CA304 Assignment 3 Report

Sockets & Websockets

Student Name: Amelia Grigoriev

Student Number: 19348241

1. Sockets

1.1. What are the three types of network socket and what are their differences?

A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. [1] The three types of network socket are stream sockets, datagram sockets and raw sockets.

Stream sockets perform like streams of information. Usually, the process sending information sends the length of the data, followed by the data itself. The process receiving information reads the length and then loops, accepting data until all of it has been transferred. [2] Stream sockets are most common as they use TCP (Transmission Control Protocol) for data transmission.

A datagram socket is a connectionless service. They're connectionless because you don't need to have an open connection as in Stream Sockets. [3] Datagrams are sent as independent packets. Data can be lost or duplicated, and datagrams can arrive out of order. [2] They use UDP (User Datagram Protocol).

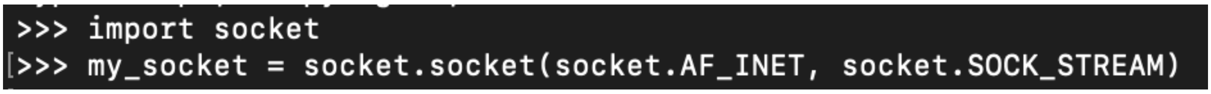
A raw socket allows direct access to lower layer protocols, such as IP and the ICMP. This interface is often used to test new protocol implementation, because the socket interface can be extended and new socket types defined to provide additional services. [2]

Stream sockets provide the most reliable connection whereas datagram sockets and raw sockets have a possibility of being discarded or duplicated during transmission.

1.2. What is the process of creating a network socket?

In Python we can create sockets using the socket library.

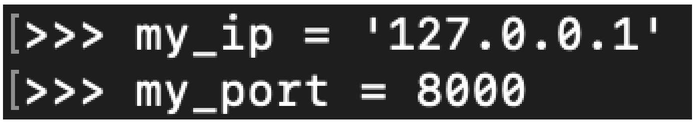
- To create a socket, first we need to install socket.
- Next, we need to call the socket constructor and pass in a family and a type. Family is the address and type is the protocol we are going to use.
- Some examples of family values are:
  - AF_INET - IPv4
  - AF_INET6 - IPv6
  - AF_BLUETOOTH
- The types are:
  - SOCK_STREAM - TCP data
  - SOCK_DGRAM - UDP data
  - SOCK_RAW - Roll your own packages
- We also have to assign our socket to a variable so that we can reference it in the future.



```
>>> import socket
[>>> my_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

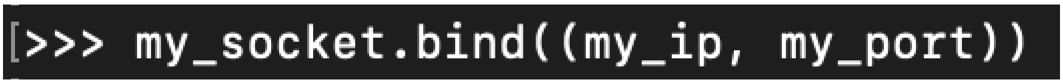
- Now we have told the OS we are looking to open up a TCP socket using the IPv4 addressing protocol.
- We now have a socket open but the server still has a bit more set up to do before it is able to handle incoming connections.
  - It needs to bind to an address

  - It needs to listen for incoming connections
  - It needs to accept connection requests
- We need to tell the socket what address we will be listening on and what port.
- To do this, we need to bind the socket to an address and a port.
- First we assign our address and port to variables so that we can reference them easily.



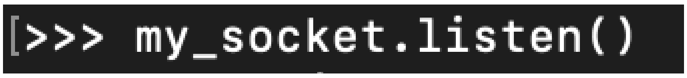
```
[>>> my_ip = '127.0.0.1'
[>>> my_port = 8000
```

- Next we bind the socket to our address and port.



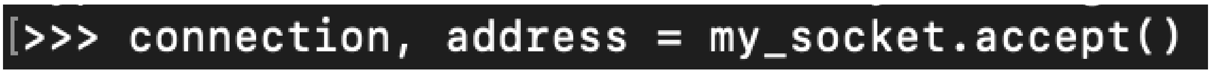
```
[>>> my_socket.bind((my_ip, my_port))
```

- Now, the socket is bound to address 127.0.0.1 and port 8000.
- Next, we need to allow the socket to start waiting for connections.
- We do this by calling socket.listen()



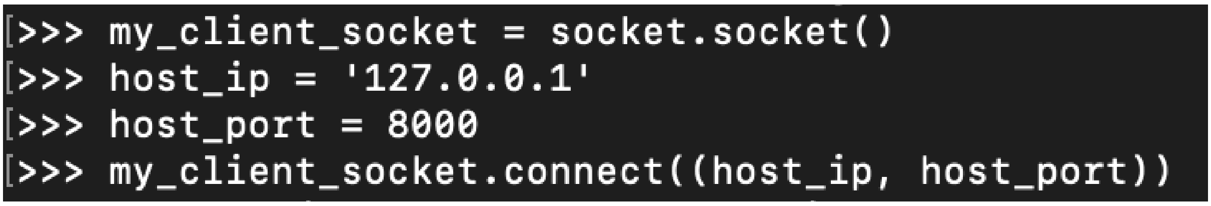
```
[>>> my_socket.listen()
```

- To accept a connection, we need to call socket.accept()



```
[>>> connection, address = my_socket.accept()
```

- This will return a connection object and the address of the machine making the connection.
- Once a socket is made for a client, it just needs to call connect to connect to the server.
- The connect call needs the address of the machine you want to connect to and the port number of the machine.



```
[>>> my_client_socket = socket.socket()
[>>> host_ip = '127.0.0.1'
[>>> host_port = 8000
[>>> my_client_socket.connect((host_ip, host_port))
```

- We have now created the network socket. [4]

1.3. The desktop computer with the IP address 192.168.0.5 is running multiple applications. What are the steps involved when the user opens their browser and visits the website discord.com, make sure to take account of the protocols involved in each step.

- Enter a URL
  - Click on the search bar of your browser and type in the address, discord.com.
  - This is known as a URL which stands for Uniform Resource Locator.

🔒 discord.com

- After a URL is entered into the search bar of your browser, the browser looks for the requested domain in its cache.
- If it’s not there, it requests the operating system’s DNS server to find the required IP address.
  - DNS stands for Domain Name System.
- Next, your browser requests a connection to the website.
- To do this it runs through a three step handshake process
  1. Your computer asks the website server if it's open to establishing new connections.

2. If the website can do so, it acknowledges that you are clear to connect.
3. Your computer then sends an acknowledgment that it received the confirmation. [5]

- The router is a link between the computer and server.
- It requests data from the internet and distributes it to networking devices such as desktop computers, laptops, and tablets. [6]
- When the IP address of the chosen webpage is identified, the browser requests the relevant data for the page from the appropriate web server.
  - This request takes place via HTTP in the form of a data packet, which contains all the information the web server needs in order to deliver the webpage data. [6]
- The router adds its own public IP address as sender and forwards the packet to the public internet. [6]
- The web server processes the information and transmits an HTTP status code. [6]
- If the request is successful, the server sends a data packet to the web browser with all the information required for the page.
- Incoming data packets from the internet are finally forwarded from the router to the computer on which the webpage is being accessed and the web browser then analyzes the packets.
- Lastly, the website which is generally composed of HTML, CSS and JavaScript files is shown.

2. Websockets

2.1. What are the differences between a websocket and a normal network socket?

WebSockets typically run from browsers connecting to Application Server over a protocol similar to HTTP that runs over TCP/IP. As a result, they're mostly used for Web applications that demand a constant connection to the server. Sockets, on the other hand, are more powerful and generic. They run over TCP/IP, but aren't limited to browsers or the HTTP protocol. They could be used to implement any kind of communication. [7]

2.2. Websockets provide full-duplex communication , what does this mean, and how did we achieve similar functionality before websockets were released.

The term full-duplex describes simultaneous data transmission and receptions over one channel. A full-duplex device is capable of bi-directional network data transmissions at the same time. [8] A commonplace example of full duplex communications is a telephone call where both parties can communicate at the same time. [9]

2.3. Detail the steps, requests and protocols involved when you:

2.3.1. Start your app server

Firstly, you run the python app.py file. This runs on https://127.0.0.1:5000/. This creates a connection. Then it uses a GET request to get the website.

2.3.2. Visit it in a browser

When a user joins a room, a GET request is used to get the username and chatroom code. A POST request is then used to send the message that a user has joined a specific room.

2.3.3. Send a message

A POST request is sent and the message is displayed on the screen.

3. Link to Demo Video

https://youtu.be/0sZGHw5WgG8

4. References

[1] https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html

[2] https://www.ibm.com/docs/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.hala001/itcsocketconselsoc.htm

[3] https://www.tutorialspoint.com/unix_sockets/what_is_socket.htm

[4] CA304 lecture notes 10 - Sockets

[5] https://www.houkconsulting.com/2018/02/what-happens-when-visit-website/

[6]

https://www.ionos.com/digitalguide/websites/web-development/how-are-websites-accessed/

[7] https://stackoverflow.com/questions/4973622/difference-between-socket-and-websocket

[8] https://www.comms-express.com/infozone/article/half-full-duplex/

[9] https://mimomax.com/about-us/our-technologies/full-duplex-communications/

Got help with code from: https://www.youtube.com/watch?v=uJC8A_7VZOA