SISTEMAS WEB
CURSO 2022/2023

HTTP - HyperText Transfer Protocol

Content Length vs Transfer Encoding

María Luz Álvarez y Oskar Casquero
Dpto. Ingeniería de Sistemas y Automática

FUNCIONAMIENTO DE HTTP:
CONTENT LENGTH Y TRANSFER ENCODING

- Un cliente web tiene que leer una respuesta enviada desde un servidor web, necesita saber dónde acaba el cuerpo del mensaje.
  - Una forma que tiene el protocolo HTTP de expresar la longitud del cuerpo del mensaje es con la cabecera "Content-Length".
    - NOTA: la cabecera “Content-Length” requiere conocer el tamaño del contenido que se va a incluir en el cuerpo del mensaje de la respuesta antes de enviarlo.

- A veces hay que enviar una gran cantidad de datos en la respuesta, pero no es posible conocer el tamaño de esos datos hasta que la petición ha sido procesada en su totalidad. Ejemplo: supongamos que los resultados de una solicitud a una base de datos se quieren visualizar en una página web.
  - Si se utiliza la cabecera "Content-Length", hay que conocer el tamaño de los datos que van en el cuerpo del mensaje antes de enviar la respuesta HTTP. Esto, por un lado, supone esperar a finalizar el procesamiento de la solicitud de la base de datos. Por otro lado, para guardar los datos se necesita un gran buffer.
  - El retraso generado por el tiempo necesario para procesar la respuesta antes de su envió influye negativamente en la experiencia del usuario.

- El protocolo HTTP proporciona un mecanismo de ir enviando partes del contenido solicitado por el cliente mientras el servidor web procesa la solicitud: cabecera "**Transfer-Encoding**" con valor "**chunked**".

FUNCIONAMIENTO DE HTTP:
CONTENT LENGTH Y TRANSFER ENCODING

- Ejemplo de "**Content-Length**": supongamos que el servidor tiene que enviar el texto "Hello World!!"

Respuesta HTTP con "Content-Length"

```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Content-Length: 13
Content-Type: text/plain

Hello World!!
```

FUNCIONAMIENTO DE HTTP:
CONTENT LENGTH Y TRANSFER ENCODING

- NOTA: si el contenido de una respuesta es mayor que la MTU (Maximum Transmission Unit) de la capa TCP, el mensaje HTTP se envía en varios segmentos.
- Ejemplo: supongamos que la MTU (Maximum Transmission Unit) de la capa TCP son 187 octetos.

Respuesta HTTP con "Content-Length"

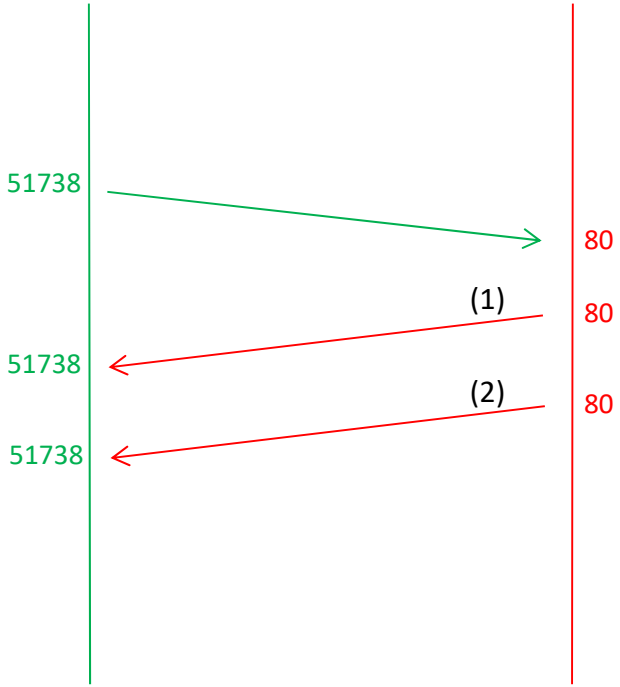
(1)

```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Content-Length: 13
Content-Type: text/plain

Hello
```

(2)

```
World!!
```



FUNCIONAMIENTO DE HTTP:

CONTENT LENGTH Y TRANSFER ENCODING

- Ejemplo "**Transfer-Encoding: chunked**": supongamos que el servidor tiene que enviar el texto "Hello World!!"

Respuesta HTTP con "Transfer-Encoding"

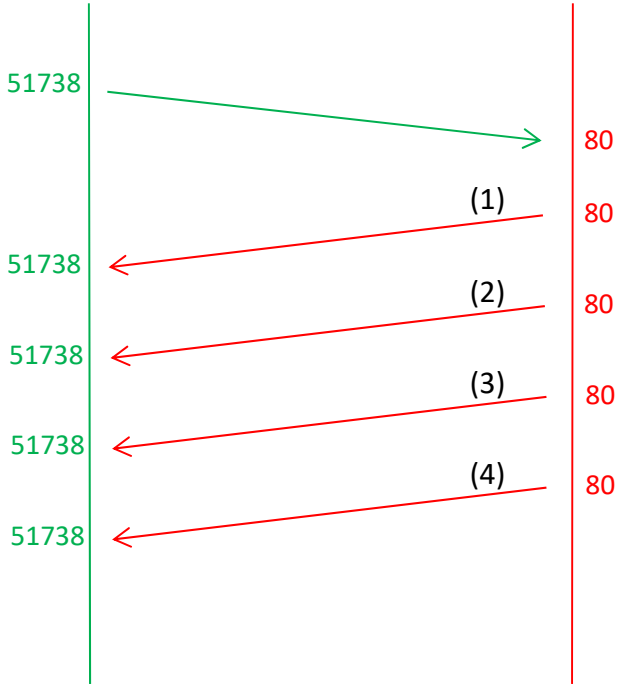
(1)
```
HTTP/1.1 200 OK
Date: Thu, 20 Nov 2015 20:25:52 GMT
Last-Modified: Tue, 17 Sep 2015 13:00:02 GMT
ETag: "1a968-3ec-4e693e61bb8b6"
Content-Type: text/plain
Transfer-Encoding: chunked

6\r\nHello \r\n
```

(2)
```
6\r\nWorld!\r\n
```

(3)
```
1\r\n!\r\n
```

(4)
```
0\r\n\r\n
```



La longitud de cada trozo se indica en hexadecimal.

Los trozos 2, 3 y 4 no llevan cabeceras HTTP. Van directamente encapsulados en segmentos TCP.

FUNCIONAMIENTO DE HTTP:
CONTENT LENGTH Y TRANSFER ENCODING

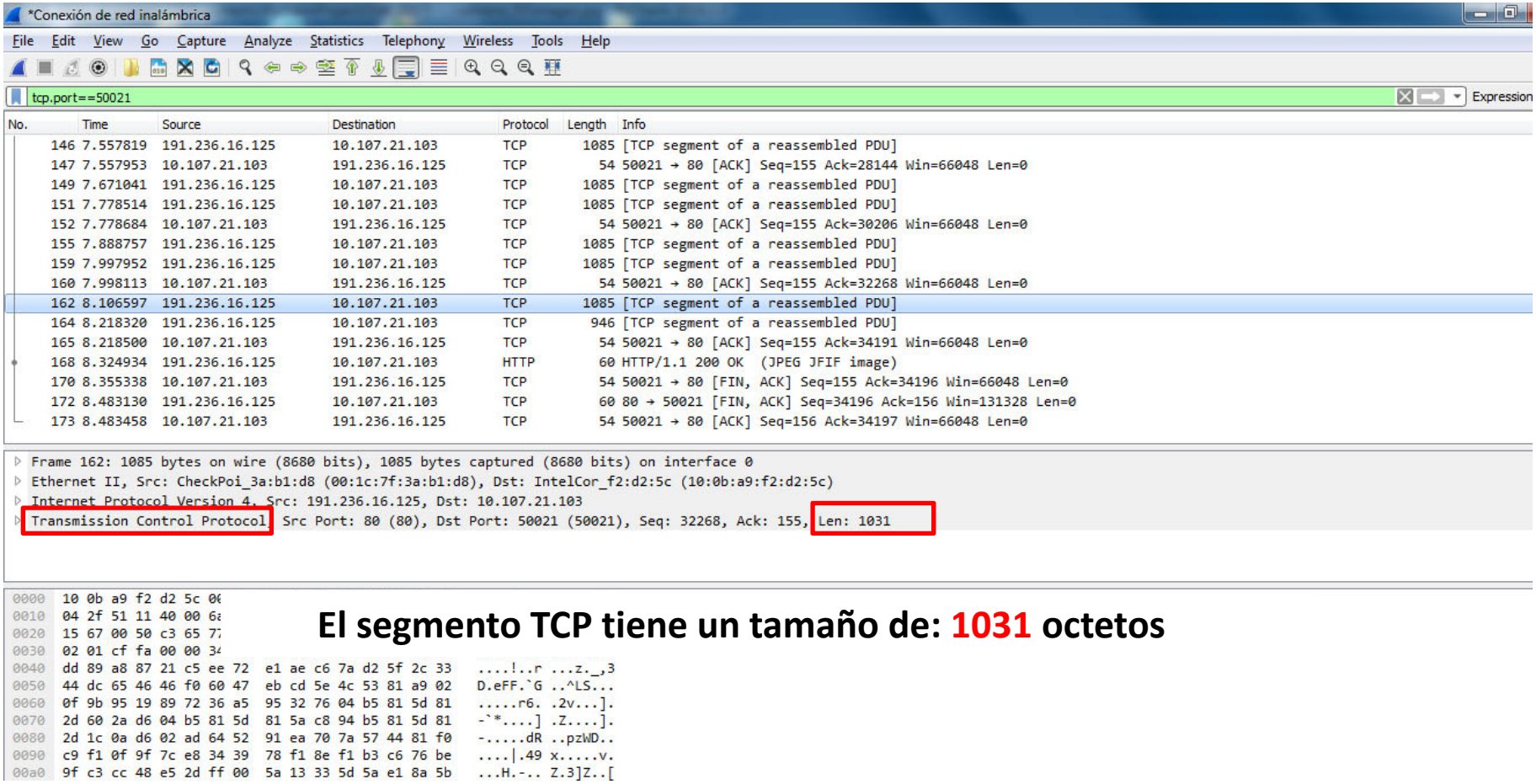
- Diferencias entre ambos mecanismos:
  - con **"Content-Length"** los mensajes no se empiezan a enviar hasta que se ha procesado toda la respuesta (latencia alta), mientras que con **"Transfer-Encoding"** los mensajes se envían según se van procesando los trozos (latencia baja)
  - con **"Content-Length"** el troceado de los mensajes (dependiente de la MTU) lo hace la capa TCP, mientras que con **"Transfer-Encoding"** lo hace la propia capa HTTP.

EJEMPLO:
ANÁLISIS DE LA DESCARGA POR "TROZOS" DE UNA IMAGEN

- Haciendo uso del navegador descarga la siguiente imagen:
  http://www.httpwatch.com/httpgallery/chunked/chunkedimage.aspx

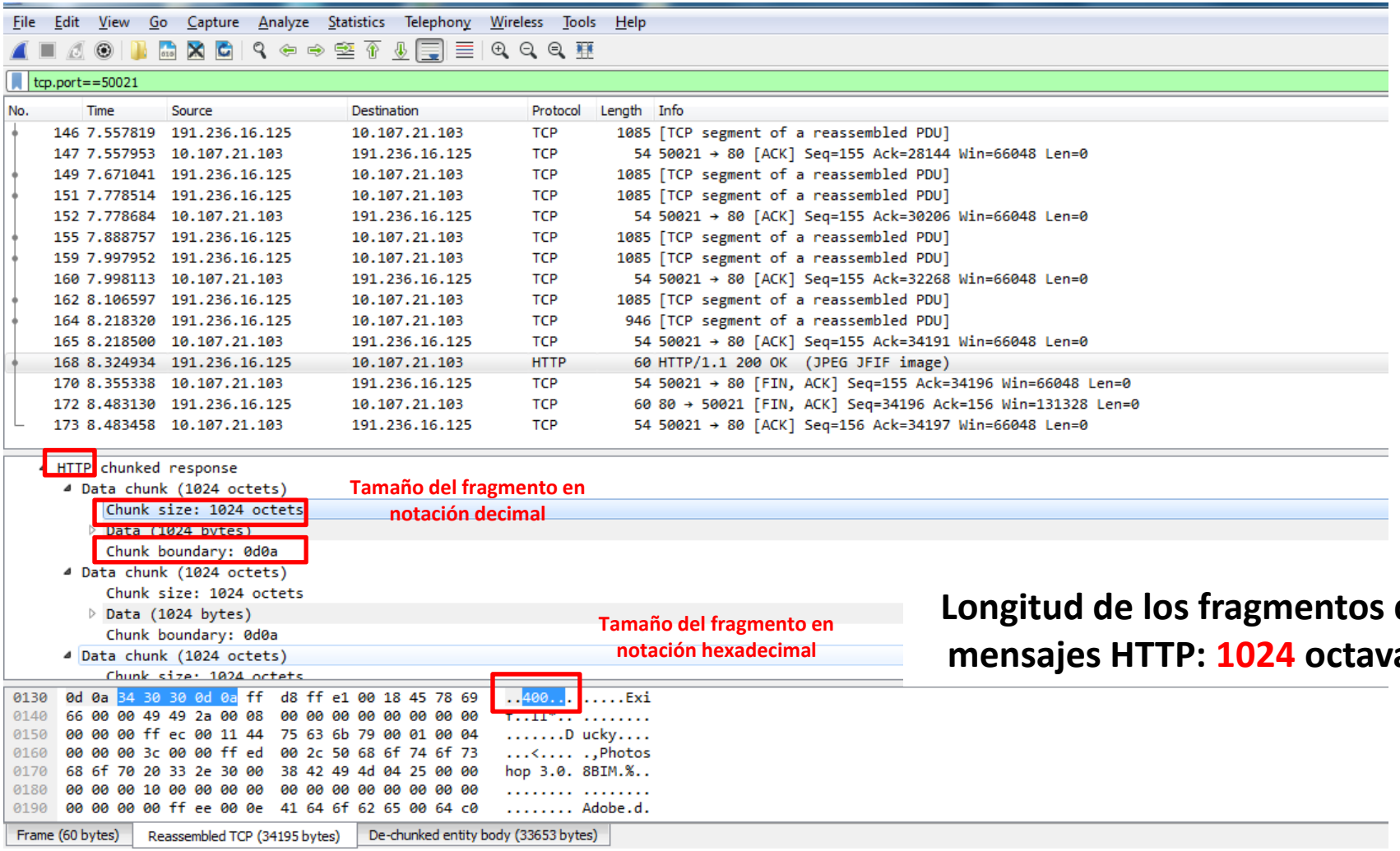
- Utilizando Wireshark analiza la captura
  - ¿Cuál es la relación entre el tamaño del campo de datos del segmento TCP y la parte del mensaje HTTP?
    - 3 + 2 + 1024 + 2 = 1031 → Explicar de dónde salen estos números

EJEMPLO:
ANÁLISIS DE LA DESCARGA POR "TROZOS" DE UNA IMAGEN



El segmento TCP tiene un tamaño de: 1031 octetos

EJEMPLO:
ANÁLISIS DE LA DESCARGA POR "TROZOS" DE UNA IMAGEN



Longitud de los fragmentos de mensajes HTTP: 1024 octavas

EJEMPLO:
ANÁLISIS DE LA DESCARGA POR "TROZOS" DE UNA IMAGEN

- RESUMEN:
  - Como se puede ver en las imágenes anteriores,
    - La longitud del contenido del segmento TCP es: **1031** octetos
    - La longitud del trozo del mensaje HTTP es: **1024** octetos
  - Un fragmento HTTP tiene la siguiente forma:
    - *LONG_FRAGMENTO*\r\nCONTENIDO_FRAGMENTO\r\n
  - El fragmento HTTP se mete en el campo de contenido de TCP, por tanto:
    - len("1024") + len("\r\n") + 1024 + len("\r\n") = 4 + 2 + 1024 + 2 = **1032**
  - 1031 ≠ 1032 --> ¿Por qué no coinciden los números?
    - ¡¡¡La longitud del fragmento debe meterse en notación hexadecimal!!!
      - 1024 en notación decimal = 400 en notación hexadecimal
      - len("400") + len("\r\n") + 1024 + len("\r\n") = 3 + 2 + 1024 + 2 = **1031**