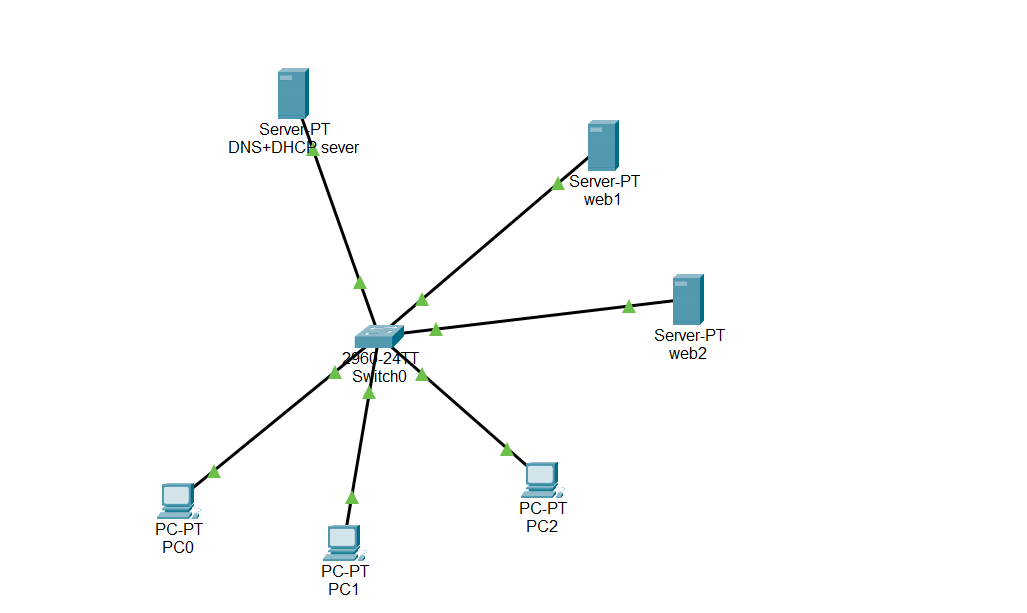
**CN LAB 4**

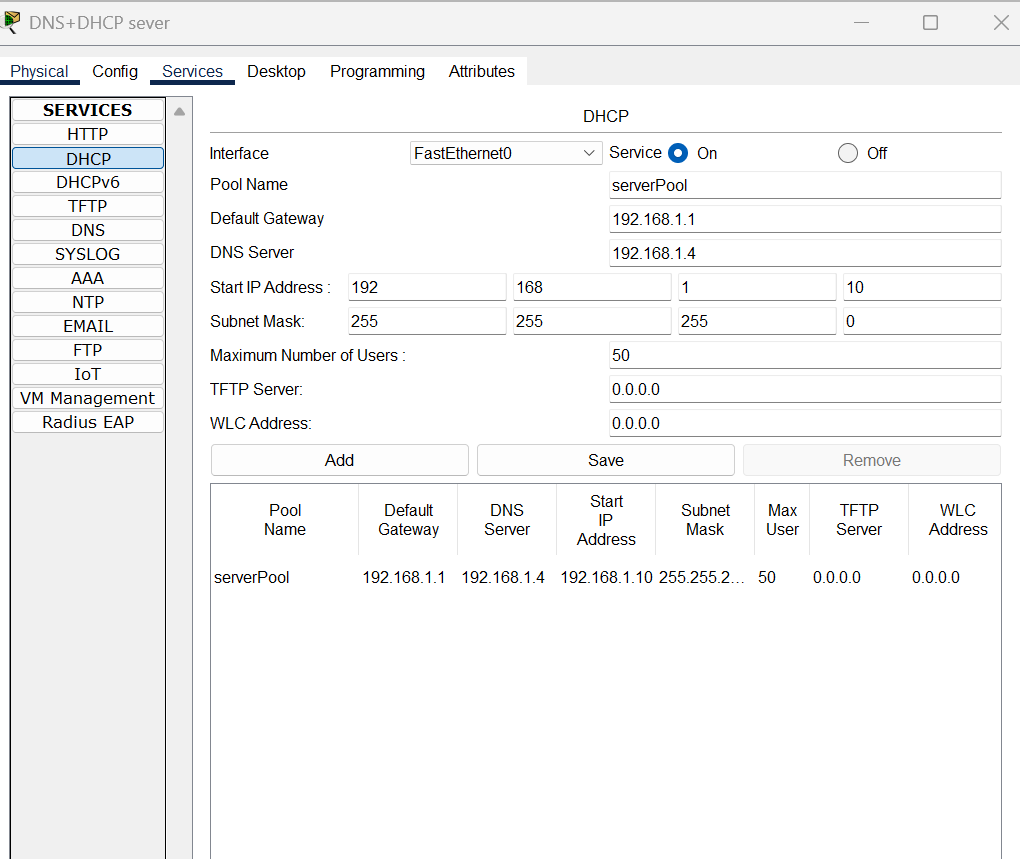
**23K-0594**

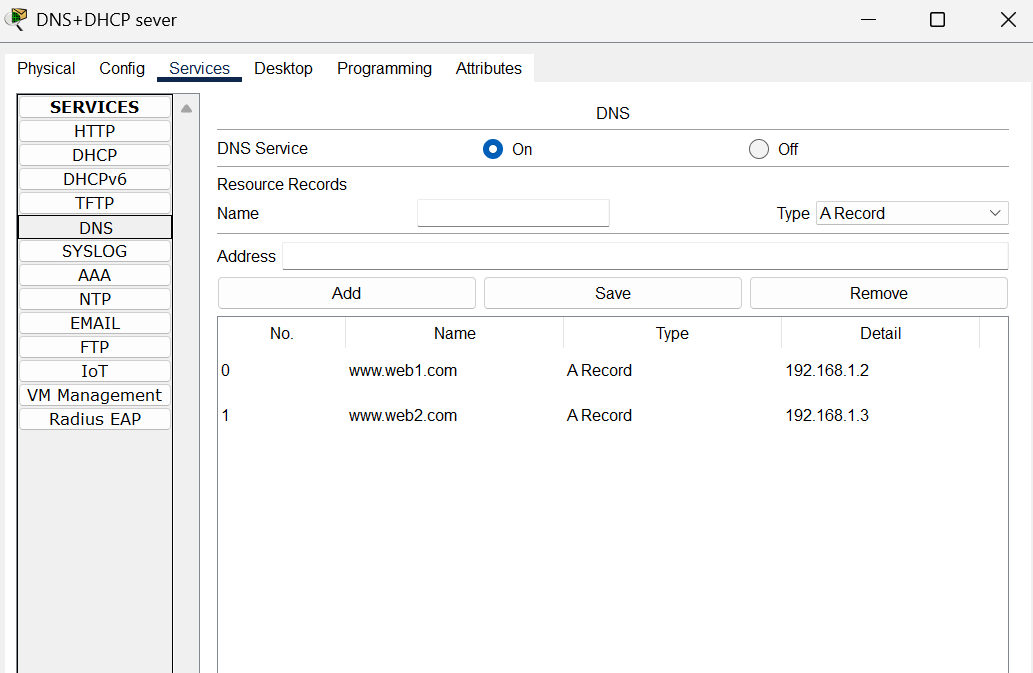
**TASK 1:**

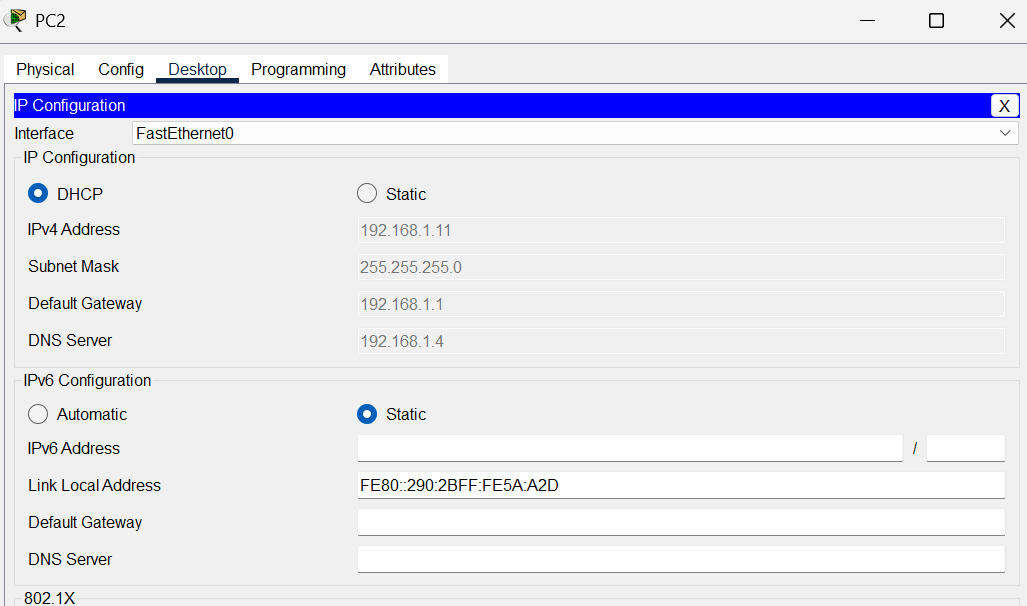
****

DNS+DHCP SERVER:

I made one server act as both **DNS and DHCP**. The **DHCP part** gives IPs to the PCs automatically (like 192.168.1.10+) so I don’t have to set them manually. The **DNS part** lets the PCs reach the web servers using names (like www.web1.com) instead of just IPs, and it also allows DNS packets to show up in simulation mode as required.

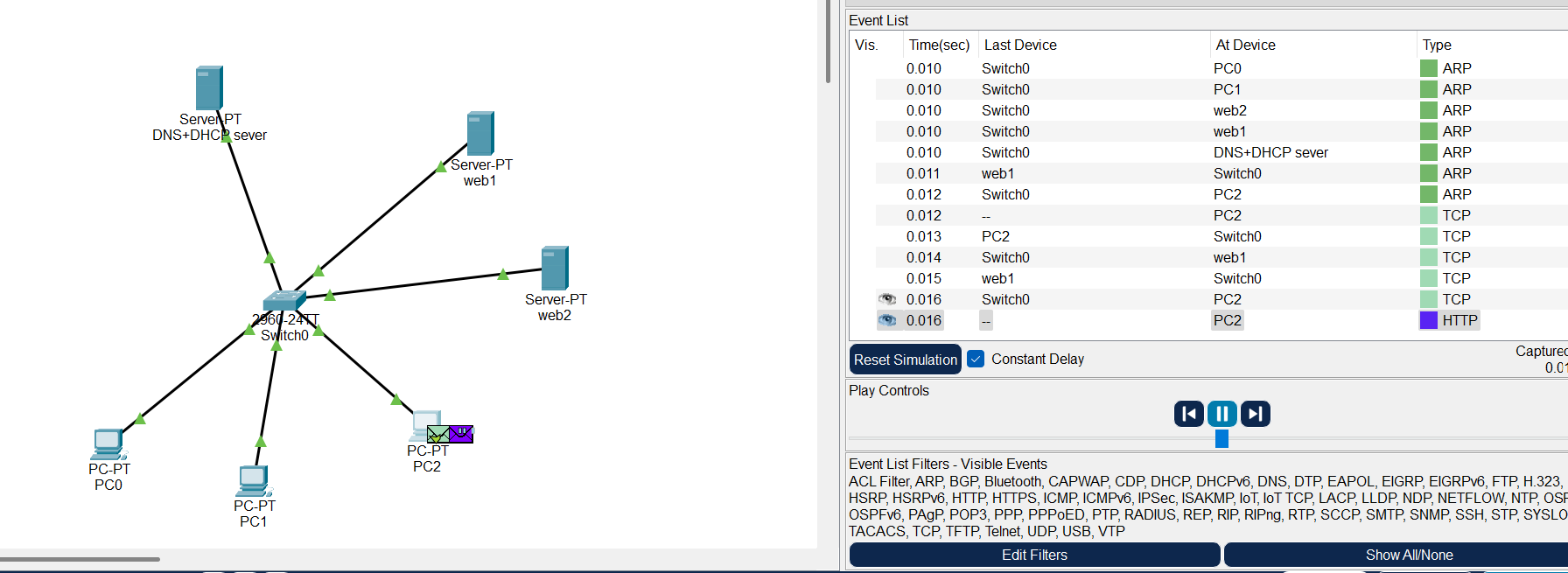
****

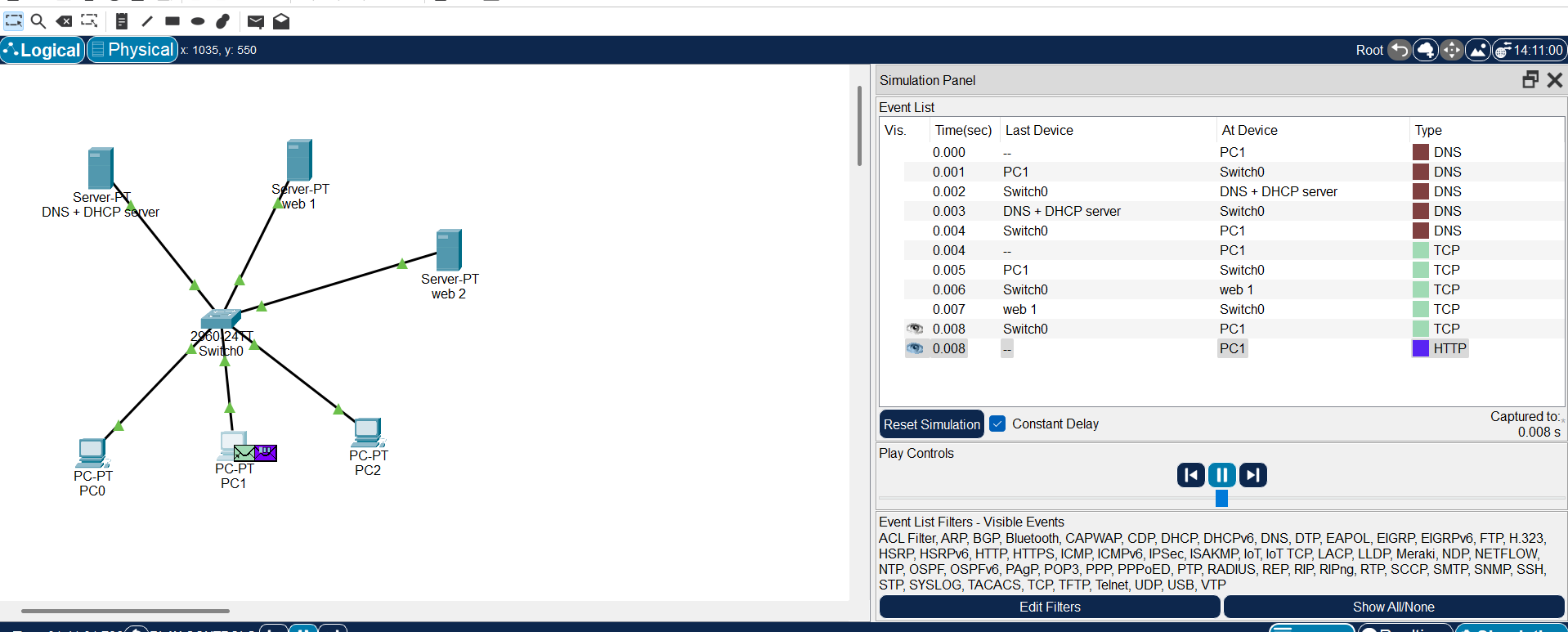
****

****

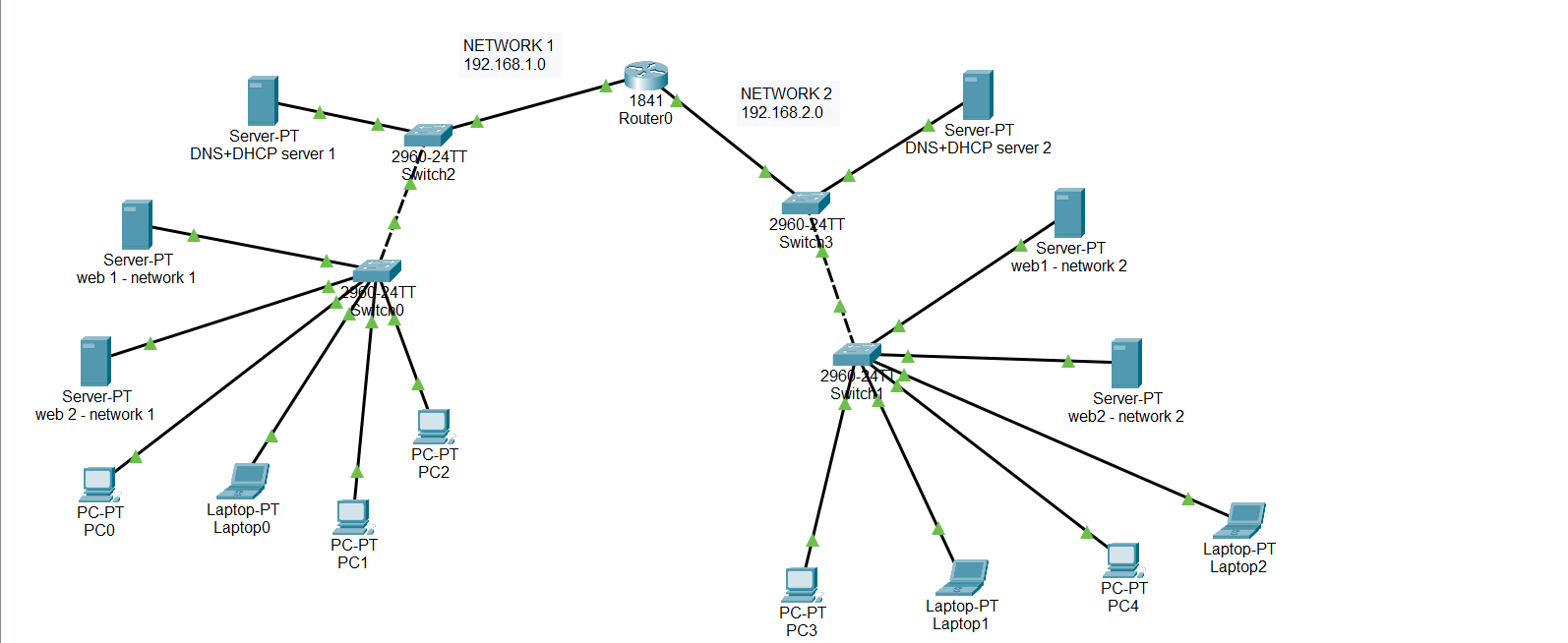
SIMULATION:



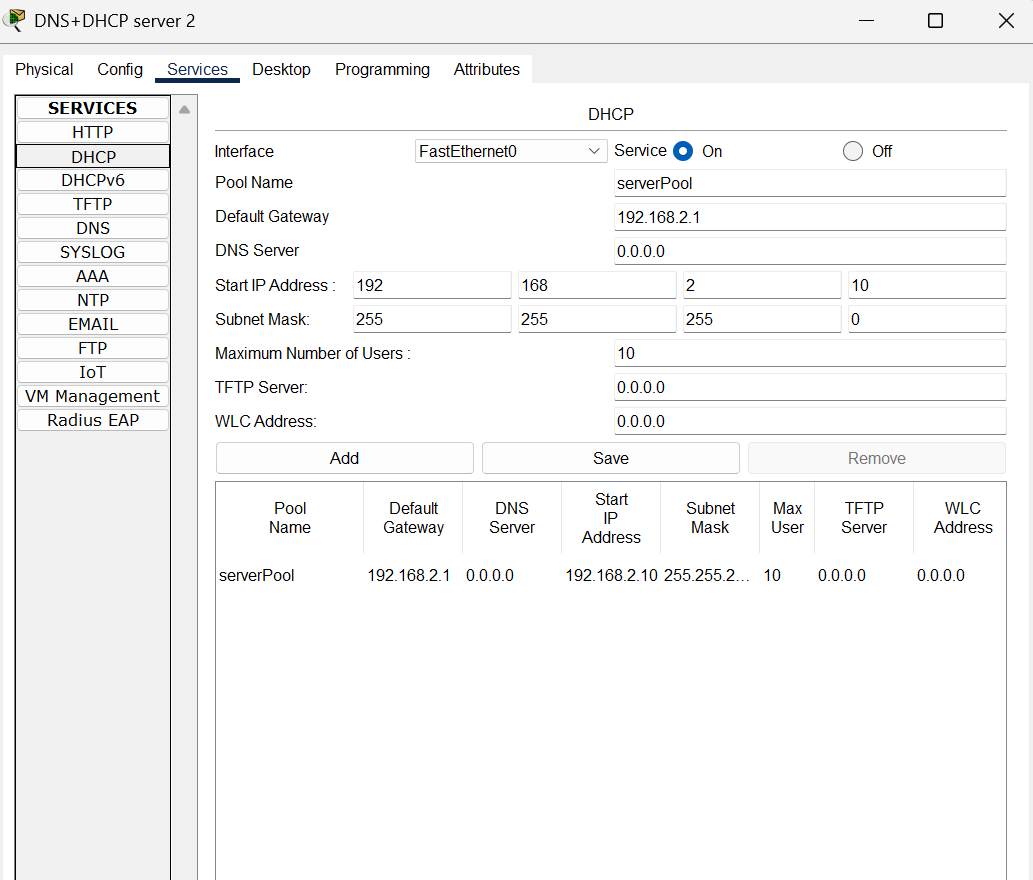


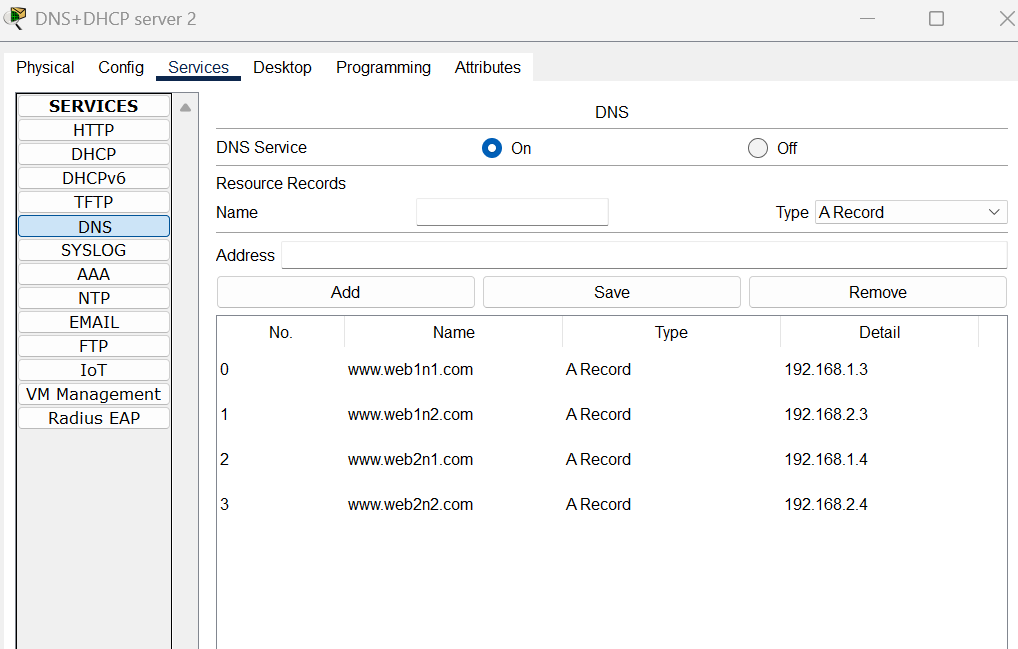


**TASK 2:**

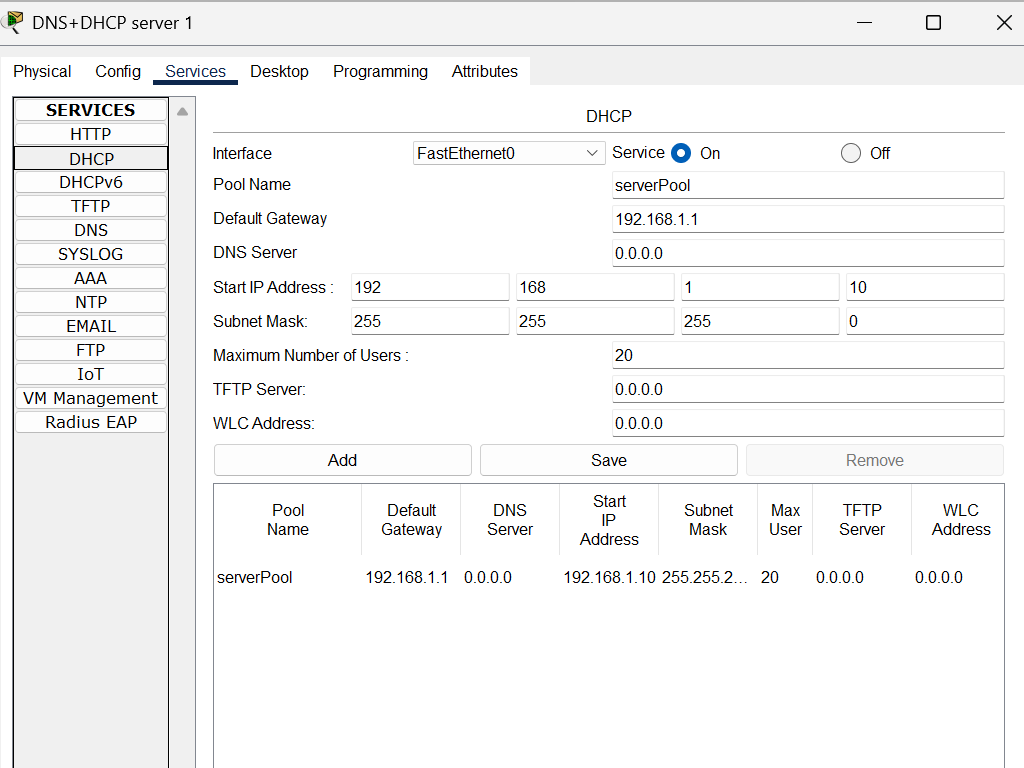
****

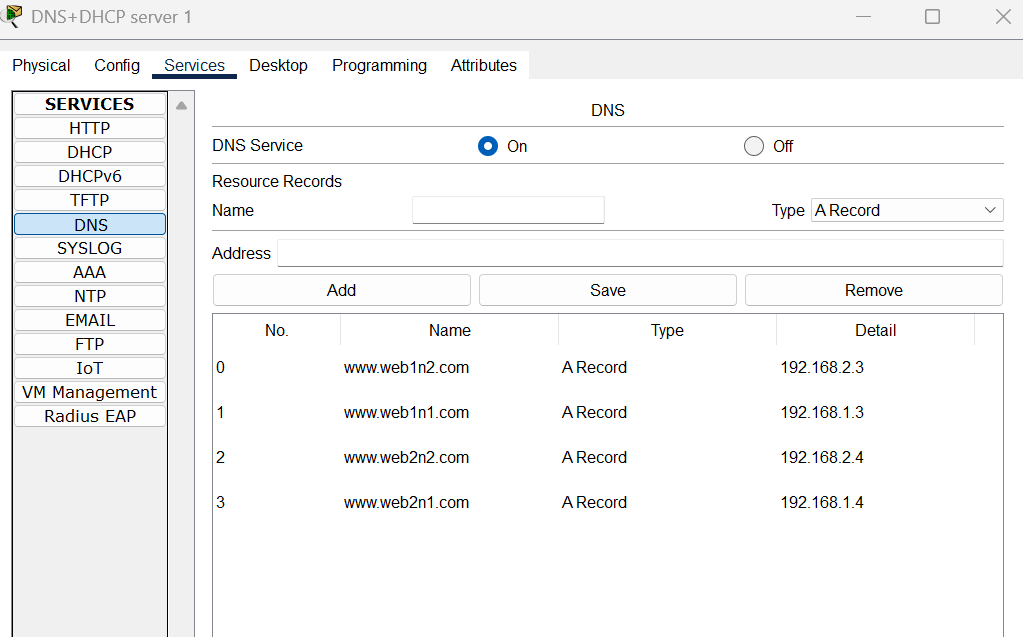
Network 2 – DNS+DHCP SERVER:

****

****

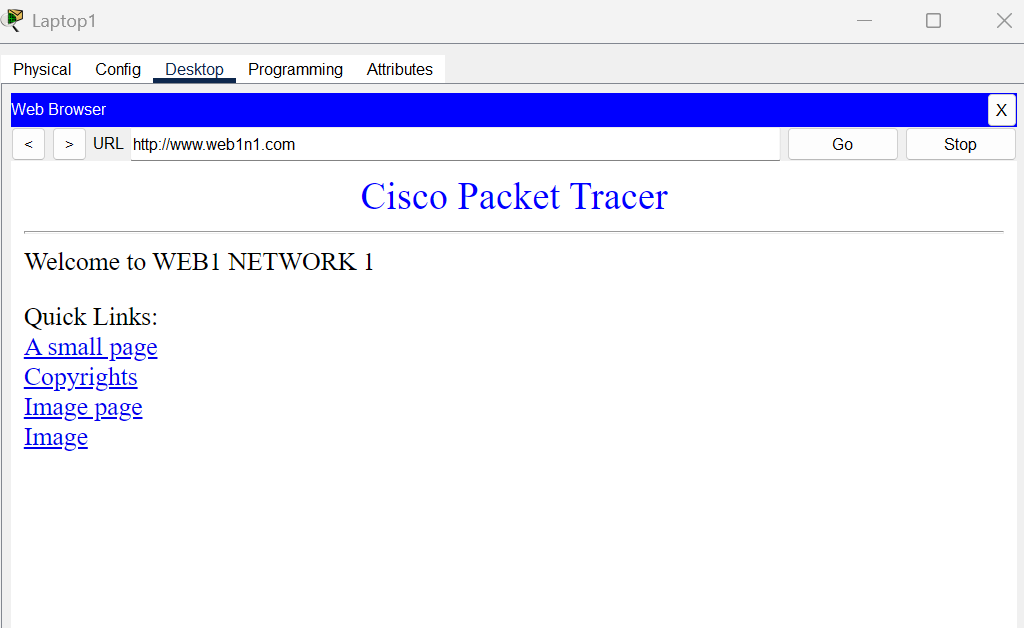
Network 1 – DNS+DHCP SERVER:

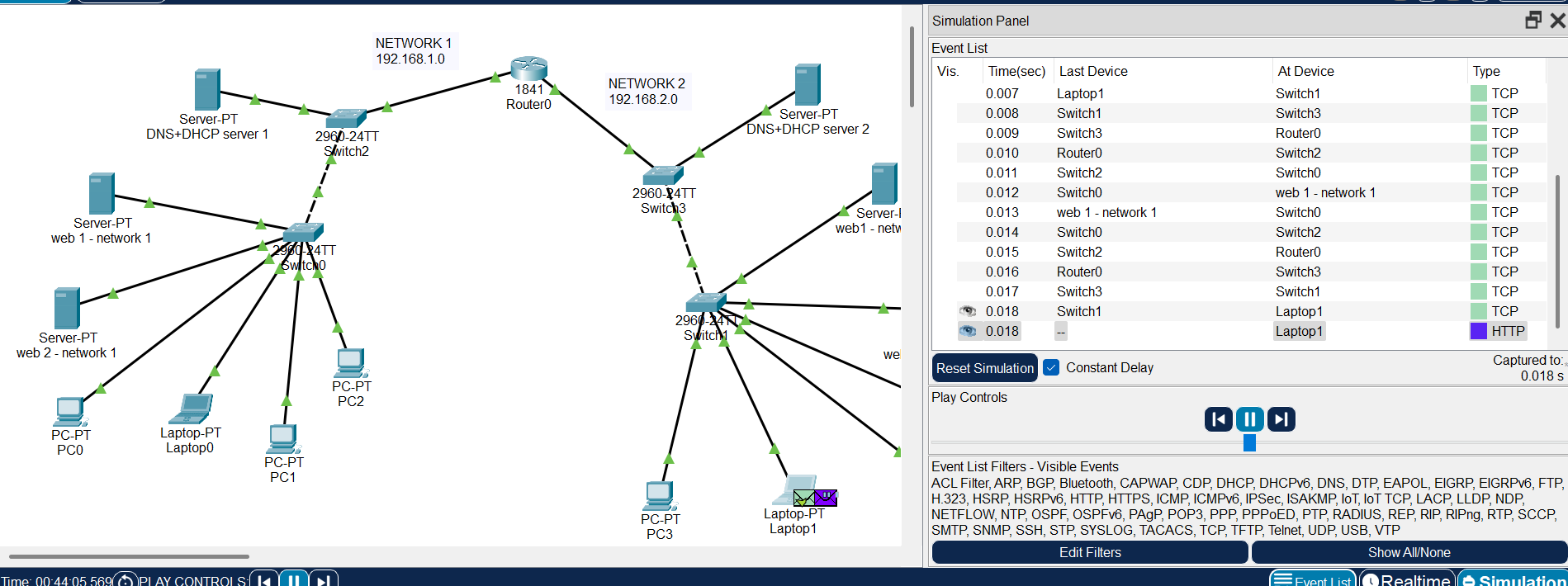
****

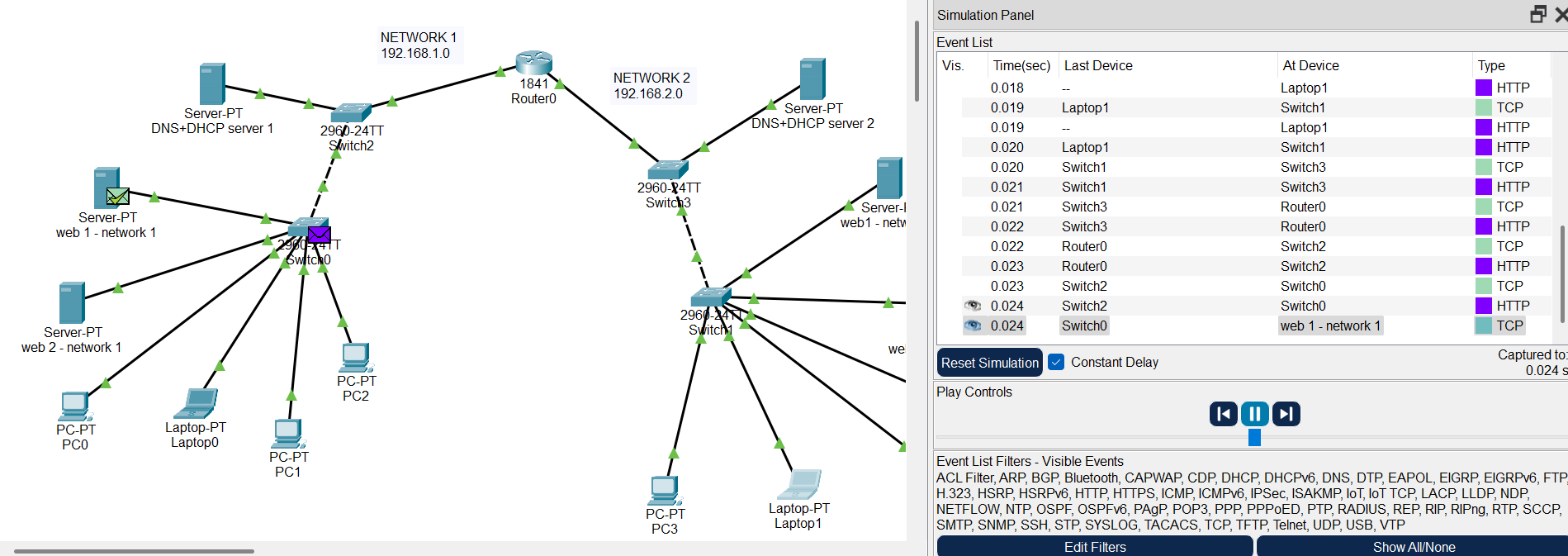
****

SIMULATION:

Laptop 1 in network 2 sends packet to web 1 in network 1.







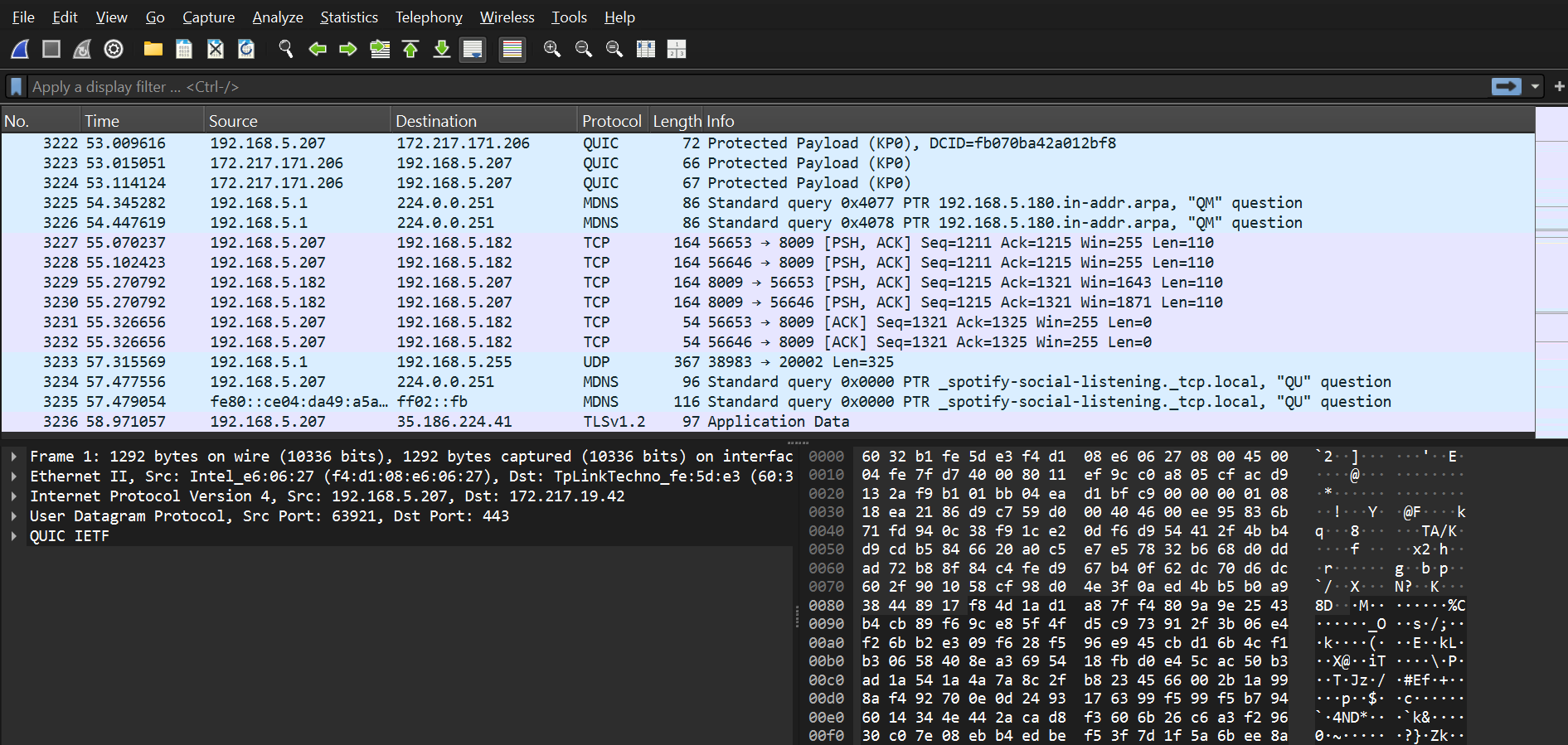
**TASK 3:**

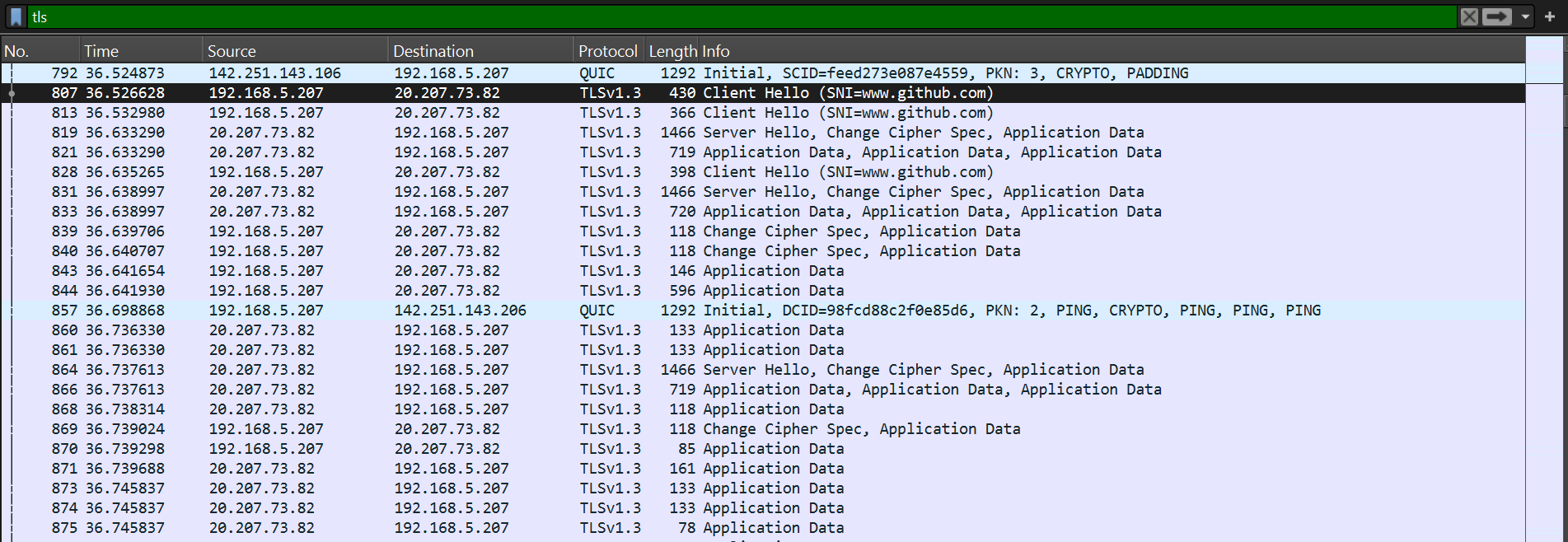
HTTP headers are divided into four main categories.

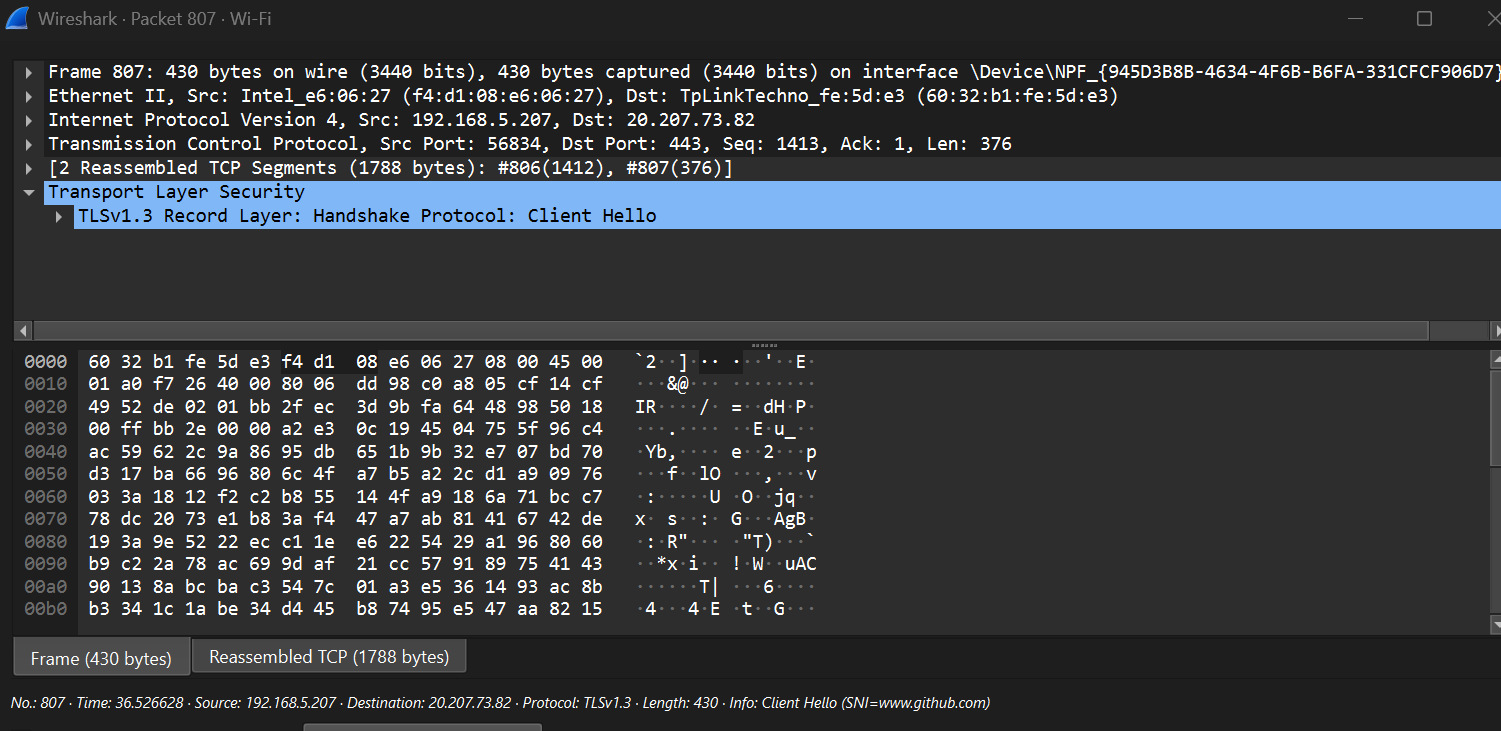
1. **General Headers**  
   These are headers that apply to both request and response messages, but they don’t describe the actual message body.  
   **Examples:** Date, Cache-Control, Connection
2. **Request Headers**  
   These headers are sent by the client and provide extra information about the request or the resource being asked for.  
   **Examples:** Host, User-Agent, Accept, Authorization
3. **Response Headers**  
   These headers are sent by the server and describe the server’s response, but not the actual content.  
   **Examples:** Server, Set-Cookie, WWW-Authenticate, Location
4. **Entity Headers**  
   These headers give details about the body of the request or response, such as its type, size, or encoding.  
   **Examples:** Content-Type, Content-Length, Content-Encoding, Last-Modified

**WIRESHARK TASKS:**

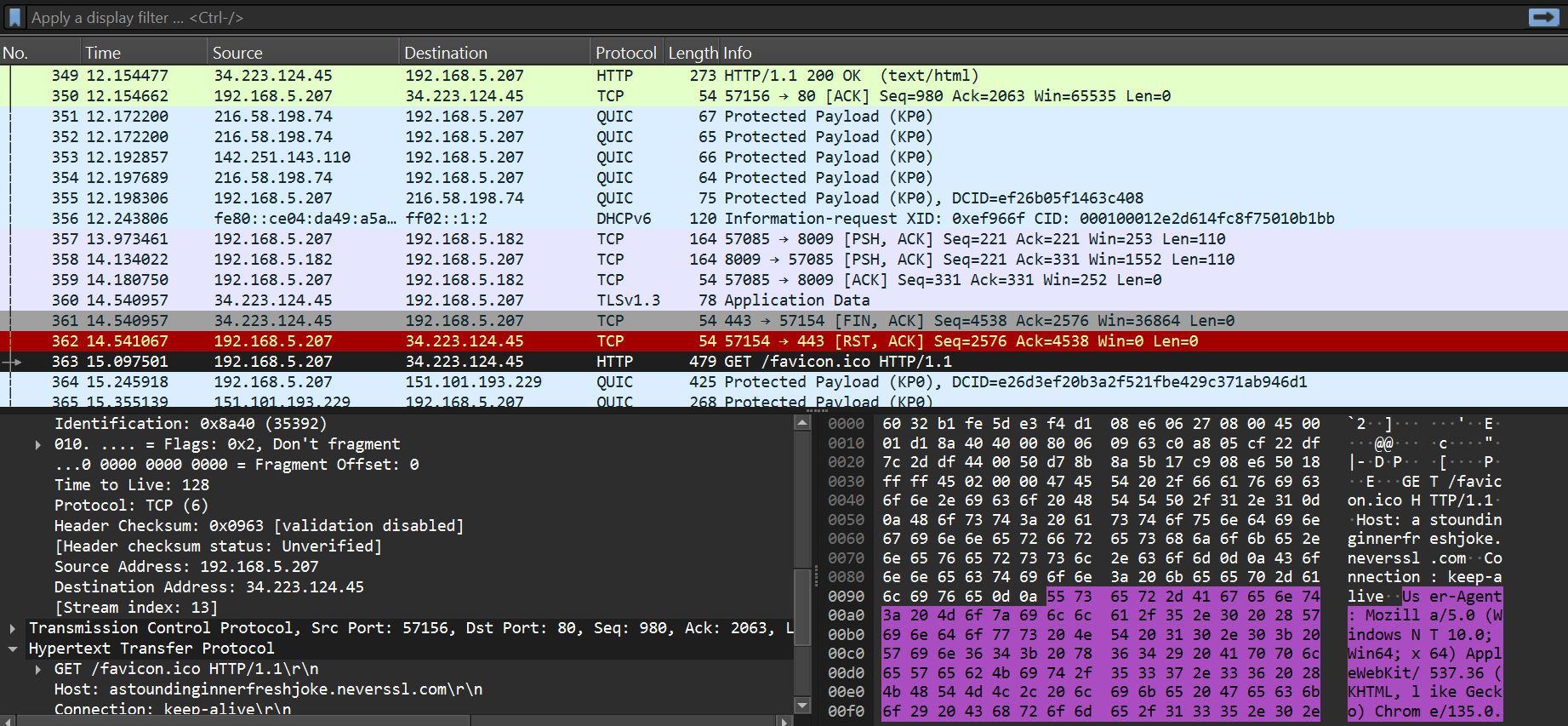
Loaded <https://www.github.com> and began capture on wireshark and stopped capture after page loaded

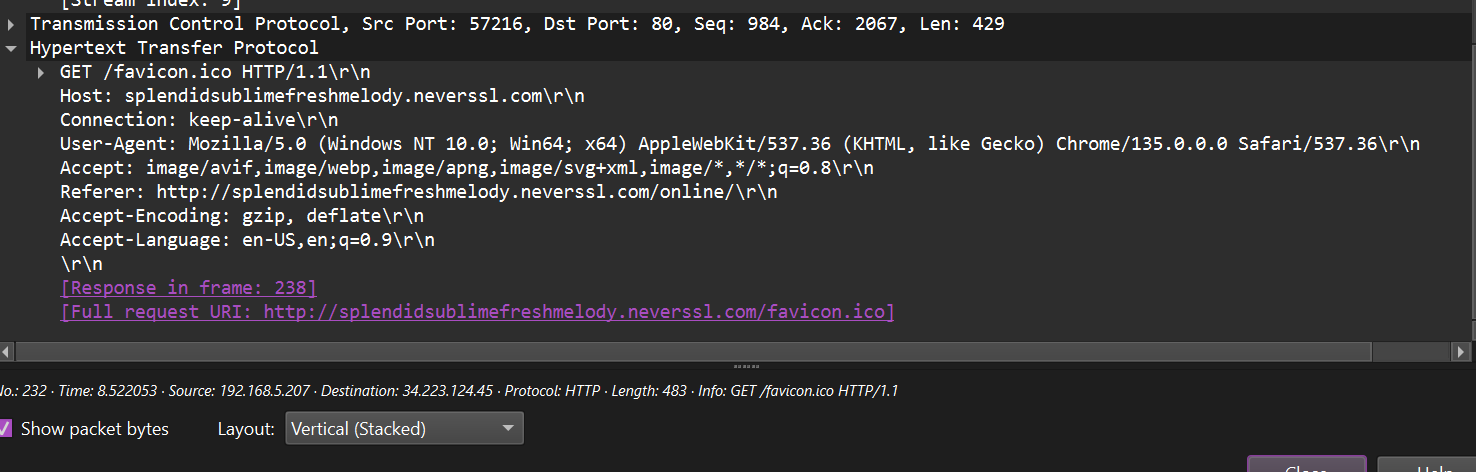


Applied the TLS filter because HTTPS packets are encrypted and cannot be read directly.



In HTTP, requests and responses (like GET and 200 OK) are visible in plain text, showing headers and content





The main difference between HTTP and HTTPS is that HTTPS encrypts the data (visible as TLS handshake and encrypted packets), while HTTP transmits data in plain text as seen in the captures.