MQTT Protocol Architecture

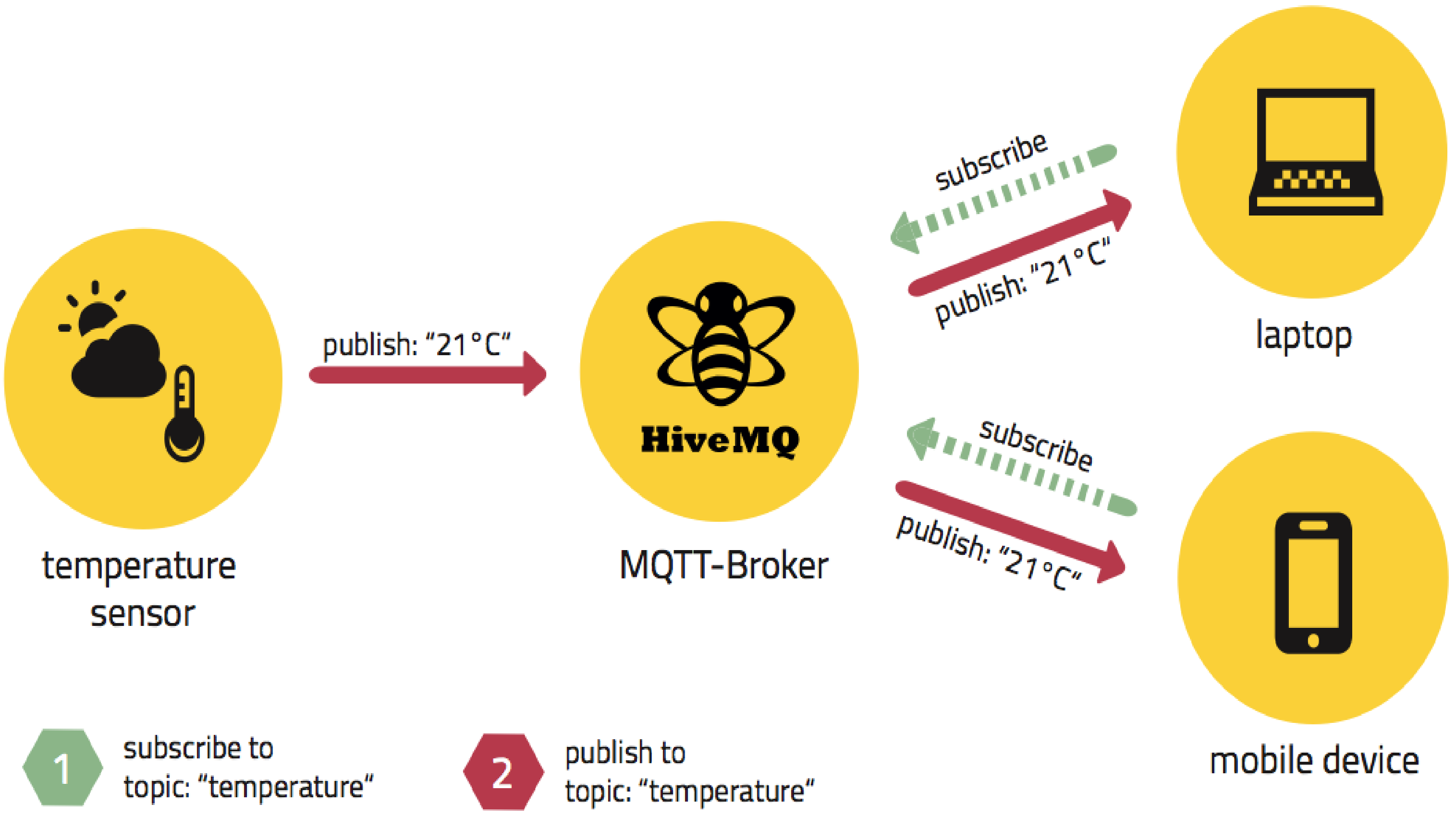
Prof SRN Reddy, IGDTUW

Message Queuing Telemetry Transport (MQTT)

- Light Weight Protocol
- Open Source and Royalty-Free
- Easy to connect with verity of Devices
- Code Optimized: Offers connectivity options optimized for sensors and remote devices.
- Data Delivery:- Delivers relevant data to any application
- Scalable:- Enables massive deployment and management of solutions.
- It is a Publish/Subscribe Messaging Model
- It facilitates one-to-many communications
- Applications: Ideal for constrained networks such as low bandwidth, high latency, data limits etc.

- Flexibility in Services:- messages will be delivered at most once, at least once, or exactly once
- Designed specifically for remote devices with little memory or processing
- Easy to use and simple set of command messages like CONNECT, PUBLISH, SUBSCRIBE, and DISCONNECT.
- Reliable: Built-in support for loss of contact between client and server

MQTT Architecture



Client:

A program or device that uses MQTT

- **Opens the Network Connection to the Server**
- **Publishes Application Messages that other Clients might be interested to use.**
- **Subscribes to request Application Messages that it is interested in receiving.**
- **Closes the Network Connection to the Server.**
- **A client can be a subscriber or Publisher**

Server / Broker

A program or device that acts as an intermediary between Clients which publish Application Messages and Clients which have made Subscriptions.

A Server:

- **Accepts Network Connections from Clients.**
- **Accepts Application Messages published by Clients.**
- **Processes Subscribe and Unsubscribe requests from Clients.**
- **Forwards Application Messages that matches with Client Subscriptions.**
- **closes the Network Connection from the Client.**

Session and Subscription

- A stateful interaction between a Client and a Server is called session.
- A Subscription comprises a Topic Filter and a maximum QoS.
- A Subscription is associated with a single Session.
- A Session can contain more than one Subscription.
- Each Subscription within a Session has a different Topic Filter.
- A **Shared Subscription** comprises a Topic Filter and a maximum QoS. A Shared Subscription can be associated with more than one Session to allow a wider range of message exchange patterns.
- A **Wildcard Subscription** is a Subscription with a Topic Filter containing one or more wildcard characters.

Terms

Topic Name: The label attached to an Application Message which is matched against the Subscriptions known to the Server.

Topic Filter: An expression contained in a Subscription to indicate an interest in one or more topics. A Topic Filter can include wildcard characters.

MQTT Control Packet: A packet of information that is sent across the Network Connection. The MQTT specification defines fifteen different types of MQTT Control Packet, for example the PUBLISH packet is used to convey Application Messages.

Malformed Packet: A control packet that cannot be parsed according to this specification.

Protocol Error: An error that is detected after the packet has been parsed and found to contain data that is not allowed by the protocol or is inconsistent with the state of the Client or Server.

MQTT Topic-based Message Routing

- MQTT protocol uses topic to route message. Topic is a hierarchical structured string, like:

Topic 1: House1/Room1/Temperature

Topic 2: House1/Room1/Humidity

Topic 3: House1/Room2/Temperature

Topic 4: House1/Room2/Humidity

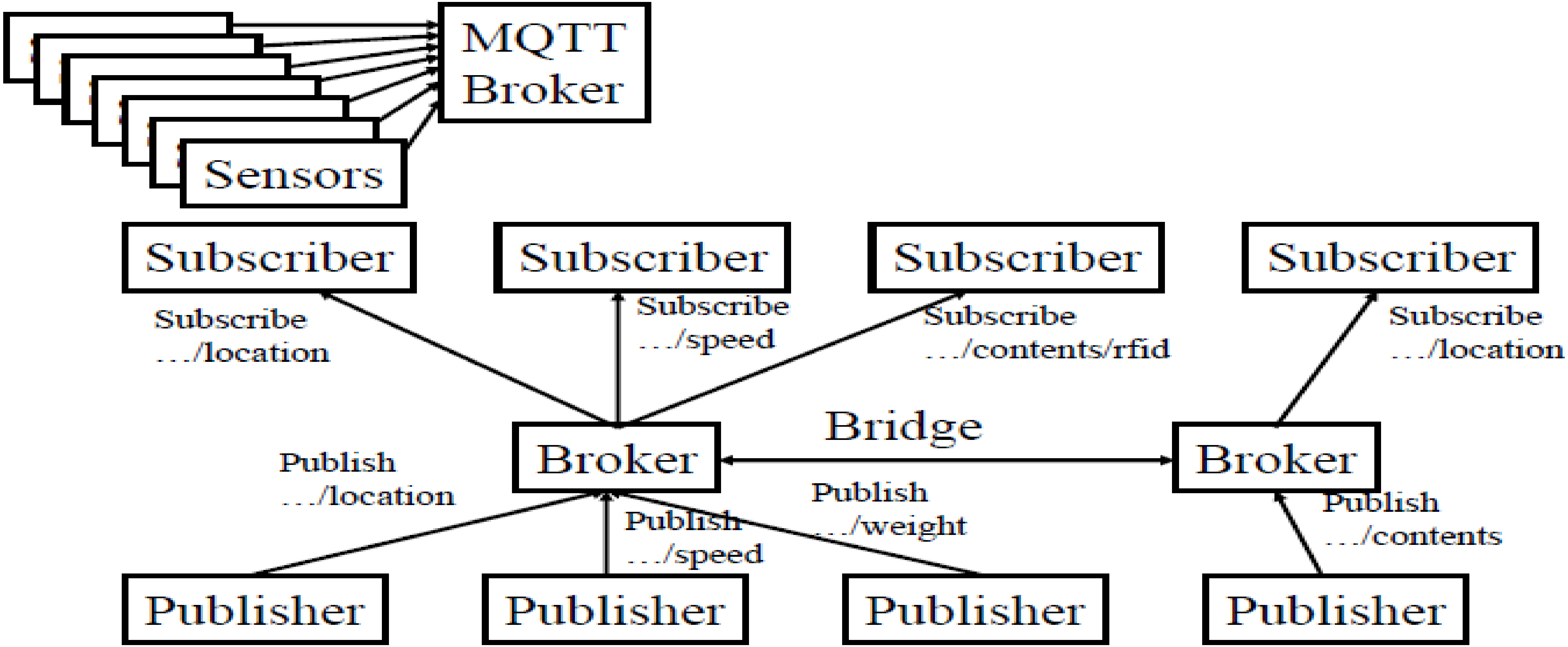
House1/Room1/+

House1/#

A forward slash (/) is used to separate levels within a topic tree and provide a hierarchical structure to the topic space.

The number sigh (#) is a wildcard for multi-level in a topic and the plus sign (+) is a wildcard for single-level.

MQTT Example



Publish and Subscribe Configuration

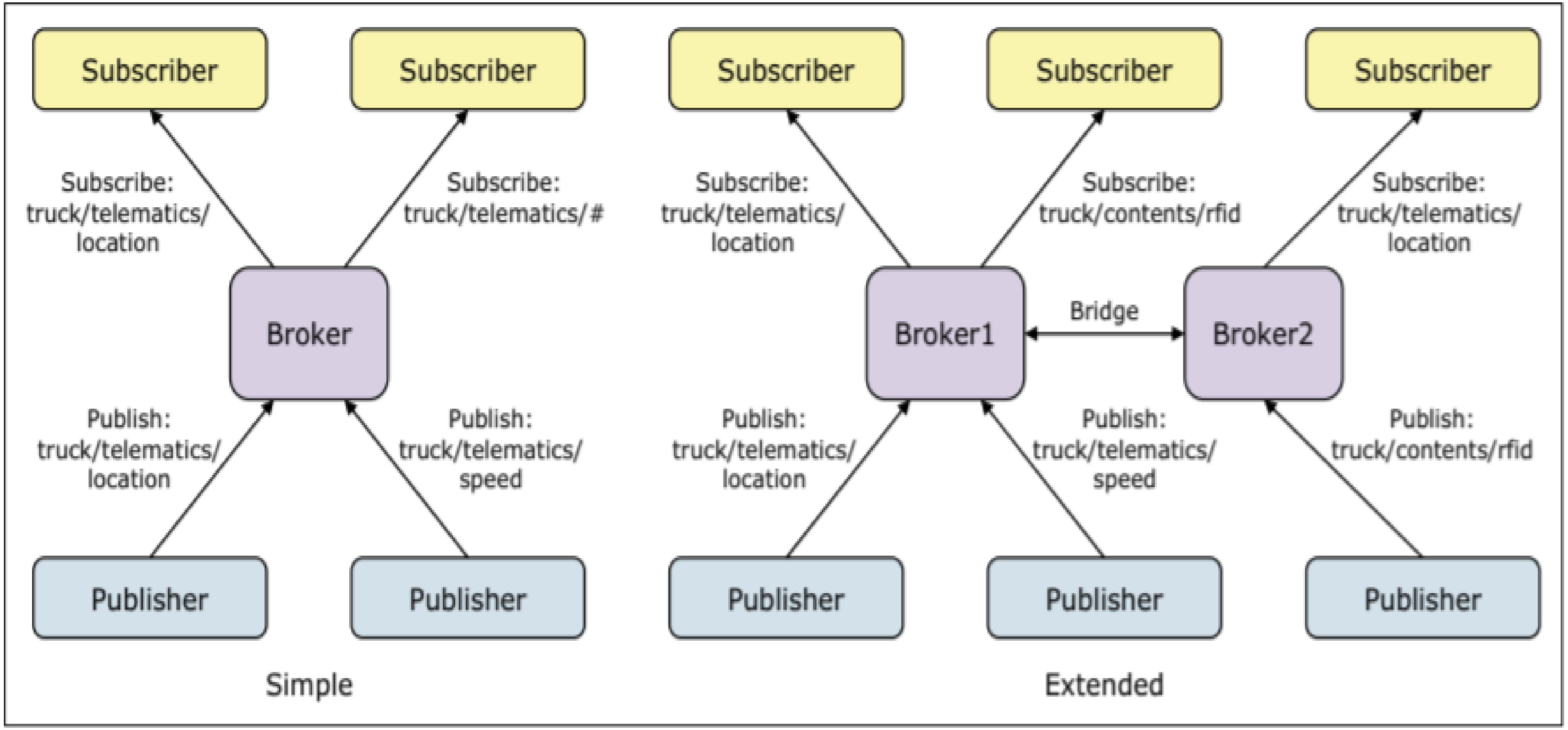
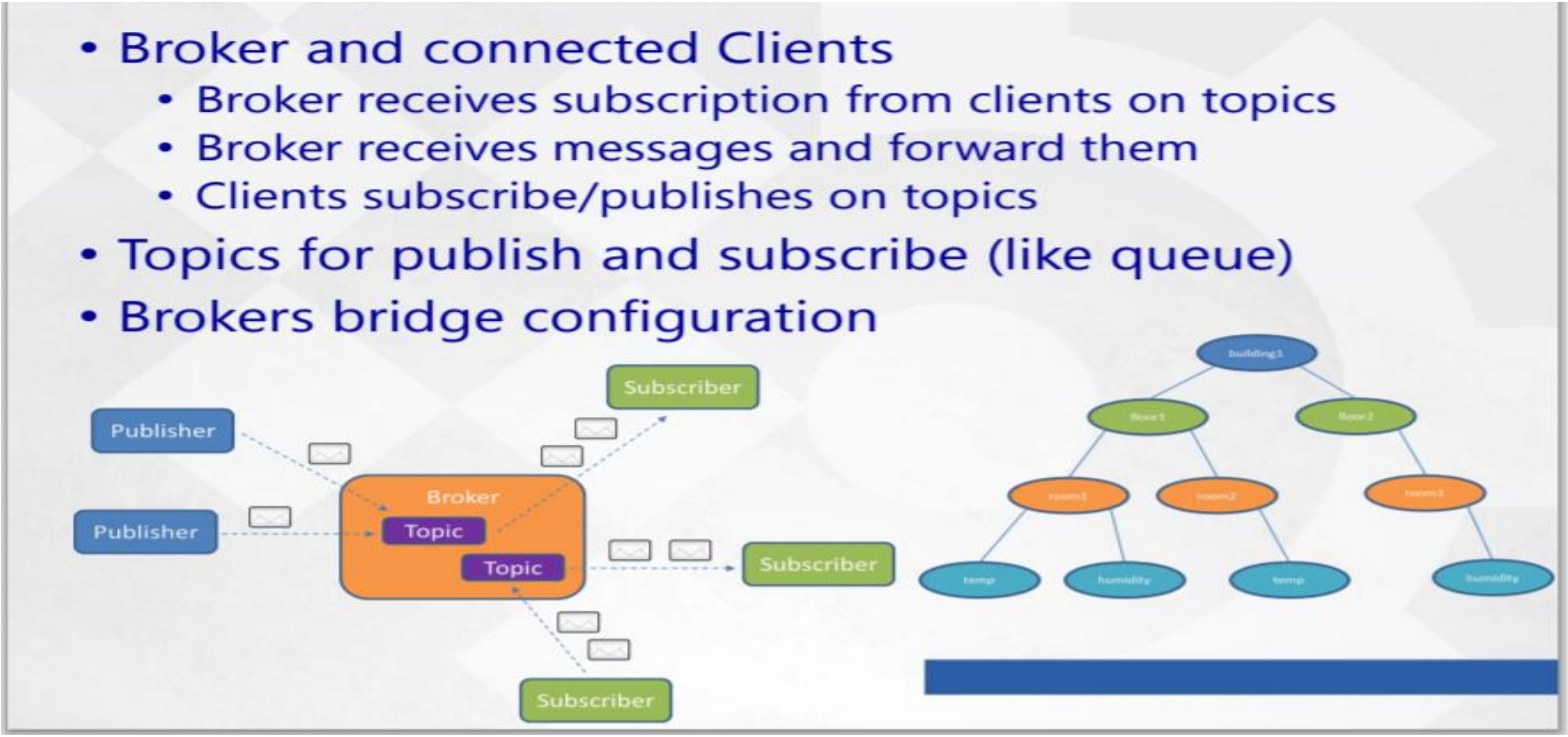

Subscriber
Subscriber
Subscriber
Subscriber
Subscriber
Subscribe: truck/telematics/location
Subscribe: truck/telematics/#
Subscribe: truck/telematics/location
Subscribe: truck/contents/rfid
Subscribe: truck/telematics/location
Broker
Broker1
Bridge
Broker2
Publish: truck/telematics/location
Publish: truck/telematics/speed
Publish: truck/telematics/location
Publish: truck/telematics/speed
Publish: truck/contents/rfid
Publisher
Publisher
Publisher
Publisher
Publisher
Simple
Extended

Figure 2-3 Two examples of publish/subscribe configurations

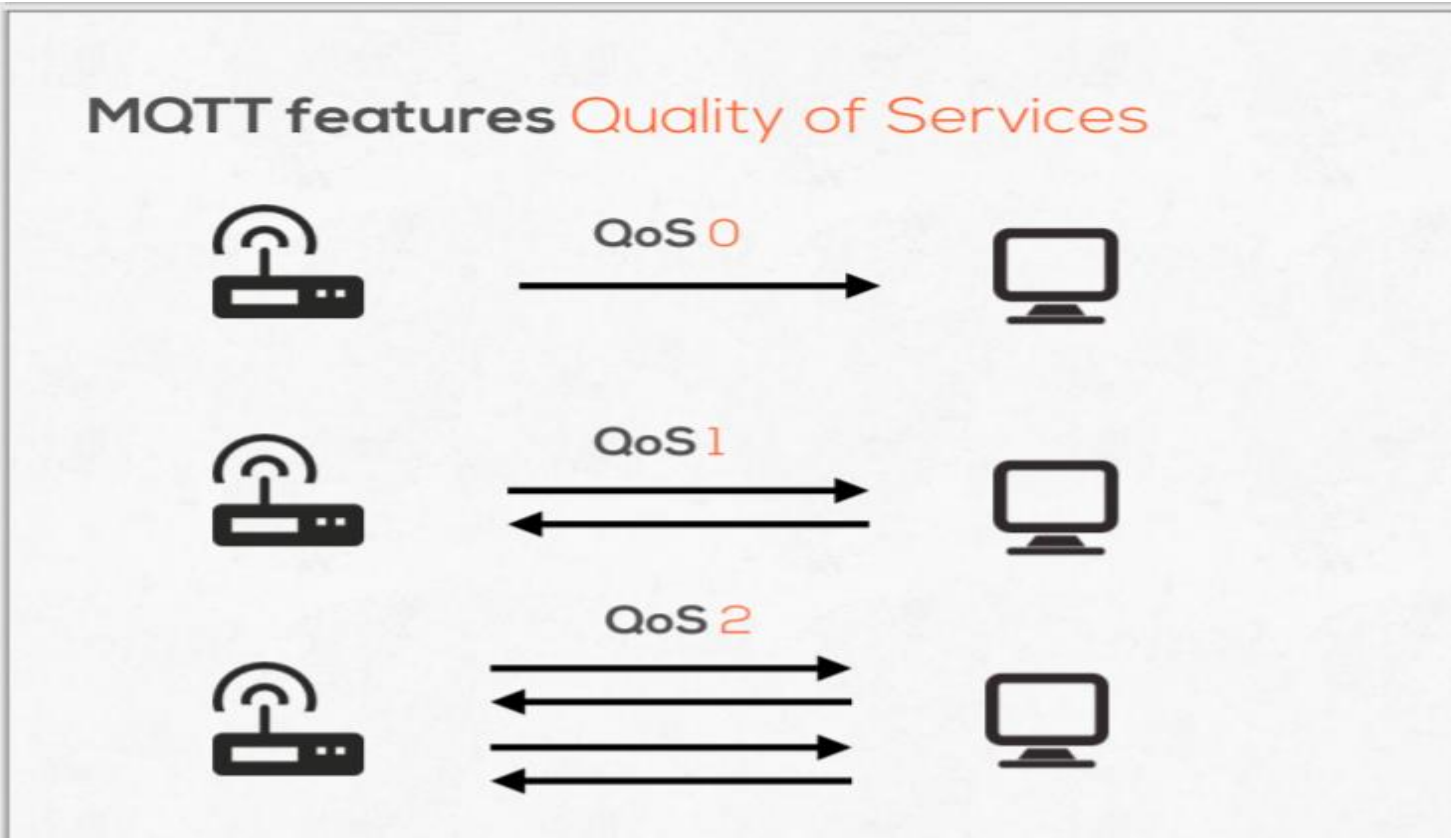
Architecture



- Broker and connected Clients
  - Broker receives subscription from clients on topics
  - Broker receives messages and forward them
  - Clients subscribe/publishes on topics
- Topics for publish and subscribe (like queue)
- Brokers bridge configuration

QoS

QoS value	**Bit 2**	**bit 1**	**Description**
0	0	0	At most once delivery
1	0	1	At least once delivery
2	1	0	Exactly once delivery
-	1	1	Reserved – must not be used



MQTT features Quality of Services

QoS 0

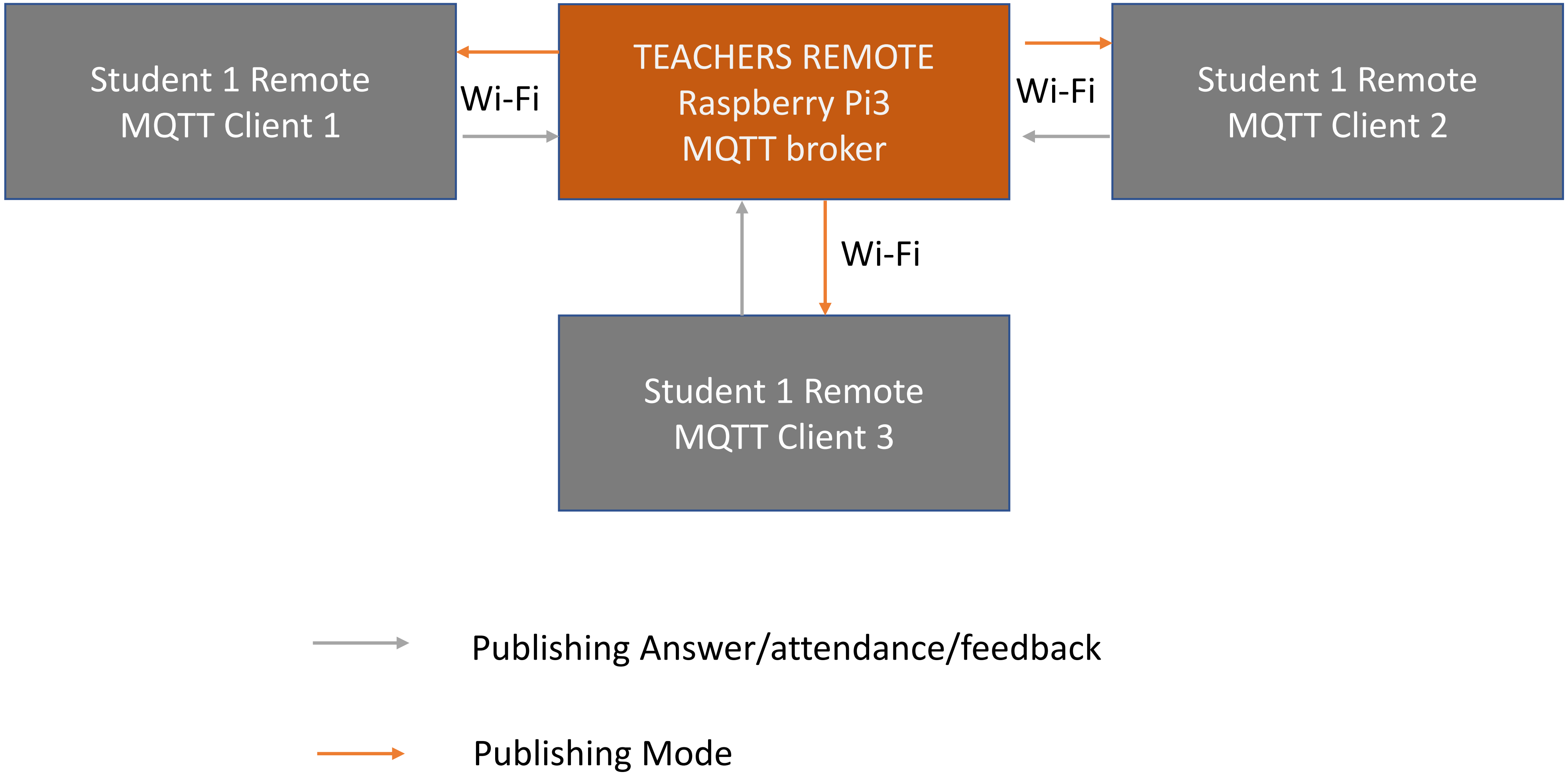
QoS 1

QoS 2

Smart Attendance system
(MQTT based Project)

- 1 remote for teacher for publishing the mode of operation
- Mode
  - Attendence
  - MCQ
  - Feedback
- 1 remote per student for giving attendance, answer and feedback

Architecture



Student 1 Remote
MQTT Client 1

Wi-Fi

TEACHERS REMOTE
Raspberry Pi3
MQTT broker

Wi-Fi

Student 1 Remote
MQTT Client 2

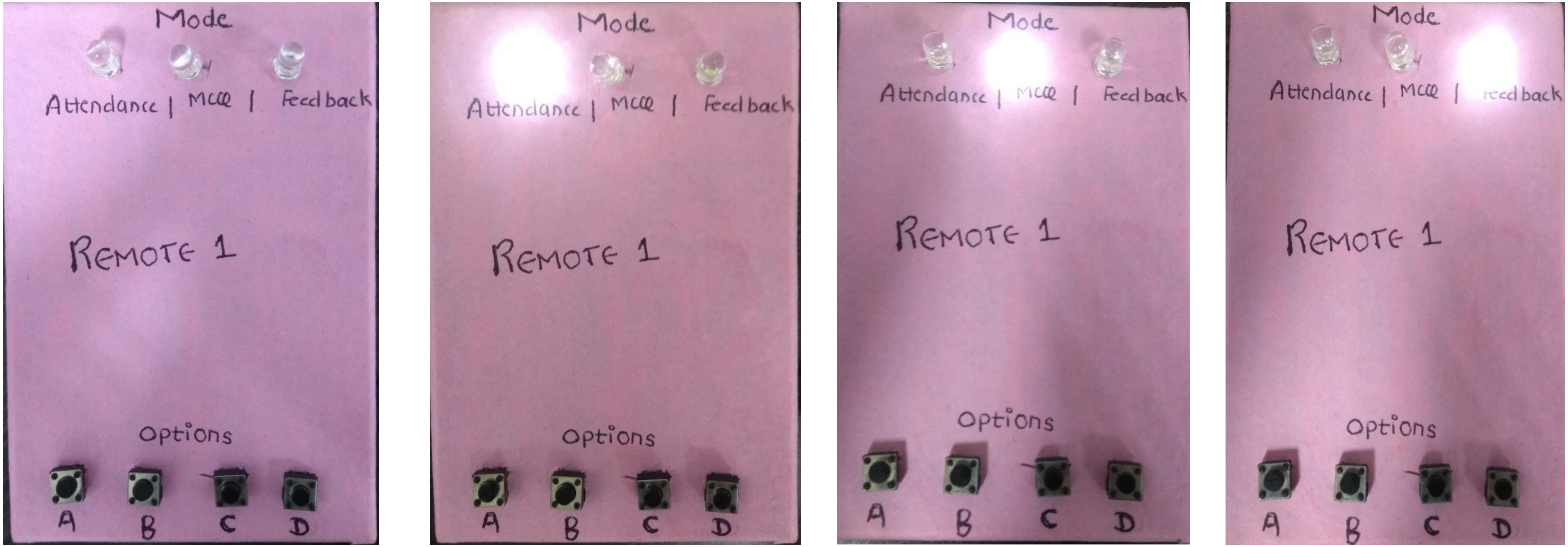
Wi-Fi

Student 1 Remote
MQTT Client 3

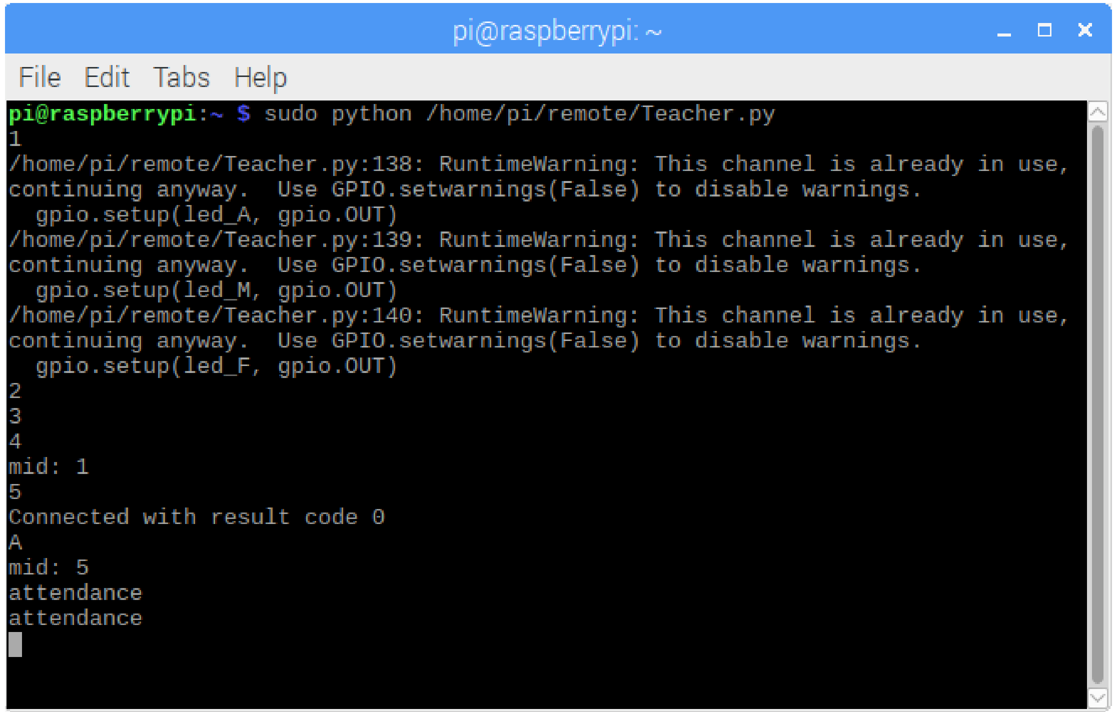
→ Publishing Answer/attendance/feedback

→ Publishing Mode

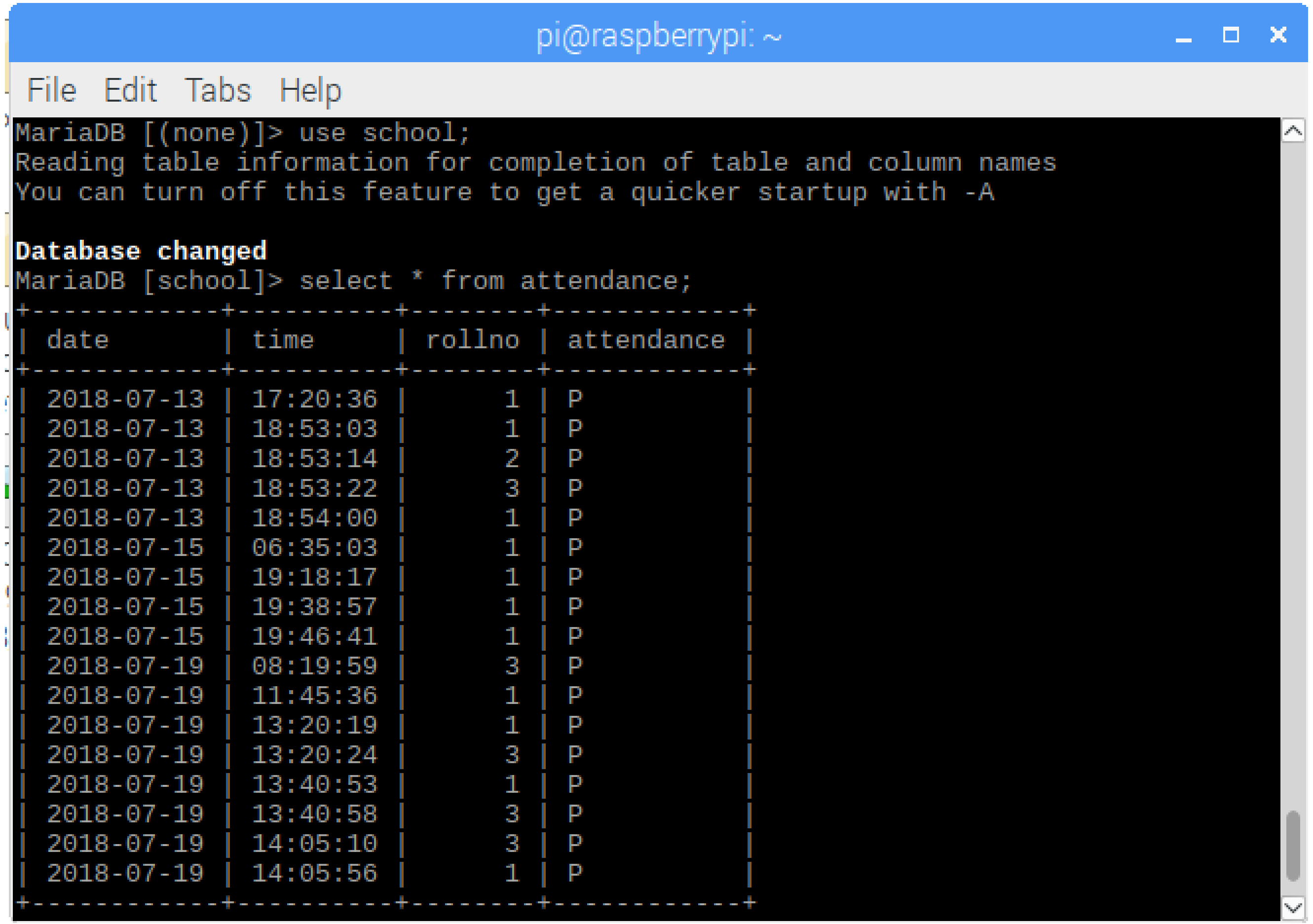
Student Remote in different modes



Attendance saved in database

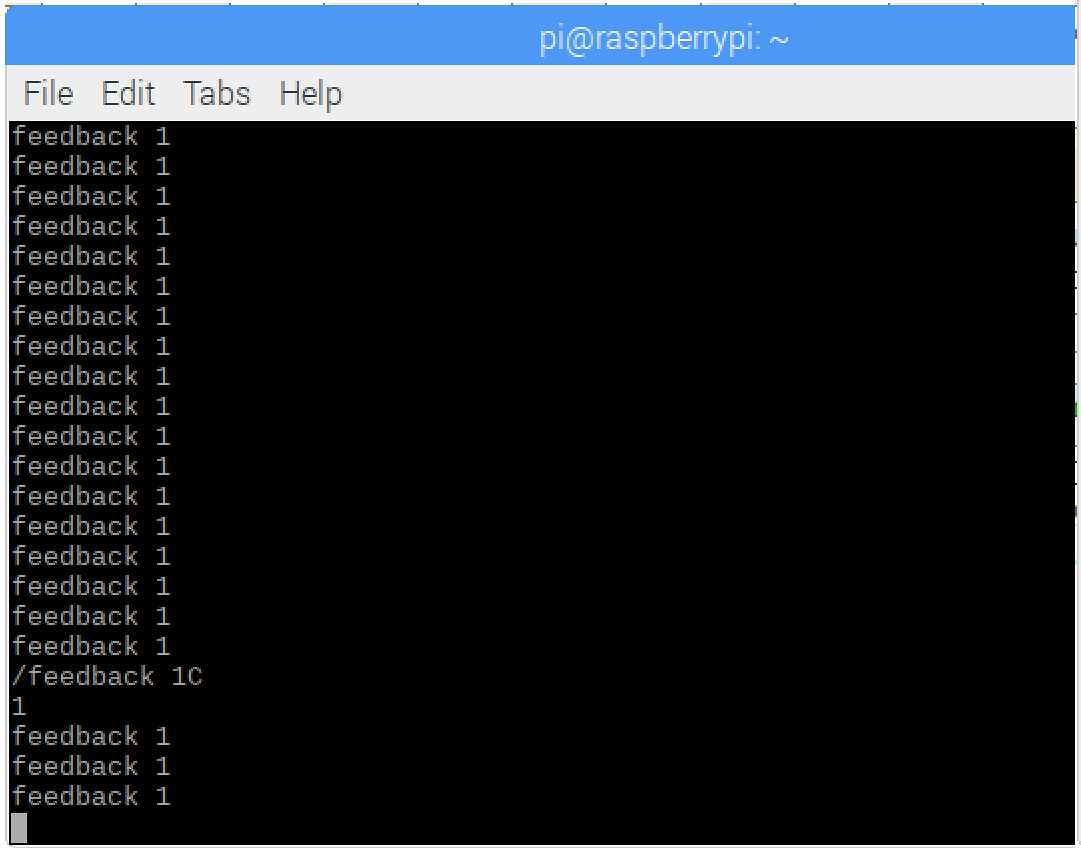


```
pi@raspberrypi:~ $ sudo python /home/pi/remote/Teacher.py
1
/home/pi/remote/Teacher.py:138: RuntimeWarning: This channel is already in use,
continuing anyway.  Use GPIO.setwarnings(False) to disable warnings.
  gpio.setup(led_A, gpio.OUT)
/home/pi/remote/Teacher.py:139: RuntimeWarning: This channel is already in use,
continuing anyway.  Use GPIO.setwarnings(False) to disable warnings.
  gpio.setup(led_M, gpio.OUT)
/home/pi/remote/Teacher.py:140: RuntimeWarning: This channel is already in use,
continuing anyway.  Use GPIO.setwarnings(False) to disable warnings.
  gpio.setup(led_F, gpio.OUT)
2
3
4
mid: 1
5
Connected with result code 0
A
mid: 5
attendance
attendance
```

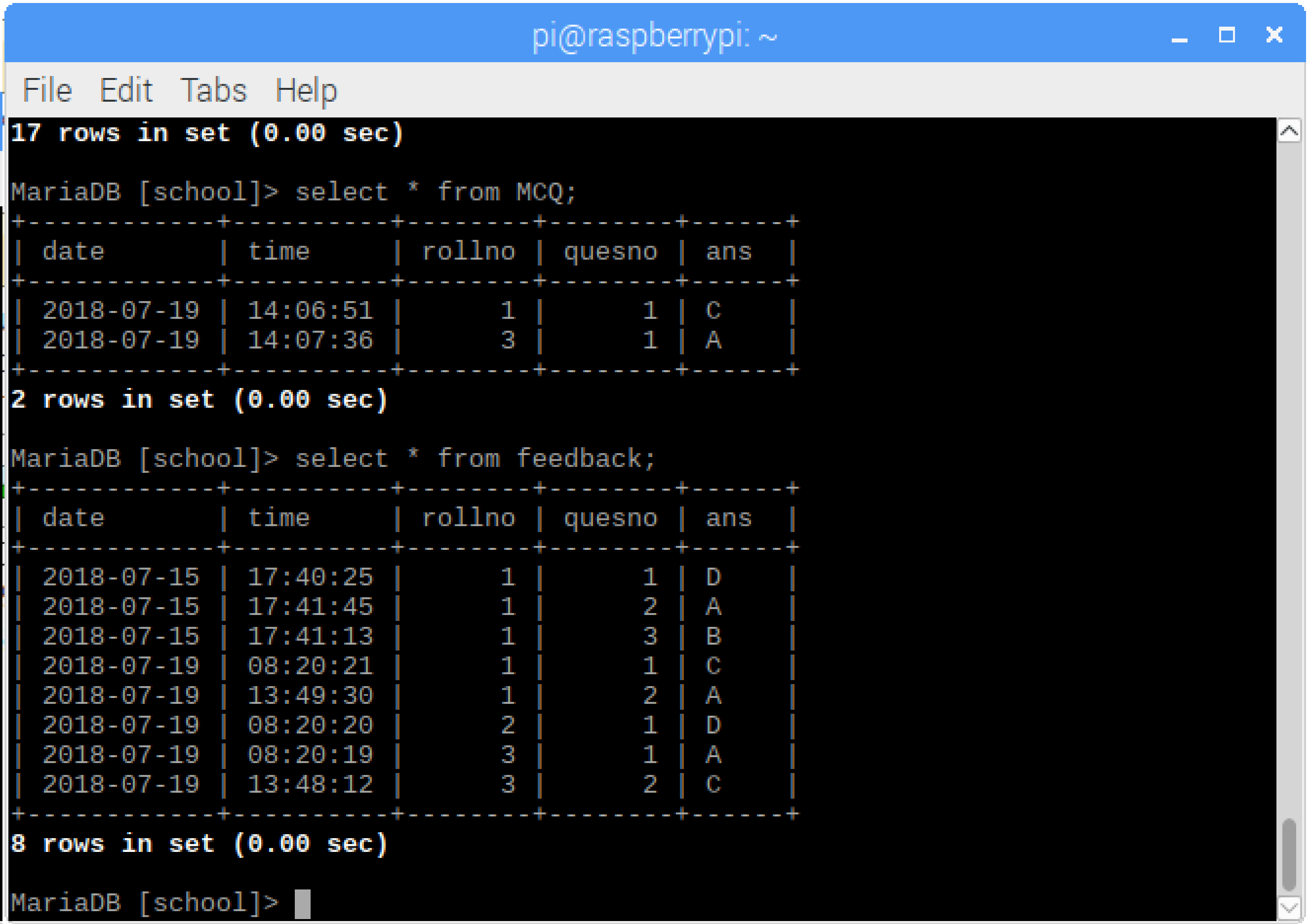


```
MariaDB [(none)]> use school;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [school]> select * from attendance;
+------------+----------+--------+------------+
| date       | time     | rollno | attendance |
+------------+----------+--------+------------+
| 2018-07-13 | 17:20:36 |      1 | P          |
| 2018-07-13 | 18:53:03 |      1 | P          |
| 2018-07-13 | 18:53:14 |      2 | P          |
| 2018-07-13 | 18:53:22 |      3 | P          |
| 2018-07-13 | 18:54:00 |      1 | P          |
| 2018-07-15 | 06:35:03 |      1 | P          |
| 2018-07-15 | 19:18:17 |      1 | P          |
| 2018-07-15 | 19:38:57 |      1 | P          |
| 2018-07-15 | 19:46:41 |      1 | P          |
| 2018-07-19 | 08:19:59 |      3 | P          |
| 2018-07-19 | 11:45:36 |      1 | P          |
| 2018-07-19 | 13:20:19 |      1 | P          |
| 2018-07-19 | 13:20:24 |      3 | P          |
| 2018-07-19 | 13:40:53 |      1 | P          |
| 2018-07-19 | 13:40:58 |      3 | P          |
| 2018-07-19 | 14:05:10 |      3 | P          |
| 2018-07-19 | 14:05:56 |      1 | P          |
+------------+----------+--------+------------+
```

Feedbacks saved in database

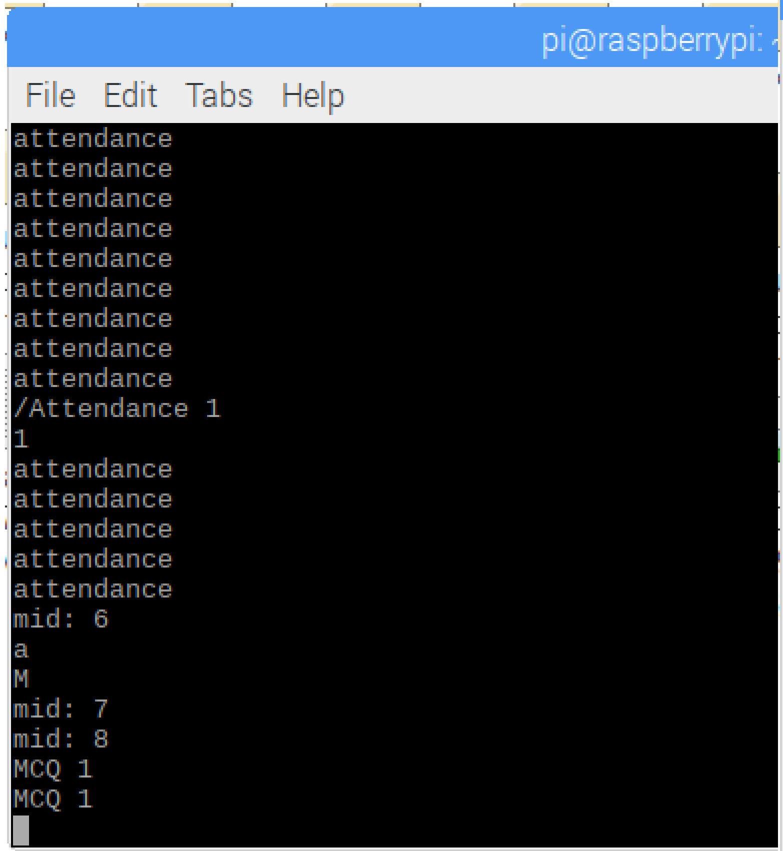


```
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
feedback 1
/feedback 1C
1
feedback 1
feedback 1
feedback 1
```

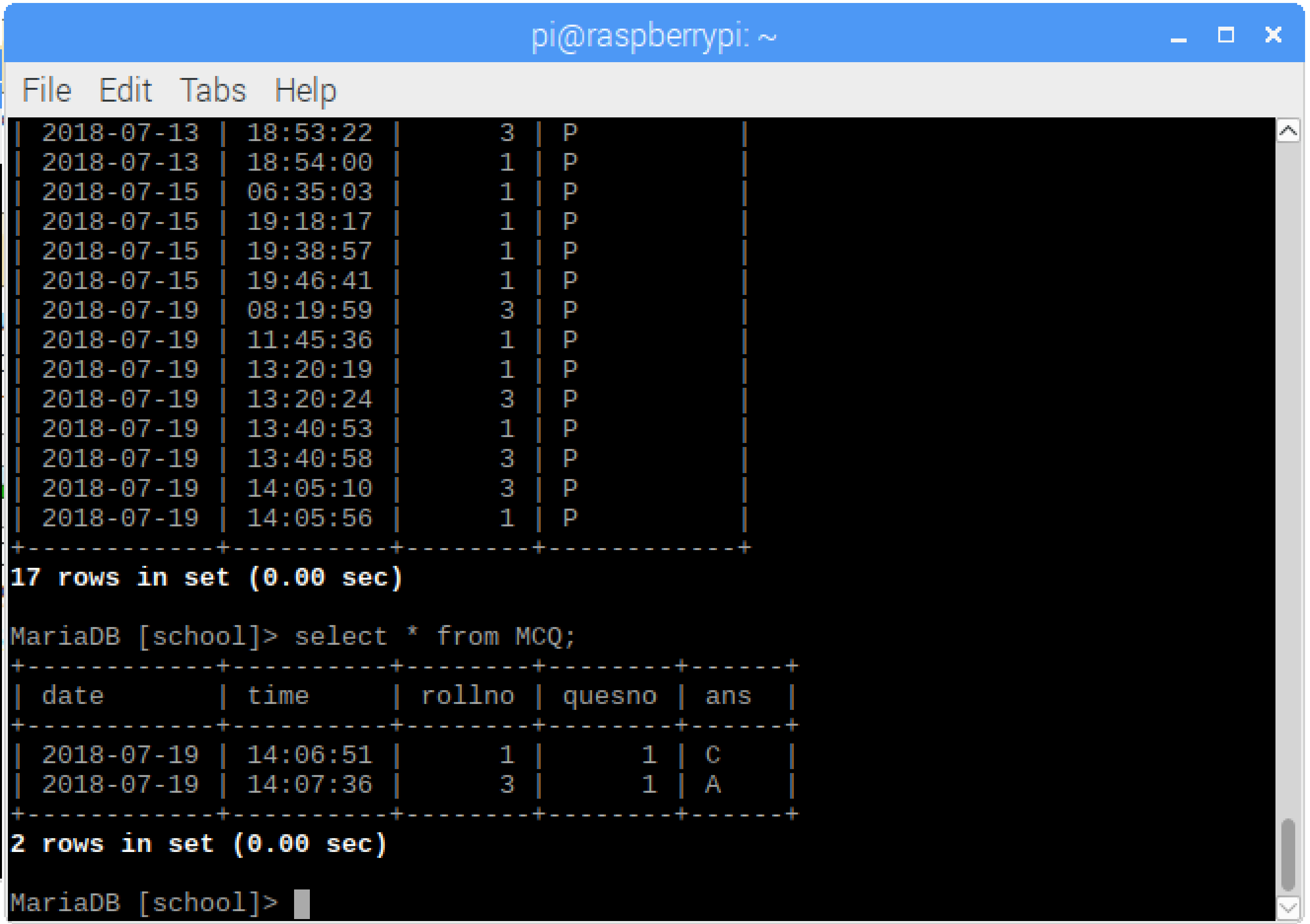


```
17 rows in set (0.00 sec)

MariaDB [school]> select * from MCQ;
+------------+----------+--------+--------+------+
| date       | time     | rollno | quesno | ans  |
+------------+----------+--------+--------+------+
| 2018-07-19 | 14:06:51 |      1 |      1 | C    |
| 2018-07-19 | 14:07:36 |      3 |      1 | A    |
+------------+----------+--------+--------+------+
2 rows in set (0.00 sec)

MariaDB [school]> select * from feedback;
+------------+----------+--------+--------+------+
| date       | time     | rollno | quesno | ans  |
+------------+----------+--------+--------+------+
| 2018-07-15 | 17:40:25 |      1 |      1 | D    |
| 2018-07-15 | 17:41:45 |      1 |      2 | A    |
| 2018-07-15 | 17:41:13 |      1 |      3 | B    |
| 2018-07-19 | 08:20:21 |      1 |      1 | C    |
| 2018-07-19 | 13:49:30 |      1 |      2 | A    |
| 2018-07-19 | 08:20:20 |      2 |      1 | D    |
| 2018-07-19 | 08:20:19 |      3 |      1 | A    |
| 2018-07-19 | 13:48:12 |      3 |      2 | C    |
+------------+----------+--------+--------+------+
8 rows in set (0.00 sec)

MariaDB [school]>
```

MCQ answer saved on database



```
attendance
attendance
attendance
attendance
attendance
attendance
attendance
attendance
attendance
/Attendance 1
1
attendance
attendance
attendance
attendance
attendance
mid: 6
a
M
mid: 7
mid: 8
MCQ 1
MCQ 1
```

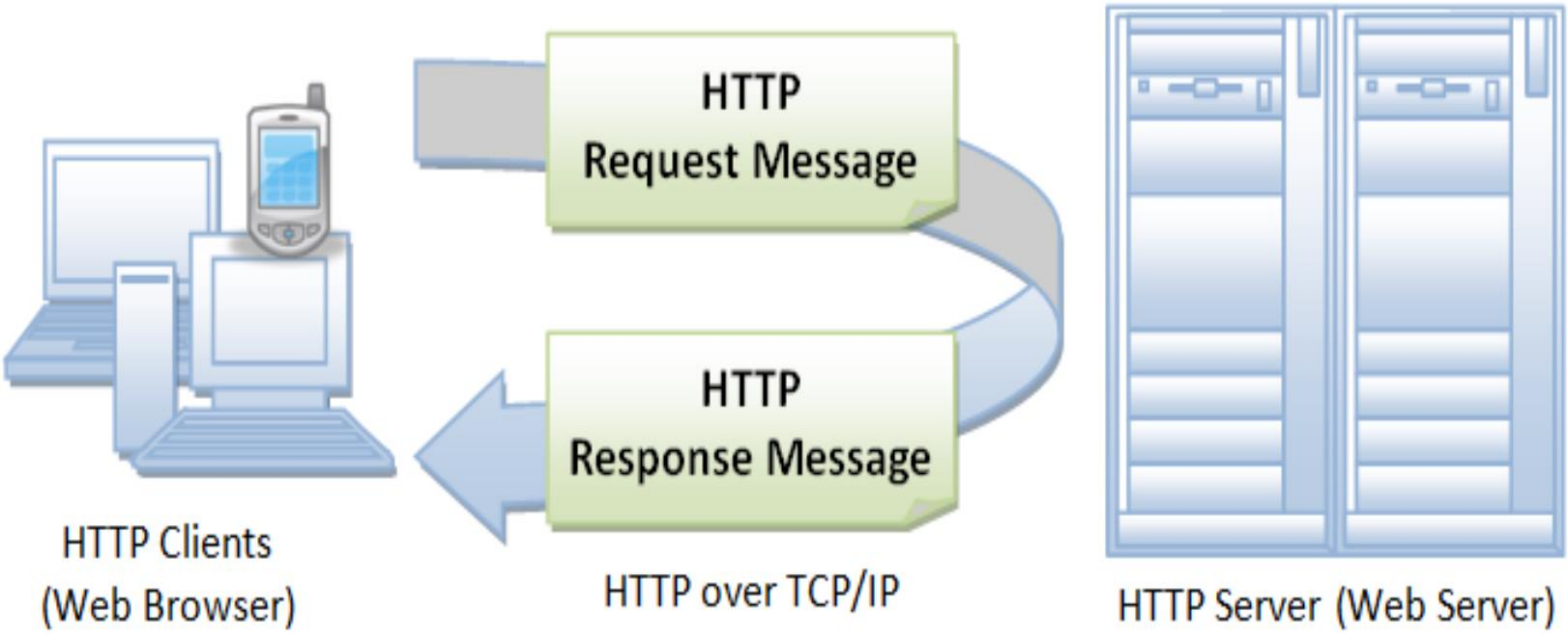


```
| 2018-07-13 | 18:53:22 |       3 | P          |
| 2018-07-13 | 18:54:00 |       1 | P          |
| 2018-07-15 | 06:35:03 |       1 | P          |
| 2018-07-15 | 19:18:17 |       1 | P          |
| 2018-07-15 | 19:38:57 |       1 | P          |
| 2018-07-15 | 19:46:41 |       1 | P          |
| 2018-07-19 | 08:19:59 |       3 | P          |
| 2018-07-19 | 11:45:36 |       1 | P          |
| 2018-07-19 | 13:20:19 |       1 | P          |
| 2018-07-19 | 13:20:24 |       3 | P          |
| 2018-07-19 | 13:40:53 |       1 | P          |
| 2018-07-19 | 13:40:58 |       3 | P          |
| 2018-07-19 | 14:05:10 |       3 | P          |
| 2018-07-19 | 14:05:56 |       1 | P          |
+------------+----------+---------+------------+
17 rows in set (0.00 sec)

MariaDB [school]> select * from MCQ;
+------------+----------+---------+---------+------+
| date       | time     | rollno  | quesno  | ans  |
+------------+----------+---------+---------+------+
| 2018-07-19 | 14:06:51 |       1 |       1 | C    |
| 2018-07-19 | 14:07:36 |       3 |       1 | A    |
+------------+----------+---------+---------+------+
2 rows in set (0.00 sec)

MariaDB [school]>
```

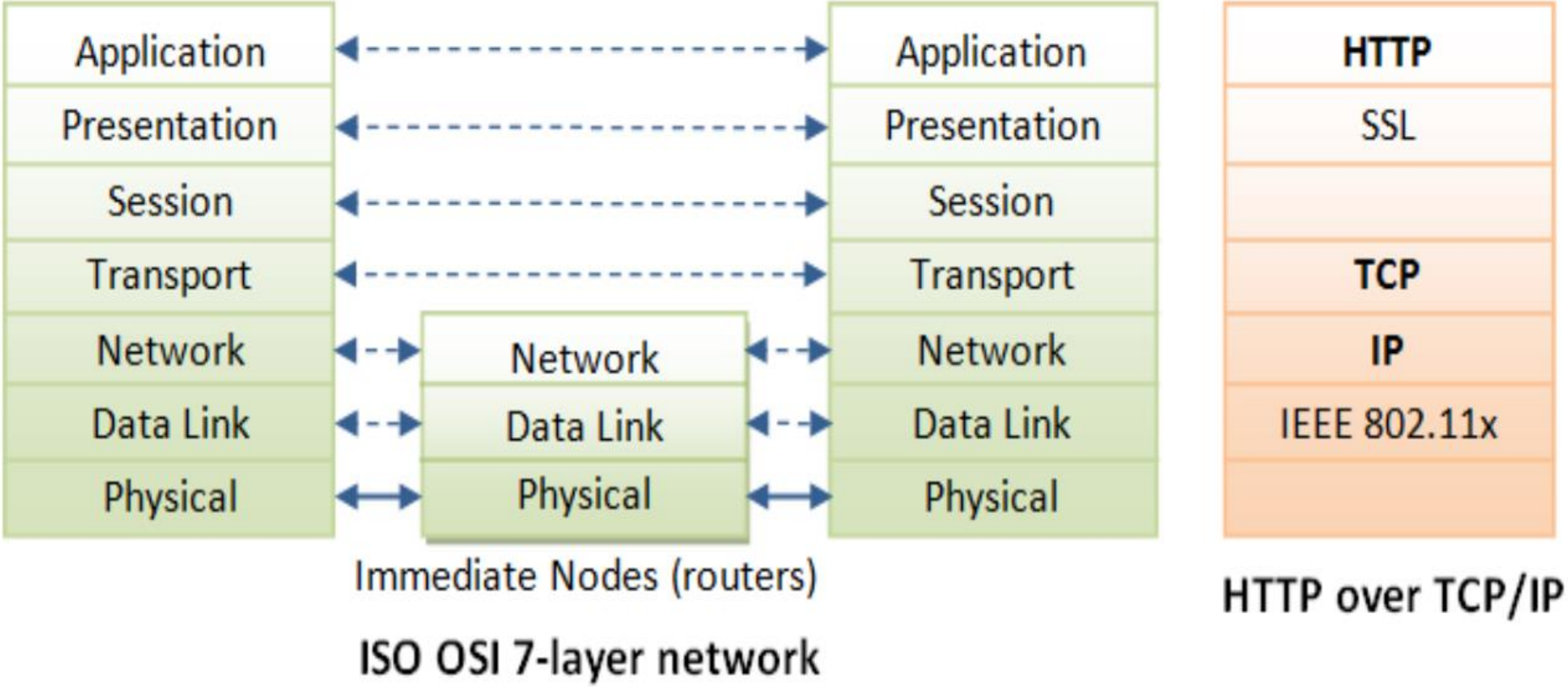
HyperText Transfer Protocol (HTTP)

- **It is the most popular application protocol used in the Internet**
- **It is an *asymmetric request-response client-server* protocol**
- **Its is a *pull protocol***
- **Client *pulls* information from the server instead of server *pushes* information down to the client**
- **It supports various kinds of data formats like text, images, multimedia etc.**
- **Its header size is more as comrated with MQTT**

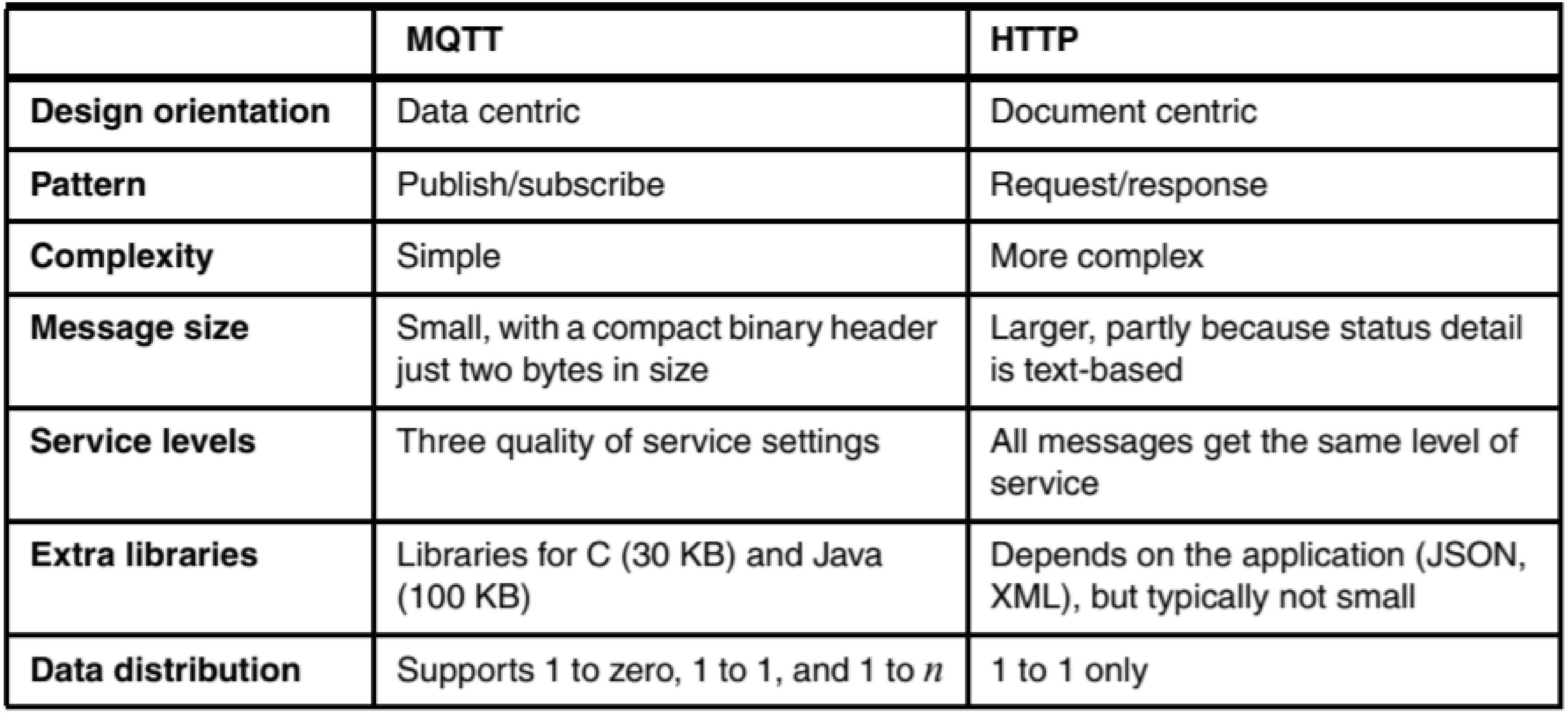
HTTP Architecture



HTTP over TCP



MQTT vs HTTP



	MQTT	HTTP
Design orientation	Data centric	Document centric
Pattern	Publish/subscribe	Request/response
Complexity	Simple	More complex
Message size	Small, with a compact binary header just two bytes in size	Larger, partly because status detail is text-based
Service levels	Three quality of service settings	All messages get the same level of service
Extra libraries	Libraries for C (30 KB) and Java (100 KB)	Depends on the application (JSON, XML), but typically not small
Data distribution	Supports 1 to zero, 1 to 1, and 1 to *n*	1 to 1 only

Refernces

1.ttps://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html [for HTTP Protocol]

2. http://docs.oasis-open.org/mqtt/mqtt/v5.0/csprd01/mqtt-v5.0-csprd01.html [MQTT Protocol]