



# TDTS04 Computer Networks and Distributed Systems

## Wireshark Lab

Author

Maximilian Bragazzi Ihrén, [maxbr431@student.liu.se](mailto:maxbr431@student.liu.se)

Markus Lewin, [marle943@student.liu.se](mailto:marle943@student.liu.se)



VT1  
Version 1.0

## 1. Audit Descriptions

Ver.	Audit Description	Date
1.0	Written for report.	150126

## 2. Task A

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

Our browser runs HTTP 1.1, as seen by checking the first GET package, and checking the "HTTP/1.1" part.

The server is also running 1.1, as seen by the response from that package.

The HTTP version is appended in every HTTP package, but we saw it in the first two.

10	4.694850	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-1.html	HTTP/1.1
12	4.718993	128.119.245.12	192.168.1.102	HTTP	439	HTTP/1.1 200 OK	(text/html)

Figure 1

### 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 only accepts American and British English. You can see this by checking the Accept-Language tag under the HTTP message display.

The browser also provides the browser's name and version, the user's operating system, operating system version and operating system language. Also indicates what charsets the user is accepting, and what types of websites the browser supports.

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) Gecko/20021120 Netscape/7.01\r\n
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.7\r\n
Accept-Language: en-us,en;q=0.50\r\n
Accept-Encoding: gzip, deflate, compress;q=0.9\r\n
Accept-Charset: ISO-8859-1, utf-8;q=0.66,*;q=0.66\r\n
```

Figure 2

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

Our IP address is 192.168.1.102, as you can see in the "Source"-column for the packages sent by our computer.

The server's IP is 128.119.245.12, as you can see in the "Source"-column for the packages received by our computer.

Source	Destination	Protocol	Length	Info
192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-1.html HTTP/1.1
128.119.245.12	192.168.1.102	HTTP	439	HTTP/1.1 200 OK (text/html)

Figure 3

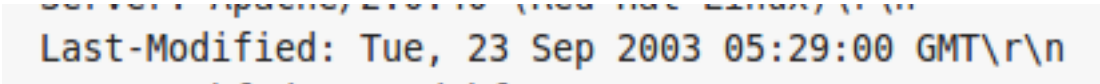
**4. What is the status code returned from the server to your browser?**

The server returns the status code "200 OK", which indicates that the webpage was gotten. This can be seen in the "Info"-column in the first package the server sent to us. See figure 3.

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

The file was last modified on Tuesday 23 September 2003 at 05:29:00 GMT.

You can tell this by checking the "Last-Modified"-tag under the HTTP message display in the "200 OK"-package.

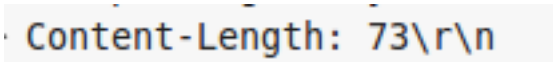


```
Last-Modified: Tue, 23 Sep 2003 05:29:00 GMT\r\n
```

Figure 4

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

The size of the recieved content is 73 bytes, which you can see by checking the "Content-Length"-tag in the "200 OK"-package (as that package sent the file).



```
Content-Length: 73\r\n
```

Figure 5

**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.**

There are no extra packets in the "packet bytes" pane.

**Discussion**

We realised that HTTP-packets are very easy to read due to them being in English and in ASCII-characters. This helps a lot in reading them in programs such as Wireshark. We also realised that there is a lot more information in each packet than we initially assumed. Of course, without a tool like Wireshark, reading these types of packets would become extremely difficult.

**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?**

No, there is no If-Modified-Since, since there is nothing in the browsers cache for the site.

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

Yes, the server sent the file. You can tell by checking the "Content-Type" and "Content-Length" tags. You can also check the data sent after the package header to see if it sent it all.

```
Accept-Ranges: bytes\r\n
Content-Length: 371\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
\r\n
```

Figure 6

**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?**

Yes, and the data that follows contains the date the file currently stored in the cache was last modified.

```
Connection: keep-alive\r\n
If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n
If-None-Match: "1bfef-173-8f4ae900"\r\n
```

Figure 7

**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 server sent no data, since it has no "Content-Length" tag, and it sent a "304 Not Modified" message, which means that the server won't send a new version (since it was "Not Modified").

tt	Info
55	GET /ethereal-labs/lab2-2.html HTTP/1.1
39	HTTP/1.1 200 OK (text/html)
68	GET /ethereal-labs/lab2-2.html HTTP/1.1
43	HTTP/1.1 304 Not Modified

Figure 8

## Discussion

We discovered that if the browser has no cached version of the website it is trying to connect to, it will ask for it without a "If-Modified-Since"-tag (how would it know which point in time to ask for?). However, if the browser has a cached version of the website it will append the tag, so the server won't send a new copy and, as a result, spend less bandwidth.

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

One, as you can see if you filter the packages after HTTP.

Info
GET /ethereal-labs/lab2-3.html HTTP/1.1
HTTP/1.1 200 OK (text/html)

Figure 9

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

4 packages, as can be seen by checking the [x Reassembled TCP...]-tag in the server's HTTP response.

You can also check by simply counting the TCP packages sent by the server (do not count the packages sent by your computer).

However, one of the packages is just an acknowledgement of the GET-request, so the total is actually 3 data-carrying TCP packets.

Destination	Protocol	Length	Info
128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-3.html HTTP/1.1
192.168.1.102	TCP	60	http > 4272 [ACK] Seq=1 Ack=502 Win=6432 Len=0
192.168.1.102	<b>1</b> TCP	1514	[TCP segment of a reassembled PDU]
192.168.1.102	TCP <b>2</b>	1514	[TCP segment of a reassembled PDU]
128.119.245.12	TCP	54	4272 > http [ACK] Seq=502 Ack=2921 Win=64240 Len=0
192.168.1.102	<b>3</b> TCP	1514	[TCP segment of a reassembled PDU]
192.168.1.102	HTTP	490	HTTP/1.1 200 OK (text/html)

Figure 10

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

"200 OK", see figure 10.

**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.**

Yes, in the first data-containing TCP packet, you can see (in the "packet bytes" section) that all of the header for the HTTP-packet is there. It is only present in the first packet, though, since the TCP packets do not use that data at all.

```

10  ...:..f.. ..w....
10  .f.P.... .d....P.
12  . ....HT TP/1.1 2
15  00 OK..D ate: Tue
15  , 23 Sep 2003 05
16  :37:02 G MT..Serv
14  er: Apac he/2.0.4
18  0 (Red H at Linux
14  )..Last- Modified
10  : Tue, 2 3 Sep 20
1d  03 05:37 :01 GMT.
11  .ETag: " 1bff2-11
13  94-96813 940"..Ac

```

Figure 11

### Discussion

If a file is too large to send with one packet, the file is transmitted by x packages, each coded with a sequence number. These packages are combined into the HTTP response package in the application layer after all the packages are recieved.

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

There were three GET-requests, all of which send their requests to different IP-addresses. The addresses are:

128.119.245.12/ethereal-labs/lab2-4.html

165.193.123.218/catalog/images/pearson-logo-footer.gif

134.241.6.82/~kurose/cover.jpg

Destination	Protocol	Length	Info
<u>128.119.245.12</u>	HTTP	555	GET /ethereal-labs/lab2-4.html HTTP/1.1
192.168.1.102	HTTP	1057	HTTP/1.1 200 OK (text/html)
<u>165.193.123.218</u>	HTTP	625	GET /catalog/images/pearson-logo-footer.gif HTTP/1.1
<u>134.241.6.82</u>	HTTP	609	GET /~kurose/cover.jpg HTTP/1.1
192.168.1.102	HTTP	912	HTTP/1.1 200 OK (GIF89a)
192.168.1.102	HTTP	1096	HTTP/1.0 200 Document follows (JPEG JFIF image)

Figure 12

### 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.

The two images were downloaded in parallel. You can tell by checking if the images were acquired before the next image request was sent.

E.g., both the requests were sent before any of the answers were received.

See figure 12.

## Discussion

Objects can be gotten from several different servers, as long as the first server you connect to gives you an address to the next.

Files can also be gotten in several different ways, for example in parallel or one-by-one. Parallel is in most cases faster to do, but one-by-one is (arguably) safer.

### 18. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

"401 Authorization Required".

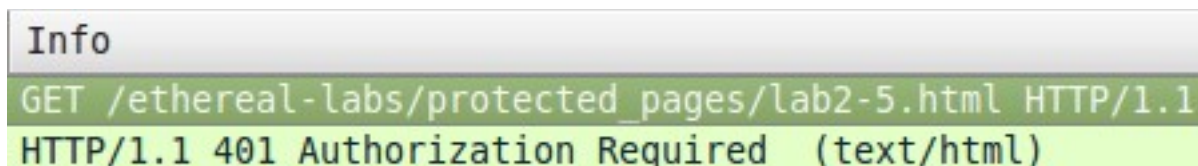


Figure 13

### 19. When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

Authorization: Basic ZXRoLXN0dWRlbmRzOm5ldHdvcmVz.

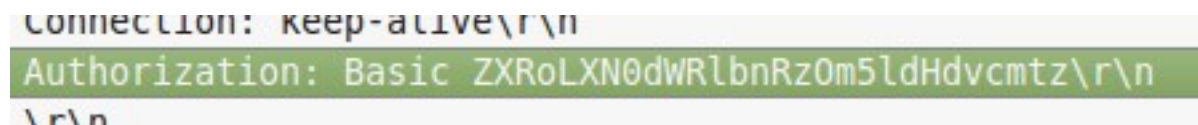


Figure 14

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

When obtaining multiple files, a connection will send a lot of extra packages if the connection is opened and closed for each file gotten. However, if you keep the connection open until packages haven't been recieved for a certain amount of time, the file transfers will go by much faster.

## Discussion

Connection: close is the first scenario, and should be used for simple ASCII websites (for example)

Connection: Keep-alive should be used for most modern websites, since you usually have to get more than one thing from them.