Telematic Applications 2019. Continuous assesment II.
Fecha: 20/12/2019

uc3m | Universidad Carlos III de Madrid

Given name:_____
Surname(s):_____

Instructions
Available time : 30 min test + 90 min problems
Explain **clearly but briefly** every step you make in the exam considering that a correct answer with no explanation may not necessarily be considered valid. Answer the exam using exam paper sheets. Explicitly describe any supposition you make, however, unless contradicted by the question, you can assume the following:

- every segment is acknoledged individually
- if the size of a request or response is given, it will comprehend also the body unless stated the contrary
- there is no need to copy from the exam into the response pieces of text, you can reference them.

Problem 1 HTTP + TCP Javier has just recorded a football match and wants to upload the video to a video share portal. The video portal works in the following way to upload a video:

Step 1 The client requests the page at URL `http://www.supervideos.com/index.html`. In the response to the request will find a web page with a form called FormPage and described below, to upload the video.

Step 2 The client fill the form with the video title and selects the video file to be uploaded. Once he clicks, the video is delivered to the server URL `http://www.supervideos.com/upload.html` and the client obtains a response with the result called ResultPage and described below.

The page FormPage is (text/html, 278 bytes long):

```
<html>
<h1>Bienvenidos a SuperVideos</h1>
<p> Sube tu video aquí: </p>
<form action="/upload.html" method="POST" enctype="multipart/form-data">
    <input type="text" name="title"/>
```

```
    <input type="file" name="video"/>
    <input type="submit" value="Submit" />
</form>
</html>
```

The page ResultPage is (text/html, 60 bytes long):

```
<html>
<h1>Video subido</h1>
<p> Muchas gracias </p>
</html>
```

Answer to the following questions:

1. (2 pt) Write down the HTTP/1.1 request made by Javier in Step 1 and the response obtained from the server. **Clearly** state how the server knows the client request has finished and how the client knows when the server response ends.

   Solution: Javier will perform a DNS query of type A for the domain name `www.supervideos.com`. He will obtain an IP address and, using port 80 (default as the port is not explicitly specified), will open a TCP connection to the obtained IP.

   The HTTP request made over that TCP connection will be (lines marked with `--` are comments that would not appear in the HTTP protocol):

   ```
   GET /index.html HTTP/1.1
   Host: www.supervideos.com
   -- blank line (1)

   HTTP/1.1 200 OK
   Date: 19/12/2019
   Content-Type: text/html
   Content-Length: 278 -- (3)
   -- blank line (2)
   [Response body of type text/html and length 278 bytes]
   ```

   Since the GET method does not have a body, the client signals the end of the request to the server using a blank line after the headers (1). The server, recognizing the request as a GET, will respond immediately after receiving the blank line. As HTTP/1.1 connections are persistent, the server will inform the client of the size of the returned resource (in this case, 278 bytes) through the `Content-Length` header, allowing the client to know when the response is complete.

2. (2 pt) Have the request or the response in the previous question any HTTP/1.1 mandatory header? If so, explain the purpose of the headers. Would it be necessary to add more headers that the mandatory ones (if any)?

   Solution: Yes, HTTP/1.1 requires the inclusion of the `Host` header in the request, which allows for the use of virtual hosts, helping the server distinguish between the possible hosts that may exist on a server. Additionally, it mandates the use of the `Date` header in responses for cache management.

   In the response, it is necessary to include the `Content-Length` header so that the client can determine the total size of the response. Furthermore, it is recommended (though not required) to include a `Content-Type` header to indicate the type of content being sent.

3. (2.5 pt) What kind of request will make Javier in step 2? Proceed as in the first question and write down the HTTP/1.1 request made by Javier in Step 1 and the response obtained from the server. Please also **clearly** state how the server knows the client request has finished and how the client knows when the server response ends.

   Consider that the form has the Content-Type "multipart/form-data" and the result of such encoding will contain the text of the title and the video (both together will be 19321 bytes long).

   Solution: When sending data to the server, the request must be of type `POST`. It will be performed over the connection already established in step 1. The HTTP request made over that TCP connection will be (lines marked with `--` are comments that would not appear in the HTTP protocol):

```
POST /upload.html HTTP/1.1
Host: www.supervideos.com
Content-Type: multipart/form-data
Content-Length: 19321 (1)
-- blank line (2)
[Body of 19321 bytes]

HTTP/1.1 200 OK
Date: 19/12/2019
Content-Type: text/html
Content-Length: 60 -- (3)
-- blank line (4)
[Response body of type text/html and length 60 bytes]
```

   The `POST` request has a body, so the blank line marked by (2) does not indicate the end of the request, as it does with `GET` requests. Therefore, it is necessary to use the

`Content-Length` header (1) to indicate the size of the body, and thus the end of the request.

For the response, since the connection remains open, it is necessary, as in the previous section, to use the `Content-Length` header.

4. (3.5 pt) Javier sends the video using is mobile phone's LTE interface that has a MTU of 1428 bytes. Considering that:

   - IP and TCP headers have no options
   - the request of step 1 is one segment long
   - the response of step 1 is one segment long
   - the request of step 2 is one segment long is 19432 bytes long (111 bytes from the request and 19321 of the body)
   - the response of step 2 is one segment long

   How long would it take (in RTTs) to upload the video to the server (the time Javier expends filling the form do not count)? What size Will have the congestion Windows of the server and the client when it finishes?

   Solution: LTE has an MTU of 1428, resulting in an MSS of $MSS = 1428 - 40 = 1388$ bytes. After establishing the connection (approximately 1.5 RTTs), both the client and the server will have a congestion window of one (there is no indication of any other limitation, so we assume $V_{ef} = cwnd$). Thus, ($cwnd_{Cli} = 1$, $cwnd_{Srv} = 1$), and we assume slow start as the *ssthresh* is estimated at 65535 bytes (47.21 segments).

   The client sends the request from step one, which occupies one segment. The server responds with an ACK, increasing the client's window ($cwnd_{Cli} = 2$, $cwnd_{Srv} = 1$). Immediately, the server sends the response, which occupies one segment. Upon receiving it, the client acknowledges the response, increasing the server's window ($cwnd_{Cli} = 2$, $cwnd_{Srv} = 2$). This process takes approximately 1.5 RTTs.

   After receiving the response, Javier sends the completed form containing the video. The request in step 2 occupies 19432 bytes, which corresponds to $\frac{19432}{1388} = 14$ segments. The client will send as much as the window allows:

   - Initially, it sends 2 segments ($cwnd_{Cli} = 2$, $cwnd_{Srv} = 2$), receiving 2 ACKs ($cwnd_{Cli} = 4$, $cwnd_{Srv} = 2$). - Then, it sends 4 segments ($cwnd_{Cli} = 4$, $cwnd_{Srv} = 2$) and receives 4 ACKs ($cwnd_{Cli} = 8$, $cwnd_{Srv} = 2$). - Finally, it sends 8 segments ($cwnd_{Cli} = 8$, $cwnd_{Srv} = 2$) and receives 8 ACKs ($cwnd_{Cli} = 16$, $cwnd_{Srv} = 2$), completing the request.

   The server will then respond with the response from step 2 ($cwnd_{Cli} = 16$, $cwnd_{Srv} = 2$), and upon receiving the final ACK ($cwnd_{Cli} = 16$, $cwnd_{Srv} = 3$), the exchange concludes with the windows at ($cwnd_{Cli} = 16$, $cwnd_{Srv} = 3$). This process takes approximately 3.5 RTTs.

To send these segments, the total time will be approximately $1{,}5 + 1{,}5 + 3{,}5 = 6{,}5$ RTTs.