TITLE

Rajdeep Gill 7934493

COURSE CODE SECTION

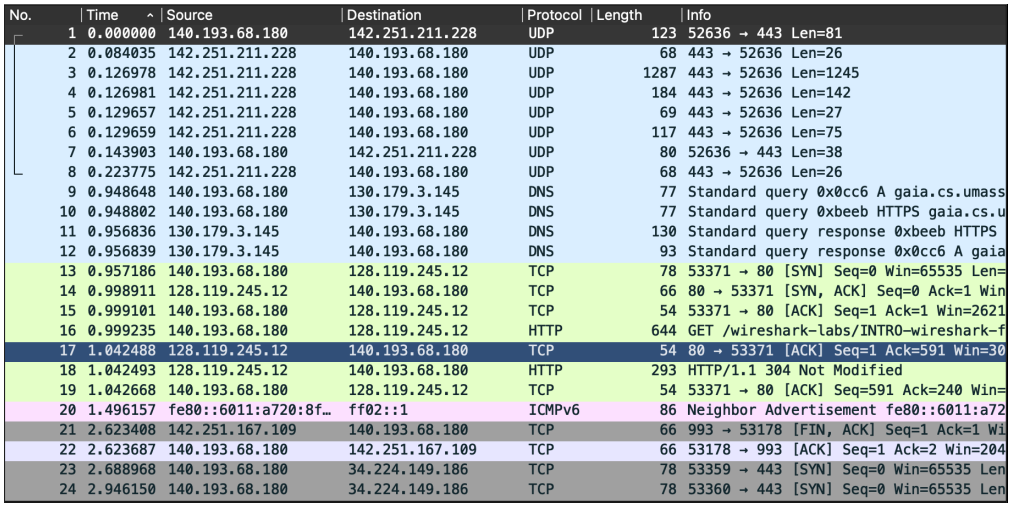
January 24, 2025

Contents

1 Wireshark Lab: Getting Started

1. The different protocols that appear in the protocol column in the unfiltered packet-listing window are:
   - UDP, TCP, DNS, HTTP, ICMPv6, TLSv1.2, LLC, TLSv1.3,

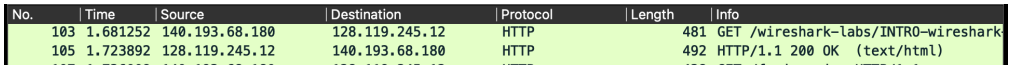
   Some of these packets can be see in Figure 1.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	140.193.68.180	142.251.211.228	UDP	123	52636 → 443 Len=81
2	0.084035	142.251.211.228	140.193.68.180	UDP	68	443 → 52636 Len=26
3	0.126978	142.251.211.228	140.193.68.180	UDP	1287	443 → 52636 Len=1245
4	0.126981	142.251.211.228	140.193.68.180	UDP	184	443 → 52636 Len=142
5	0.129657	142.251.211.228	140.193.68.180	UDP	69	443 → 52636 Len=27
6	0.129659	142.251.211.228	140.193.68.180	UDP	117	443 → 52636 Len=75
7	0.143903	140.193.68.180	142.251.211.228	UDP	80	52636 → 443 Len=38
8	0.223775	142.251.211.228	140.193.68.180	UDP	68	443 → 52636 Len=26
9	0.948648	140.193.68.180	130.179.3.145	DNS	77	Standard query 0x0cc6 A gaia.cs.umass
10	0.948802	140.193.68.180	130.179.3.145	DNS	77	Standard query 0xbeeb HTTPS gaia.cs.u
11	0.956836	130.179.3.145	140.193.68.180	DNS	130	Standard query response 0xbeeb HTTPS
12	0.956839	130.179.3.145	140.193.68.180	DNS	93	Standard query response 0x0cc6 A gaia
13	0.957186	140.193.68.180	128.119.245.12	TCP	78	53371 → 80 [SYN] Seq=0 Win=65535 Len=
14	0.998911	128.119.245.12	140.193.68.180	TCP	66	80 → 53371 [SYN, ACK] Seq=0 Ack=1 Win
15	0.999101	140.193.68.180	128.119.245.12	TCP	54	53371 → 80 [ACK] Seq=1 Ack=1 Win=2621
16	0.999235	140.193.68.180	128.119.245.12	HTTP	644	GET /wireshark-labs/INTRO-wireshark-f
17	1.042488	128.119.245.12	140.193.68.180	TCP	54	80 → 53371 [ACK] Seq=1 Ack=591 Win=30
18	1.042493	128.119.245.12	140.193.68.180	HTTP	293	HTTP/1.1 304 Not Modified
19	1.042668	140.193.68.180	128.119.245.12	TCP	54	53371 → 80 [ACK] Seq=591 Ack=240 Win=
20	1.496157	fe80::6011:a720:8f…	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::6011:a72
21	2.623408	142.251.167.109	140.193.68.180	TCP	66	993 → 53178 [FIN, ACK] Seq=1 Ack=1 Wi
22	2.623687	140.193.68.180	142.251.167.109	TCP	66	53178 → 993 [ACK] Seq=1 Ack=2 Win=204
23	2.688968	140.193.68.180	34.224.149.186	TCP	78	53359 → 443 [SYN] Seq=0 Win=65535 Len
24	2.946150	140.193.68.180	34.224.149.186	TCP	78	53360 → 443 [SYN] Seq=0 Win=65535 Len

Figure 1: Packet-listing window

2. The time taken for when the HTTP GET message was sent to when the HTTP OK reply was recieved was 0.04264 seconds. This was calculated by subtracting the time the HTTP GET message was sent from the time the HTTP OK reply was recieved. This can be seen in Figure 2.



No.	Time	Source	Destination	Protocol	Length	Info
103	1.681252	140.193.68.180	128.119.245.12	HTTP	481	GET /wireshark-labs/INTRO-wireshark
105	1.723892	128.119.245.12	140.193.68.180	HTTP	492	HTTP/1.1 200 OK (text/html)

Figure 2: Time taken for HTTP GET message to HTTP OK reply

3. The Internet address of the gaia.cs.umass.edu is 128.119.245.12 and the Internet address of my computer is 140.193.68.100. These addresses are sseen in Figure 2.

4. Making 20 different requests and finding the delay between the HTTP GET message and the HTTP OK reply, the average delay was 0.08513 seconds. The plot of the 20 different requests can be seen in Figure 3.

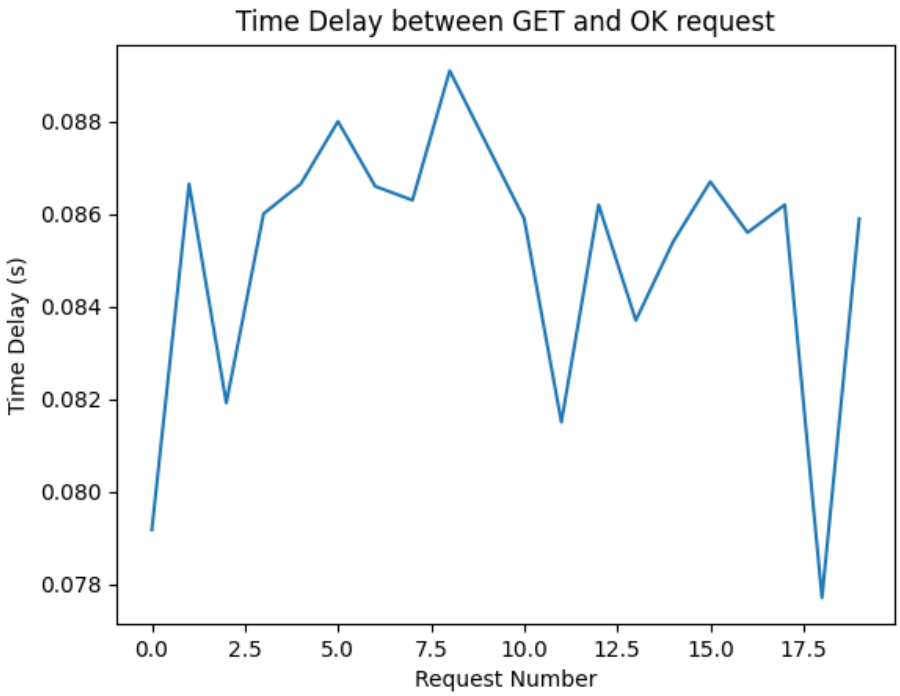


Figure 3: 20 different requests

2 Wireshark Lab: HTTP

1. The browser is running HTTP version 1.1 and so is the server. This can be deduced from the GET and OK requests both having HTTP/1.1 in the info column.
2. Todo
3. The IP address of my computer is 140.193.68.100 and the gaia.cs.umass.edu server is 128.119.245.12. This can be seen in the GET request and OK response.
4. The status code returned by the server is 200. This can be seen in the info column of the OK request.
5. The file was last modified at: . This can be seen in OK response from the server in the Hypertext Transfer Protocol section.
6. 576 bytes of content was returned to the browser, and can be seen in the frame section of the GET response.
7. perhaps

3 The HTTP CONDITIONAL GET/response interaction

8. The first HTTP GET request does not have an If-Modified-Since header.
9. The server does return the contents of the file as we can see the content length of is 371.

10. Inspecting the second HTTP GET request, we do see an If-Modified-Since header. The info followed by the header is "If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT\r\n". This can be seen in the Hypertext Transfer Protocol section of the GET request.
11. The HTTP status code of is 304 with a response phrase of Not Modified. The server did not explicitly return the contents of the file as the browser already had the file cached. There is also no content length in the response.

4 Retrieving Long Documents

12. 1 HTTP GET request messages were sent by the browser.
13. 5 TCP segments were needed to carry the single HTTP response message.
14. The status code and phrase associated with the response to the HTTP GET request is 200 OK.
15. There are 3 packets with status lines stating a continuation.

5 HTML Documents with Embedded Objects

16. 3 HTTP GET request messages were sent by the browser. One for the html and the other two for the image. The internet address of the first request was: 128.119.245.12, the second request was: 165.193.123.218 and the third request was: 134.241.6.82.
17. The two images were done in parallel as the browser sent the request for the two images before any of the responses were received. In the packet listing window we see that the requests were both made before any of the responses were received.

6 HTTP Authentication

18. In the initial HTTP GET request, the server responseds with a 401 code and a message of Authorization Required.
19. The new field in the second HTTP GET request is the Authorization field. It is a basic authentication field with the value encoded in base64.

7 Additional Questions

1. In the TCP/IP stack, HTTP belongs in the application layer.
2. The underlying transport layer protocol used by TCP is IP.
3. The HTTP response for a successful request is 200 OK.
4. When a file exceeds the payload size of a single packet, the file is split into multiple TCP segments and are sent individually. The segments are then reassembled at the destination.
5. The components of the HTTP status line are the status code and the status phrase.
6. The encoding method used in HTTP authentication is base64.
7. Basic authentication is not secure as base64 can be easily decoded, allowing the information in the header to be read. If it contains sensitive information, it can be easily stolen.l