6. Networking with TCP and HTTP

github repo

Computer networking refers to interconnected computing devices that can exchange data and share resources with each other. Essentially two or multiple devices communicating with one another.

We have rules or protocols set in place, so that the exchange of information goes smoothly and there isn't any lost information. If there is, then there is a protocol in place to fix it.

- There is always one listening(client) and another sending(server) information. But the roles aren't fixed, they can change.

OSI (Open Systems Interconnection Model)

There are 7 layers in OSI(Refer to image OSI.webp) - 1 - Physical (Ethernet / USB / IEEE1394) - 2 - Datalink Layer (WiFi / ARP / MAC / PPP) - 3 - Network (IP / NAT / ARP) - 4 - Transport (TCP / UDP / FCP) - 5 - Session (TCP* / SMB / NetBEUI/Sockets) - 6 - Presentation (TLS / SSL / GZIP) - 7 - Application (HTTP / SMTP / SSH)

The ones we focused the most on were 1,2,3,4,5. These don't have to be memorized, as the OSI model isn't used, however it helps when debuggin networking issues in code.

TCP/UDP

Both are in transport layer. They set protocols on how data is transfered from client to server. * Messages are broken down into packets * Messages contained source and destination address in their header * Each packet travels through the network independently * Packets are reassembled into a full message when reaching destination

- TCP ensures that there are no lost packages.(No miscommunication)
- UDP is a connectionless protocol, it will continue the communication even if there is a lost package.(Live streaming websites)

A great example of TCP is the chat server, which was built using NET. Use telnet in another terminal to connect to the server. If the port is 3000, you would run `javascript telnet localhost 3000`

What is HTTP ?

HyperText Transfer Protocol - a protocol that deals with the transmission of hypermedia(HTML, JPEGs, etc.) It is apart of the 7th layer of OSI.

How a web client (browser, etc) communicates with a web server

The initial step of an HTTP exchange is the request. In the request, we want to make an action, decided by the method, to a specific path, the URL, with specific headers and an optional body content.

In httpRequest.js, we can see what information we can get out of a http request, by using the node package Request. To run the code please install the package by running `javascript npm install` A request will return information about the header, the body, and a response code for the request made.

Response Status

More information regarding status codes - 100 series - Informational responses - 200 series - Successful responses - 300 series - Redirection messages - 400 series - Client error messages - 500 series - Server error messages

View on Compass | Leave Feedback