Computación en Internet I

Andrés A. Aristizábal P.
aaaristizabal@icesi.edu.co

Departamento de Tecnologías de Información y Comunicaciones

2023-1

Agenda

1. **First Midterm**

2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP

3. **Exercises**

Question 1

Empareje cada capa del modelo OSI con su función

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio).

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión.

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima.

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega.

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.)

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.) **Sesión**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.) **Sesión**
- Actúa como una capa de “traducción” de datos, que les da formato, los comprime o los cifra, entre otras acciones.

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.) **Sesión**
- Actúa como una capa de “traducción” de datos, que les da formato, los comprime o los cifra, entre otras acciones. **Presentación**

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.) **Sesión**
- Actúa como una capa de “traducción” de datos, que les da formato, los comprime o los cifra, entre otras acciones. **Presentación**
- Es la capa que ve el usuario. Provee los servicios que ofrece la red de datos (correo electrónico, transferencia de archivos, conexión remota...).

Question 1

Empareje cada capa del modelo OSI con su función

- Especifica el medio de transmisión (cable, fibra, ondas electromagnéticas) y el formato de los bits (pulsos de luz, señales eléctricas o de radio). **Física**
- Especifica la manera cómo las diferentes estaciones se identifican entre sí, y cómo se turnan para poder hacer uso del medio de transmisión. **Enlace de datos**
- Especifica la manera cómo se hace el enrutamiento, es decir, cómo llevar un mensaje desde un punto origen hasta un punto destino siguiendo una ruta óptima. **Red**
- Especifica a cuál proceso dentro del destino se van a entregar los datos, y el nivel de confiabilidad de la entrega. **Transporte**
- Establece los parámetros para la creación de una sesión entre dos nodos (claves de cifrado, parámetros de la conexión, etc.) **Sesión**
- Actúa como una capa de “traducción” de datos, que les da formato, los comprime o los cifra, entre otras acciones. **Presentación**
- Es la capa que ve el usuario. Provee los servicios que ofrece la red de datos (correo electrónico, transferencia de archivos, conexión remota...). **Aplicación**

Question 2

Empareje el medio físico con su descripción

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45.

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45. **Cable UTP**

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45. **Cable UTP**
- Es un medio confiable para distancias medias, protegido contra la interferencia electromagnt́ica. Emplea conectores roscados.

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45. **Cable UTP**
- Es un medio confiable para distancias medias, protegido contra la interferencia electromagnt́ica. Emplea conectores roscados. **Cable coaxial**

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45. **Cable UTP**
- Es un medio confiable para distancias medias, protegido contra la interferencia electromagnt́ica. Emplea conectores roscados. **Cable coaxial**
- Es un medio confiable para distancias cortas, medias y largas. Se puede fabricar con plástico o vidrio.

Question 2

Empareje el medio físico con su descripción

- Es un medio barato y confiable para distancias menores o iguales a 100 metros. Emplea conectores RJ-45. **Cable UTP**
- Es un medio confiable para distancias medias, protegido contra la interferencia electromagnt́ica. Emplea conectores roscados. **Cable coaxial**
- Es un medio confiable para distancias cortas, medias y largas. Se puede fabricar con plástico o vidrio. **Fibra óptica**

Question 3

En un cable coaxial, la señal de datos circula por

Question 3

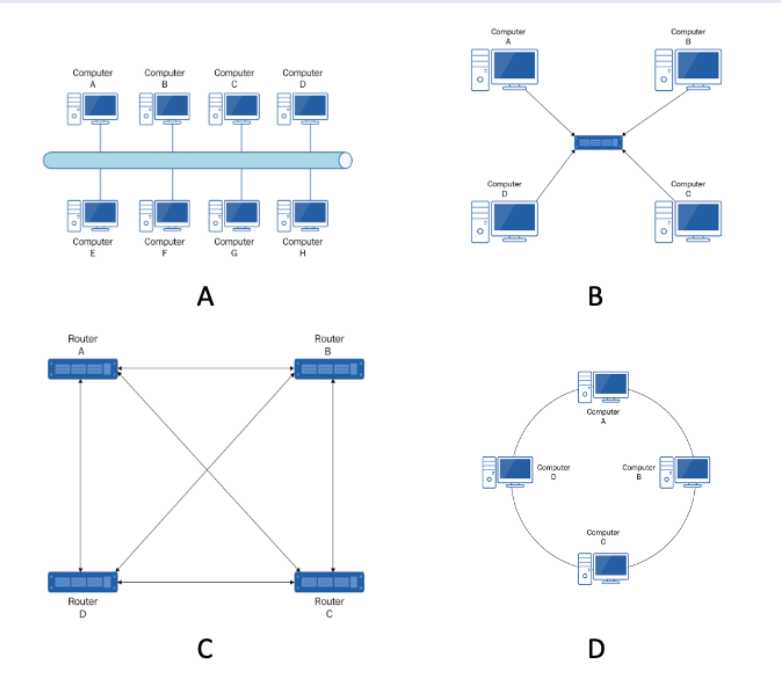
En un cable coaxial, la señal de datos circula por **el conductor central**, y

Question 3

En un cable coaxial, la señal de datos circula por **el conductor central**, y **la malla** actúa como línea de retorno o tierra.

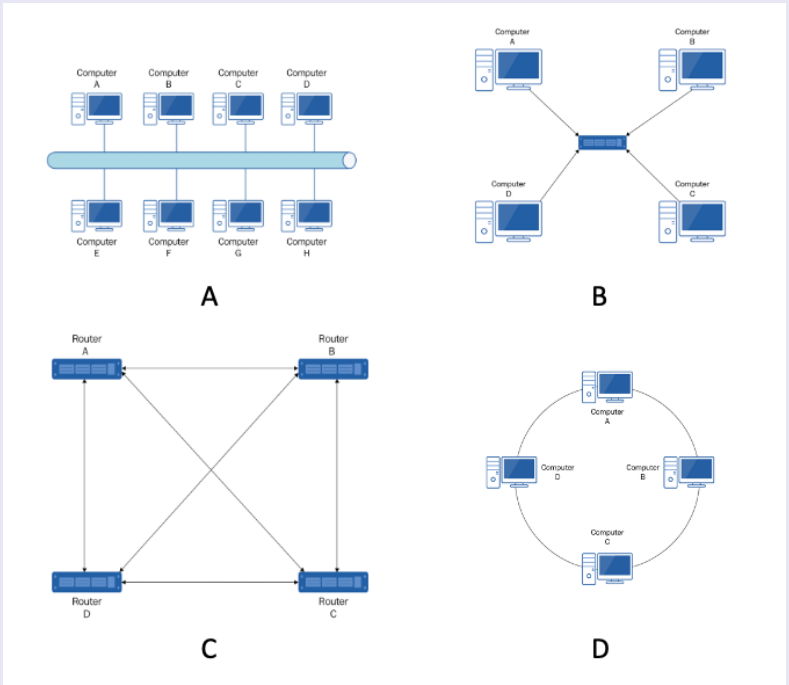
Question 4

Arrastre el nombre de la topología debajo del gráfico correspondiente.



Question 4

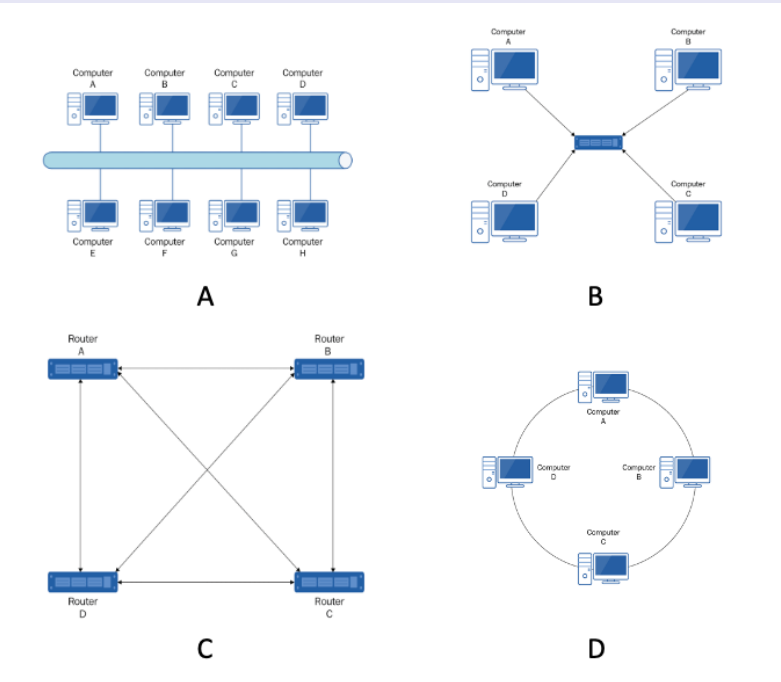
Arrastre el nombre de la topología debajo del gráfico correspondiente.



A Bus

Question 4

Arrastre el nombre de la topología debajo del gráfico correspondiente.

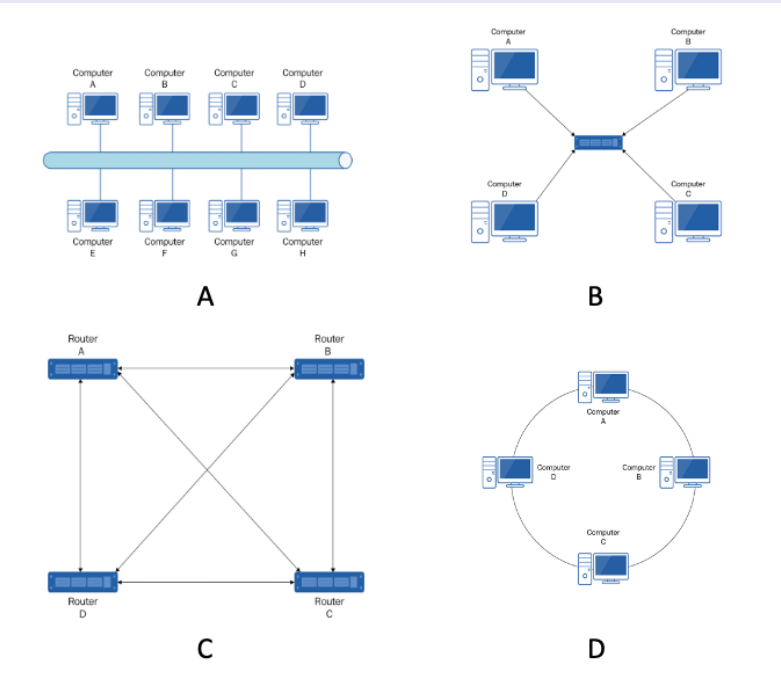


A Bus

B Estrella

Question 4

Arrastre el nombre de la topología debajo del gráfico correspondiente.



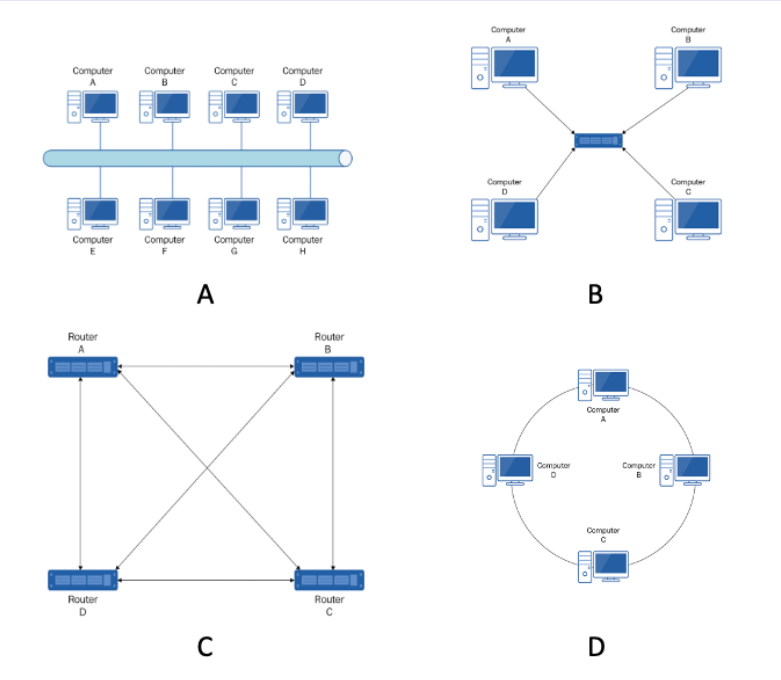
A Bus

B Estrella

C Malla

Question 4

Arrastre el nombre de la topología debajo del gráfico correspondiente.



A Bus

B Estrella

C Malla

D Anillo

Question 5

Considere la subred 192.168.131.0/28

Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de

Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de **14** hosts.

Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de **14** hosts.
- La dirección de la subred es

Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de **14** hosts.
- La dirección de la subred es **192.168.131.0**

Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de **14** hosts.
- La dirección de la subred es **192.168.131.0**
- La dirección de broadcast de la subred es

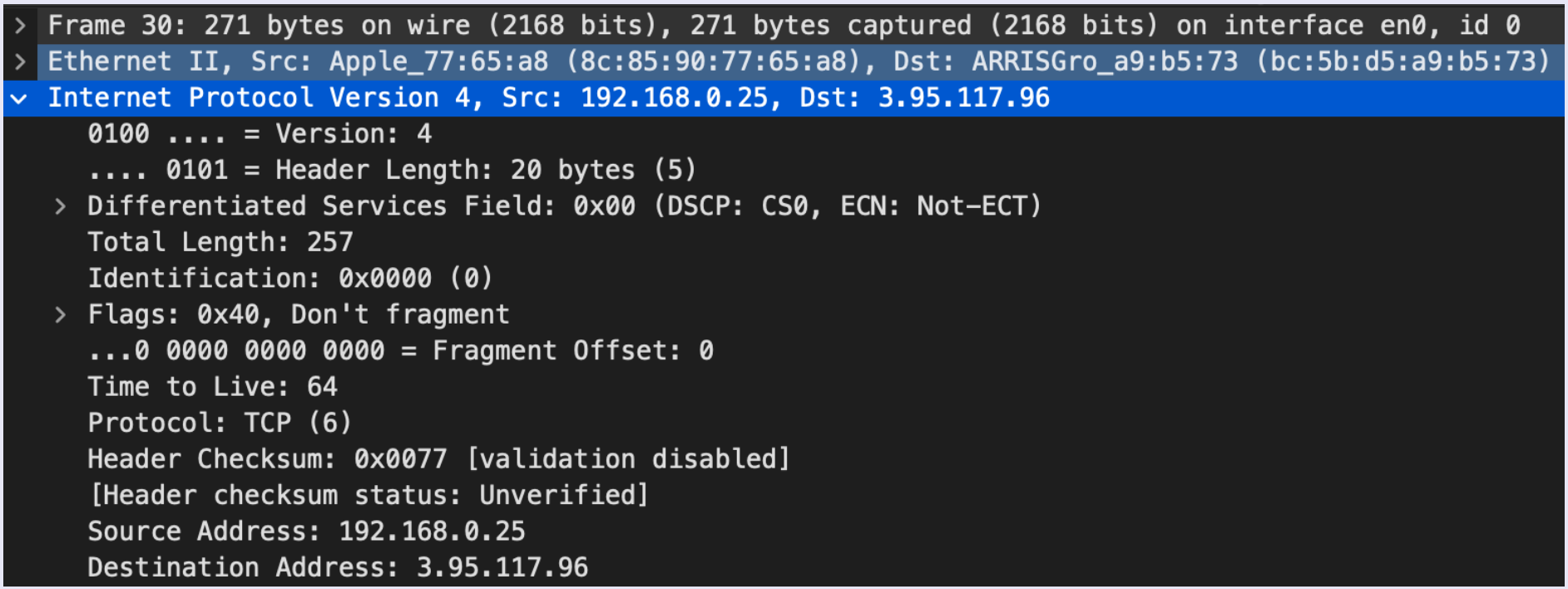
Question 5

Considere la subred 192.168.131.0/28

- La subred tiene un total de **14** hosts.
- La dirección de la subred es **192.168.131.0**
- La dirección de broadcast de la subred es **192.168.131.15**

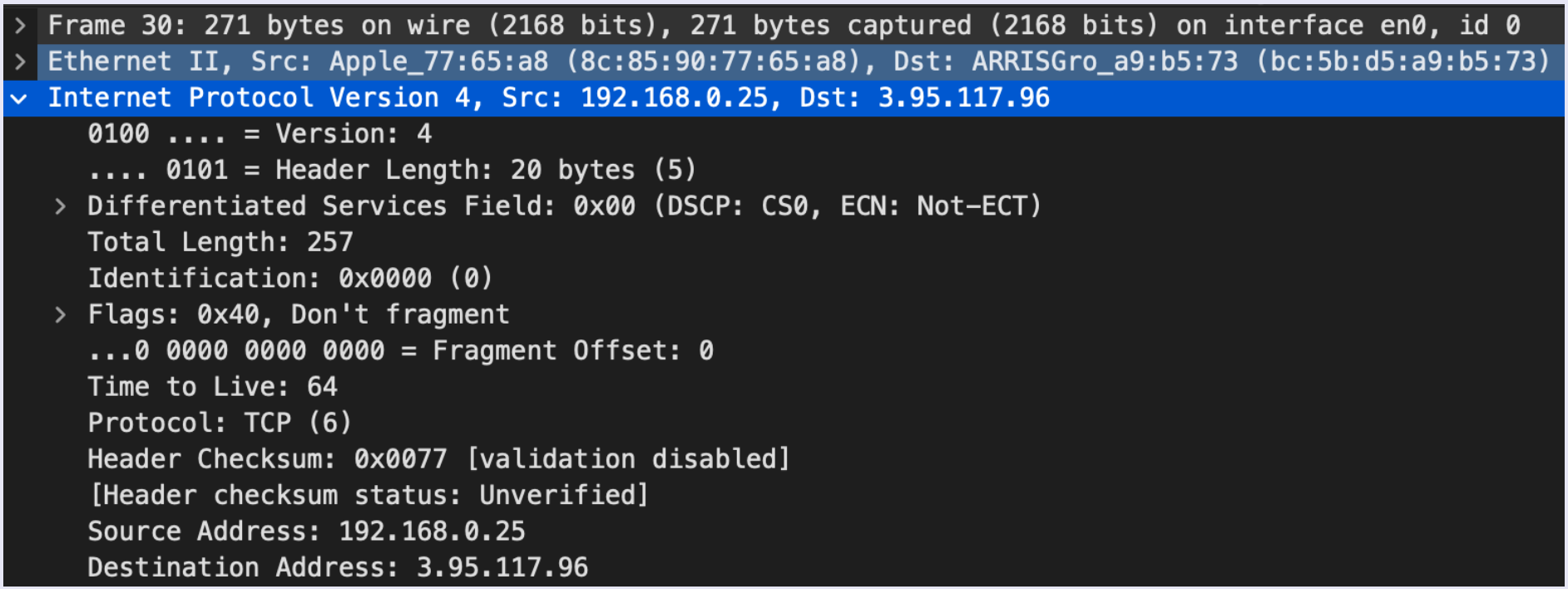
Question 6

En la siguiente captura:



Question 6

En la siguiente captura:



La dirección IP fuente del datagrama es

Question 6

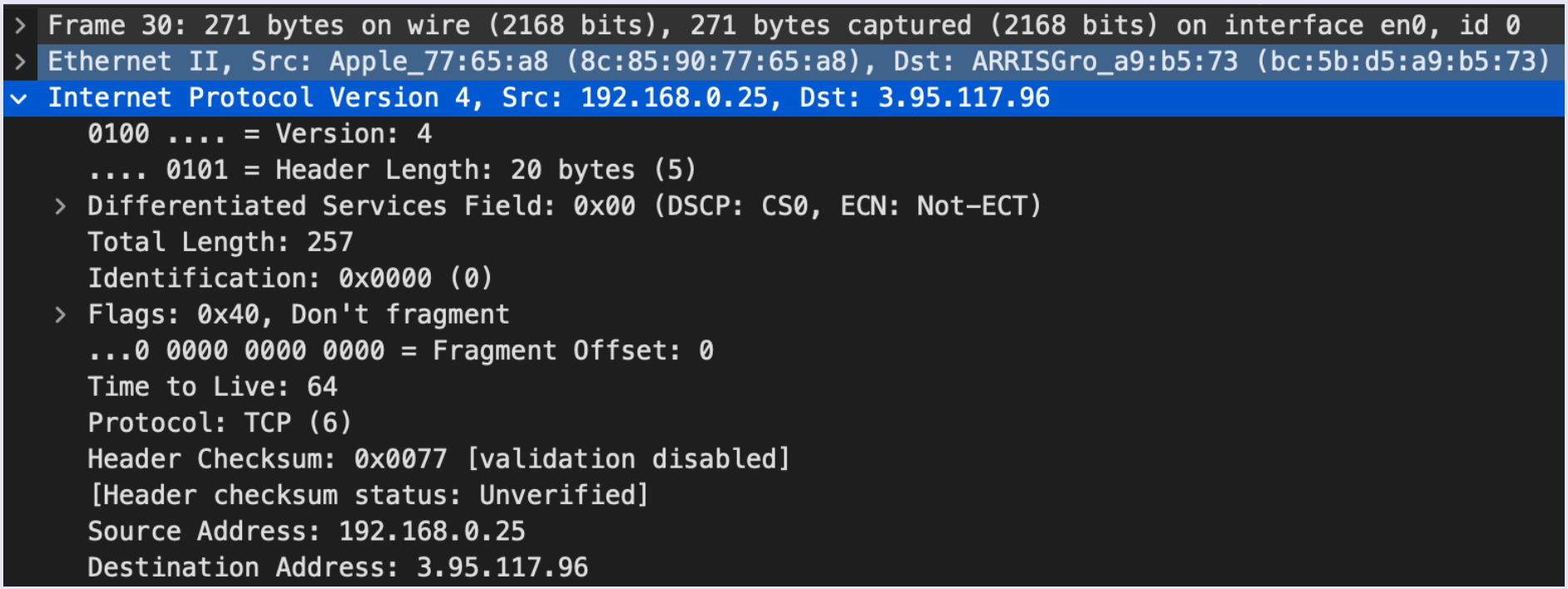
En la siguiente captura:

```
> Frame 30: 271 bytes on wire (2168 bits), 271 bytes captured (2168 bits) on interface en0, id 0
> Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
v Internet Protocol Version 4, Src: 192.168.0.25, Dst: 3.95.117.96
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 257
    Identification: 0x0000 (0)
  > Flags: 0x40, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0x0077 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0.25
    Destination Address: 3.95.117.96
```

La dirección IP fuente del datagrama es **192.168.0.25**,

Question 6

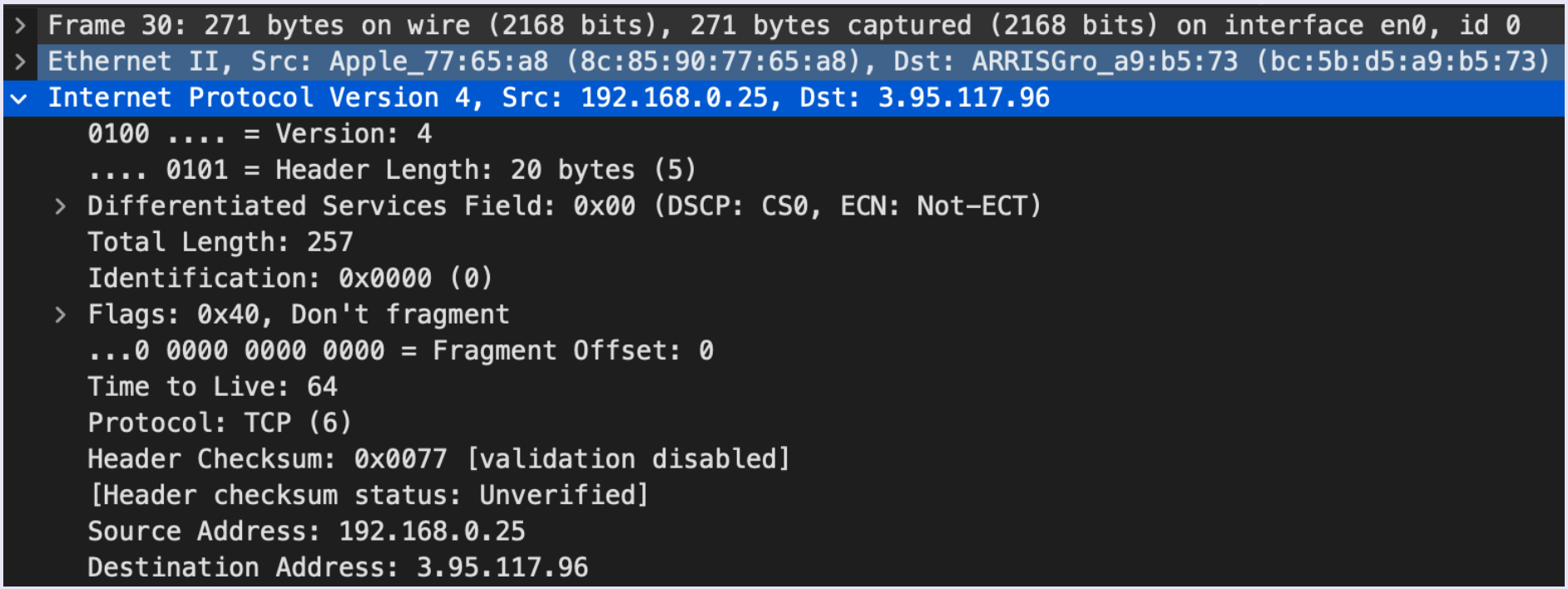
En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es

Question 6

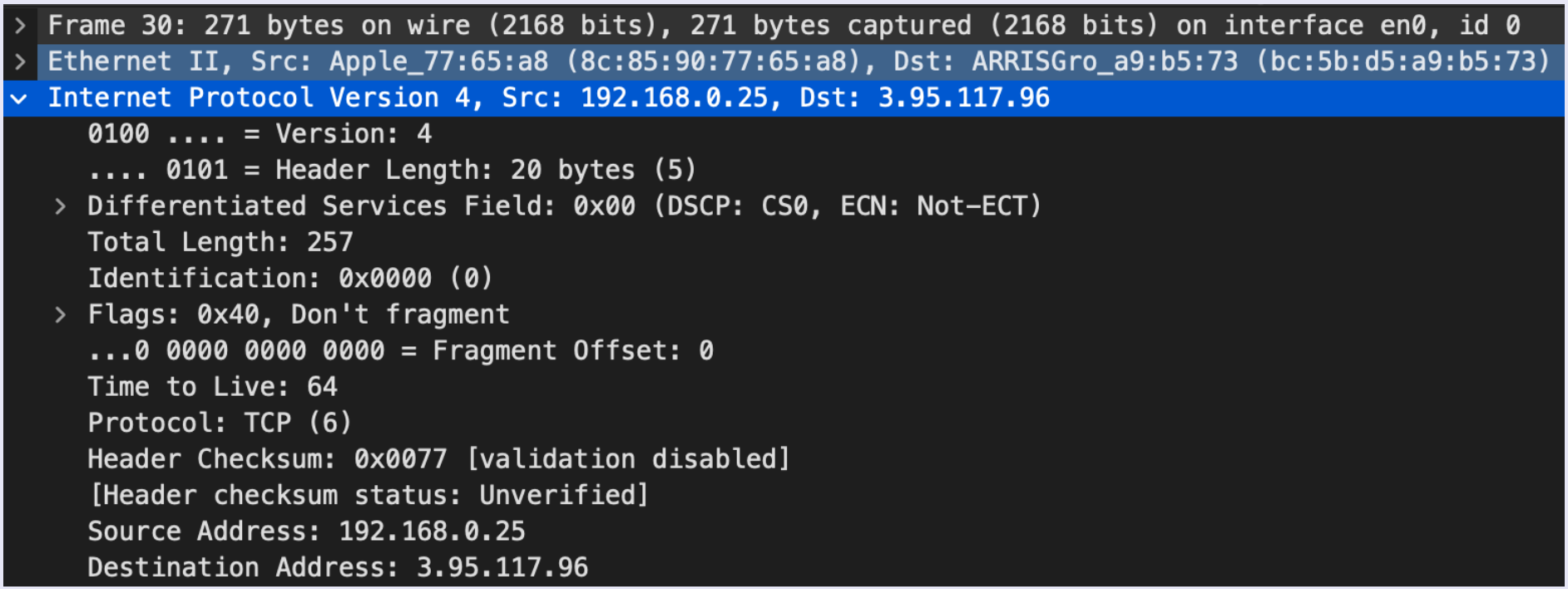
En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es **3.95.117.96**,

Question 6

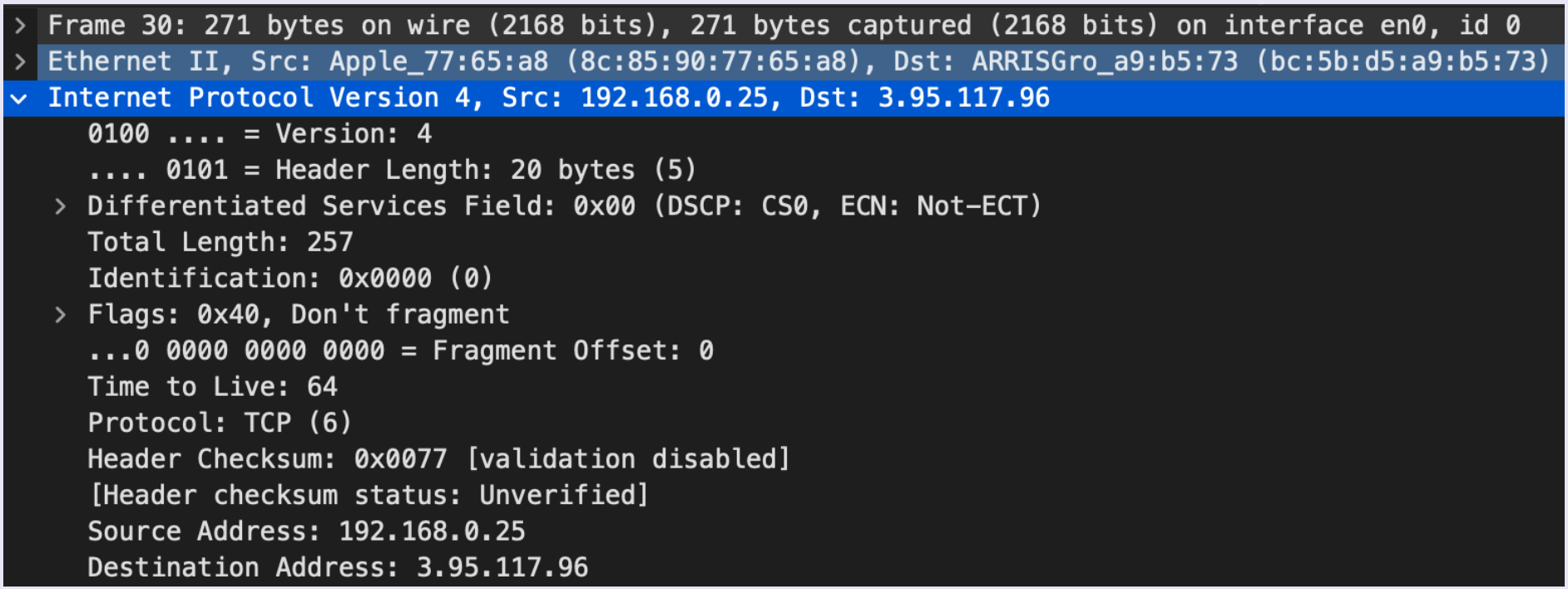
En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es **3.95.117.96**, el payload corresponde al protocolo

Question 6

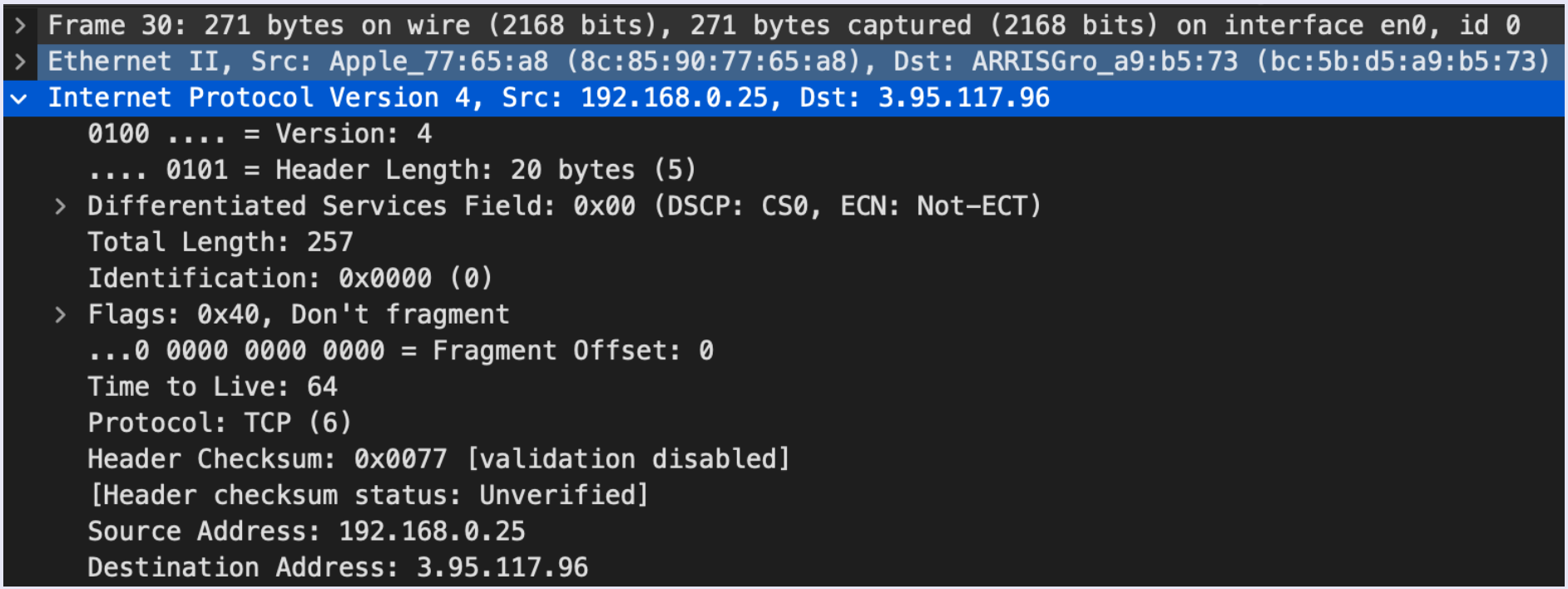
En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es **3.95.117.96**, el payload corresponde al protocolo **tcp**

Question 6

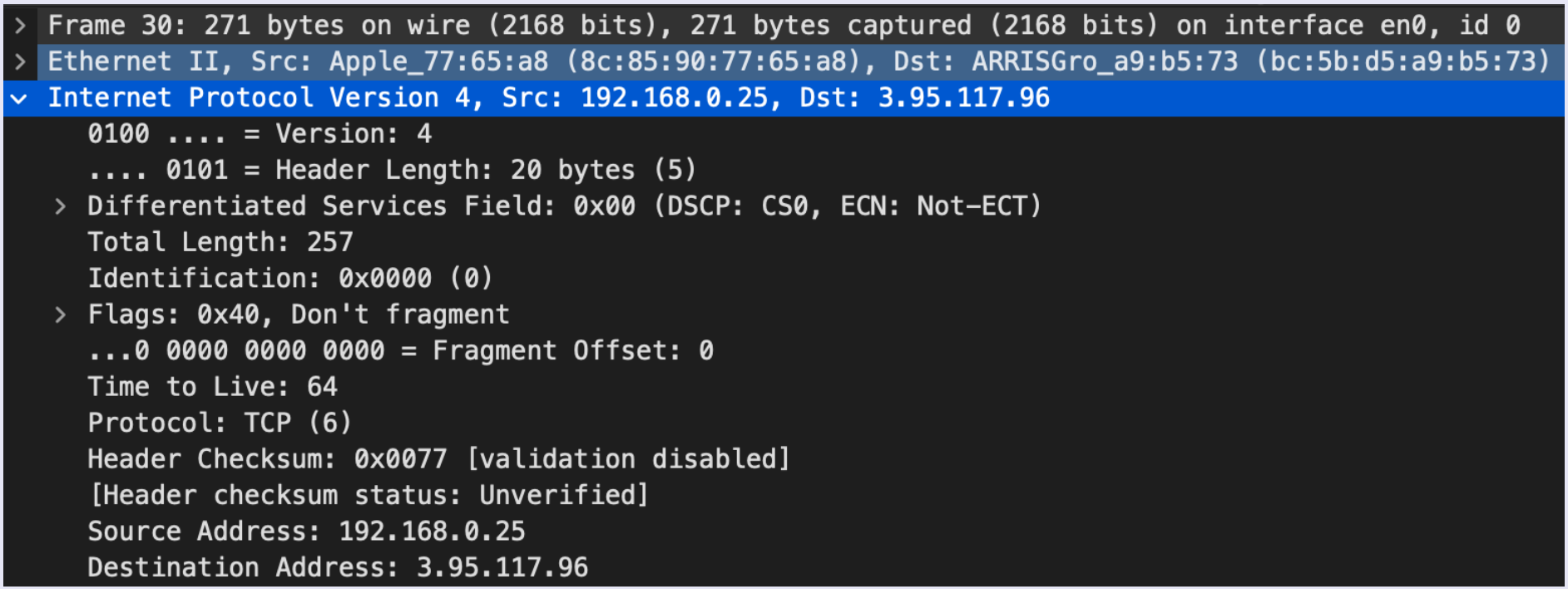
En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es **3.95.117.96**, el payload corresponde al protocolo **tcp** y la longitud de dicho payload es de

Question 6

En la siguiente captura:



La dirección IP fuente del datagrama es **192.168.0.25**, la dirección IP destino es **3.95.117.96**, el payload corresponde al protocolo **tcp** y la longitud de dicho payload es de **237** bytes.

Question 7

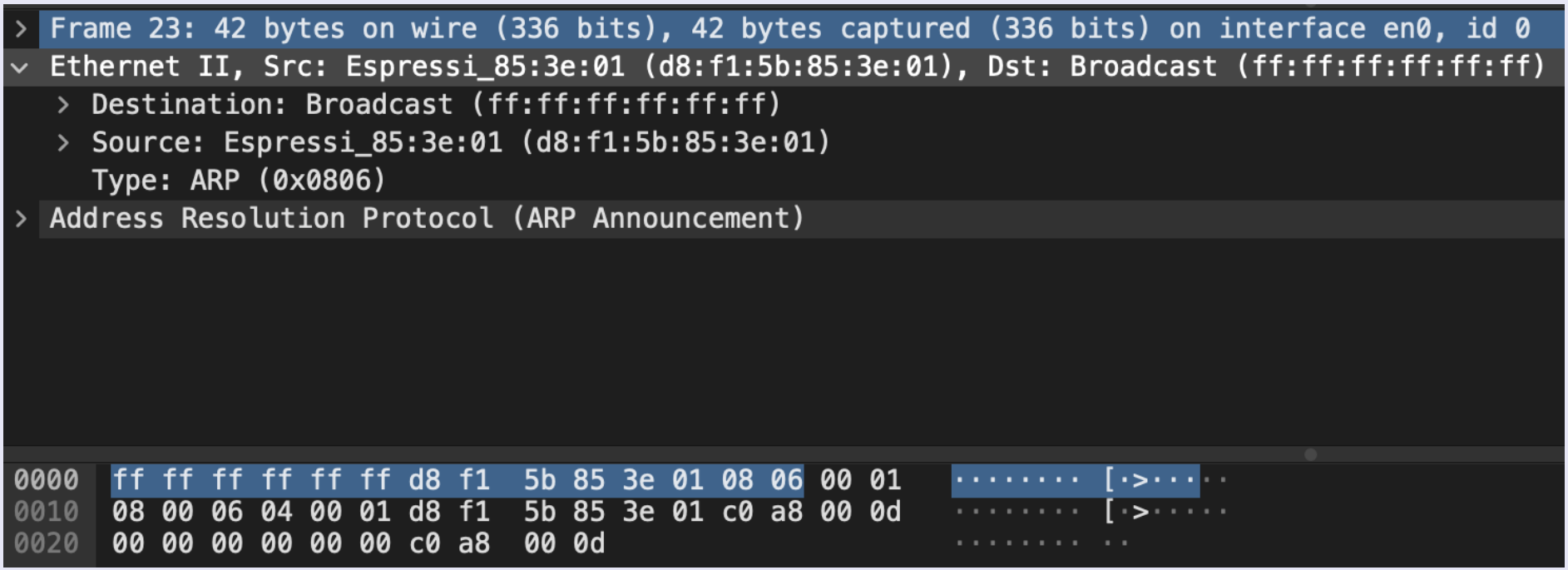
En la siguiente captura:

```
> Frame 23: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
v Ethernet II, Src: Espressi_85:3e:01 (d8:f1:5b:85:3e:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    > Source: Espressi_85:3e:01 (d8:f1:5b:85:3e:01)
      Type: ARP (0x0806)
> Address Resolution Protocol (ARP Announcement)
```

```
0000  ff ff ff ff ff ff d8 f1  5b 85 3e 01 08 06 00 01   ........ [.>.....
0010  08 00 06 04 00 01 d8 f1  5b 85 3e 01 c0 a8 00 0d   ........ [.>.....
0020  00 00 00 00 00 00 c0 a8  00 0d                     ........ ..
```

Question 7

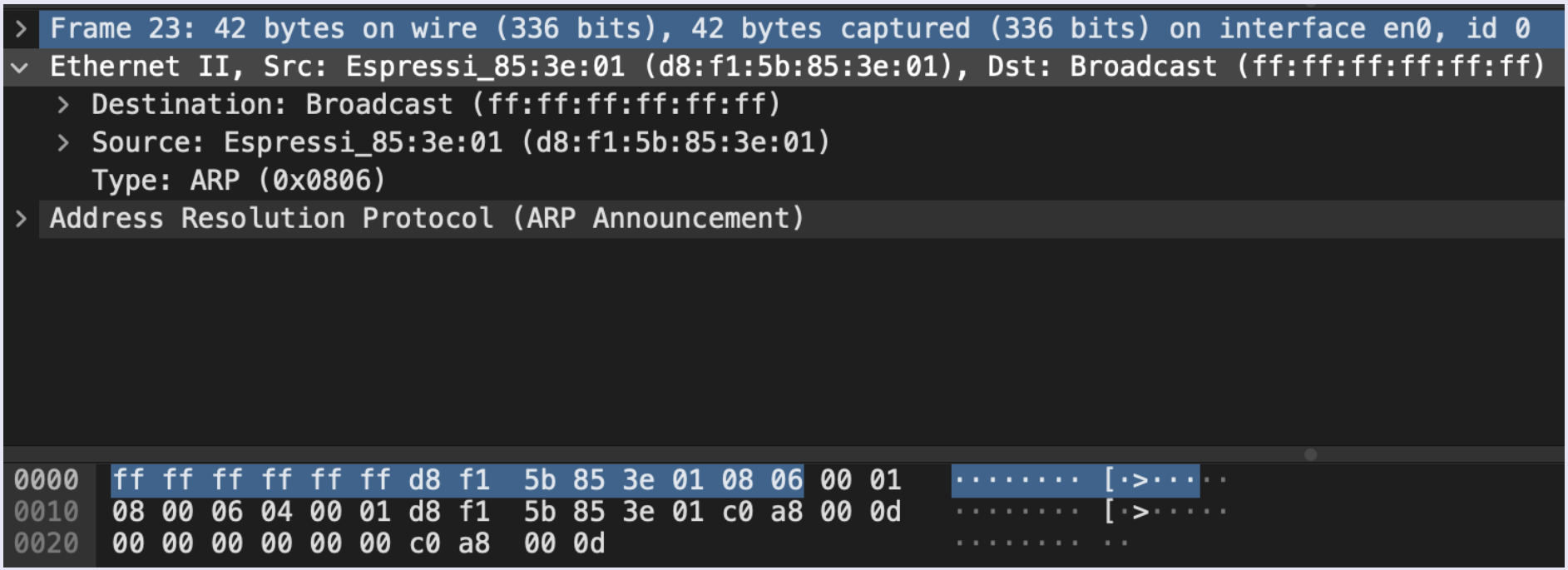
En la siguiente captura:



El encabezado de capa 2 tiene una longitud de

Question 7

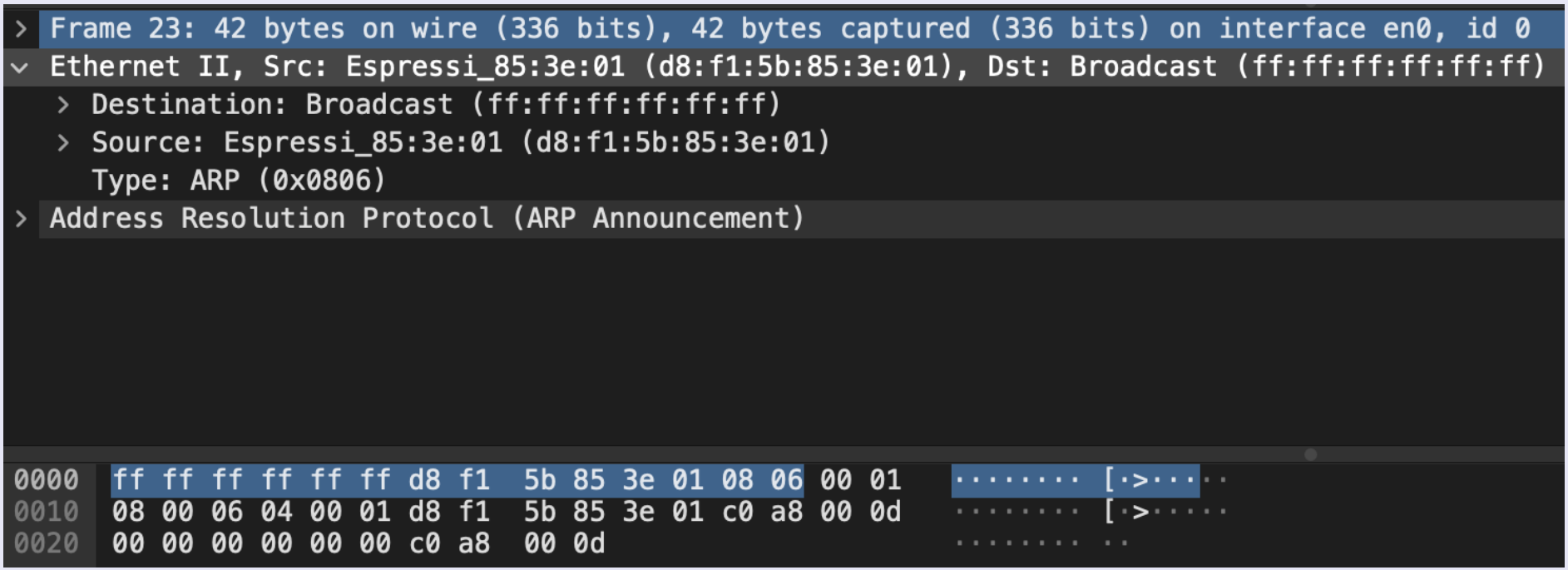
En la siguiente captura:



El encabezado de capa 2 tiene una longitud de **14** bytes,

Question 7

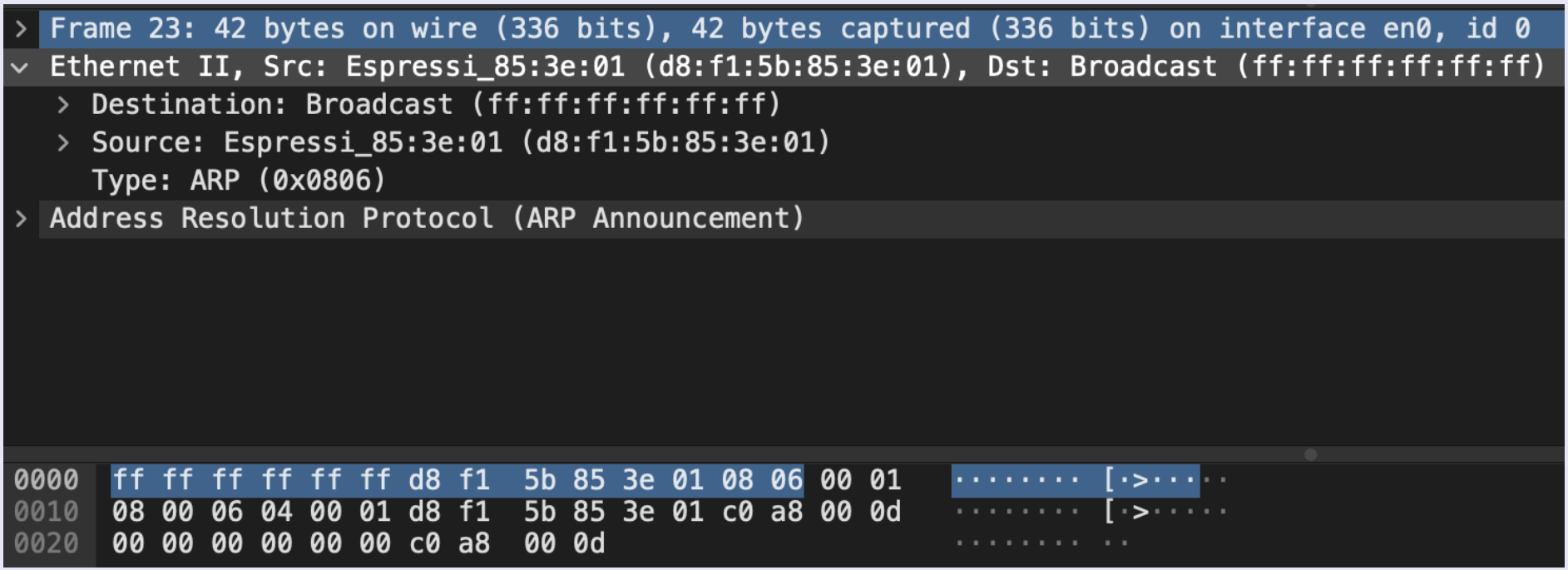
En la siguiente captura:



El encabezado de capa 2 tiene una longitud de **14** bytes, el equipo que transmitió la trama tiene la dirección

Question 7

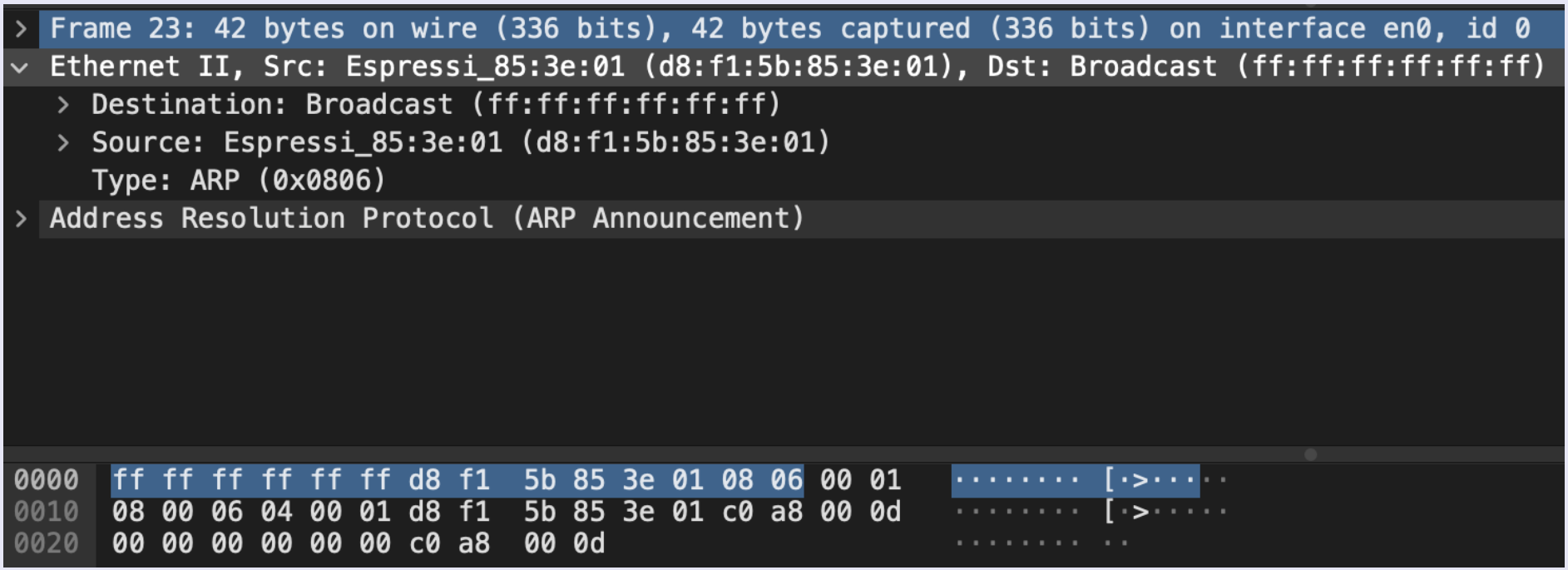
En la siguiente captura:



El encabezado de capa 2 tiene una longitud de **14** bytes, el equipo que transmitió la trama tiene la dirección **d8:f1:5b:85:3e:01**,

Question 7

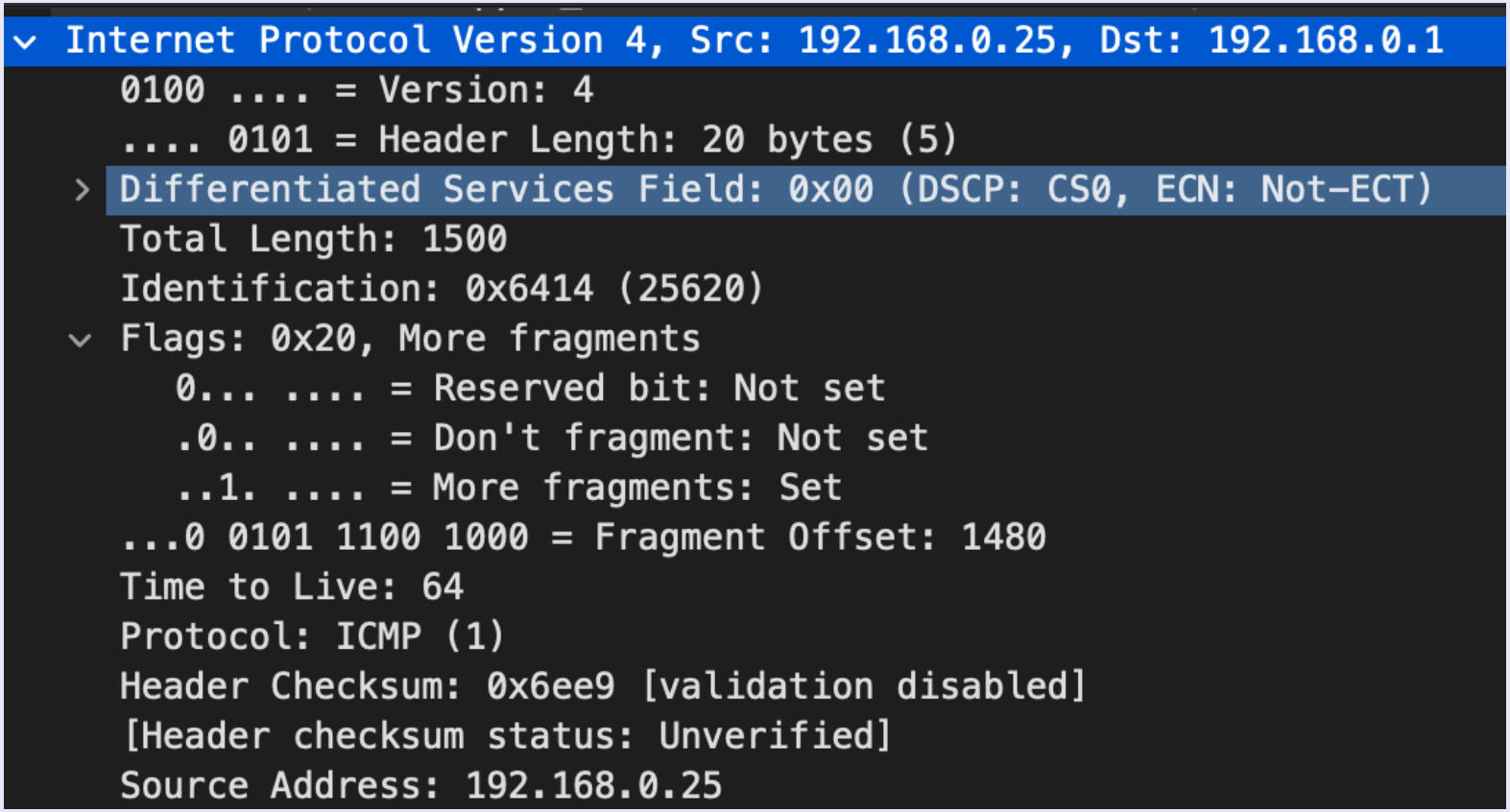
En la siguiente captura:



El encabezado de capa 2 tiene una longitud de **14** bytes, el equipo que transmitió la trama tiene la dirección **d8:f1:5b:85:3e:01**, y el payload de la trama contiene un mensaje **ARP**.

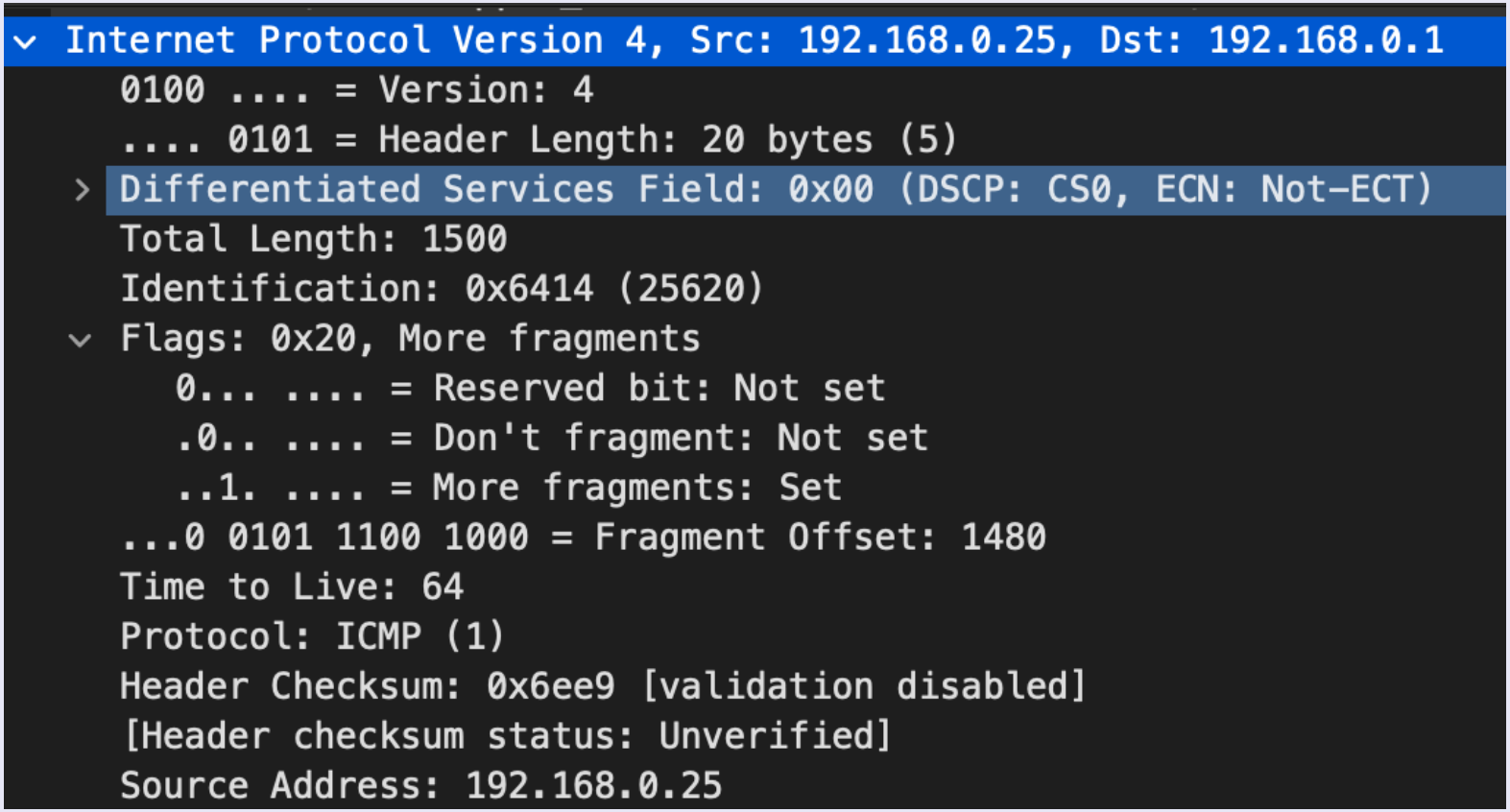
Question 8

En la siguiente captura:



Question 8

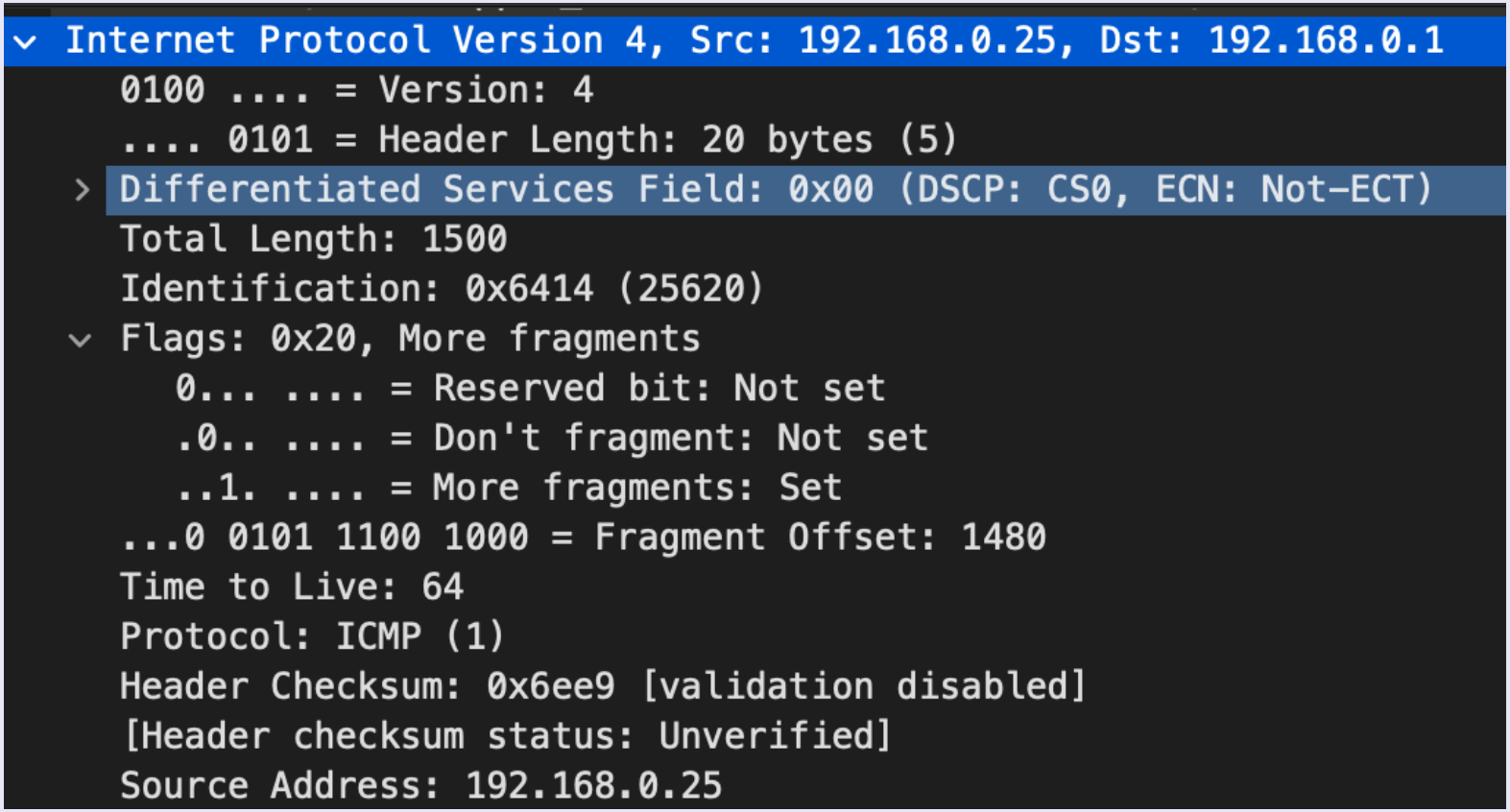
En la siguiente captura:



¿Esta captura hace parte de un datagrama fragmentado?

Question 8

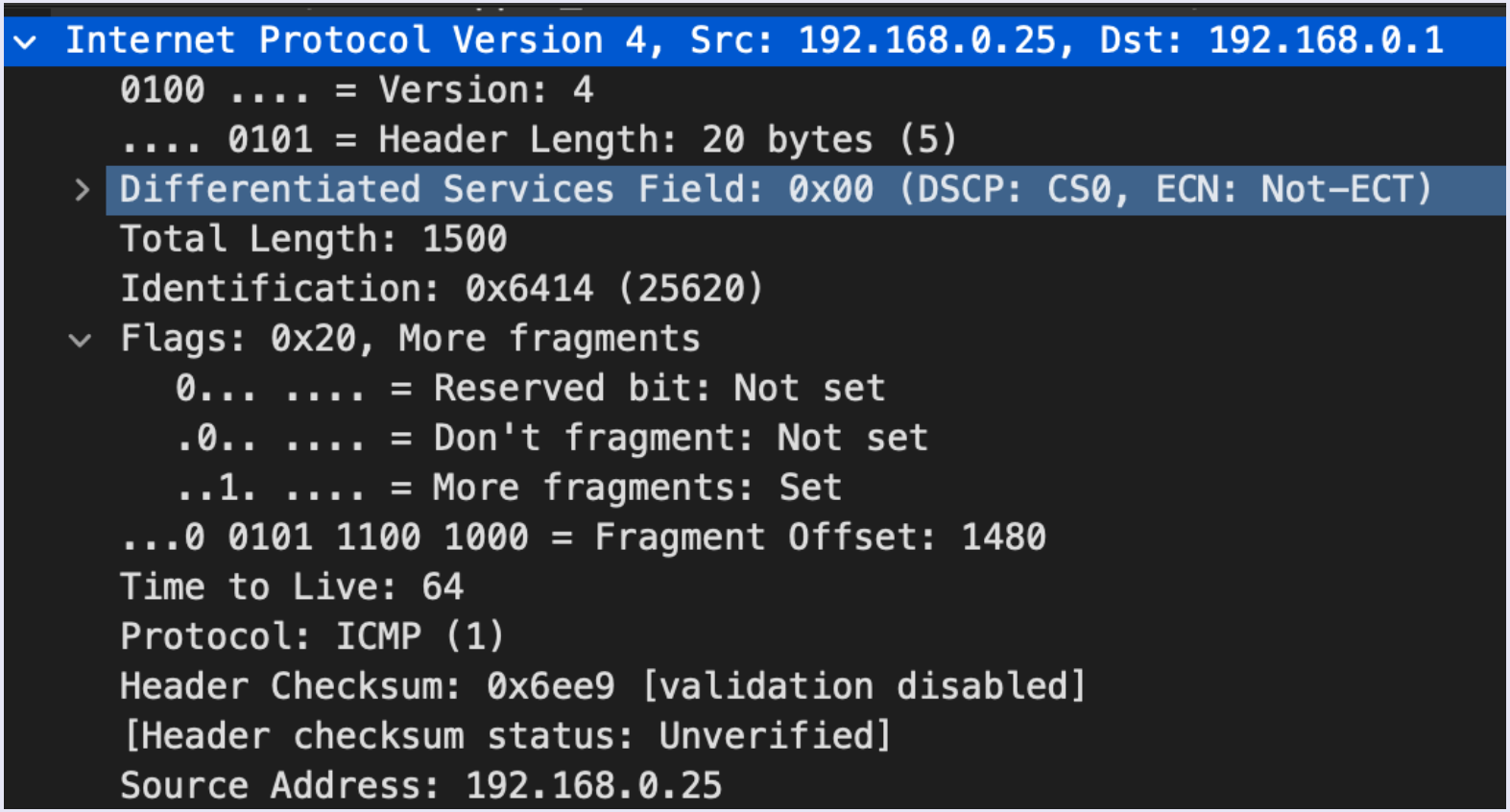
En la siguiente captura:



¿Esta captura hace parte de un datagrama fragmentado? **Sí**.

Question 8

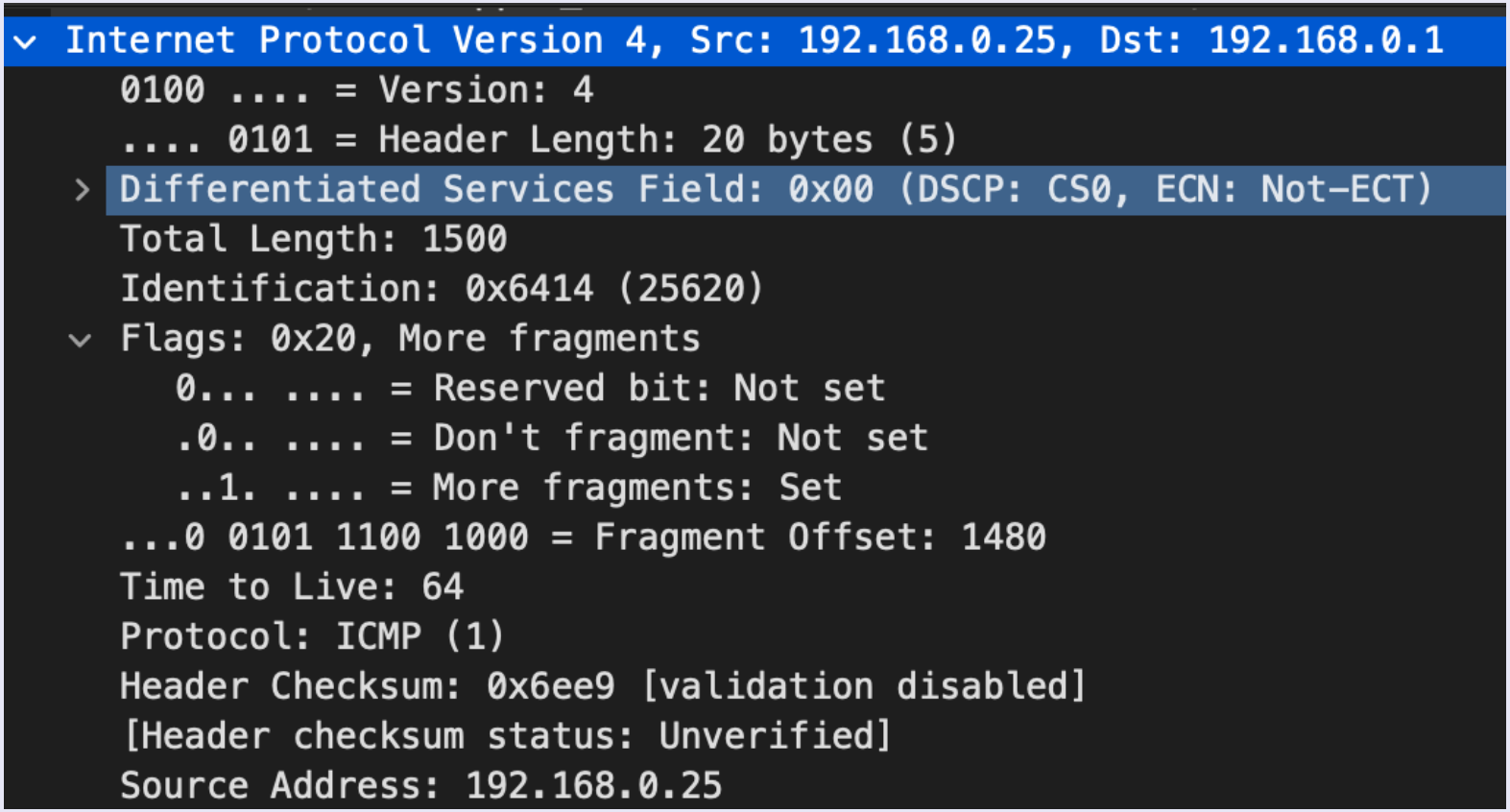
En la siguiente captura:



¿Esta captura hace parte de un datagrama fragmentado? **Sí**.
En caso de serlo, qué posición ocupa el fragmento en la secuencia?

Question 8

En la siguiente captura:



¿Esta captura hace parte de un datagrama fragmentado? **Sí**.
En caso de serlo, qué posición ocupa el fragmento en la secuencia?
Ni la primera ni la última.

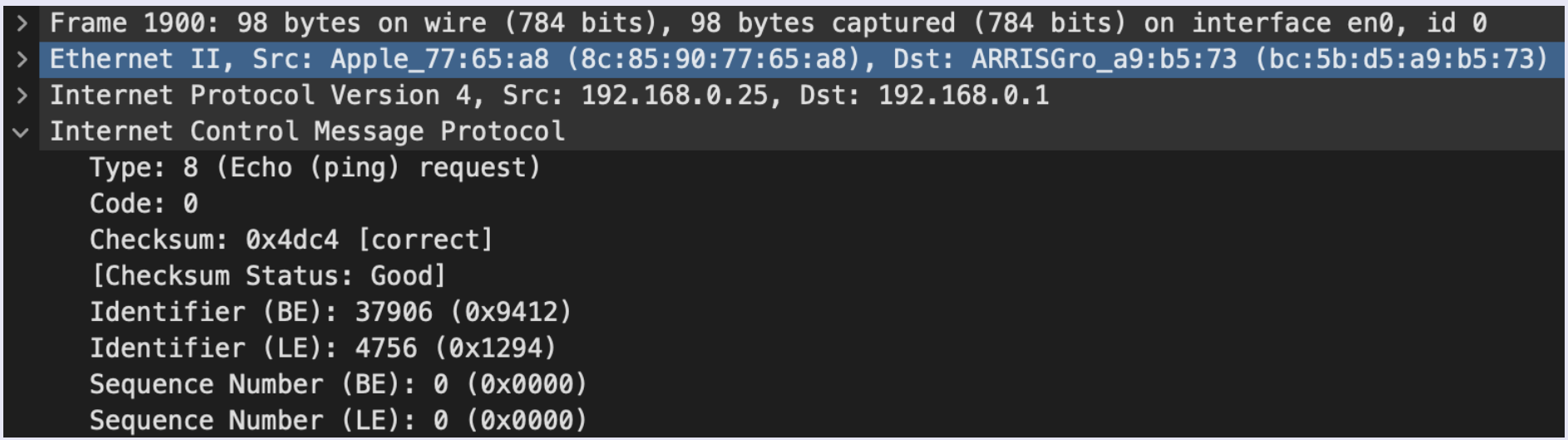
Question 9

En la siguiente captura:

```
> Frame 1900: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 192.168.0.1
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4dc4 [correct]
    [Checksum Status: Good]
    Identifier (BE): 37906 (0x9412)
    Identifier (LE): 4756 (0x1294)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
```

Question 9

En la siguiente captura:



Se está efectuando una

Question 9

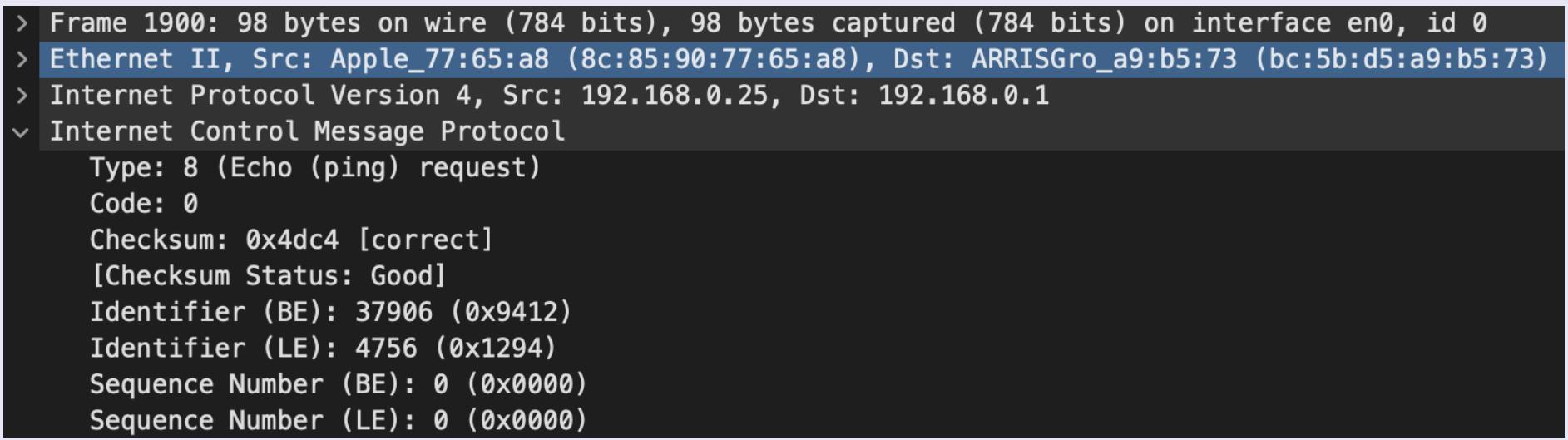
En la siguiente captura:

```
> Frame 1900: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 192.168.0.1
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4dc4 [correct]
    [Checksum Status: Good]
    Identifier (BE): 37906 (0x9412)
    Identifier (LE): 4756 (0x1294)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
```

Se está efectuando una **solicitud de eco**

Question 9

En la siguiente captura:



Se está efectuando una **solicitud de eco** al equipo con la dirección IP

Question 9

En la siguiente captura:

```
> Frame 1900: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface en0, id 0
> Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 192.168.0.1
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4dc4 [correct]
    [Checksum Status: Good]
    Identifier (BE): 37906 (0x9412)
    Identifier (LE): 4756 (0x1294)
    Sequence Number (BE): 0 (0x0000)
    Sequence Number (LE): 0 (0x0000)
```

Se está efectuando una **solicitud de eco** al equipo con la dirección IP **192.168.0.1**

Question 10

Una dirección MAC consta de

Question 10

Una dirección MAC consta de **48** bits,

Question 10

Una dirección MAC consta de **48** bits, de los cuales

Question 10

Una dirección MAC consta de **48** bits, de los cuales **24** son asignados por la IEEE al fabricante de la tarjeta, y los otros

Question 10

Una dirección MAC consta de **48** bits, de los cuales **24** son asignados por la IEEE al fabricante de la tarjeta, y los otros **24** son asignados por el fabricante.

Question 11

En la siguiente captura de red:

```
> Frame 29: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface en0, id 0
v Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Destination: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Source: Apple_77:65:a8 (8c:85:90:77:65:a8)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 3.95.117.96
> Transmission Control Protocol, Src Port: 50650, Dst Port: 443, Seq: 1033, Ack: 239, Len: 35
> Transport Layer Security
```

Question 11

En la siguiente captura de red:

```
> Frame 29: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface en0, id 0
v Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Destination: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Source: Apple_77:65:a8 (8c:85:90:77:65:a8)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 3.95.117.96
> Transmission Control Protocol, Src Port: 50650, Dst Port: 443, Seq: 1033, Ack: 239, Len: 35
> Transport Layer Security
```

La dirección MAC fuente es

Question 11

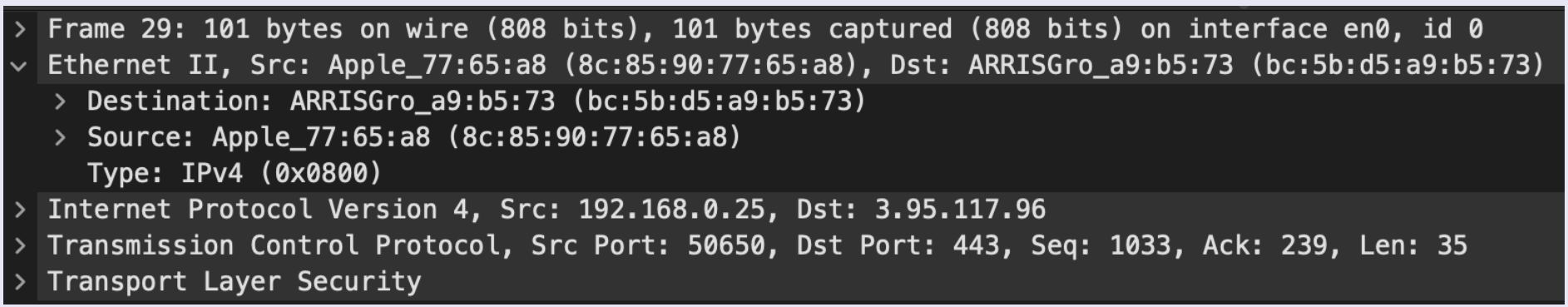
En la siguiente captura de red:

```
> Frame 29: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface en0, id 0
v Ethernet II, Src: Apple_77:65:a8 (8c:85:90:77:65:a8), Dst: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Destination: ARRISGro_a9:b5:73 (bc:5b:d5:a9:b5:73)
  > Source: Apple_77:65:a8 (8c:85:90:77:65:a8)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.0.25, Dst: 3.95.117.96
> Transmission Control Protocol, Src Port: 50650, Dst Port: 443, Seq: 1033, Ack: 239, Len: 35
> Transport Layer Security
```

La dirección MAC fuente es **8c:85:90:77:65:a8**

Question 11

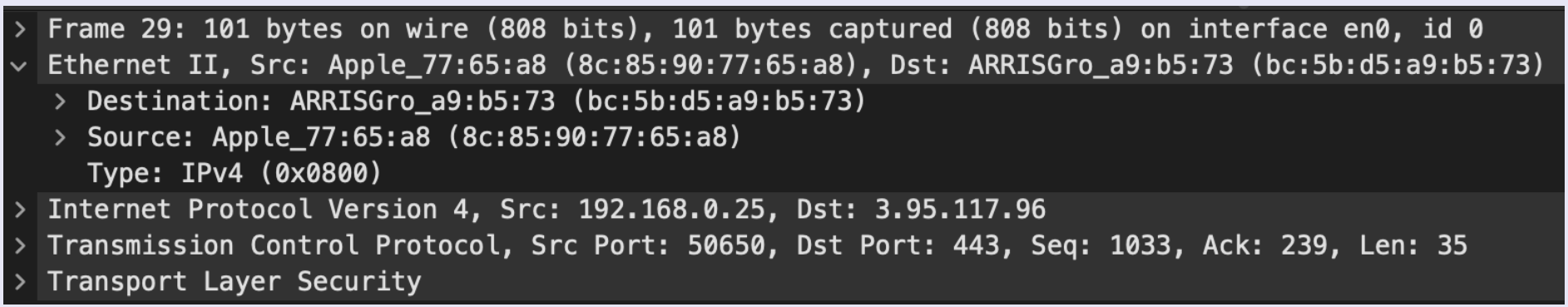
En la siguiente captura de red:



La dirección MAC fuente es **8c:85:90:77:65:a8** y la dirección MAC destino es

Question 11

En la siguiente captura de red:



La dirección MAC fuente es **8c:85:90:77:65:a8** y la dirección MAC destino es **bc:5b:d5:a9:b5:73**

Question 12

El protocolo IP en la capa de red emplea un mecanismo de confiabilidad en la entrega de datos conocido como

Question 12

El protocolo IP en la capa de red emplea un mecanismo de confiabilidad en la entrega de datos conocido como **mejor esfuerzo**.

Question 13

En las capturas de tramas con Wireshark, hay un campo que normalmente no se captura ni se muestra. Dicho campo es el siguiente:

Question 13

En las capturas de tramas con Wireshark, hay un campo que normalmente no se captura ni se muestra. Dicho campo es el siguiente: **Secuencia de chequeo de trama (FCS)**.

Question 14

El campo de TTL (Time-to-live) del encabezado IP muestra:

Question 14

El campo de TTL (Time-to-live) del encabezado IP muestra: **Cuántos saltos puede dar el datagrama entre enrutadores antes de que se lo considere como perdido**.

Question 15

La técnica de NAT se emplea para poder enrutar una subred privada a través de una dirección pública. El enrutador que hace NAT se encarga de hacer la traducción o equivalencia entre la dirección pública y las privadas.

Question 15

La técnica de NAT se emplea para poder enrutar una subred privada a través de una dirección pública. El enrutador que hace NAT se encarga de hacer la traducción o equivalencia entre la dirección pública y las privadas. **Verdadero**.

Agenda del día

1. First Midterm

2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP

3. Exercises

What is the Transport Layer?

What is the Transport Layer?

- It is a central piece of the layered network architecture.
- Resides between the application and network layers.
- It has the critical role of providing communication services directly to the application processes running on different hosts.

What is the Transport Layer?

- It is a central piece of the layered network architecture.
- Resides between the application and network layers.
- It has the critical role of providing communication services directly to the application processes running on different hosts.

What does a transport layer protocol provides?

What is the Transport Layer?

- It is a central piece of the layered network architecture.
- Resides between the application and network layers.
- It has the critical role of providing communication services directly to the application processes running on different hosts.

What does a transport layer protocol provides?

- A logical communication between application processes running on different hosts.
  - As if the hosts running the processes were directly connected.
  - In reality, the hosts may be on opposite sides of the planet, connected via numerous routers and a wide range of link types.
  - Application processes use the logical communication provided by the transport layer to send messages to each other.
  - Free from the worry of the details of the physical infrastructure used to carry these messages.

Where are the transport layer protocols implemented?

Where are the transport layer protocols implemented?

- They are implemented in the end systems.
- But not in network routers.

Where are the transport layer protocols implemented?

- They are implemented in the end systems.
- But not in network routers.

How do they work on the sending side?

Where are the transport layer protocols implemented?

- They are implemented in the end systems.
- But not in network routers.

How do they work on the sending side?

- The transport layer converts the application-layer messages into transport-layer packets (transport-layer segments).
  - Application messages are broken into smaller chunks.
  - A transport-layer header is added to each chunk to create the transport-layer segment.
- The transport layer then passes the segment to the network layer at the sending end system.
- The segment is encapsulated within a network-layer packet (a datagram) and sent to the destination.
- Network routers act only on the network-layer fields of the datagram.
  - They do not examine the fields of the transport-layer segment encapsulated with the datagram.

How do they work on the receiving side?

How do they work on the receiving side?

- The network layer extracts the transport-layer segment from the datagram and passes the segment up to the transport layer.
- The transport layer then processes the received segment, making the data in the segment available to the receiving application.

How do they work on the receiving side?

- The network layer extracts the transport-layer segment from the datagram and passes the segment up to the transport layer.
- The transport layer then processes the received segment, making the data in the segment available to the receiving application.

What about these transport-layer protocols?

How do they work on the receiving side?

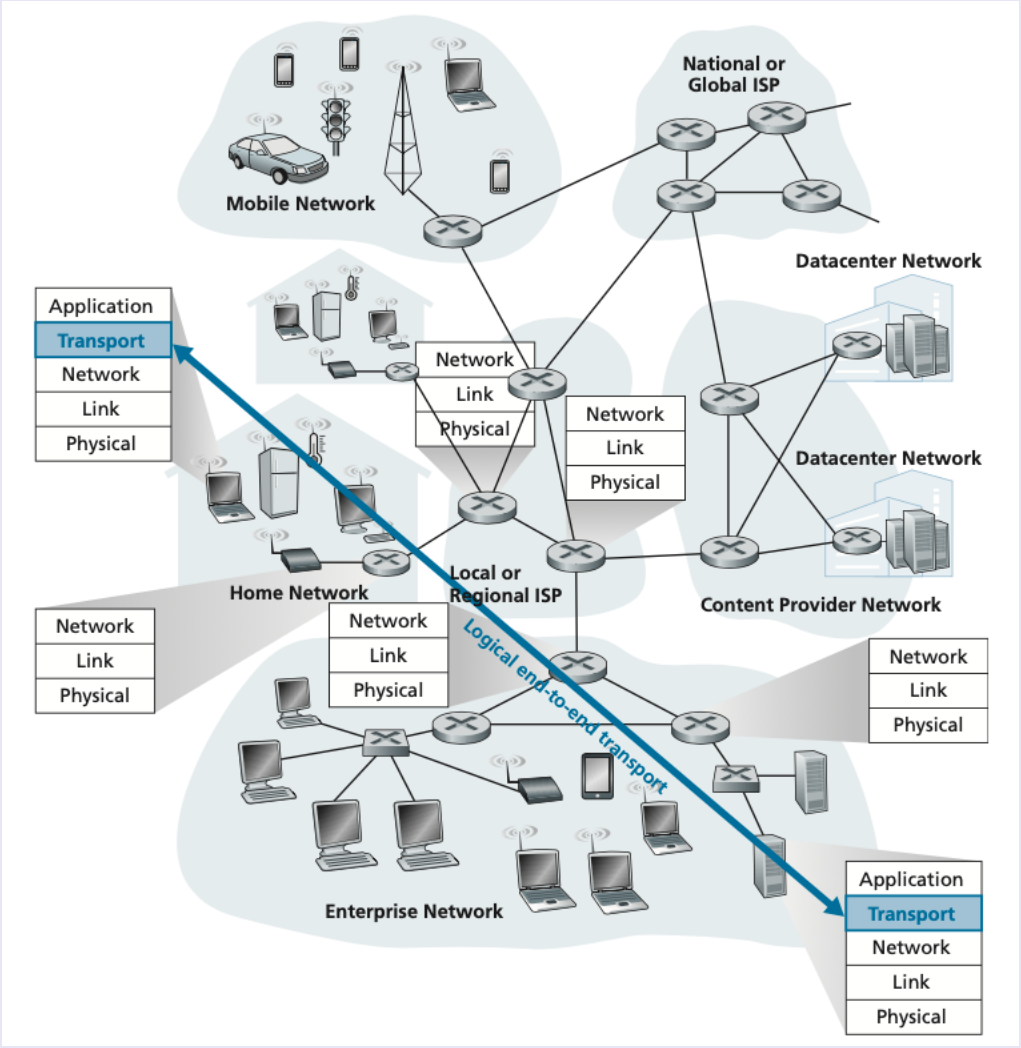
- The network layer extracts the transport-layer segment from the datagram and passes the segment up to the transport layer.
- The transport layer then processes the received segment, making the data in the segment available to the receiving application.

What about these transport-layer protocols?

- More than one transport-layer protocol may be available to network applications.
- The Internet has two protocols:
  - TCP
  - UDP.
- Each of these protocols provides a different set of transport-layer services to the invoking application.

How does the transport layer provide a logical communication between application processes?

How does the transport layer provide a logical communication between application processes?



Agenda del día

1. First Midterm
2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP
3. Exercises

Relationship Between Transport and Network Layers

What is the relationship between Transport and Network Layers?

Relationship Between Transport and Network Layers

What is the relationship between Transport and Network Layers?

- The transport layer lies just above the network layer in the protocol stack.
- A transport-layer protocol provides logical communication between processes running on different hosts.
- A network-layer protocol provides logical communication between hosts.
- Transport-layer protocols live in the end systems.
- Within an end system, a transport protocol moves messages from application processes to the network edge (i.e., the network layer) and vice versa.
  - It doesn't have any say about how the messages are moved within the network core.
  - Intermediate routers neither act on, nor recognize, any information that the transport layer may have added to the application messages.

Relationship Between Transport and Network Layers

What about the services between these two layers?

Relationship Between Transport and Network Layers

What about the services between these two layers?

- The services that a transport protocol can provide are often constrained by the service model of the underlying network-layer protocol.
- If the network-layer protocol cannot provide delay or bandwidth guarantees for transport-layer segments sent between hosts, then the transport-layer protocol cannot provide delay or bandwidth guarantees for application messages sent between processes.
- Although, certain services can be offered by a transport protocol even when the underlying network protocol doesn't offer the corresponding service at the network layer.
  - A transport protocol can offer reliable data transfer service to an application even when the underlying network protocol is unreliable.
  - A transport protocol can use encryption to guarantee that application messages are not read by intruders, even when the network layer cannot guarantee the confidentiality of transport-layer segments.

Agenda del día

1. First Midterm
2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP
3. Exercises

Overview of the Transport Layer in the Internet

Which are the two transport-layer protocols the internet makes available to the application layer?

Which are the two transport-layer protocols the internet makes available to the application layer?

- UDP (User Datagram Protocol)
  - Provides an unreliable, connectionless service to the invoking application.
- TCP (Transmission Control Protocol)
  - Provides a reliable, connection-oriented service to the invoking application.
- When designing a network application, the application developer must specify one of these two transport protocols.
- The application developer selects between UDP and TCP when creating sockets.

Which are the service models provided by UDP and TCP?

Which are the service models provided by UDP and TCP?

- The most fundamental responsibility is to extend IP's delivery service between two end systems to a delivery service between two processes running on the end systems.
  - Extending host-to-host delivery to process-to-process delivery is called transport-layer multiplexing and demultiplexing.
- Provide integrity checking by including error detection fields in their segments'headers.
- Additionally TCP offers several additional services to applications:
  - Reliable data transfer (flow control, sequence numbers, acknowledgments, and timers).
  - Ensures that data is delivered from sending process to receiving process, correctly and in order.
  - Converts IP's unreliable service between end systems into a reliable data transport service between processes.
  - Provides congestion control.

Agenda del día

1. First Midterm
2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP
3. Exercises

Multiplexing and Demultiplexing

In general terms, what is multiplexing and demultiplexing?

Multiplexing and Demultiplexing

In general terms, what is multiplexing and demultiplexing?

- It is extending the host-to-host delivery service provided by the network layer to a process-to-process delivery service for applications running on the hosts.

Multiplexing and Demultiplexing

In general terms, what is multiplexing and demultiplexing?

- It is extending the host-to-host delivery service provided by the network layer to a process-to-process delivery service for applications running on the hosts.

What happens at the destination host?

Multiplexing and Demultiplexing

In general terms, what is multiplexing and demultiplexing?

- It is extending the host-to-host delivery service provided by the network layer to a process-to-process delivery service for applications running on the hosts.

What happens at the destination host?

- The transport layer receives segments from the network layer just below.
- The transport layer has the responsibility of delivering the data in these segments to the appropriate application process running in the host.
- A process (as part of a network application) can have one or more sockets (two-way communication links).
  - Doors through which data passes from the network to the process and through which data passes from the process to the network.
- The transport layer in the receiving host does not actually deliver data directly to a process, but instead to an intermediary socket.
- Each socket has a unique identifier.

Multiplexing and Demultiplexing

How does a receiving host direct an incoming transport-layer segment to the appropriate socket?

Multiplexing and Demultiplexing

How does a receiving host direct an incoming transport-layer segment to the appropriate socket?

- Each transport-layer segment has a set of fields in the segment for this purpose.
- At the receiving end, the transport layer examines these fields to identify the receiving socket and then directs the segment to that socket.
- This job of delivering the data in a transport-layer segment to the correct socket is called demultiplexing.

How does a receiving host direct an incoming transport-layer segment to the appropriate socket?

- Each transport-layer segment has a set of fields in the segment for this purpose.
- At the receiving end, the transport layer examines these fields to identify the receiving socket and then directs the segment to that socket.
- This job of delivering the data in a transport-layer segment to the correct socket is called demultiplexing.

So what is multiplexing?

Multiplexing and Demultiplexing

How does a receiving host direct an incoming transport-layer segment to the appropriate socket?

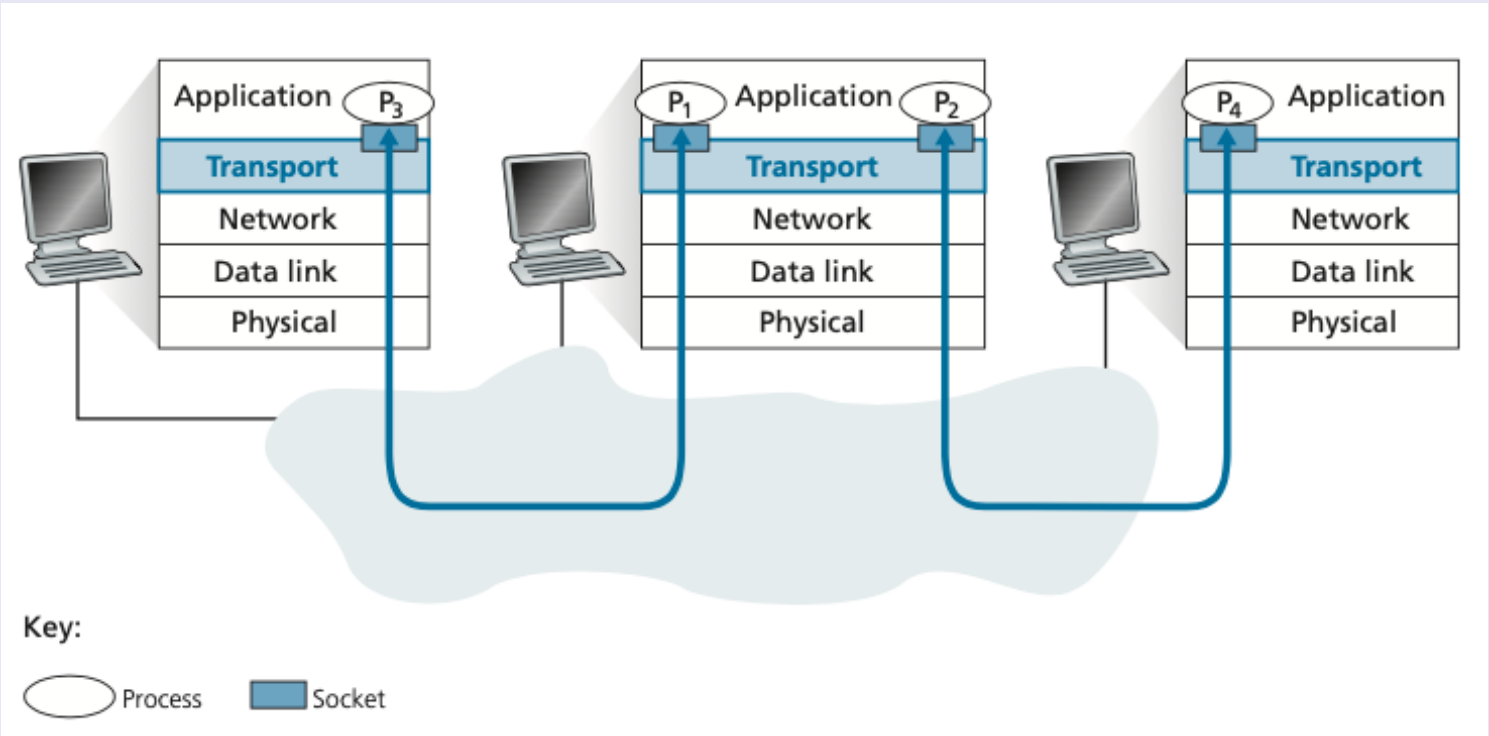
- Each transport-layer segment has a set of fields in the segment for this purpose.
- At the receiving end, the transport layer examines these fields to identify the receiving socket and then directs the segment to that socket.
- This job of delivering the data in a transport-layer segment to the correct socket is called demultiplexing.

So what is multiplexing?

- Is the job of gathering data chunks at the source host from different sockets.
- Encapsulating each data chunk with header information (that will later be used in demultiplexing) to create segments
- Passing the segments to the network layer.

How about an example of multiplexing/demultiplexing?

How about an example of multiplexing/demultiplexing?



What does multiplexing require?

What does multiplexing require?

- That sockets have unique identifiers.
- That segments have special fields that indicate the socket to which the segment is to be delivered.
  - The source port number field and the destination port number field.
  - Each port number is a 16-bit number, ranging from 0 to 65535.
  - The port numbers ranging from 0 to 1023 are called well-known port numbers and are restricted.

Multiplexing and Demultiplexing

What does multiplexing require?

- That sockets have unique identifiers.
- That segments have special fields that indicate the socket to which the segment is to be delivered.
  - The source port number field and the destination port number field.
  - Each port number is a 16-bit number, ranging from 0 to 65535.
  - The port numbers ranging from 0 to 1023 are called well-known port numbers and are restricted.

How can the transport layer implement the demultiplexing service?

Multiplexing and Demultiplexing

What does multiplexing require?

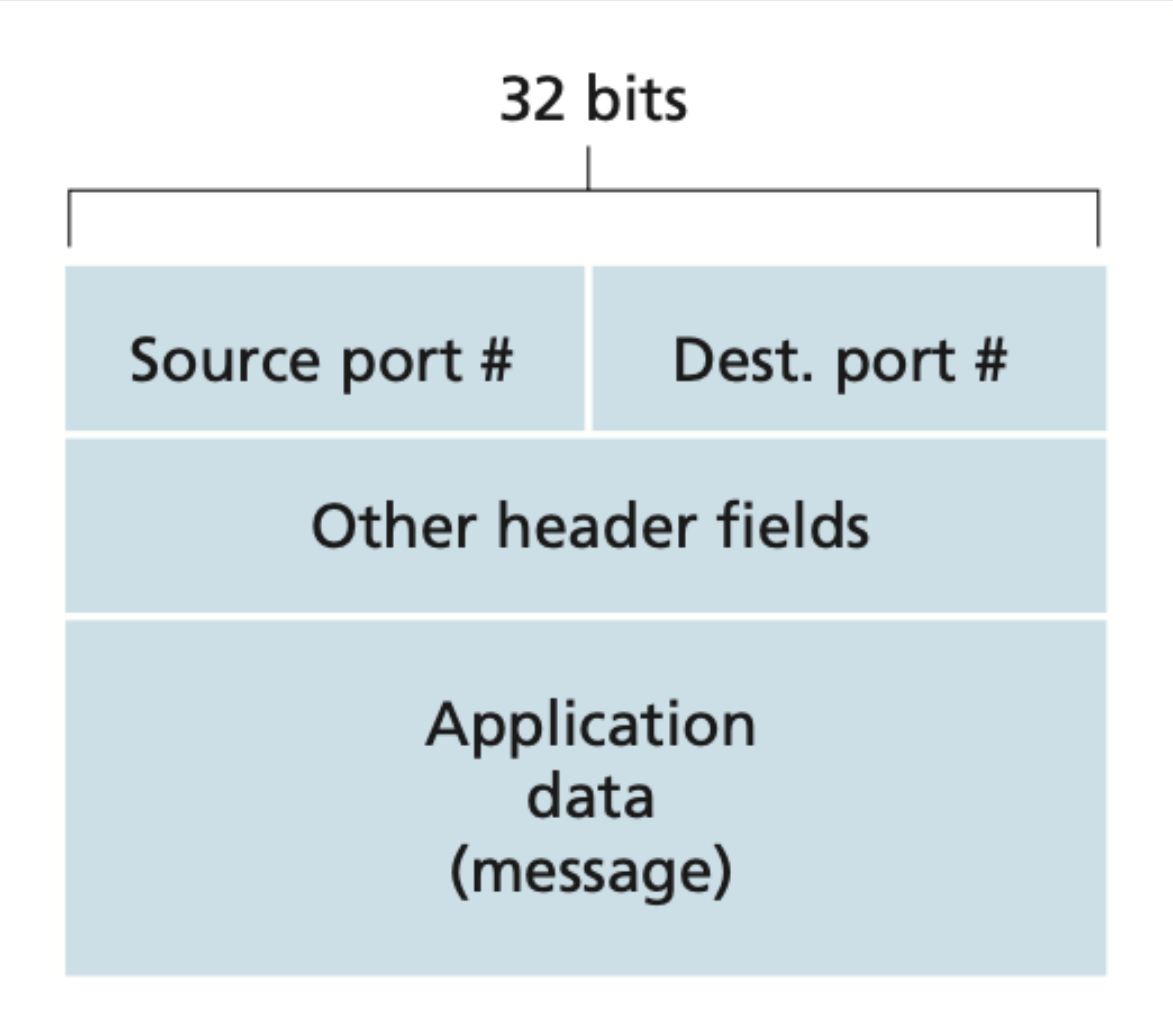
- That sockets have unique identifiers.
- That segments have special fields that indicate the socket to which the segment is to be delivered.
  - The source port number field and the destination port number field.
  - Each port number is a 16-bit number, ranging from 0 to 65535.
  - The port numbers ranging from 0 to 1023 are called well-known port numbers and are restricted.

How can the transport layer implement the demultiplexing service?

- Each socket in the host could be assigned a port number.
- When a segment arrives at the host, the transport layer examines the destination port number in the segment and directs the segment to the corresponding socket.
- The segment's data then passes through the socket into the attached process.

What is the structure of a transport-layer segment?

What is the structure of a transport-layer segment?



How do connectionless multiplexing and demultiplexing work?

How do connectionless multiplexing and demultiplexing work?

- When creating a UDP socket it gets assigned a port number, but you can bind it to a specific port afterwards.
- A process in Host A, with UDP port 19157, wants to send a chunk of application data to a process with UDP port 46428 in Host B.
- The transport layer in Host A creates a transport-layer segment.
- The transport layer passes the segment to the network layer.
- The network layer encapsulates it in an IP datagram and makes a best effort delivery.

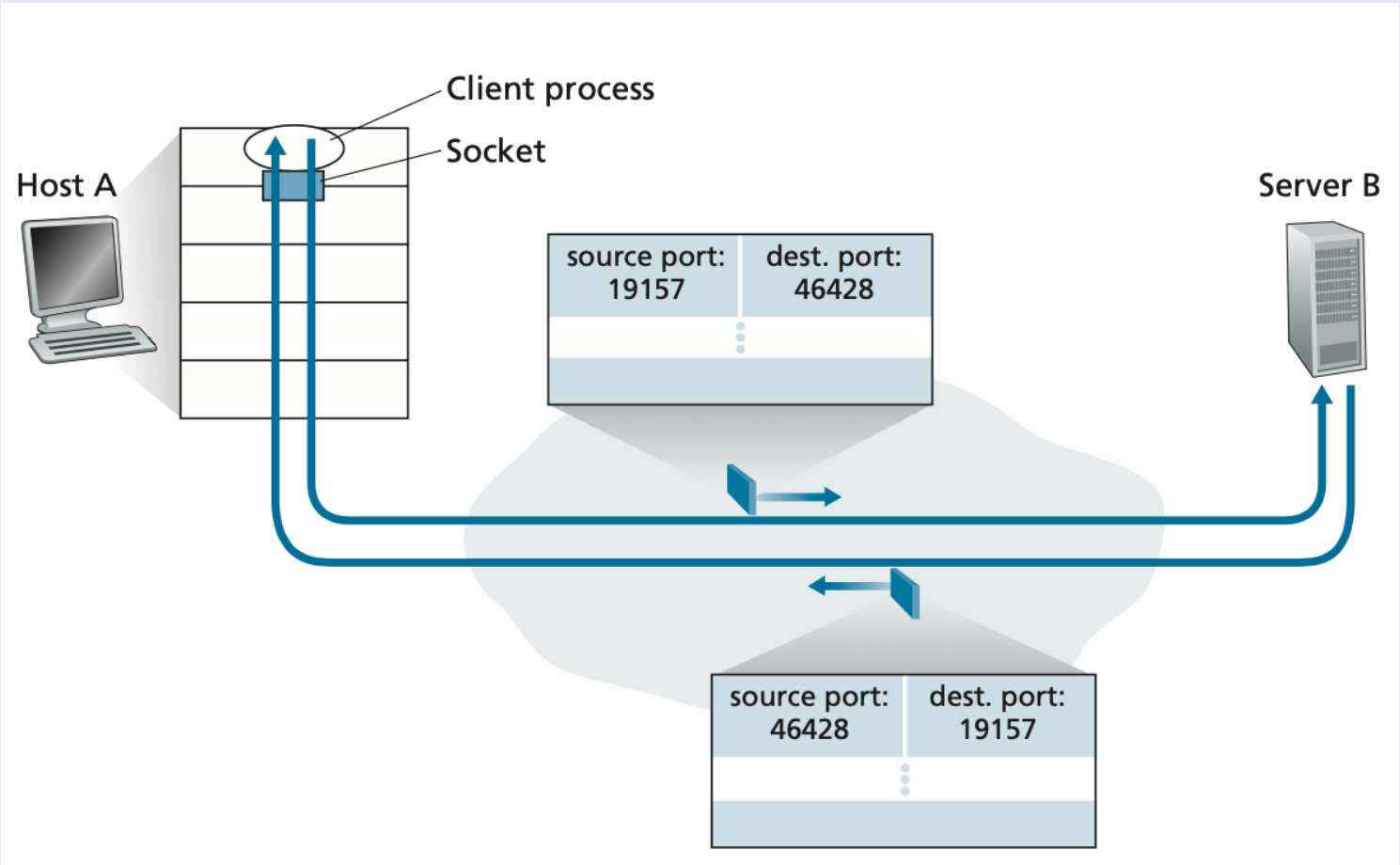
How do connectionless multiplexing and demultiplexing work?

- The transport layer at the receiving host examines the destination port number in the segment and delivers the segment to its socket identified by port 46428.
- Host B could be running multiple processes, each with its own UDP socket and associated port number.
- As UDP segments arrive from the network, host B directs (demultiplexing) each segment to the appropriate socket by examining the segments destination port number.
- If two segments have different source IP address and or source port, but the same destination IP address and port then both will go to the same socket.
- The source port and IP address is used as a return address if the server were to send/respond back to the client.

What is the purpose of the source port number?

Multiplexing and Demultiplexing

What is the purpose of the source port number?



How do connection-oriented multiplexing and demultiplexing work?

How do connection-oriented multiplexing and demultiplexing work?

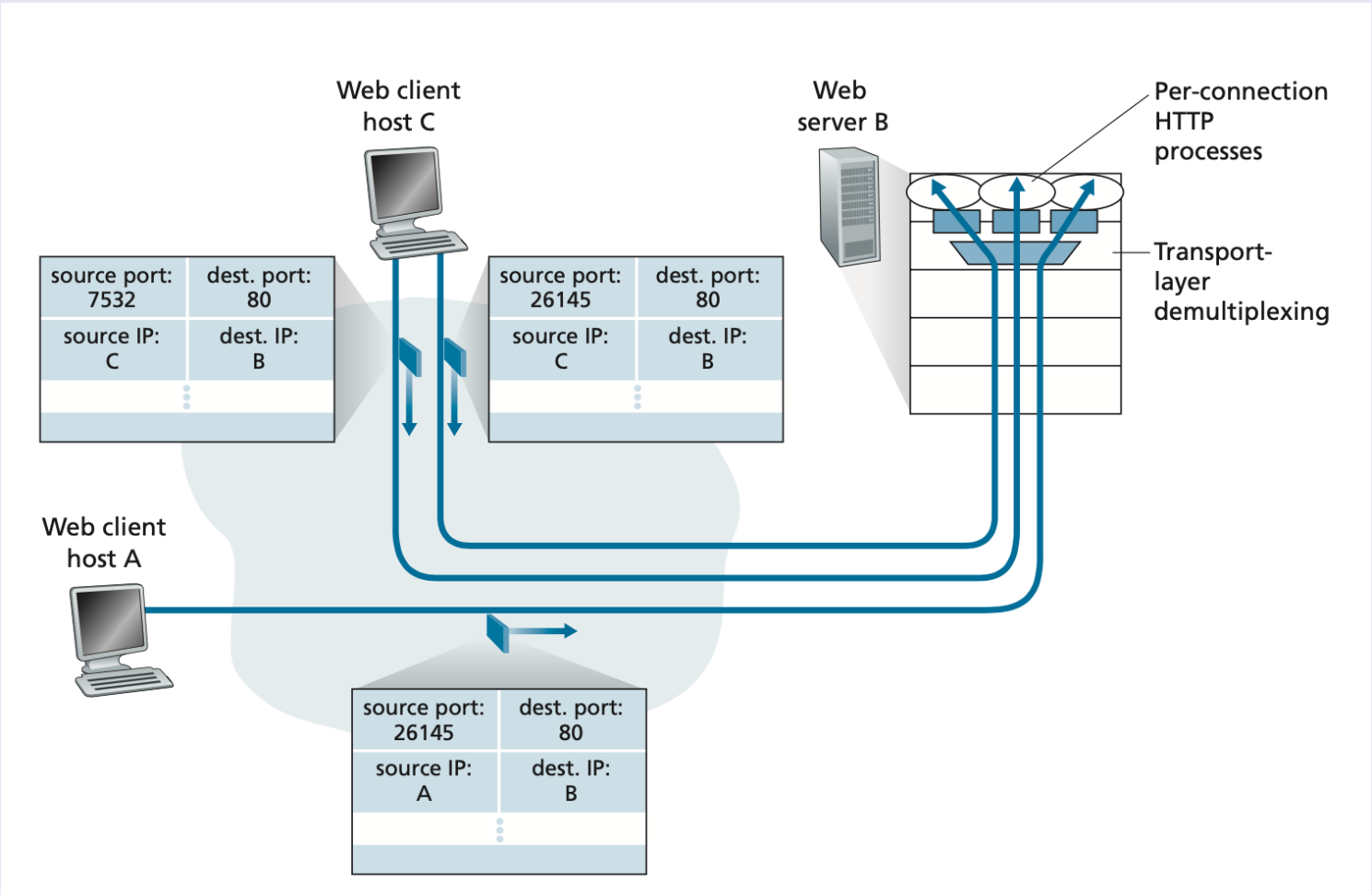
- The TCP socket is identified by a four-touple:
  - Source IP address
  - Source port number
  - Destination IP address
  - Destination port number
- When a TCP segment arrives to a host, the host uses all four values to direct (demultiplex) the segment to the appropriate socket.
- Two arriving segments with different source IP addresses or source port number will be directed to two different sockets.

Multiplexing and Demultiplexing

How can two clients, using the same destination port number, communicate with the same Web server application?

Multiplexing and Demultiplexing

How can two clients, using the same destination port number, communicate with the same Web server application?



Another example?

Another example?

- Consider a host running a Web server on port 80.
- All sent segments will have destination port 80.
- The server distinguishes the segments using source IP addresses and source port numbers.
- There is not always a one-to-one correspondence between connection sockets and processes.
- In fact, todays web servers often use only one process and create a new thread with a new connection socket for each new client connection.
- If the connection to the server is persistent then it uses the same socket during the whole connection.
- If it is a non-persistent connection a new TCP connection is created and closed for every request/response.

Agenda del día

1. First Midterm
2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP
3. Exercises

What does the User Datagram Protocol do?

What does the User Datagram Protocol do?

- Aside from the multiplexing/demultiplexing function and some light error checking, it adds nothing to IP.
- UDP takes messages from the application process.
- Attaches source and destination port number fields for the multiplexing/demultiplexing service.
- Adds two other small fields, and passes the resulting segment to the network layer.
- The network layer encapsulates the transport-layer segment into an IP datagram.
- It makes a best-effort attempt to deliver the segment to the receiving host.
- If the segment arrives at the receiving host, UDP uses the destination port number to deliver the segment's data to the correct application process.
- With UDP there is no handshaking between sending and receiving transport-layer entities before sending a segment.
- For this reason, UDP is said to be connectionless.

What application protocol often uses UDP?

What application protocol often uses UDP?

- DNS is an example of an application-layer protocol that typically uses UDP.
- When the DNS application in a host wants to make a query, it constructs a DNS query message and passes the message to UDP.
- Without performing any handshaking with the UDP entity running on the destination end system, the host-side UDP adds header fields to the message and passes the resulting segment to the network layer.
- The network layer encapsulates the UDP segment into a datagram and sends the datagram to a name server.
- The DNS application at the querying host then waits for a reply to its query.
- If it doesn't receive a reply (possibly because the underlying network lost the query or the reply), it might try resending the query, try sending the query to another name server, or inform the invoking application that it can't get a reply.

Connectionless Transport: UDP

Why would you choose to build an application over UDP rather than over TCP?

Why would you choose to build an application over UDP rather than over TCP?

- Finer application-level control over what data is sent, and when.
  - UDP will package the data inside a UDP segment and immediately pass the segment to the network layer.
  - Real-time applications often require a minimum sending rate, do not want to overly delay segment transmission, and can tolerate some data loss.
- No connection establishment.
  - UDP does not introduce any delay to establish a connection.
  - DNS would be much slower if it ran over TCP.
- No connection state.
  - UDP does not maintain connection state and does not track any parameters.
  - A server devoted to a particular application can typically support many more active clients when the application runs over UDP rather than TCP.
- Small packet header overhead.
  - UDP has only 8 bytes of overhead.

Which are some popular internet applications and their underlying transport protocols?

Which are some popular internet applications and their underlying transport protocols?

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Secure remote terminal access	SSH	TCP
Web	HTTP, HTTP/3	TCP (for HTTP), UDP (for HTTP/3)
File transfer	FTP	TCP
Remote file server	NFS	Typically UDP
Streaming multimedia	DASH	TCP
Internet telephony	typically proprietary	UDP or TCP
Network management	SNMP	Typically UDP
Name translation	DNS	Typically UDP

Connectionless Transport: UDP

Which is UDP's main drawback?

Which is UDP's main drawback?

- UDP has no congestion control.
- Congestion control is needed to prevent the network from entering a congested state in which very little useful work is done.
- The high loss rates induced by the uncontrolled UDP senders would cause the TCP senders to dramatically decrease their rates.
- The lack of congestion control in UDP can result in high loss rates between a UDP sender and receiver, and the crowding out of TCP sessions.

Connectionless Transport: UDP

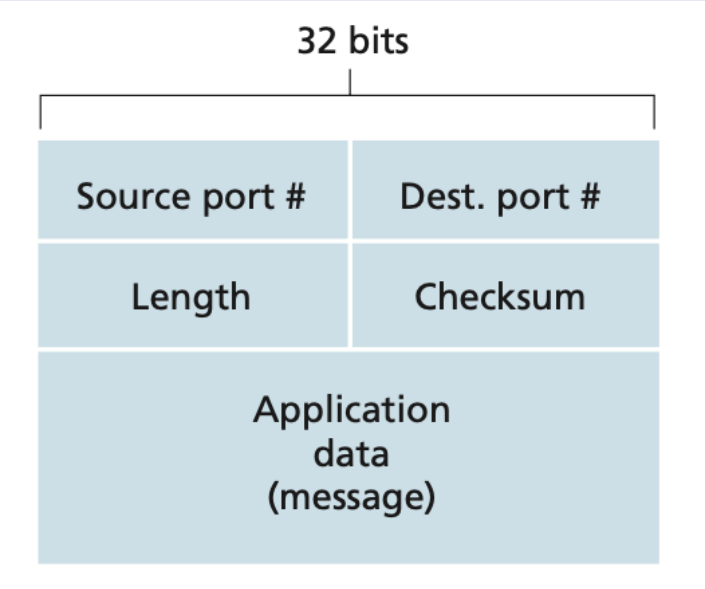
Ho can we have a reliable data transfer when using UDP?

Ho can we have a reliable data transfer when using UDP?

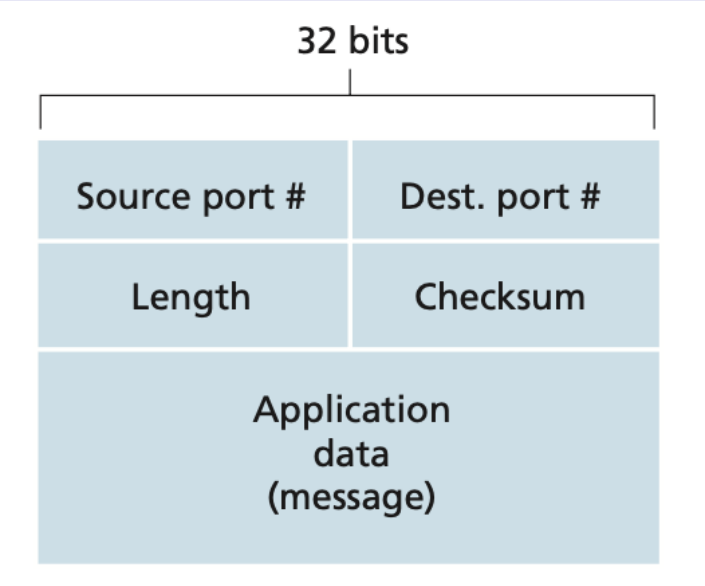
- This can be done if reliability is built into the application itself.
- Building reliability directly into the application allows application processes to communicate reliably without being subjected to the transmission-rate constraints imposed by TCP's congestion-control mechanism.

What is UDP's segment structure?

What is UDP's segment structure?



What is UDP's segment structure?



- The application data occupies the data field of the UDP segment.
- For DNS, the data field contains either a query message or a response message.
- For a streaming audio application, audio samples fill the data field.

What is UDP's segment structure?

- The UDP header has only four fields, each consisting of two bytes.
- The port numbers allow the destination host to pass the application data to the correct process running on the destination end system.
- The length field specifies the number of bytes in the UDP segment (header plus data).
- An explicit length value is needed since the size of the data field may differ from one UDP segment to the next.
- The checksum is used by the receiving host to check whether errors have been introduced into the segment.

How does the UDP Checksum work?

How does the UDP Checksum work?

- The UDP checksum provides for error detection.
- It is used to determine whether bits within the UDP segment have been altered as it moved from source to destination.
- UDP at the sender side performs the 1s complement of the sum of all the 16-bit words in the segment, with any overflow encountered during the sum being wrapped around.
- This result is put in the checksum field of the UDP segment.

What about an example?

What about an example?

We have the following three 16-bit words:

- 0110011001100000
- 0101010101010101
- 1000111100001100

Why does UDP provide a checksum?

Why does UDP provide a checksum?

- The reason is that there is no guarantee that all the links between source and destination provide error checking.
- One of the links may use a link-layer protocol that does not provide error checking.
- Even if segments are correctly transferred across a link, it's possible that bit errors could be introduced when a segment is stored in a router's memory.
- Given that neither link-by-link reliability nor in-memory error detection is guaranteed, UDP must provide error detection at the transport layer, on an end-end basis.
- Because IP is supposed to run over just about any layer-2 protocol, it is useful for the transport layer to provide error checking as a safety measure.
- Although UDP provides error checking, it does not do anything to recover from an error.
- Some implementations of UDP simply discard the damaged segment.
- Others pass the damaged segment to the application with a warning.

Agenda del día

1. First Midterm
2. **Transport Layer**
   - Introduction
   - Relationship Between Transport and Network Layers
   - Overview of the Transport Layer in the Internet
   - Multiplexing and Demultiplexing
   - Connectionless Transport: UDP
   - Socket Programming with UDP
3. Exercises

How does UDP work?

How does UDP work?

- UDP is a connectionless service (no initial handshaking phase during which a pipe is established between the two processes).
- When a process wants to send a batch of bytes to another process, the sending process must attach the destination process's address to the batch of bytes.
- This must be done for each batch of bytes the sending process sends.
- UDP provides an unreliable message-oriented service model.
- It is message-oriented in that batches are bytes that are sent in a single zero operation at the sending side, will be delivered as a batch at the receiving side.
- After having created a packet, the sending process pushes the packet into the network through a socket.
- UDP makes no guarantees that a packet will reach its ultimate destination.

Let us build a a simple application using UDP

Let us build a a simple application using UDP

1. A client reads a line from its standard input (keyboard) and sends the line out its socket to the server.
2. The server reads a line from its socket.
3. The server converts the line to uppercase.
4. The server sends the modified line out its socket to the client.
5. The client reads the modified line from its socket and prints the line on its standard output (monitor).

What should we expect from the code?

What should we expect from the code?

1. No initial handshaking between the two processes and therefore no need for a welcoming socket.
2. No streams are attached to the sockets.
3. The sending hosts create packets by attaching the IP destination address and port number to each batch of bytes it sends.
4. The receiving process must unravel each received packet to obtain the packet's information bytes.

How would the code for the client side of the application be?

How would the code for the client side of the application be?

```java
import java.io.*;
import java.net.*;

public class UDPClient {
  public static void main(String args[]) throws Exception {
    BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));
    DatagramSocket clientSocket = new DatagramSocket();
    InetAddress IPAddress = InetAddress.getByName("127.0.0.1");
    byte[] sendData = new byte[1024];
    byte[] receiveData = new byte[1024];
    String sentence = inFromUser.readLine();
    sendData = sentence.getBytes();
    DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,IPAddress,9876);
    clientSocket.send(sendPacket);
    DatagramPacket receivePacket = new DatagramPacket(receiveData,receiveData.length);
    clientSocket.receive(receivePacket);
    String modifiedSentence = new String(receivePacket.getData());
    System.out.println("FROM SERVER:" + modifiedSentence);
    clientSocket.close();
  }
}
```

How would the code for the server side of the application be?

How would the code for the server side of the application be?

```
1  import java.io.*;
2  import java.net.*;
3  public class UDPServer {
4    public static void main(String args[]) throws Exception {
5      DatagramSocket serverSocket = new DatagramSocket(9876);
6      byte[] receiveData = new byte[1024];
7      byte[] sendData = new byte[1024];
8      while(true) {
9        DatagramPacket receivePacket = new DatagramPacket(receiveData,receiveData.length);
10       serverSocket.receive(receivePacket);
11       String sentence = new String(receivePacket.getData());
12       InetAddress IPAddress = receivePacket.getAddress();
13       int port = receivePacket.getPort();
14       String capitalizedSentence = sentence.toUpperCase();
15       sendData = capitalizedSentence.getBytes();
16       DatagramPacket sendPacket = new DatagramPacket(sendData,sendData.length,IPAddress,port);
17       serverSocket.send(sendPacket);
18     }
19   }
20 }
21
```

Exercises

1. Write a UDP client/server system in which the client program sends an integer number and the server program returns its square.
2. Write a Simple Calculator via UDP.