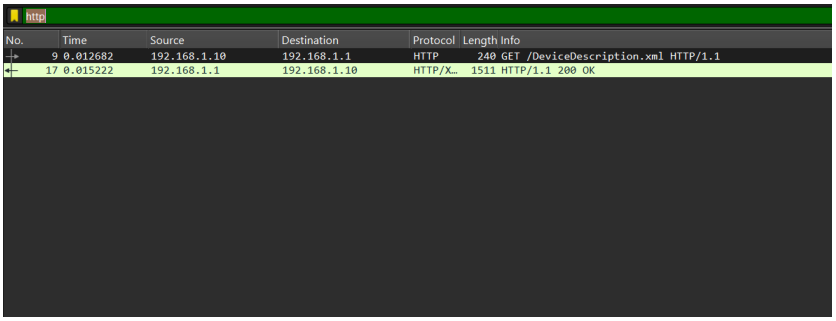
Experiment – 8

Aim: To capture and analyze HTTP traffic to understand how a basic website operation works, such as request and response between a client (browser) and a server.

Objective: Understand the HTTP protocol, including GET requests, response codes, and the data exchanged between the client and server.

Theory:
When you visit a website, your browser sends an HTTP request to the web server to fetch resources (HTML, images, scripts, etc.). The server responds with the requested data. Wireshark can capture these HTTP packets, allowing you to inspect the communication and understand the details of the request-response cycle.

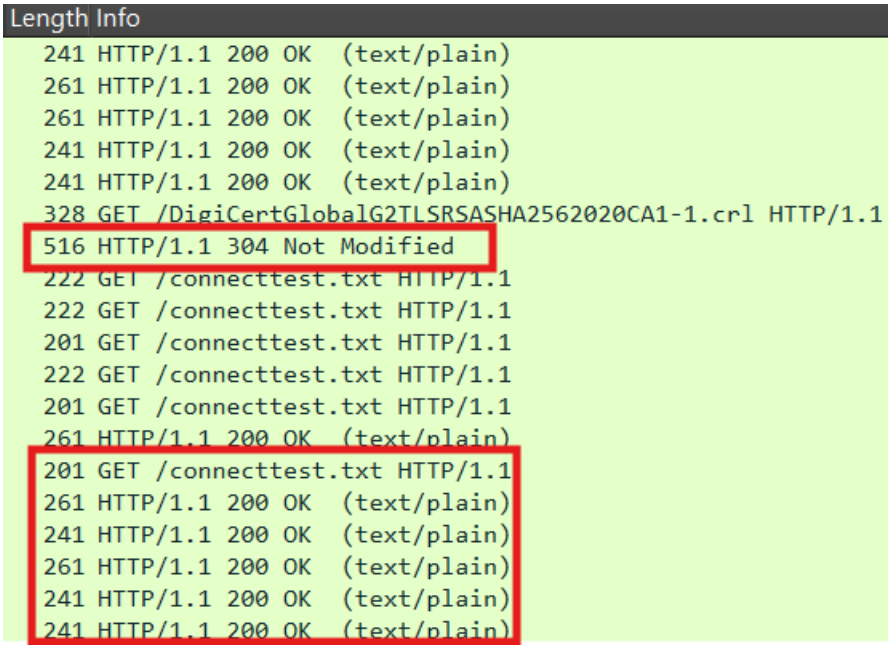
Used **Commands** in Wireshark:
1. Capture HTTP Traffic:
- Start capturing packets (Capture > Start).
- Use the filter http to show only HTTP traffic. This will help you focus on the communication between the client and the server.



2. Analyze HTTP Request:
- Look at the packets captured to find HTTP GET/POST requests. For example, you will see GET requests like GET /index.html HTTP/1.1.

No.	Time	Source	Destination	Protocol	Length	Info
2918	28.399576	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)
2924	28.418292	2401:4900:8898:e0bb…	2600:1417:6e::170f:…	HTTP	229	GET /connecttest.txt HTTP/1.1
2926	28.437916	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
2931	28.453713	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)
2936	28.454579	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
2941	28.463737	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)
2946	28.503987	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
3545	63.909038	2401:4900:8898:e0bb…	2600:1417:6e::170f:…	HTTP	229	GET /connecttest.txt HTTP/1.1
3548	63.927864	2401:4900:8898:e0bb…	2600:1417:6e::170f:…	HTTP	229	GET /connecttest.txt HTTP/1.1
3551	63.929772	192.168.1.10	23.76.157.113	HTTP	208	GET /connecttest.txt HTTP/1.1
3554	63.944287	2401:4900:8898:e0bb…	2600:1417:6e::170f:…	HTTP	229	GET /connecttest.txt HTTP/1.1
3557	63.948096	192.168.1.10	23.76.157.113	HTTP	208	GET /connecttest.txt HTTP/1.1
3559	63.950092	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
3565	63.954720	192.168.1.10	23.76.157.113	HTTP	208	GET /connecttest.txt HTTP/1.1
3567	63.982203	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)
3572	63.987797	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
3581	64.016226	2600:1417:6e::170f:…	2401:4900:8898:e0bb…	HTTP	261	HTTP/1.1 200 OK (text/plain)
3586	64.022190	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)
3591	64.038175	23.76.157.113	192.168.1.10	HTTP	241	HTTP/1.1 200 OK (text/plain)

3. HTTP Response:
- Find HTTP response packets, which will have status codes like 200 OK, 404 Not Found, etc. The response will include the body of the HTML or other resources.



4. Filter by Host:
- Use the filter http.host == "example.com" to see the HTTP traffic specifically for a particular website.

Conclusion: In this experiment, we successfully captured and analyzed HTTP traffic using Wireshark to understand the fundamental operation of a website. By focusing on the communication between a client (browser) and a server, we observed the **request-response cycle** of the HTTP protocol.