

# **Redes de Computadores**

## *Nivel de Red:*

### *Introducción*

Área de Ingeniería Telemática  
Dpto. Automática y Computación  
<http://www.tlm.unavarra.es/>

# Bloque 3: Nivel de Red

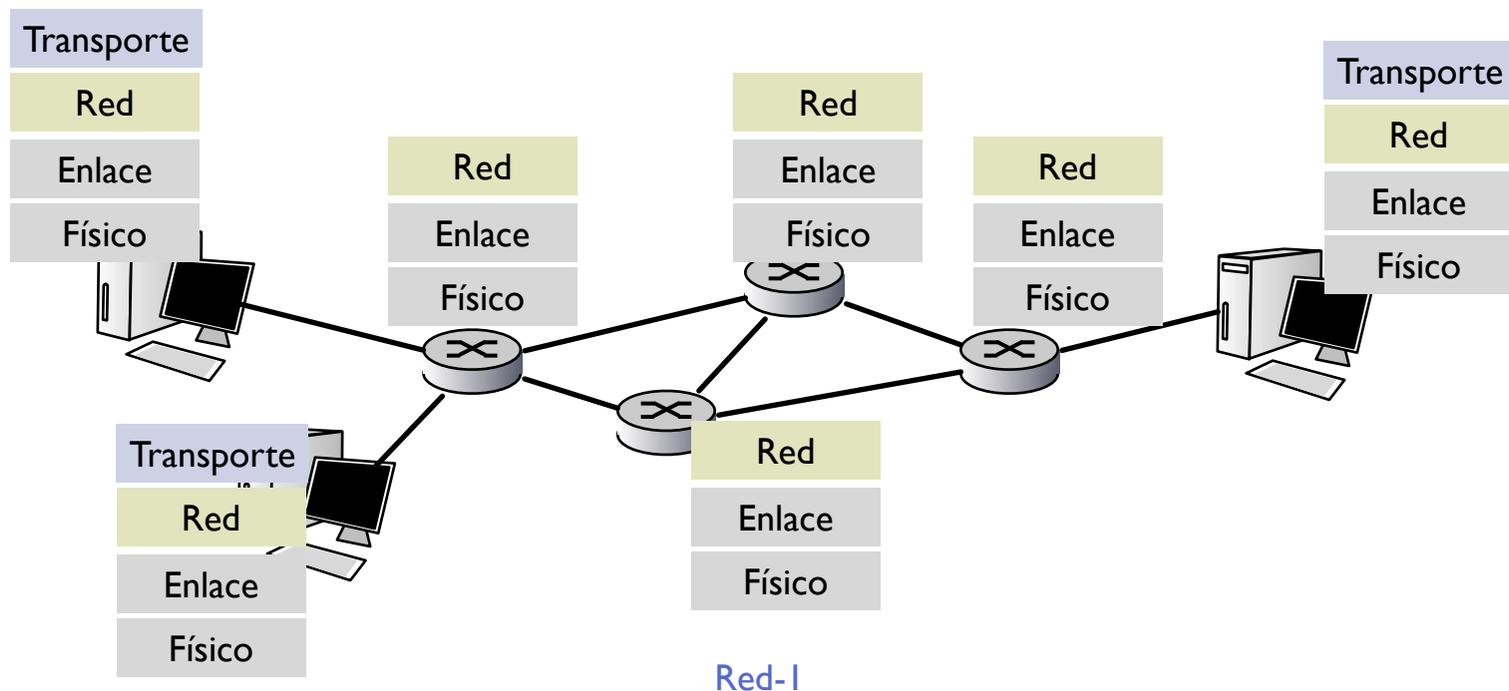
- ▶ El nivel de red: principios generales
- ▶ El nivel de red en Internet: IP
- ▶ IP e ICMP
- ▶ Enrutamiento
- ▶ Intranets y redes privadas

# En esta clase

- ▶ Funciones y principios básicos del nivel de red
- ▶ Reenvío y encaminamiento
- ▶ Servicios y arquitecturas del nivel de red
- ▶ Circuitos virtuales y datagramas

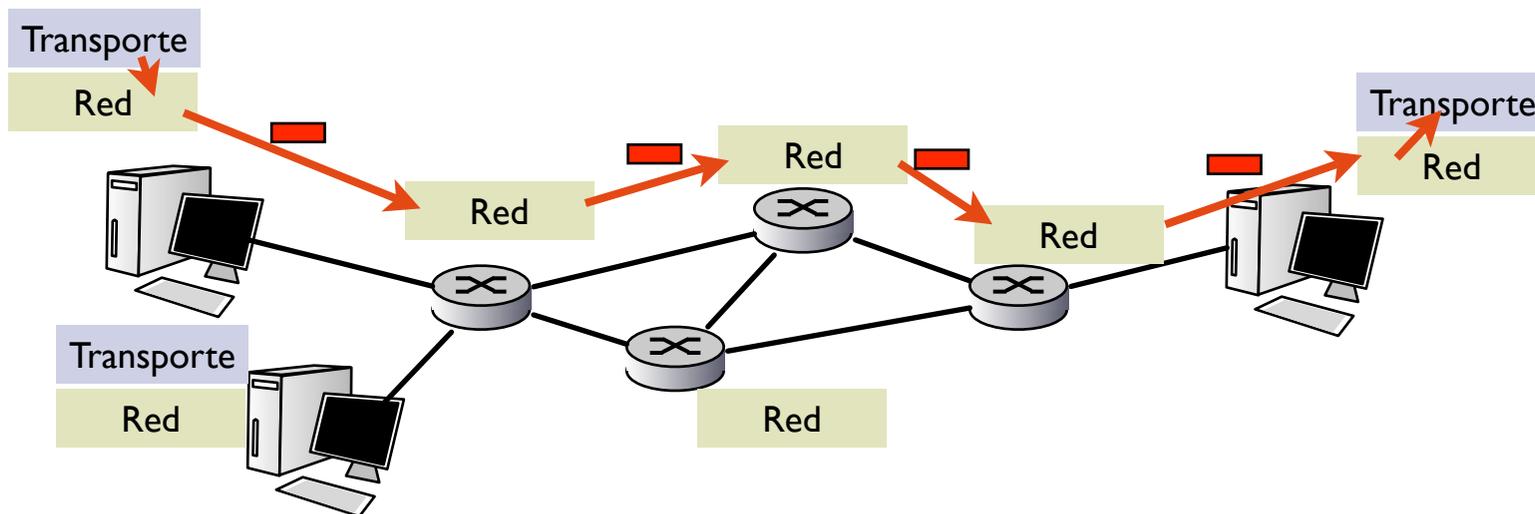
# Nivel de Red

- ▶ Transportar datos **entre hosts**
- ▶ Canal de comunicación entre niveles de transporte
  - > Emisor: encapsular segmentos de transporte en datagramas
  - > Receptor: recuperar segmentos y entregarlos al nivel de transporte
- ▶ Presente en todos los nodos (hosts, routers) de la red



# Nivel de Red

- ▶ El nivel de transporte entrega el paquete a su nivel de red, indicando la dirección del destino
- ▶ El nivel de red pasa ese paquete a algún nivel de red vecino
- ▶ Cada nivel de red puede hablar con otros niveles de red adyacentes
- ▶ 2 tareas para llegar al final
  - > Enviar paquetes a los niveles de red vecinos
  - > Saber a que vecino debo pasarselo para llegar al destino



# Funciones básicas del nivel de red

## ▶ Reenvío (Forwarding)

- > Enviar los paquetes recibidos de otros niveles de red al siguiente nodo del camino
  - + Llevar un paquete del interfaz de entrada al interfaz de salida

## ▶ Enrutamiento (Routing)

- > Conocer el camino para llevar los paquetes hacia su destino.
  - + Calcular los caminos en una malla de niveles de red
  - + Cuando deba reenviar un paquete saber a quien debo reenviarlo

# Forwarding y routing

## Routing:

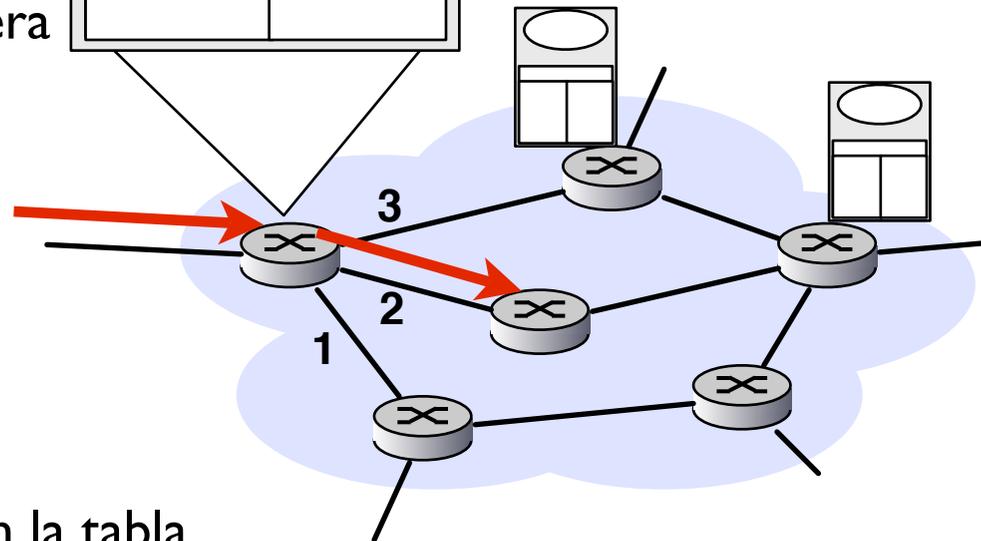
mantener la tabla con caminos correctos para cada dirección posible

La dirección de la cabecera se consulta en la tabla

0111

Algoritmo de enrutamiento

Tabla de rutas	
cabecera	camino
0100	3
0101	2
0111	2
1001	1



## Forwarding:

extraer dirección  
mirar dirección en la tabla  
enviar por el enlace indicado

# Funciones básicas del nivel de red

## ▶ Reenvío (Forwarding)

- > Enviar los paquetes recibidos de otros niveles de red al siguiente nodo del camino
  - + Llevar un paquete del interfaz de entrada al interfaz de salida

## ▶ Enrutamiento (Routing)

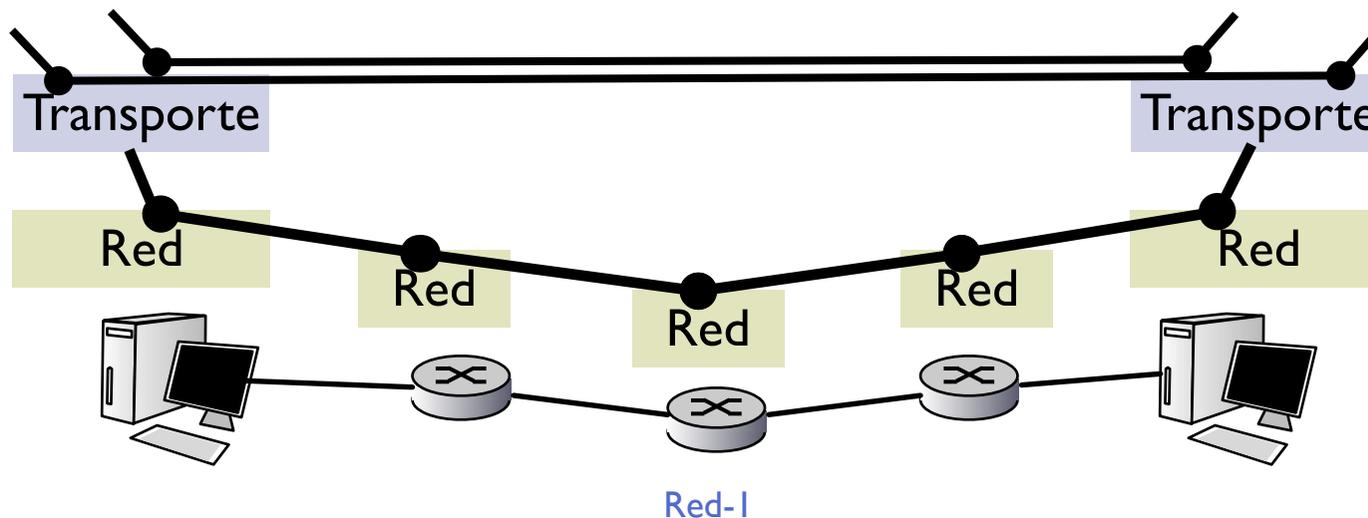
- > Conocer el camino para llevar los paquetes hacia su destino.
  - + Calcular los caminos en una malla de niveles de red
  - + Cuando deba reenviar un paquete saber a quien debo reenviarlo

## ▶ Establecimiento de conexión

- > Solo en algunas tecnologías de red (ATM, X25...)
  - + Preparar los recursos en los nodos
  - + Enrutamiento previo

# Establecimiento de conexión

- ▶ En algunas redes el nivel de red ofrece un servicio orientado a conexión
  - > X.25, Frame relay, ATM
- ▶ Antes de enviar datos, establecer la conexión
  - > Los routers a lo largo del camino conocen la conexión
- ▶ Había conexiones en el nivel de transporte
  - > Red: conexión entre hosts, conexión en la red
  - > Transporte: conexión entre procesos, existe sólo en los extremos



# Servicios de Red

- ▶ Servicios posibles en el nivel de red
  - > Garantías sobre un paquete
    - + Entrega garantizada
    - + Entrega con retardo máximo garantizado
  - > Garantías sobre un flujo de paquetes
    - + Entrega en orden
    - + Ancho de banda mínimo garantizado
    - + Jitter garantizado (espaciado entre paquetes se mantiene)
- ▶ ¿Cuales de estos ofrece Internet?

# Servicios en redes reales

Red	Servicio	Garantiza?				indica congestión
		BW	pérdidas	orden	retardo	
Internet	best effort	no	no	no	no	no
ATM	CBR	constante	si	si	si	no hay
ATM	VBR	garantizado	si	si	si	no hay
ATM	ABR	garantizado un mínimo	no	si	no	si
ATM	UBR	no	no	si	no	no

- ▶ La red que ha acabado imponiéndose es la que menos servicios ofrece !!
- ▶ Filosofía: red simple, haz solo una cosa pero bien

# Dos formas de organizar el nivel de red...

## ▶ Nivel de red **orientado a conexión**

- > Establecimiento de conexión enviando un paquete de señalización
- > Las decisiones de enrutamiento se toman una vez
- > Los paquetes llevan un **identificador de circuito virtual**
- > Este tipo de redes se llaman de **Circuitos Virtuales**

## ▶ Nivel de red **no orientado a conexión**

- > Se envían datos en cualquier momento a cualquier destino
- > Se debe decidir el camino independientemente para cada paquete
- > Cada paquete lleva la dirección completa del destino
- > Este tipo de redes se llaman de **Datagramas**

# Circuitos virtuales

- ▶ **Los caminos (circuitos) se comportan como circuitos telefónicos**
  - > En eficiencia (tiempo de establecimiento de conexión)
  - > En acciones y mensajes de establecimiento
    - + Establecimiento y liberación de circuitos
- ▶ **Los paquetes llevan un identificador de circuito virtual en lugar de la dirección de destino**
  - > Ventaja: es mas rápido reenviar en un circuito virtual
- ▶ **Los routers mantienen estado de los circuitos**
  - > Ventaja: se puede asignar recursos por conexión

# Circuitos virtuales: implementación

- ▶ Circuito virtual (VC) formado por:
  - > Camino de origen a destino
  - > Números de circuito virtual para cada enlace
  - > Entrada en las tablas de cada nodo
- ▶ El paquete lleva el número de VC
- ▶ El número de VC cambia en cada salto siguiendo las entradas en la tabla

# Circuitos virtuales: implementación

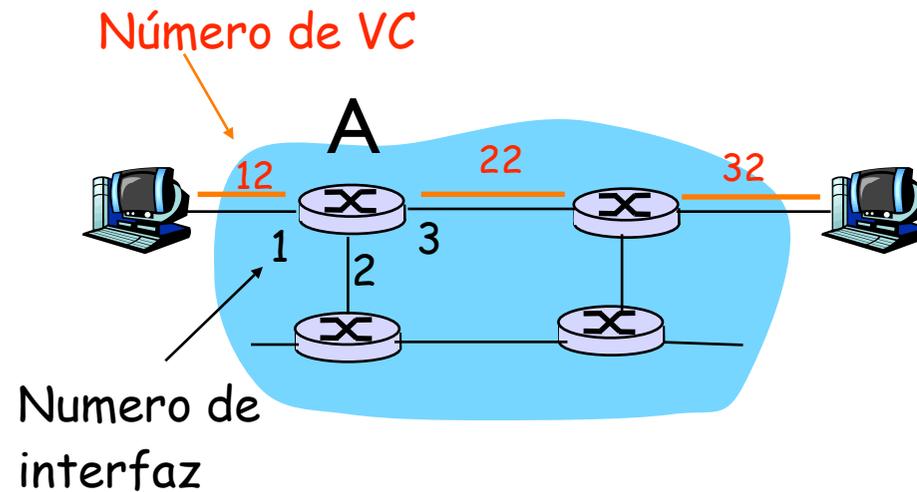


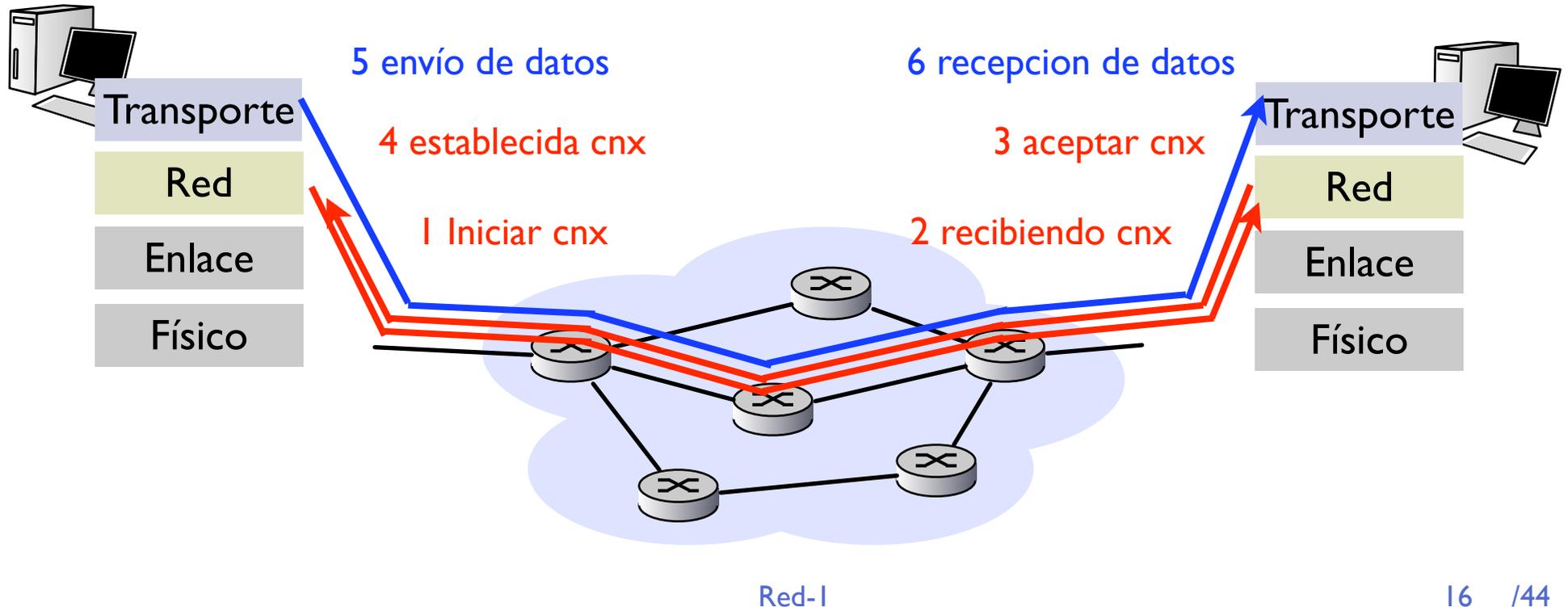
Tabla del router A

Interfaz IN	#VC IN	Interfaz OUT	#VC OUT
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

- ▶ Los routers mantienen el estado de las conexiones

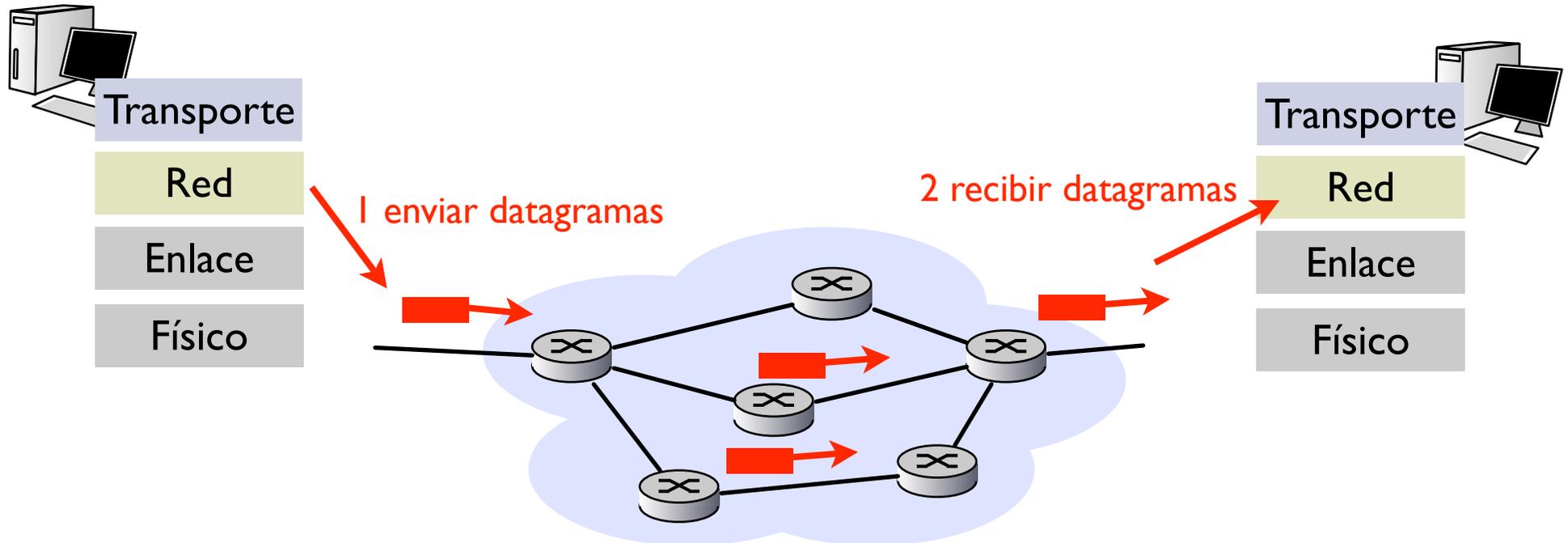
# Circuitos virtuales: señalización

- ▶ Establecer, mantener y cortar un CV
- ▶ En ATM, Frame Relay, X.25



# Red de Datagramas

- ▶ No hay establecimiento
- ▶ No hay estado en los routers
  - > Cada paquete lleva la dirección de destino
  - > Es procesado de forma independiente
  - > Pueden incluso seguir diferentes caminos (orden no garantizado)



# Datagramas y tabla de rutas

- ▶ Para cada paquete consultar en la tabla su dirección de destino
  - > Problema: hay mas destinos que circuitos virtuales...
- ▶ En Internet: **direcciones de 32 bits = 4294967296 destinos !**
- ▶ Si organizamos la red de forma que a las direcciones seguidas se vaya normalmente por el mismo enlace...

Destino	Enlace salida
11001000 00010111 00010000 00000000 ... 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 ... 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 ... 11001000 00010111 00011111 11111111	2
El resto	3

- ▶ Encaminamiento basado en el prefijo común

# Datagramas y tabla de rutas

## ► Búsqueda del prefijo más largo

- > Tabla de prefijos
- > Buscamos el prefijo más largo que coincida con la dirección de destino del paquete

Destino	Enlace salida
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
El resto	3

Ejemplos: qué interfaz corresponde a estos destinos?

11001000 00010111 00010110 10100001      0

11001000 00010111 00011000 10101010      ~~2~~ 1

# ¿ Datagramas o Circuitos Virtuales?

## ▶ Datagramas

- > Comunicaciones entre ordenadores
- > Sistemas “inteligentes” que se adaptan a las pérdidas
- > Red simple, complejidad en los extremos
  - + Facilidad para integrar diferentes tipos de enlaces

## ▶ Circuitos virtuales

- > Evolución de la telefonía, optimizado para conversaciones
  - + requisitos temporales, garantizar calidad de servicio
  - + terminales “tontos”
  - + la complejidad está en la red

## ▶ ¿De qué tipo es Internet?

# Internet

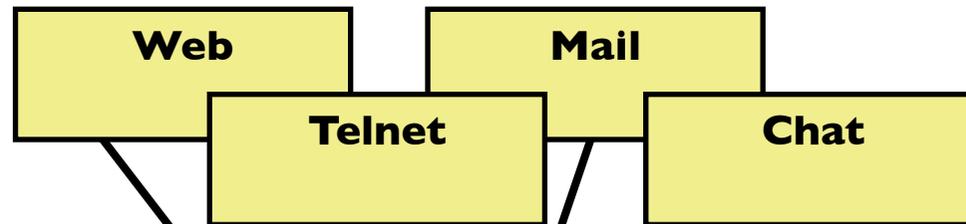
- ▶ Es una red de **Datagramas**
    - > Entrega no garantizada
    - > Tiempo de entrega no garantizado
    - > Orden no garantizado
    - > Servicio: best-effort
  - ▶ Internet = red formada con la pila de protocolos TCP/IP
- Nivel de red: **IP (Internet Protocol)**

# Resumen

- ▶ El nivel de red proporciona transporte entre niveles de transporte.
- ▶ Para ello los niveles de red vecinos se envían paquetes y salto a salto el paquete se va acercando a su destino
- ▶ Funciones forwarding y routing
- ▶ Filosofías de circuitos virtuales (orientados a conexión) y datagramas (sin conexiones)
- ▶ Internet es una red de datagramas

# TCP/IP: torre de protocolos de Internet

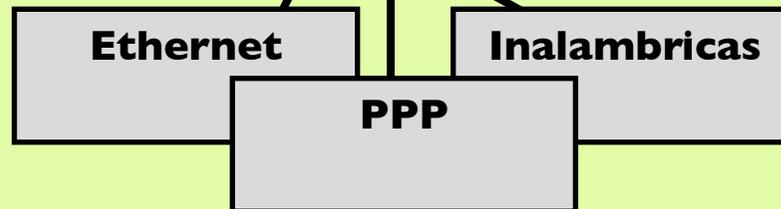
## Aplicaciones/servicios



Envío entre aplicaciones/servicios  
(direcciones IP + puertos)



Envío entre ordenadores  
(direcciones IP)

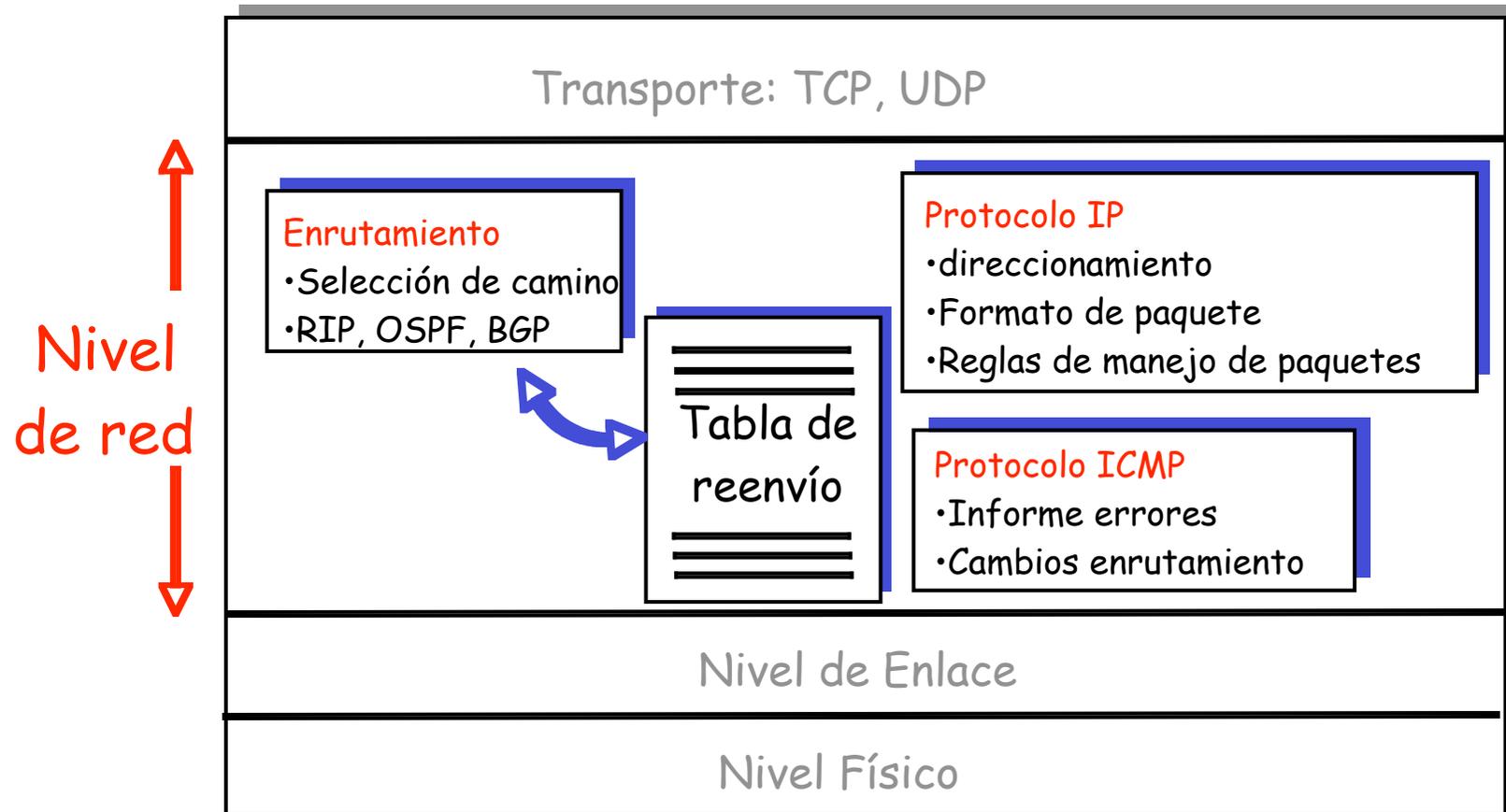


# Protocolo IP

- ▶ IP: Internet Protocol (RFC-791)
- ▶ ICMP: Internet Control Message Protocol (RFC-792)
- ▶ Varios componentes
  - > IP reenvío de paquetes a los vecinos
  - > ICMP reenvío de mensajes de error a los vecinos
  - > Tabla de reenvíos: tabla de rutas
  - > Separada la funcionalidad de encaminamiento: cálculo de la tabla de rutas. Originalmente sólo los routers tenían esta función

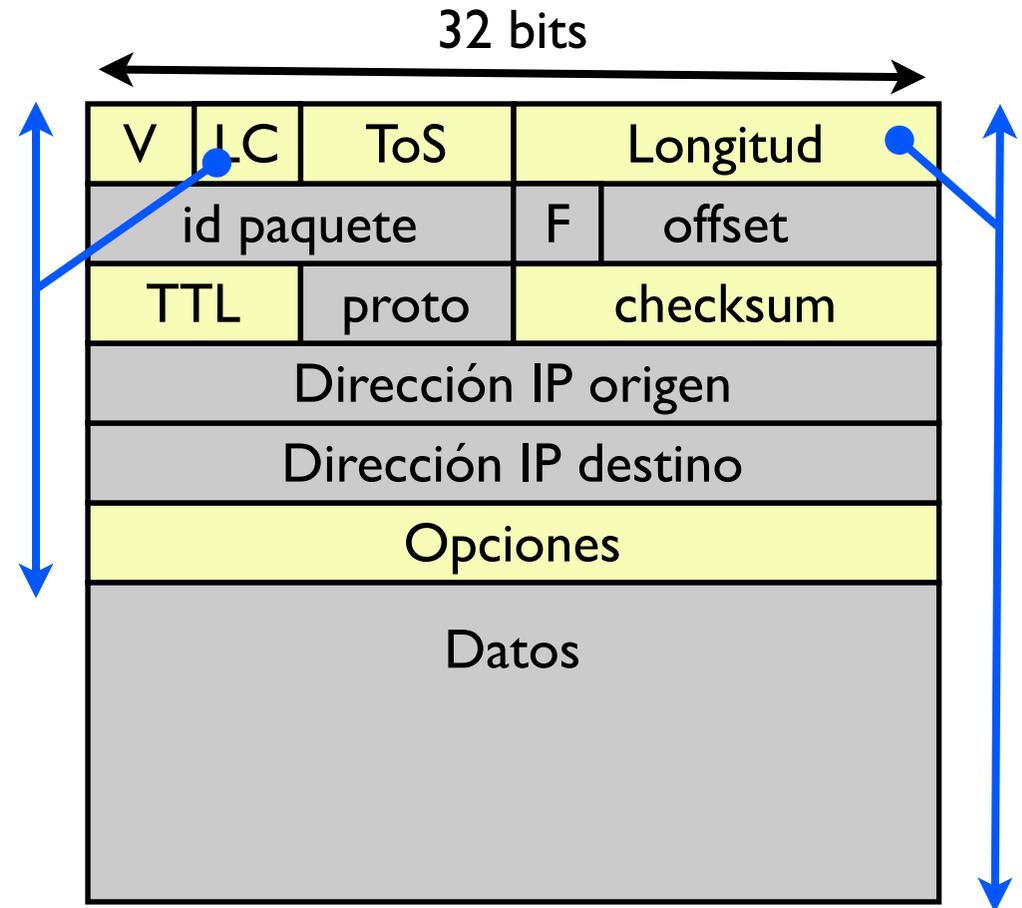
# Nivel de Red en Internet

- ▶ Nivel de red en un Host: componentes



# IP: Formato del paquete

- ▶ V: version de IP
- ▶ LC: longitud de la cabecera
- ▶ ToS: tipo de servicio
- ▶ Longitud total del paquete
- ▶ checksum de la cabecera
- ▶ time to live
- ▶ opciones



# IP: Formato del paquete: fragmentación

- ▶ proto: protocolo de nivel superior transportado

1 ICMP

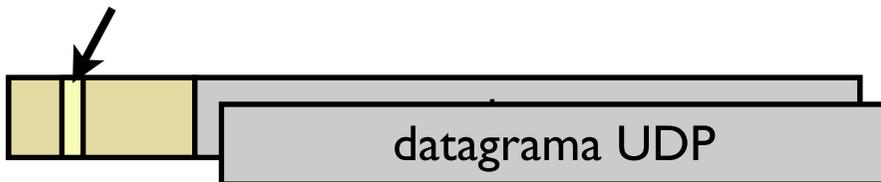
6 TCP

17 UDP

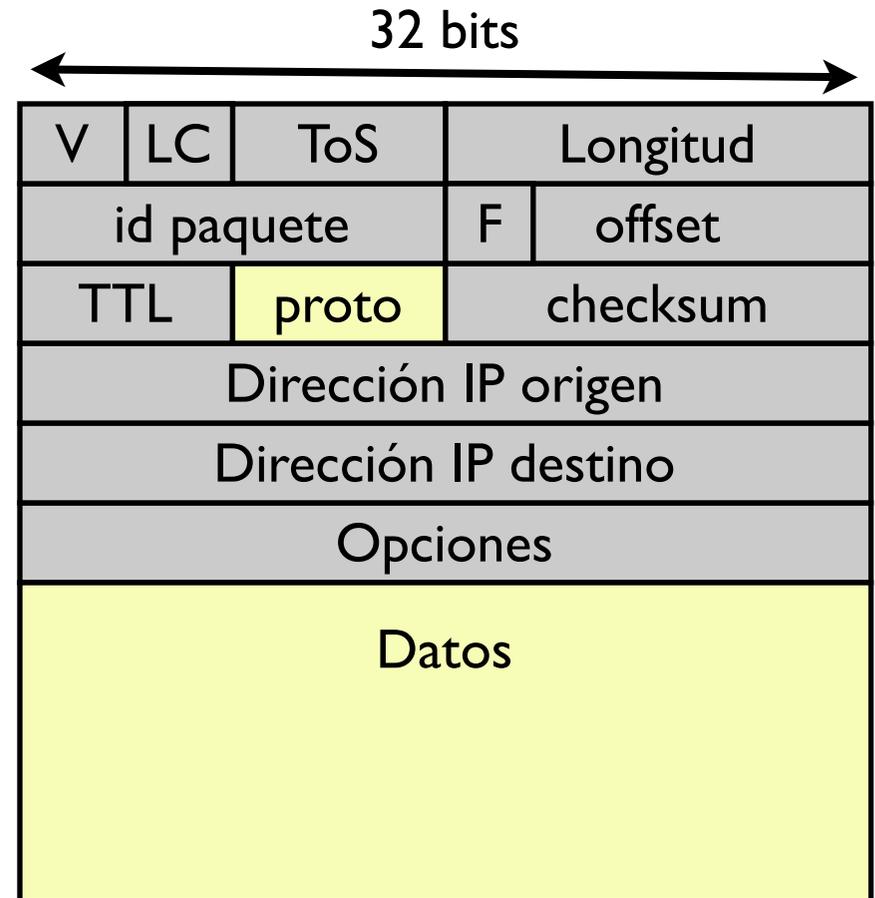
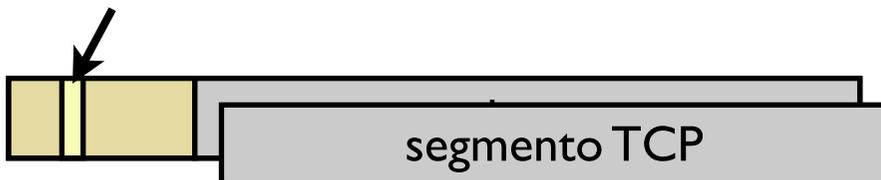
...

- ▶ Datos: datos de nivel superior

proto=17



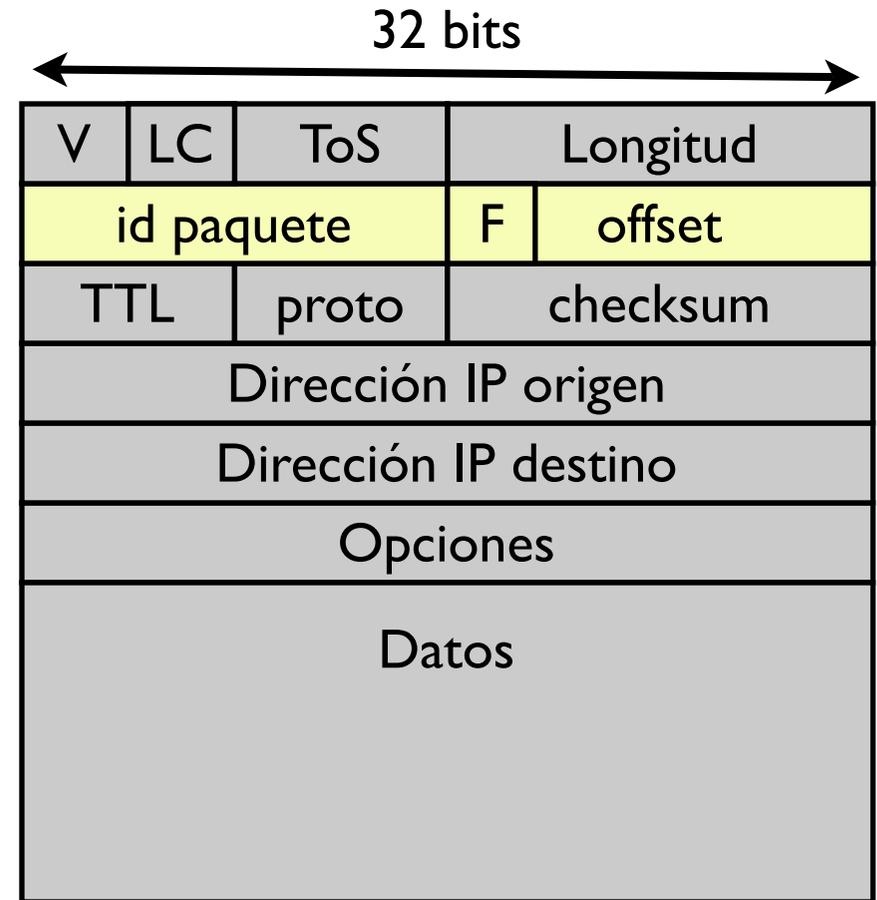
proto=6



# IP: Formato del paquete: fragmentación

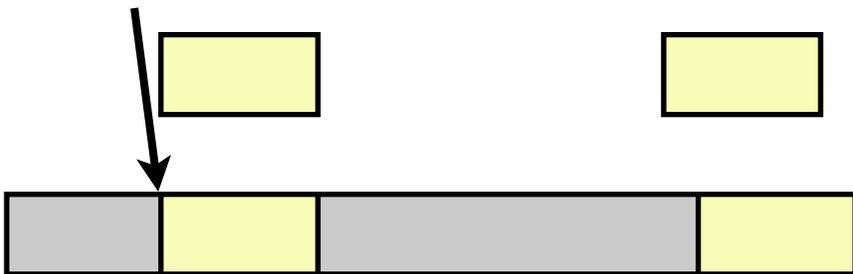
## ► Información de fragmentación

- > id (único) del paquete
- > F flags
  - + reservado 0
  - + no fragmentar
  - + mas fragmentos
- > offset (en palabras de 8 bytes)



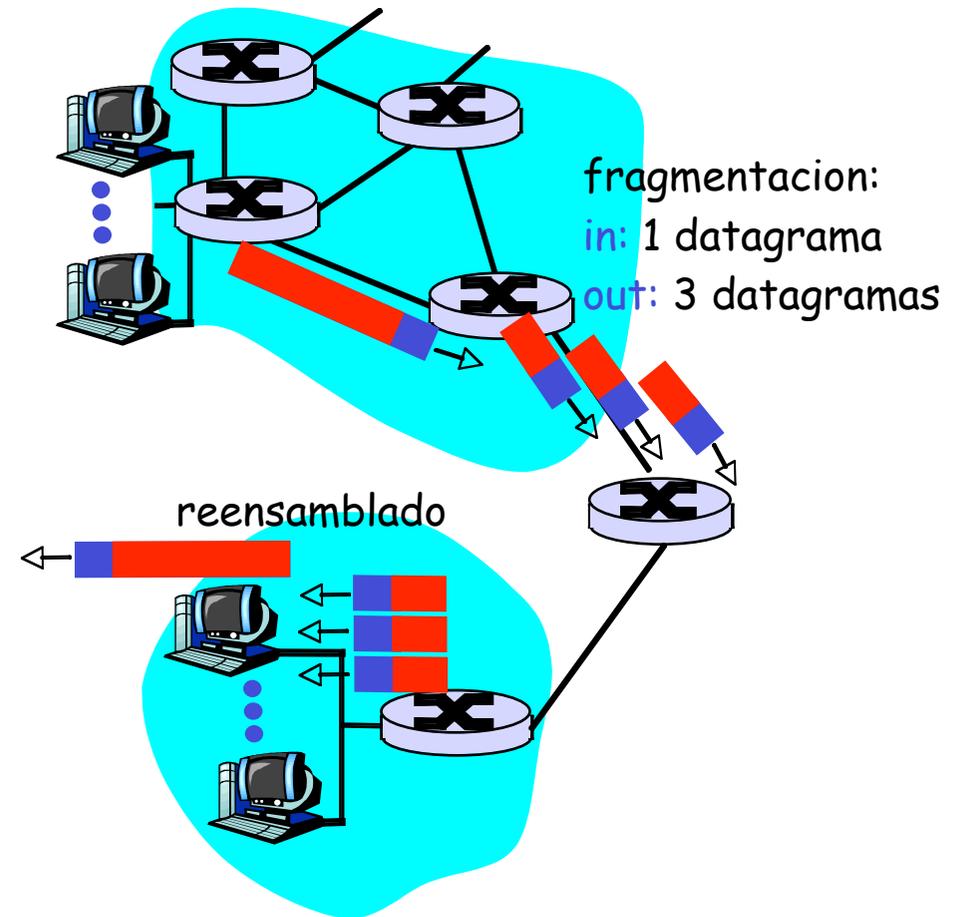
f: 1 (más)  
offset

f: 0 (último)



# Fragmentación y reensamblado IP

- ▶ Diferentes enlaces con diferentes MTU (max. transfer unit)
- ▶ Los routers fragmentan paquetes
  - > 1 paquete se divide en varios
  - > se reensamblan en destino

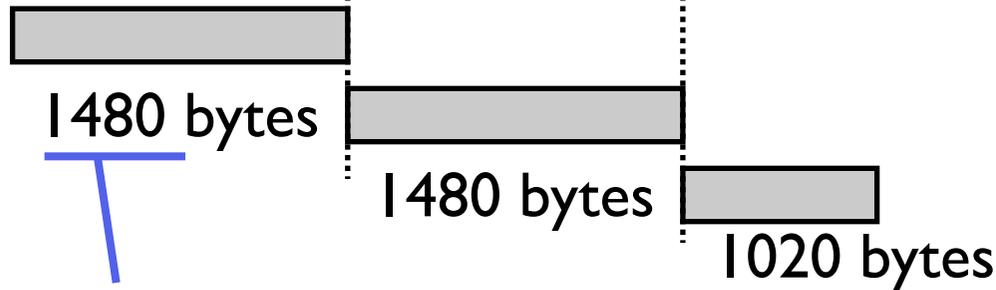


# Ejemplo

4000 bytes  
MTU=1500

Paquete original sin fragmentar

	<b>length</b> =4000	<b>id</b> =x	<b>fragflag</b> =0	<b>offset</b> =0	
--	------------------------	-----------------	-----------------------	---------------------	--



3 fragmentos

offset en palabras de 8 by  
1480/8

reservamos 20 bytes  
para cabecera IP

La longitud es del  
paquete IP

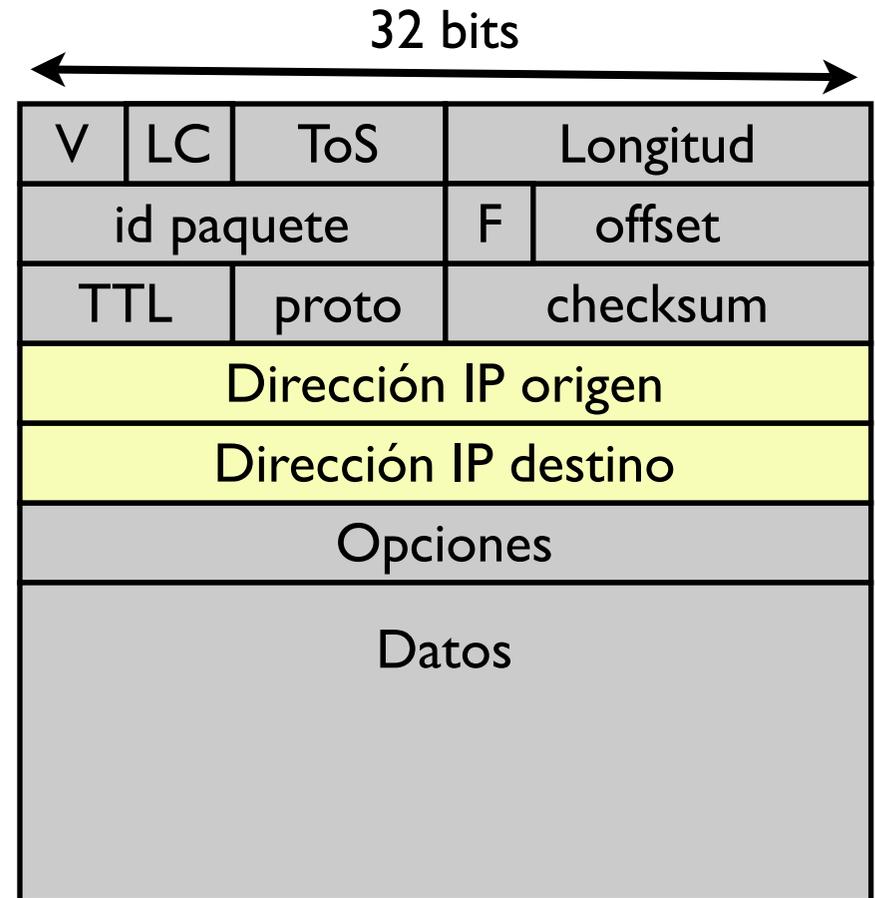
	<b>length</b> =1500	<b>id</b> =x	<b>fragflag</b> =1	<b>offset</b> =0	
--	------------------------	-----------------	-----------------------	---------------------	--

	<b>length</b> =1500	<b>id</b> =x	<b>fragflag</b> =1	<b>offset</b> =185	
--	------------------------	-----------------	-----------------------	-----------------------	--

	<b>length</b> =1040	<b>id</b> =x	<b>fragflag</b> =0	<b>offset</b> =370	
--	------------------------	-----------------	-----------------------	-----------------------	--

# Formato paquete IP: direcciones

- ▶ Dirección origen
- ▶ Dirección destino
  
- ▶ Dirección de protocolo IP
  - > Identifica a un Host de Internet



# IP v4 direccionamiento

- ▶ Dirección IP

- > **32 bits** 4294967296 direcciones?

- ▶ Ejemplo:

www.tlm.unavarra.es tiene por dirección:

10000010 11001110 10100000 11010111

Que también puede escribirse:

2194579671

0x 82 CE A0 D7

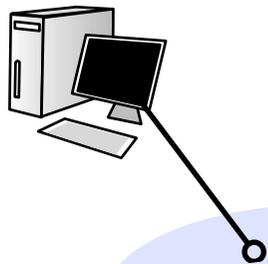
○ más comúnmente:

130 . 206 . 160 . 215

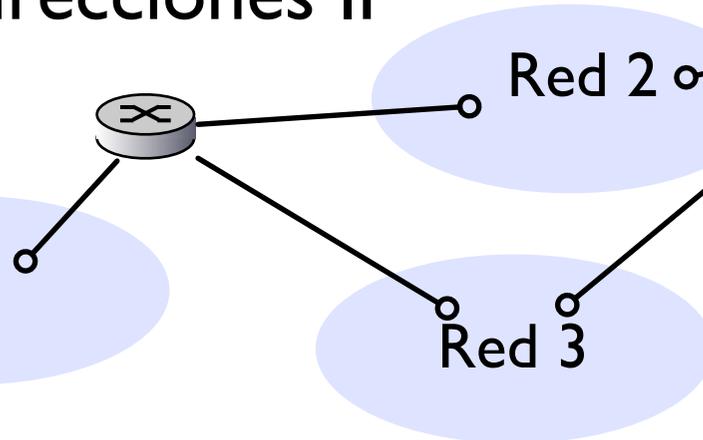
# Interfaces

- ▶ La dirección IP identifica a un **interfaz** de un **host** en Internet
  - > **Interfaz IP**: conexión a una red IP
  - > ¿ sólo nivel IP pero varios interfaces con varias direcciones?

Host típico  
1 dirección IP



Router  
3 direcciones IP

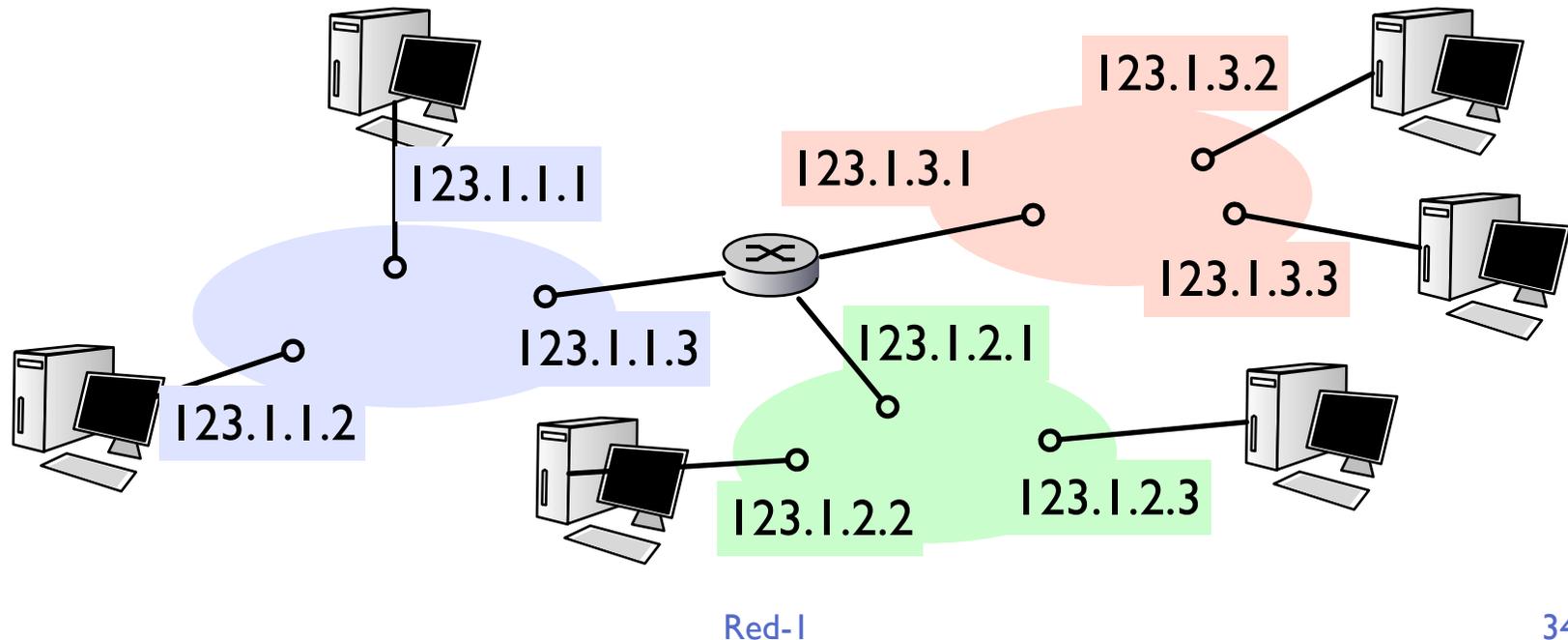


PC multihomed  
2 direcciones IP



# Subredes

- ▶ Reenvío por búsqueda de prefijo
  - Es conveniente agrupar las direcciones IPs con prefijos comunes en la misma red de área local
- ▶ Conjunto de direcciones con un mismo prefijo **subred / red IP**



# Subredes

- ▶ La dirección de un host se divide en 2 partes
  - > **Identificador de red (net id)**
  - > **Identificador de host (host id)**



Prefijo usado  
para enrutamiento

Identifica al host  
dentro de su red

- ▶ **¿Cómo organizamos los identificadores?**
  - > reparto de bits fijo?
  - > reparto variable?

# Direccionamiento con Clases (Classful)

- ▶ Reparto de direcciones original de Internet  
5 clases de direcciones

Clase A 

0	-NetID-	-----	-HostID-	-----
---	---------	-------	----------	-------

128 subredes con 16777216 hosts en cada una

Ejemplo:    netId:            103.0.0.0  
              dirección:    103.21.205.3

Clase B 

10	-----NetID-----	-----	-----HostID-----	-----
----	-----------------	-------	------------------	-------

16384 subredes con 65536 hosts en cada una

Ejemplo:    netId:            132.27.0.0  
              dirección:    132.27.5.123

# Direccionamiento con Clases (Classful)

▶ ...

Clase C



2097152 subredes con 256 hosts en cada una

Ejemplo: netId: 202.123.31.0  
dirección: 202.123.31.12

Clase D



**Multicast...** ver clase correspondiente

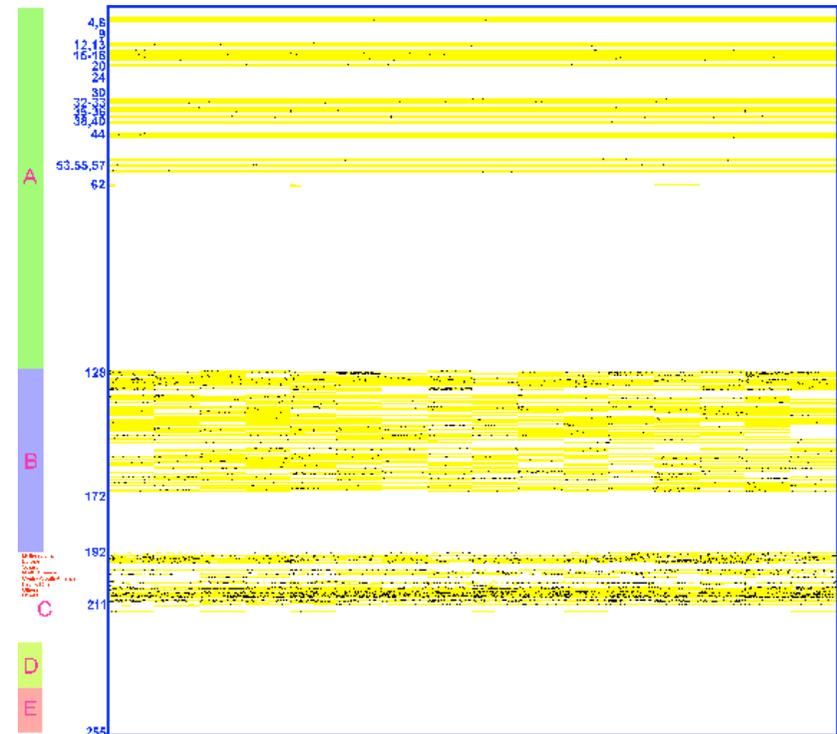
Clase E



Reservadas para usos futuros

# Direccionamiento con Clases (Classful)

- ▶ El reparto Classfull no pudo prever bien el crecimiento de Internet
  - > Redes de clase C muy pequeñas para empresas de 1000 ordenadores
  - > Sólo 16384 redes de tamaño mediano se agotan rápidamente
- ▶ Es necesario
  - > subdividir redes A ?
  - > unir redes C ? (supernetting)



# CIDR

- ▶ Actualmente se utiliza CIDR

## Classless InterDomain Routing (RFC 1519)

- > Utiliza longitud variable del prefijo (o identificador de red)
- > El prefijo se indica con un número de bits

a.b.c.d / x      x: número de bits del prefijo

Ejemplo:

direccion IP:      130.206.169.215/20

prefijo/netID:      130.206.160.0

# Tipos de direcciones IP

- ▶ Unicast (a un host)  
Las correspondientes a las clases A,B y C
- ▶ Multicast: Clase D  
(a varios hosts, los que han decidido recibirlo)
- ▶ Broadcast (a todos los hosts de una red)  
255.255.255.255 + broadcast dirigido
  
- ▶ + algunas excepciones y subredes reservadas

# Direccionamiento IP

## ▶ Casos especiales



## ▶ Todo 0s = this

- > 0.0.0.0 este host
- > netID=0 hostID en esta red

## ▶ Todo 1s = todos (broadcast)

- > 255.255.255.255 broadcast (en esta red)
- > 10.3.255.255 / 16 broadcast en la red 10.3.0.0/16

## ▶ Consecuencia

- > En la red 10.3.1.0/24 solo caben 254 hosts  
10.3.1.0 no se usa y se deja para nombre de la red  
10.3.1.255 es todos los hosts de esa red

# Direccionamiento IP

- ▶ Redes reservadas para uso privado
  - > 10/8
  - > 172.16/12
  - > 192.168/16
  - > 169.254/16
- ▶ 127/8    127.0.0.1 = loopback

# IP: Obteniendo direcciones

## ¿Cómo se obtiene un bloque de direcciones?

- ▶ Los ISP (Internet service providers) asignan direcciones IP o bloques a sus clientes
- ▶ Los registros de Internet locales(LIC) nacionales(NIC) o regionales(RIC) reparten los bloques entre los ISPs
- ▶ La autoridad mundial es el ICANN/IANA

## ¿Cómo obtiene un host la dirección IP?

- ▶ Configuración manual del administrador
- ▶ DHCP, BOOTP y similares
  - > protocolos de autoconfiguración

# Conclusiones

- ▶ El nivel de red de Internet proporciona un servicio de datagramas best-effort
- ▶ Cada paquete se maneja independientemente e incluye las direcciones origen y destino
- ▶ Para ello las direcciones deben repartirse siguiendo criterios de agrupar los prefijos comunes en la misma subred

Próxima clase:

- ▶ Como funciona IP: reenvío (forwarding)
- ▶ ICMP