A Day in the Life of a Web Page Request

Zaeem Yousaf(L19-1196 Section 5C)

Submitted To sir Sulman Mubarak

Contents

1 Scenario

Zeem boots up his PC and afterward interfaces it to an Ethernet link associated with the school's Ethernet switch, which thusly is associated with the school's switch. The school's switch is associated with an ISP, in this model, PTCL.net. PTCL.net is giving the DNS administration to the school.

2 summary of major steps

At the point when Zaeem first interfaces his PC to the organization, he can do nothing without an IP address. Accordingly, the main organization related activity taken by Zaeem's PC is to run the DHCP convention to get an IP address, just as other data, from the nearby DHCP server.

The working framework on Zaeem's PC makes a DHCP demand message and puts this message inside a UDP section with objective port 67 DHCP

server and source port 68 DHCP customer. The UDP fragment is then positioned inside an IP datagram with a transmission IP objective location 255.255.255.255 and a source IP address of 0.0.0.0, since Zaeem's PC doesn't yet have an IP address.

The IP datagram containing the DHCP demand message is then positioned inside an Ethernet outline. The Ethernet outline has an objective MAC locations of FF:FF:FF:FF:FF:FF so the edge will be communicated to all gadgets associated with the switch ideally including a DHCP server; the edge's source MAC address is that of Zaeem's PC, 00:16:D3:23:68:8A.

The transmission Ethernet outline containing the DHCP demand is the primary edge sent by Zaeem's PC to the Ethernet switch. The switch communicates the approaching edge on every single active port, including the port associated with the switch.

The switch gets the transmission Ethernet outline containing the DHCP demand on its point of interaction with MAC address 00:22:6B:45:1F:1B and the IP datagram is separated from the Ethernet outline. The datagram's transmission IP objective location shows that this IP datagram ought to be handled by upper layer proto-cols at this hub, so the datagram's payload a UDP portion is consequently demultiplexed up to UDP, and the DHCP demand message is removed from the UDP section. The DHCP server currently has the DHCP demand message.

We should assume that the DHCP server running inside the switch can distribute IP addresses in the CIDR block 68.85.2.0/24. In this model, all IP addresses utilized inside the school are hence inside PTCL's location block.Let's guess the DHCP server assigns address 68.85.2.101 to Zaeem's PC. The DHCP server makes a DHCP ACK message containing this IP address, just as the IP address of the DNS server 68.87.71.226, the IP address for the default passage switch 68.85.2.1, and the subnet block 68.85.2.0/24 proportionally, the "network veil". The DHCP message is put inside a UDP portion, which is put inside an IP datagram, which is put inside an Ethernet outline. The Ethernet outline has a source MAC address of the switch's connection point to the home organization 00:22:6B:45:1F:1B and an objective MAC address of Zaeem's PC 00:16:D3:23:68:8A.

Zaeem's PC gets the Ethernet outline containing the DHCP ACK, extricates the IP datagram from the Ethernet outline, removes the UDP portion from the IP datagram, and separates the DHCP ACK message from the UDP section. Zaeem's DHCP customer then, at that point, records its IP address and the IP address of its DNS server. It likewise introduces the location of the default entryway into its IP sending table Section 4.1. Zaeem's PC will send all datagrams with objective location outside of its subnet 68.85.2.0/24

to the default door. Now, Zaeem's PC has introduced its systems administration parts and is prepared to start handling the Web page get. Note that main the last two DHCP steps of the four introduced in Chapter 4 are really fundamental.

3 Detail

3.1 DNS and ARP

At the point when Zaeem types the URL for www.github.com into his Web program, he starts the long chain of occasions that will ultimately bring about Google's landing page being dis-played by his Web program. Zaeem's Web program starts the cycle by making a TCP attachment Section 2.7 that will be utilized to send the HTTP demand Section 2.2 to www.github.com. To make the attachment, Zaeem's PC should realize the IP address of www.github.com. We learned in Section 2.5, that the DNS proto-col is utilized to give this name-to-IP-address interpretation administration.

The operating system on Zaeem's laptop thus creates a DNS query message and then putting the string "www.github.com" in the inquiry segment of the DNS message. This DNS message is then positioned inside a UDP section with a destina-tion port of 53 DNS server. The UDP portion is then positioned inside an IP information gram with an IP objective location of 68.87.71.226 the location of the DNS server returned in the DHCP ACK in sync 5 and a source IP address of 68.85.2.101.

Zaeem's PC then, at that point, puts the datagram containing the DNS inquiry message in an Ethernet outline. This casing will be sent tended to, at the connection layer to the entryway switch in Zaeem's school's organization. Notwithstanding, despite the fact that Zaeem's PC realizes the IP address of the school's door switch 68.85.2.1 through the DHCP ACK message in sync 5 above, it doesn't have the foggiest idea about the passage switch's MAC address. To acquire the MAC address of the passage switch, Zaeem's PC should utilize the ARP convention Section 5.4.1.

Zaeem's PC makes an ARP inquiry message with an objective IP address of 68.85.2.1 the default passage, puts the ARP message inside an Ethernet outline with a transmission objective location FF:FF:FF:FF:FF:FF and sends the Ethernet casing to the switch, which conveys the edge to every associated gadget, including the entryway switch.

The passage switch gets the casing containing the ARP inquiry message on the point of interaction to the school organization, and observes that the

objective IP address of 68.85.2.1 in the ARP message matches the IP address of its connection point. The entryway switch along these lines readies an ARP answer, showing that its MAC address of 00:22:6B:45:1F:1B compares to IP address 68.85.2.1. It puts the ARP answer message in an Ethernet outline, with an objective location of 00:16:D3:23:68:8A Zaeem's PC and sends the casing to the switch, which conveys the edge to Zaeem's PC.

Zaeem's PC gets the edge containing the ARP answer message and concentrates the MAC address of the entryway switch 00:22:6B:45:1F:1B from the ARP answer message. Zaeem's laptop can now finally address the Ethernet frame containing the DNS question to the door switch's MAC address. Note that the IP datagram in this edge has an IP objective location of 68.87.71.226 the DNS server, while the edge has an objective location of 00:22:6B:45:1F:1B the door switch. Zaeem's PC sends this edge to the switch, which conveys the casing to the entryway switch.

3.2 Intra-Domain Routing to the DNS Server

The door switch gets the edge and concentrates on the IP datagram containing the DNS inquiry. The switch looks into the objective location of this datagram 68.87.71.226 and decides from its sending table that the datagram ought to be shipped off the furthest left switch in the PTCL network. The IP information is set inside a connection layer outline suitable for the connection associating the school's switch to the furthest left PTCL switch and the edge is sent over this connection.

The furthest left switch in the PTCL network gets the edge, removes the IP datagram, looks at the datagram's objective location 68.87.71.226 and decides the active point of interaction on which to advance the datagram towards the DNS waiter from its sending table, which has been filled in by Comcast's intra-space convention, e.g RIP.

At last the IP datagram containing the DNS question shows up at the DNS server. The DNS server extricates the DNS inquiry message, looks into the name www.github.com in its DNS data set Section 2.5, and observes the DNS asset record that contains the IP address 64.233.169.105 for www.github.com. expecting that it is as of now reserved in the DNS server. Review that this reserved information began in the definitive DNS server Section 2.5.2 for googlecom. The DNS server frames a DNS answer message containing this hostname-to-IP-address planning, and places the DNS answer message in a UDP section, and the fragment inside an IP datagram addressed to Zaeem's PC 68.85.2.101. This datagram will be sent back through the PTCL organization to the school's switch and from that point,

by means of the Ethernet change to Zaeem's PC.

Zaeem's PC separates the IP address of the server www.github.com from the DNS message. At long last, later a great deal of work, Zaeem's PC is currently prepared to contact the www.github.com server!

4 Web Client-Server Interaction: TCP and HTTP

4.1 GET and POST

Since Zaeem's PC has the IP address of www.github.com,it can make the TCP attachment that will be utilized to send the HTTP GET message to www.github.com. At the point when Zaeem makes the TCP attachment, the TCP in Zaeem's PC should initially play out a three-way handshake with the TCP in www.github.com. Zaeem's PC hence first makes a TCP SYN fragment with objective port 80 for HTTP, puts the TCP section inside an IP information datagram with an objective IP address of 64.233.169.105 www.github.com, places the datagram inside an edge with an objective MAC address of 00:22:6B:45:1F:1B the entryway switch and sends the casing to the switch.

The switches in the school organization, PTCL's organization, and Google's organization forward the datagram containing the TCP SYN towards www.github.com, utilizing the sending table in every switch, as in stages 14–16 above. Review that the switch sending table passages administering sending of bundles over the between area interface between the PTCL and Google networks are controlled by the BGP convention.

4.2 Data flow

Ultimately, the datagram containing the TCP SYN shows up at www.github.com. The TCP SYN message is removed from the datagram and demultiplexed to the welcome attachment related with port 80. An association attachment Section 2.7 is made for the TCP association between the Google HTTP server and Zaeem's PC. A TCP SYNACK Section 3.5.6 fragment is created, set inside a datagram addressed to Zaeem's PC, lastly positioned inside a connection layer outline fitting for the connection associating www.github.com to its first-jump switch.

With the attachment on Zaeem's PC now at last! ready to send by www.google.com, Zaeem's program makes the HTTP GET message containing the URL to be gotten. The HTTP GET message is then composed into the attachment, with the GET message turning into the payload of a

TCP section. The TCP portion is set in a datagram and sent and conveyed to www.github.com.

The HTTP server at www.github.com peruses the HTTP GET message from the TCP attachment, makes a HTTP reaction message Section 2.2, places the mentioned Web page content in the body of the HTTP reaction message, and sends the message into the TCP attachment.

The datagram containing the HTTP answer message is sent through the Google, PTCL, and school organizations, and shows up at Zaeem's PC. Zaeem's Web program peruses the HTTP reaction from the attachment, extricates the html for the Web page from the body of the HTTP reaction and then at the end page is displayed on client side (at zaeem's laptop)