COL334 Assignment 2

Dipen Kumar (2018CS50098)

October 31, 2020

1. Wireshark

1. I applied "dns" filter on packet trace and saw DNS queries and responses for www.cse.iitd.ac.in. My local DNS server was 192.168.1.1. It took 0.226575 seconds for the DNS request-response to complete.

2. I applied "http" filter on the packet trace. The approximate number of HTTP requests that were generated to download all the objects on the home-page was 39. Web page consists of objects. An object is simply a file- such as an HTML file, a JPEG image, a java applet, or a video clip- that is addressable by a single URL. Browser first request for base HTML file and text for the user to start reading and latter with time it request other referred objects such as images and video.

3. I applied the filter as "((ip.src == 192.168.1.105 && ip.dst == 103.27.9.152) || (ip.src == 103.27.9.152 && ip.dst == 192.168.1.105)) tcp" to filter for TCP packets moving between my browser and the web-server. The number of TCP connections that were opened between my browser and the web-server was 6 with port number of my browser as client socket from 51946 to 51951 whereas web application running on IITD sever with port number for server socket as 80.

4. Yes, several content objects are fetched over the same TCP connection. I observed that more than one HTTP requests were generated to fetch several objects over same TCP connection identified by the 4-tuple (source IP, destination IP, source port, destination port).

5. Yes, 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. Six parallel TCP connections were opened between my browser and the web-server and it took them 0.105122, 0.121896, 0.121707, 0.134349, 0.172114 and 0.172001 seconds for this handshake, before the connection can be used to send/receive data. Given this latency, browser might want to open many parallel TCP connections and might want to use persistence connections with pipelining and will not want to establish one TCP connection for every objects to load which has delay (latency) of two RTT. In this way browser can minimize the overall page-load time.

6. Page load time for www.cse.iitd.ac.in was 14.453096 seconds.

7. I did trace for http://www.indianexpress.com and filter for "http". I find there a HTTP request message using GET method and HTTP/1.1 protocol which was responded by HTTP/1.1 version, 302 status code and "Moved Permanently" phrase. I find no HTTP traffic. I browsed through the entire trace without any filters and I was not able to see the contents of any HTML and Javascript files being transferred because http://www.indianexpress.com responded with "HTTP/1.1 301 Moved Permanantly" and "Location: https://www.indianexpress.com". This new URL (HTTPS means HTTP

over TLS) where we are directed is encrypted and secure. We were able to do it easily earlier for http://www.cse.iitd.ac.in because it used "http" which is not encrypted and not secure.

2. Chrome Developer Tools

1. I am able to see the different content objects in the browser, which you were earlier not able to see through Wireshark because request was made by browser using HTTPS which encrypt the response message that could not be read by third party like wireshark but when the message reaches the browser, the other end of TCP connection where it is decrypted and visible through browser.

2. Approximately 350 content objects were downloaded to render the home-page of www.indianexpress.com. Many of these objects are not from the indianexpress.com domain. They are from ad networks like Double Click, analytics services like Google Analytics, and other third-party service providers like Google APIs for variety of uses. Objects provided by ad networks are for advertisements. Indian Express makes money from ad networks by advertising third-party products and services on its web-page. Objects by analytics services takes data recording web-page activities and do all statistics and analysis for Indian Express. APIs are very useful to avoid doing every thing from scratch. Instead of coding every thing used by the web-page, Indian Express uses pre implemented servies of google through APIs. Like one can use google sheet API, one can use google API for user login.

3. Average throughput that was observed during the content download period was 320 KBps, when observed on 1.7 MB object which took 5.43 seconds to download.

4. Total amount of content downloaded to render the NY Times home-page was 3.2 MB whereas the total amount of content downloaded to render the Indian Express home-page was 1.7 MB. This says that we should keep our contents light in terms of content data because greater the amount of content, greater is the page-load time. For the NY Times it was 16.77 seconds and for the Indian Express it was 13.10 seconds with an average download speed of 300 KBps. We can partition our data and keep it stored at other links and we will just provide the links for that in our website under some well organised tabs rather than providing all the data at once which may not be useful for user and user may get overwhelmed. We can also do some advanced optimization in browser and first focus on to load the portion of web-page which is currently holding the window and hence we can reduce fold time. Browser should first load some primary text files for users to start reading and keep going while it may further continue to load big objects. I have also observed that for websites with many small objects, roundtrip delay becomes more significant and must be countered using pipelining or we can use persistence TCP connection and put all objects at same domain so that we can continue with our existing TCP connection with creating new connections and hence we are saving handshaking roundtrip delays.

5. Yes, web-pages which 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. This is because for many small objects hosted on many different domains we need to create many TCP connection for at least each of these many different domains in case of persistence connections. In case case of non persistence connection this number of connections may go even higher. With many TCP connections come many three way handshakes and increases roundtrip delay. Since roundtrip delay will dominate, we will need to reduce this because no matter how fast we download content objects there will still exist this roundtrip delay and it will still remain dominant. One classical way to deal with delay is

piplining, here we fully exploit our resources and don't waste time siting idle. Here we don't wait for our first job to complete rather we start our second job sooner and saves time and reduce delay. In this way we can shift the dominance from delays to throughput and then we can think of increasing the throughput. Hence 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.

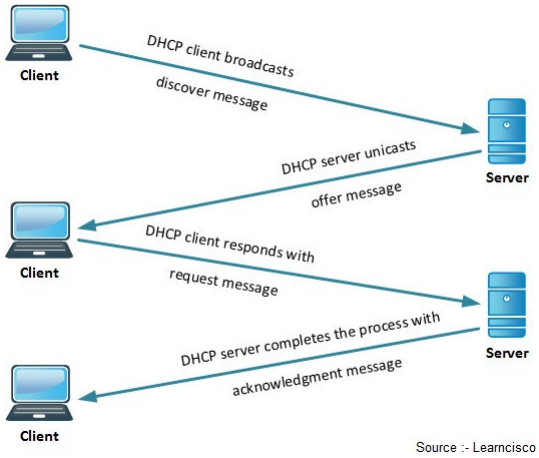
6. Chrome is able to emulate different networks by reducing its TCP window size. The slower the network to emulate the more chrome reduces its TCP window size. Device computation capabilities may begin to affect the user experience. Suppose our device is connected to a high speed network and receives data at high speed but our device is itself a slow performing in terms of computation. Computation here may be something like decrypting the encrypted message back to original user readable form. In this case despite our high speed network we won't be experencing high speed service also it may happen that packets at our end system (device) may start dropping out once the buffer is full.

	Indian Express (data downloaded)	Indian Express (page-load time)	NY Times (data downloaded)	NY Times (page-load time)
Regular 2G	202 KB	10.87 sec	458 KB	16.96 sec
Fast 3G	2.5 MB	32.30 sec	3.3 MB	23.52 sec
Slow 3G	625 KB	50.88 sec	454 KB	17.98 sec

7. Third-party domain are securepubads.g.doubleclick.net, www.google-analytics.com. Cookies is being sent to these third-party domains. These third-party domains are requesting user browsing history to be saved locally to show more personalized and relevant advertisements. No, I don't have third-party cookies blocked on my google chrome.

3. Back to Wireshark

1. I am using macOS and command used here to release and renew the DHCP address was "sudo ipconfig set en0 BOOTP and sudo ipconfig set en0 DHCP". Dynamic Host Configuration Protocol is used to dynamically assign IP address to each devices on the network. Initially devices send a Discover message as source 0.0.0.0 to special ip address 255.255.255.255 and get a Offer in response message and then it request for that ip address and finally our device gets acknowledgment. Underlying transport layer protocol being used is UDP



Source :- Learncisco

2. Messages are sent and received by our device for fetching domain name for intermediate routers. Underlying transport layer protocol being used is UDP

34	10.321920	192.168.1.105	192.168.1.1	DNS	74	Standard query 0x1627 A www.google.com
35	10.363481	192.168.1.1	192.168.1.105	DNS	90	Standard query response 0x1627 A www.google.com A 216.58.200.164
38	10.366917	192.168.1.105	192.168.1.1	DNS	84	Standard query 0x1b7e PTR 1.1.168.192.in-addr.arpa
39	10.409210	192.168.1.1	192.168.1.105	DNS	84	Standard query response 0x1b7e Server failure PTR 1.1.168.192.in-addr.arpa
46	11.425344	192.168.1.105	192.168.1.1	DNS	85	Standard query 0x835a PTR 1.128.244.49.in-addr.arpa
47	11.469080	192.168.1.1	192.168.1.105	DNS	157	Standard query response 0x835a PTR 1.128.244.49.in-addr.arpa PTR 1-adsl.ntc.net.np NS munal.ntc.net.np…
54	11.646173	192.168.1.105	192.168.1.1	DNS	86	Standard query 0xba5d PTR 181.88.70.202.in-addr.arpa
55	11.689679	192.168.1.1	192.168.1.105	DNS	144	Standard query response 0xba5d No such name PTR 181.88.70.202.in-addr.arpa SOA danphe.ntc.net.np
62	11.818095	192.168.1.105	192.168.1.1	DNS	85	Standard query 0xf0aa PTR 41.213.26.10.in-addr.arpa
63	11.861118	192.168.1.1	192.168.1.105	DNS	85	Standard query response 0xf0aa Server failure PTR 41.213.26.10.in-addr.arpa
81	17.938917	192.168.1.105	192.168.1.1	DNS	85	Standard query 0x341c PTR 97.93.70.202.in-addr.arpa
82	17.981620	192.168.1.1	192.168.1.105	DNS	143	Standard query response 0x341c No such name PTR 97.93.70.202.in-addr.arpa SOA danphe.ntc.net.np
107	23.031550	192.168.1.105	192.168.1.1	DNS	86	Standard query 0x0b5a PTR 190.93.70.202.in-addr.arpa
108	23.076878	192.168.1.1	192.168.1.105	DNS	86	Standard query response 0x0b5a Server failure PTR 190.93.70.202.in-addr.arpa
119	24.165678	192.168.1.105	192.168.1.1	DNS	86	Standard query 0xbdb1 PTR 158.93.70.202.in-addr.arpa
120	24.208746	192.168.1.1	192.168.1.105	DNS	169	Standard query response 0xbdb1 PTR 158.93.70.202.in-addr.arpa PTR bhr.core-but.core.ntc.net.np NS danp…
127	24.364892	192.168.1.105	192.168.1.1	DNS	86	Standard query 0xccb9 PTR 92.119.125.74.in-addr.arpa
132	24.502812	192.168.1.1	192.168.1.105	DNS	86	Standard query response 0xccb9 Server failure PTR 92.119.125.74.in-addr.arpa
139	25.590214	192.168.1.105	192.168.1.1	DNS	86	Standard query 0xa44b PTR 97.243.125.74.in-addr.arpa
140	25.726561	192.168.1.1	192.168.1.105	DNS	146	Standard query response 0xa44b No such name PTR 97.243.125.74.in-addr.arpa SOA ns1.google.com
143	25.811829	192.168.1.105	192.168.1.1	DNS	87	Standard query 0x4a69 PTR 193.244.125.74.in-addr.arpa
144	25.947582	192.168.1.1	192.168.1.105	DNS	87	Standard query response 0x4a69 Server failure PTR 193.244.125.74.in-addr.arpa
153	26.981559	192.168.1.105	192.168.1.1	DNS	86	Standard query 0x6b74 PTR 85.67.253.172.in-addr.arpa
154	27.658997	192.168.1.1	192.168.1.105	DNS	86	Standard query response 0x6b74 Server failure PTR 85.67.253.172.in-addr.arpa
157	28.069612	192.168.1.105	192.168.1.1	DNS	86	Standard query 0xca70 PTR 87.67.253.172.in-addr.arpa
158	28.785279	192.168.1.1	192.168.1.105	DNS	86	Standard query response 0xca70 Server failure PTR 87.67.253.172.in-addr.arpa
165	29.243600	192.168.1.105	192.168.1.1	DNS	87	Standard query 0xb3b4 PTR 164.200.58.216.in-addr.arpa
166	29.282223	192.168.1.1	192.168.1.105	DNS	156	Standard query response 0xb3b4 PTR 164.200.58.216.in-addr.arpa PTR del11s06-in-f4.1e100.net PTR nrt12s…
217	46.143146	192.168.1.105	192.168.1.1	DNS	89	Standard query 0xc030 A apidata.googleusercontent.com
218	46.189447	192.168.1.1	192.168.1.105	DNS	389	Standard query response 0xc030 A apidata.googleusercontent.com CNAME googlehosted.l.googleusercontent.…
244	46.851406	192.168.1.105	192.168.1.1	DNS	89	Standard query 0x9d7c A d27xxe7juh1us6.cloudfront.net
245	46.893728	192.168.1.1	192.168.1.105	DNS	153	Standard query response 0x9d7c A d27xxe7juh1us6.cloudfront.net A 204.246.164.22 A 204.246.164.2 A 204.…

3. Traceroute seems to be sending ping messages to destination servers with ttl value starting from one and increasing by one every time till it reaches destination. In response it gets ip address and a message which says ttl exceeded if it has still not reached till destination.

37	10.365933	192.168.1.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
41	11.385044	192.168.1.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
43	11.386395	192.168.1.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
45	11.424127	49.244.128.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
49	11.509409	49.244.128.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
51	11.548047	49.244.128.1	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
53	11.645117	202.70.88.181	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
57	11.734772	202.70.88.181	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
59	11.777903	202.70.88.181	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
61	11.816937	10.26.213.41	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
68	12.857126	10.26.213.41	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
70	12.896062	10.26.213.41	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
80	17.937868	202.70.93.97	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
106	23.030458	202.70.93.190	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
110	24.077015	202.70.93.190	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
116	24.122939	202.70.93.190	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
118	24.164747	202.70.93.158	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
122	24.251734	202.70.93.158	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
124	24.295456	202.70.93.158	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
126	24.363792	74.125.119.92	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
134	25.436992	74.125.119.92	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
136	25.503942	74.125.119.92	192.168.1.105	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
138	25.589181	74.125.243.97	192.168.1.105	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
142	25.811136	74.125.244.193	192.168.1.105	ICMP	94	Time-to-live exceeded (Time to live exceeded in transit)
150	26.898147	74.125.244.193	192.168.1.105	ICMP	94	Time-to-live exceeded (Time to live exceeded in transit)
152	26.980650	172.253.67.85	192.168.1.105	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
156	28.068476	172.253.67.87	192.168.1.105	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
162	29.156069	172.253.67.85	192.168.1.105	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
164	29.242057	216.58.200.164	192.168.1.105	ICMP	70	Destination unreachable (Port unreachable)
168	29.365639	216.58.200.164	192.168.1.105	ICMP	70	Destination unreachable (Port unreachable)
171	29.451565	216.58.200.164	192.168.1.105	ICMP	70	Destination unreachable (Port unreachable)

4. Underlying transport layer protocols used in youtube, google meet and zoom are TCP, UDP and UDP respectivily.