

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO ĐỀ TÀI BÀI TẬP LỚN**  
**HỌC PHẦN LẬP TRÌNH MẠNG - 2122I\_INT3304-1**  
**TÊN ĐỀ TÀI: CHƯƠNG TRÌNH KIỂU PUBLISH/SUBSCRIBE**  
**CHO CÁC THIẾT BỊ CẢM BIẾN**

Giảng viên hướng dẫn:

- 1. PGS.TS. Nguyễn Hoài Sơn**
- 2. CN. Ngô Minh Hoàng**

Nhóm sinh viên thực hiện:

- |                            |   |                               |
|----------------------------|---|-------------------------------|
| <b>1. Nguyễn An Bằng</b>   | - | Mã sinh viên: <b>18020195</b> |
| <b>2. Lê Quang Quân</b>    | - | Mã sinh viên: <b>18020046</b> |
| <b>3. Hoàng Trung Thực</b> | - | Mã sinh viên: <b>18021257</b> |

*Hà Nội, tháng 11-12 năm 2021*



# Lời cảm ơn và Giới thiệu về dự án

Internet kết nối vạn vật (Internet of Things - IoT) là một trong những chủ đề được quan tâm và nhắc đến rất nhiều trong bối cảnh thời đại khoa học công nghệ ngày càng phát triển như hiện nay. Nhiều người thường ví von rằng nó như là một kịch bản của thế giới, khi mà mọi thứ đều được kết nối qua một mạng duy nhất, thậm chí không cần đến sự tương tác giữa người với người hay người với máy. IoT hội tụ những thành tựu phát triển của công nghệ không dây, công nghệ vi cơ điện tử và mạng Internet.

Để đạt được những mục tiêu đó, thời đại IoT đòi hỏi chúng ta cần có những giao thức kết nối mới để có thể hỗ trợ đầy đủ các yêu cầu của các thiết bị vật lý thực tế. Giao thức Message Queuing Telemetry Transport (MQTT) là một trong số đó, và nó đang dần trở nên phổ biến trong bối cảnh hiện nay. Trong tài liệu này, chúng tôi sẽ trình bày về một giao thức kiểu publish/subscribe như MQTT cho các thiết bị cảm biến.

Sản phẩm này là kết quả của nhóm chúng tôi trong quá trình học tập và nghiên cứu tại trường Đại học Công nghệ - Đại học Quốc gia Hà Nội. Chúng tôi muốn gửi lời cảm ơn tới các thầy cô giáo vì đã truyền đạt những kiến thức bổ ích và có nhiều ý nghĩa cho việc trau dồi, tích lũy kinh nghiệm để theo đuổi ngành học.

Đặc biệt, chúng tôi muốn gửi lời cảm ơn chân thành nhất tới PGS.TS. Nguyễn Hoài Sơn và giảng viên Ngô Minh Hoàng - những người trực tiếp giảng dạy bộ môn Lập trình mạng và hướng dẫn cho chúng tôi những bước tiếp cận cơ bản về các giao thức mạng, IoT.

Bên cạnh tài liệu này, chúng tôi có giới thiệu mã nguồn được viết bằng ngôn ngữ Java và các tài liệu liên quan của dự án tại <https://github.com/honest113/mqtt>. Mặc dù đã có nhiều cố gắng nhưng tài liệu khó tránh khỏi những thiếu sót nhất định. Chúng tôi rất mong nhận được những phản hồi tích cực cùng ý kiến đóng góp của các thầy cô giáo cùng các bạn để dự án tiếp tục được cải tiến, hoàn thiện hơn.

*Hà Nội, tháng 12 năm 2021*

***Nhóm tác giả***

# Mục lục

<b>1</b>	<b>Tổng quan</b>	<b>6</b>
<b>2</b>	<b>Phân tích thiết kế của hệ thống</b>	<b>7</b>
2.1	Thiết kế chương trình . . . . .	7
2.2	Thiết kế giao thức . . . . .	8
2.2.1	Giao thức giữa publisher và broker . . . . .	8
2.2.2	Giao thức giữa subscriber và broker . . . . .	9
2.3	Các yêu cầu và phản hồi trong chương trình . . . . .	10
2.3.1	Các yêu cầu được hỗ trợ . . . . .	10
2.3.2	Các phản hồi trả về từ chương trình . . . . .	11
<b>3</b>	<b>Triển khai xây dựng chương trình</b>	<b>12</b>
3.1	Các công nghệ được sử dụng . . . . .	12
3.2	Giải thích một số thuật toán chính . . . . .	12
3.2.1	Server . . . . .	12
3.2.2	Publisher . . . . .	12
3.2.3	Subscriber . . . . .	13
<b>4</b>	<b>Cách khởi chạy chương trình</b>	<b>14</b>
4.1	Chạy bằng cách câu lệnh giao thức . . . . .	14
4.2	Chạy tự động sử dụng tệp script . . . . .	14
<b>5</b>	<b>Định hướng phát triển trong tương lai</b>	<b>15</b>
<b>6</b>	<b>Tài liệu tham khảo</b>	<b>16</b>

## **Danh sách bảng**

1	Bảng mô tả các yêu cầu được hỗ trợ trong chương trình . . . . .	10
2	Bảng mô tả các phản hồi trả về từ chương trình . . . . .	11

## **Danh sách hình vẽ**

1	Các thành phần chính của chương trình . . . . .	7
2	Giao thức giữa publisher và broker . . . . .	8
3	Giao thức giữa subscriber và broker . . . . .	9

## **1 Tổng quan**

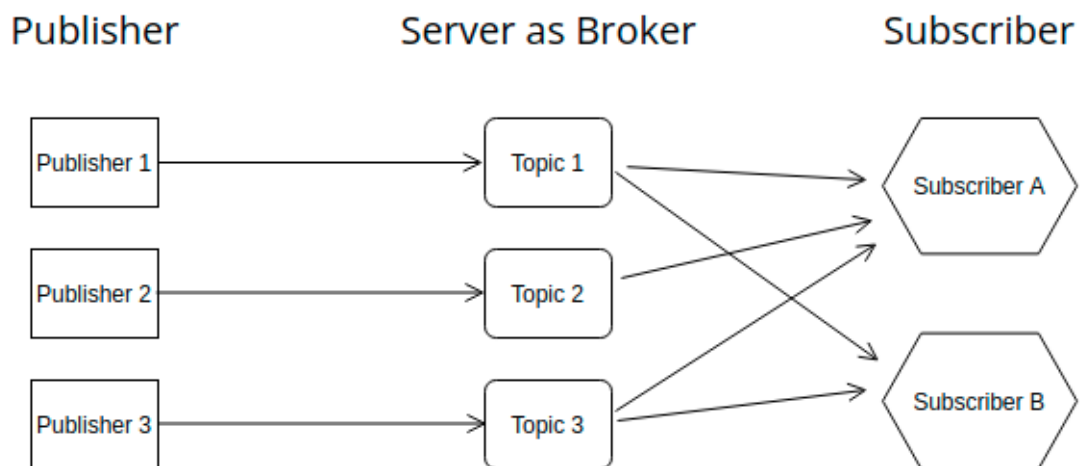
Chương trình của chúng tôi hoạt động kiểu Publish/Subscribe như MQTT, gồm ba thành phần chính: Server đóng vai trò Broker, chương trình sinh dữ liệu Publisher và chương trình hiển thị thông tin Subscriber với các chức năng chính như sau:

- Subscriber kết nối với Broker để đăng ký nhận thông tin về topic (Subscriber được đăng ký nhiều topic).
- Publisher kết nối với Broker để đăng ký topic và publish data (Publisher được publish thông tin về 1 topic).
- Subscriber nhận được dữ liệu từ topic đã đăng ký và hiển thị thông tin cho người dùng.
- Nhiều Publisher cùng kết nối đến Broker và publish dữ liệu với các topic khác nhau.
- Nhiều Subscriber cùng kết nối với Broker, nhận và hiển thị thông tin từ các topic khác nhau đã đăng ký.

## 2 Phân tích thiết kế của hệ thống

### 2.1 Thiết kế chương trình

Chương trình của chúng tôi có 3 thành phần chính: Server đóng vai trò là Broker, Publisher và Subscriber. Trong đó các publisher và subscriber độc lập với nhau, giao tiếp bí mật thông qua trung gian là server.



**Hình 1:** Các thành phần chính của chương trình

- Publisher sẽ có các thông tin như id, name của publisher đó và topic mà publisher phát thông tin, mỗi publisher chỉ được phát thông tin về một topic.
- Subscriber bao gồm các thông tin như id, name và một danh sách (list) các topic mà subscriber đăng ký nhận thông tin, mỗi subscriber được phép đăng ký nhận thông tin từ nhiều topic.
- Server đóng vai trò là broker, có nhiệm vụ xử lý các kết nối của client (publisher và subscriber), xử lý dữ liệu được publish bởi publisher và truyền dữ liệu cho đúng subscriber.

## 2.2 Thiết kế giao thức

Các publisher và subscriber độc lập với nhau, có nghĩa là 2 bên không biết gì về nhau cả, publisher không biết những subscriber nào đăng ký nhận thông tin về topic mà mình publish và ngược lại.

### 2.2.1 Giao thức giữa publisher và broker

```
Publisher: <Enter name>
Server:    200 Hello <Publisher name>
Publisher: GET LIST TOPIC
Server:    201 LIST TOPIC OK:
           <--- list topic info --->
Publisher: SET TOPIC
Server:    202 SET TOPIC OK
Publisher: <Valid Topic Name>
Server:    221 REGISTER DONE
Publisher: START PUB
Server:    203 START PUB OK
Publisher: STOP PUB
Server:    204 STOP PUB OK
Publisher: QUIT
Server:    500 BYE
```

**Hình 2:** Giao thức giữa publisher và broker

Theo quy ước của chúng tôi, tên topic được đặt theo quy chuẩn:

***topic\_name/locationA/sensorA***

Trong giao thức trên, tên topic chỉ được chấp nhận nếu nó phân thành ít nhất 3 cấp.



### 2.2.2 Giao thức giữa subscriber và broker

```
Subscriber: <Enter name>
Server:      200 Hello <Subscriber name>
Subscriber: GET LIST TOPIC
Server:      201 LIST TOPIC OK:
              <--- list topic info --->
Subscriber: SUB TOPIC
Server:      212 SUB TOPIC OK
Subscriber: <Topic name>
Server:      206 UPDATE TOPIC SUCCESSFUL
Subscriber: GET MY TOPIC
Server:      208 TOPIC:
              <--- topic info --->
Subscriber: UNSUB TOPIC
Server:      213 UNSUB TOPIC OK
Subscriber: <Topic name>
Server:      206 UPDATE TOPIC SUCCESSFUL
Subscriber: QUIT
Server:      500 BYE
```

**Hình 3:** Giao thức giữa subscriber và broker

Lưu ý rằng các subscriber có thể thực hiện lọc theo dõi các topic bằng cách sử dụng kí tự \* đại diện cho thành phần tùy biến trong tên topic. Ví dụ: sử dụng *\*/Room\_A/* để lọc, đăng ký nhận tin từ tất cả các topic liên quan đến các sensor trong phòng A.

## 2.3 Các yêu cầu và phản hồi trong chương trình

### 2.3.1 Các yêu cầu được hỗ trợ

Câu lệnh	Đối tượng	Giải thích
GET LIST TOPIC	Publisher & Subscriber	Lấy thông tin về topic hiện có trên server
SET TOPIC	Publisher	Set topic cho publisher
START PUB	Publisher	Publisher sau khi đăng ký topic có thể dùng lệnh này để bắt đầu phát data (ngẫu nhiên)
STOP PUB	Publisher	Dừng việc publish data
GET MY TOPIC	Subscriber	Hiển thị các topic mà subscriber đăng ký
SUB TOPIC	Subscriber	Subscriber đăng ký nhận thông tin từ topic
UNSUB TOPIC	Subscriber	Subscriber hủy đăng ký nhận tin từ topic
QUIT	Publisher & Subscriber	Kết thúc phiên làm việc

**Bảng 1:** Bảng mô tả các yêu cầu được hỗ trợ trong chương trình

### 2.3.2 Các phản hồi trả về từ chương trình

Phản hồi	Đối tượng nhận	Giải thích
200 Hello	Publisher & Subscriber	Hello response
201 LIST TOPIC OK	Publisher & Subscriber	Thông tin về list topic
202 SET TOPIC OK	Publisher	Server đã chấp nhận yêu cầu set topic của publisher
212 SUB TOPIC OK	Subscriber	Server đã chấp nhận yêu cầu đăng ký topic của subscriber
213 UNSUB TOPIC OK	Subscriber	Server đã chấp nhận yêu cầu hủy đăng ký topic của subscriber
221 REGISTER DONE	Publisher	Publisher đã hoàn tất thông tin
203 START PUB OK	Publisher	Publisher tự động sinh dữ liệu
204 STOP PUB OK	Publisher	Publisher ngừng tự động sinh dữ liệu
205 RECEIVE MESSAGE OK	Publisher	Publisher publish thông tin thành công
206 UPDATE TOPIC SUCCESSFUL	Subscriber	Subscriber cập nhật list topic thành công
208 TOPIC:	Subscriber	List topic đang đăng ký của subscriber
400 BAD REQUEST	Publisher & Subscriber	Không rõ lệnh thực thi
411 SET TOPIC ERROR	Publisher	Publisher đăng ký topic thất bại
412 START PUB FAIL - REGISTER REQUIRE	Publisher	Publisher chưa hoàn tất thông tin
413 TOPIC NOT FOUND	Subscriber	Không tìm thấy topic
423 INVALID TOPIC NAME	Publisher	Tên topic không hợp lệ
500 BYE	Publisher & Subscriber	Kết thúc kết nối

**Bảng 2:** Bảng mô tả các phản hồi trả về từ chương trình

## 3 Triển khai xây dựng chương trình

### 3.1 Các công nghệ được sử dụng

Chúng tôi đã triển khai xây dựng chương trình với thành phần chính sử dụng ngôn ngữ Java và có sử dụng một đoạn mã Shell nhằm hỗ trợ việc thực thi chương trình một cách tự động, đơn giản hơn.

Trong quá trình thực hiện, chúng tôi đã sử dụng các IDE là *IntelliJ IDEA* và *Visual Studio Code*. Khi chương trình đã được khởi tạo nền tảng cơ bản, chúng tôi sử dụng Github nhằm mục đích lưu trữ, quản lý nâng cấp và công bố.

### 3.2 Giải thích một số thuật toán chính

#### 3.2.1 Server

Server quản lý các topic, publisher và subscriber lần lượt bằng các thuộc tính sau:

```
ArrayList<Topic> listTopics = new ArrayList<Topic>();  
ArrayList<PublisherData> IPublisherDatas = new ArrayList<PublisherData>();  
ArrayList<SubscriberData> ISubscriberDatas = new ArrayList<SubscriberData>();
```

#### 3.2.2 Publisher

- Mỗi khi một publisher kết nối đến server, server sẽ tạo một đối tượng *PublisherData* chứa tên của publisher đó và thêm đối tượng này vào *IPublisherDatas*.
- Khi publisher thiết lập topic, chương trình sẽ kiểm tra *listTopics* và tính hợp lệ của tên topic. Nếu hợp lệ, server sẽ tạo một đối tượng topic mới (*newTopic*) và đồng thời thiết lập thuộc tính topic của publisher là *newTopic* và thêm *newTopic* vào *listTopics*.
- Sau khi đã có đầy đủ thông tin (name và topic), publisher có thể tiến hành phát thông tin về topic đó. Server nhận được thông tin từ publisher sẽ tiến hành làm giàu (enrich data) và chuyển cho các subscriber quan tâm.
- Khi publisher yêu cầu kết thúc phiên làm việc (QUIT), server sẽ xóa publisher khỏi *IPublisherDatas* và tiến hành đóng kết nối.

### 3.2.3 Subscriber

- Khi có một subscriber mới kết nối đến, server sẽ tạo một đối tượng Subscriber-Data chứa tên của subscriber đó và thêm đối tượng này vào ***lSubscriberDatas***.
- Subscriber gửi yêu cầu “GET LIST TOPIC” để truy vấn các topic hiện có trong ***listTopics***, sau đó đăng ký nhận thông tin của topic từ listTopics này.
- Khi một subscriber đăng ký một topic thì topic đó sẽ được thêm vào ***listTopics*** của subscriber này (listTopics này của riêng từng subscriber và khác ***listTopics*** của server được nói đến ở trên).
- Khi subscriber hủy đăng ký một topic thì chương trình sẽ loại bỏ topic đó ra khỏi ***listTopics*** của subscriber.
- Khi nhận được thông tin publish của một publisher nào đó, server sẽ kiểm tra topic của publisher và tìm kiếm những subscriber đang đăng ký nhận tin của topic đó (dựa vào ***listTopics*** của các subscriber) nhằm chuyển thông tin đến các subscriber.
- Khi subscriber yêu cầu kết thúc phiên làm việc (QUIT), server sẽ xóa subscriber khỏi ***lSubscriberDatas*** và tiến hành đóng kết nối.

## 4 Cách khởi chạy chương trình

### 4.1 Chạy bằng cách câu lệnh giao thức

```
# compile common
$ javac -cp lib/gson-2.8.8.jar common/*.java

# compile server
$ javac server/*.java

# compile publisher and subscriber
$ javac clients/*.java

# run server
$ java -cp ../lib/gson-2.8.8.jar server.Server

# run publisher
$ java -cp ../lib/gson-2.8.8.jar clients.Publisher 127.0.0.1

# run subscriber
$ java -cp ../lib/gson-2.8.8.jar clients.Subscriber 127.0.0.1
```

### 4.2 Chạy tự động sử dụng tệp script

```
$ chmod +x script.sh
$ ./script.sh
```

## **5 Định hướng phát triển trong tương lai**

Hiện tại, sản phẩm của chúng tôi đang được xây dựng nhằm hoàn thiện và đáp ứng các yêu cầu, tính năng chính. Tuy nhiên, việc giao tiếp và hiển thị thông tin qua màn hình câu lệnh khiến cho việc theo dõi chưa được trực quan, ít nhiều tạo ra bất tiện cho người dùng. Trong thời gian tới, chúng tôi sẽ cố gắng tiếp tục phát triển các tính năng và bổ sung việc thao tác, giao tiếp với chương trình thông qua giao diện để việc sử dụng được thuận tiện hơn.

## **6 Tài liệu tham khảo**

- [1] PGS.TS. Nguyễn Hoài Sơn (2021), Giáo trình môn học Lập trình mạng, UET.
- [2] CN. Ngô Minh Hoàng (2021), Hướng dẫn thực hành Lập trình mạng, UET.
- [3] Các nguồn tài liệu khác về MQTT và giao thức kiểu publish/subscribe.