Lab 9: MQTT

50.012 Networks (2018)

Hand out: 21^{st} Nov (C1) / 23^{rd} Nov (C2)
Hand in: 27^{th} Nov (C1) / 29^{th} Nov (C2)

1 Overview

- In this execise, we will write our own MQTT client
- This exercise is using the tutorial at `https://sakshambhatla.wordpress.com/2014/08/11/simple-mqtt-broker-and-client-in-python/`
- Brief documentation of the paho library is available at `https://pypi.python.org/pypi/paho-mqtt`
- A MQTT server is available on port 1337 at `scy-phy.net`

2 Setup

- During the setup, you should be connected to `SUTD_Student` to have Internet access.
- Use `sudo apt-get install paho-mqtt` to install the paho python library
  - If you would like to have a local broker for testing, install mosquitto with `sudo apt install mosquitto`

3 Simple Subscriber and Publisher Client

- Start by implementing the client from the tutorial, and try to read content posted to the `hello/world` topic.
  - Your client should connect to the broker, and then print all received messages to the stdout
- Use the same or a different script to publish to the same `hello/world` topic.
  - Send a few test messages - are they displayed by your subscriber client?

3.1 Persistence

- By default, your client will miss messages posted to the topic if it is not running at that time
- By default, your client will create a random ID. Change the call to `Client()` to use a static ID of your choice to enable persistent sessions
- The `clean_session` argument to the `Client()` constructor can be used to change the persistence behaviour
  - Set `clean_session` to `False`, the broker should deliver missed messages to you when you connect next time
  - This will only work for messages that were published using the "persistent" flag

3.2 Subscribing to multiple topics

- Modify your client to subscribe to a second topic of your choice
- Start your subscriber client, and use the publisher client to post a message to that topic
  - You should now see messages posted to both topics in your subscriber client stdout
- Test the `*` and `#` wildcard as discussed in the lecture

4 Interactive client

- Use the provided clientSkeleton.py to write a more interactive "chat" client
  - Set the username variable to a name of your choice
  - Fill the TODO gaps to enable publishing and subscription
- The skeleton only allows to send "hello" messages to other users - extend it to allow to send arbitrary messages
- Add a function to subscribe and unsubscribe from arbitrary channels
- Add persistence as before to allow reception of messages that were missed while the client was down
- Use the client to exchange messages with others!

5 Optional extensions (not required)

- MQTT exchanges binary data, not ascii.
- Extend your client to interpret data from a specific topic as binary data (e.g. as an image)
- Allow your client to publish images to that topic, and receive images from that topic

6 What to Hand in

6.1 eDimension submission:

- Please hand in your modified `clientSkeleton.py` file.
- Make sure to include your name in the header of the file
- Your client should do the following:
  - Automatically subscribe to a number of topics
  - Allow manual subscription/unsubscription
  - Allow manual sending of arbitrary messages to other users
  - Display of incoming messages for your user
  - Allow display of missed messaged since last run of client

6.2 Checkoff:

- No checkoff required if you submitted your reply

NOTE: Please note that the due dates for submission are 27^{th} Nov for Cohort 1 and 29^{th} Nov for Cohort 2. We will look at the timestamps of your submission to evaluate timely / late submission.