

# Índice hora 4

## *Hora 1*

### 1 Características de IP

### 2 Cabecera IPv4

#### 2.1 Opciones de cabecera IPv4

## *Hora 2*

### 3 Direccionamiento IP

#### 3.1 Direccionamiento classful

#### 3.2 Subnetting

##### 3.2.1 Variable-Length Subnet Mask (FLSM)

##### 3.2.2 Variable-Length Subnet Mask (VLSM)

## *Hora 3*

#### 3.3 Supernetting

#### 3.4 Direccionamiento classless

##### 3.4.1 Classless Interdomain Routing (CIDR)

#### 3.5 Direcciones IP especiales

#### 3.6 Direccionando una red

## *Hora 4*

### 4 Router

### 5 Reenvío

#### 5.1 Reenvío en redes classful

#### 5.2 Reenvío en redes con subnetting VLSM y FLSM

#### 5.3 Reenvío en redes classless – CIDR

### 6 Fragmentación y reensamblado

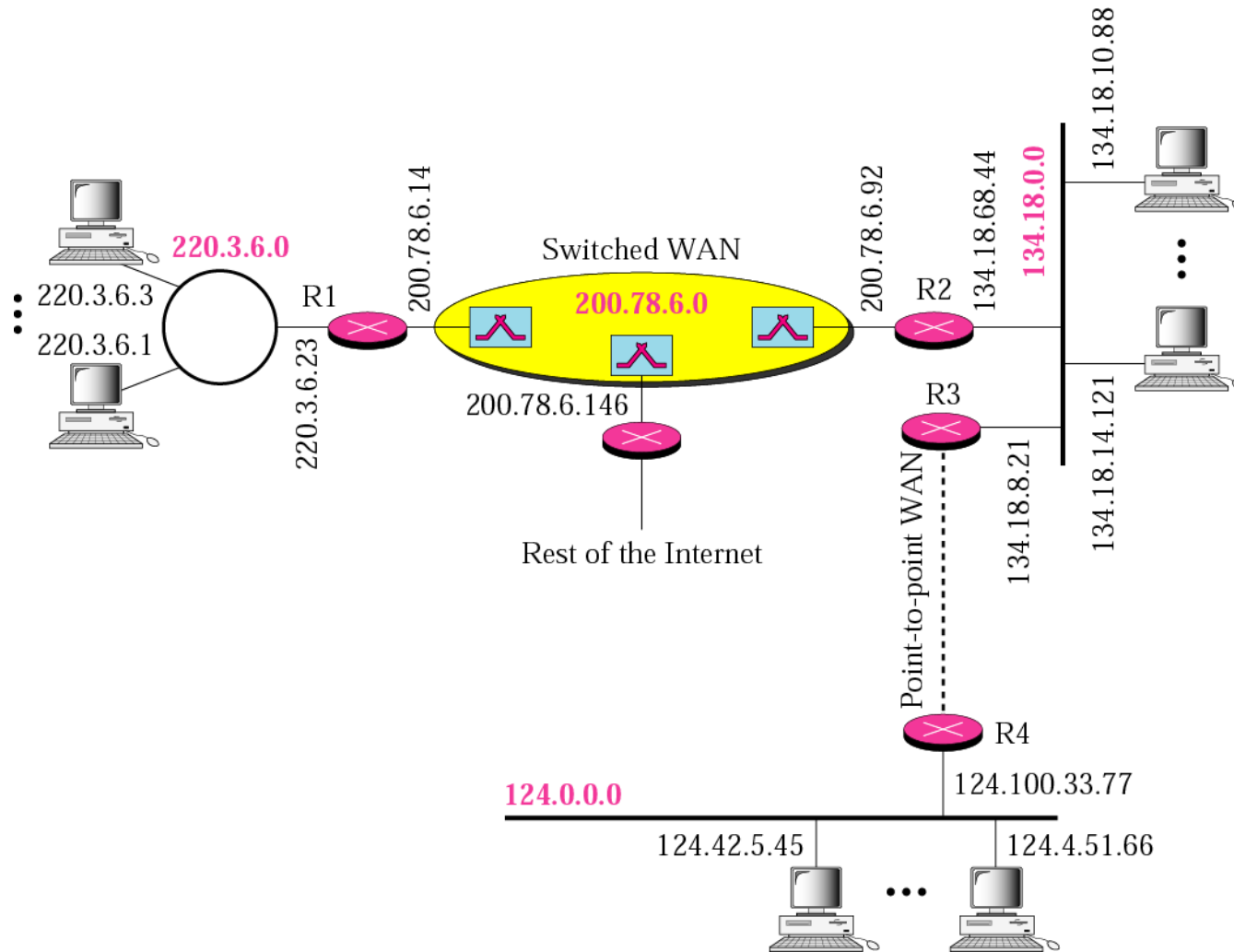
## Objetivos

- Conocer el router como dispositivo de interconexión de nivel 3.
- Caracterizar el proceso de reenvío de paquetes IP en un router.
- Conocer qué es una tabla de rutas y cómo funciona.
- Conocer el proceso de fragmentación y reensamblado IP.

## 4 Router

- Router: dispositivo de interconexión a nivel 3 de la torre OSI.  
Trabaja:
  - A nivel físico regenerando la señal que recibe.
  - A nivel de enlace comprobando las direcciones físicas (origen y destino) de los paquetes.
  - A nivel de red, comprobando las direcciones de nivel de red (direcciones IP)
- Un router permite interconectar LANs y WANs
  - Permite interconectar redes con distinta tecnología (distinto nivel de enlace).
  - Un router posee uno o más interfaces conectados a diferentes redes IP
  - Aisla a nivel de enlace los segmentos de red que une.
    - Diferentes dominios de broadcast: no reenvía tráfico de broadcast.
    - Diferentes dominios de colisión: no reenvía colisiones

# Router

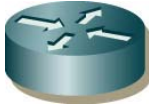


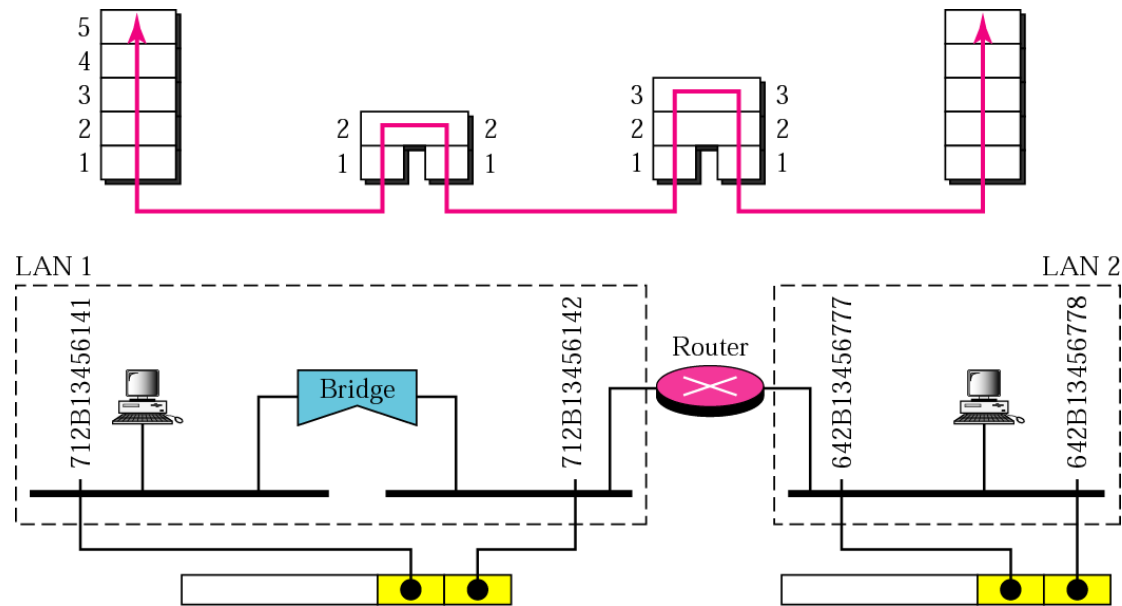
# Router

- A diferencia de un hub o switch, un router:
  - Tiene una dirección física y lógica (IP) por cada uno de sus interfaces. Puede llegar a tener varias direcciones IPs en el mismo interfaz.
  - Actúa sólo sobre aquellos paquetes cuya dirección física destino coincida con la del interfaz del router por el que llega.
- Reenvío, encaminamiento, enrutamiento, forwarding, routing...
  - Un router provee funcionalidades de reenvío de paquetes IP
    - Según la dirección IP destino del paquete, reenvía el paquete por el puerto adecuado a ese destino.
  - El proceso de reenvío:
    - Cambia la dirección física origen y destino de los paquetes cuando los reenvía.
    - Mantiene las direcciones IP extremo a extremo.
  - Tabla de rutas: configuración interna que determina el mejor puerto de salida para determinado destino.

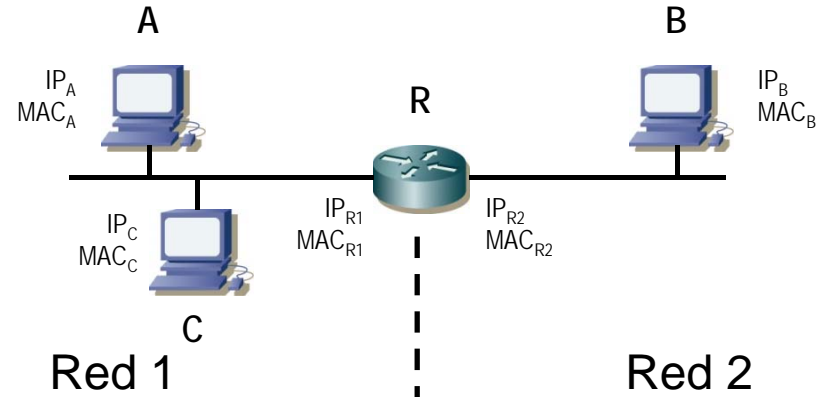
Máquina destino	Puerto de salida

# Router

- Representación habitual: 
- Conmutador de nivel 3 = router de alta velocidad.
- Ejemplo router, cambia las direcciones físicas de los paquetes.



# Router



- Paquete de A a C:
- Paquete de C a A:
- Paquete de A a R:
- Paquete de A a B:
- Paquete de B a A:

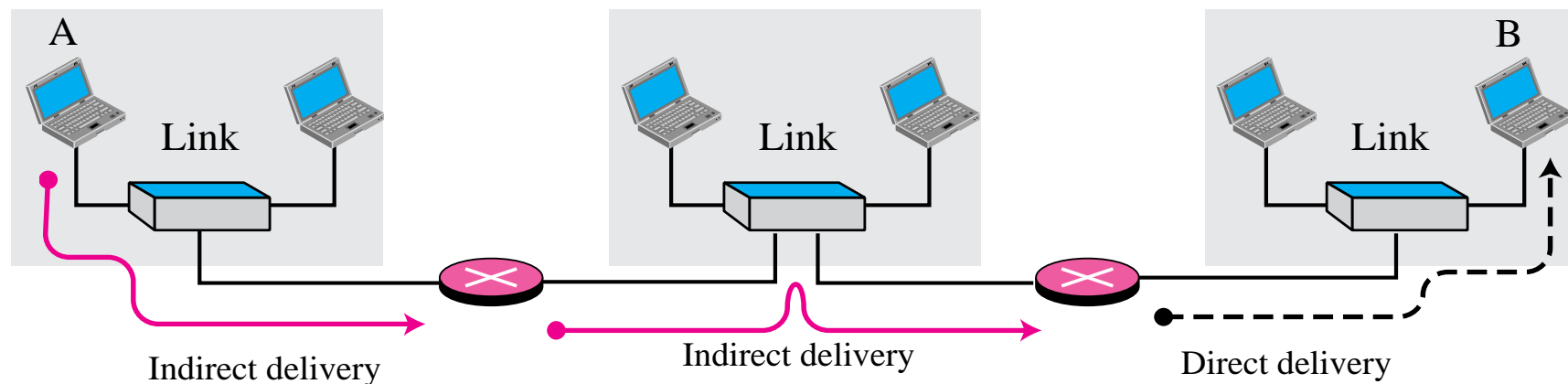
MAC <sub>origen</sub>	MAC <sub>destin</sub>	IP <sub>origen</sub>	IP <sub>destino</sub>
MAC <sub>A</sub>	MAC <sub>C</sub>	IP <sub>A</sub>	IP <sub>C</sub>
MAC <sub>C</sub>	MAC <sub>A</sub>	IP <sub>C</sub>	IP <sub>A</sub>
MAC <sub>A</sub>	MAC <sub>R1</sub>	IP <sub>A</sub>	IP <sub>R1</sub>
MAC <sub>A</sub>	MAC <sub>R1</sub>	IP <sub>A</sub>	IP <sub>B</sub>
MAC <sub>R1</sub>	MAC <sub>A</sub>	IP <sub>B</sub>	IP <sub>A</sub>

MAC <sub>origen</sub>	MAC <sub>destin</sub>	IP <sub>origen</sub>	IP <sub>destino</sub>
MAC <sub>R2</sub>	MAC <sub>B</sub>	IP <sub>A</sub>	IP <sub>B</sub>
MAC <sub>B</sub>	MAC <sub>R2</sub>	IP <sub>B</sub>	IP <sub>A</sub>

## 5 Reenvío

- Una máquina puede comunicarse directamente con otra máquina de su misma red por nivel de enlace.
- Sin embargo, si la máquina destino se encuentra en otra red, se hace necesario enviar el paquete al router por defecto de la red para que él sea el encargado de reenviarlo.
- El reenvío consiste en poner al paquete en su ruta al destino. Para ello el router necesitará tener configurada su tabla de rutas.





## Tabla de rutas manual

- En escenarios de red pequeños (p.e. una pequeña empresa) es posible definir una tabla de rutas manual para cada router que especifique el camino a seguir para cada posible red destino.
  - Las redes directamente conectadas las conoce un router por tener configurado el direccionamiento IP en sus interfaces.
  - Las redes separadas más de un salto se tendrán que configurar manualmente colocando como router siguiente salto la dirección IP de aquel router vecino en el camino al destino.
  - Podrá existir una ruta por defecto (0.0.0.0/0) que es la ruta que seguirán los paquetes que no cumplan ninguna otra entrada de la tabla de rutas.
  - En el caso de máquinas finales, habitualmente no existe una tabla de rutas como tal sino que el tráfico dirigido fuera de su red automáticamente se reenvía al router por defecto que tenga configurado.

# Tabla de rutas manual

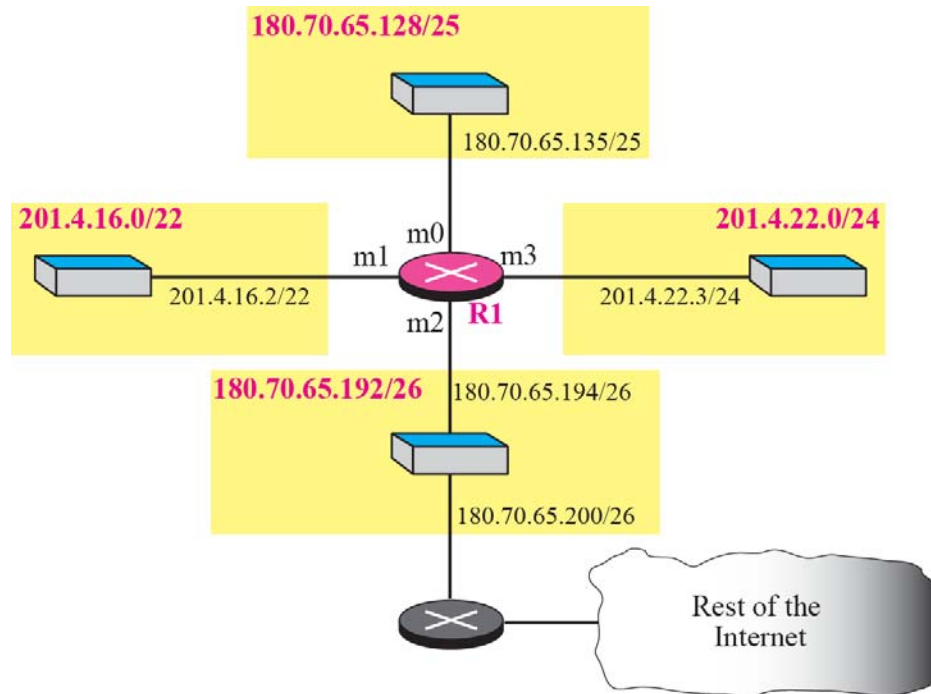


Tabla de rutas de R1

Mask	Network Address	Next Hop	Interface
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0	....	m1
Default	Default	180.70.65.200	m2

## Tabla de rutas dinámica

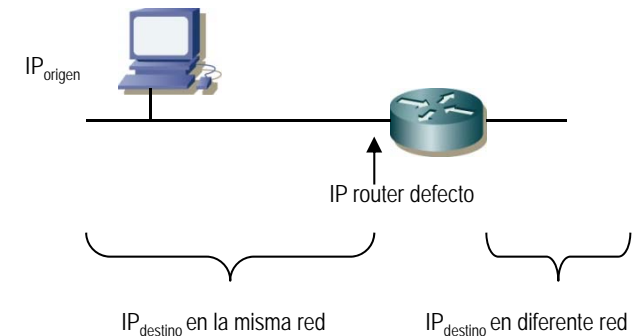
- Para redes más grandes o incluso Internet, no es posible utilizar tablas de rutas manuales.
  - No escala su complejidad.
  - No se adaptan a cambios en la red (caída de un enlace, aparición de nuevos caminos, etc.)
- En Internet se utilizan protocolos de enrutamiento dinámico como RIP, OSPF, IS-IS y BGP para que los propios routers descubran la topología y los caminos de ruta óptima (menor coste) a cada destino [Nota: estos protocolos de enrutamiento dinámico se revisarán en una asignatura de la especialidad].

## 5.1 Reenvío en redes classful

- En máquinas (un interfaz de red)

- Configuración necesaria:

- Dirección  $IP_{origen}$ .
    - Dirección IP del router por defecto de su red.
    - No imprescindible: servidores DNS.



- Proceso de reenvío de un paquete a  $IP_{destino}$ :

- La máquina calcula su  $NetworkID_{origen}$  a partir de la dirección  $IP_{origen}$ , según la clase a la que pertenece.
    - Para cada destino, calcula el  $NetworkID_{destino}$  a partir de la dirección  $IP_{destino}$ , según la clase a la que pertenece.
    - Si  $NetworkID_{destino} = NetworkID_{origen}$ , ambas máquinas se encuentran en la misma red y se envía el paquete por nivel de enlace al destino directamente ( $MAC_{destino}$ ).
    - Si no, se envía el paquete por nivel de enlace al router por defecto ( $MAC_{router}$ ).

# Ejemplo de reenvío en máquinas

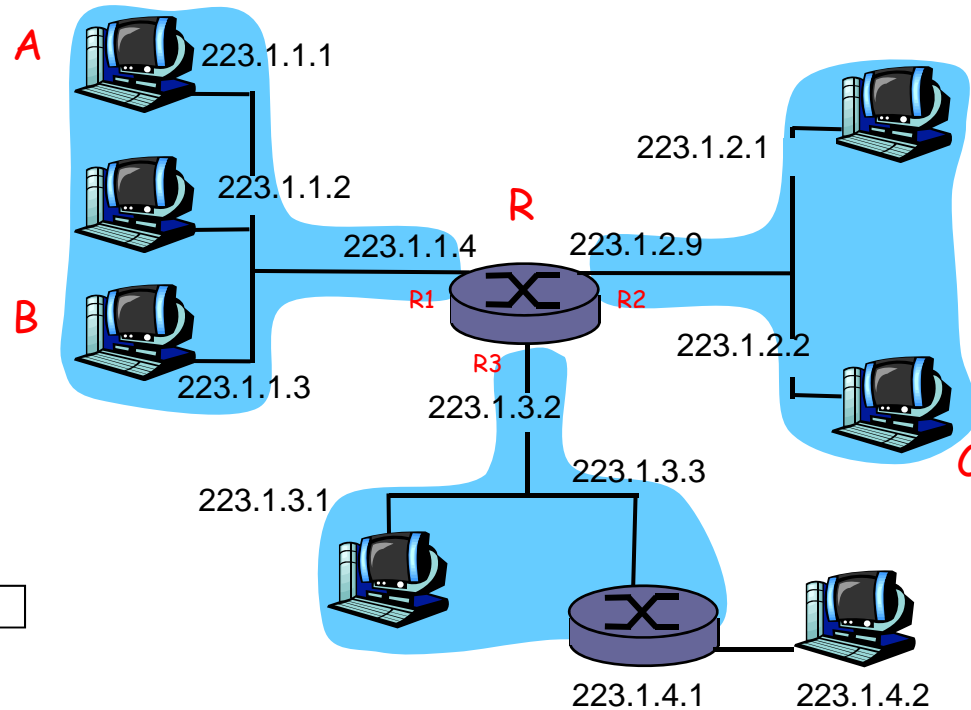
¿Cuál es la red IP en la que se encuentra cada máquina?

A: 223.1.1.0

B: 223.1.1.0

C: 223.1.2.0

(clases C, 24 bits NetID y 8 bits HostID puesto a 0)



Formato de trama resumida:

MAC <sub>origen</sub>	MAC <sub>destino</sub>	IP <sub>origen</sub>	IP <sub>destino</sub>
-----------------------	------------------------	----------------------	-----------------------

Paquete de A a B, visto en la red 223.1.1.0:

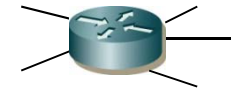
MAC <sub>A</sub>	MAC <sub>B</sub>	IP <sub>A</sub>	IP <sub>B</sub>
------------------	------------------	-----------------	-----------------

Paquete de A a C, visto en la red 223.1.1.0:

MAC <sub>A</sub>	MAC <sub>R1</sub>	IP <sub>A</sub>	IP <sub>C</sub>
------------------	-------------------	-----------------	-----------------

## Reenvío en redes classful

- En routers (varios interfaces de red)



- Configuración necesaria:

- Dirección  $IP_{interfaz}$  por cada interfaz, dentro del rango de red de la LAN a la que se conecta cada interfaz.
    - Tabla de rutas:
 

Red destino	Siguiete Salto	Interfaz

      - Red destino: Network ID.
      - Siguiete salto: IP del router que lleva a esa red.
      - Otros campos: Interfaz (ID del interfaz del router), Métrica, etc.

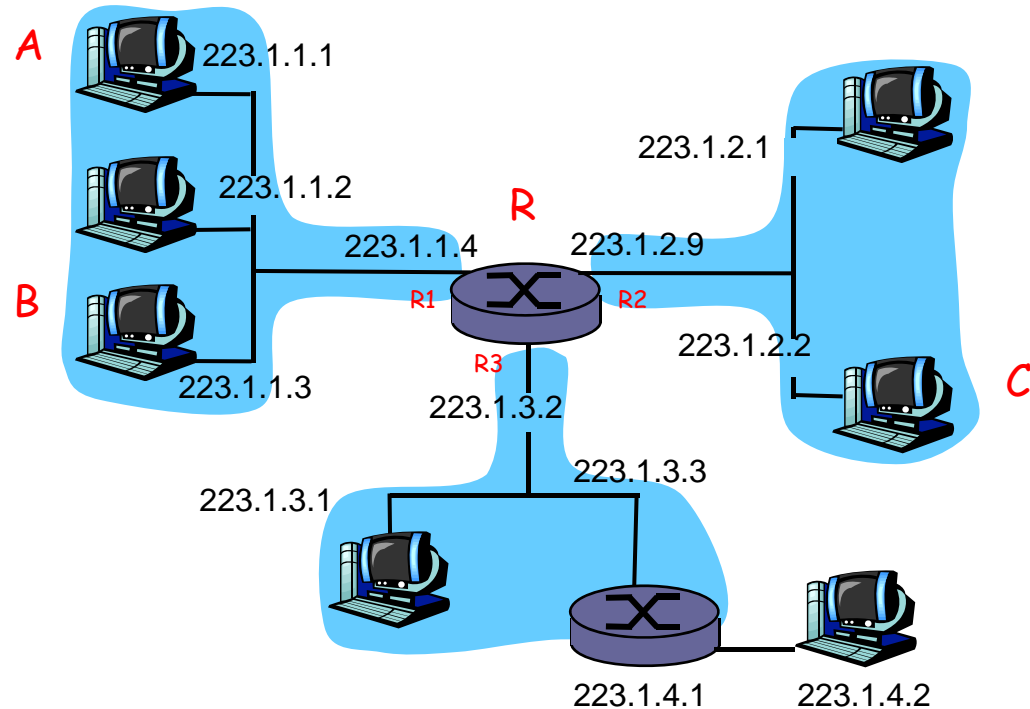
- Proceso de reenvío de un paquete a  $IP_{destino}$ :

- Busca una ruta directa a esa máquina en la tabla de rutas, buscando  $IP_{destino}$  en la columna Red destino. Si la encuentra hace lo que indique.
      - Si no, calcula el  $NetworkID_{destino}$  y lo busca en la tabla de rutas en la columna Red destino. Si lo encuentra hace lo que indique.
      - Si no, busca si existe ruta por defecto (Red destino=0.0.0.0) y si existe hace lo que indique esa fila.
      - Si no, tira el paquete. Se desconoce como reenviar el paquete.

# Ejemplo de reenvío en routers

Tabla de rutas de R:

Red destino	Siguiente Salto	Interfaz
223.1.1.0	-	1
223.1.2.0	-	2
223.1.3.0	-	3
223.1.4.0	223.1.3.3	3
0.0.0.0	223.1.3.3	3



Formato de trama resumida:

MAC <sub>origen</sub>	MAC <sub>destino</sub>	IP <sub>origen</sub>	IP <sub>destino</sub>
-----------------------	------------------------	----------------------	-----------------------

Paquete de A a C:

Paso 1: visto en la red 223.1.1.0 – de A a R (máquina A envía):

MAC <sub>A</sub>	MAC <sub>R1</sub>	IP <sub>A</sub>	IP <sub>C</sub>
------------------	-------------------	-----------------	-----------------

Paso 2: visto en la red 223.1.2.0 - de R a C (router R reenvía):

MAC <sub>R2</sub>	MAC <sub>C</sub>	IP <sub>A</sub>	IP <sub>C</sub>
-------------------	------------------	-----------------	-----------------

## 5.2 Reenvío en redes con subnetting FLSM y VLSM

- En máquinas
  - Como en redes classful pero considerando la máscara de la subred para obtener el ExtendedNetID.
- En routers
  - En subnetting básico sólo una entrada de la tabla de rutas puede ser válida para un destino concreto.
  - Proceso, para una  $IP_{\text{destino}}$  dada:
    - Buscar ruta directa a la máquina  $IP_{\text{destino}}$ 
      - Si: lo envía según lo que indique la tabla.
      - No: extrae el  $NetID_{\text{destino}}$  classful.
    - Entradas de la tabla de rutas con máscara (VLSM)
      - Si: extrae los  $ExtendedNetID_{\text{destino}}$  con la máscara de cada entrada.
      - No: algún interfaz con máscara (FLSM)
        - Si: extrae el  $ExtendedNetID_{\text{destino}}$  con la máscara del interfaz.
    - Busca ruta a la subred (ExtendedNetID) y si no a la red (NetID) destino
      - Si: lo envía según la tabla de rutas.
      - No: busca ruta por defecto y si no lo tira.



# Ejemplo reenvío con subnetting

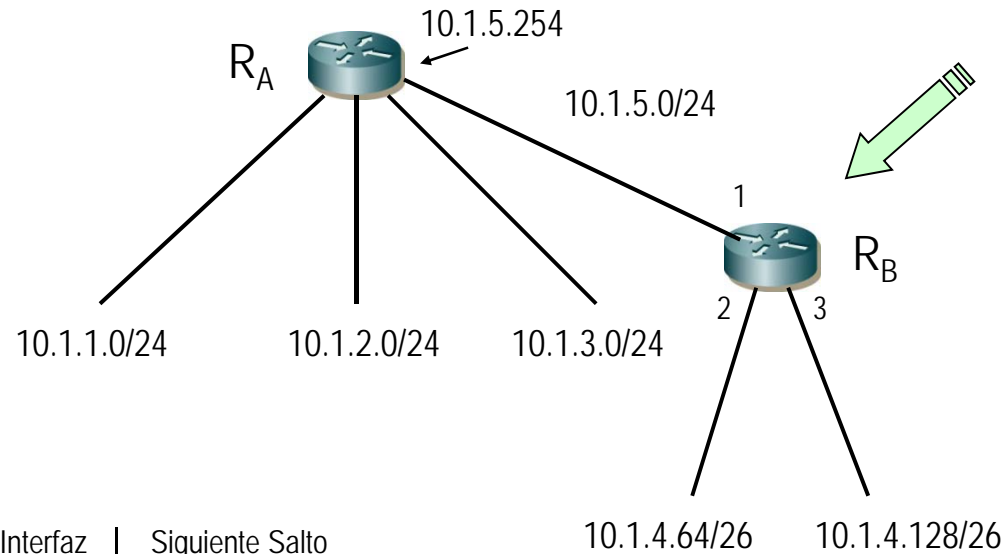


Tabla de rutas de R<sub>B</sub>:

Red/Máscara	Interfaz	Siguiente Salto
10.1.5.0/255.255.255.0	1	-
10.1.4.64/255.255.255.192	2	-
10.1.4.128/255.255.255.192	3	-
10.1.1.0/255.255.255.0	1	10.1.5.254
10.1.2.0/255.255.255.0	1	10.1.5.254
10.1.3.0/255.255.255.0	1	10.1.5.254
0.0.0.0/0.0.0.0	1	10.1.5.254

## 5.3 Reenvío en redes classless – CIDR

- Búsqueda de direcciones classless: algoritmo “longest-match” o búsqueda del prefijo más largo
  - Se queda con la entrada de la tabla de rutas cuya máscara sea mayor y el prefijo (identificador de red) coincida con el buscado.
  - Si hay varias entradas válidas de igual tamaño de máscara, se elige la que tenga el mismo TOS que el paquete, y si no la hay, la de TOS=0.

■ Ej:

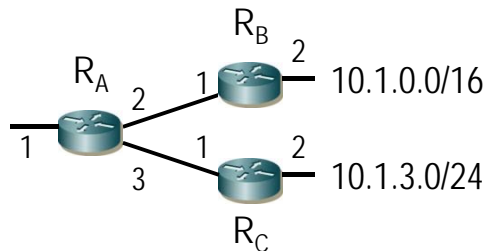


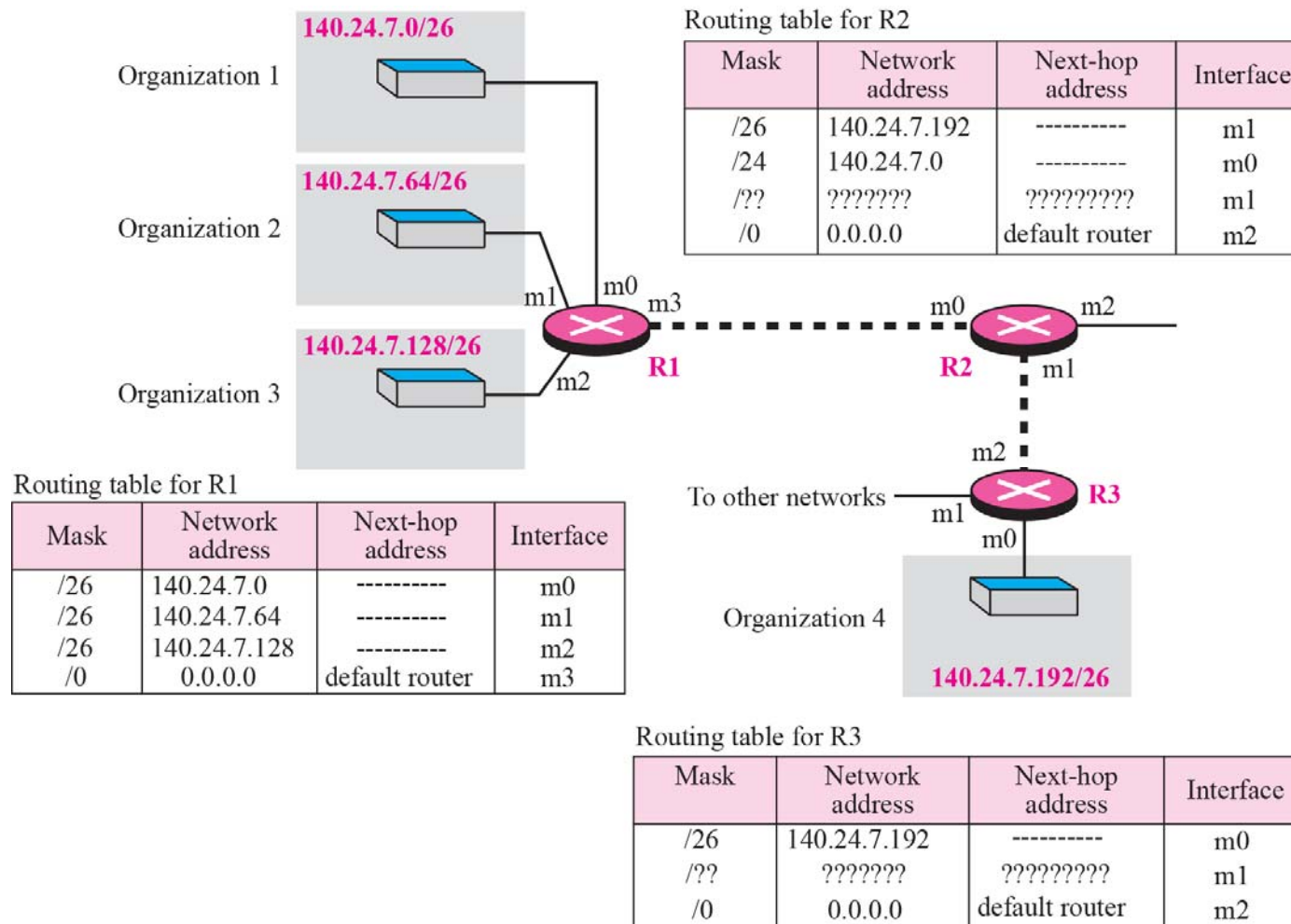
Tabla de rutas de  $R_A$ :

Red/Máscara	Interfaz	Siguiente Salto
10.1.0.0/16	2	$IP_{B,1}$
10.1.3.0/24	3	$IP_{C,1}$
0.0.0.0/0	1	-

(\*) *Faltan entradas ¿Cuáles?*

- Si  $IP_{dst}=10.1.3.7$ , la entrada 10.1.3.0/24 es la más larga que verifica.
- Si  $IP_{dst}=10.1.4.27$ , la entrada 10.1.0.0/16 es la más larga que verifica.
- Si  $IP_{dst}=10.2.5.5$ , ninguna coincide por lo que se toma la ruta por defecto.

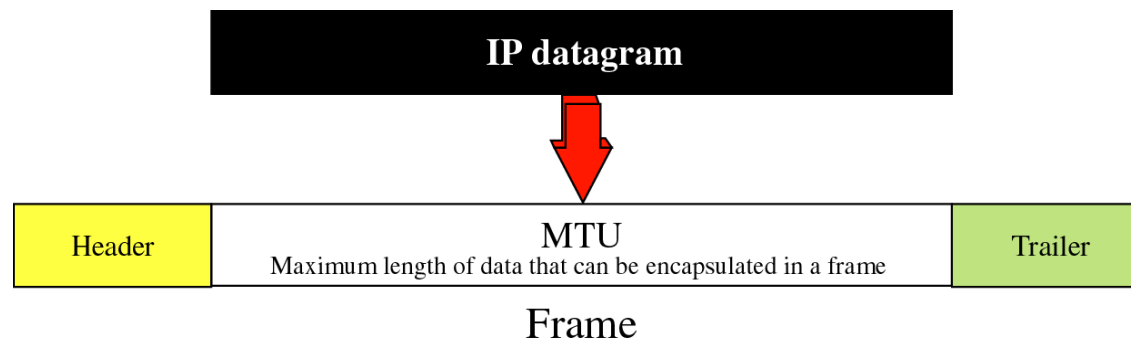
# Reenvío en CIDR: longest-match



Por longest-match la red 140.24.7.192/25 es accesible aunque haya otra red solapada con ella en otra ubicación

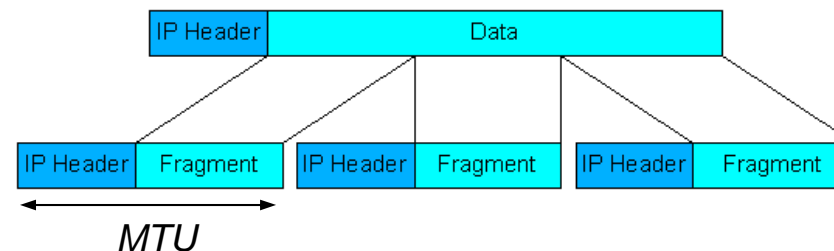
## 6 Fragmentación y reensamblado

- Existen redes de diferente tecnología con sus propios protocolos de nivel de enlace que especifican el máximo tamaño de datagrama IP que pueden encapsular: MTU.



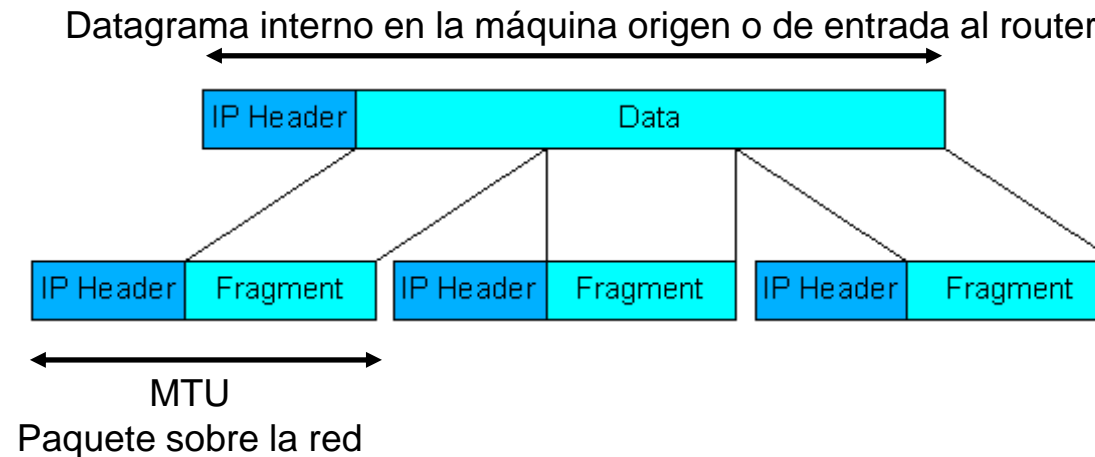
- Tamaño máximo de datagrama IP: 65535 bytes. Si un datagrama pasa a una red de MTU menor  $\Rightarrow$  necesidad de fragmentación.

Tecnología	MTU(bytes)
Token Ring	17.914
FDDI	4.352
Ethernet	1.500
Ethernet jumbo frames	9.000



## Quien fragmenta

- La máquina origen, si el datagrama IP es mayor que la MTU de la red en la que se encuentra conectada.
  - Situación que se suele desear evitar: el propio nivel de transporte genere datagramas limitados por el tamaño de la MTU.
- Un router, si la MTU de su enlace de salida por el que tiene que reenviar el paquete es más pequeña que el tamaño de paquete IP.



# Campos relacionados con fragmentación

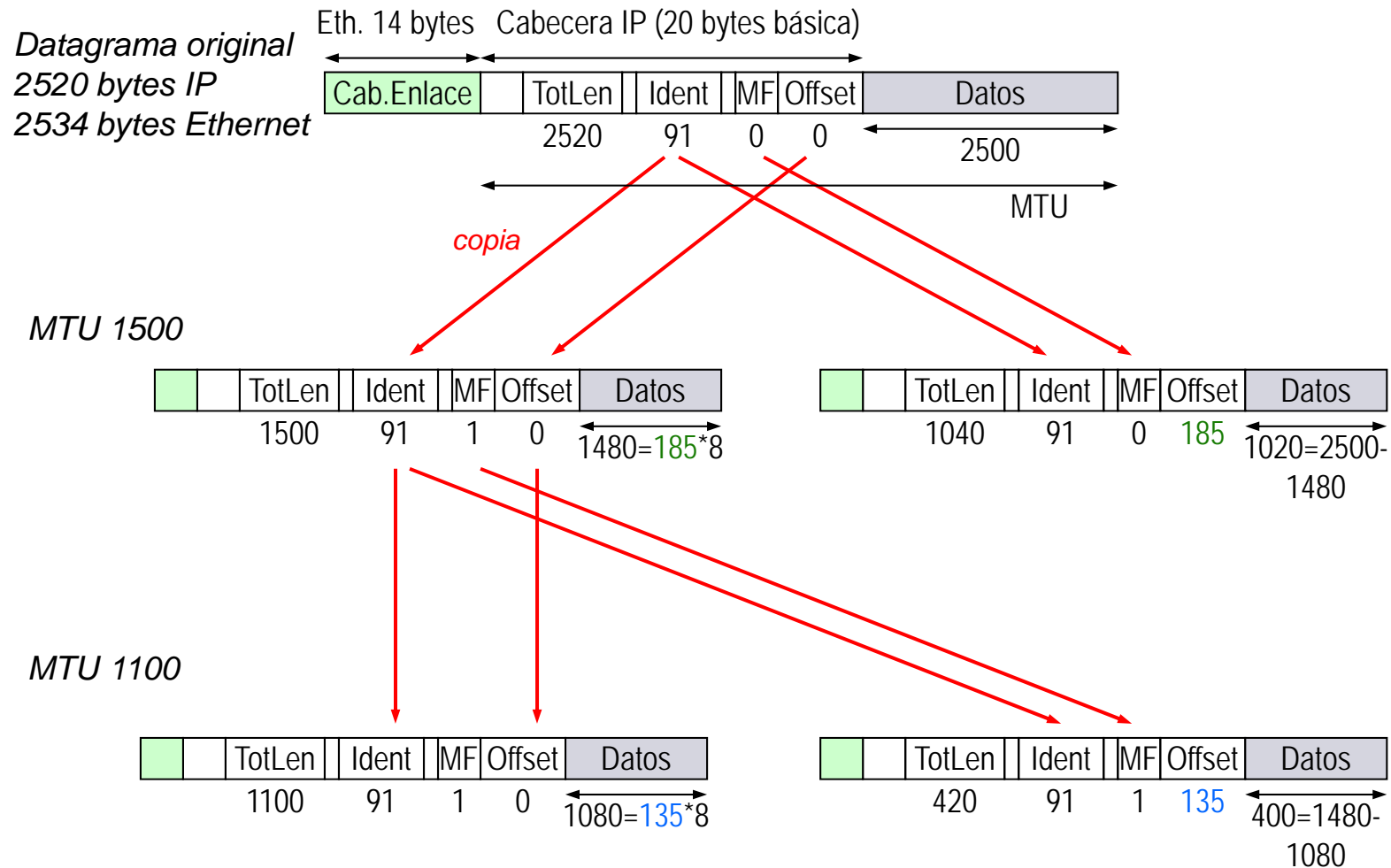
- Campos de la cabecera IP relacionados (en amarillo):
  - Identificador
  - Flag Más Fragmentos (MF)
  - Offset de fragmentación
- Otros campos de la cabecera IP modificados:
  - Longitud total
  - Checksum cabecera

VER 4 bits	HLEN 4 bits	Service type 8 bits	Total length 16 bits	
Identification 16 bits			Flags 3 bits <b>0 DF MF</b>	Fragmentation offset 13 bits
Time to live 8 bits		Protocol 8 bits	Header checksum 16 bits	
Source IP address				
Destination IP address				
<b>Option</b>				

## Proceso de fragmentación

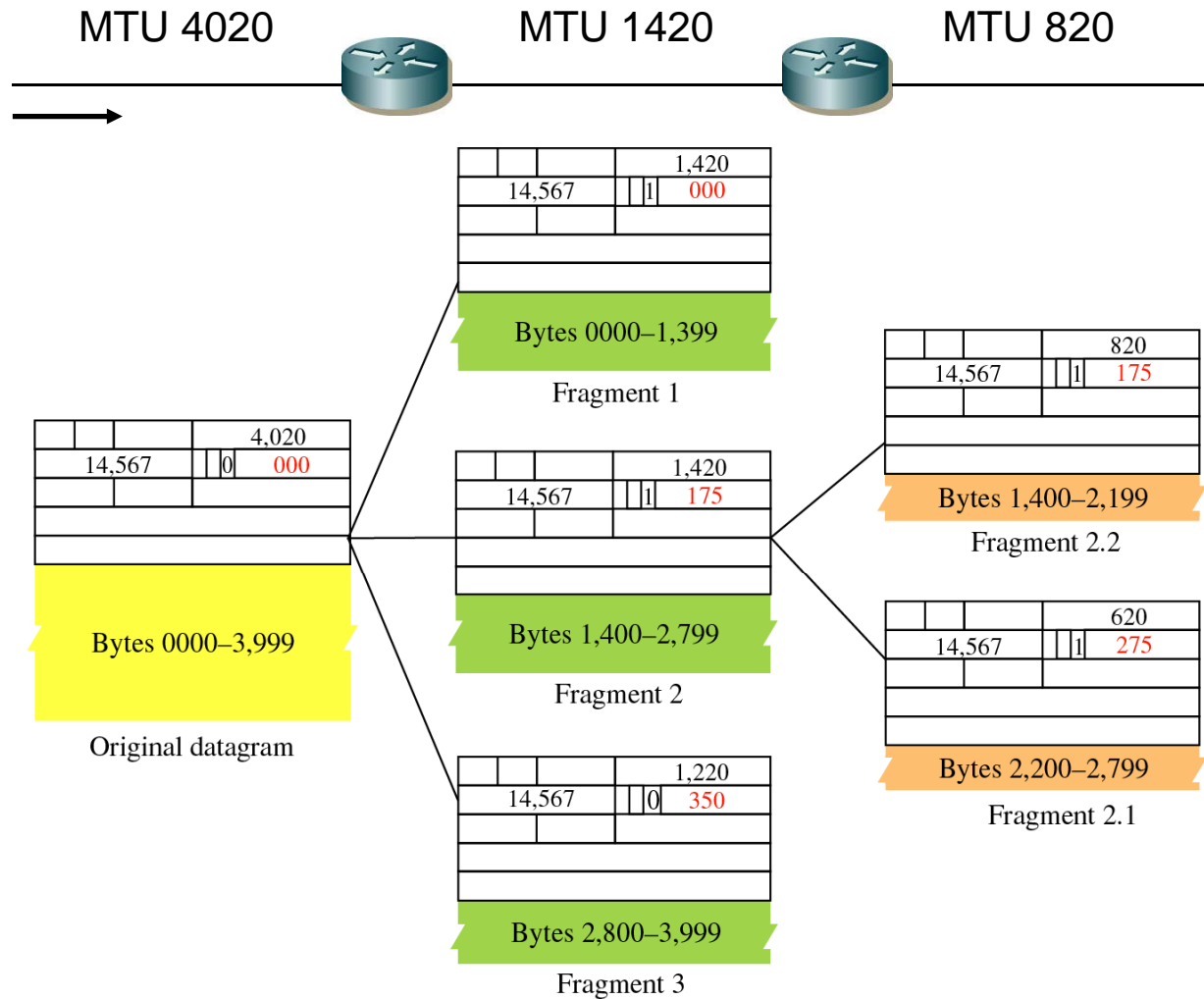
- Proceso de fragmentación a nivel IP:
  1. Todos los fragmentos copian la cabecera IP del original como base excepto CRC, TotalLength, flag MF y offset que se recalculan para cada fragmento.
  2. El tamaño de datos en cada fragmento deberá ser múltiplo de 8 bytes (excepto el último), por ser el offset medible en palabras de 8 bytes. Se escoge el tamaño de fragmento múltiplo de 8 bytes más cercano al límite impuesto por (MTU-HLEN).
  3. El primer fragmento copia el campo de offset del original. Para el resto de fragmentos se calcula según el tamaño de los fragmentos anteriores.
  4. Todos los fragmentos tendrán MF=1 excepto el último que copia el MF del original.
  5. El TotalLength y CRC se calcula para cada fragmento resultante.
  
- Analizando los fragmentos generados no se puede saber si proceden de uno o varios procesos de fragmentación.

# Fragmentación, ejemplo 1





# Fragmentación, ejemplo 2

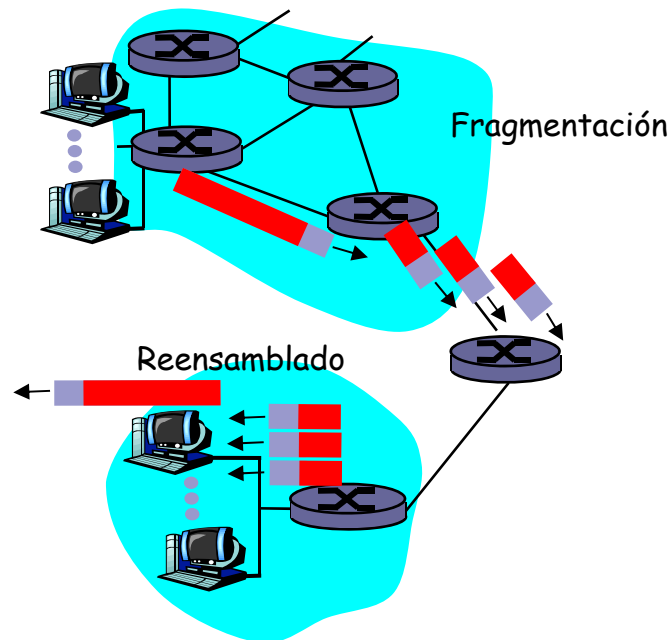


## Reensamblado

- Reconstruir el datagrama original mandado por el emisor.
- Se realiza en el destino final.
  - Los routers no reensamblan porque pueden no recibir todos los fragmentos ya que pueden ir por caminos diferentes.
  - Impacto negativo al no usar la máxima MTU en cada momento (sobrecarga de cabeceras).
- Reensamblado en base a (identificación, IPorigen, IPdestino, protocolo)  $\Rightarrow$  todos los fragmentos que verifican la tupla pertenecen al mismo datagrama original.
- Proceso de reensamblado:
  1. Existe un temporizador que se activa al recibir un fragmento de un datagrama nuevo.
  2. Al recibir un fragmento con MF=0 se puede conocer la longitud total del datagrama original.
  3. Si caduca el temporizador sin haber recibido todos los fragmentos, se tiran y el origen deberá retransmitir todo.

# Reensamblado

- Valor del temporizador depende de la implementación. Típicamente valores de 30sg.
- Todo nodo IP ha de ser capaz al menos de reensamblar datagramas de 576 bytes [RFC1032,STD43]. Limitaciones de memoria superadas en la actualidad.



## Resumen

- Router, dispositivo de interconexión de nivel 3
  - Núcleo del funcionamiento de Internet
  - Separa redes, dominios de colisión y dominios de broadcast
  - Funciones de reenvío basadas en la tabla de rutas
- Tabla de rutas manual
  - Ruta por defecto
  - Router siguiente salto
  - Router por defecto
- Reenvío por “longest-match” en el caso de direccionamiento classless CIDR, que permite la existencias de huecos y redes solapadas.
- Proceso de fragmentación para paquetes que no caben en la MTU de determinado segmento de red

# Referencias

- [Forouzan]
  - Capítulo 3, secciones 3.5 “Connectiong devices”
  - Capítulo 6, secciones 6.1- 6.2 “Delivery”, “Forwarding”
  - Capítulo 7, secciones 7.3 “Fragmentation”
- [Stevens]
  - Capítulo 9, secciones 9.1-9.2 “Introduction”, “Routing principles”
  - Capítulo 11, sección 11 “IP fragmentation”