Práctica 2: Fragmentación y reensamblado en IP

*Lectura previa: Leed el apartado "**Fragmentación del datagrama IPv4**" en el libro recomendado de la asignatura (Kurose). Dependiendo de la edición, está en distintos puntos del capítulo 4. Se recomiendan las versiones 5ª y 7ª en castellano (que están en la biblioteca de la Universidad). En la 5ª edición está en la sección 4.4.1. En la 7ª, en la 4.3.2. En la 8ª edición no se explica ya que en IPv6 no se permite la fragmentación en los routers, aunque sí en origen, que fragmentará, si es necesario, el datagrama original de una manera similar a la que se describe en esta práctica para IPv4.*

1. Introducción

En esta práctica vamos a estudiar el problema de la fragmentación de datagramas IPv4. Como hemos visto en las sesiones de teoría, aunque el tamaño teórico máximo de un datagrama IP es igual a 64 KB, en realidad se envían datagramas más pequeños. La razón es que para transmitirse, el datagrama debe encapsularse en una trama de nivel de enlace, ocupando su campo de datos de la misma forma que hemos visto que los datos de aplicación se transportan en el campo de datos de los segmentos TCP. Por lo tanto, el tamaño del datagrama estará limitado por el tamaño máximo del campo de datos de la trama que lo va a transportar. 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.

En el tema 6 del curso estudiaremos en detalle el formato de las tramas *Ethernet*. No obstante, podemos destacar ahora los aspectos más relevantes que debemos saber para la realización de esta práctica. Como se muestra en la Figura 1, preámbulos y delimitadores de inicio y fin aparte, la trama *Ethernet* tiene un máximo de 1518 bytes. Si descontamos los 12 bytes que ocupan las dirección físicas origen y destino, los 2 bytes que indicarán el tipo o protocolo embebido en la zona de datos (p.ej, el código 0x800 significa que los datos contienen un datagrama IP) y los 4 bytes de CRC, nos quedarán 1518 – 18 = 1500 bytes, como máximo, para los datos. En esta zona de datos estará el datagrama IP (ver Figura 2). Ya sabemos, por tanto, que la longitud máxima del datagrama (MTU) no podrá exceder esos 1500 bytes en el caso de una trama *Ethernet*.

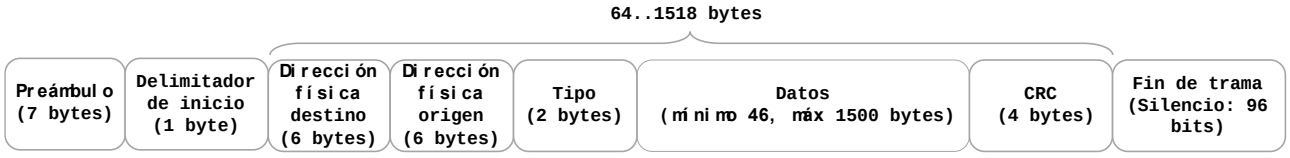


Figura 1. Formato de la trama Ethernet

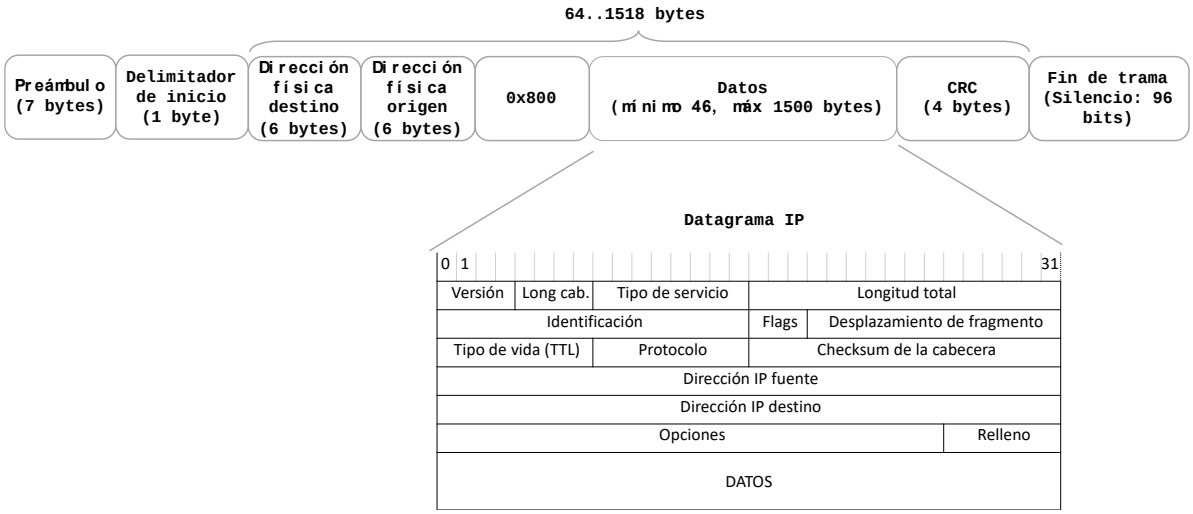


Figura 2. Datagrama IP en la trama Ethernet

El concepto de MTU es similar al ya conocido de MSS (*Maximum Segment Size*) en el nivel de transporte. ¿Recuerdas que cuando estudiamos el tema sobre TCP e hicimos prácticas obteníamos un MSS igual a 1460 bytes? ¿Ya tienes más claro por qué? Reflexiona sobre este aspecto teniendo en cuenta que las cabeceras IP y TCP (sin opciones) ocupan 20 bytes cada una. Piensa, viendo la Figura 2, qué información se encapsula en la zona de datos del datagrama IP.

Cuando se emplea TCP, el tamaño máximo del segmento 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 viaja por separado hasta que llega al equipo destino, que tendrá que reensamblar el datagrama original una vez recibidos todos los fragmentos. Estas ideas se muestran en la Figura 3. El datagrama que el equipo A quiere enviar al equipo B se genera con un tamaño acorde a la MTU de la red en la que se encuentra (Red 1). En su viaje, cuando llega al primer *router*, éste sabe que la Red 2 tiene una MTU menor a la longitud de los datos que lleva el datagrama que acaba de recibir. En ese momento determina que tiene que fragmentarlo para cumplir las restricciones de la Red 2. Por simplicidad podemos suponer que el datagrama original se va a fragmentar en 2 datagramas porque su longitud original está, por ejemplo, entorno a los 900 o 1000 bytes. (Si la longitud original fuera la máxima habría que fragmentarlo en 3 datagramas). De esta forma, se generará un primer datagrama con hasta los primeros 576 bytes de los datos originales (posteriormente concretaremos este aspecto) y los restantes hasta el total se incorporarán a un segundo datagrama. Y cada uno de esos datagramas tendrá su cabecera acorde con la información que lleva, su longitud total, checksum y sus campos relacionados con la **fragmentación**. Por eso se ha cambiado el color de las mismas, para resaltar que no son iguales. Como hemos dicho antes, una vez que un datagrama se fragmenta no se reensambla hasta que sus fragmentos llegan al destino. Como se ve en la figura, aunque la Red 3 vuelve a tener una MTU mayor, los datagramas siguen fragmentados hasta llegar al Equipo B, donde se hace el reensamblado.

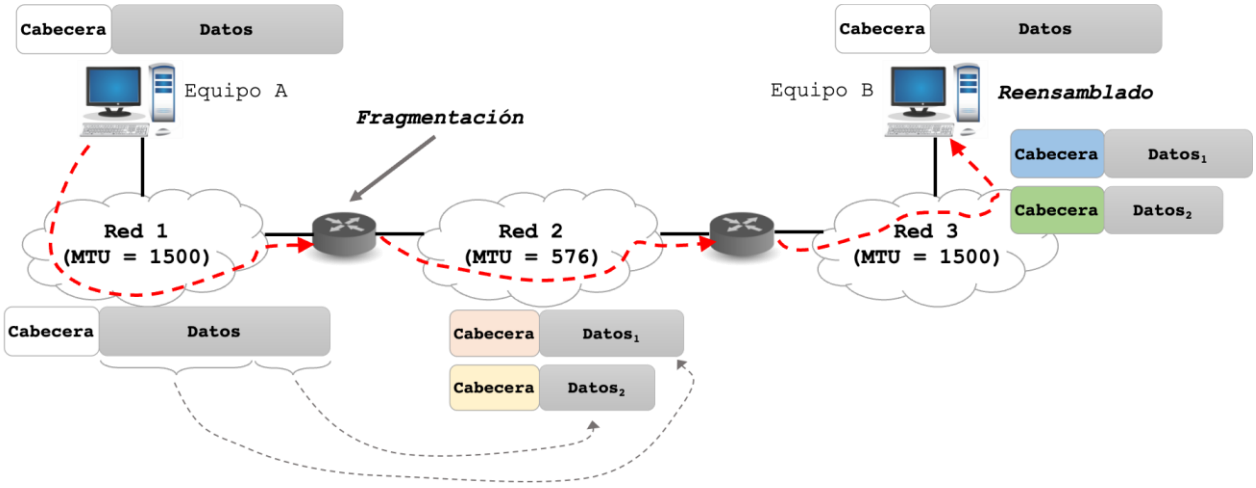


Figura 3. Fragmentación del datagrama al cambiar la MTU de la Red 2

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.

Todas las implementaciones de IP deben aceptar datagramas de hasta 576 bytes (tanto si llegan completos o en fragmentos), aunque la mayoría podrá manipular valores mayores, que suelen estar por encima de 8192 bytes o incluso superiores.

2. Fragmentación en IPv4

En IPv4, algunos de los campos de la cabecera del datagrama están involucrados en el proceso de fragmentación. Son los que aparecen coloreados en el formato del datagrama mostrado en la Figura 4.

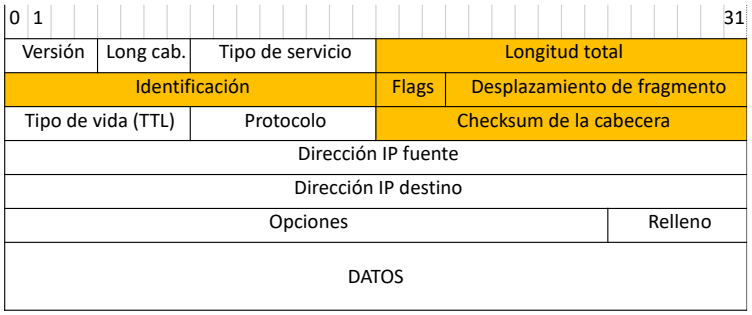


Figura 4. Datagrama IPv4

- El campo de **longitud total** define el tamaño total del datagrama (cabecera + datos) en bytes. En caso de ser necesaria la fragmentación, pasa a indicar el tamaño del fragmento.
- El campo de **identificación** es un número 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 se puede fragmentar. 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 en 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 que se muestra en la Figura 5, supongamos que el equipo A desea enviar un datagrama de una longitud total igual a 1620 bytes. Al estar en una red (Red 1) que soporta esa longitud (y mayores) generará un único datagrama de esa longitud (el denominado "original"). Cuando este datagrama llegue al Router 1, éste tendrá que realizar la fragmentación, ya que el tamaño máximo admitido en las tramas de la Red 2 es inferior. Por eso, el datagrama original va a ser dividido en tres fragmentos. Así, mientras que en la Red 1 ha circulado un datagrama de 1620 bytes, en la Red 2 van a circular 3 fragmentos de ese datagrama. Los dos primeros fragmentos de 620 bytes, y el último de 420 bytes. A continuación analizaremos cómo obtener la longitud de cada fragmento.

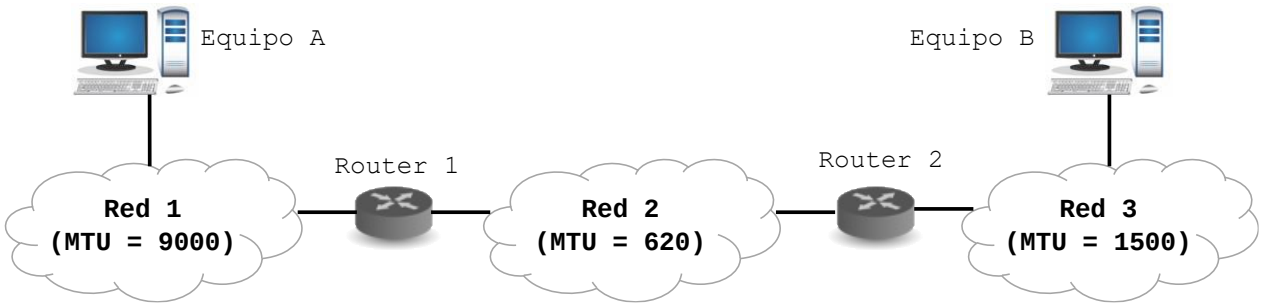


Figura 5. Ejemplo de fragmentación

Datagrama original	Longitud total	Identificación	DF	MF	Desplazamiento	Datos
	1620	32	0	0	0	1600 bytes

La fragmentación realizada en el Router 1 genera los siguientes datagramas:

	Longitud total	Identificación	DF	MF	Desplazamiento	Datos
Fragmento 1	**620**	32	0	**1**	0	**Del byte 0 al 599**
Fragmento 2	**620**	32	0	**1**	**75** (75×8 = 600)	**Del byte 600 al 1199**
Fragmento 3	**420**	32	0	0	**150** (150×8 = 1200)	**Del byte 1200 a 1599**

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. Así, en este caso tendríamos que 576 – 20 = 556 y 556 ÷ 8 = 69.5, es decir no es divisible entre 8. Entonces, en estos casos, buscaremos el múltiplo de 8 más cercano al tamaño máximo. En este ejemplo sólo se podrían aprovechar 552 (69 × 8) 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 (69 × 8 = 552).

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

Aunque no podemos observar directamente la fragmentación que se produce en los routers, 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 equipo se verá obligada a fragmentar el datagrama que contiene el paquete ICMP.

3.1 Formato de los mensajes ICMP echo request y reply

La orden ping utiliza los mensajes ICMP *Echo request* e ICMP *Echo reply*. Aunque el protocolo ICMP se estudiará en detalle en la práctica 4, destacamos ahora las ideas básicas para entender su uso en esta práctica. El formato de estos mensajes, que se encapsularán en la zona de datos del datagrama IP, se puede apreciar en la Figura 6. Como se puede ver, son unos mensajes sencillos compuestos por una cabecera y unos datos. En la cabecera se especifica el *Tipo de mensaje* ICMP (echo *request* o *reply*), el campo *Código* no tiene utilidad en estas órdenes y a continuación se añade un código (*checksum*) para detectar errores, similar a los que hemos estudiado en otros protocolos. En la siguiente palabra los campos *Identificador* y *Número de secuencia* sirven para poder hacer un seguimiento de los mensajes que se envían y reciben. Podemos pensar en el campo de identificador como un número que identifica la petición y el número de secuencia como el número que se incrementa cada vez que se hace una nueva petición a un destino. Así el emisor puede identificar claramente a qué mensaje de los que ha enviado le están contestando. Por último, en el campo de datos el emisor pone la información que quiere que le reenvíe el receptor a la solicitud de eco (de ahí su nombre). Normalmente se rellena con unas decenas de bytes (cada sistema operativo pone una cantidad por defecto, 32, 56 o 64 bytes en Windows, macOS y Linux respectivamente). En definitiva, tendremos una longitud total igual a 8 bytes (2 palabras de 32 bits que constituyen la cabecera) más la longitud de los datos que se envíen para solicitar el eco.

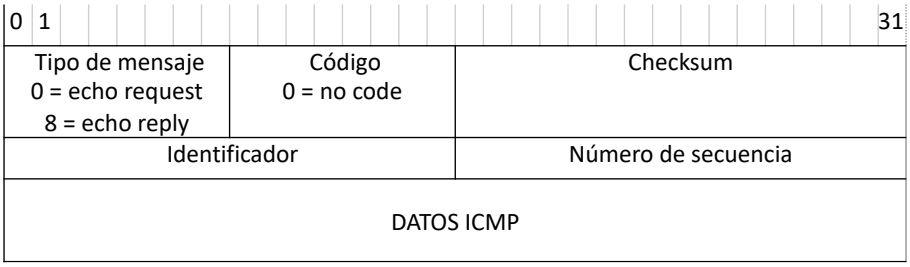


Figura 6. Formato de los mensajes ICMP echo request y reply

En la Figura 7 se aprecia la estructura completa de los diferentes niveles involucrados en la comunicación. En los datos de la trama ethernet (que posteriormente se pasará al nivel físico para que la envíe bit a bit) se pone el datagrama IP (el campo *tipo* en la trama ethernet igual a 0x800 indica que se encapsula en la trama un datagrama IP). El datagrama IP lleva en este caso un mensaje ICMP. Como se muestra en la figura, en el datagrama IP se indica en el campo *Protocolo* (= 1) que en la zona de datos se encapsula un mensaje ICMP. Si el protocolo fuera el 6 significaría que en la zona de datos estaría embebido un segmento TCP.

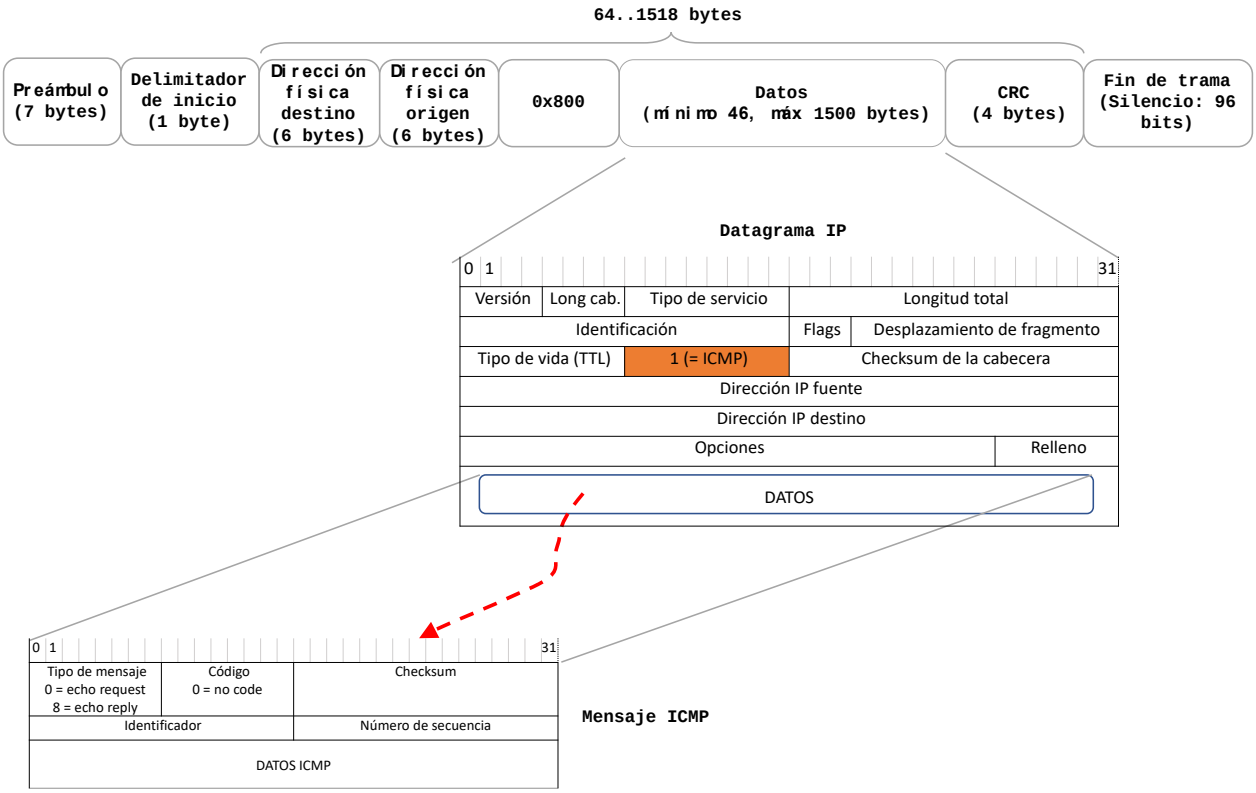


Figura 7. Mensaje ICMP - Datagrama IP - Trama Ethernet

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:

Captura→Opciones→Filtro de captura para interfaces seleccionados: icmp and host xx.xx.xx.xx

Debes sustituir ***xx.xx.xx.xx*** por la dirección IP de tu máquina. Recuerda que en la práctica 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 una terminal de Linux y teclea:

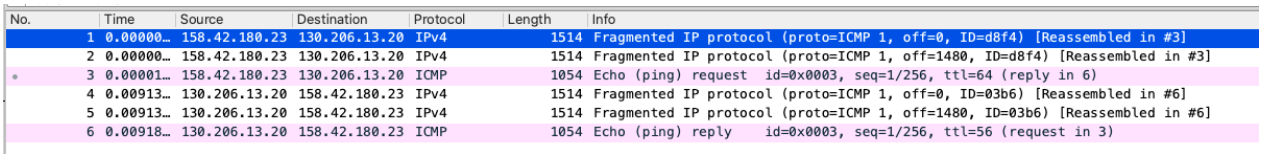
```
> ping -c 1 -s 3972 www.rediris.es
```

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.

Obtendrás una captura similar a la mostrada en la Figura 8.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000…	158.42.180.23	130.206.13.20	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=d8f4) [Reassembled in #3]
2	0.00000…	158.42.180.23	130.206.13.20	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=d8f4) [Reassembled in #3]
3	0.00001…	158.42.180.23	130.206.13.20	ICMP	1054	Echo (ping) request id=0x0003, seq=1/256, ttl=64 (reply in 6)
4	0.00913…	130.206.13.20	158.42.180.23	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=03b6) [Reassembled in #6]
5	0.00913…	130.206.13.20	158.42.180.23	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=1480, ID=03b6) [Reassembled in #6]
6	0.00918…	130.206.13.20	158.42.180.23	ICMP	1054	Echo (ping) reply id=0x0003, seq=1/256, ttl=56 (request in 3)

Figura 8. Captura de la orden ping con Wireshark

La opción -c 1 en Linux y macOS o -n 1 en Windows 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 por defecto envía paquetes de forma ininterrumpida. La opción -s 3972 (Linux y macOS) o -l 3972 (Windows) indica que el mensaje ICMP lleva un campo de datos de 3972 bytes, cuyo contenido no es relevante. Ten en cuenta que el paquete ICMP también tiene una cabecera de 8 bytes como hemos visto antes. A continuación, la orden ping muestra por pantalla información acerca de la respuesta, que será un mensaje ICMP de respuesta de eco.

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. No malinterpretes lo que estás viendo en Wireshark: hay 3 fragmentos tanto en la petición como en la respuesta. Wireshark etiqueta los dos primeros como "*Fragmented IP protocol*" y el tercero, el último, como la solicitud de "ping" que conformaban los 3 fragmentos. Es una "comodidad" que nos ofrece Wireshark para que interpretemos las tramas capturadas de una forma más sencilla. Analiza en detalle el contenido de los 3 fragmentos para contestar las siguientes preguntas:

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? Justifica la respuesta.

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.

e) 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.

f) ¿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?