Lab 2:
HTTP Protocol

1 Previous work.

It is essential for the correct development of this lab-2 to study HTTP protocol (versions 1.0 and 1.1). Thus, you will be able to correctly answer the questions and better understand the captured traffic.

You can read about HTTP in:

- Kurose's book : chapter 2.2
- HTTP: http://www.rfc-editor.org/rfc/rfc2616.txt

Also, in the development of the lab, you can find useful the documents referred in the following links:

- Guía usuario Wireshark: http://www.wireshark.org/docs/
- Monitor de red de Firefox https://developer.mozilla.org/en-US/docs/Tools/Network_Monitor

2 The lab-2 aims.

At the end of the lab-2 you should be able to:

- Related to Netcat
  - know how to use nc program as a basic client TCP.
    - e.g. to know how to send a HTTP 1.0 or HTTP 1.1 request.
- Related to Wireshark
  - know how to use Wiershark capture filters.
    - e.g. the use of capture filters to capture only HTTP traffic.
  - know how to use Wireshark to see the client/server dialog.
    - e.g. the use of "Follow TCP Stream" command.
- Related to browser tools
  - know how to use network tools to analyse request headers and responses.

- know how to use network tool to carry out a performance analysis.

- Related to HTTP
  - know the mandatory and the optional header in both HTTP 1.0 and 1.1 version.
  - know the request/ response HTTP message format.
  - know how and when conditional GET is used.
  - know the meaning and use of headers: *Date*, *Last-Modified* y *Content-Length*.

3 First Part. HTTP request.

3.1 Use of netcat to send a HTTP request.

As we learned in Lab-1, Netcat (often abbreviated to `nc`) is a computer networking utility for reading from and writing to network connections using TCP or UDP.

Since the HTTP (1.0 and 1.1) protocol is a character-oriented protocol its syntax is perfectly readable, so we can perfectly use the `nc` command to compose requests and get answers.

Exercise 1: Netcat Use.

Use netcat to send a GET HTTP 1.0 request to the web server running in the IP address 158.42.180.62 (serveis.redes.upv.es) and port 80, to get the document whose URL is http://serveis.redes.upv.es/ejercicio1.html.

To stablish the TCP connection type the following command:

```
nc -C 158.42.180.62 80
```

The -C option tells netcat to send the character pair <CR> <LF> when the <Enter> key is pressed (in OSX the "c" must be lowercase).

Once the TCP connection is stablished with the server in the IP address 158.42.180.62 (*`serveis-rdc.redes.upv.es`*), you can send the GET HTTP request:

```
GET /ejercicio1.html HTTP/1.0 ↵
↵
```

Note: The carriage return key must be pressed twice after the request line. As a result of this action you will receive a text similar to the following:

```
HTTP/1.1 200 OK
```

```
Date: Thu, 06 Oct 2016 17:13:16 GMT
Server: Apache
Accept-Ranges: bytes
X-UA-Compatible: IE=EmulateIE7, IE=9
Connection: close
Content-Type: text/html; charset=ISO-8859-1
Content-Language: es
                ...
```

The received content was shown by the standard output (screen). If you want to save this text in a file, you can redirect the output to a file using the ">" operator.

Questions:

Question 1.1: What does the "-C" option do in the "nc" command? Why is it necessary?

Question 1.2: What is the response code? What does that code mean?

Very important note: If the code of the answer is 400 ask your teacher.

Exercise 2: Uso de netcat (continuación).

Repeat exercise 1,but this time using the HTTP 1.1 version.

Questions:

Question 2.1. Which headers are mandatory in the client request?

Question 2.2. How is indicated the end of headers in the client request and in the server response?

Question 2.3. When was the last time the html file was modified on the server?

Question 2.4. What is the difference between `Date` and `Last-modified` headers?

Question 2.5. What is the size in bytes of the page sent by the server indicated in the "`Content-Length`" header?

Exercise 3: Virtual server access.

Connect to the same server using the command:

```
nc -C 158.42.180.62 80
```

Make the same request as in the previous exercise, but this time using the header:

```
Host: zoltar.redes.upv.es
```

Questions:

Question 3.1: Do you get the same response as in the previous exercise?

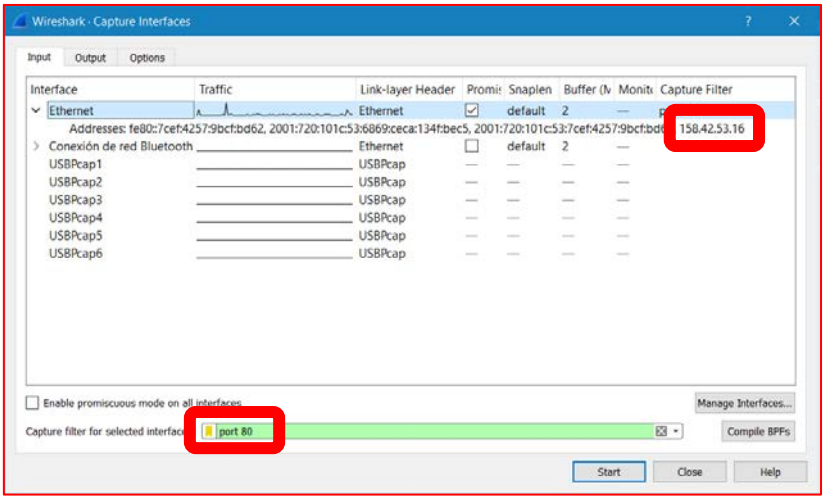
Question 3.2: Can you explain what is happening?

3.2 Use of Wireshark as protocol analyser.

In this part of the lab we will work with the Wireshark protocol analyser, to analyse the HTTP traffic exchanged between our browser and the web servers.

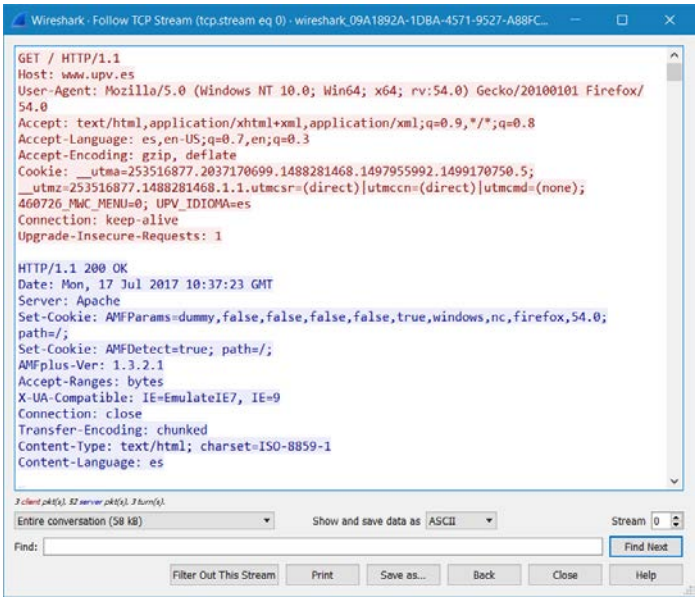
Remember that before generating the traffic (request the indicated web pages) you must configure the traffic capture:

- Select the interface with the public IP address (it will start with 158.42.)
- Define the capture filter for HTTP traffic ("`port 80`")..

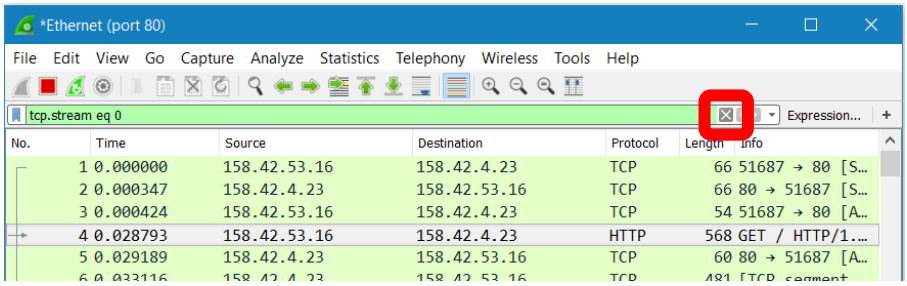


Following TCP streams

If you are working with TCP based protocols it can be very helpful to see the data from a TCP stream in the way that the application layer sees it. Perhaps you are trying to make sense of a data stream. Maybe you just need a display filter to show only the packets of that TCP stream. If so, Wireshark's ability to follow a TCP stream will be useful to you. Simply select a TCP packet in the packet list of the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Wireshark Tools menu (or use the context menu in the packet list). Wireshark will set an appropriate display filter and pop up a dialog box with all the data from the TCP stream laid out in order, as shown in the following Figure.



If different TCP connections appeared in the original capture, you will now only see one. If you want to re-view them all you must delete the display filter that was automatically added when selecting the "Follow TCP Stream" option. Press the "X" button to delete it. The following figure will help you.



Exercise 4: Capture HTTP traffic (port 80).

Configure the Wireshark to capture port 80 traffic using a capture filter ("port 80" syntax). Start the capture and using the browser Firefox get the web page in:

http://www.upv.es/

Using the "Follow TCP Stream" option, analyse the first HTTP request / response exchange of the captured traffic. You should have obtained from the server the response "HTTP/1.1 200 OK", if not, check the URL that you entered in the browser (you must include the protocol field of the URL).

If you have to re-capture the web page before you must empty the browser cache.

Questions:

Question 4.1: Does your browser use HTTP 1.0 or 1.1? What version of HTTP is the server using?

Question 4.2: What are the languages that your browser indicates to the server you prefer?

Question 4.3: What is the IP address of your computer? And the IP address of the server? (IP addresses can be viewed in the packet list pane, labeled as "`Source`" and "`Destination`").

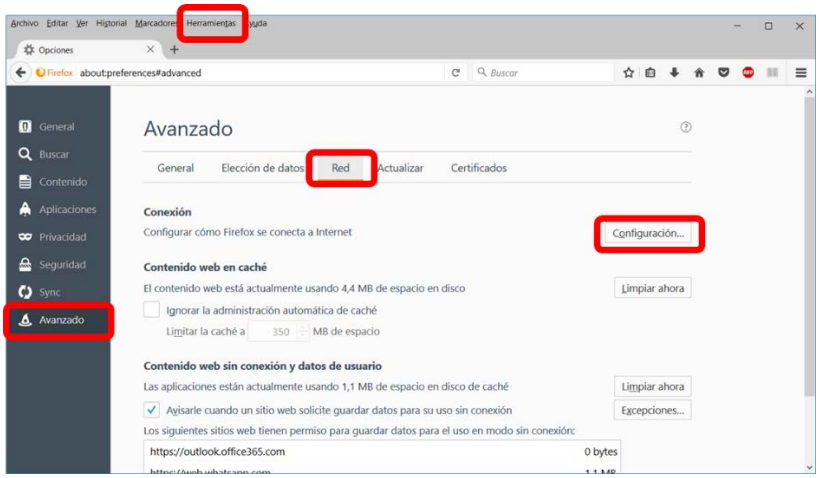
Question 4.4.: Analyzes the "`Connection`" headers of client and server, are they using persistent or non-persistent connections?

Question 4.5: Does the "`Transfer-Encoding`" header appear in the response? What is its value? What does that method of transfer mean?

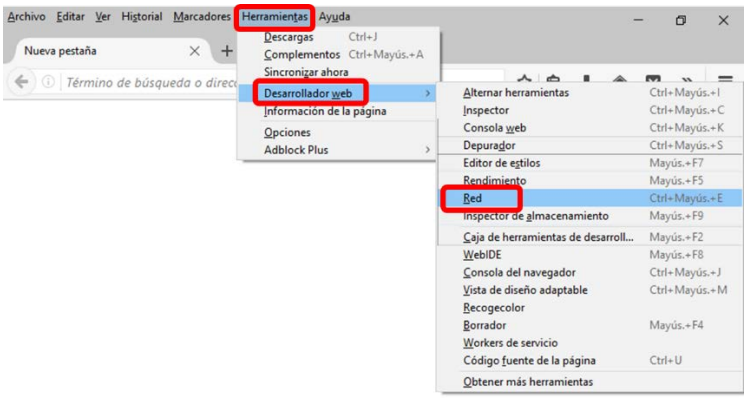
Question 4.6: Why does not the "`Content-Length`" header appear?

4 Second part. Advanced HTTP Interactions

In the following exercises we will use the browser itself as an analysis tool. To do this, we will launch the Firefox browser first. To make sure we are browsing without intermediate proxies that can introduce modifications in some headers, we will access "Tools"> "Options"> "Advanced"> "Network"> "Configuration" and select "No Proxy"



Next, we will move through the submenus of "Tools"> "Web Developer"> "Network" (or directly by pressing "Ctr + Shift + E"). A frame will be opened in the right part of the window.

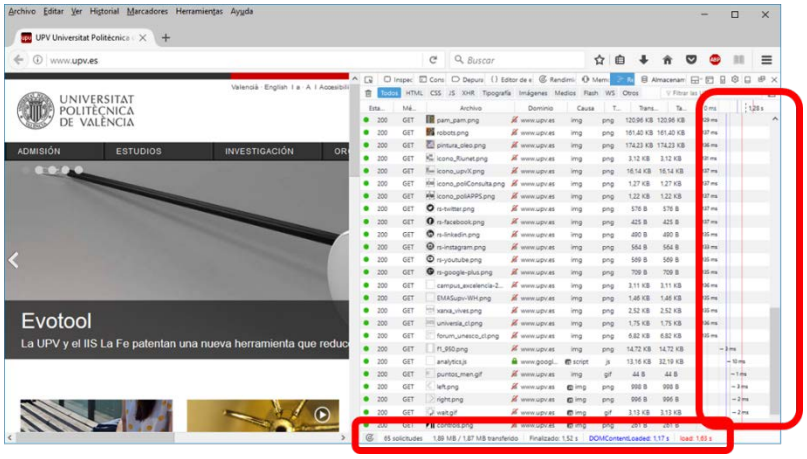


We will delete the data of the cache (pressing "Ctr + Shift + Del") and then we will get the following URL:

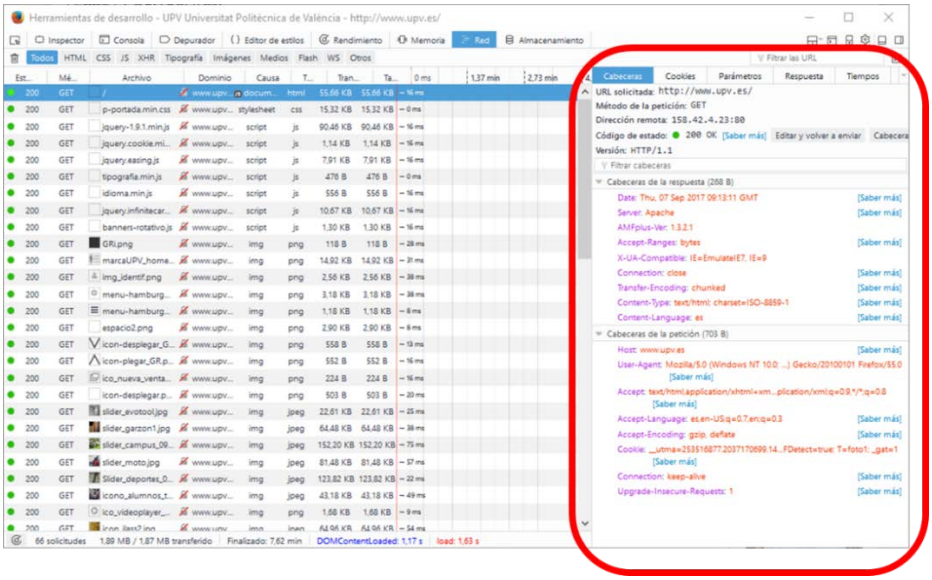
http://www.upv.es/

You should get something similar to what is shown in the following figure. The right highlighted part is called "the waterfall" and displays the way in which the

various resources that compose the document are obtained. Below, the status bar contains the following values: number of requests, total size / transferred size (note that some resources may be compressed prior to transfer, there may be cache hits, etc.), end time, and the times in which the *DOMContentLoaded* and *load* events occur (in the first one the HTML document was obtained and analysed and the interface is started, in the second, in addition, the rest of the resources were also obtained).



To analyse the HTTP headers of a request/response pair, simply select it and then display the "`Headers`" tab.



Let's see now how the browser's cache affects to requests made by the client. If you already have the resource in your cache, and you suspect it may have been modified since you obtained it, the client will send a conditional request, using the "`If-modified-since`" header. If the resource has not been modified on the

server, the response it sends will not be the usual one (200 OK). Let's look at the differences in both the client request and the server response.

Exercise 5: The cache.

Clear the cache and access the URL:

http://www.upv.es/

Check that all the responses have "200" as a status code. Write down the times and sizes that appear in the status bar of the tool. Without deleting the cache, reload the web page (Press F5 or Ctrl + R) and write down again the times and sizes.

Questions:

Question 5.1: Are there any differences in time and size?

Question 5.2: How many responses with codes other than 200 have you obtained? What do these codes mean?

Question 1.3: In the request headers, does a conditional header appear in addition to "`If-modified-since`"? What does it mean?

Question 1.4: Does any conditional header appear on the first request (HTML document request)? Why does it happen?

Exercise 6: Redirects (optional exercise).

Clear the cache and access the URL:

http://www.elpais.com/

Questions:

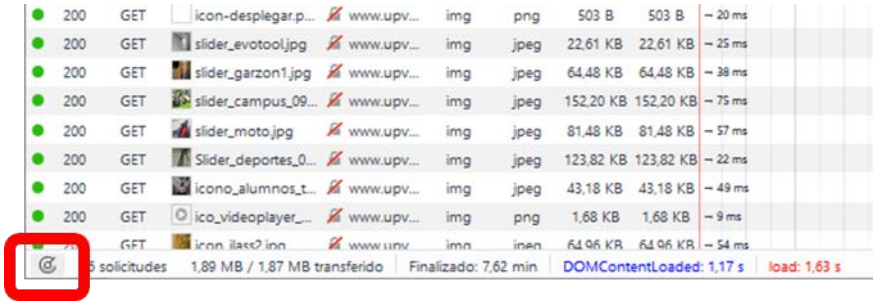
Question 6.1: When attempting to download the root resource the response code is "301". What does this code mean?

Question 6.2: What other "3XX" codes are obtained and what do they mean?

Question 6.3: Write down the values in the status bar and reload the document without deleting the cache. What are the differences?

Exercise 7: Optional exercise. Analysis of cache usage.

To perform a comparative analysis of cache usage, click on the icon to the left of the status bar.



The hit rate is the ratio of the number of resources that are valid in the cache to the number of total resources.

Questions:

Question 7.1: Calculate the obtained hit rate.

Question 7.2: Now calculate the success rate in bytes.

Question 7.3: Is there a difference between the two? What does each measure?