

# Práctica 2: Fragmentación y reensamblado en IP

## 1. Sesión L2

Lectura previa: Kurose 4.4.1 **subapartado “Fragmentación del datagrama IP”**

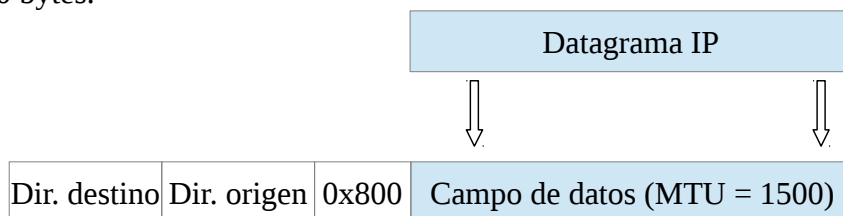
Trabajo previo antes de la sesión de laboratorio: Introducción (lectura), Fragmentación en IPv4 (estudio), Ejercicio 1 (realización).

## 2. Introducción

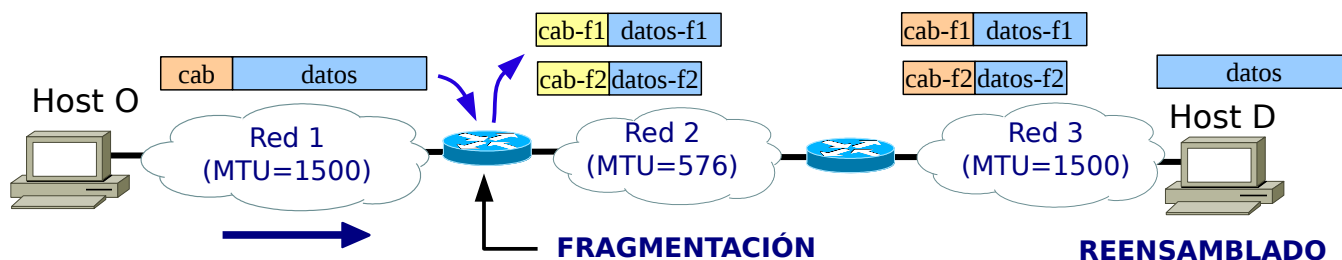
En esta práctica vamos a estudiar el problema de la fragmentación de datagramas IPv4.

Como ya hemos visto en clase, el tamaño máximo de un datagrama IP es de 64 KB, pero es más bien un valor máximo teórico y en la práctica suelen enviarse datagramas más pequeños.

Antes de transmitirse, el datagrama se encapsula en una trama, ocupando el campo de datos de la misma. Por lo tanto, el tamaño del datagrama estará limitado por el tamaño máximo del campo de datos de la trama que lo va a llevar. Este valor depende de la tecnología de red que se utilice. La mayoría de las tecnologías definen tamaños máximos, también conocidos como MTUs (*Maximum Transfer Unit*). Así, por ejemplo, Ethernet define una MTU de 1.500 bytes, PPPoE de 1.492 bytes o FDDI de 4.470 bytes.



Cuando se emplea TCP, el tamaño inicial del segmento TCP ya se elige de forma que el datagrama IP resultante quepa en el campo de datos de la trama en la que se va a encapsular. Desgraciadamente, incluso con esta precaución, el datagrama puede necesitar fragmentarse en trozos más pequeños si en su tránsito hacia el destino tiene que atravesar una red con una MTU menor que el original. El router que separa las dos redes se encargará de esta tarea, antes de reenviar el datagrama a la red de salida. Posteriormente, el host destino tendrá que reensamblar el datagrama original una vez recibidos todos los fragmentos.

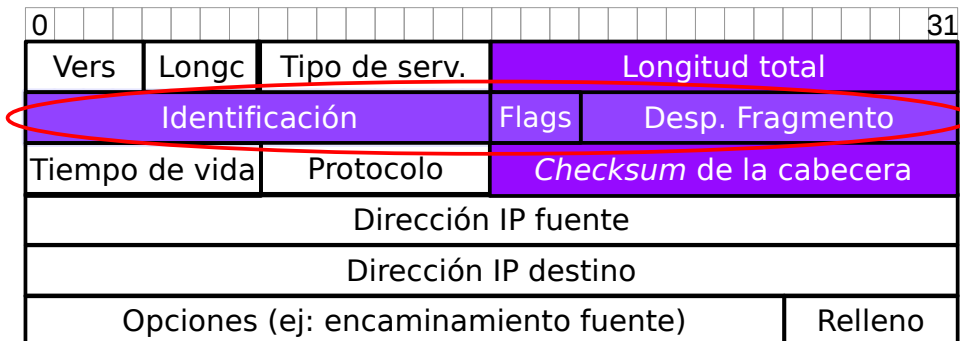


Cuando se emplean otros protocolos distintos de TCP, como UDP o ICMP, el problema de la fragmentación puede incluso plantearse en el propio host origen, ya que UDP o ICMP no tienen en cuenta la MTU a la hora de generar sus unidades de datos.

Las implementaciones de IP no están obligadas a manejar datagramas sin fragmentar mayores de 576 bytes, aunque la mayoría podrá manipular valores mayores, que suelen estar por encima de 8192 bytes o incluso superiores, y de forma ocasional menores de 1500.

### 3. Fragmentación en IPv4

Solo algunos de los campos de la cabecera del datagrama intervienen en el proceso de fragmentación. Son los que aparecen coloreados en el siguiente esquema de la cabecera:



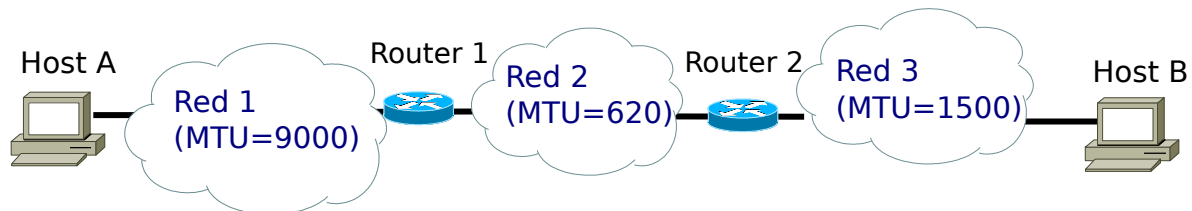
- El campo de **longitud total**, que define el tamaño total del datagrama (cabecera + datos) en bytes, pasa a indicar el tamaño del fragmento.
- El campo de **identificación** es un entero de 16 bits que identifica de forma única al datagrama original. Permite identificar a los fragmentos que pertenecen al mismo datagrama, dado que todos los fragmentos de un datagrama heredan el identificador del datagrama original.
- **Flags**: 3 bits (pero el de más peso no se emplea). Se utilizan sólo para especificar valores relativos a la fragmentación de paquetes:
  - Do not Fragment (**DF**): Indica que el datagrama no debe fragmentarse.
  - More Fragments (**MF**): Si está activado indica que este fragmento no es el último de la serie. Se utiliza en el destino final del datagrama durante el reensamblado.
- **Desplazamiento** del fragmento: Es un campo de 13 bits. Indica la posición del fragmento dentro del datagrama original en múltiplos de 8 bytes, es decir referido a bloques de 64 bits. El primer fragmento será el de desplazamiento cero (y bit MF=1).
- **Checksum** de la cabecera: Tiene la finalidad de proteger frente a posibles errores en la cabecera del datagrama. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el tiempo de vida).

Puede ser necesario volver a fragmentar un datagrama ya fragmentado. En ese caso, todos los fragmentos tienen el mismo nivel y el desplazamiento se refiere al datagrama original.

El reensamblado se realiza siempre en el receptor, y requiere recibir todos los fragmentos del datagrama en un tiempo acotado, antes de que venza un temporizador. El temporizador se inicia al recibir el primer fragmento del datagrama (el que llega primero, aunque no sea el de desplazamiento cero). Si el temporizador vence se descartan los fragmentos ya recibidos. En caso necesario, si el protocolo de nivel superior, por ejemplo TCP, solicita una retransmisión habrá que volver a enviar el datagrama completo de nuevo.

## Ejemplo de fragmentación

Dada la red del esquema siguiente, el host A envía un datagrama de longitud total 1620 bytes al host B. Dado que el datagrama tiene una longitud mayor de 620 bytes (MTU de la red 2), cuando el router 1 lo reenvíe se verá obligado a fragmentarlo.



El datagrama original y los fragmentos son los siguientes:

Lon. total 1620	Identif. 32	DF=0 MF=0	Desplaz. 0	Datos 1 (600 oct)	Datos 2 (600 oct)	Datos 3 (400)
--------------------	----------------	--------------	------------	-------------------	-------------------	---------------

Lon. total 620	Identif. 32	DF=0 MF=1	Desplaz. 0	Datos 1	MTU = 620 octetos	
Lon. total 620	Identif. 32	DF=0 MF=1	Desplaz. 75 (600)	Datos 2		
Lon. total 420	Identif. 32	DF=0 MF=0	Desplaz. 150 (1200)	Datos 3		

A la hora de calcular la cantidad de datos IP que caben en una trama hay que tener en cuenta:

- Que la cabecera IP ocupa 20 bytes, si como es habitual no lleva opciones. El resto de la MTU, en este caso  $620 - 20 = 600$ , es lo que queda disponible para los datos IP. En nuestro ejemplo, el datagrama original llevaba 1.600 bytes de datos IP que tendrán que ser distribuidos en fragmentos que **como máximo** lleven 600 bytes de datos IP, si la condición analizada en el apartado b) lo permite. 3 paquets
- La cantidad de datos que se incluye en cada fragmento exceptuando el último debe ser divisible entre 8, debido a la forma en que se expresa el desplazamiento del fragmento. En este caso,  $600 \div 8 = 75$ , dado que 600 es divisible entre 8, todo cuadra perfectamente. Además, el desplazamiento, que será múltiplo de 600 en los diferentes fragmentos, realmente aparecerá en la cabecera IP de los fragmentos expresado en múltiplos de 75 dado que el campo desplazamiento tiene 13 bits de longitud y esto obliga a expresar los desplazamientos en múltiplos de 8 bytes.

Cabe destacar que cuando se usan otros tamaños típicos de MTU, como 576, no todo cuadra tan bien. En este caso tenemos que  $576 - 20 = 556$ ,  $556 \div 8 = 69,5$ . Dado que 556 no es divisible entre 8, en este caso sólo se podrían aprovechar 552 de los 556 bytes disponibles en la MTU, para que la división dé un valor exacto. En el caso de una secuencia de fragmentos, el desplazamiento real sería múltiplo de 552 pero aparecería expresado en el campo de desplazamiento en múltiplos de 69.

**Ejercicio 1.** Supón que la capa de red quiere pasar un datagrama de 3500 bytes a la capa de enlace que tiene una MTU de 1500 bytes.

- ¿Cuántos fragmentos se van a generar? Justifica la respuesta.  $3480 / 1480 = 2,35 = 3$  paquets
- ¿Qué valor tienen los campos desplazamiento del fragmento en cada uno de ellos? (recuerda que el valor del campo de desplazamiento es la posición del primer byte del fragmento dividida por ocho).  $\text{múltiplos de } 185: 0, 185, 370.$
- Suponiendo que el datagrama IP lleva encapsulado un datagrama UDP, ¿cuántos bytes transmitidos pertenecen a cabeceras y cuántos a datos? Suponiendo que el protocolo del nivel de enlace emplea una cabecera de 16 bytes, ¿cuál es el valor en porcentaje de bytes de datos frente a bytes totales transmitidos?  $\text{cabecera} = (3 * 20) + 8 + 16 = 84 \text{ B}$  2,14% de capçaleres  
 $\text{dades} = 3500 - (8 + 16) - 20 = 3456 \text{ B}$  97,86% de dades

NOTA: tamaño de la cabecera IP sin opciones: 20 bytes. Tamaño cabecera UDP: 8 bytes.

## 4. Análisis de tráfico

No podemos observar directamente en el laboratorio la fragmentación que se produce en los routers, pero podemos utilizar un pequeño truco para generar fragmentación en nuestro propio equipo de prácticas.

Como hemos comentado en la introducción, los protocolos ICMP y UDP no tienen en cuenta el tamaño de la MTU local a la hora de generar sus unidades de datos: paquetes ICMP o datagramas UDP, respectivamente. Muchos sistemas operativos como Linux, Microsoft Windows o MAC OS nos proporcionan una orden llamada “ping” que nos permite enviar a un destino paquetes ICMP de eco con la cantidad de datos ICMP que especifiquemos, y esperar la respuesta asociada. Si el tamaño total del paquete ICMP (cabecera y campo de datos) que se va a enviar más el tamaño de la cabecera IP exceden la MTU local, la capa IP de nuestro host se verá obligada a fragmentar el datagrama que contiene el paquete ICMP.

**Ejercicio 2.** Prepara una captura de tráfico con el programa wireshark. Indica en el filtro de tráfico el protocolo “icmp”, en el wireshark Capture → Options → Capture filter (para ver únicamente el tráfico enviado o recibido por tu computador añade al filtro tu dirección IP, esto es, icmp and host 158.42.180.xx). A continuación abre un shell y teclea:

```
> ping -c 1 -s 3972 zoltar.redes.upv.es
```

La opción -c es para que se envíe un único paquete, ya que por defecto como se verá en la práctica donde se estudia con detalle el protocolo ICMP, la orden ping en Linux envía paquetes de forma ininterrumpida. La opción -s indica el tamaño del campo de datos ICMP (el paquete ICMP también tendrá una cabecera).

Como estamos conectados a una red Ethernet, un envío con -s 3972 exigirá la fragmentación del paquete en varios paquetes IP (¿por qué se producirá dicha fragmentación?).

- Para el datagrama enviado por tu ordenador, compara las cabeceras de los fragmentos generados, fijándote especialmente en los campos **longitud total**, **flags** y **desplazamiento del fragmento** (*fragment offset* en la captura de wireshark). Para ello ayúdate de la tabla siguiente, donde puedes anotar los valores de estos campos.

<i>Identificador Fragmento</i>	<i>Flag DF</i>	<i>Flag MF</i>	<i>Desplazamiento</i>	<i>Longitud total</i>
0xda7a	0	1	0	1500
0xda7a	0	1	1480	1500
0xda7a	0	0	2960	1040

- b) ¿Cuál es el valor del campo protocolo de la cabecera de los tres fragmentos? ¿Debe ser el mismo para todos los fragmentos? *los dos primeros ipv4 y el ultimo icmp*
- c) ¿Cuál es el valor del campo desplazamiento enviado en la cabecera IP del segundo fragmento? Wireshark muestra el valor del desplazamiento ya calculado, no el que realmente se envía. Comprueba en la pestaña inferior que muestra los bytes enviados en hexadecimal cuál ha sido el valor realmente enviado. Recuerda que el tamaño del campo de desplazamiento es de 13 bits. *185*
- d) Calcula el tamaño del mensaje que deberíamos enviar para que se generaran cuatro fragmentos de tamaño máximo. Para este cálculo hay que tener en cuenta cuánto ocupa la cabecera ICMP. La longitud de la cabecera ICMP hay que calcularla viendo cuánto ocupa cada uno de sus campos en la pestaña inferior de la captura. *1500 \* 4 = 6000 B*  
*6000 - (4 \* 20) = 5920 B -> 5920 - 8 = 5912 B de datos*
- Comprueba que dicho tamaño de mensaje es correcto capturando el tráfico generado tras ejecutar nuevamente la orden **ping** sustituyendo 3972 por el tamaño de mensaje calculado.
- e) ¿Cuántos bytes de datos IP viajan en cada paquete? ¿Y de datos ICMP? Para el cálculo puedes ayudarte de las cabeceras “Header Length” y “Total Length” del datagrama IP. *1500 - 20 = 1480 B datos ip y 20 B cabecera*

**Ejercicio 3.** Las MTUs de las redes 1 y 2 son 4500 y 800 respectivamente. En el computador B de la red 2 se han recibido los siguientes datagramas IP. El emisor de dichos datagramas es el computador A de la red 1.

<i>Campos de la cabecera IP</i>				
<i>Longitud total</i>	<i>Identificador</i>	<i>DF</i>	<i>MF</i>	<i>Desplazamiento</i>
796	16	0	0	194 <i>3.16</i>
40	28	0	0	194 <i>3.28</i>
796	16	0	1	0 <i>1.16</i>
796	28	0	1	0 <i>1.28</i>
780	63	0	0	0 <i>paquet unic</i>
796	16	0	1	97 <i>2.16</i>
796	95	0	1	291 <i>4.95</i>
796	28	0	1	97 <i>2.28</i>
54	95	0	0	388 <i>5.95</i>

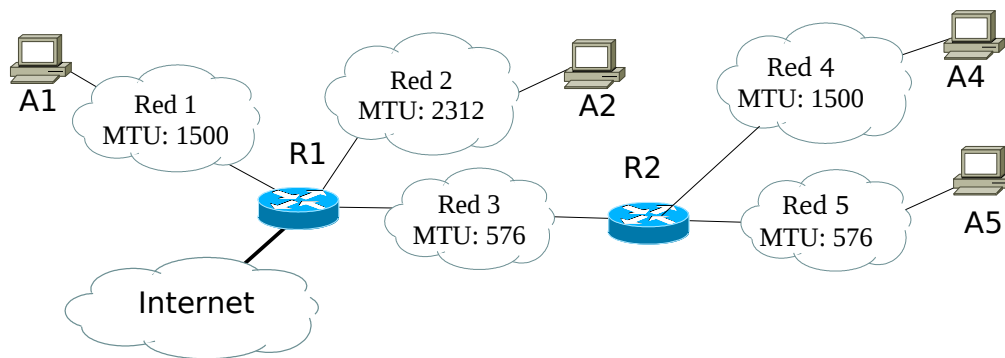
- a) ¿Tienen alguna relación entre sí los distintos datagramas recibidos? Justifica la respuesta. *si, que el desplazamiento es 97*
- b) Rellena la tabla con los valores de los datagramas cuando los emitió A.

Longitud total	Identificador	Flag DF	Flag MF	Desplazamiento
2348	16	0	0	0
1572	28	0	0	0
780	63	0	0	0
3158	95	0	0	0

- c) ¿Serán entregados al nivel superior todos los datagramas recibidos?

*no, del paquete 95 aun les faltan 3 fragmentos*

**Ejercicio 4. (OPCIONAL)** Considera la red de la figura:



- a) Supongamos que A1 envía a A2 un datagrama de 1500 bytes de longitud total con el flag DF desactivado. Deduce qué operación va a realizar el router R1, qué información enviará y a quién. *reenviaría els 1500 B porque el MTU de la red 2 es mayor*
- b) Supongamos que A1 envía otro datagrama de 1500 bytes a A2 pero con el flag DF activado. Deduce qué operación hará el router R1, que información va a enviar y a quién. *no pasa nada porque no tiene que fragmentar el paquete*
- c) Ahora, A2 envía un datagrama de 2312 bytes de longitud total a A1 con el flag DF desactivado. Indica qué información recibirá A1. ¿Y si A2 enviara otro datagrama con el flag DF activo? Si es necesario haz uso de la tabla siguiente:

Fragmento	Flag DF	Flag MF	Desplazamiento	Longitud total
1	0	1	0	1500
1	0	0	185	832
<i>no puede fragmentar el paquete por tanto lo descartaria</i>				

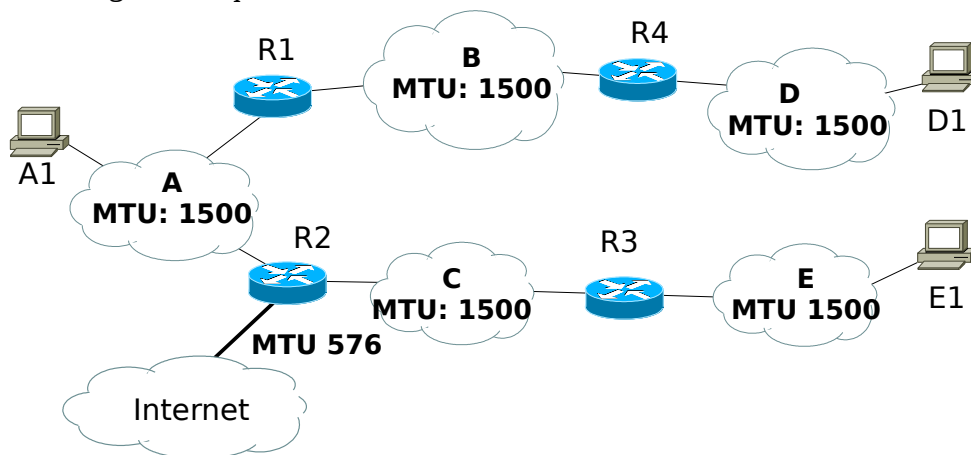
d) A1 envía un datagrama de 1500 bytes a A4 con el flag DF desactivado. Indica los fragmentos que R1 enviará a R2 y los que R2 enviará a A4.

Fragmento	Flag DF	Flag MF	Desplazamiento	Longitud total
1	0	1	0	572
1	0	1	69	572
1	0	0	138	396

e) Como el apartado d) pero el datagrama que envía A1 está destinado a A5.

Fragmento	Flag DF	Flag MF	Desplazamiento	Longitud total
igual que antes pero ahora el router r2 reenviara los 3 fragmentos como esta				

**Ejercicio 5. (OPCIONAL)** En el *host* D1 tenemos un proceso que a través de UDP quiere enviar un mensaje de 688 octetos a otro proceso en un *host* de una red remota (Internet). Suponiendo que el enlace con Internet (vía R2) tiene una MTU de 576 octetos, y que la cabecera IP no lleva opciones, indica los valores de los siguientes campos de la cabecera IP: Identificación, bit MF, desplazamiento, tamaño total y dirección IP origen del datagrama original que envía D1 y de los fragmentos que salen de R2 hacia Internet.



fragmento	df	mf	desplazamiento	longitud total
1	0	1	0	572
1	0	0	69	164