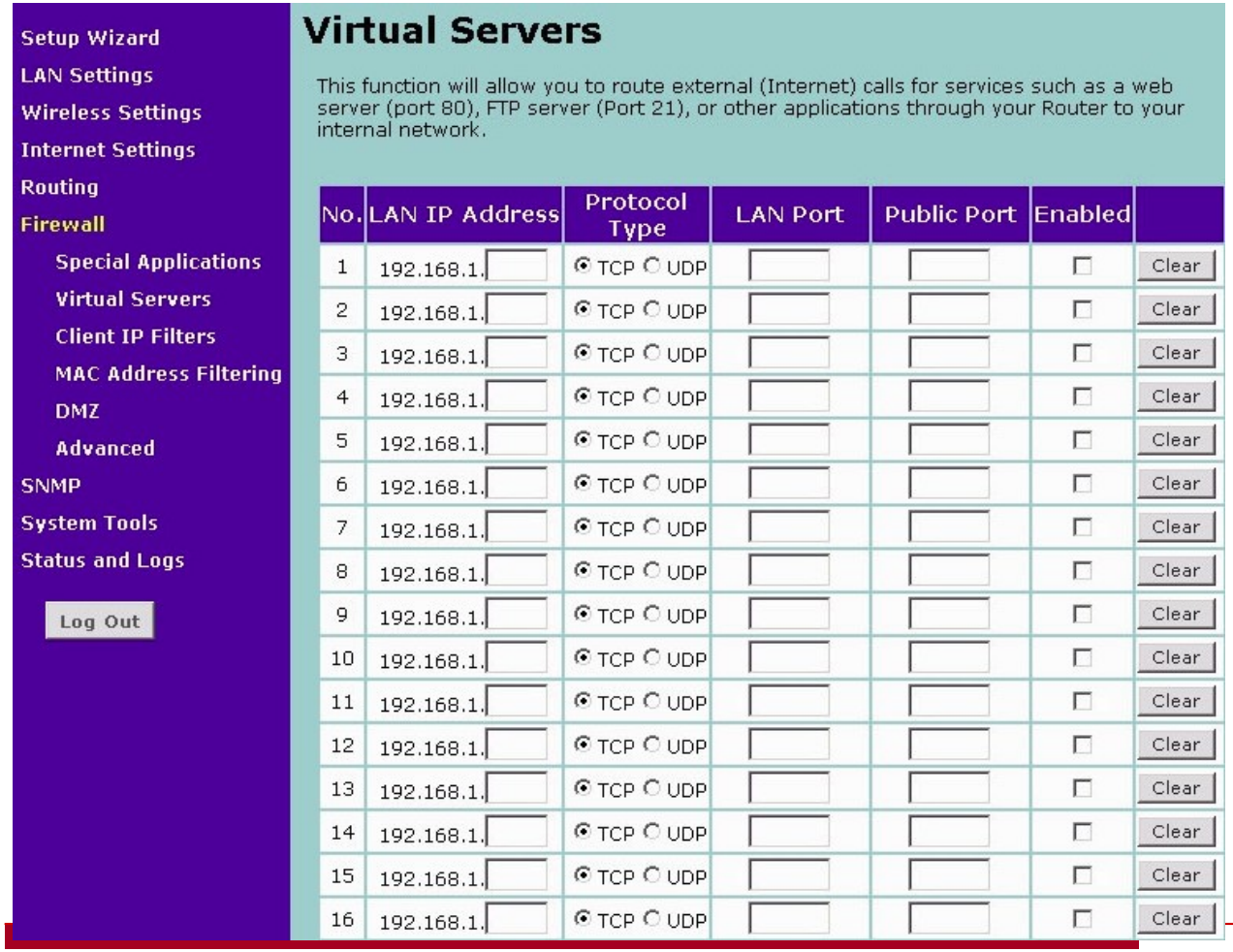
- La activación de NAT en el router es equivalente la operación de Masquerading en Linux
  - Permite compartir la conexión a Internet entre todos los PCs conectados al router
  - La IP privada de cada PC se traduce a la única IP pública disponible en la interfaz externa del router

- Para activar NAT, seleccionar "Firewall" → "Advanced"



Configuración de un Router ADSL: 3Com Wireless 11g

- **Port Forwarding o Virtual Servers**
  - Utilidad de Port Forwarding
    - Permite tener un servidor en la red interna visible desde Internet
      - El router redirige la conexión entrante a la IP privada del servidor
  - Seleccionar "Firewall" → "Virtual Servers"



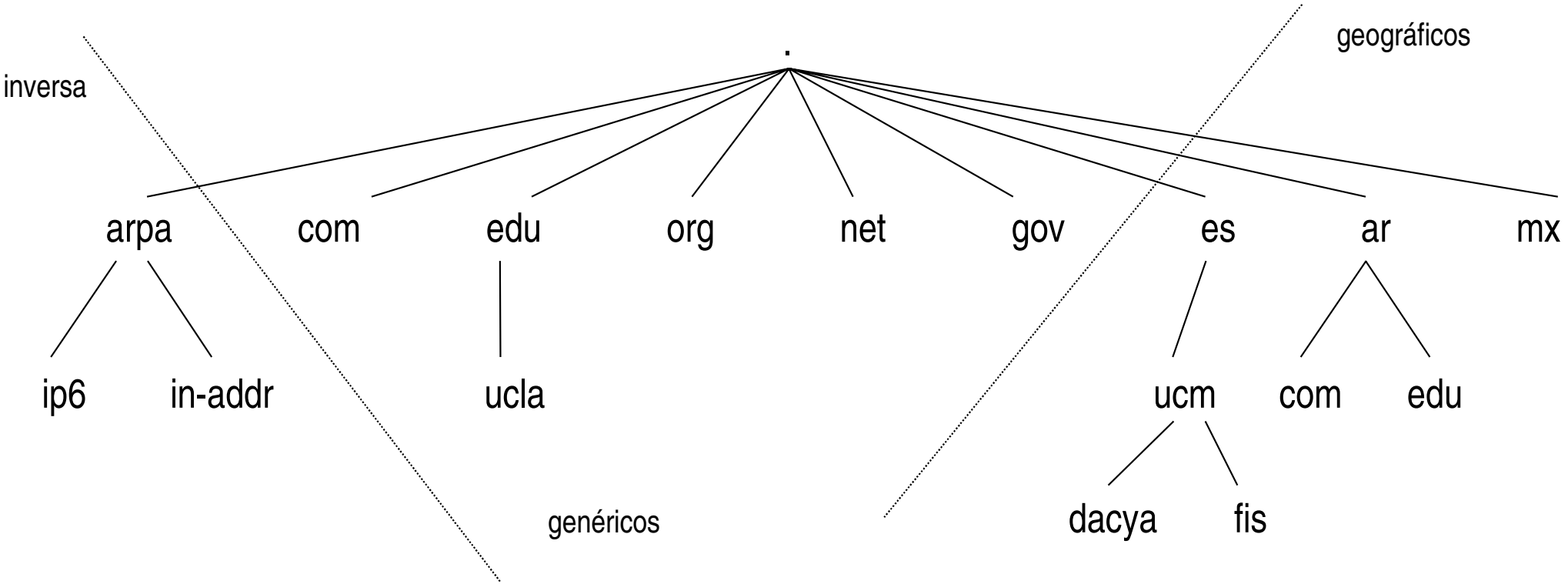
- Public Port
  - Puerto externo que recibe la conexión
- LAN IP Address
  - IP privada a la que se redirige la conexión
- LAN Port
  - Puerto del equipo privado al que redirige la conexión

Domain Name System

- Es una base de datos jerárquica y distribuída.
- Guarda información para hacer la correspondencia entre nombres simbólicos y direcciones IP.
- Las aplicaciones de Internet resuelven los nombres mediante un **resolutor** local.
- El resolutor envía las consultas a uno o más servidores de nombres hasta que obtiene la respuesta.

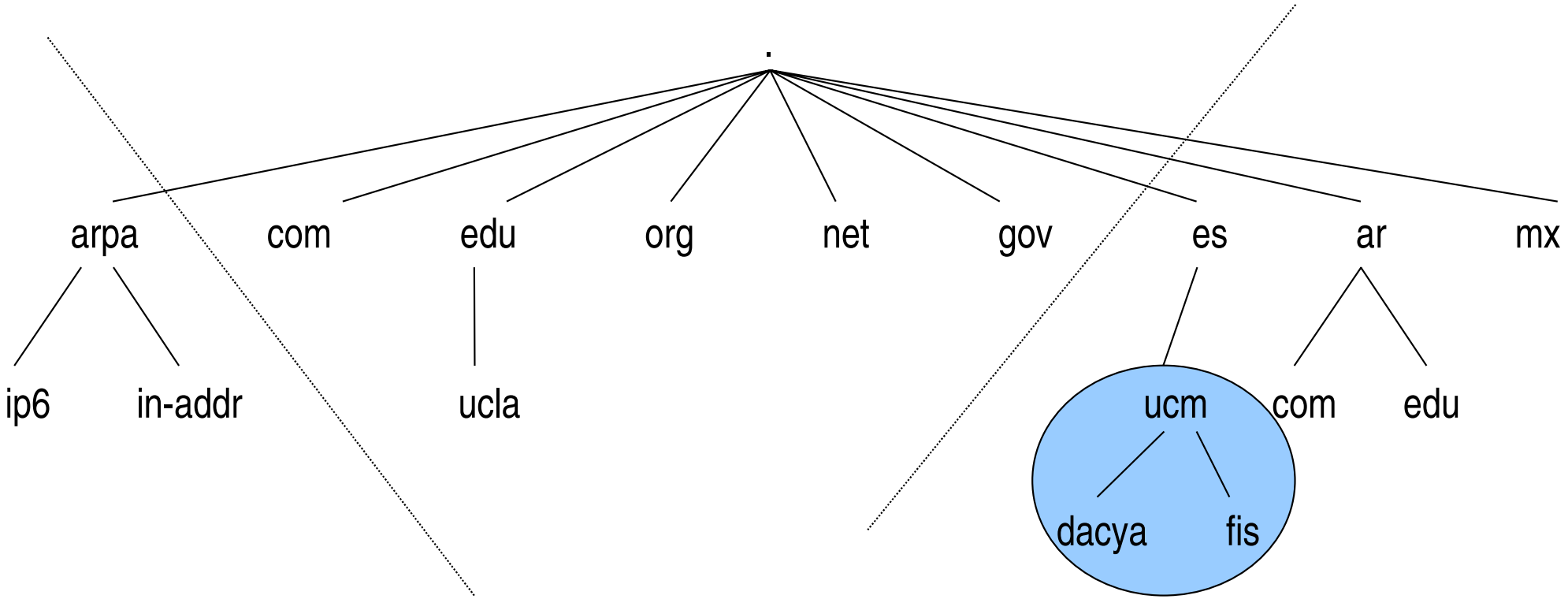
Estructura jerárquica

- Por ejemplo:
  - phobos.dacya.ucm.es.



Zonas

- Una zona es una sección (conexa) del árbol de nombres que está gestionada por una única autoridad.



Tipos de servidores

- Primario.
  - Mantiene la base de datos con la información sobre la zona.
- Secundario.
  - Posee una copia de la base de datos del servidor primario. Proporciona redundancia frente a posibles fallos.
- Cache.
  - No mantiene ninguna zona.
  - Sólo almacena en su memoria temporal las consultas que recibe de los clientes, para utilizarlas en caso de una nueva consulta.

- **Tipos de zona:**
  - **hint**: hace referencia al dominio raíz. El archivo de la definición de zona contiene las direcciones IP de los servidores raíz.
  - **master**: el servidor tiene autoridad sobre la zona. Es un servidor primario.
  - **slave**: el servidor es un servidor secundario. No mantiene un archivo de definición de zona, sino que lo copia desde el servidor primario.

```
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
zone "casa.loc" in {
        type master;
        file "master/casa.loc.zone";
        notify no;
        allow-transfer { fec0::/16; };
        allow-query { fec0::/16; fe80::/16; 192.168.1.0/24; 127.0.0.1; };
        };
```

Configuración en Linux

- Para una máquina cliente:
  - Modificar el archivo /etc/resolv.conf
  - search -> añade automáticamente ese dominio a los nombres incompletos.
  - nameserver : dirección IP del servidor de nombres. Puede haber varios. Si hay más de uno, se van preguntando en orden.

```
# Archivo /etc/resolv.conf
search midominio.edu.ar
nameserver 192.168.1.1
nameserver 192.168.1.2
```

Configuración en Linux

- Para una máquina servidor:
  - Es necesario instalar el programa bind9 (Berkeley Internet Name Domain version 9).
  - Archivos de configuración: generalmente en /etc/bind/
  - Archivos de zona: en Debian suelen estar en /var/cache/bind
  - Definición de zona (en /etc/bind/named.conf.local):

```
zone "casa.loc" in {
        type master;
        file "master/casa.loc.zone";
        notify no;
        allow-transfer { fec0::/16; };
        allow-query { fec0::/16; fe80::/16; 192.168.1.0/24; 127.0.0.1; };
        //allow-update { key casa.loc; };
};
```

Definición de zona

```
$TTL 1D
;$ORIGIN casa.loc.
@       IN SOA  caramon.casa.loc.   root.caramon.loc. (
                        2004072022      ; serial (d. adams)
                        1D              ; refresh
                        4H              ; retry
                        1D              ; expiry
                        1H )            ; minimum
@               NS                      caramon
@               MX              10      mail
ns              CNAME                   caramon
gateway         A               192.168.1.1
caramon         A               192.168.1.2
caramon         AAAA                    fec0::80:1
raistlin        A               192.168.1.7
raistlin        AAAA                    fec0::80:7
$GENERATE 2-99  local-${0,2,x} A                192.168.1.$
$GENERATE 2-99  local-${0,2,x} AAAA             fec0::1:$
```

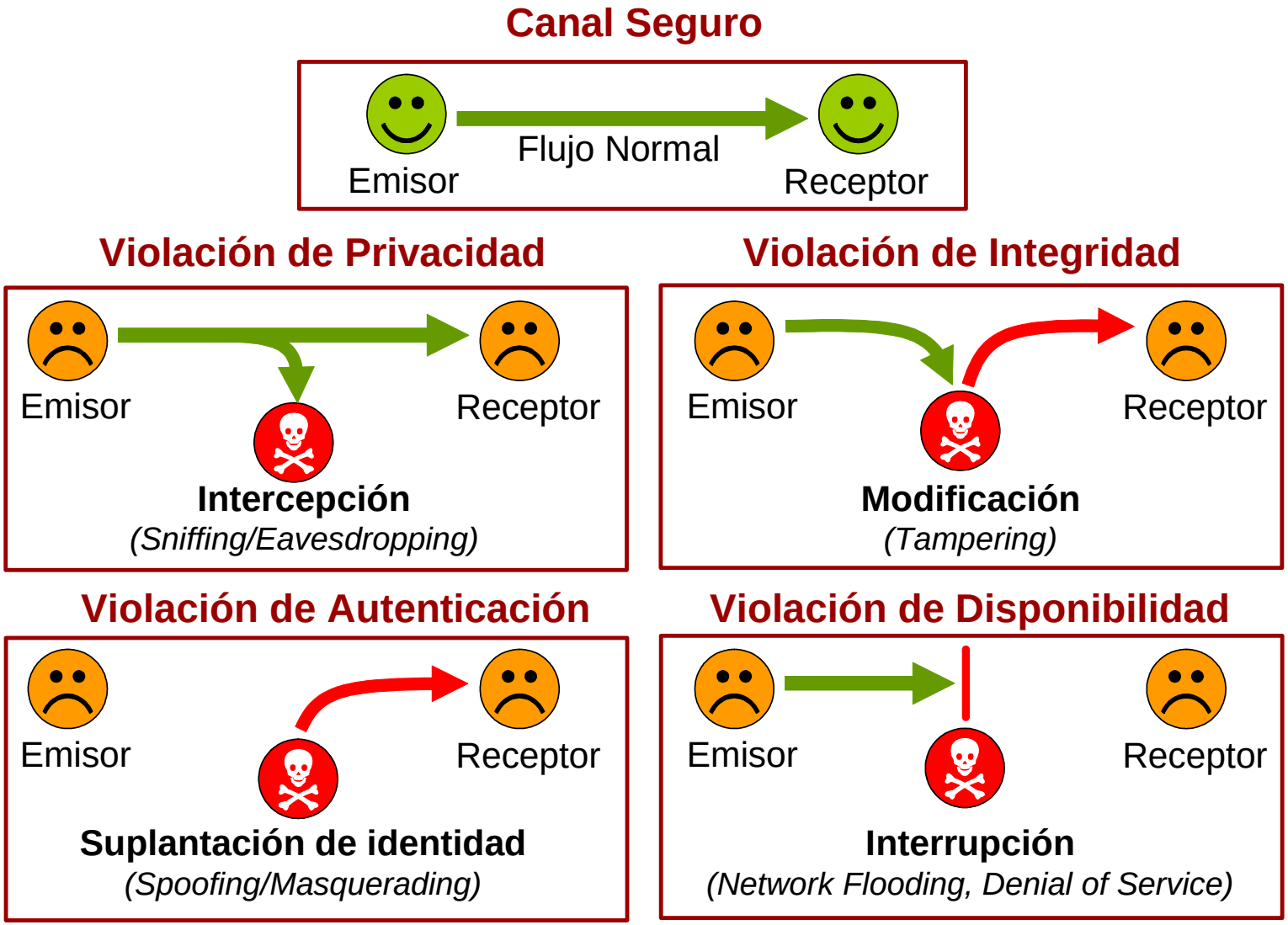
Contenidos

- Tema 13. Redes de área local y arquitectura de protocolos TCP/IP
- Tema 14. La interfaz de red Ethernet
- Tema 15. Protocolo de resolución de direcciones (ARP)
- Tema 16. Configuración de IPv4. Redes y subredes.
- Tema 17. Configuración de routers y protocolos de routing
- Tema 18. Configuración de protocolos de transporte (TCP y UDP): puertos y servicios
- Tema 19. Conceptos avanzados de redes: DHCP, IPv6, Firewalls, …..
- **Tema 20. Seguridad de la red**

Aspectos clave de la seguridad

- Confidencialidad
  - Debe garantizarse que la información enviada sólo puede ser leída por personas debidamente autorizadas.
- Integridad
  - Debe garantizarse que la información no puede ser alterada en el transcurso hacia su destino.
- Autenticación
  - Debe garantizarse que los participantes en el intercambio de información son realmente quienes dicen ser
- Disponibilidad
  - Debe garantizarse la información está disponible en el momento adecuado para las personas autorizadas

Clasificación de las amenazas



Amenzas a un sistema: Escaneo de puertos

Objetivo

- Recolectar información de un sistema acerca de:
  - **Listado de servicios de red abiertos y cerrados**
    - Útil para determinar si la máquina ofrece servicios de tipo Telnet, FTP, rlogin, etc.
  - **Reconocimiento del sistema operativo**
    - Cada sistema operativo inicializa los campos de las cabeceras TCP, UDP e IP con unos valores específicos. Esto se denomina **Fingerprint** (huella dactilar)
    - Mediante un escaneo de puertos, analizando las cabeceras de los paquetes que devuelve la máquina atacada, se puede averiguar el tipo y la versión del S.O.
  - **Búsqueda de exploits**
    - Busca aquellos puertos donde puede haber instalado un Troyano, una puerta trasera, etc.
    - Cada Troyano escucha por un puerto específico
  - **Búsqueda de vulnerabilidades**
    - Busca aquellos puertos correspondientes a servicios con fallos de programación conocidos
    - Buffer overflows, fallos de configuración, etc.
- El escaneo de puertos por sí mismo no es peligroso
  - Es el paso previo para lanzar posteriormente otros tipos de ataques

Técnicas de escaneo de puertos TCP (1)

Envío de SYN (técnica *connect*)

- La máquina atacante envía un segmento TCP SYN
- La máquina atacada
  - Si el puerto está abierto, devuelve un segmento SYN-ACK
  - si el puerto está cerrado, devuelve un segmento RST (Reset)
- Ventajas
  - Técnica muy simple de implementar
- Desventajas
  - Muy fácil de detectar por la máquina atacada, ya que cualquier el SO registra todos los intentos de conexión a cualquier puerto que esté abierto
  - La mayoría de los firewalls filtran los segmentos TCP SYN

Técnicas de escaneo de puertos TCP (2)

Envío de SYN-ACK (técnica *stealth*)

- La máquina atacante envía un segmento TCP SYN-ACK (confirmación de conexión)
- La máquina atacada
  - Si el puerto está abierto, no hace nada, simplemente descarta el paquete ya que lo considera erróneo
  - Si el puerto está cerrado devuelve un segmento RST
- Ventaja
  - Difícil de detectar, ya que no deja rastro en la máquina atacada

Técnicas de escaneo de puertos UDP

Técnica habitual

- La máquina atacante envía un **segmento UDP vacío**
- La máquina atacada actuará de la siguiente forma
  - Si el **puerto está abierto**, pueden ocurrir dos cosas
    - La máquina atacada devuelve un mensaje de error
    - La máquina atacada descarta el paquete recibido y no devuelve nada
  - Si el **puerto está cerrado**
    - La máquina atacada devolverá un paquete ICMP Port Unreachable

PRACTICA (1)

- Utilizar la utilidad hping para generar paquetes TCP “a medida”
  - Sintáxis: **`hping ip_destino –FLAGS –p port_destino`**

    FLAGS= –A → ACK
    –S → SYN
    –F → FIN
    –R → RST
    –P → PSH
- Realizar un escaneo de puertos TCP usando la utilidad hping
  - Escaneo mediante técnica **connect**
    - **`hping ip-destino –S –p ++1`**
    - Con **ethereal** ver los paquetes de respuesta
  - Escaneo mediante técnica **stealth**
    - **`hping ip-destino –AS –p ++1`**
    - Con **ethereal** ver los paquetes de respuesta

PRACTICA (2)

- Utilizar la utilidad hping para generar paquetes UDP “a medida”
  - Sintáxis: **`hping -2 ip_destino –p port_destino`**

- Realizar un escaneo de puertos UDP usando la utilidad hping
  - Escaneo mediante técnica **connect**
    - **`hping -2 ip-destino –p 1`**
    - Usando **^z** vamos aumentando el número de puerto
    - Con **ethereal** ver los paquetes de respuesta

PRACTICA (3)

- Utilizar la utilidad nmap realizar un escaneo de puertos TCP y UDP
  - Ver ayuda de nmap para aprender las distintas opciones

PRACTICA (4)

- Utilizar la utilidad nessus realizar un escaneo de vulnerabilidades

Amenzas a la disponibilidad: Denegación de servicios

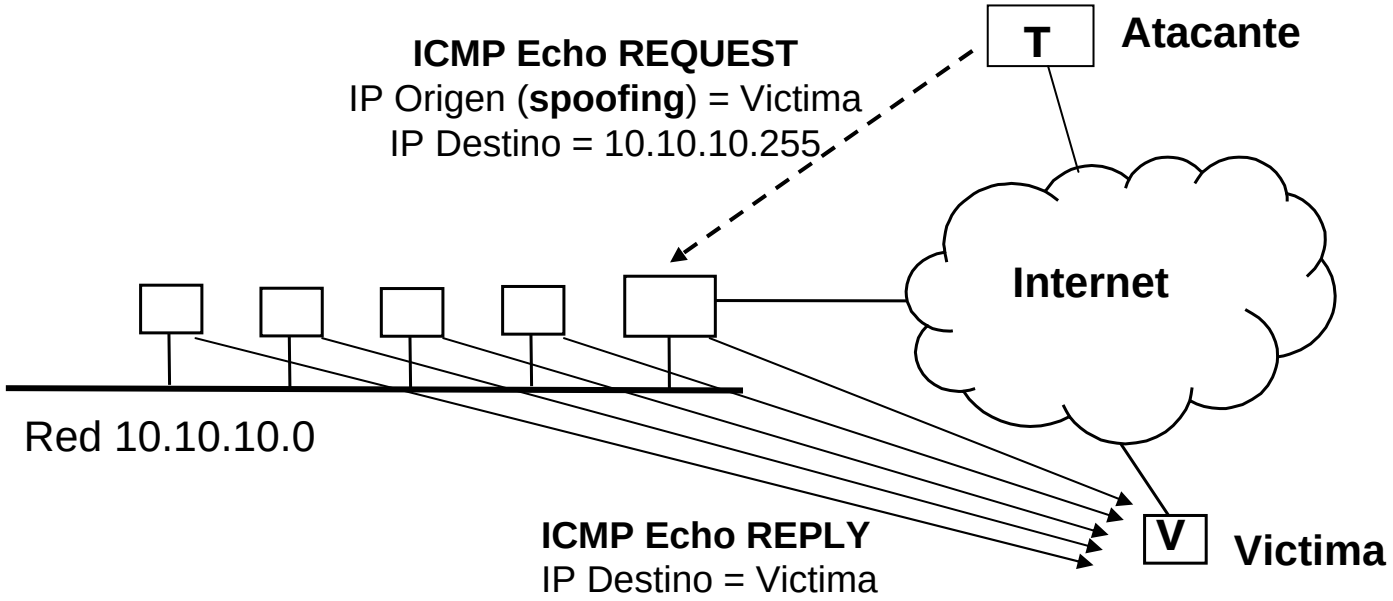
DoS = Denial os Service

- Algunas Técnicas DoS conocidas
  - Envenenamiento de la cache ARP (ARP poissoning)
  - Inundación de Pings (Ping Flood o Smurf)
  - Ping de la muerte (Ping fo Dead)
  - Bombas de Redirección
  - Fragmentación de IP (Teardrop)
  - Inundación por conexiones TCP semiabiertas (TCP SYN Flood)
  - Agotamiento de conexiones TCP (TCP Conection Exhaustion)
  - Inundación de UDP (UDP flood)
  - Ataque DoS Distribuido (DDoS)
  - Etc.
- Técnicas para combatir el DoS
  - Cerrar todos los puertos del sistema que no sean necesarios
  - Proteger los sistemas y la red mediante un FIREWALL

Ataques DoS (1)

Inundación de Pings (ICMP flood o Smurf)

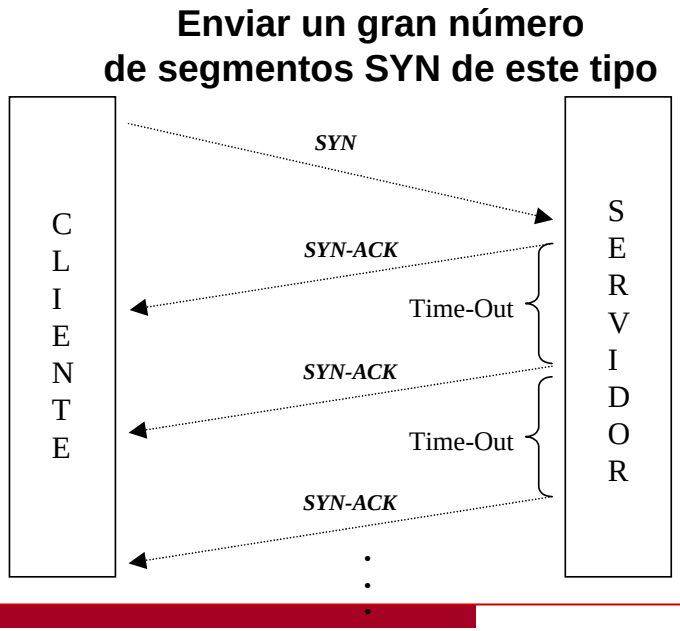
- El ataque por inundación de ping consiste en enviar un ping con tamaño de mensaje grande enviado a la dirección broadcast de una red
  - Esto puede llegar a colapsar la red
- **Método reflectivo** (RDoS=Reflective DoS)
  - Atacar una máquina haciendo un ICMP Flood a una red, usando una IP origen falsa (SPOOFING)



Ataques DoS (2)

Inundación de TCP SYN (SYN Flood)

- Consiste en saturar un puerto TCP servidor con un gran número de solicitudes de conexión (SYN) sin completar
- El servidor sufre dos problemas:
  - Consume tiempo enviando una y otra vez los SYN-ACK de las múltiples conexiones parcialmente establecidas
  - Puede llegar al límite de conexiones abiertas, y a partir de ese momento empezaría a descartar nuevas solicitudes de conexión.



Ataques DoS (3)

Inundación de UDP (UDP Flood)

- Consisten en enviar múltiples paquetes UDP a un puerto UDP abierto la máquina atacada
- Cada paquete UDP enviado consume un socket (ocupa memoria) y esto va reduciendo los recurso de la máquina atacada

Ataques DoS Distribuidos (DDoS)

- Los ataques DDoS suelen ser ataques de agotamiento de conexiones que se realizan simultáneamente desde múltiples máquinas sobre un mismo servidor
- Las máquinas atacantes son, por lo general, también víctimas de un Troyano
  - El Troyano infecta miles de máquinas repartidas por todo el mundo
  - Este troyano se activa un día y una hora determinada y realiza un ataque contra un servidor específico (Yahoo, Amazon, etc.)
  - De esta forma, el ataque se realiza de forma sincronizada desde miles de máquinas
    - Consiguen provocar un agotamiento de conexiones en el servidor y por tanto una denegación de servicio

PRACTICAS

Ataque ICMP Flood (Smurf)

- Atacar una máquina víctima usando la utilidad **hping**
  - Se envía un ICMP Echo Request a toda la red (p. ej. 192.168.1.255)
  - Se usa como dirección origen la IP de la víctima (**SPOOFING**)

```
hping -1 -a IP_victima 192.168.1.255
```

Ataque TCP SYN Flood

- Atacar el puerto 23 de una máquina víctima usando la utilidad **hping**
  - Se envían TCP SYN a la máquina víctima a un ritmo muy elevado (cada 100 ms)
  - Se usa una IP origen falsa (SPOOFING) para no saturar nuestra propia máquina con las respuestas

```
hping -a IP_origen_falsa IP_victima –S  -p 23 –i u100
```

Amenzas a la confidencialidad: Sniffers de red (1)

- Herramientas capaces de capturar toda la información que circula a través de la tarjeta red
  - Un sniffer activa la tarjeta de red en **modo promiscuo**
    - Una tarjeta en **modo normal** sólo captura el tráfico que va dirigido a ella o el tráfico broadcast
    - Una tarjeta en **modo promiscuo** captura todo el tráfico que pasa por la red
- Utilidades legítimas de los sniffers
  - Resolución de problemas de la red
    - Analizar el tráfico de la red y detectar posible problemas de configuración:
      - Cables de red desconectados
      - Tablas de rutas mal configuradas
  - Detección de intrusos
    - Sniffers ubicados estratégicamente para monitorizar el tráfico de la red y detectar posible intrusos

Amenzas a la confidencialidad: Sniffers de red (2)

- Utilidades ilegítimas de los sniffers
  - Capturar nombres de usuarios y contraseñas de acceso a un sistema
  - Capturar información privada que viaja por la red
    - Correo electrónico y ficheros transmitidos por FTP, NFS, etc.
    - Nº de tarjetas de crédito, PINs de acceso a cuentas bancarias, etc
- Técnicas para combatir los sniffers
  - **Uso de comunicaciones encriptadas**
    - Sustituir protocolos inseguros (Telnet, FTP, etc.) por protocolos encriptados (SSH, Telnet Seguro)
    - Utilizar un sistema de encriptación de clave pública para
      - Envío de correo electrónico
      - Transacciones de comercio electrónico, banca electrónica, etc.
  - **Uso de redes con switch**
    - En este tipo de redes el campo de acción del sniffer es mucho más limitado
  - **Uso de herramientas anti-sniffers**
    - Existen herramientas para detectar sniffers y tarjetas en modo promiscuo

Amenzas a la confidencialidad: Sniffers de red (3)

- Utilidades ilegítimas de los sniffers
  - Capturar nombres de usuarios y contraseñas de acceso a un sistema
  - Capturar información privada que viaja por la red
    - Correo electrónico y ficheros transmitidos por FTP, NFS, etc.
    - Nº de tarjetas de crédito, PINs de acceso a cuentas bancarias, etc
- Técnicas para combatir los sniffers
  - **Uso de comunicaciones encriptadas**
    - Sustituir protocolos inseguros (Telnet, FTP, etc.) por protocolos encriptados (SSH, Telnet Seguro)
    - Utilizar un sistema de encriptación de clave pública para
      - Envío de correo electrónico
      - Transacciones de comercio electrónico, banca electrónica, etc.
  - **Uso de redes con switch**
    - En este tipo de redes el campo de acción del sniffer es mucho más limitado
  - **Uso de herramientas anti-sniffers**
    - Existen herramientas para detectar sniffers y tarjetas en modo promiscuo

PRACTICAS

- Usar **ethereal** para ver nombres de usuarios y contraseñas de las aplicaciones **telnet** y **ftp**

Técnicas de criptografía o cifrado de los datos (1)

- Utilidad
  - Permiten garantizar la **confidencialidad** de los datos
- Principales técnicas de criptografía o cifrado de la información
  - Encriptación por clave simétrica
  - Encriptación por clave pública
  - Encriptación por clave secreta compartida
  - Encriptación por clave de sesión

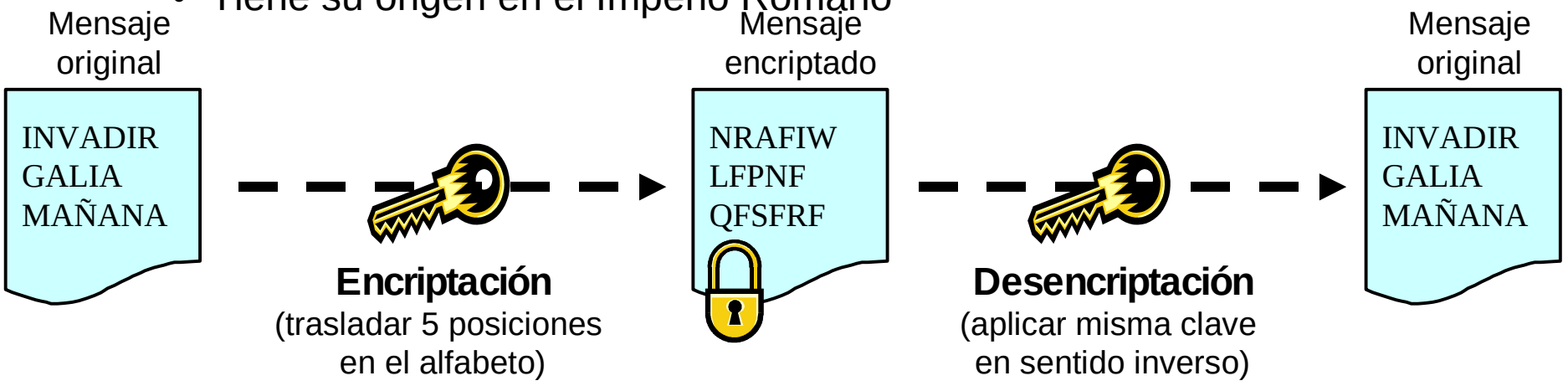
Técnicas de criptografía o cifrado de los datos (2)

¿Qué es la criptografía?

- Técnica que permite
  convertir un texto legible o plano ("*plain text*")
  en un texto encriptado o cifrado ("*cipher text*")
  a través de la aplicación del un algoritmo de cifrado
  basado en una clave criptográfica (LLAVE)

El origen de la criptografía

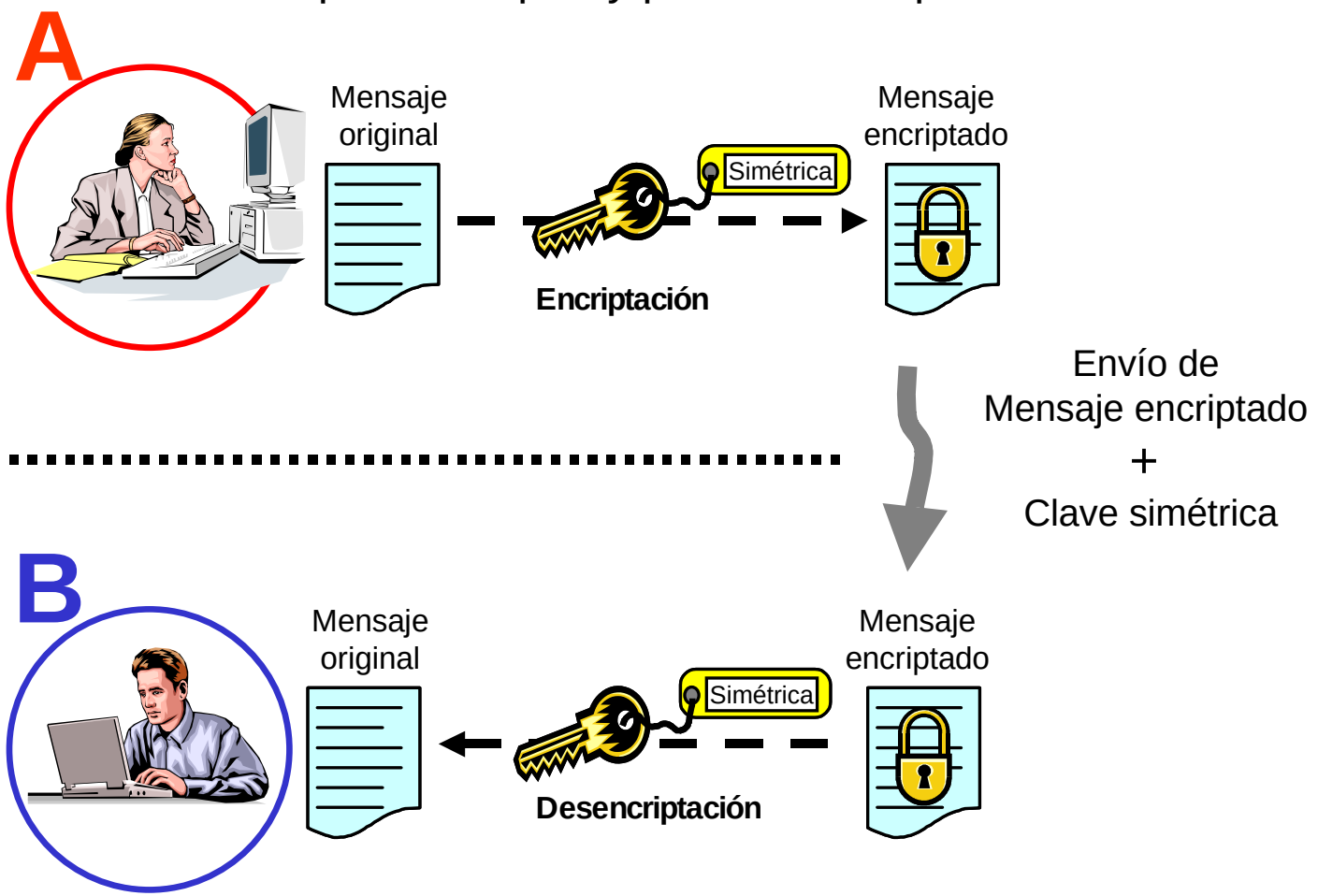
- Tiene su origen en el Imperio Romano



Técnicas de criptografía o cifrado de los datos (3)

Criptografía de clave simétrica (i)

- Usa la misma clave para encriptar y para desencriptar



Técnicas de criptografía o cifrado de los datos (4)

Criptografía de clave simétrica (ii)

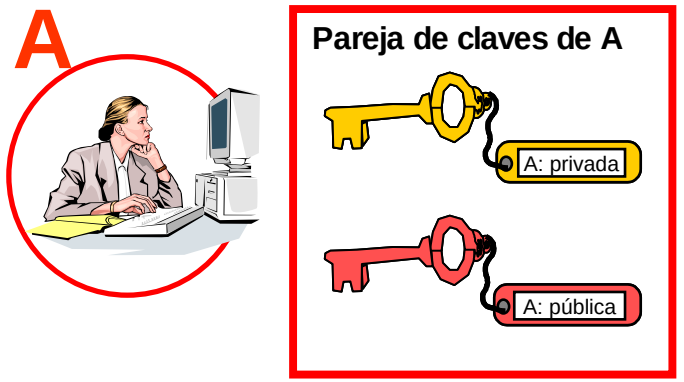
- **Ventajas**
  - Mecanismo muy rápido
- **Problemas**
  - Necesidad de distribuir la clave
  - Si alguien consigue leer el mensaje y la clave podrá descifrar el mensaje

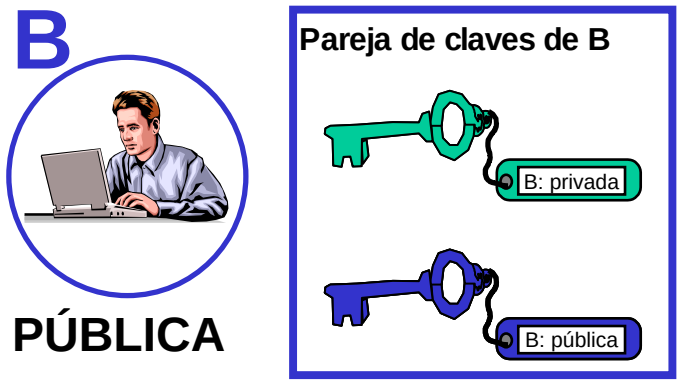
- **Ejemplos de algoritmos de clave simétrica**
  - DES (Data Encryption Standard)
  - 3DES (Triple DES)
  - RC2, RC4, RC5 (Ron´s Code, version 2, 4, 5)
  - AES (Advanced Encription Standard)

Técnicas de criptografía o cifrado de los datos (5)

Criptografía de clave asimétrica o pública (i)

- Cada usuario del sistema ha de poseer una pareja de claves:
  - **Clave privada:** será custodiada por su propietario y no se dará a conocer a nadie
  - **Clave pública**: será conocida por todos los usuarios



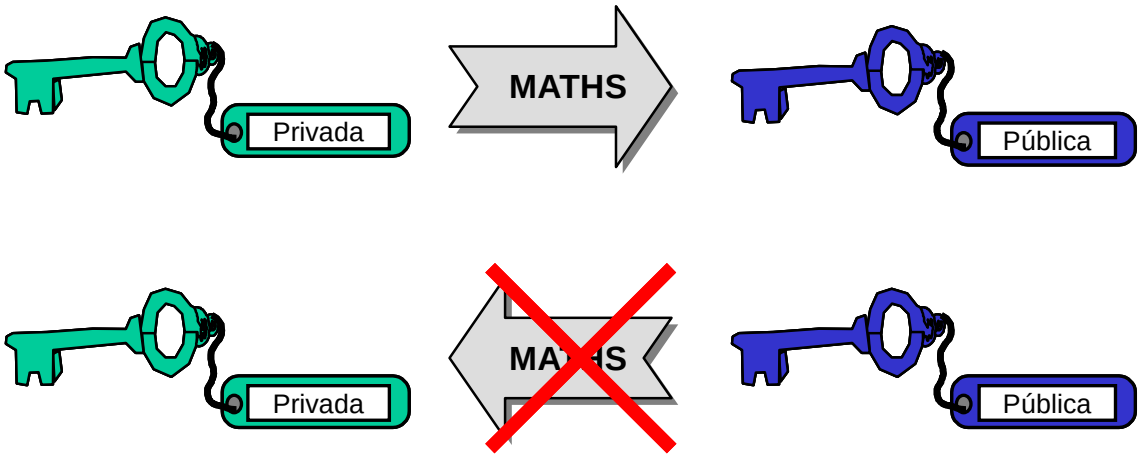


- La pareja de claves es **complementaria**:
  - Mensaje cifrado con **CLAVE PRIVADA**
    → Sólo puede ser descifrado con **CLAVE PÚBLICA**
  - Mensaje cifrado con **CLAVE PÚBLICA**
    → Sólo puede ser descifrado con **CLAVE PRIVADA**

Técnicas de criptografía o cifrado de los datos (6)

Criptografía de clave asimétrica o pública (ii)

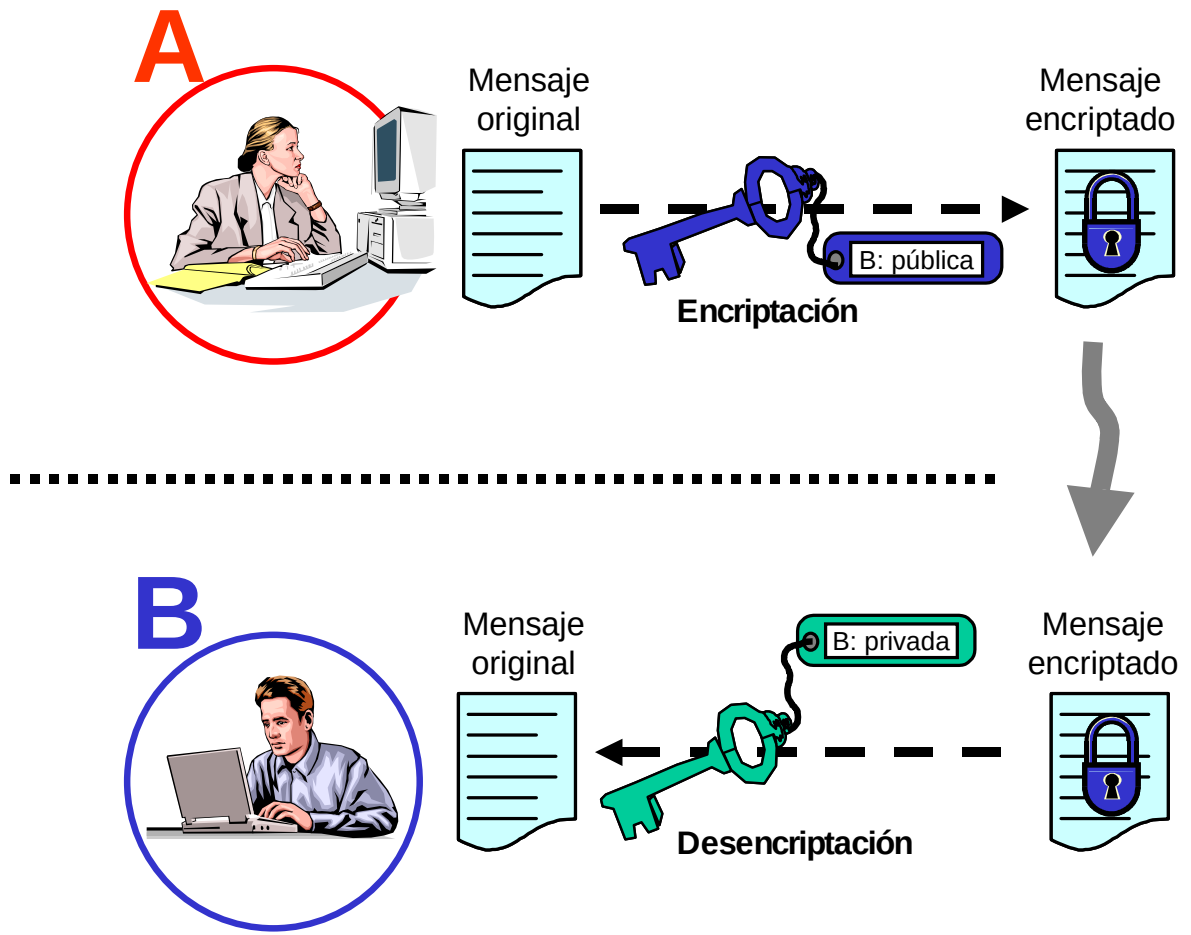
- **Relación matemática entre las claves**
  - La clave pública se genera matemáticamente a partir de la privada
  - Sin embargo, obtener la clave privada a partir de la pública es matemáticamente y computacionalmente imposible



Técnicas de criptografía o cifrado de los datos (7)

Criptografía de clave asimétrica o pública (iii)

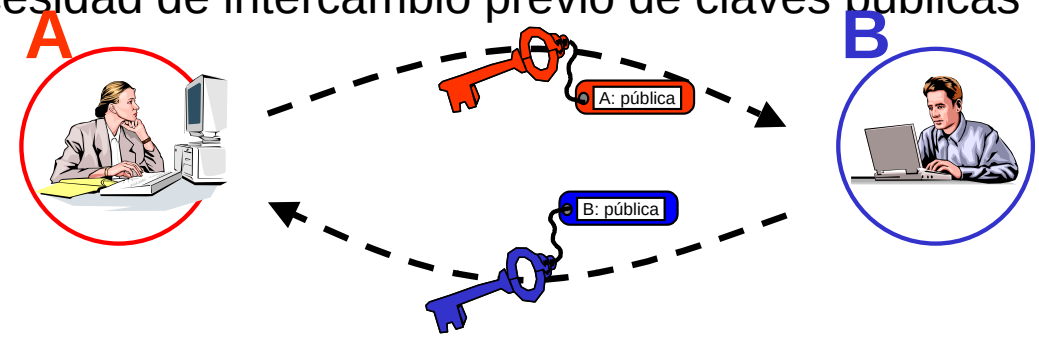
- **Funcionamiento**



Técnicas de criptografía o cifrado de los datos (8)

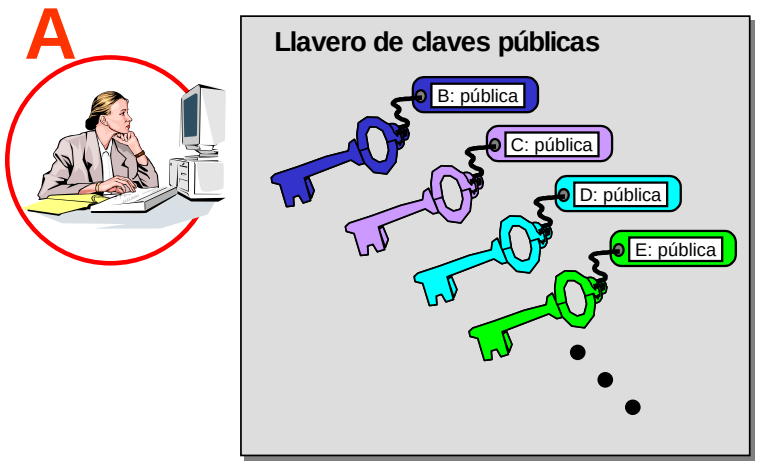
Criptografía de clave asimétrica o pública (iv)

- **Funcionamiento**
  - Necesidad de intercambio previo de claves públicas



  - Colección de claves públicas de otros usuarios:

    Llavero de claves públicas



Técnicas de criptografía o cifrado de los datos (9)

Criptografía de clave asimétrica o pública (v)

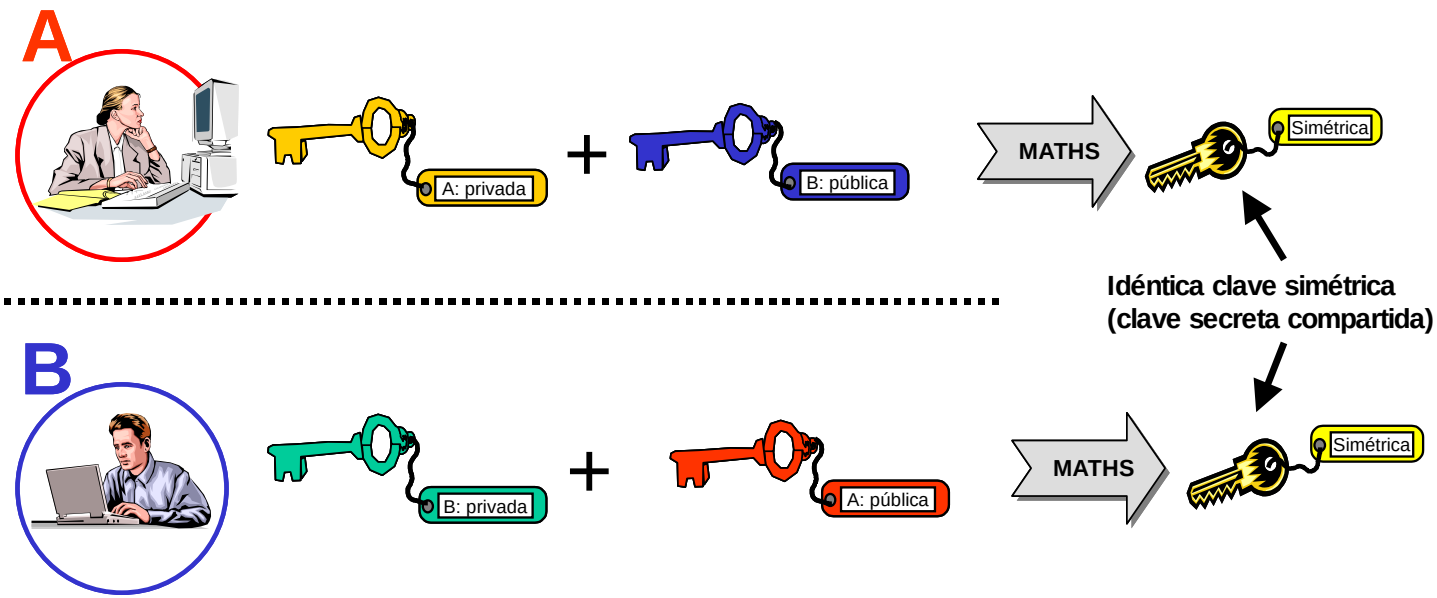
- **Ventajas**
  - Muy seguro
- **Desventaja**
  - Proceso de cifrado lento
  - Poco recomendable para mensajes muy largos
- **Solución:**
  - Combinar mecanismo de clave simétrica con mecanismo de clave asimétrica

- **Ejemplos de algoritmos de clave asimétrica o pública**
  - RSA (Rivest, Shamir y Adleman)

Técnicas de criptografía o cifrado de los datos (10)

Encriptación por clave secreta compartida (i)

- Basado en clave simétrica
  - La clave simétrica no se intercambia
  - La genera cada uno de los extremos de la siguiente manera



Técnicas de criptografía o cifrado de los datos (11)

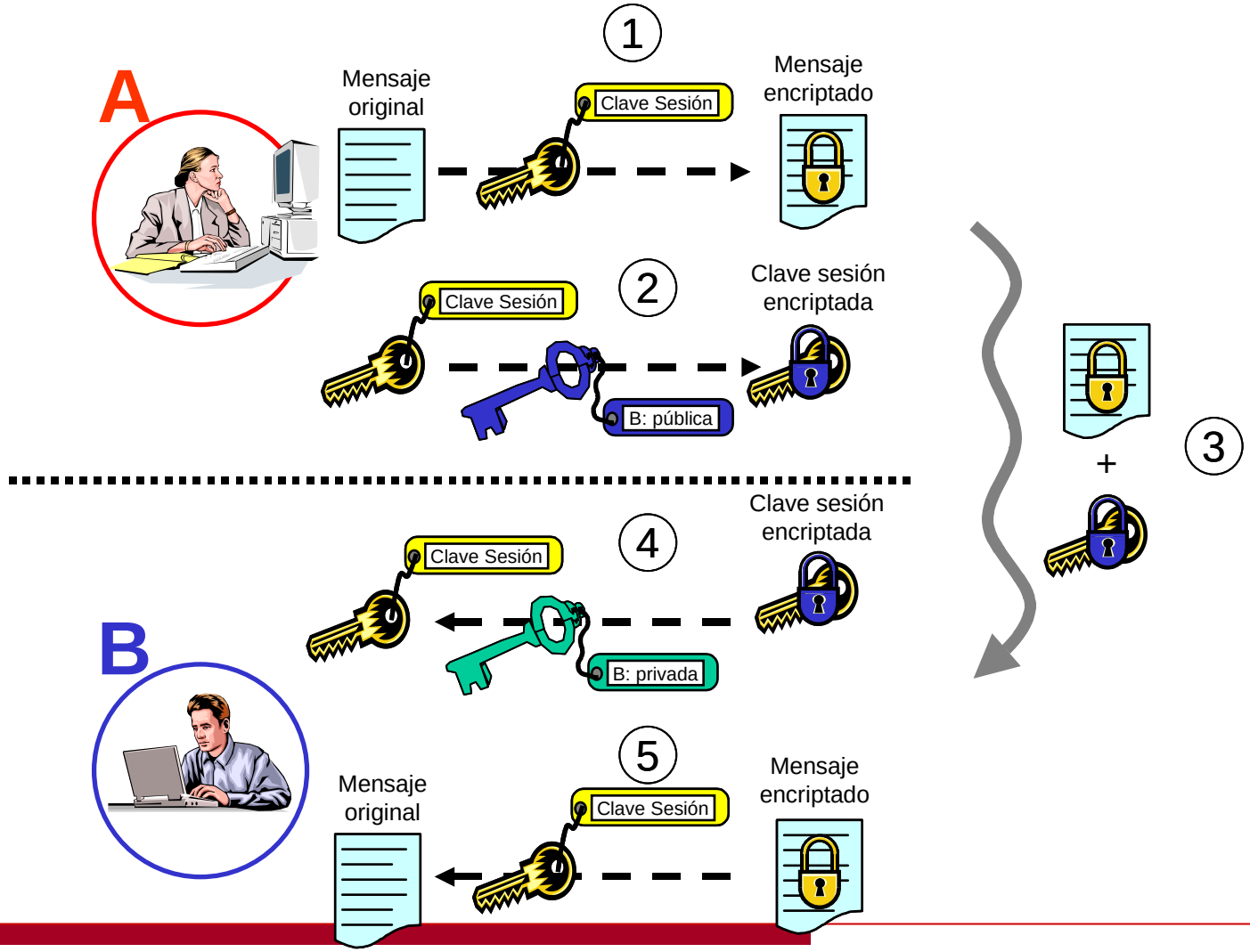
Encriptación por clave secreta compartida (ii)

- **Ventajas**
  - Rápido al ser simétrico
  - Seguro, al no tener que intercambiar la clave
- **Inconvenientes**
  - Capa pareja de usuarios utiliza siempre la misma clave
  - Cuanto más veces se utiliza una clave, más fácil es averiguar dicha clave

- **Ejemplos**
  - Diffie-Hellman

Técnicas de criptografía o cifrado de los datos (12)

Encriptación por clave de sesión (i)



Técnicas de criptografía o cifrado de los datos (13)

Encriptación por clave de sesión (ii)

- **Ventajas**
  - Rápido al ser simétrico
  - Seguro
    - La clave de sesión se envía encriptada
    - Para cada transmisión se utiliza una clave de sesión distinta

- **Ejemplos**
  - SSL (Secure Socket Layer)
    - Utilizado en servidores Web seguros (https)

PRACTICAS

- Comparar el funcionamiento de un protocolo de comunicaciones que utilice cifrado con otro sin cifrado
  - Protocolo con cifrado: **ssh** (cifrado por clave de sesión)
  - Protocolo plano: **telnet**
- Usar **ethereal** para las diferencias entre ambos

Autenticación e Integridad: firmas y certificados digitales

¿Para qué sirve una firma digital?

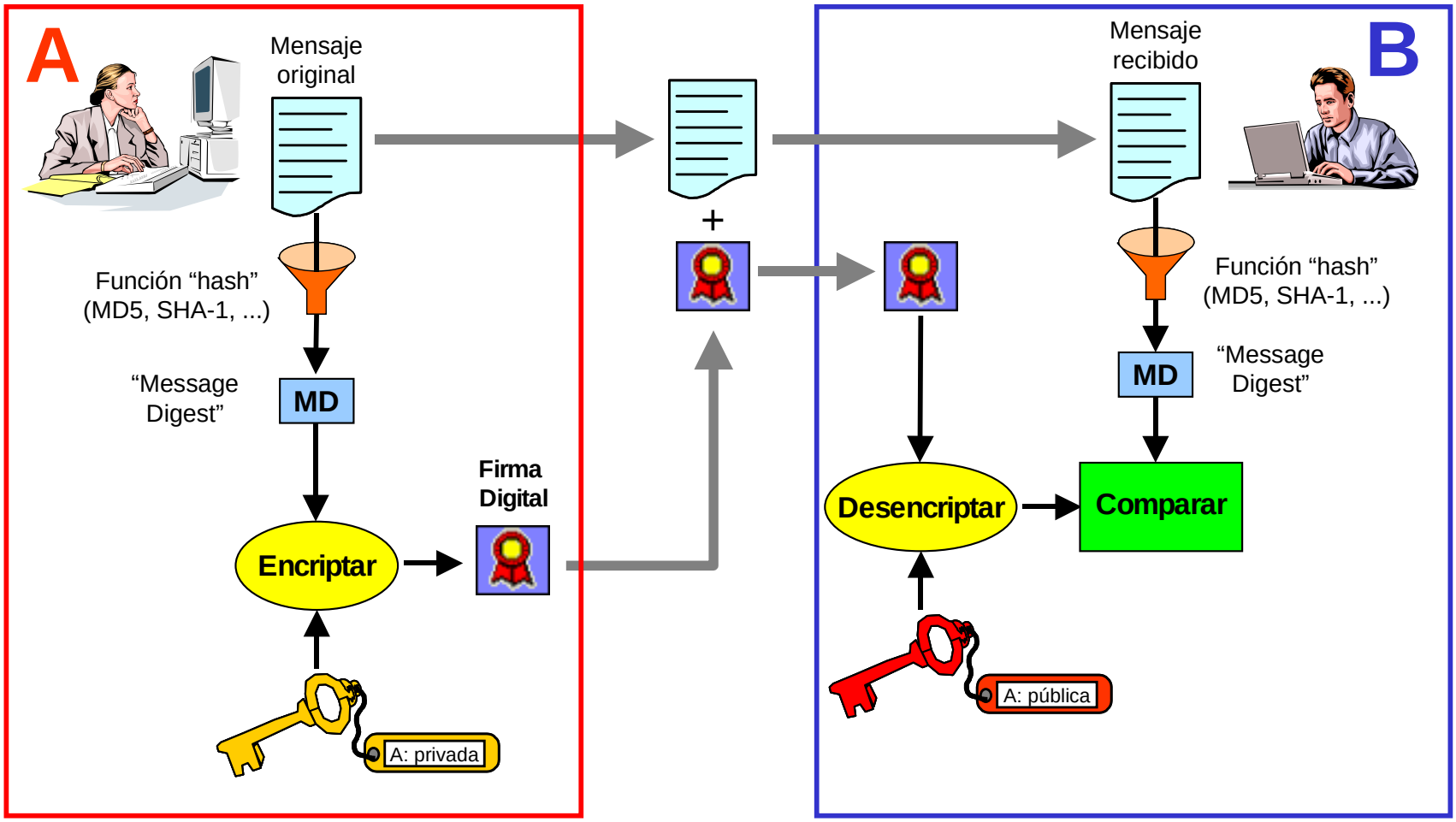
- Permite al receptor verificar la identidad del remitente
  ➔ Autenticación
- Permite al receptor verificar que la información no ha sido modificada
  ➔ Integridad

La firma se genera a partir de:

- La clave privada del remitente
  - El remitente debe disponer de una pareja de claves pública y privada
  - La clave privada se emplea para generar la firma digital
  - Esto garantiza la autenticación
- El mensaje original
  - A partir del mensaje original, se aplica una función de resumen (hash)
  - Este resumen se encripta con la clave privada, formando la firma digital del remitente
  - Esto garantiza la integridad

Autenticación e Integridad: firmas y certificados digitales

Generación de la firma digital



- **Autenticación e Integridad: firmas y certificados digitales**
  - Generación de la firma digital: la función HASH
    - Características
      - Transformar un texto de longitud variable en un bloque de longitud fija
      - Longitud pequeña (algunas son de 16 bits).
      - Irreversible
      - Propiedad de no colisión
      - Sencilla de implementar
    - Ejemplos
      - MD4 (Message Digest 4).
      - MD5 (Message Digest 5).
      - SHA-1 (Secure Hash Algorithm 1).

Autenticación e Integridad: firmas y certificados digitales

Combinación de firma digital y encriptación

	No encriptado No firmado	Encriptado No firmado	No encriptado Firmado	Encriptado Firmado
Confidencialidad	NO	SÍ	NO	SÍ
Integridad	NO	NO	SÍ	SÍ
Autenticación	NO	NO	SÍ	SÍ

Problemas de la firma digital

- ¿Cómo tener certeza de que la clave pública de un usuario corresponde realmente a ese individuo y no ha sido falsificada por otro?
- ¿Quién verifica la identidad del poseedor de la clave pública?

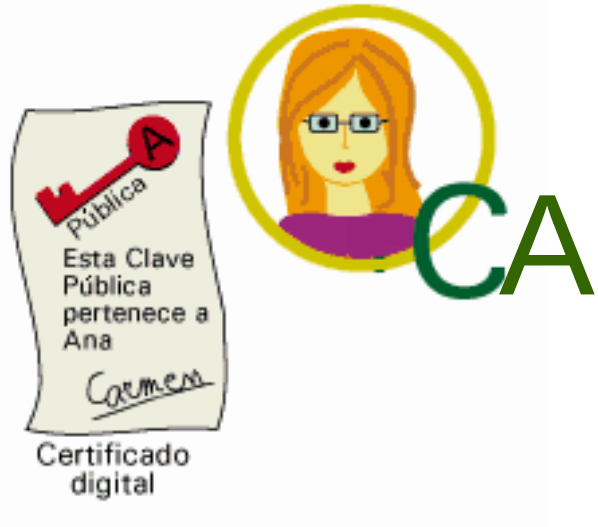
Solución

- Certificados digitales

Autenticación e Integridad: firmas y certificados digitales

Certificados digitales

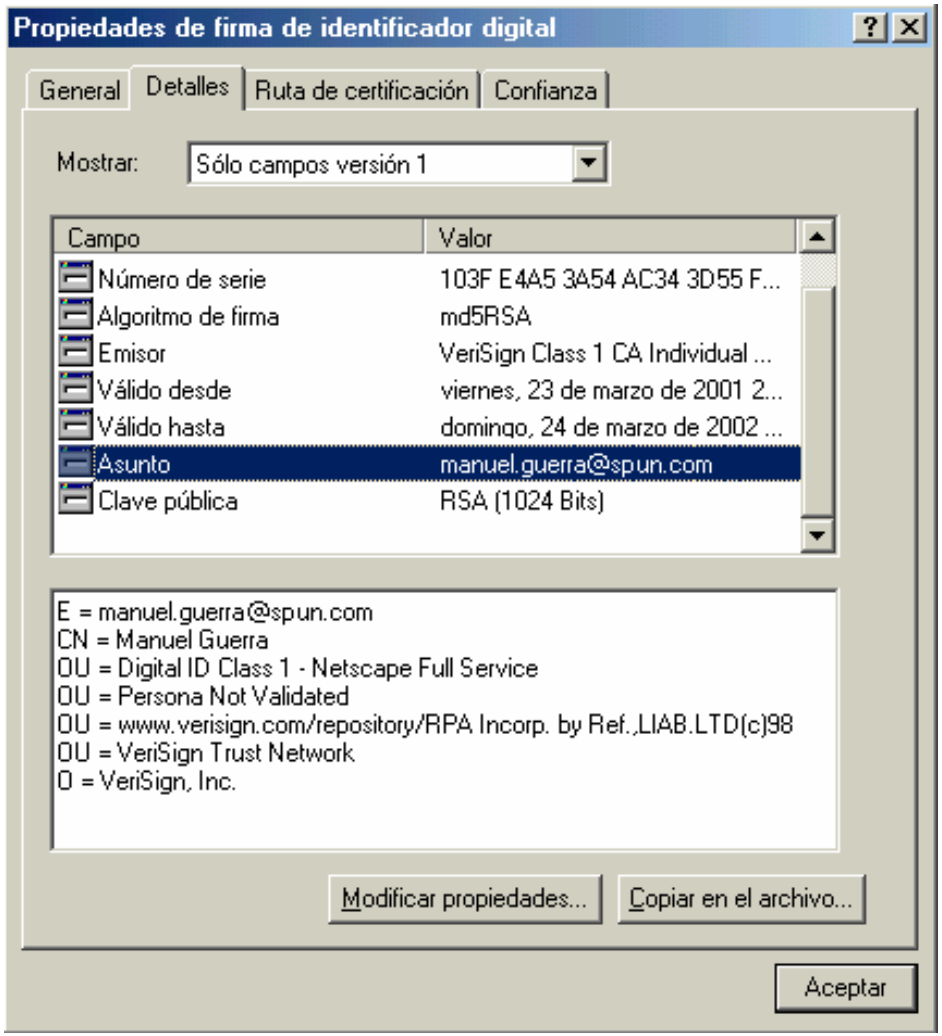
- ¿Qué es un certificado digital?
  - Fichero digital intransferible y no modificable,
  - Emitido por una tercera parte de confianza (**Autoridad de Certificación**)
  - Que asocia a una persona o identidad una clave pública.
- Autoridad de Certificación (CA)
  - Organización emisora de certificados digitales



- **Autenticación e Integridad: firmas y certificados digitales**
  - Información contenida en un certificado digital

Autenticación e Integridad: firmas y certificados digitales

Ejemplo de certificado digital



Autenticación e Integridad: firmas y certificados digitales

Tipos de certificados digitales

	Nivel de seguridad	Confirmación de la Identidad	Aplicaciones
Clase 1	Mínima	Únicamente comprueba la no duplicidad del nombre y e-mail en la base de datos	Correo electrónico, Navegación por Internet
Clase 2	Intermedia	Contrasta todos los datos identificativos facilitados por el usuario con las bases de datos relacionadas	Servicios de subscripción, Transacciones de mediano riesgo
Clase 3	Máxima	Requiere presencia personal ante terceras partes (notario, registro)	Comercio electrónico, Transacciones comerciales de alto riesgo

Seguridad en la Web: protocolo SSL

Utilidad del protocolo SSL en servidores Web

- Se utiliza para transacciones seguras a través de WEB (**servicio https**)
  - Compras por Internet
  - Banca Electrónica
  - Etc.
- Basado en el mecanismo de **encriptación por clave de sesión**
  - El servidor debe disponer de un certificado digital válido
    - Este certificado contiene la clave pública del servidor
  - El cliente no necesita disponer de un certificado digital
    - La clave simétrica se encripta con la clave pública del servidor

Seguridad en la Web: protocolo SSL

Funcionamiento del protocolo SSL

- El cliente se conecta a una página Web de tipo https
- El servidor envía su Certificado Digital al cliente, que incluye la clave pública del servidor
- El cliente comprueba que el certificado ha sido emitido por una CA de confianza y que dicho certificado es válido
- El cliente y el servidor acuerdan un algoritmo de encriptación soportado por ambas partes
- El cliente genera una clave simétrica de sesión
- El cliente encripta la clave de sesión con la clave pública del servidor y envía dicha clave encriptada
- El servidor recibe la clave de sesión y la desencripta con su clave privada
- El cliente y el servidor, a partir de este momento, se comunican de forma cifrada usando la clave de sesión compartida

Seguridad en la Web: protocolo SSL

Comprobación de la validez del certificado del servidor