

## Assignment 3

ENE 419 – Computer Networks.

**Due: Check LMS for online submission and hardcopy in class**

### Programming assignment

(Streaming Protocol) You will develop an asynchronous web socket client. First, your client program is subscribed to a streaming channel of Binance server (<https://www.binance.com/en/markets>) that constantly transmits textual information of cryptocurrency.

The receiving information is so called market data of bitcoin, aka 'btc'. This is equivalent to buying/selling information about btc at the moment, it's same as the stock market. Let's try to receive the market data of btc@usdt (referred to as 'btcusdt', 'usdt' implies US dollars) and print them out.

For testing:

Try wscat to receive the data first and take a look.

How to install:

```
$ npm install -g wscat
$ wscat -c wscat -c wss://stream.binance.com:443/ws/btcusdt@depth5
connected (press CTRL+C to quit)
```

After connection, you will receive the data from the web socket:

```
<{"lastUpdateId":27547033276,"bids":[[{"16702.06000000","0.00069000"],["16701.76000000","0.00861000"],["16701.75000000","0.00149000"],["16701.32000000","0.04320000"],["16701.31000000","0.01571000"]],{"asks":[[{"16702.84000000","0.05546000"],["16702.85000000","0.02665000"],["16702.86000000","0.03324000"],["16703.34000000","0.01139000"],["16703.39000000","0.03000000"]]]}
<{"lastUpdateId":27547033811,"bids":[[{"16702.75000000","0.00087000"],["16702.74000000","0.02161000"],["16702.73000000","0.01909000"],["16702.63000000","0.00600000"],["16702.19000000","0.03417000"]],{"asks":[[{"16703.33000000","0.00893000"],["16703.34000000","0.00069000"],["16703.87000000","0.00649000"],["16703.92000000","0.00599000"],["16704.00000000","0.00337000"]]]}
...
```

Please refer to the following references for sample code:

For python3 <https://websockets.readthedocs.io/en/stable/>

You may need to know how to process json format in python.

If you choose to do this assignment in other languages, like C++, it will be also great.  
BIG BONUS!

### Submission:

1. Show your source code and output of each step in PDF and submit it to LMS. The output of step A, B, and C should just contain the last 10 (or 20) lines of each data file.
2. You do not have to create three separate python files to do step A, B, and C. You can do it in a single file separately (perhaps using a function). It's completely up to you.
3. Again, if you do this in C++, then there will be bonus.

### \*Step A)

Write a streaming client program receiving 'btcusdt@depth5' and a trade stream of 'btcusdt' information in the same stream connection. The data will be updated every 100-500 milliseconds. Your program will save the data into a text file (do not use shell redirect), called stream-data-binance.txt. You will add a new line of data as they arrive. The trade stream shows "p" and "q" values only. The stream-data-single.txt file should be like the following:

```
$ ./step_a.py
$ tail -n 10 stream-data-binance.txt
```

```
btcusdt@orderbook,bids:[["16702.06000000","0.00069000"],["16701.76000000","0.00861000"],["16701.75000000","0.00149000"],["16701.32000000","0.04320000"],["16701.31000000","0.01571000"],["16702.84000000","0.05546000"],["16702.85000000","0.02665000"],["16702.86000000","0.03324000"],["16703.34000000","0.01139000"],["16703.39000000","0.03000000"],["16702.84000000","0.05546000"],["16702.85000000","0.02665000"],["16702.86000000","0.03324000"],["16703.34000000","0.01139000"],["16703.39000000","0.03000000"]]
btcusdt@trade, p:16728.50000000,q:0.00626000
btcusdt@orderbook, ...
```

Please read Table 3: WebSocket Market Streams to figure out how to receive two or more streams. (<https://www.binance.com/en/support/faq/binance-options-api-interface-and-websocket-fe0be251ac014a8082e702f83d089e54>)

### \*Step B)

Modify your program from Step A receiving a similar btc data from the Upbit server (a Korean local stream server). Let's try.

```
$ wscat -c wss://api.upbit.com/websocket/v1
connected (press CTRL+C to quit)
```

After connection, your input to web socket:

```
[{"ticket":"UNIQUE_TICKET"}, {"type":"orderbook", "codes":["KRW-BTC"]}]]
```

After the input, you will receive the data from the web socket:

```
< {"type":"orderbook","code":"KRW-BTC","timestamp":1668495774440,"total_ask_size":11.400931790000001,"total_bid_size":3.1081664699999996,"orderbook_units":[{"ask_price":2.2855E7,"bid_price":2.2845E7,"ask_size":2.21720611,"bid_size":0.00164598}, {"ask_price":2.286E7,"bid_price":2.2844E7,"ask_size":1.07409042,"bid_size":0.39629050}, {"ask_price":2.2861E7,"bid_price":2.2843E7,"ask_size":0.13138600,"bid_size":0.07457449}, {"ask_price":2.2862E7,"bid_price":2.2842E7,"ask_size":1.11673137,"bid_size":0.41861210}, {"ask_price":2.2863E7,"bid_price":2.2841E7,"ask_size":0.695,"bid_size":0.23034224}, {"ask_price":2.2864E7,"bid_price":2.284E7,"ask_size":0.24999199,"bid_size":0.00146232}, {"ask_price":2.2865E7,"bid_price":2.283E7,"ask_size":0.06318533,"bid_size":0.50251088}, {"ask_price":2.2866E7,"bid_price":2.2827E7,"ask_size":0.46969651,"bid_size":0.08409668}, {"ask_price":2.2867E7,"bid_price":2.2826E7,"ask_size":1.68992953,"bid_size":0.00087619}, {"ask_price":2.2868E7,"bid_price":2.2825E7,"ask_size":0.86078502,"bid_size":0.00044389}, {"ask_price":2.2869E7,"bid_price":2.2821E7,"ask_size":0.01956419,"bid_size":0.058}, {"ask_price":2.2872E7,"bid_price":2.282E7,"ask_size":2.06000000,"bid_size":0.45073815}, {"ask_price":2.2874E7,"bid_price":2.2818E7,"ask_size":0.74898200,"bid_size":0.00043825}, {"ask_price":2.2875E7,"bid_price":2.2817E7,"ask_size":0.00354223,"bid_size":0.17213480}, {"ask_price":2.2876E7,"bid_price":2.2816E7,"ask_size":0.00084109,"bid_size":0.716}], "stream_type":"REALTIME"}
```

Your text data file is now called stream-data-upbit.txt. Print out the data after "orderbook\_units" only. You will add a new line of data as they arrive. It should be like the following:

```
$ ./step_b.py
$ tail -n 10 stream-data-upbit.txt
```

```
btckrw@orderbook,[{"ask_price":2.2855E7,"bid_price":2.2845E7,"ask_size":2.21720611,"bid_size":0.00164598},{"ask_price":2.286E7,"bid_price":2.2844E7,"ask_size":1.07409042,"bid_size":0.39629050},{"ask_price":2.2861E7,"bid_price":2.2843E7,"ask_size":0.13138600,"bid_size":0.07457449},{"ask_price":2.2862E7,"bid_price":2.2842E7,"ask_size":1.11673137,"bid_size":0.41861210},{"ask_price":2.2863E7,"bid_price":2.2841E7,"ask_size":0.695,"bid_size":0.23034224},{"ask_price":2.2864E7,"bid_price":2.284E7,"ask_size":0.24999199,"bid_size":0.00146232},{"ask_price":2.2865E7,"bid_price":2.283E7,"ask_size":0.06318533,"bid_size":0.50251088},{"ask_price":2.2866E7,"bid_price":2.2827E7,"ask_size":0.46969651,"bid_size":0.08409668},{"ask_price":2.2867E7,"bid_price":2.2826E7,"ask_size":1.68992953,"bid_size":0.00087619},{"ask_price":2.2868E7,"bid_price":2.2825E7,"ask_size":0.86078502,"bid_size":0.00044389},{"ask_price":2.2869E7,"bid_price":2.2821E7,"ask_size":0.01956419,"bid_size":0.058},{"ask_price":2.2872E7,"bid_price":2.282E7,"ask_size":2.06000000,"bid_size":0.45073815},{"ask_price":2.2874E7,"bid_price":2.2818E7,"ask_size":0.74898200,"bid_size":0.00043825},{"ask_price":2.2875E7,"bid_price":2.2817E7,"ask_size":0.00354223,"bid_size":0.17213480},{"ask_price":2.2876E7,"bid_price":2.2816E7,"ask_size":0.00084109,"bid_size":0.716}]
btckrw@orderbook,[{"ask_price":2.2855E7,"bid_price":2.2846E7,"ask_size":2.21720611,"bid_size":0.875},{"ask_price":2.286E7,"bid_price":2.2845E7,"ask_size":1.07409042,"bid_size":0.00164598},{"ask_price":2.2861E7,"bid_price":2.2844E7,"ask_size":0.13138600,"bid_size":0.39629050},{"ask_price":2.2862E7,"bid_price":2.2843E7,"ask_size":1.11673137,"bid_size":0.07457449},{"ask_price":2.2863E7,"bid_price":2.2842E7,"ask_size":0.695,"bid_size":0.41861210},{"ask_price":2.2864E7,"bid_price":2.2841E7,"ask_size":0.24999199,"bid_size":0.23034224},{"ask_price":2.2865E7,"bid_price":2.284E7,"ask_size":0.06318533,"bid_size":0.00146232},{"ask_price":2.2866E7,"bid_price":2.283E7,"ask_size":0.46969651,"bid_size":0.50251088},{"ask_price":2.2867E7,"bid_price":2.2827E7,"ask_size":1.68992953,"bid_size":0.08409668},{"ask_price":2.2868E7,"bid_price":2.2826E7,"ask_size":0.86078502,"bid_size":0.00087619},{"ask_price":2.2869E7,"bid_price":2.2825E7,"ask_size":0.01956419,"bid_size":0.00044389},{"ask_price":2.2872E7,"bid_price":2.2821E7,"ask_size":2.06000000,"bid_size":0.058},{"ask_price":2.2874E7,"bid_price":2.282E7,"ask_size":0.74898200,"bid_size":0.45073815},{"ask_price":2.2875E7,"bid_price":2.2818E7,"ask_size":0.00354223,"bid_size":0.00043825},{"ask_price":2.2876E7,"bid_price":2.2817E7,"ask_size":0.00084109,"bid_size":0.17213480}]
btckrw@orderbook, ...
```

\*Step C) This is a kinda BONUS.

You will take the codes from Step A and B to combine two stream data into a single file. You need to figure out how to open two stream connections for Binance and Upbit simultaneously. You may use thread (multi-thread or multi-process) to run two receiving clients, however, the output file must be a single text file: stream-data-multi.txt. The order of the streams may be random.

```
$ ./step_c.py
$ tail -n 10 stream-data-multi.txt
```

```
btcusdt@orderbook,bids:[["16702.06000000","0.00069000"],["16701.76000000","0.00861000"],["16701.75000000","0.00149000"],["16701.32000000","0.04320000"],["16701.31000000","0.01571000"]],asks:[["16702.84000000","0.05546000"],["16702.85000000","0.02665000"],["16702.86000000","0.03324000"],["16703.34000000","0.01139000"],["16703.39000000","0.03000000"]],asks:[["16702.84000000","0.05546000"],["16702.85000000","0.02665000"],["16702.86000000","0.03324000"],["16703.34000000","0.01139000"],["16703.39000000","0.03000000"]]
btcusdt@trade, p:16728.50000000,q:0.00626000
btckrw@orderbook,[{"ask_price":2.2855E7,"bid_price":2.2846E7,"ask_size":2.21720611,"bid_size":0.875},{"ask_price":2.286E7,"bid_price":2.2845E7,"ask_size":1.07409042,"bid_size":0.00164598},{"ask_price":2.2861E7,"bid_price":2.2844E7,"ask_size":0.13138600,"bid_size":0.39629050},{"ask_price":2.2862E7,"bid_price":2.2843E7,"ask_size":1.11673137,"bid_size":0.07457449},{"ask_price":2.2863E7,"bid_price":2.2842E7,"ask_size":0.695,"bid_size":0.41861210},{"ask_price":2.2864E7,"bid_price":2.2841E7,"ask_size":0.24999199,"bid_size":0.23034224},{"ask_price":2.2865E7,"bid_price":2.284E7,"ask_size":0.06318533,"bid_size":0.00146232},{"ask_price":2.2866E7,"bid_price":2.283E7,"ask_size":0.46969651,"bid_size":0.50251088},{"ask_price":2.2867E7,"bid_price":2.2827E7,"ask_size":1.68992953,"bid_size":0.08409668},{"ask_price":2.2868E7,"bid_price":2.2826E7,"ask_size":0.86078502,"bid_size":0.00087619},{"ask_price":2.2869E7,"bid_price":2.2825E7,"ask_size":0.01956419,"bid_size":0.00044389},{"ask_price":2.2872E7,"bid_price":2.2821E7,"ask_size":2.06000000,"bid_size":0.058},{"ask_price":2.2874E7,"bid_price":2.282E7,"ask_size":0.74898200,"bid_size":0.45073815},{"ask_price":2.2875E7,"bid_price":2.2818E7,"ask_size":0.00354223,"bid_size":0.00043825},{"ask_price":2.2876E7,"bid_price":2.2817E7,"ask_size":0.00084109,"bid_size":0.17213480}]
btcusdt@orderbook, ...
btckrw@orderbook, ...
btcusdt@trade, ...
```