Práctica 2: Fragmentación y reensamblado en IP

Esta práctica se realiza preferiblemente en un entono Linux. No obstante, es posible realizarla en Windows 10, con resultado equivalente

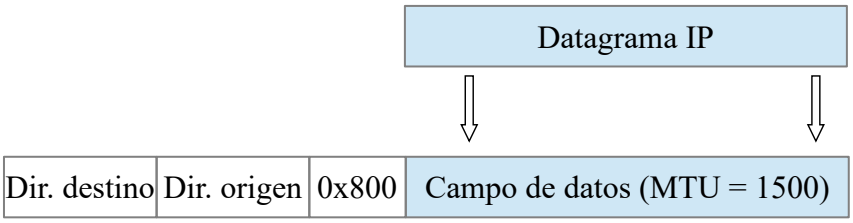
Lectura previa: Kurose 4.3.2 ***subapartado "Fragmentación del datagrama IPv4"***

1. Introducción

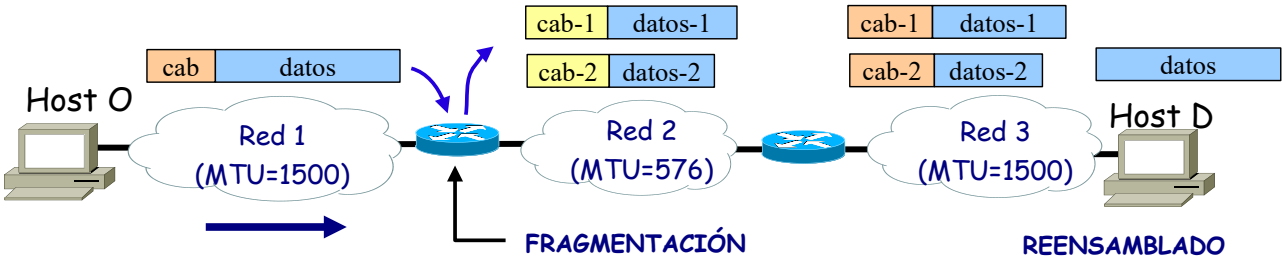
En esta práctica vamos a estudiar el problema de la fragmentación de datagramas IPv4.

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. En la práctica suelen enviarse datagramas más pequeños.

Para transmitirse, el datagrama debe encapsularse 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 transporta. 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 máximo 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 la red original. El *router* que separa las dos redes se encargará de esta tarea antes de reenviar el datagrama a la red de salida. Posteriormente, cada uno de los fragmentos viajan por separado hasta que llegan al host destino, que tendrá que reensamblar el datagrama original una vez recibidos todos los fragmentos.

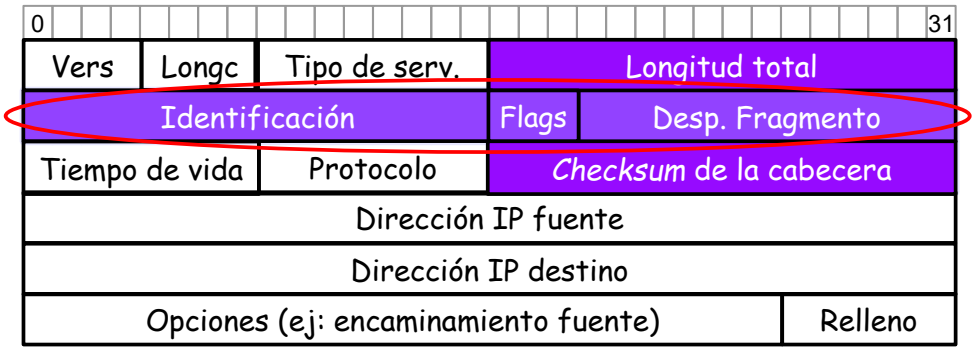


Cuando se emplean otros protocolos distintos de TCP, como UDP o ICMP, el problema de la fragmentación puede incluso plantearse en el propio host origen, ya que UDP o ICMP no tienen en cuenta la MTU a la hora de generar sus unidades de datos.

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.

2. Fragmentación en IPv4

Solo algunos de los campos de la cabecera del datagrama están involucrados 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. Tras la fragmentación, pasa a indicar el tamaño del fragmento.
- El campo de **identificación** es un entero de 16 bits que identifica de forma única a cada datagrama transmitido por un host, etiquetando 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: Son tres bits, aunque el de más peso no se emplea. Los dos restantes se utilizan para especificar condiciones relativas a la fragmentación de paquetes:
  - Do not Fragment (**DF**): Cuando está a '1', indica que el datagrama no debe fragmentarse. Si para reenviar un paquete IP con este bit activo es necesario fragmentar, no se reenviará, sino que se descartará y se informará al origen mediante un mensaje ICMP.
  - More Fragments (**MF**): Si está a '1' indica que este fragmento no es el último de la serie. Así, tendrá este valor en todos los fragmentos menos el último. Se utiliza en el destino final del datagrama durante el reensamblado.
- **Desplazamiento** del fragmento: Es un campo de 13 bits, que indica la posición del fragmento dentro del datagrama original. Puesto que la longitud de este campo (13 bits) es

tres bits menor que la del campo Longitud total (16 bits), el desplazamiento de los datos se expresa en múltiplos de 8 bytes, es decir referido a bloques de 64 bits. Ello conlleva que el campo de datos de un fragmento que no sea el último debe tener un tamaño múltiplo de 8 bytes para poder expresar correctamente el desplazamiento del siguiente fragmento. el último fragmento no debe cumplir esta restricción en tamaño, al no existir fragmentos posteriores, y se marca mediante el bit MF=0. Por otro lado, el primer fragmento será el de desplazamiento cero y bit MF=1.

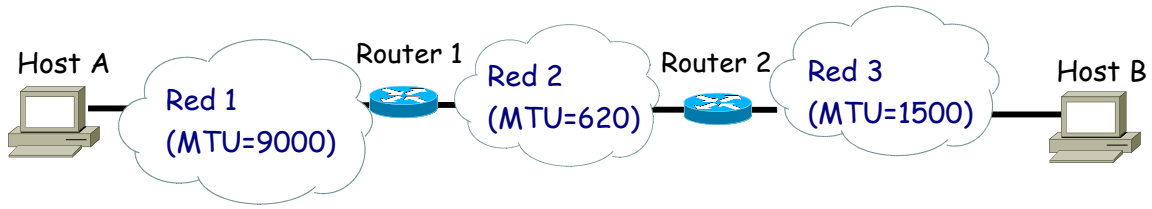
- ***Checksum*** de la cabecera: Tiene la finalidad de proteger frente a posibles errores en la cabecera del datagrama. Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el tiempo de vida). En concreto, tras la fragmentación se alteran múltiples campos de la cabecera y por tanto es necesario recalcularlo.

Puede ser necesario volver a fragmentar un datagrama ya fragmentado, por ejemplo, si atraviesa otra red con una MTU menor. En ese caso, el desplazamiento de todos los fragmentos se refiere al datagrama original.

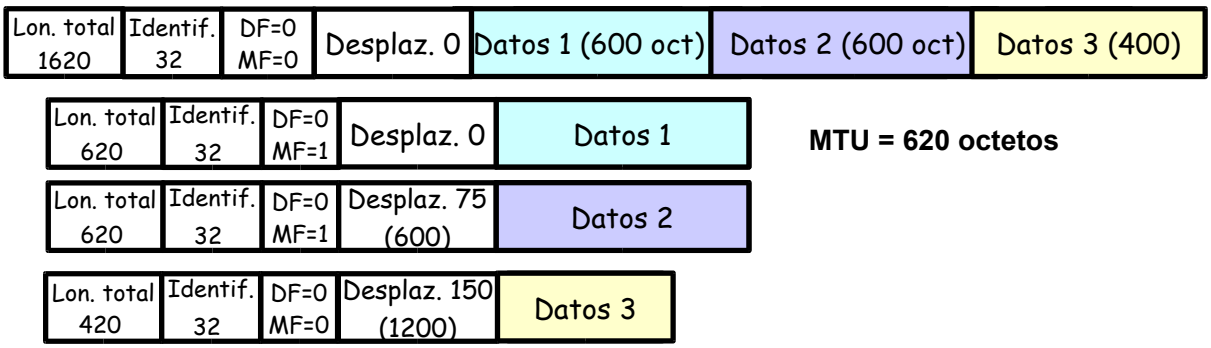
El reensamblado se realiza siempre en el receptor, y requiere recibir todos los fragmentos del datagrama en un tiempo acotado, antes de que venza un temporizador. El temporizador se inicia al recibir el primer fragmento del datagrama (el que llega primero, aunque no sea el de desplazamiento cero). Si 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:



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

a) Que la cabecera IP ocupa 20 bytes, si no lleva opciones, como es habitual. 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.

b) 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 ÷ 8 = 75, dado que 600 es divisible entre 8, todo cuadra perfectamente. Además, el desplazamiento será múltiplo de 600 en los diferentes fragmentos. Sin embargo, los valores que realmente aparecerán en la cabecera IP de los fragmentos serán múltiplos de 75.

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 ÷ 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.

Un router recibe un datagrama de 3500 bytes. La red de salida en la que debe transmitirlo para que llegue a su destino tiene una MTU de 1500 bytes, por lo que el router debe fragmentar el datagrama.

Calcula el número de fragmentos que se generarán y el tamaño de cada fragmento. Incluye en tu respuesta los cálculos realizados.
Indica el valor que tiene el campo desplazamiento de la cabecera IP en cada uno de los fragmentos generados (Recuerda que el tamaño del campo de datos de todos los fragmentos exceptuando el último fragmento debe ser un valor divisible por 8).

Completa la tabla siguiente con los valores obtenidos.

Número de Fragmentos	*Longitud total/fragmento*	*Desplazamiento*	*Bit MF*

3. Análisis de tráfico

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

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. En la práctica anterior estudiamos la orden **ping**, que nos permite enviar a un destino paquetes ICMP de petición 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.

Vamos a preparar una captura de tráfico con el programa Wireshark. Para ello abre el Wireshark y aplica un filtro para ver únicamente el tráfico **icmp** enviado o recibido por tu computador:

Capture→Options→Capture filter for selected interfaces: *icmp and host xx.xx.xx.xx*

Debes sustituir *xx.xx.xx.xx* por la dirección IP de tu máquina. Recuerda que en la sesión anterior usamos el comando **ip address** de Linux para averiguarlo. De forma análoga, en Windows puedes emplear **ipconfig**.

Una vez lo tengas claro empieza la captura.

A continuación, abre un intérprete de órdenes de Linux y teclea:

```
> ping -c 1 -s 3972 www.rediris.es
```

El equivalente en Windows sería:

```
C:\> ping -n 1 -l 3972 www.rediris.es
```

Una vez terminado el **ping**, detén la captura de paquetes del wireshark.

La opción **`-c 1`** es para que se envíe un único mensaje ICMP de petición de eco al host especificado. Como se verá en la práctica donde se estudia con detalle el protocolo ICMP, la orden ping en Linux por defecto envía paquetes de forma ininterrumpida. La opción **`-s 3972`** indica que el mensaje ICMP lleva un campo de datos opcional de 3972 bytes, cuyo contenido no es relevante. Ten en cuenta que el paquete ICMP también tendrá una cabecera. A continuación, la orden ping muestra por pantalla información acerca de la respuesta, que será un mensaje ICMP de respuesta de eco. Estos mensajes se verán más detalladamente en la práctica correspondiente.

Como estamos conectados a una red Ethernet (cuya MTU es de 1500 bytes), un envío con -s 3972 exigirá la fragmentación del paquete en varios paquetes IP.

a) 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.

Identificador Fragmento	***Flag DF***	***Flag MF***	***Desplazamiento***	***Longitud total***

b) ¿Cuál es el valor del campo protocolo de la cabecera de los tres fragmentos? ¿Debe ser el mismo para todos los fragmentos?

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.

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.

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.

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.

Campos de la cabecera IP				
Longitud total	***Identificador***	***DF***	***MF***	***Desplazamiento***
796	16	0	0	194
40	28	0	0	194
796	16	0	1	0
796	28	0	1	0
780	63	0	0	0
796	16	0	1	97
796	95	0	1	291
796	28	0	1	97
54	95	0	0	388

a) ¿Tienen alguna relación entre sí los distintos datagramas recibidos? Justifica la respuesta.

b) Rellena la tabla con los valores de los datagramas cuando los emitió A.

Longitud total	***Identificador***	***Flag DF***	***Flag MF***	***Desplazamiento***

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