

This part implements the data transfer between PCs using MQTT protocol in P/S mode.

## 1 Install

Src: `wget http://mosquitto.org/files/source/mosquitto-1.4.5.tar.gz`

(You can also download newer versions in that website)

Then extract the file, after that, in terminal, cd to the directory, type:

`make`

`sudo make install`

## 2. Possible Problems

【1】 can't find openssl/ssl.h

`sudo apt-get install libssl-dev`

【2】 can't find ares.h

`sudo apt-get install libc-ares-dev`

【3】 can't find uuid/uuid.h

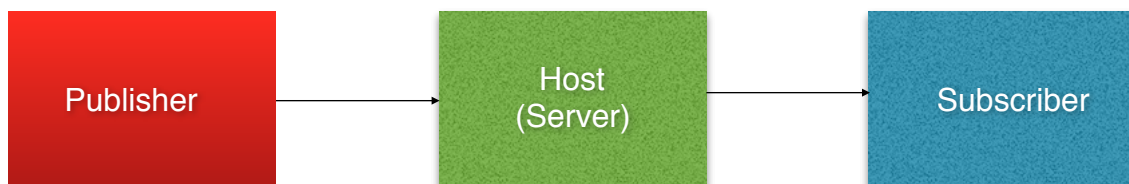
`sudo apt-get install uuid-dev`

【4】 error while loading shared libraries: libmosquitto.so.1: cannot open shared object file: No such file or directory

`sudo ln -s /usr/local/lib/libmosquitto.so.1 /usr/lib/libmosquitto.so.1`

`sudo ldconfig`

## 3. Publisher/Subscriber Mode



There is a machine acting as the Host, we could call it central server.

Publisher would publish the information towards host, which will board cast the info to all subscribers listening to the same channel(topic).

Just like we done in lab3.

So at least we need 3 machines. (Actually, we could make it using 3 terminals in one PC for testing)

## 4. Commands

1.In `"/usr/local/sbin"` you can see the file `"mosquitto"`.

2.Start the server:

`mosquitto -c /etc/mosquitto/mosquitto.conf.example`

3.Subscribe:

`mosquitto_sub -h ... -p ... -t ...`

-h: host's ip, default value is localhost  
-p: port, default value is 1883  
-t : topic

4.Publish :

mosquitto\_pub -h ... -p ... -t ... -m ...  
-p, -t, -h are similar  
-m: message

## 5. Test

A simple example could be shown in one single PC.

3 Terminals needed.

Terminal1: The server, type "mosquitto -c /etc/mosquitto/mosquitto.conf.example"

Terminal2: The subscriber, type "mosquitto\_sub -t XXX"

Terminal3: The publisher, type "mosquitto\_pub -t XXX -m ...."

Then you can see results in window.

## 6. Format

The basic commands can only transfer simple string messages.

For our project , we can set the message in the format of :

"name\_value"

Then using output redirect to save the message in local txt file. Then we could extract the value of each machine to do something more. For example, make a website page to supervise and show all the data of connected PCs.

That is to say, originally, the msg received was shown at terminal, but using commands like this:

" mosquitto\_sub -t XXX > 1.txt "

The msg would be saved in "1.txt" by output redirecting.