物聯網通訊與安全

第2章 物聯網通訊協定
IoT Protocols

蘇維宗 (Wei-Tsung Su)
suwt@au.edu.tw
564D

歷史版本

版本	說明	日期	負責人
v1.0	初版	2019/02/18	蘇維宗

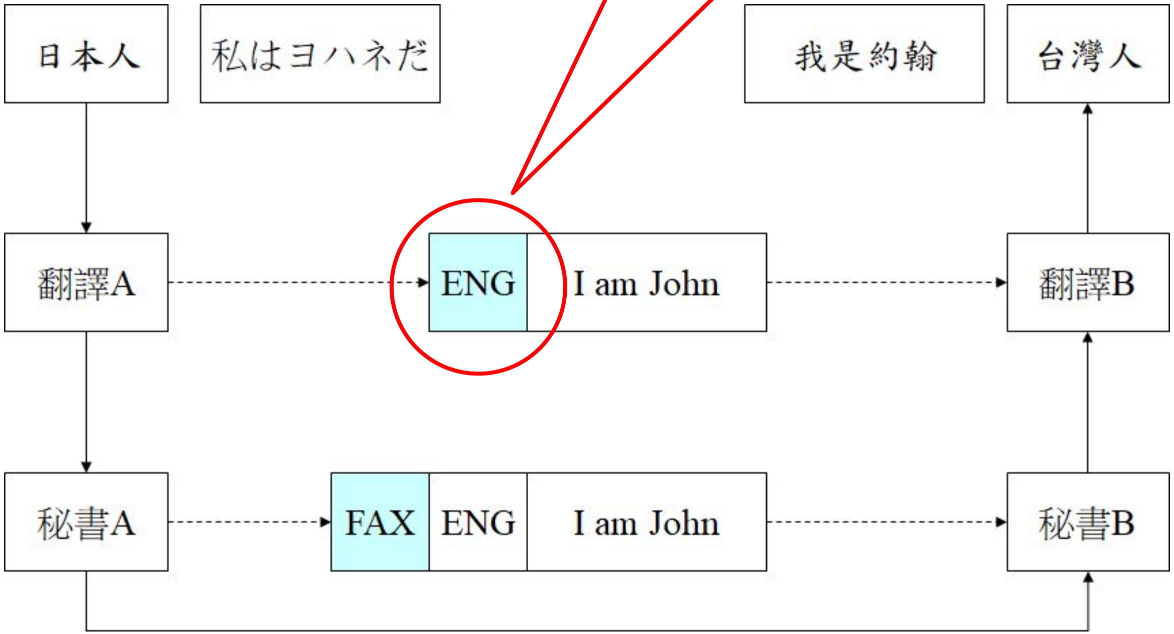
Protocols

What's Protocols?

In computer networking, **protocols (通訊協定)** define

- format and order of messages exchamged among network entities
- actions taken on the transmission and receipt of messages
- other events

Why Need Protocols?



Header (標頭) includes the information indicating how this message can be handled.

TCP/IP Network Layer Model

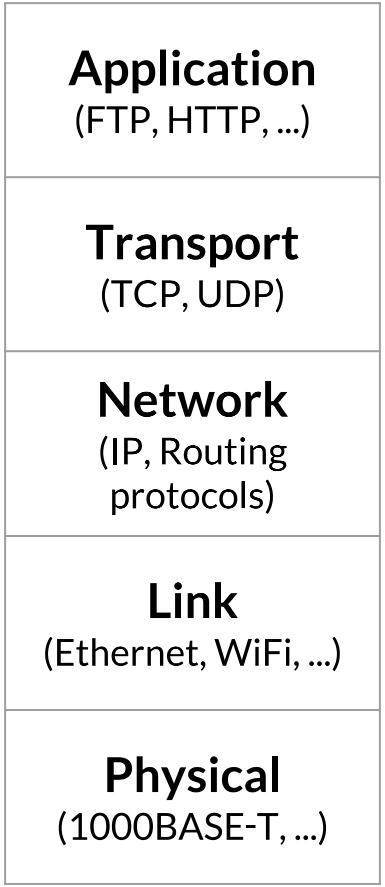
Application: supporting network applications

Transport: process-process data transfer

Network: routing of datagrams from source to destination

Link: data transfer between neighboring network elements

Physical: bits “on the wire”



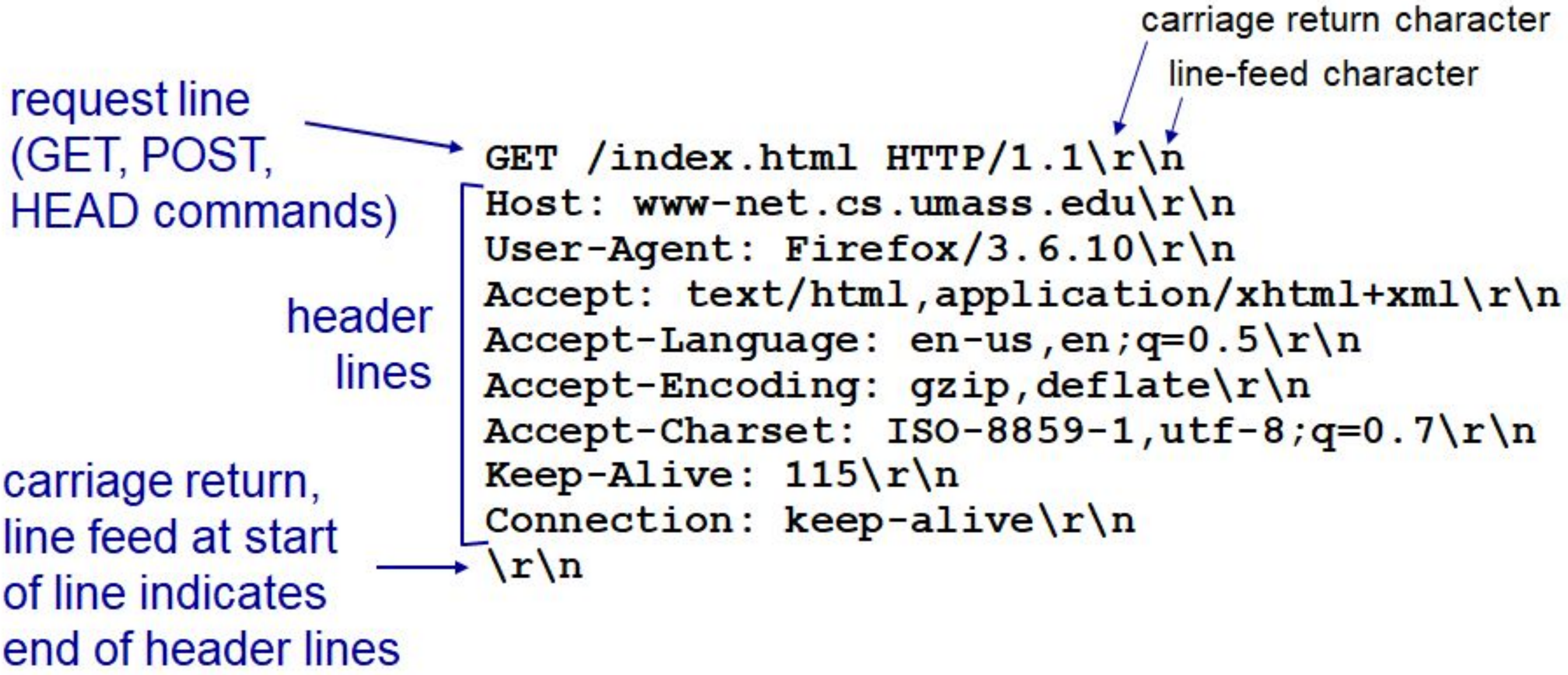
Application
(FTP, HTTP, ...)

Transport
(TCP, UDP)

Network
(IP, Routing protocols)

Link
(Ethernet, WiFi, ...)

Physical
(1000BASE-T, ...)



carriage return character

line-feed character

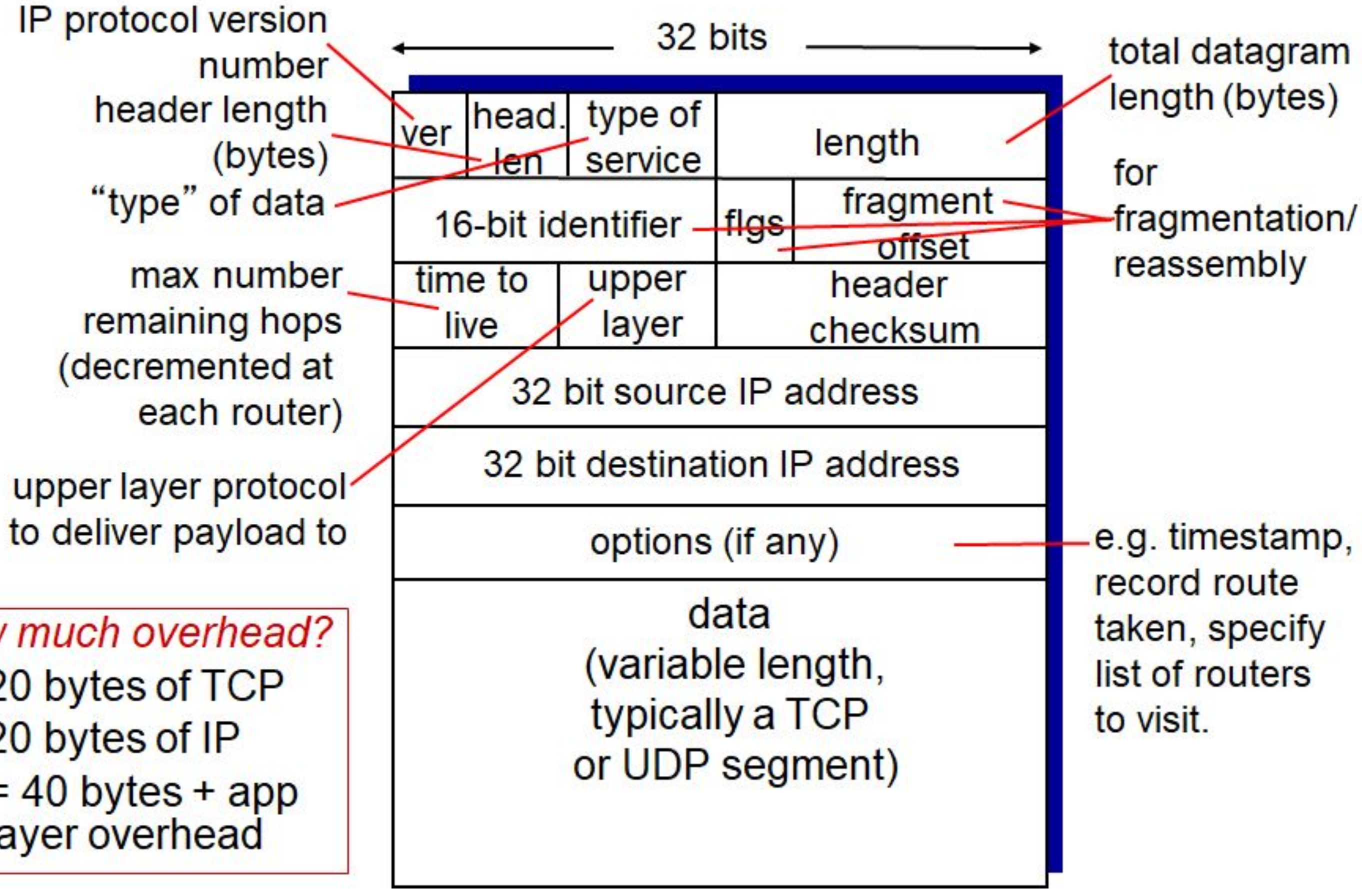
request line
(GET, POST,
HEAD commands)

header
lines

carriage return,
line feed at start
of line indicates
end of header lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

Case Study: HTTP (Application Layer)


IP protocol version number
header length (bytes)
"type" of data
max number remaining hops (decremented at each router)
upper layer protocol to deliver payload to
32 bits
ver
head. len
type of service
length
16-bit identifier
flgs
fragment offset
time to live
upper layer
header checksum
32 bit source IP address
32 bit destination IP address
options (if any)
data (variable length, typically a TCP or UDP segment)
total datagram length (bytes)
for fragmentation/ reassembly
e.g. timestamp, record route taken, specify list of routers to visit.

how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Case Study: IPv4 (Network Layer)

IoT Protocols

IoT Protocols for Data Exchange

Constrained Application Protocol (CoAP)

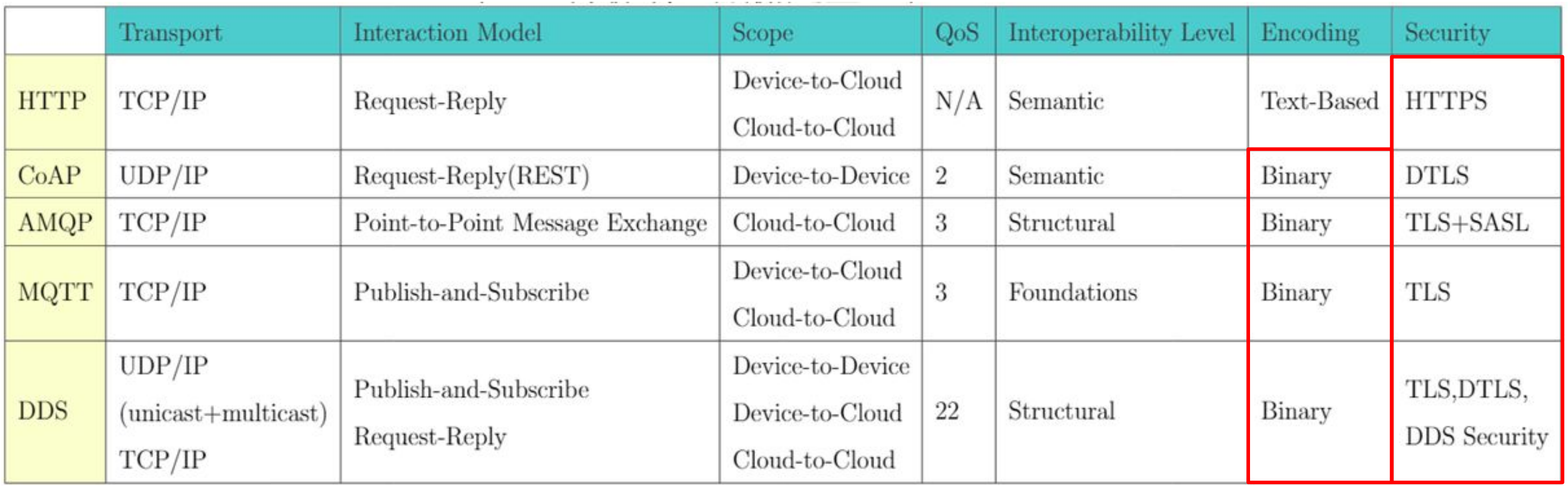
Extensibel Messaging and Presence Protocol (XMPP)

Message Queue Telemetry Transport (MQTT)

Advanced Message Queuing Protocol (AMQP)

Data Distributed Service (DDS)

Comparison



	Transport	Interaction Model	Scope	QoS	Interoperability Level	Encoding	Security
HTTP	TCP/IP	Request-Reply	Device-to-Cloud Cloud-to-Cloud	N/A	Semantic	Text-Based	HTTPS
CoAP	UDP/IP	Request-Reply(REST)	Device-to-Device	2	Semantic	Binary	DTLS
AMQP	TCP/IP	Point-to-Point Message Exchange	Cloud-to-Cloud	3	Structural	Binary	TLS+SASL
MQTT	TCP/IP	Publish-and-Subscribe	Device-to-Cloud Cloud-to-Cloud	3	Foundations	Binary	TLS
DDS	UDP/IP (unicast+multicast) TCP/IP	Publish-and-Subscribe Request-Reply	Device-to-Device Device-to-Cloud Cloud-to-Cloud	22	Structural	Binary	TLS,DTLS, DDS Security

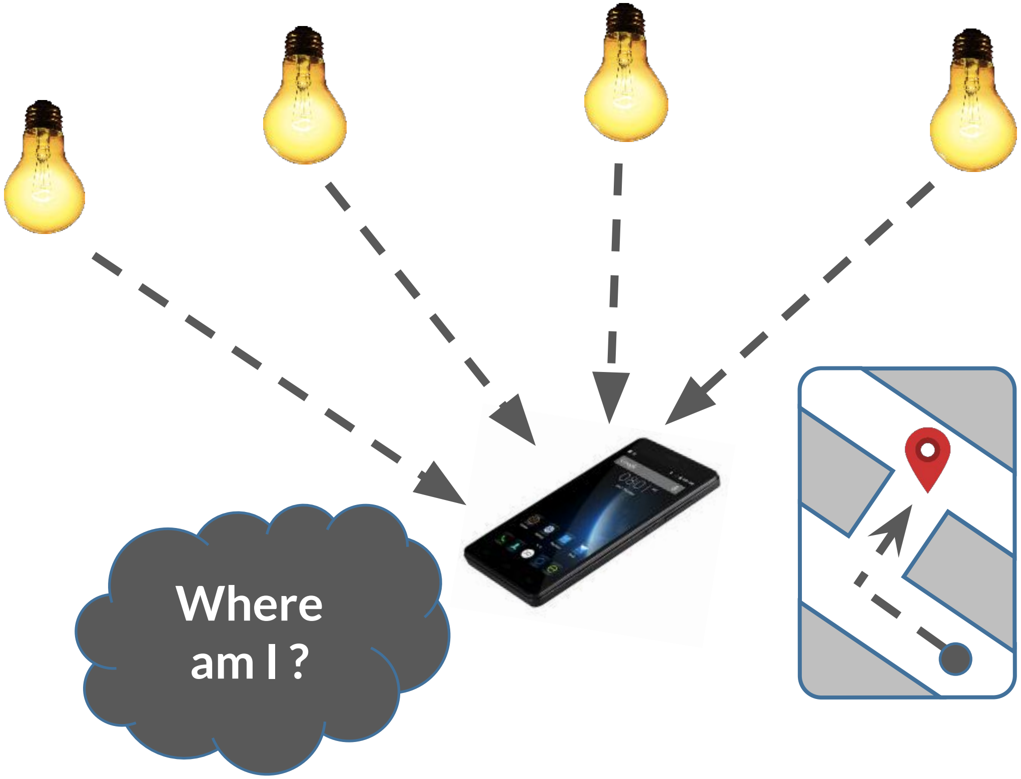
MQTT

Message Queue Telemetry Transport

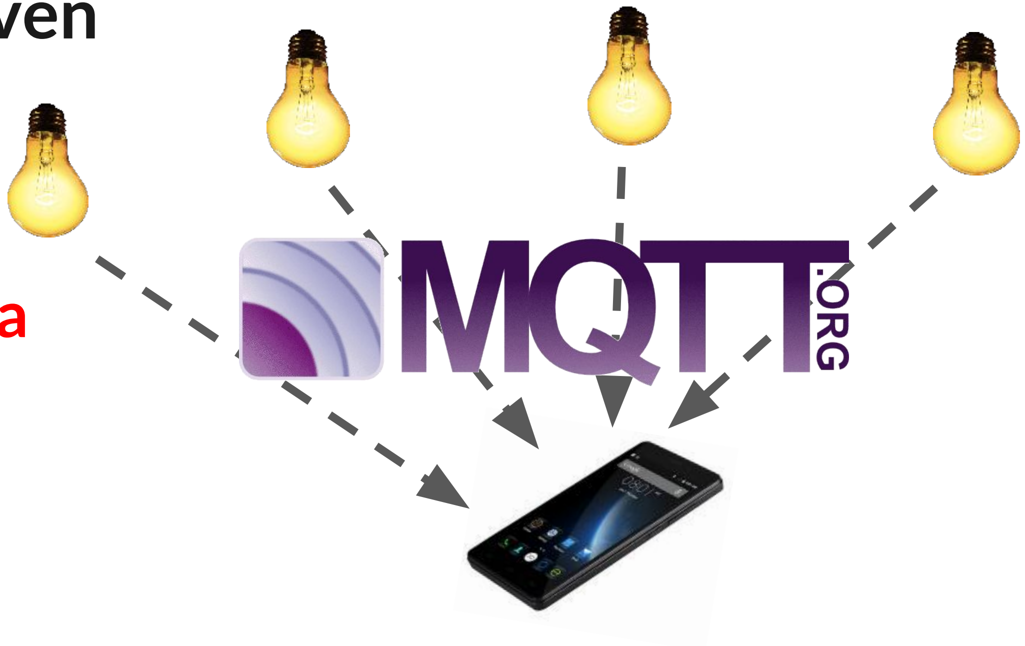
IoT Applications

Data-Driven Application

For example, visual light communication (VLC) can be used to provide indoor localization.



Challenges of Data-Driven IoT Applications



How to obtain sensor data from VLC nodes?

MQTT can be one of the solutions.

1999

Andy Stanford-Clark與Arlen Nipper發表MQTT v1.0。

2011

Facebook Messenger採用MQTT作為訊息交換協定。

2013

MQTT v3.1成為OASIS輕量化訊息交換的公開標準。

History of MQTT

Watson IoT
(IBM)

IoT Core
(Google)

Azure IoT
(Microsoft)

AWS IoT
(Amazon)

Lightweight

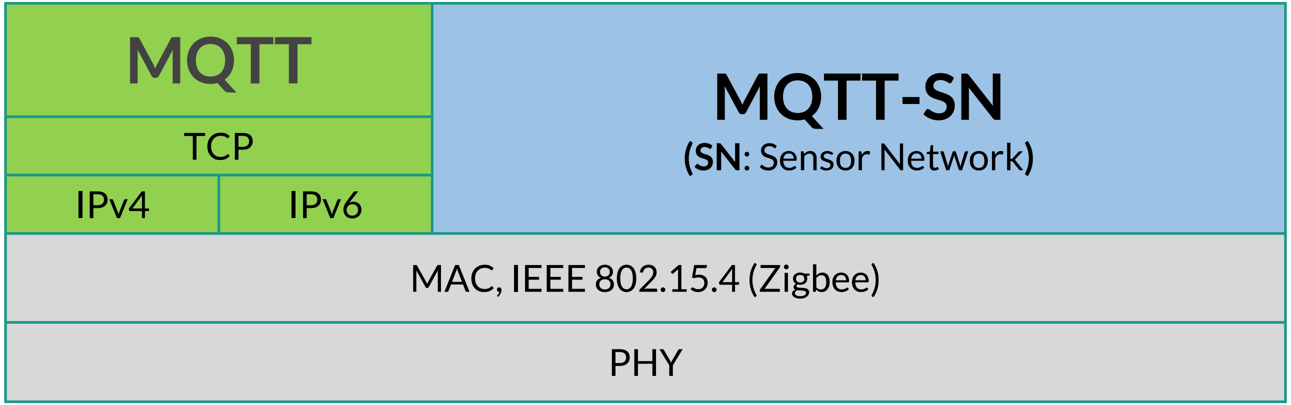
MQTT.ORG

Widely
Accepted

Why using MQTT?

MQTT is based on TCP/IP

MQTT-SN is for devices without TCP/IP



MQTT

TCP

IPv4

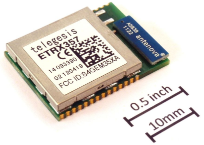
IPv6

MQTT-SN

(**SN**: Sensor Network)

MAC, IEEE 802.15.4 (Zigbee)

PHY



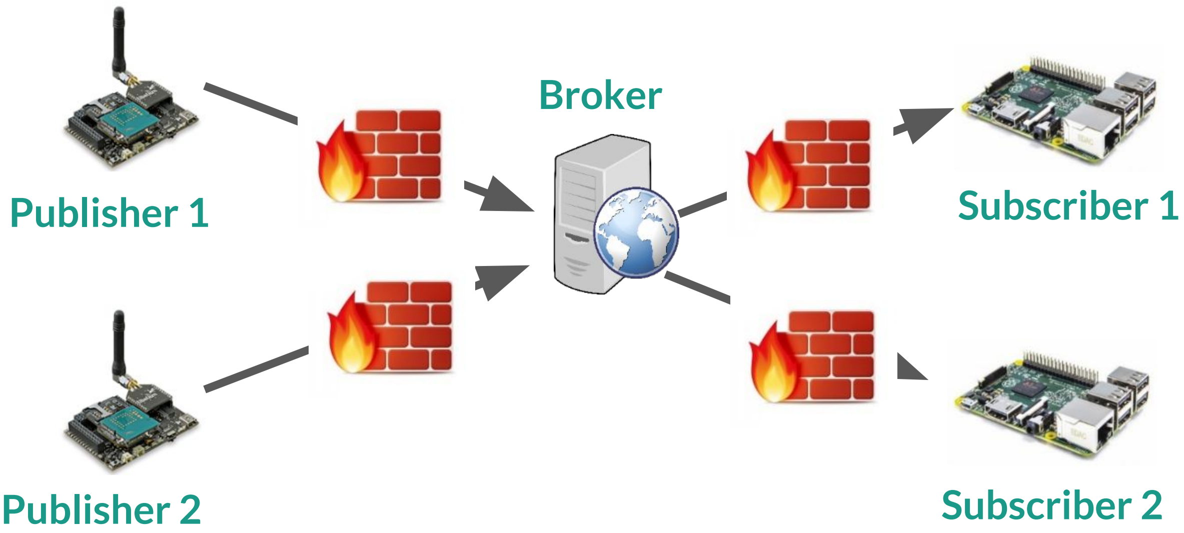
MQTT vs. MQTT-SN

Roles in MQTT

Publisher
Data provider

Broker
Data forwarder

Subscriber
Data Consumer



Publisher 1

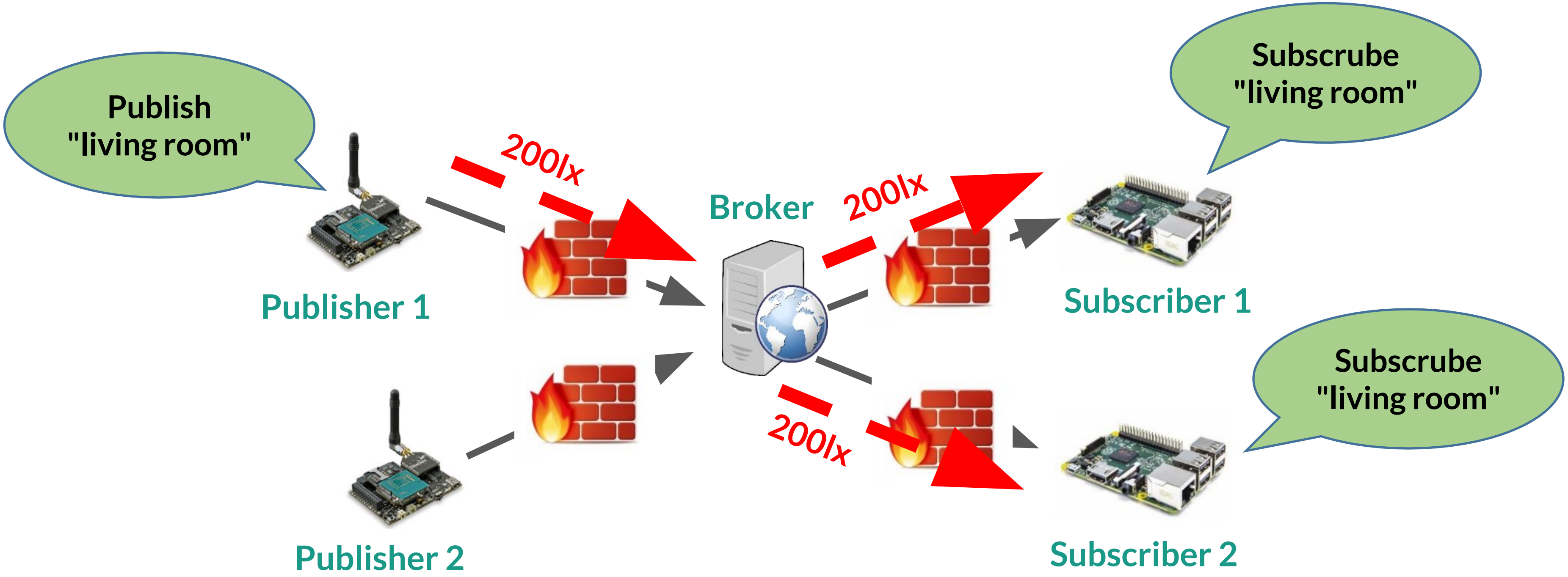
Publisher 2

Broker

Subscriber 1

Subscriber 2

Based on **publish-subscribe** model



Features of MQTT

MQTT Supports Quality of Service (QoS)

QoS Level 0

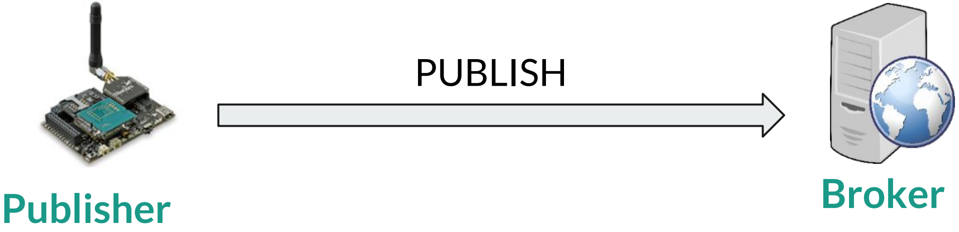
Only Once

QoS Level 1

At Least Once

QoS Level 2

Exactly Once



MQTT Supports Quality of Service (QoS)

QoS Level 0

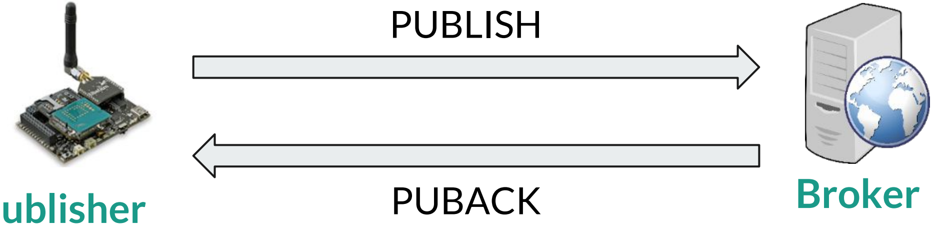
Only Once

QoS Level 1

At Least Once

QoS Level 2

Exactly Once



MQTT Supports Quality of Service (QoS)

QoS Level 0

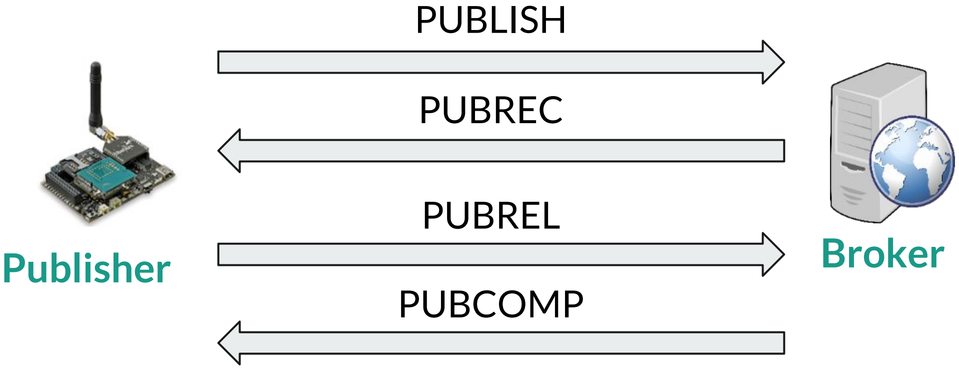
Only Once

QoS Level 1

At Least Once

QoS Level 2

Exactly Once



PUBLISH

PUBREC

PUBREL

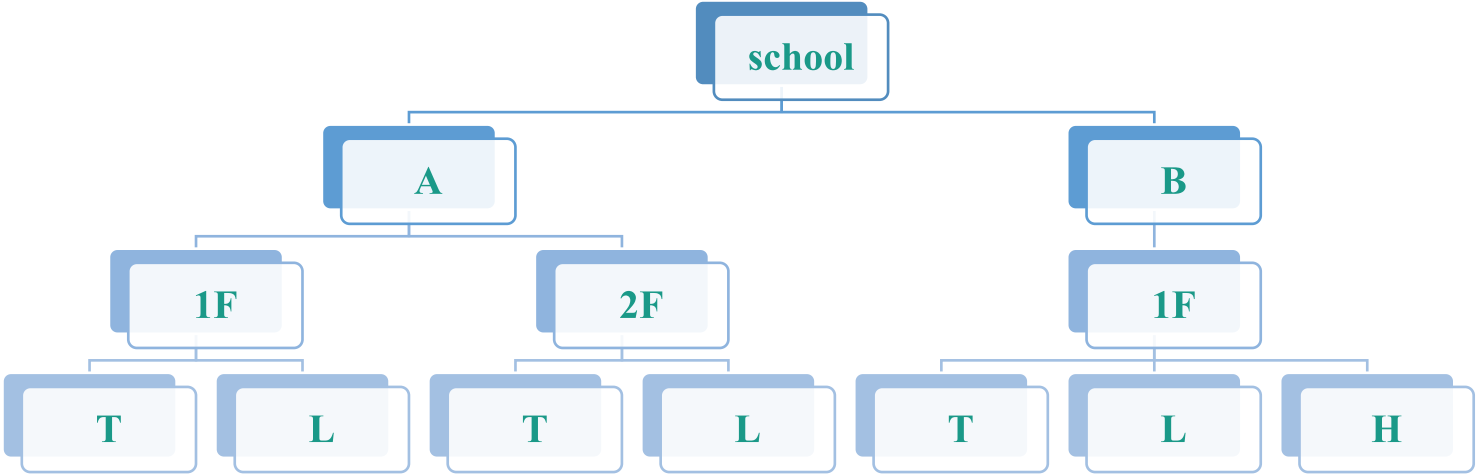
PUBCOMP

Publisher

Broker

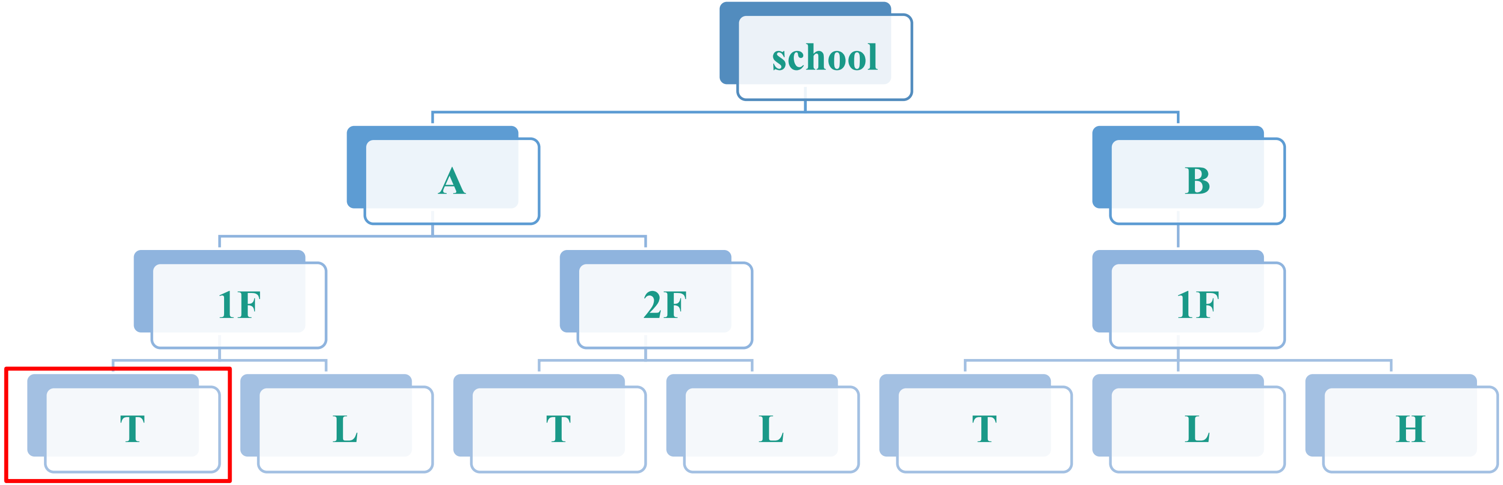
MQTT Topic

MQTT topic can be described as a tree structure. For example,



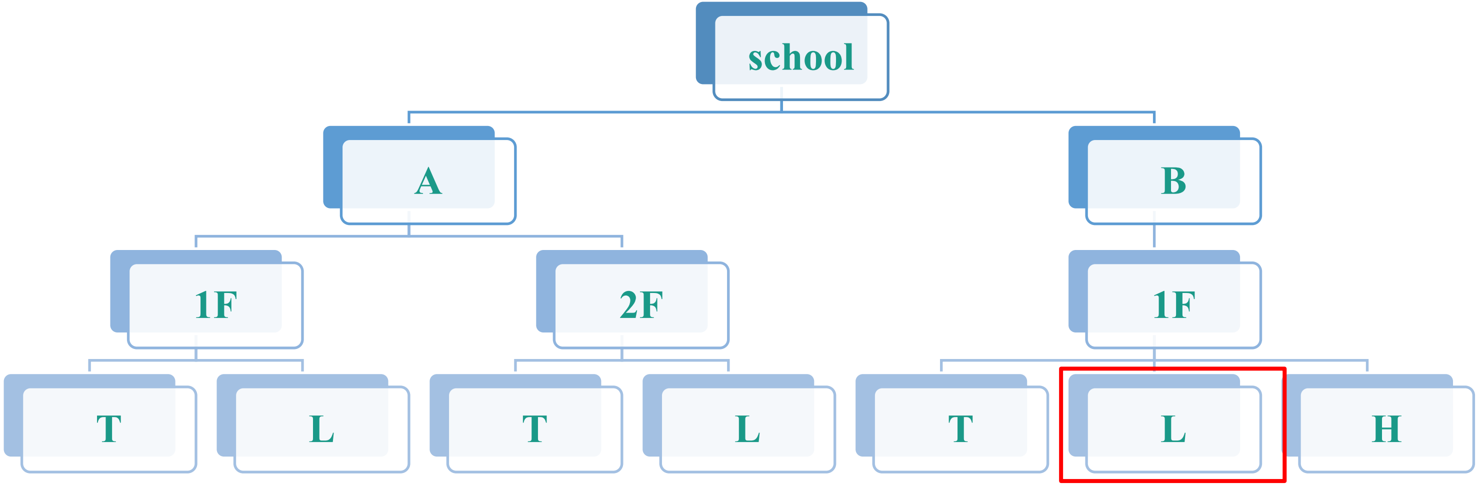
MQTT Topic (con't)

`school/A/1F/T`



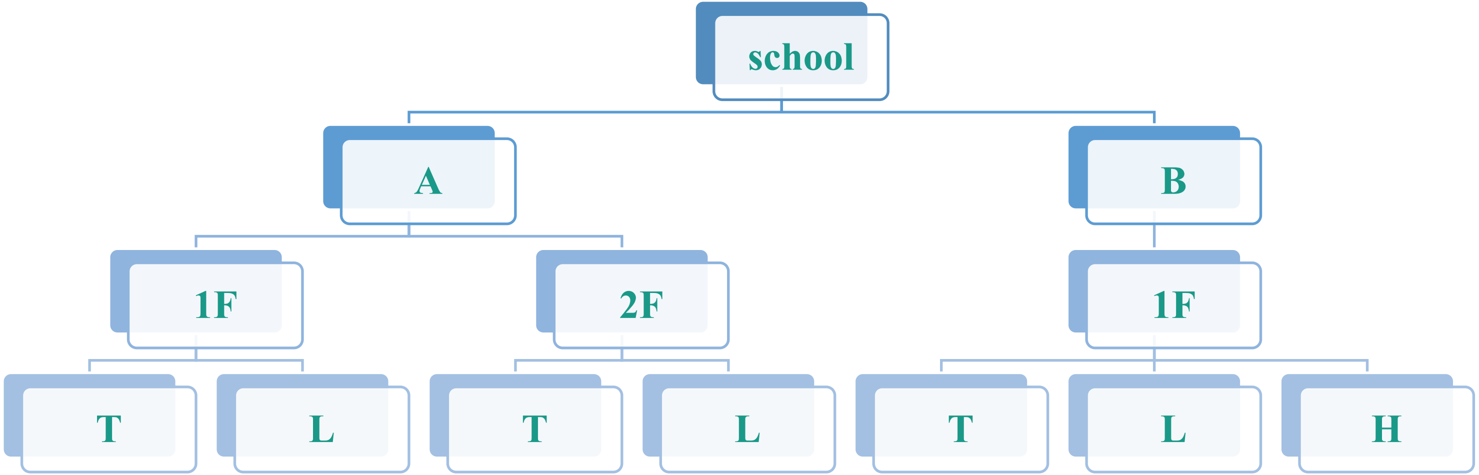
MQTT Topic (con't)

```
school/B/1F/L
```



Wildcard in MQTT Topic

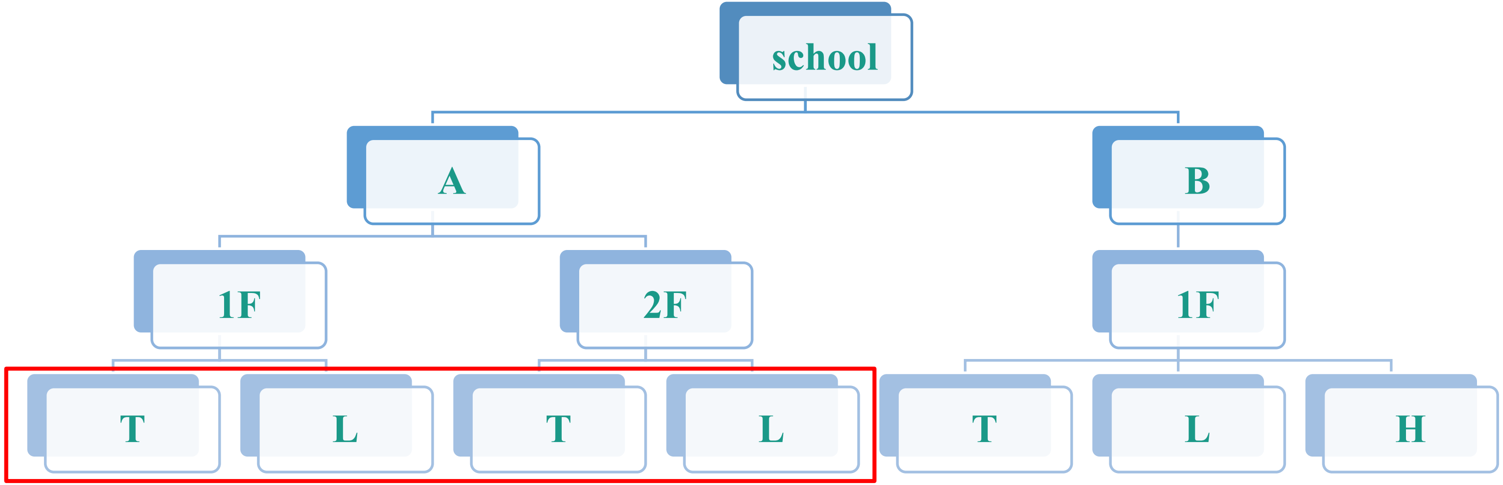
Subscribers can subscribe multiple topics using wildcard in MQTT topic



Wildcard in MQTT Topic (#)

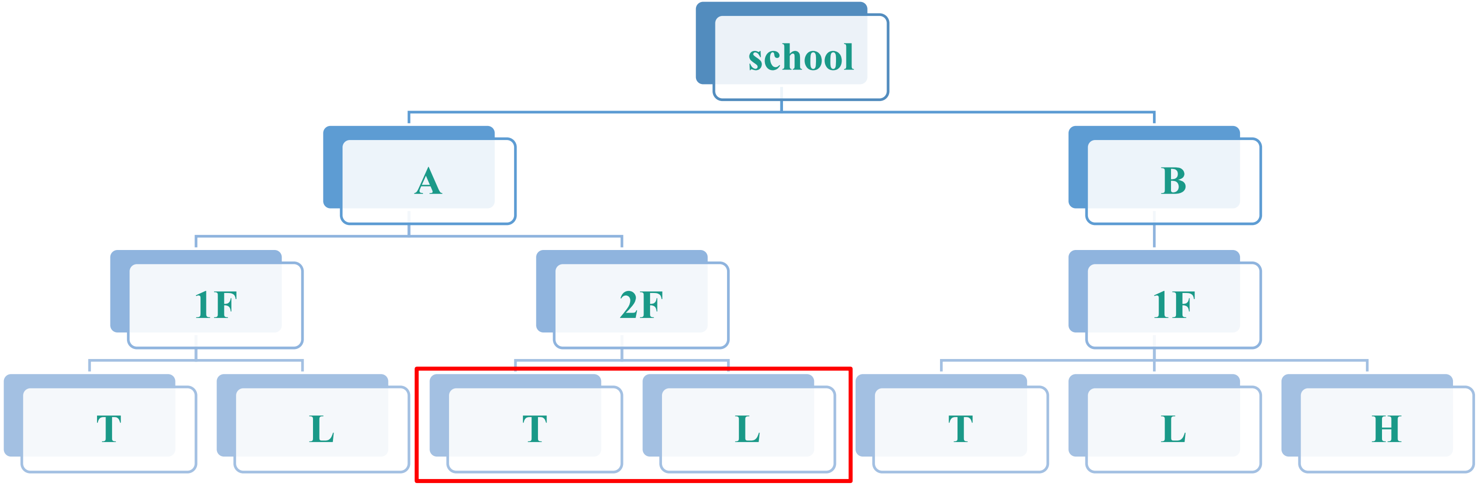
Remark:
Wildcard # can only use once in the tail of topic.

```
school/A/#
```



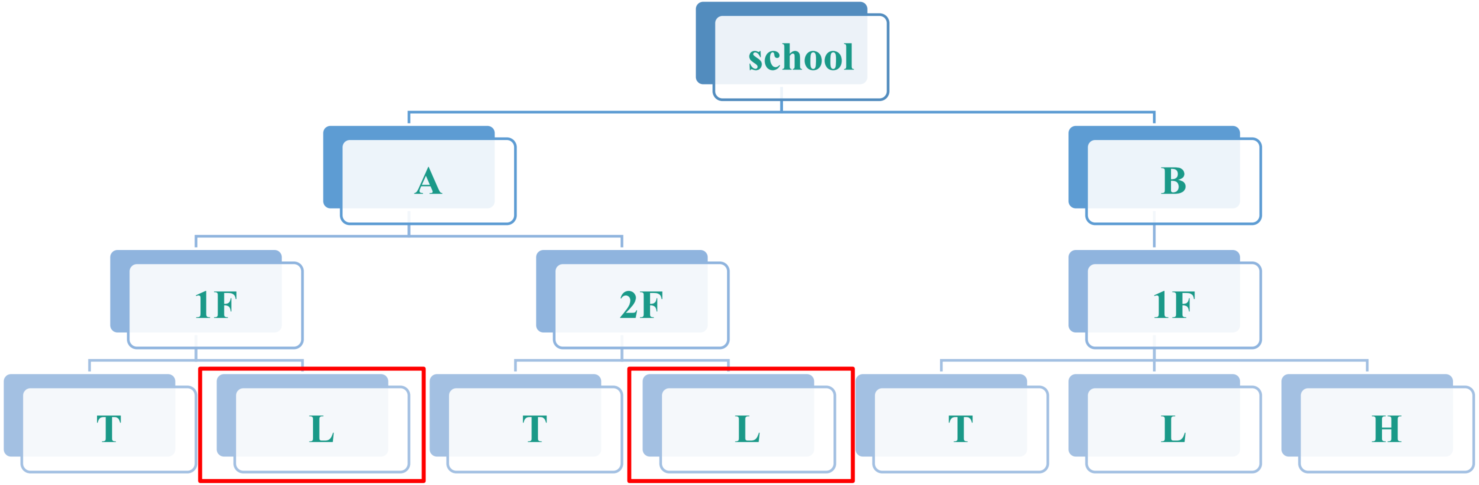
Wildcard in MQTT Topic (#)

```
school/A/2F/#
```



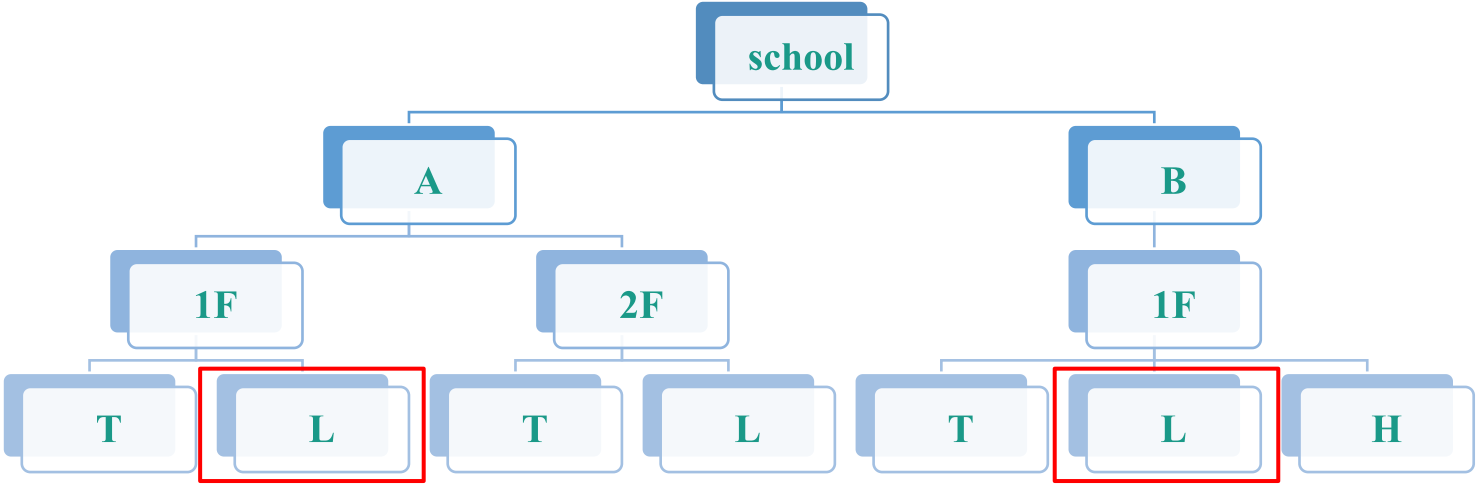
Wildcard in MQTT Topic (+)

```
school/A/+/L
```



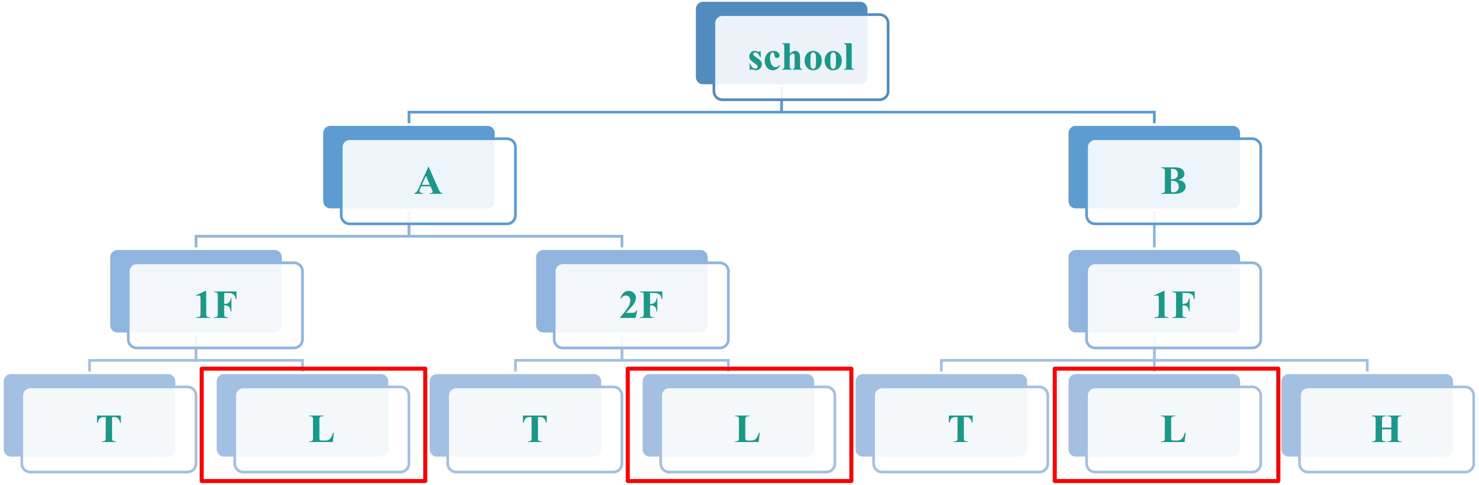
Wildcard in MQTT Topic (+)

```
school/+/1F/L
```



Wildcard in MQTT Topic (+)

```
school/+/+/L
```



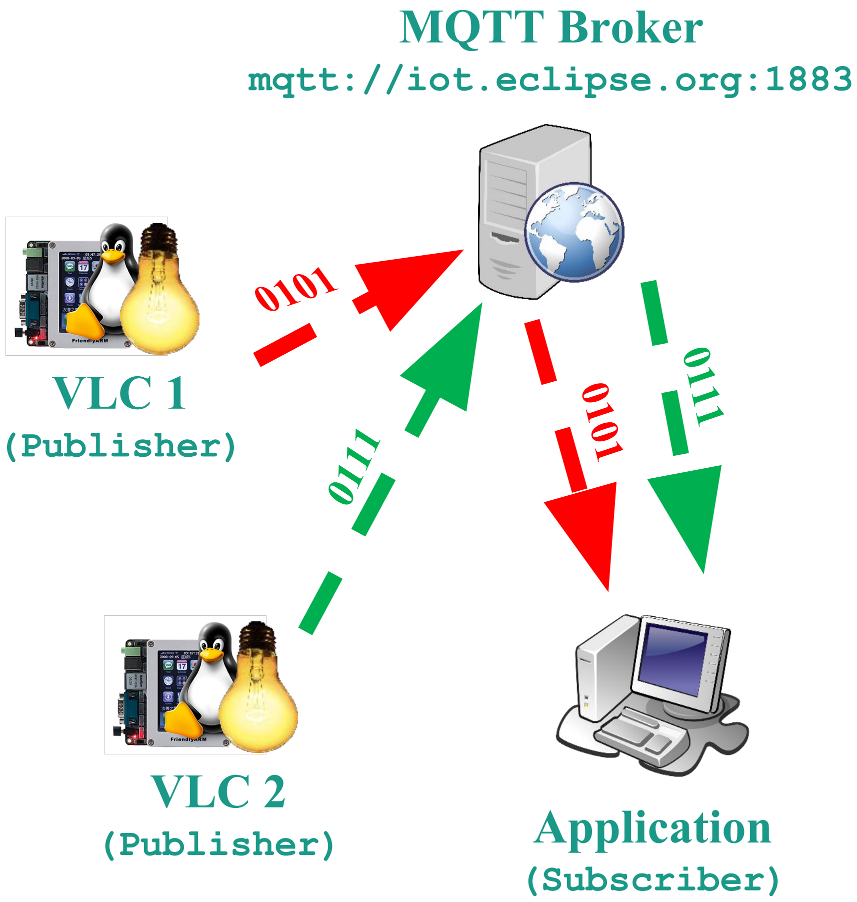
Q & A

Computer History Museum, Mt. View, CA

Experiment 1

Purpose
Data exchange with MQTT

Configuration
All VLC nodes (publisher) send data to application (subscriber).



Experiment 1 (con't)

MQTT Topic

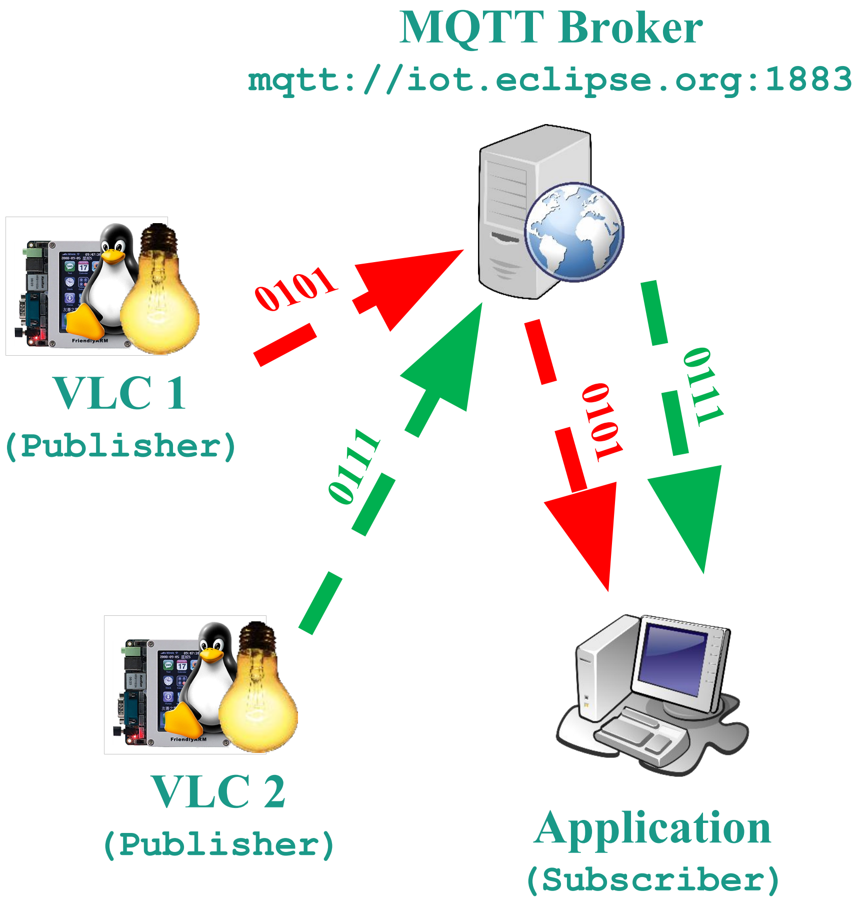
VLCIP/[node id]/data

Publisher

VLCIP/1/data
VLCIP/2/data

Subscriber

VLCIP/1/data
VLCIP/+/data



Eclipse Paho for Python

Development Environment

Install Python (either Anaconda or Python directly)

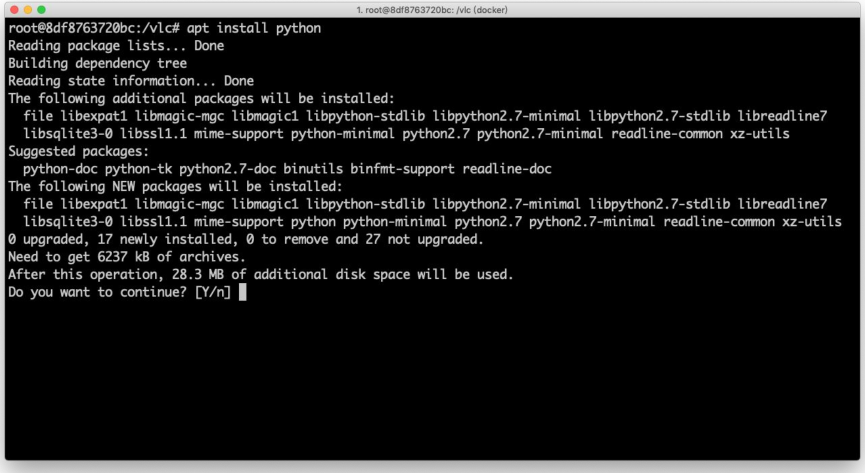
Install Python package manager (`pip`)

Install Python virtual environment (`virtualenv`)

Install Eclipse Paho for Python

Install Python

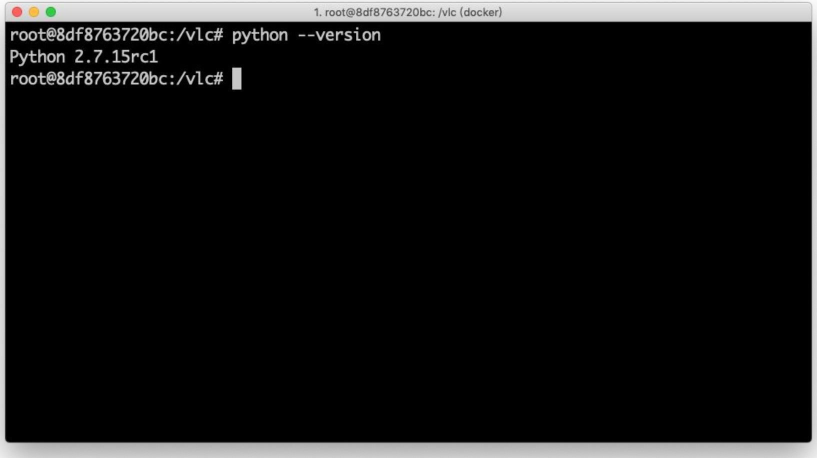
```
$ sudo apt install python
```



```
root@8df8763720bc:/vlc# apt install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  file libexpat1 libmagic-mgc libmagic1 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libreadline7
  libsqlite3-0 libssl1.1 mime-support python-minimal python2.7 python2.7-minimal readline-common xz-utils
Suggested packages:
  python-doc python-tk python2.7-doc binutils binfmt-support readline-doc
The following NEW packages will be installed:
  file libexpat1 libmagic-mgc libmagic1 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libreadline7
  libsqlite3-0 libssl1.1 mime-support python python-minimal python2.7 python2.7-minimal readline-common xz-utils
0 upgraded, 17 newly installed, 0 to remove and 27 not upgraded.
Need to get 6237 kB of archives.
After this operation, 28.3 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Install Python (con't)

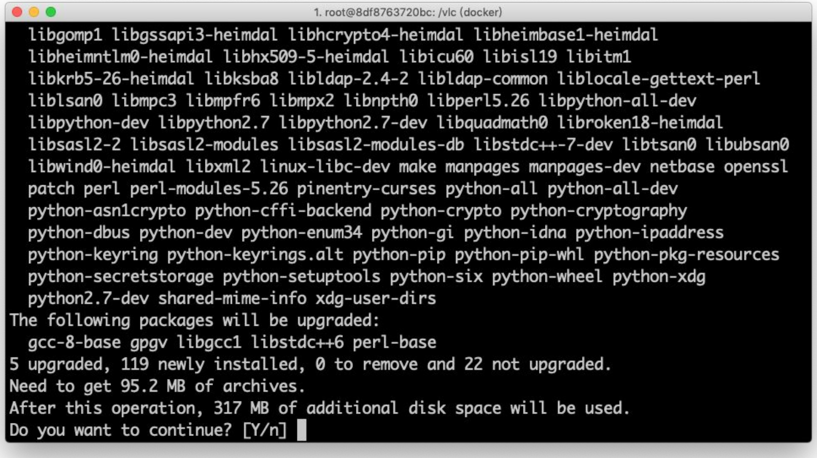
```
$ python --version
```



```
root@8df8763720bc:/vlc# python --version
Python 2.7.15rc1
root@8df8763720bc:/vlc# 
```

Install Python Package Manager

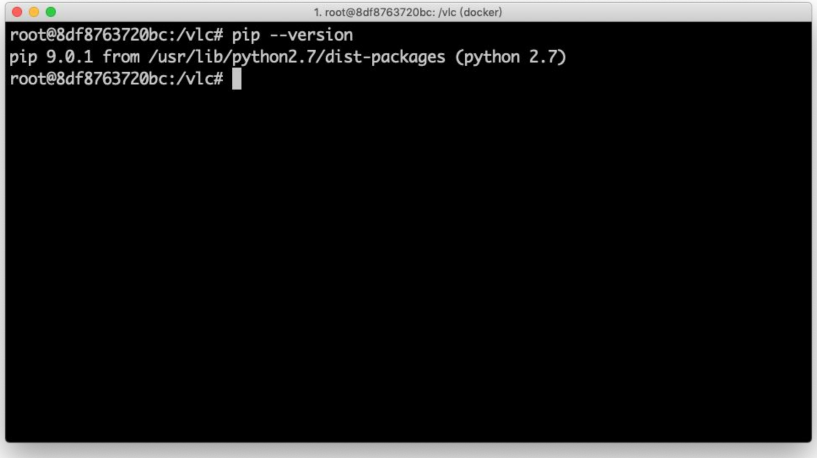
```
$ sudo apt install python-pip
```



```
  libgomp1 libgssapi3-heimdal libhcrypto4-heimdal libheimbase1-heimdal
  libheimntlm0-heimdal libhx509-5-heimdal libicu60 libisl19 libitm1
  libkrb5-26-heimdal libksba8 libldap-2.4-2 libldap-common liblocale-gettext-perl
  liblsan0 libmpc3 libmpfr6 libmpx2 libnpth0 libperl5.26 libpython-all-dev
  libpython-dev libpython2.7 libpython2.7-dev libquadmath0 libroken18-heimdal
  libsasl2-2 libsasl2-modules libsasl2-modules-db libstdc++-7-dev libtsan0 libubsan0
  libwind0-heimdal libxml2 linux-libc-dev make manpages manpages-dev netbase openssl
  patch perl perl-modules-5.26 pinentry-curses python-all python-all-dev
  python-asn1crypto python-cffi-backend python-crypto python-cryptography
  python-dbus python-dev python-enum34 python-gi python-idna python-ipaddress
  python-keyring python-keyrings.alt python-pip python-pip-whl python-pkg-resources
  python-secretstorage python-setuptools python-six python-wheel python-xdg
  python2.7-dev shared-mime-info xdg-user-dirs
The following packages will be upgraded:
  gcc-8-base gpgv libgcc1 libstdc++6 perl-base
5 upgraded, 119 newly installed, 0 to remove and 22 not upgraded.
Need to get 95.2 MB of archives.
After this operation, 317 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Install Python Package Manager (con't)

```
$ pip --version
```



```
root@8df8763720bc:/vlc# pip --version
pip 9.0.1 from /usr/lib/python2.7/dist-packages (python 2.7)
root@8df8763720bc:/vlc#
```

Install Python Virtual Environment

Install python virtual environment

```
$ pip install virtualenv
```

Check if virtualenv is installed

```
$ virtualenv --version
15.2.0
```

Use Python Virtual Environment

Create virtual environment

```
$ virtualenv [name]
```

Enter virtual environment

```
$ cd [name]
$ source bin/activate
```

```
[name]$
```

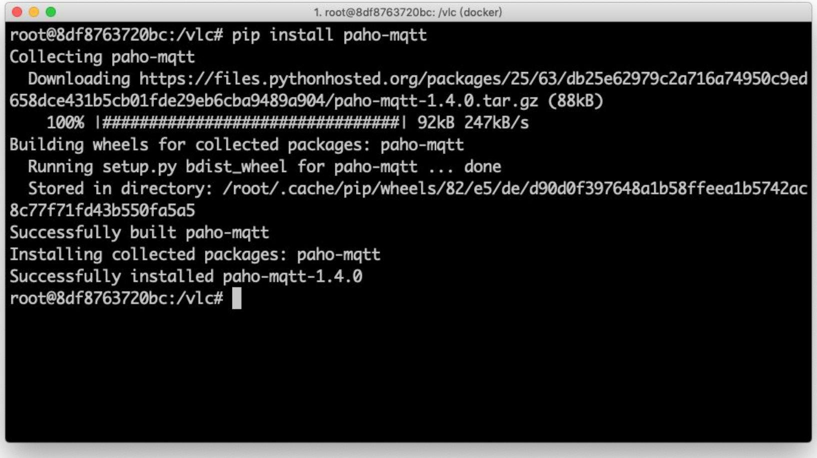
Leave virtual environment

```
[name] $ deactiivate
```

https://pypi.org/project/paho-mqtt/

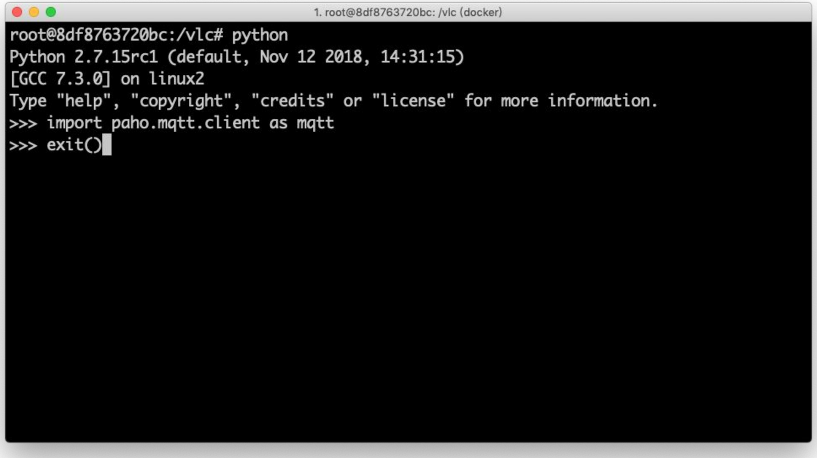
Install Eclipse Paho for Python

```
$ pip install paho-mqtt
```



```
root@8df8763720bc:/vlc# pip install paho-mqtt
Collecting paho-mqtt
  Downloading https://files.pythonhosted.org/packages/25/63/db25e62979c2a716a74950c9ed
658dce431b5cb01fde29eb6cba9489a904/paho-mqtt-1.4.0.tar.gz (88kB)
    100% |################################| 92kB 247kB/s
Building wheels for collected packages: paho-mqtt
  Running setup.py bdist_wheel for paho-mqtt ... done
  Stored in directory: /root/.cache/pip/wheels/82/e5/de/d90d0f397648a1b58ffeea1b5742ac
8c77f71fd43b550fa5a5
Successfully built paho-mqtt
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-1.4.0
root@8df8763720bc:/vlc# 
```

Install Eclipse Paho for Python (con't)



1. root@8df8763720bc: /vlc (docker)

```
root@8df8763720bc:/vlc# python
Python 2.7.15rc1 (default, Nov 12 2018, 14:31:15)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import paho.mqtt.client as mqtt
>>> exit()
```

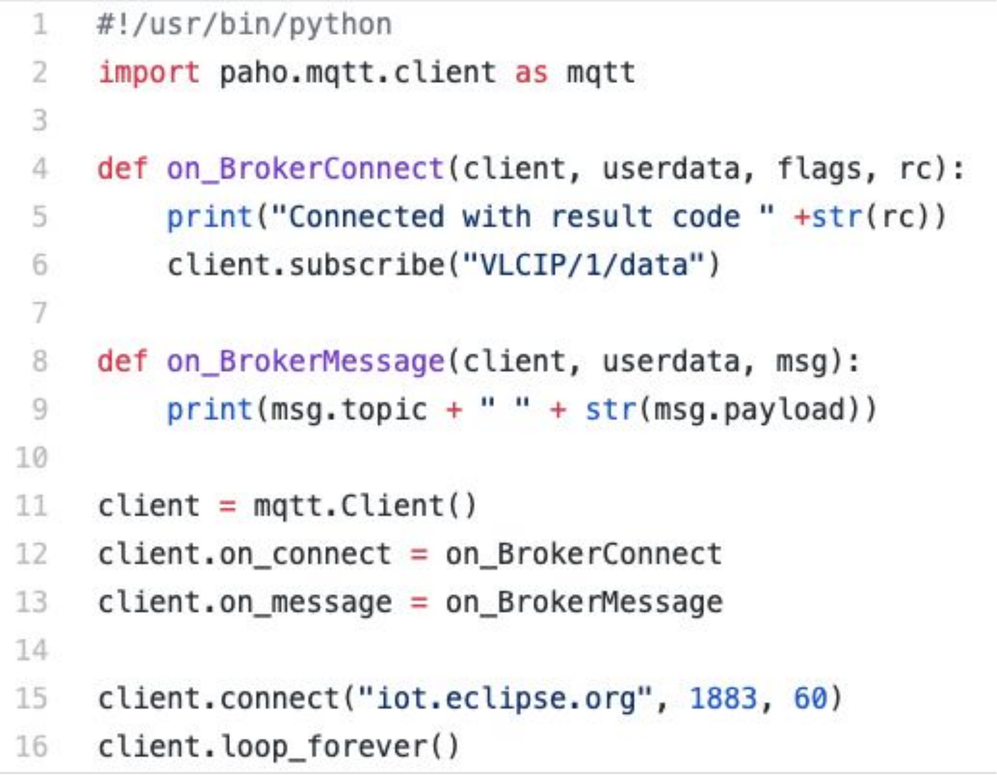
Source Code: https://github.com/ellingtonsu/vlcexp/blob/master/exp1-1/vlcapp.py

Application

```
$ python vlcapp.py
```

Try to subscribe different topics (line 6), such as

```
VLCIP/1/data
VLCIP/2/data
VLCIP/+/data
```



```
1   #!/usr/bin/python
2   import paho.mqtt.client as mqtt
3
4   def on_BrokerConnect(client, userdata, flags, rc):
5       print("Connected with result code " +str(rc))
6       client.subscribe("VLCIP/1/data")
7
8   def on_BrokerMessage(client, userdata, msg):
9       print(msg.topic + " " + str(msg.payload))
10
11  client = mqtt.Client()
12  client.on_connect = on_BrokerConnect
13  client.on_message = on_BrokerMessage
14
15  client.connect("iot.eclipse.org", 1883, 60)
16  client.loop_forever()
```

Source Code: https://github.com/ellingtonsu/vlcexp/blob/master/exp1-1/vlcnode.py

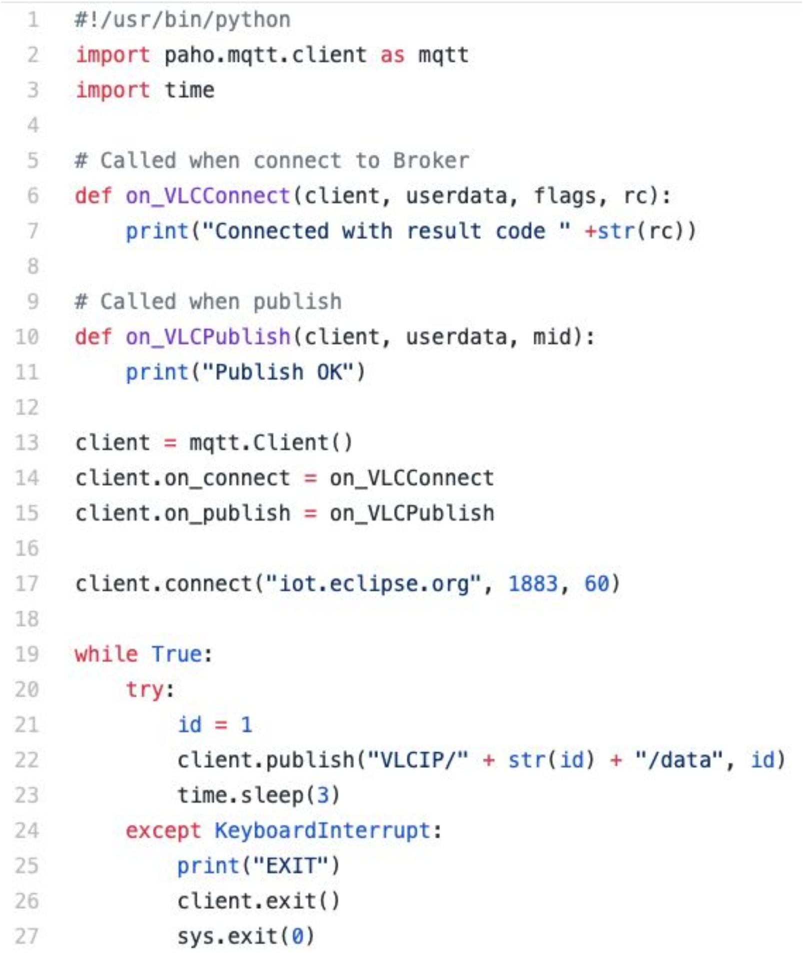
Nodes

Modify node id (line 21)

```
id = n
```

and

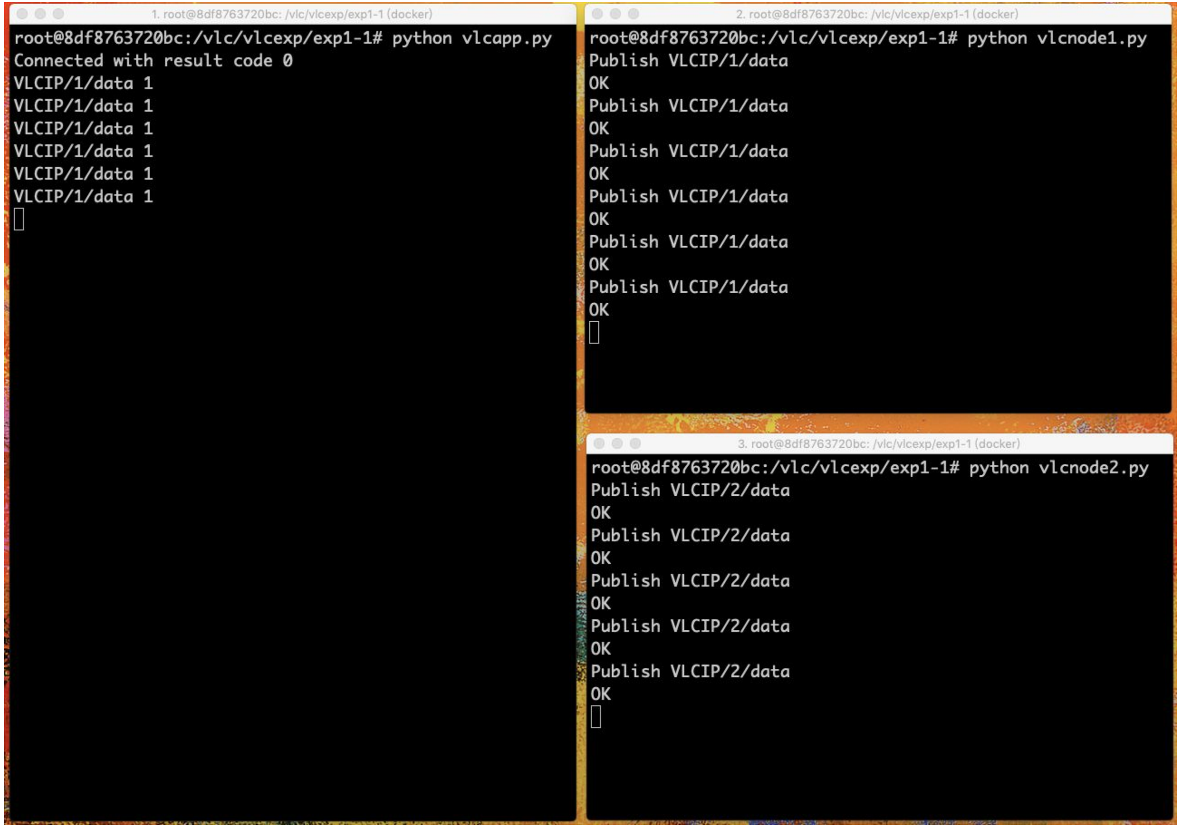
```
$ python vlcnode.py
```



```
 1  #!/usr/bin/python
 2  import paho.mqtt.client as mqtt
 3  import time
 4
 5  # Called when connect to Broker
 6  def on_VLCConnect(client, userdata, flags, rc):
 7      print("Connected with result code " +str(rc))
 8
 9  # Called when publish
10  def on_VLCPublish(client, userdata, mid):
11      print("Publish OK")
12
13  client = mqtt.Client()
14  client.on_connect = on_VLCConnect
15  client.on_publish = on_VLCPublish
16
17  client.connect("iot.eclipse.org", 1883, 60)
18
19  while True:
20      try:
21          id = 1
22          client.publish("VLCIP/" + str(id) + "/data", id)
23          time.sleep(3)
24      except KeyboardInterrupt:
25          print("EXIT")
26          client.exit()
27          sys.exit(0)
```

VLCIP/1/data

Only data from node 1 can be received by application.



1. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)
2. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)
3. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

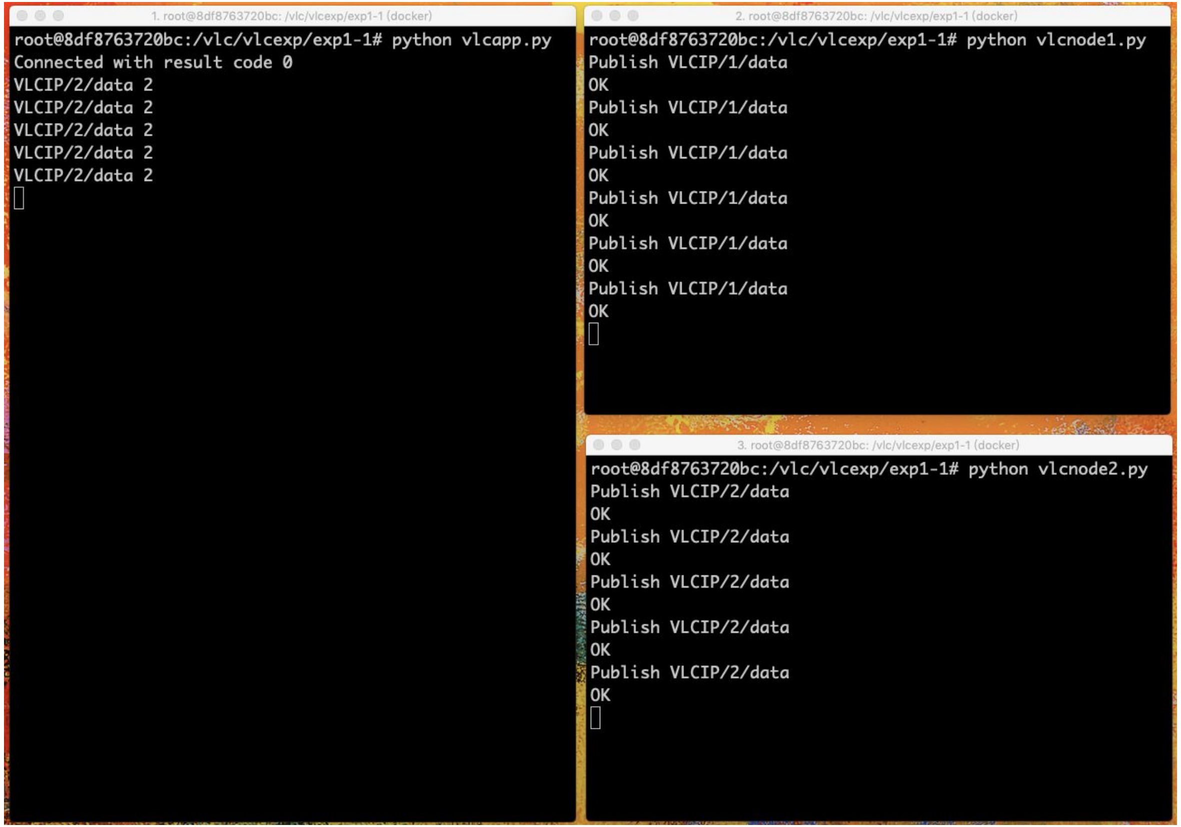
```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcapp.py
Connected with result code 0
VLCIP/1/data 1
VLCIP/1/data 1
VLCIP/1/data 1
VLCIP/1/data 1
VLCIP/1/data 1
VLCIP/1/data 1
```

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode1.py
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
```

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode2.py
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
```

VLCIP/2/data

Only data from node 2 can be received by application.



1. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcapp.py
Connected with result code 0
VLCIP/2/data 2
VLCIP/2/data 2
VLCIP/2/data 2
VLCIP/2/data 2
VLCIP/2/data 2
```

2. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

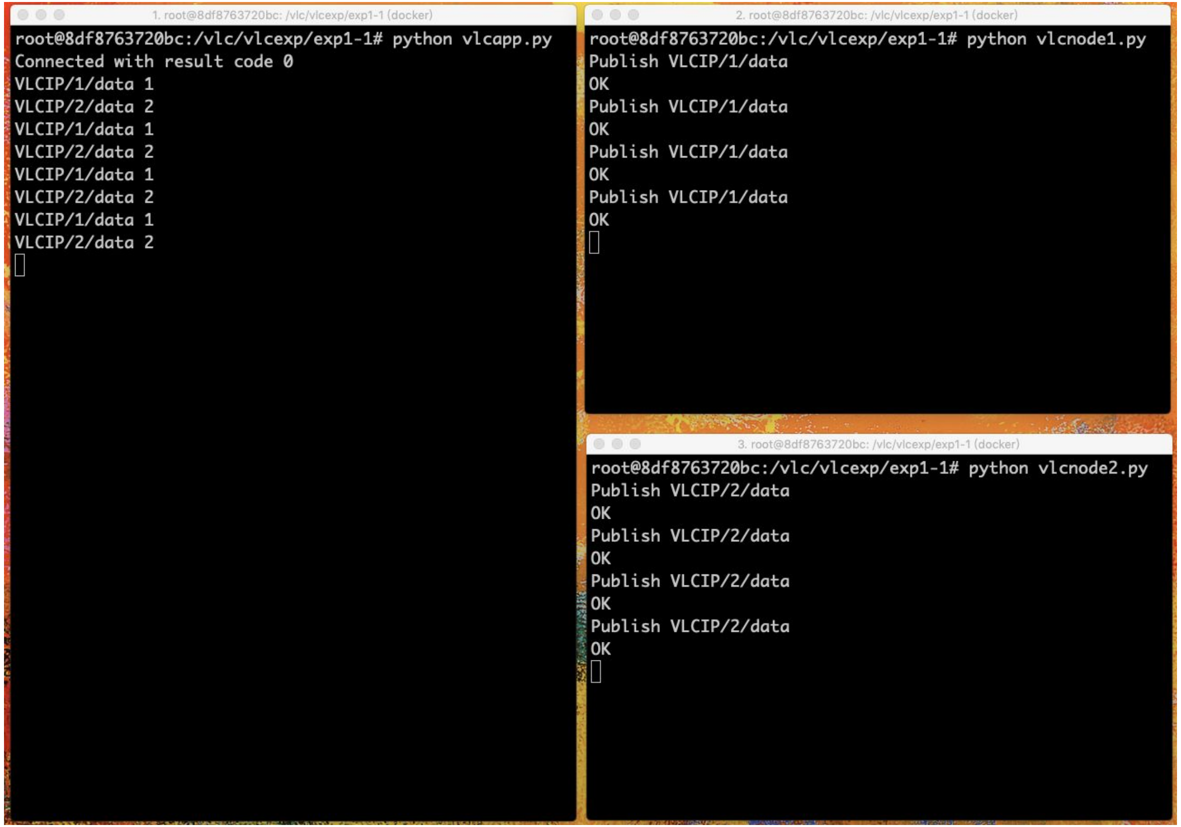
```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode1.py
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
```

3. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode2.py
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
```

VLCIP/+/data

Data from node 1 and node 2 can be both received by application.



1. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcapp.py
Connected with result code 0
VLCIP/1/data 1
VLCIP/2/data 2
VLCIP/1/data 1
VLCIP/2/data 2
VLCIP/1/data 1
VLCIP/2/data 2
VLCIP/1/data 1
VLCIP/2/data 2
```

2. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode1.py
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
Publish VLCIP/1/data
OK
```

3. root@8df8763720bc: /vlc/vlcexp/exp1-1 (docker)

```
root@8df8763720bc:/vlc/vlcexp/exp1-1# python vlcnode2.py
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
Publish VLCIP/2/data
OK
```

MQTT for Node.js

Development Environment

Install Node.js

```
$ sudo apt-get install nodejs
```

Install Node.js package manager (`npm`)

```
$ sudo apt-get install npm
```

Install mqtt

Install mqtt package (https://www.npmjs.com/package/mqtt)

```
    $ npm install mqtt --save
```

Package will be installed in `node_modules` directory

Check version of MQTT for Node.js

```
    $ npm info mqtt version
```

Subscriber

sub.js

```
var mqtt = require('mqtt')
var client = mqtt.connect('mqtt://iot.eclipse.org')

client.on('connect', function() { client.subscribe('VLCIP/1/data')})
client.on('message', function(topic, message) {
     console.log(message.toString())
     client.end()
})
```

```
$ node sub.js
```

Publisher

pub.js

```
var mqtt = require('mqtt')
var client = mqtt.connect('mqtt://iot.eclipse.org')

client.on('connect', function() {
     client.publish('VLCIP/1/data','Hello World')
     client.end()
})
```

```
$ node pub.js
```

Practice

Write programs to simulate school example in this slide with mixed subscribers implemented by Python and Node.js.

Q & A

Computer History Museum, Mt. View, CA

CoAP

Constrained Application Protocol

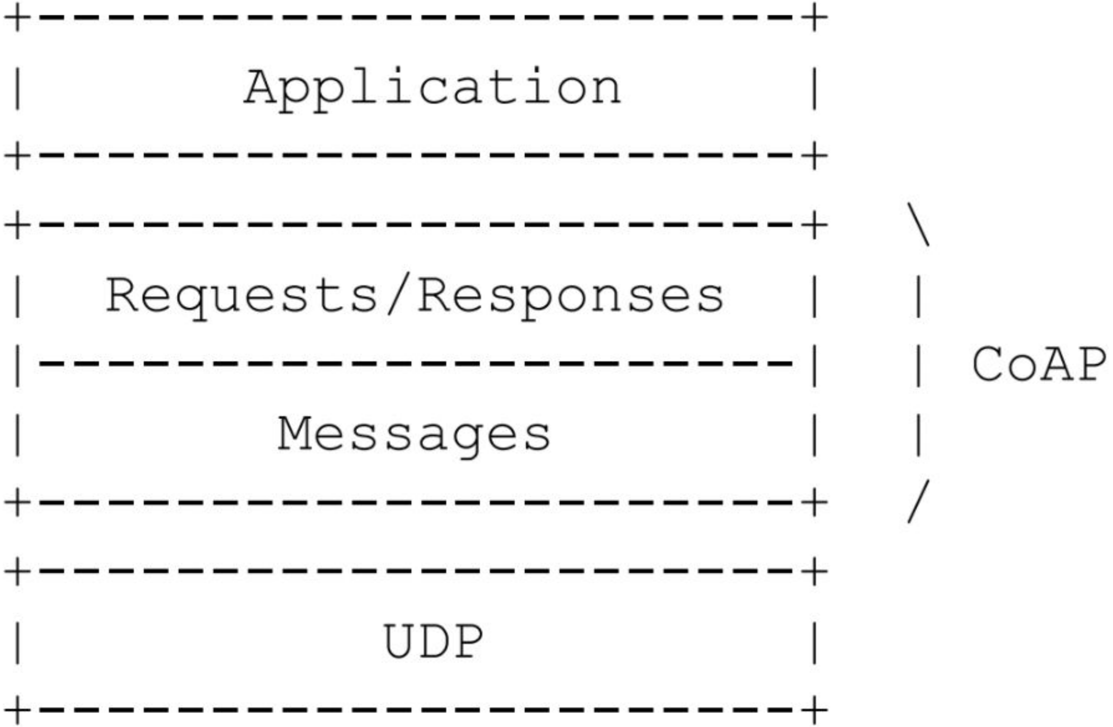
CoAP

CoAP is an IETF protocol standard (RFC 7252).

CoAP is based on client-server model.

Similar to HTTP (GET, PUT, DELETE, POST, ...) but lightweight.

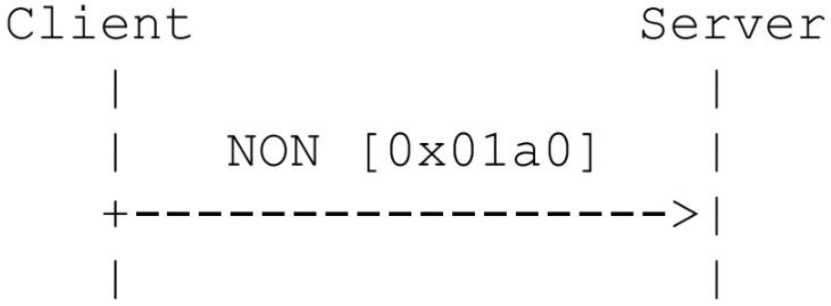
On top of CoAP, Open Mobile Alliance (OMA) has defined Lightweight M2M (LWM2M).



```
+----------------------+
|      Application     |
+----------------------+
+----------------------+  \
|  Requests/Responses  |  |
|----------------------|  | CoAP
|       Messages       |  |
+----------------------+  /
+----------------------+
|          UDP         |
+----------------------+
```

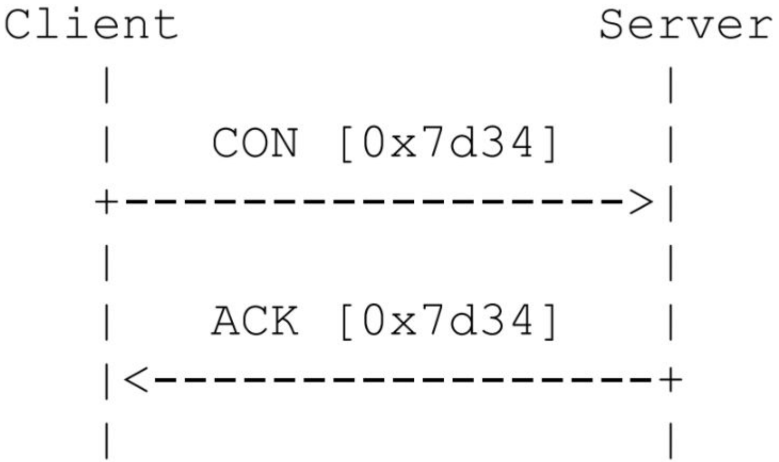
Abstract Layeringof CoAP

Reliable Message Transmission over UDP



```
Client              Server
   |                   |
   |   NON [0x01a0]    |
   +----------------->|
   |                   |
```

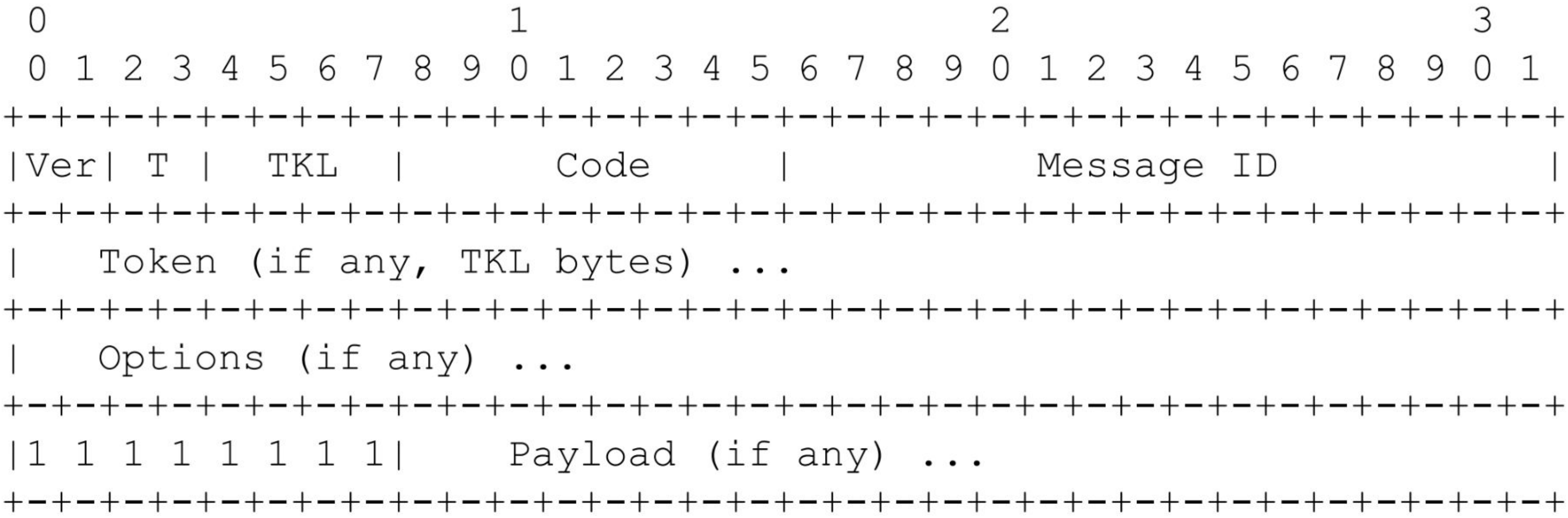
Unreliable Message Transmission



```
Client              Server
   |                   |
   |   CON [0x7d34]    |
   +----------------->|
   |                   |
   |   ACK [0x7d34]    |
   |<-----------------+
   |                   |
```

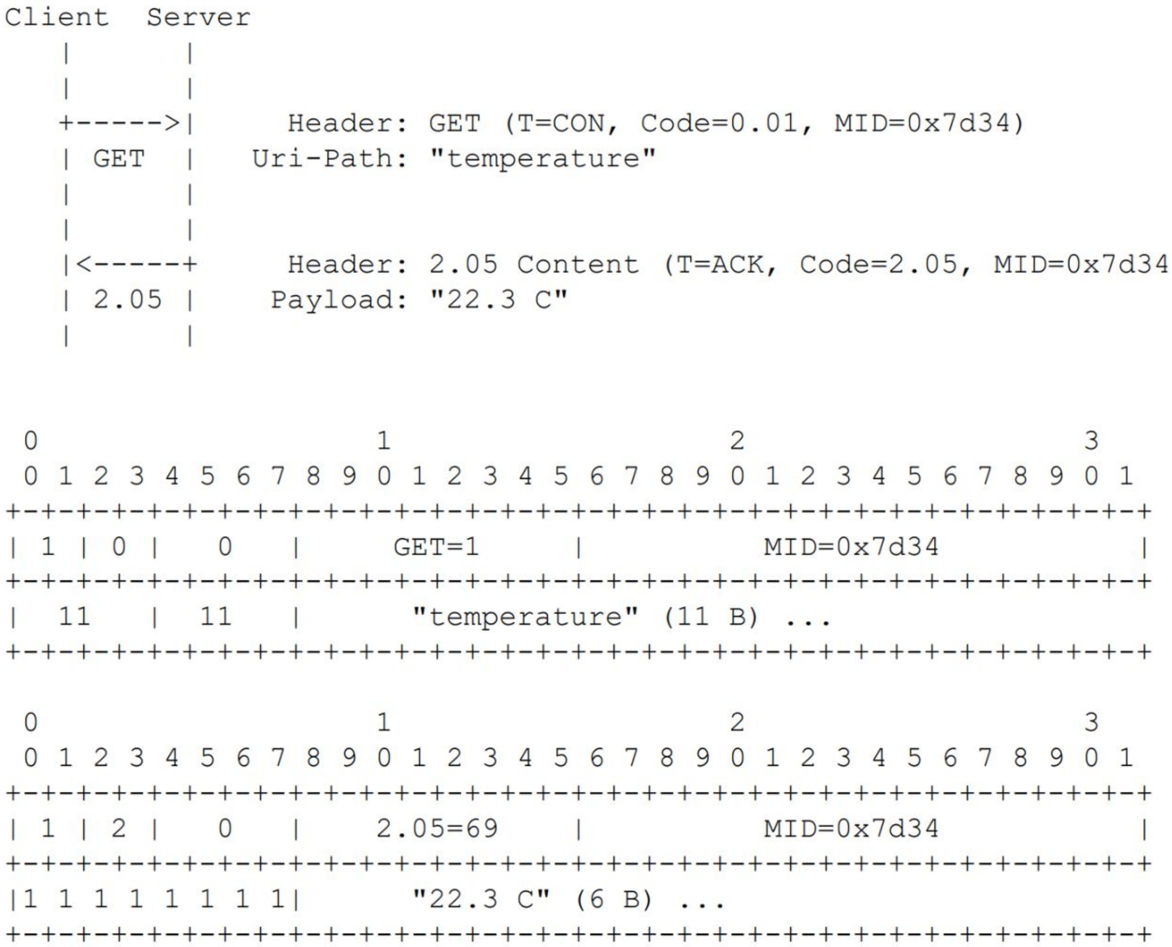
Reliable Message Transmission
(Confirmable)

Message Format



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver| T |  TKL  |      Code     |          Message ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Token (if any, TKL bytes) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|    Payload (if any) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example



```
Client  Server
   |      |
   |      |
   +----->|     Header: GET (T=CON, Code=0.01, MID=0x7d34)
   | GET  |   Uri-Path: "temperature"
   |      |
   |      |
   |<-----+     Header: 2.05 Content (T=ACK, Code=2.05, MID=0x7d34
   | 2.05 |    Payload: "22.3 C"
   |      |


 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |   0   |     GET=1     |          MID=0x7d34           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  11   |  11   |      "temperature" (11 B) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 2 |   0   |    2.05=69    |          MID=0x7d34           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 1 1 1 1 1|      "22.3 C" (6 B) ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Q & A

Computer History Museum, Mt. View, CA