

Practical IoT Activity

Advanced Computer Networks, CSC 454/554 and CYEN 489-001, Spring 2020

Total points available: 100 (+20 possible bonus points)

Due date: All materials are due by 11:59 PM on May 15, 2020

Summary

In this activity, you will implement a class-wide Internet of Things system within the constraints placed on us by being unable to meet in person. This document describes several tasks. You will need to complete each of these tasks to get full credit for the assignment. After the required tasks are listed some possibilities for extra credit.

For this assignment, you will create a publisher program that publishes your current CPU usage percentage to a broker server using the MQ Telemetry Transport (MQTT) protocol under a unique topic. You will also create a client/subscriber program that subscribes to your own topic on the broker as well as topics published by several other students. Your client will record its observations in memory within the Python program and plot them using Matplotlib in real time.

Preparation

1. If not already installed, install a Python 3 environment
 - a. Whatever environment you choose, you'll need numpy, scipy, and matplotlib
 - b. If you don't have a preference, Anaconda is good:
<https://www.anaconda.com/products/individual>
2. Install paho-mqtt in your Python 3 environment
 - a. pip install paho-mqtt
3. If not already installed, install psutil in your Python 3 environment
 - a. pip install psutil
4. Download the zip file containing the source files from Moodle.
5. Find the documentation for the Paho MQTT client here: <https://pypi.org/project/paho-mqtt/>

Set up publisher

6. Open the mqttPublisher.py file
7. Modify the topic1 variable to set your unique topic name. Use the following format:

"Lastname/x"

where x is up to you.

8. Change the delay if you would like to. Do not set it to be less than 2 seconds.

Set up subscriber

9. Open the mqttSubscriber.py file

10. In the `on_connect()` callback function, add `client.subscribe()` calls for each topic you want to subscribe to. A list of available topics from the class will be published somewhere; be on the look-out for more information on this.
11. You will need to subscribe to your own topic(s) as well as **four other topics** from other students.
12. Carefully read through and understand the `on_message()` callback function. This function is called every time the subscriber receives a message from the broker. You will need to modify this function to handle each topic you are subscribed to.
13. In the beginning of the main block, empty lists are created to store time stamps and CPU percentages received from the broker. You will need to add a time list and a sensor value list for each topic you are subscribed to. Each empty list will need to be passed to the `userdata` parameter when you instantiate the `mqtt.Client`.
14. Modify the plotting functionality to plot each set of sensor data you are receiving from each other user you are subscribed to. Each set of sensor data can be plotted on its own figure or on subplots within one figure. Use informative labels to make sure it is obvious who the sensor data is coming from and what sensor data it is. (For most of you, it will just be CPU usage percentage, but some may implement other sensors. If you subscribe to data from those sensors, make it clear what you are plotting.)
15. You can make any changes to the plotting procedure as long as the baseline functionality already present is maintained.
16. Modify the infinite while loop to proceed for a finite amount of time.

Get everything running

17. Use screen recording software or your phone camera to start recording your screen.
 - a. Make sure the code you are running and your plots are visible during the recording.
18. In one Python interpreter, start running your publisher. You can start this along with the rest of the class and leave it running for a while.
19. In another interpreter, start running your subscriber.
20. Once the subscriber program has finished, save all your figures.

What to submit

21. Prepare and upload a zip file containing the following materials to Moodle, under the IoT activity assignment:
 - a. Your two modified Python source files
 - b. Your figure image files
 - c. Your screen recording of the subscriber program running and generating its plots.

Bonus

22. **BONUS:** Expand this activity in some way. Up to 20 bonus points are available depending on my judgement of how much extra work you did. A non-exhaustive list of things you could do is given below. You could do many of these in combination for additive bonus, up to a maximum of 20 points total. The points listed after each item are approximate and subject to change based on the details of your implementation.

- a. Implement your client and server in one program using threading (or multiprocessing, or whatever else) (+5 points)
- b. Use a public weather API to gather weather data for your area. Publish this data to a unique topic on the broker and display it on your client interface. Make sure your messages provide enough information to infer things like location, quantity being measured, and units. Use '@@@' for your string delimiter. (+10 points)
- c. Use a Raspberry Pi hooked up to a sensor (or sensors) of your choice to observe some phenomenon (or phenomena) and publish the data to a unique topic on the broker. Your message should allow the rest of us to identify what you are measuring and what the units are. Use '@@@' for your string delimiter. (+10 points)
- d. Use separate devices or virtual machines for your client and server (+2 points)
- e. If you use separate devices for your client and server, use a networking technology other than infrastructure mode WiFi (802.11a/b/g/n/ac, etc) or ethernet to connect them. Examples: Ad-hoc or mesh WiFi, Bluetooth, 802.15.4 or ZigBee if you have the hardware (+8 points)

Other remarks

This activity requires the real-time sharing of data among students. The best way to do that is to attend the class-time Zoom meeting and coordinate with the rest of the class there. However, if you would prefer to work with a small group outside of class instead to share your data, you are free to do so. Keep in mind that you will need to interact with at least four other students to receive full credit.

Acknowledgments

Thanks to Christopher Smith for getting me set up with the TechXplore cluster for the broker and for providing me with a handy starting point for scripts for working with Paho.