# Assignment 1: “Wireshark lab: Getting started + HTTP”

*Andreas Järvelä(andja235)*
*Sebastian Lindmark(sebli821)*

## Questions:

**1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?**

Our browser is running HTTP version 1.1, it can be displayed in the GET package info. Server is also running HTTP version 1.1, it can be displayed in the response packet info.

**2. What languages (if any) does your browser indicate that it can accept to the server? In the captured session, what other information (if any) does the browser provide the server with regarding the user/browser?**

The browser indicates that it can accept en-us as “Accept-Language”. It also gives the server information about what OS is running and browser. In this case: Mozilla/5.0 as browser and Windows as OS. Other information is also given like accepted encoding, charsets and formats.

**3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?**

The destination ip is displayed in the GET packet, in this case it is 128.119.245.12.
Server IP: 128.119.245.12:80
We are given the local ip address in wireshark.
Client IP: 192.168.1.102:4127

**4. What is the status code returned from the server to your browser?**

In the response packet we see that the server gave us status code 200, OK!

**5. When was the HTML file that you are retrieving last modified at the server?**

The response packet has a Last-Modified header saying: Tue, 23 Sep 2003 05:29:00 GMT

**6. How many bytes of content are being returned to your browser?**

The content length header in the response packet displays the number of bytes returned to the client. The length of the packet was 73 bytes. 73 bytes were returned.

**7. By inspecting the raw data in the "packet bytes" pane, do you see any http headers within the data that are not displayed in the "packet details" pane? If so, name one.**

No headers are shown in the raw data that is not shown in the display.

**8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?**

There's no IF-MODIFIED-SINCE in the first HTTP GET request.

*Andreas Järvelä(andja235)*
*Sebastian Lindmark(sebli821)*

**9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?**

The server returned status code 200 OK and 371 bytes. You can tell by looking in the HTTP.

**10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?**

If-Modified-Since header contains a date. The date is specified so that if the data has been modified we want a new version of it.

**11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.**

The response status code is 304 Not Modified. This means we already have the newest version of the file, therefore no bytes are returned.

**TASK B:**

Lets say we request an html file for the first time. The browser will store the downloaded file temporary and display the contents to the user. If there is a new http request  for that file, the client can pass along an “If-modified-since” header in the http GET request. The server can compare this http header with the last modified date of its file. If the file on the server has been update, the server will resend the file. Otherwise the server will return the 304 status code to tell the client that it hasn't been modified and the client can reuse its copy of the file.

**12. How many HTTP GET request messages were sent by your browser?**

One, we filter out to HTTP and see that there's only one HTTP GET request made for 4500 bytes.

**13. How many data-containing TCP segments were needed to carry the single HTTP response?**

The MTU for the packages returned are 1500 bytes, therefore we will get 4 packages.

**14. What is the status code and phrase associated with the response to the HTTP GET request?**

Once the acknowledgements are made and all the TCP packets containing the bytes requested are done. The server responds with the status code 200 , OK which also contains the remaining data. It's the last TCP packet combined with the HTTP response.

*Andreas Järvelä(andja235)*
*Sebastian Lindmark(sebli821)*

**15. Is there any HTTP header information in the transmitted data associated with TCP segmentation? For this question you may want to think about at what layer each protocol operates, and how the protocols at the different layers interoperate.**

It contains the sequence numbers, acts as an offset to which part of the packet data that has been sent.

**TASK C:**

When we request a file, the file might get fragmented on the way, depending on the capacity between the routers. It uses the sequence number to keep track of order of the packets. Each packet has a length and the sequence number is, for each packet, incremented by the length of the previous packet. This way the client can validate that it has received the data in correct order.

**16. How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?**

Three GET requests are made. The first one is for the HTML, which consists of two links to images, therefore we do 2 additional GET requests for those images.

HTML is located at: 128.119.245.12

And the images are located at 165.193.123.218(http://www.pearsonhighered.com/assets/hip/us/hip_us_pearsonhighered/images/pearson_logo.gif) respectively 134.241.6.82(http://manic.cs.umass.edu/~kurose/cover_5th_ed.jpg).

**17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.**

We download the two images serially, we can see that it's a pipelined and persistent connection. But the respective data from the images are given serially, first the ip ending with 218, when we get the HTTP 200 OK from that we start getting data from ip ending with 82.

**TASK D:** First we get the HTML file. When this file is downloaded the client sees the two images. The client requests the first image and the second image. This is pipelined and persistent connection so no new handshake has to be made. The server then sends the first image and then sends second image.

**20. What does the "Connection: close" and "Connection: Keep-alive" header field imply in HTTP protocol? When should one be used over the other?**

"Connection:close" is used with non persistent TCP
"Connection:Keep-alive" is used with persistent TCP
"Connection:Keep-alive" is faster and does not require a handshake to be made between each new request.

*Andreas Järvelä(andja235)*
*Sebastian Lindmark(sebli821)*

**TASK E:** When using a proxy we might need to consider if we want to use persistent or non-persistent connection with it. So now that we are about to code a proxy we will consider the advantages of a persistent connection with faster data transfer and less impact on the processors. Or if we want more scalability or a more simple architecture we can use the non-persistent option.