Valerio Colitta
Théo Foray

TDST06 – Computer Networks

Wireshark Lab : HTTP

Assignment 1

Valerio Colitta
Théo Foray

1. The basic HTTP GET/response interaction:

Practical Questions :

1. **Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?**
   Both my computer and the server are running HTTP version 1.1, as it can be seen for example in the lines below:
   Get request: (file Task A - Http Get)
   `Request Version: HTTP/1.1`
   Response: (file Task A - Response)
   `Response Version: HTTP/1.1`
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?**
   My browser indicates that it accepts French and American English, as shown (file Task A - Http get.
   `Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3\r\n`
   Moreover, the browser provides information about which browser and which OS I'm using to the server. (file Task A - Http get):
   `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0\r\n`
3. **What is the IP address of your computer? Of the gaia.cs.umass.edu server?**
   My IP Address is the 192.168.1.9 and the server's address is 128.119.245.12. (file Task A - Http get)
   `192.168.1.9 128.119.245.12`
   The source being my computer, and the destination the server since I request the page that the server is holding.
4. **What is the status code returned from the server to your browser?**
   The status code returned is 200, which means the request has been successful. (file Task A - Response).
   `Status Code: 200`
5. **When was the HTML file that you are retrieving last modified at the server?**
   The HTML file retrieved has been modified on Tuesday the 4th of September at 05:59:01 GMT. (file Task A – Response)
   `Last-Modified: Tue, 04 Sep 2018 05:59:01 GMT`
6. **How many bytes of content are being returned to your browser?**
   128 bytes of content have been returned to my browser. (file Task A - Response)
   `Content-Length: 128`
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 is no other header in the raw Data, because Wireshark already displays everything in the Details Pane.

Paragraph:

In this section, it has been shown that when two host (a client and a server) interact, they exchange information to be identified which permits the server to give an adapted response to the client (such as the Language of the information, the client OS).

Valerio Colitta
Théo Foray

2. The HTTP CONDITIONAL GET/response interaction

Practice Questions:

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?**
   We cannot see the field IF-MODIFIED-SINCE. (file Task B – Http Get 1)
9. **Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?**
   In the first GET the server responded with the text. In fact, it can be found in the Line-based text data field). (file Task B – Response 1)
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?**
    I see a "IF-MODIFIED-SINCE:" line in the second Http Get request. (file Task B – Http Get 1)
    `If-Modified-Since: Thu, 06 Sep 2018 05:59:01 GMT`
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 HTTP Status Code responded is 304, and the phrase associated is Not Modified. (file Task B – Response 2)
    `Status Code: 304`
    `[Status Code Description: Not Modified]`
    `Response Phrase: Not Modified`
    The server did not return explicitly the content of the file since there is no `Line-based text data: text/html (10 lines)` line in this response, even if there was on the first response. It has happened because when the second request has been done, it has been noticed the page wasn't modified, and since it has cached it, there is no need to ask for the content again.

Paragraph:

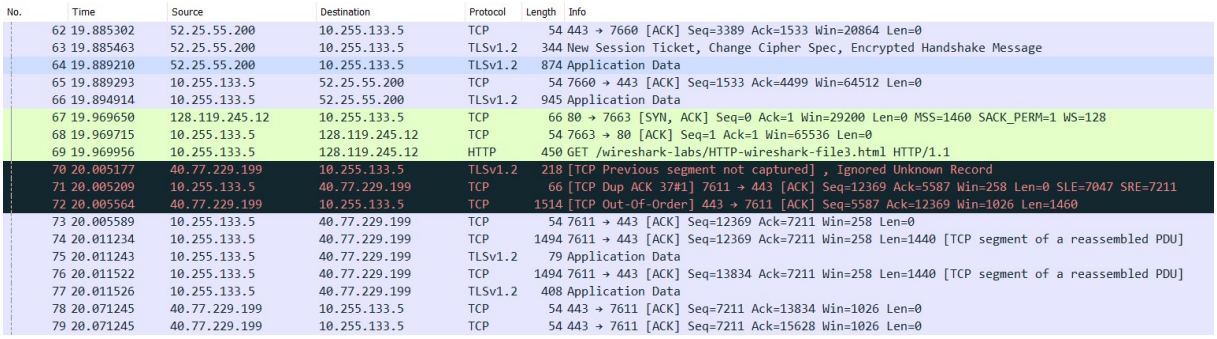
The IF-MODIFIED-SINCE field should be included if the page is cached in the browser. This is useful because then, the server doesn't have to send explicitly the data to the client. It happens when the content hasn't been modified since the date specified in the field. It saves times and bandwidth which can be very interesting with big files.

3. Retrieving long documents:

Practice Questions:

12. **How many HTTP GET request messages were sent by your browser?**
    Only one HTTP GET request was sent by my browser.

Valerio Colitta
Théo Foray



No.	Time	Source	Destination	Protocol	Length	Info
62	19.885302	52.25.55.200	10.255.133.5	TCP	54	443 → 7660 [ACK] Seq=3389 Ack=1533 Win=20864 Len=0
63	19.885463	52.25.55.200	10.255.133.5	TLSv1.2	344	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
64	19.889210	52.25.55.200	10.255.133.5	TLSv1.2	874	Application Data
65	19.889293	10.255.133.5	52.25.55.200	TCP	54	7660 → 443 [ACK] Seq=1533 Ack=4499 Win=64512 Len=0
66	19.894914	10.255.133.5	52.25.55.200	TLSv1.2	945	Application Data
67	19.969650	128.119.245.12	10.255.133.5	TCP	66	80 → 7663 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
68	19.969715	10.255.133.5	128.119.245.12	TCP	54	7663 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
69	19.969956	10.255.133.5	128.119.245.12	HTTP	450	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
70	20.005177	40.77.229.199	10.255.133.5	TLSv1.2	218	[TCP Previous segment not captured] , Ignored Unknown Record
71	20.005209	10.255.133.5	40.77.229.199	TCP	66	[TCP Dup ACK 37#1] 7611 → 443 [ACK] Seq=12369 Ack=5587 Win=258 Len=0 SLE=7047 SRE=7211
72	20.005564	40.77.229.199	10.255.133.5	TCP	1514	[TCP Out-Of-Order] 443 → 7611 [ACK] Seq=5587 Ack=12369 Win=1026 Len=1460
73	20.005589	10.255.133.5	40.77.229.199	TCP	54	7611 → 443 [ACK] Seq=12369 Ack=7211 Win=258 Len=0
74	20.011234	10.255.133.5	40.77.229.199	TCP	1494	7611 → 443 [ACK] Seq=12369 Ack=7211 Win=258 Len=1440 [TCP segment of a reassembled PDU]
75	20.011243	10.255.133.5	40.77.229.199	TLSv1.2	79	Application Data
76	20.011522	10.255.133.5	40.77.229.199	TCP	1494	7611 → 443 [ACK] Seq=13834 Ack=7211 Win=258 Len=1440 [TCP segment of a reassembled PDU]
77	20.011526	10.255.133.5	40.77.229.199	TLSv1.2	408	Application Data
78	20.071245	40.77.229.199	10.255.133.5	TCP	54	443 → 7611 [ACK] Seq=7211 Ack=13834 Win=1026 Len=0
79	20.071245	40.77.229.199	10.255.133.5	TCP	54	443 → 7611 [ACK] Seq=7211 Ack=15628 Win=1026 Len=0

13. **How many data-containing TCP segments were needed to carry the single HTTP response?**
    To carry the single HTTP response 4 TCP Segments were needed (3 single segments + 1 in the HTTP response). (file Task C - Trace)
14. **What is the status code and phrase associated with the response to the HTTP GET request?**
    It has status code 200, with phrase OK. (file Task C - Trace)
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.**
    No there isn't, and it shouldn't. TCP layer operates one layer below HTTP. This means that during encapsulation, the newly generated TCP packet won't pass through the application layer and won't get an HTTP header.

Paragraph:

The browser sends only one HTTP Request while there are 4 TCP requests because the file exceeds the MTU (1500 bytes). Thus, it must be split into chunks which in this case are 1460 bytes long.

4. HTML Documents with Embedded Objects:

Practice Questions:

16. **How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?**
    Four HTTP GET requests were done by the browser.
    The requests were respectively sent to http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html, http://gaia.cs.umass.edu/pearson.png, http://manic.cs.umass.edu/~kurose/cover_5th_ed.jpg, http://caite.cs.umass.edu/~kurose/cover_5th_ed.jpg
    This is checkable in the file Task D – trace.
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.**
    By looking at the trace given, we see we don't need to get the response to the first request to throw a new one, so there are two GET in a row, and then we get the two responses. Hence, it is not serially but parallel.
    Here, we used the traces provided for this assignment.

Paragraph:

Since there are many resources on a single page, the browser must send one HTTP request for each of those. If HTTP Persistence is used, it may happen that if two resources are on the same server, they can be sent over the same TCP response with wasting time and avoiding unnecessary communication.

Valerio Colitta
Théo Foray

5. HTTP Authentication:

Practice Questions:

18. **What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?**
    The first request gets a 401 code and an Unauthorized message, because of the need for a password, whenever a password is inserted a new request is sent. (file HTTP Authentication – Trace).
19. **When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?**
    In the second GET request, it sends a new field called Authorization, with the credentials in plain-text since the request is a simple HTTP request and not an encrypted one. (file HTTP Authentication – Trace).

6. Preparation for Assignment 2

20. **What does the "Connection: close" and "Connection: Keep-alive" header field imply in HTTP protocol? When should one be used over the other?**
    The Connection header indicates whether the TCP connection has to be persistent or closed immediately after the response for the first resource (the webpage file). With keep-alive, we use the same TCP connection over and over for all the resources on the page, otherwise with the close field we would have to restart another TCP connection for every other resource (style.css, script.js,…) resulting in a waste of time. So we should use the keep-alive field whenever we have a page with a lot of resources to be downloaded.