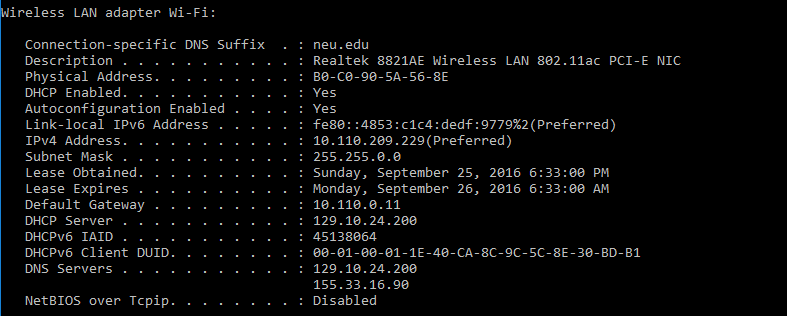
**Lab report for Internetworking**

**Name: Rishabh Jain**

Step 1 (DHCP):

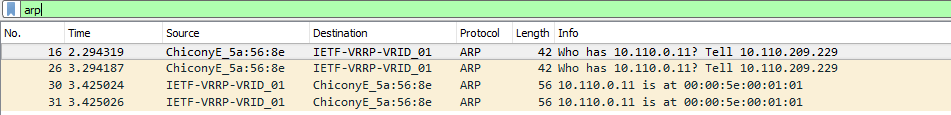
If a PC is added to network, it is assigned a ip address automatically using DHCP protocol (Dynamic Host Control Protocol). Once the PC is assigned ip address, DHCP server also sends out information such as reachable DNS servers and default gateways.



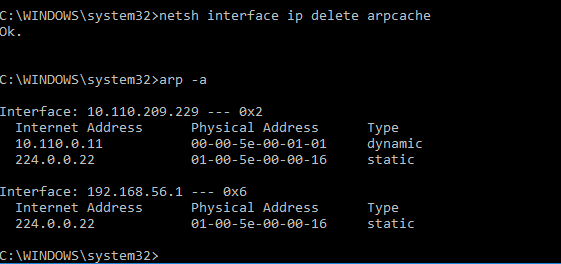
So we can see in above screenshot that, My PC’s ip address is 10.110.209.229, gateway information (10.110.0.11) and DNS server (129.10.25.200 and 155.33.16.90) to reach out.

Step 2 (ARP):

Usually the mac address of default gateway is already present in arp table. But, as informed by instructor, I deleted arp cache table so we can see that ARP sends a broadcast ARP request asking for mac address of default gateway 10.110.0.11 in below picture.



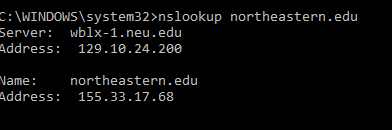
This important because any packet leaving my PC passes through default gateway and knowing mac address of default gateway is done using ARP request broadcast. Once it gets ARP reply with mac address it updates local arp table for future reference

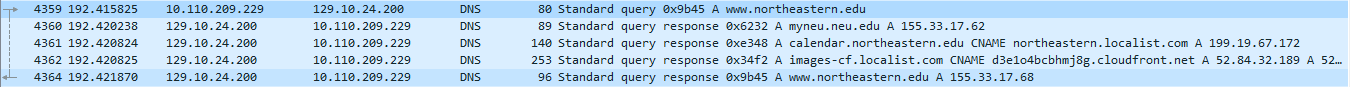


Now we can see mac address for default gateway (10.110.0.11) is 00-00-5e-00-01-01 which is same we saw in Wireshark screenshot above.

Step 3 (DNS):

DNS protocol is used to map domains names with ip address. Because remembering “northeastern.edu” is easier than remembering 155.33.17.68. So browser sends out standard DNS query to resolve [www.northeastern.edu](http://www.northeastern.edu) and DNS server responds with resolved ip address for that domain. i.e, [www.northeastern.edu](http://www.northeastern.edu) . DNS uses UDP as protocol for transmission and uses udp port 53





We can see in last line of screenshot it has standard query response with resolved ip for northeastern.edu i.e, 155.33.17.68

The arrow marks on the left show the request and response mapping.

Step 4 (TCP):

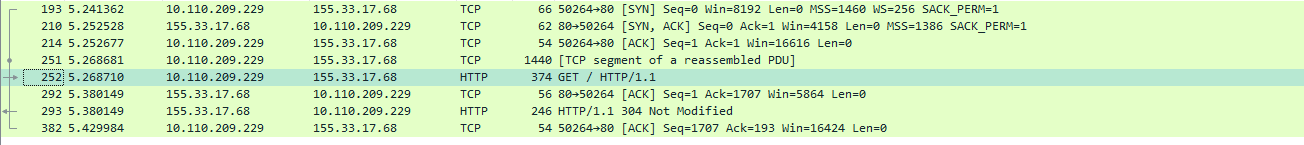
Once the browser has resolved IP address for “www.northeastern.edu”, browser initiates TCP connection by performing TCP handshake. We can see that first packet is from my PC to server with SYN flag which marks TCP connection initiation. Then server replies with SYN/ACK flags ON and then my system completes the TCP handshake by send out packet with ACK.

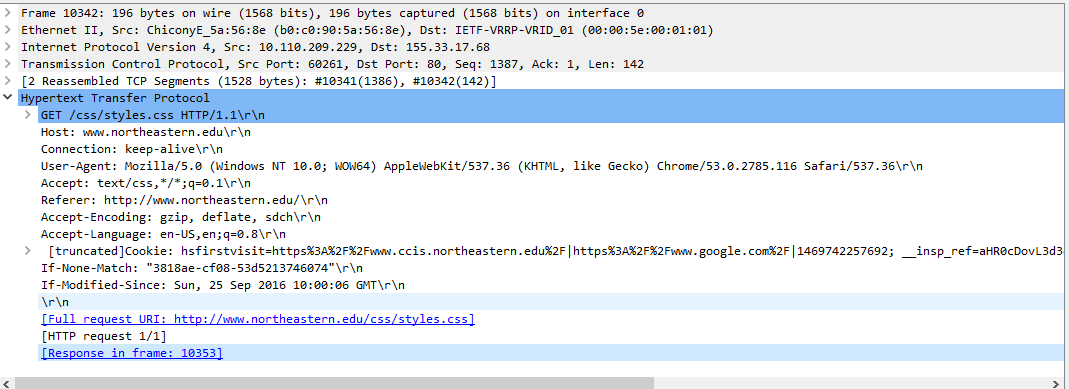


TCP connection is reliable and guarantees delivery of data. Also it uses sequence numbers to make sure that the packets are received in order.

Step 5 (HTTP):

HTTP uses TCP as protocol for transmission, HTTP uses port 80 and web server listens on this port for incoming connections and sends requested resources. So once TCP connection is established. Browser starts asking for resources using GET or POST messages and server responds with OK 200 response if it finds the requested resources.





In the above screenshot we see that “Get /css/style.css HTTP/1.1” HTTP request to server and the server replies HTTP/1.1 304 Not Modified response. We get this 304 response because the browser in HTTP GET request used If-Modified-Since header to determine if it can use the cached response instead of server sending it all again. This means the browser now uses the cached style.css rather that receiving it from server.

**To be precise:**

1. First PC gets assigned IP address (10.110.209.229) along with information such as default gateway and DNS server. DHCP uses port 67, 68 for client and server respectively
2. Arp resolves the mac address of default gateway (10.110.0.11) by broadcasting arp request
3. Browser send out DNS query to resolve ‘www.northeastern.edu’. DNS works on UDP port 53. The ip address for web server is 155.33.17.68
4. TCP connection is established with (SYN, SYN/ACK, ACK) and then HTTP request is sent using Get or Post command followed by resource requested.
5. HTTP uses port 80 on webserver and listens for incoming connection and send out requested resources. (in our case / index file or default file)

**Network stack:**

HTTP 🡪 Application layer

TCP 🡪 Transport layer

DNS 🡪 Between network and Transport layer

ARP 🡪Between Link layer and Network Layer

DHCP