COL374/672 Computer Networks: 2020-21 semester I

Assignment 2

In this assignment, we will understand more about inspecting network traffic using tools like Wireshark and the Chrome developer tools.

1. Use *wireshark* to grab all packets on your wireless interface, while visiting the website http://www.cse.iitd.ac.in from your browser. Do an *ipconfig /flushdns*[1] before you do this activity to clear your local DNS cache. And also clear your browser cache. Report the following:

    a. Apply a "dns" filter on the packet trace, and see if you can find DNS queries and responses for www.cse.iitd.ac.in. What DNS server was used? How long did it take for the DNS request-response to complete?

    b. Apply an "http" filter on the packet trace and report the approximate number of HTTP requests that were generated to download all the objects on the home-page. What can you tell from this observation about how web-pages are structured, and how browsers render complex pages with multiple images and files?

    c. Apply a filter such as "((*ip.src==192.168.1.3 && ip.dst==10.7.174.111) || (ip.src==10.7.174.111 && ip.dst==192.168.1.3)) && tcp*". As would be self-explanatory, this will filter for TCP packets moving between your browser and the web-server. Recall that the source and destination IP addresses are a part of the network layer header, which is also called the IP layer since IP (Internet Protocol) is the most common network layer protocol in use. Find the number of TCP connections that were opened between your browser and the web-server. Recall that a TCP connection is identified by the 4-tuple (source IP, destination IP, source port, destination port).

    d. In the previous part, do you find that several content objects are fetched over the same TCP connection?

    e. Notice that before an HTTP message is sent on a new TCP connection, a 3-way handshake is first performed to establish the TCP connection. The client sends a SYN message to the server, the server replies with a SYN-ACK message, and the client then sends an ACK. You will find that several TCP connections were opened between your browser and the web-server. How much time does it take for this handshake, before the connection can be used to send/receive data? Given this latency, what kind of optimizations do you think the browser might want to follow to minimize the overall page-load time?

    f. Report the total time taken for download of the entire webpage measured as the time at which the first DNS request was sent and the time when the last content object was

[1] On Linux, use: sudo /etc/init.d/dns-clean restart; sudo /etc/init.d/networking force-reload

received. This is called the page load time. Another useful metric that browsers try to optimize is called the above fold time, which is the time taken to first download objects that are sufficient to render the part of the web-page that's visible on the screen, ie. above the fold.

g. Try doing a trace for http://www.indianexpress.com and filter for "http". What do you find, is there any HTTP traffic? Browse through the entire trace without any filters, are you able to see the contents of any HTML and Javascript files being transferred? Why is that, while you were able to do it easily earlier for http://www.cse.iitd.ac.in?

2. Now open the Chrome browser and go to Developer Tools. Open the Network tab. Go to http://www.indianexpress.com. Report the following:

   a. Why are you able to see the different content objects in the browser, which you were earlier not able to see through Wireshark?

   b. How many content objects were downloaded to render the home-page of www.indianexpress.com? You will see that many of these objects are not from the indianexpress.com domain. Where are they from? What do you think is the purpose of these objects?

   c. Look at the timing information for any object. Try finding one of the larger objects and look at the Timing sub-tab for this object to see the breakup of the time taken to download the object. What average throughput was observed during the content download period, ie. when the content was actually being downloaded, not counting other latencies like DNS lookup or TCP connection establishment delay?

   d. Now do the same thing for http://www.nytimes.com and compare the total amount of content downloaded to render the NY Times home-page, with the content downloaded to render the Indian Express home-page. What does this tell you about creating websites?

   e. Do you agree that from a user-experience point of view, since web-pages are constituted of many small objects and which could be hosted on multiple domains, factors like the roundtrip delay and optimizations by the browser to pipeline downloads of multiple objects, are more important than the network throughput that is obtained? Explain your answer.

   f. Chrome allows you to test websites by emulating different network conditions. Look at the Throttling dropdown in the Network tab and change the network to Fast 3G, Slow 3G, etc. You can also build your own custom network profile by specifying the mean downlink throughput, uplink throughput, and latency. Experiment with different values, such as the following:

      (technology, downlink throughput in kbps, uplink throughput in kbps, latency in ms)
      Regular 2G, 250, 50, 300
      Good 2G, 450, 150, 150

Regular 3G, 750, 250, 100
Good 3G, 1000, 750, 40
Regular 4G, 4000, 3000, 20
DSL, 2000, 1000, 5

How do you think Chrome is able to emulate such different networks?

Chrome also allows you to emulate different devices which is particularly useful when the computation capabilities of the device may begin to affect the user experience. Why do you think the device computation capabilities can have this effect?

g. Go to http://www.indianexpress.com and export the Chrome trace as a HAR file. Open this file using a text editor understand the information it contains, look up on the web if needed to understand the structure of HAR files. Examine the requests going to ad networks like Double Click or analytics services like Google Analytics and report what third-party domains do you see being accessed? What user-specific information seems is being sent to these websites? What kind of information do you think these third-party domains are requesting to be saved as cookies locally? Go to your Chrome privacy settings, then to the cookie settings. Do you have third party cookies blocked?

3. Let's go back to Wireshark now. Disconnect from the network, release the DHCP address by invoking *ipconfig /release*[2], then start Wireshark, and connect back to the network.

   a. Filter for DHCP packets and describe what seems to be the protocol through which DHCP is operating. Draw a transaction diagram of the DHCP messages that you see being sent and received by your device. Report what underlying transport layer protocol is being used.

   b. Run a traceroute to www.google.com and similarly report for DNS messages, of what messages are sent and received by your device, and the underlying transport layer protocol.

   c. Now filter for ICMP messages and similarly report what messages traceroute seems to be sending.

   d. Run different streaming applications while capturing their packets, like when watching a video on youtube, or talking on Skype, or on Zoom. Report what underlying protocols seem to be in use for the data streams.

What to submit: A neat report in pdf, formatted using Latex. Please use the exact same question numbering as above.

[2] On Linux, use: sudo dhclient –r; sudo dhclient