



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

IT4735

Internet of Things and Applications

Giảng viên: TS. Phạm Ngọc Hưng
Viện Công nghệ Thông tin và Truyền thông
hungpn@soict.hust.edu.vn

Nội dung

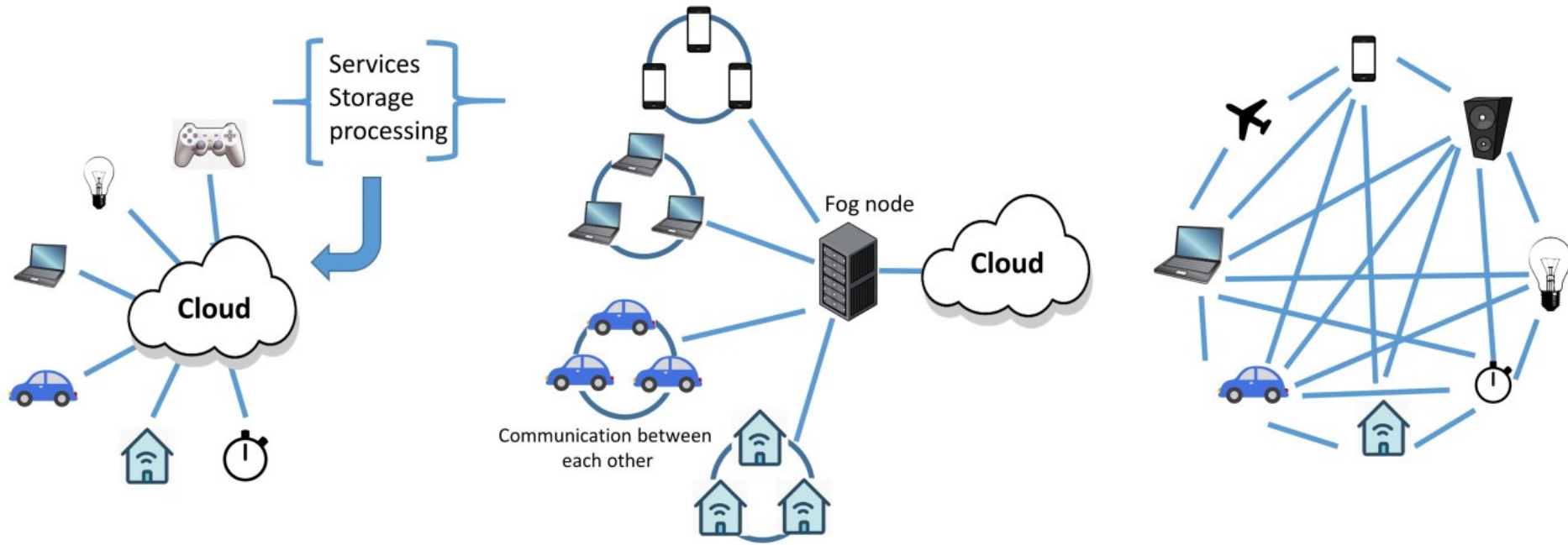
- Chương 1. Tổng quan về IoT
- Chương 2. Hệ thống IoT và các công nghệ
- Chương 3. Lập trình IoT
- Chương 4. An toàn và Bảo mật IoT
- Chương 5. Xây dựng ứng dụng IoT

Chương 2. Hệ thống IoT và các công nghệ

- 2.1. Kiến trúc hệ thống IoT
- 2.2. Truyền thông trong IoT
- 2.3. Các giao thức truyền thông cho IoT
- 2.4. Nền tảng đám mây IoT

2.1. Kiến trúc hệ thống IoT

Kiến trúc tổng quan của hệ thống IoT

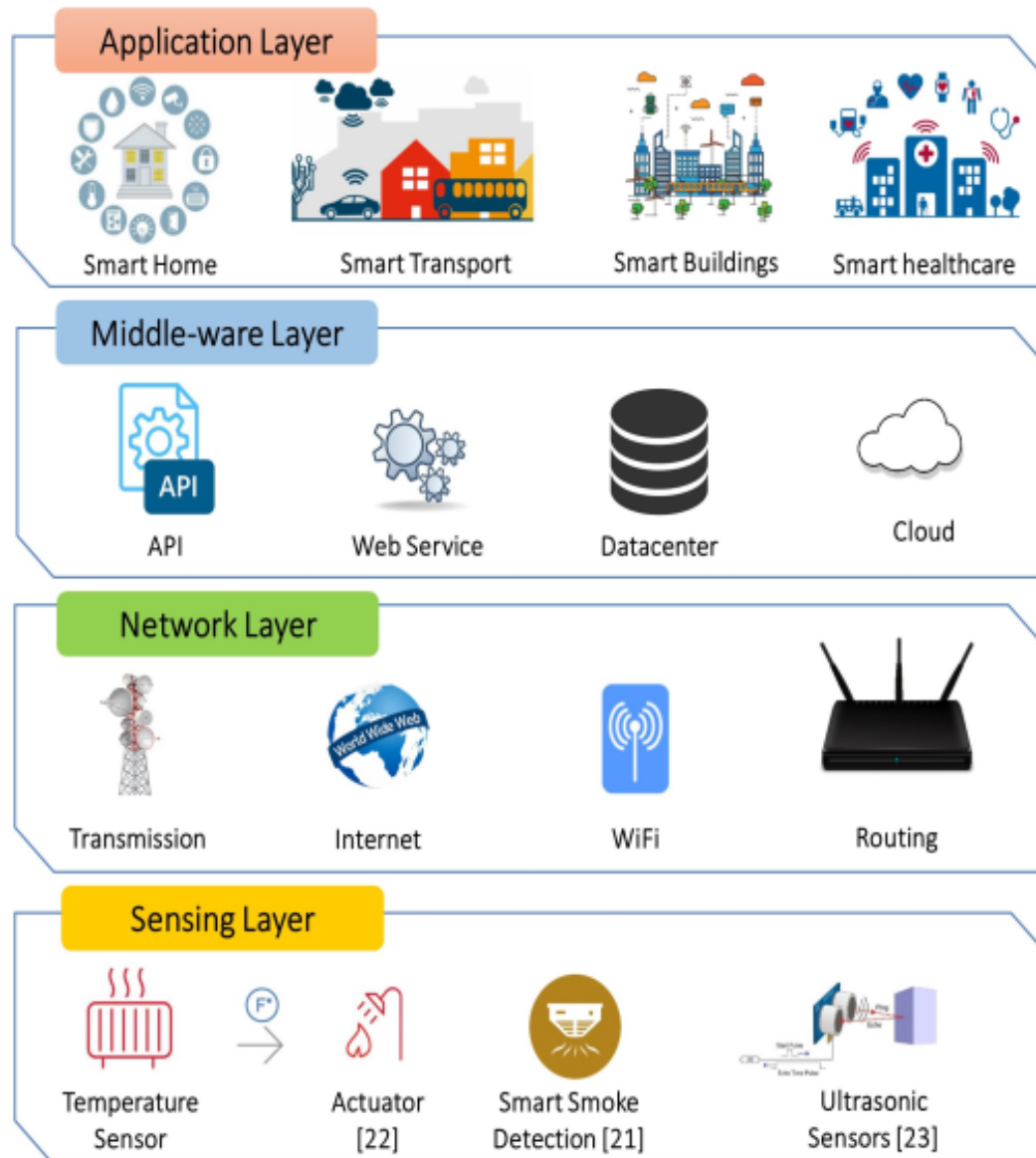


Present and Future Architectures of IoT

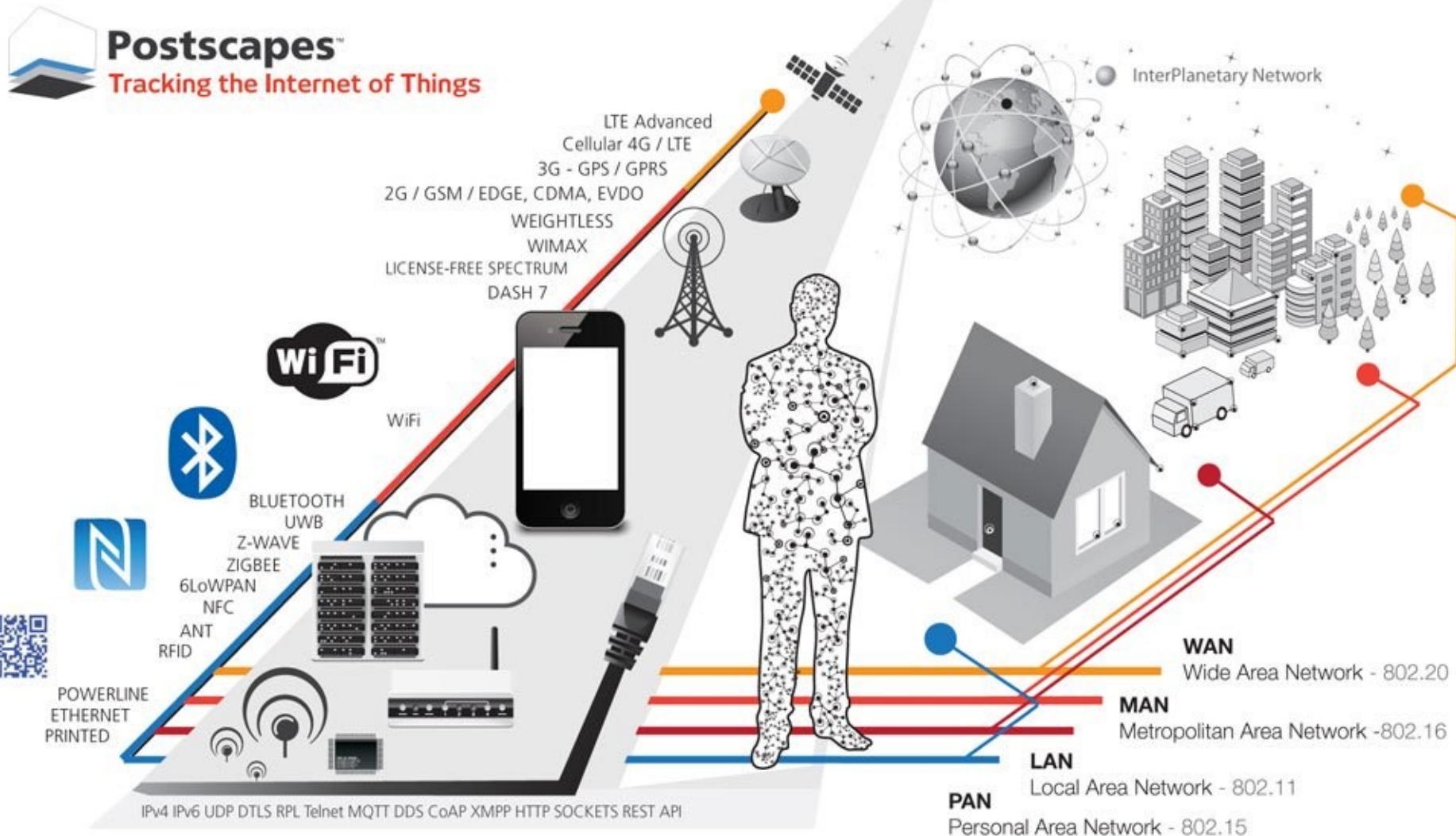
Source:

https://www.researchgate.net/publication/333909259_A_Survey_on_IoT_Security_Application_Are_as_Security_Threats_and_Solution_Architectures

Kiến trúc phân tầng của hệ thống IoT



2.2. Truyền thông trong IoT

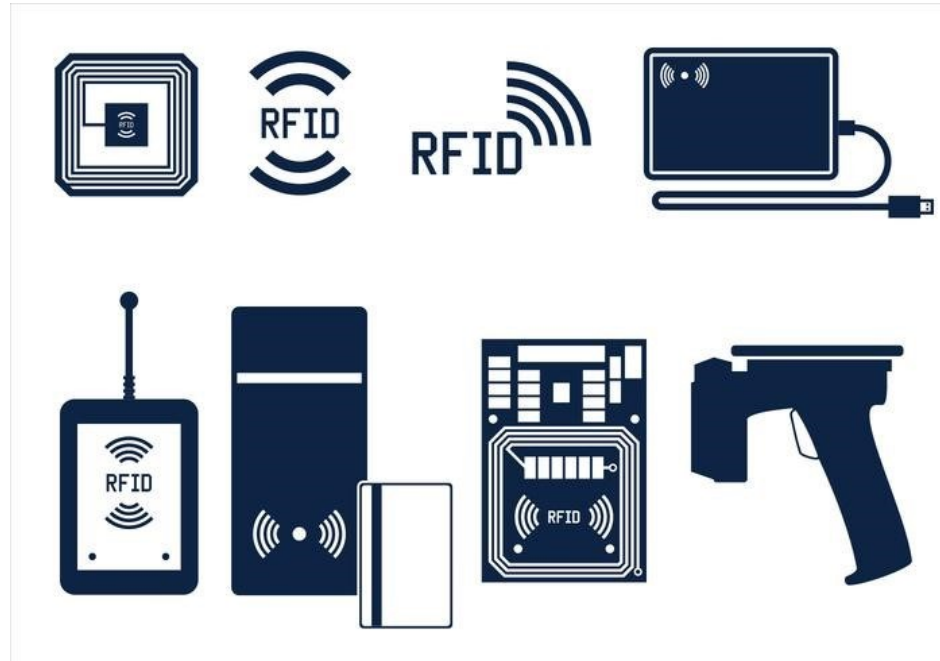


Truyền thông trong IoT

- NFC and RFID
- Low-Energy Bluetooth
- Low-Energy Wireless
- Radio Protocols
- LTE-A
- WiFi-Direct (P2P)

Truyền thông trong IoT

- RFID (Radio-frequency identification)



<https://www.youtube.com/watch?v=MAA9JpGraoU>

Frequency: 120–150 kHz (LF), 13.56 MHz (HF), 433 MHz (UHF), 865-868 MHz (Europe) 902-928 MHz (North America) UHF, 2450-5800 MHz (microwave), 3.1–10 GHz (microwave)

Range: 10cm to 200m

Examples: Road tolls, Building Access, Inventory

Truyền thông trong IoT

- NFC (Near-Field Communications)



Frequency: 13.56 MHz

Range: < 0.2 m

Examples: Smart Wallets/Cards, Action Tags, Access Control

Truyền thông trong IoT

- Bluetooth



Frequency: 2.4GHz

Range: 1-100m

Examples: Hands-free headsets, key dongles, [fitness trackers](#)

Truyền thông trong IoT

- WiFi



Frequency: 2.4 GHz, 3.6 GHz and 4.9/5.0 GHz bands.

Range: Common range is up to 100m but can be extended.

Applications: Routers, Tablets, etc

More: <https://www.postscapes.com/internet-of-things-technologies/>

Truyền thông trong IoT

- GSM (Global System for Mobile communications)



Frequency: Europe: 900MHz & 1.8GHz, US: 1.9GHz & 850MHz, Full List can be found [here](#).

Data Rate: 9.6 kbps

Examples: Cell phones, M2M, smart meter, asset tracking

More: <https://www.postscapes.com/internet-of-things-technologies/>

Truyền thông trong IoT

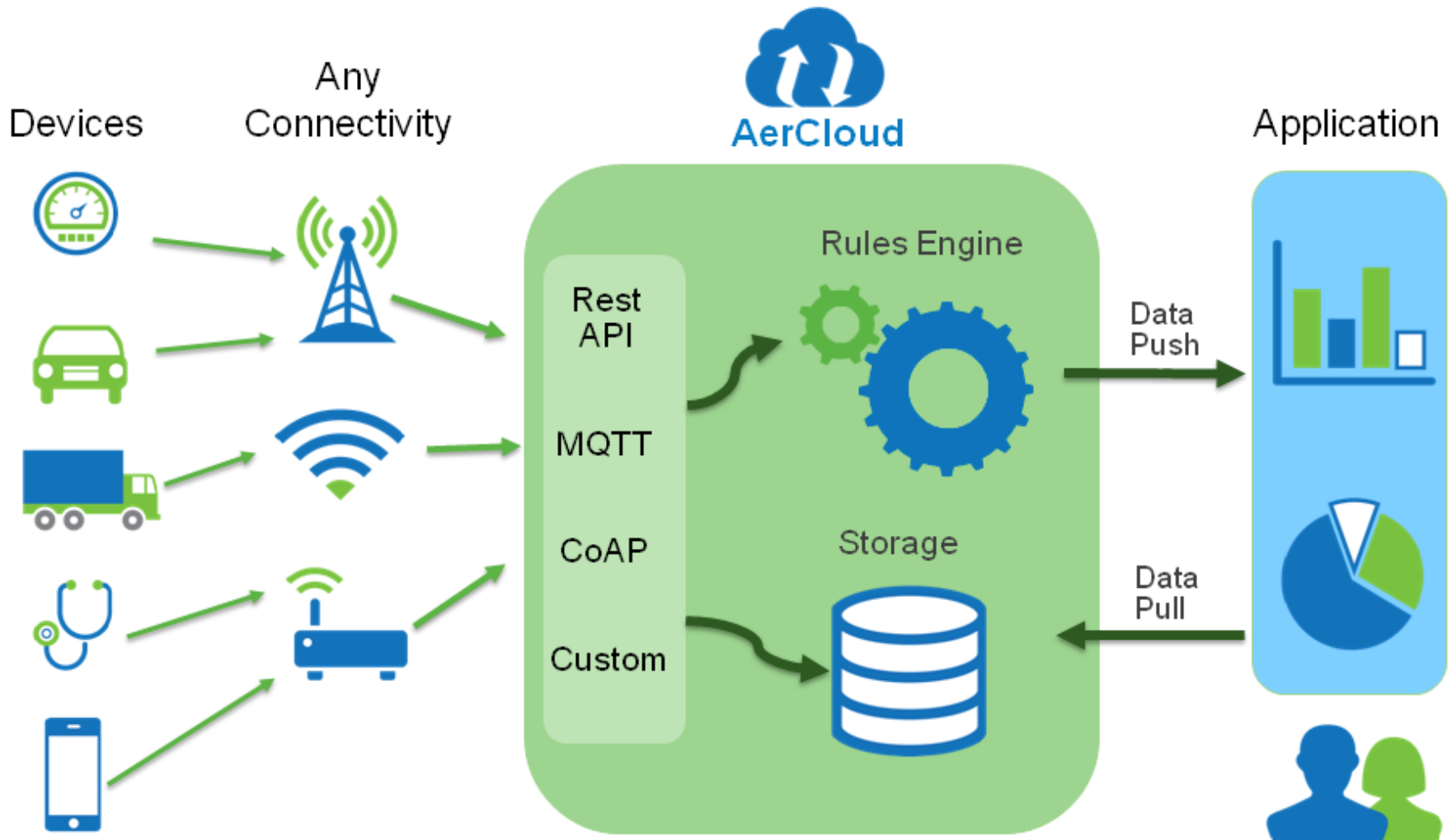
Additional:

- [3G](#)
- [4G LTE](#)
- [ANT](#)
- [Dash7](#)
- [Ethernet](#)
- [GPRS](#)
- [PLC / Powerline](#)
- [QR Codes, EPC](#)
- [WiMax](#)
- [X-10](#)
- [802.15.4](#)
- [Z-Wave](#)
- [Zigbee](#)

Backbone: IPv4, IPv6, TCP/UDP

More: <https://www.postscapes.com/internet-of-things-technologies/>

2.3. Các giao thức truyền thông cho IoT



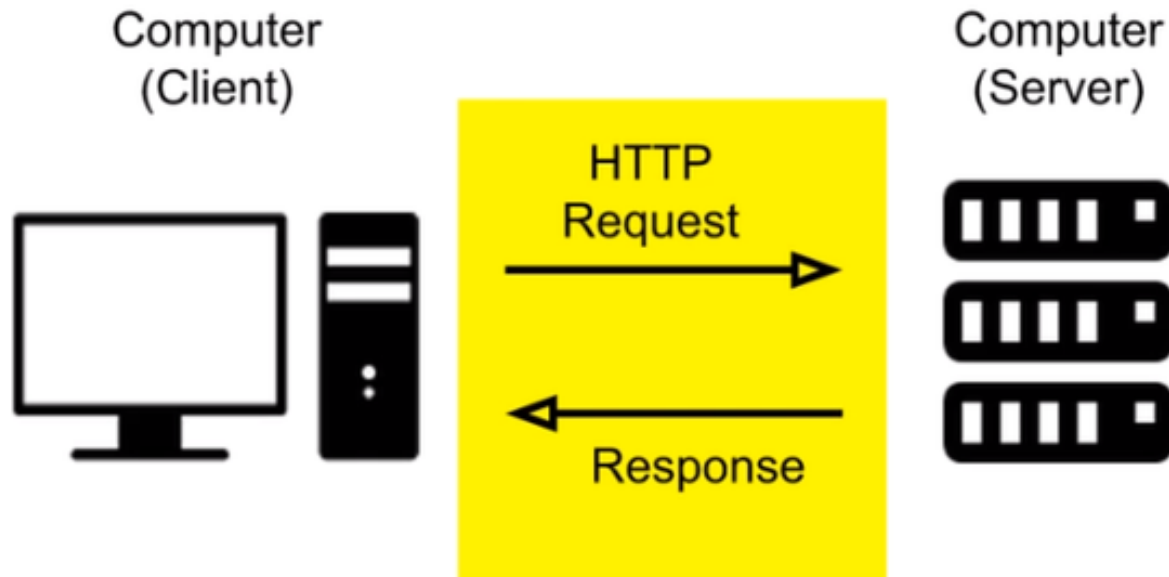
Các giao thức truyền thông

- CoAP (Constrained Application Protocol)
- MQTT (Message Queue Telemetry Transport)
- RESTful HTTP (Representational State Transfer)
- XMPP (Extensible Messaging and Presence Protocol)

<https://www.postscapes.com/internet-of-things-technologies/>

2.3.1. HTTP

HTTP = HyperText Transfer Procol



- Minh họa lập trình trao đổi dữ liệu với HTTP request

Ví dụ ứng dụng sử dụng HTTP

Smart Locker Project

Server back-end
& Web front-end

Application on
Android device
controller

Lockers
Controller
(ATmega2560)

Locker

RFID tag
reader



Ví dụ trao đổi dữ liệu HTTP request

■ Sử dụng thư viện trên Android

```
public String requestUserInfor(String devId, String qrDevId, String
qrCode, int timeout){
    String result = "";
    try {
        String urlApi =
"https://203.171.20.94:8080/api/AccessControl/GetUserInfor";
        URL url = new URL(urlApi);
        HttpURLConnecton conn = (HttpURLConnection)
url.openConnection();
        conn.setRequestMethod("POST");
        conn.setConnectTimeout(timeout);
        conn.setReadTimeout(timeout);
        conn.setRequestProperty("Content-Type", "application/json;
charset=utf-8");
```

Ví dụ trao đổi dữ liệu HTTP request

■ Sử dụng thư viện trên Android

```
public String requestUserInfor(String devId, String qrDevId, String
qrCode, int timeout){
    ...
    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(os, "UTF-8"));
    JSONObject jsonObj = new JSONObject();
    try {
        jsonObj.accumulate("deviceId", devId);
        jsonObj.accumulate("qrCodeId", qrDevId);
        jsonObj.accumulate("qrCodeValue", qrCode);
    } catch (JSONException e) {
        e.printStackTrace();
    }

    writer.write(jsonObj.toString());
    writer.flush();
    writer.close();
    os.close();
    conn.connect();
}
```

Ví dụ trao đổi dữ liệu HTTP request

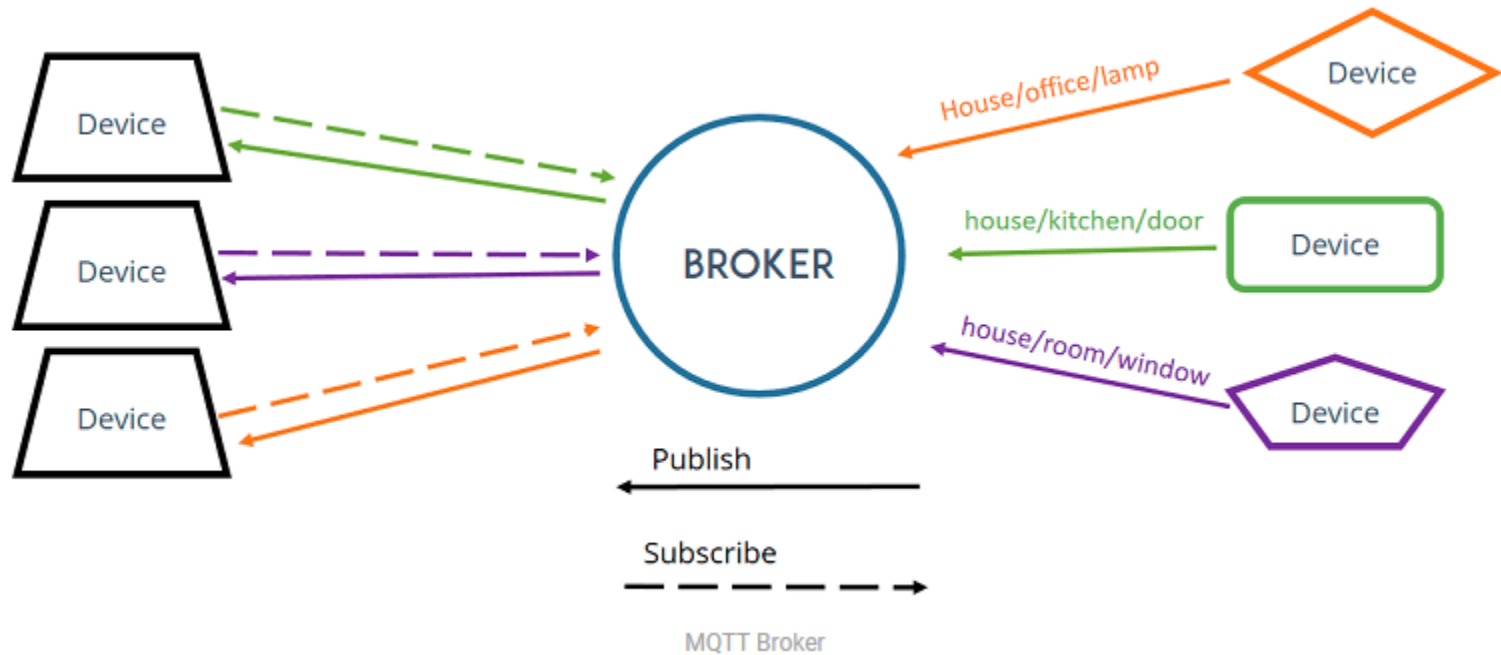
```
public String requestUserInfor(String devId, String qrDevId, String
qrCode, int timeout){
    ...
    StringBuffer sb = new StringBuffer();
    if (conn.getResponseCode() == HttpURLConnection.HTTP_OK) {
        InputStream in = conn.getInputStream();
        int chr;
        while ((chr = in.read()) != -1) {
            sb.append((char) chr);
        }
        in.close();
    } else {
        sb.append(conn.getResponseCode());
    }
    result = sb.toString();
    conn.disconnect();
}
catch(IOException e){
    e.printStackTrace();
}
return result;
}
```

2.3.2. Giao thức MQTT

- **The Messaging and Data Exchange Protocol of the IoT**
- **MQTT** (Message Queuing Telemetry Transport):
 - Giao thức truyền thông điệp (message) theo mô hình publish/subscribe (xuất bản – theo dõi)
 - Sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

Giao thức MQTT

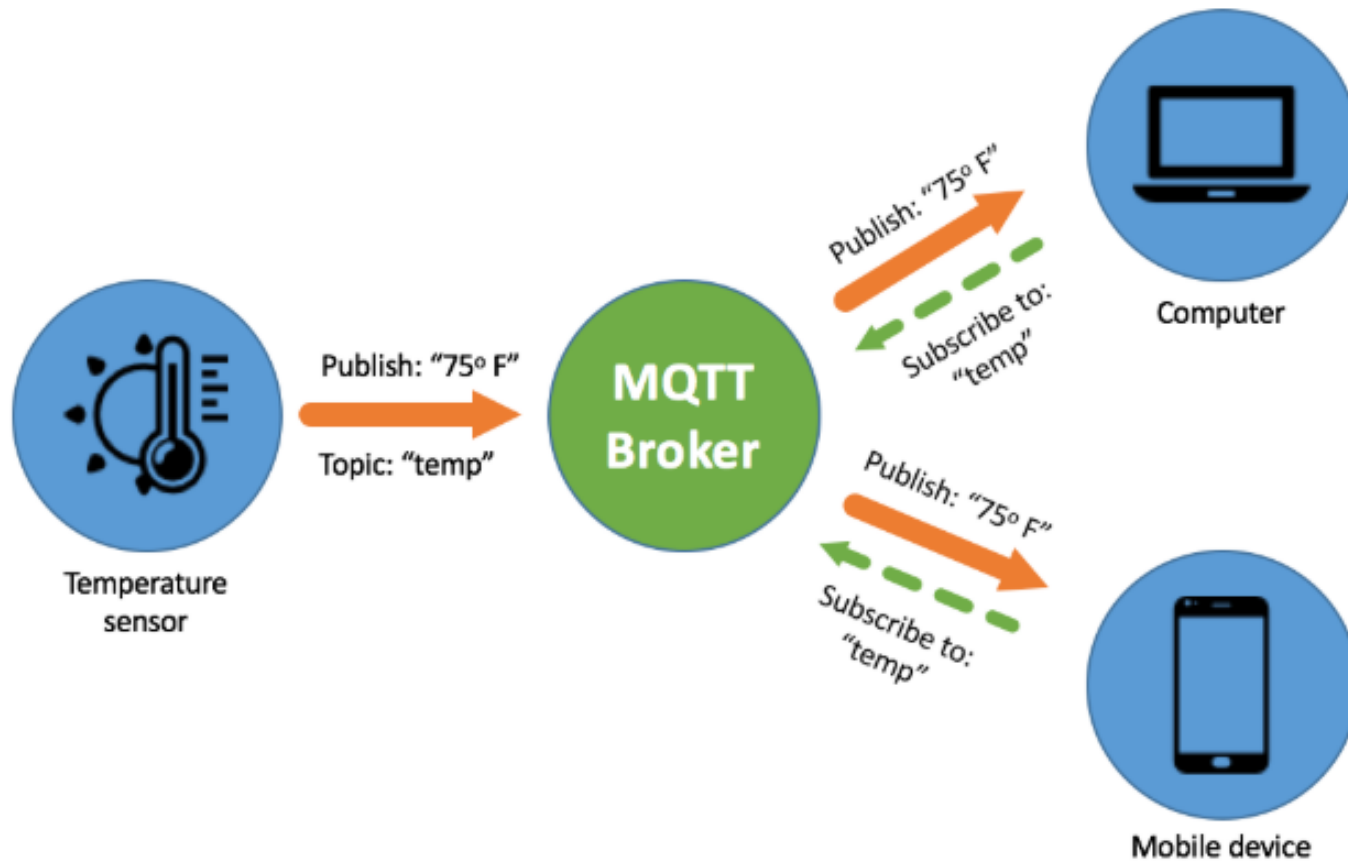
- Kiến trúc mức cao của MQTT gồm 2 thành phần chính:
 - Broker
 - Clients



■ Publish/Subscribe:

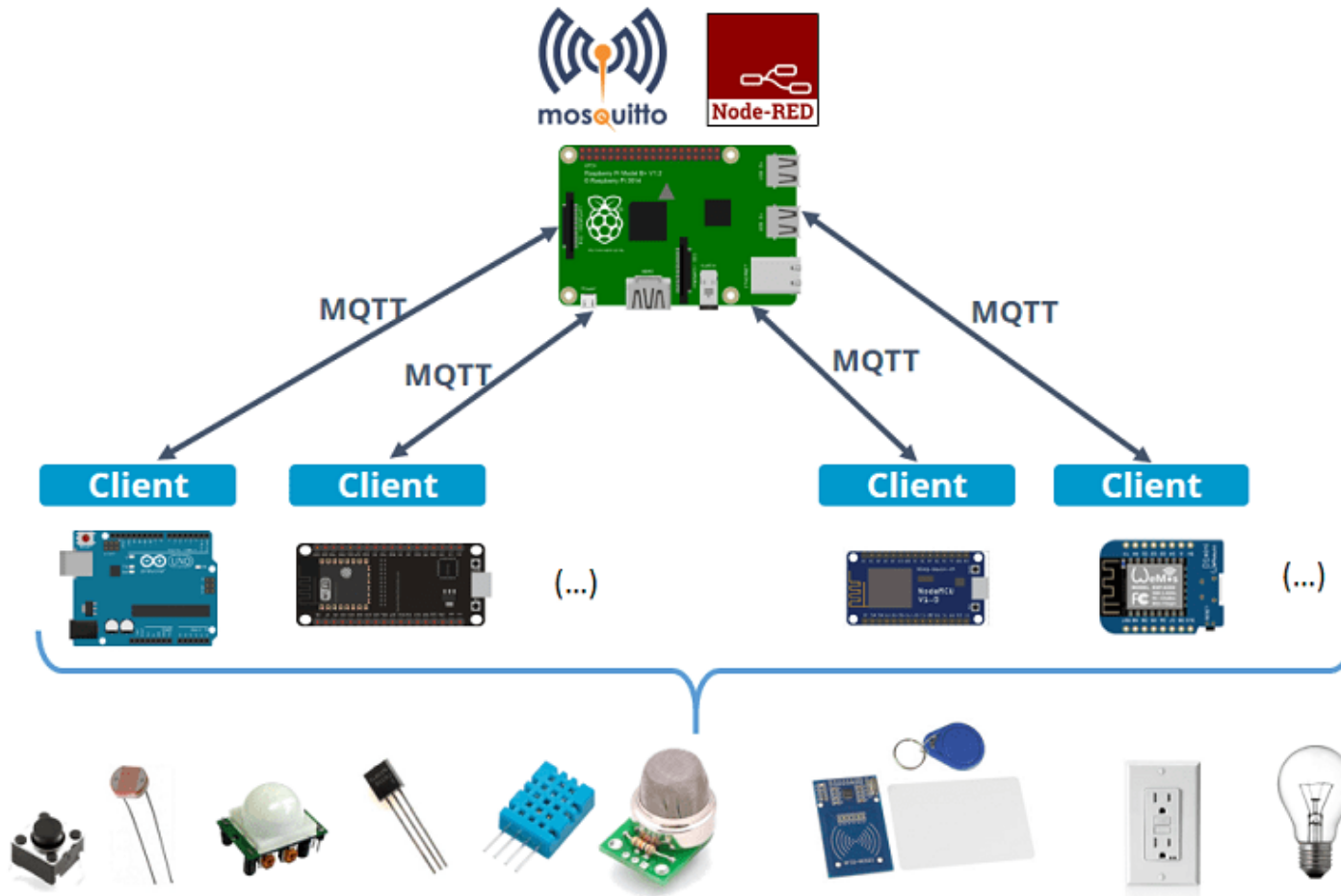
- Trong một hệ thống sử dụng giao thức MQTT, nhiều node trạm (gọi là mqtt client – gọi tắt là client) kết nối tới một MQTT server (gọi là broker).
- Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như “/client1/channel1”, “/client1/channel2”. Quá trình đăng ký này gọi là “**subscribe**”. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký.
- Khi một client gửi dữ liệu tới kênh đó, gọi là “**publish**”.

Giao thức MQTT



<https://www.hivemq.com/mqtt-essentials/>

Giao thức MQTT



<https://smartfactoryvn.com/technology/internet-of-things/giao-thuc-mqtt-la-gi-nhung-ung-dung-cua-mqtt-nhu-the-nao/>

■ QoS (Quality of Service):

- Có 3 tùy chọn **QoS** khi “publish” và “subscribe”:
- **QoS0** Broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP, (“fire and forget”).
- **QoS1** Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
- **QoS2** Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay

Giao thức MQTT

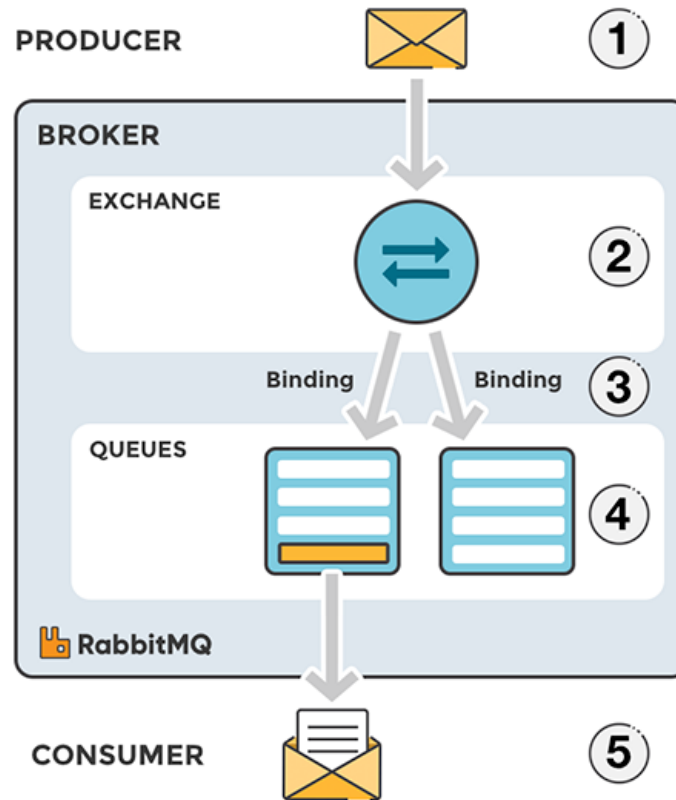
- Ví dụ lập trình với MQTT
- Sử dụng tool MQTTBox

2.3.3. Giao thức AMQP

- RabbitMQ:
 - Một message broker sử dụng giao thức AMQP (Advanced Message Queue Protocol) – Giao thức truyền nhận thông điệp sử dụng hàng đợi.
 - RabbitMQ broker đóng vai trò trung gian lưu trữ cũng như điều phối thông điệp (message) giữa bên gửi (producer) và bên nhận (consumer)

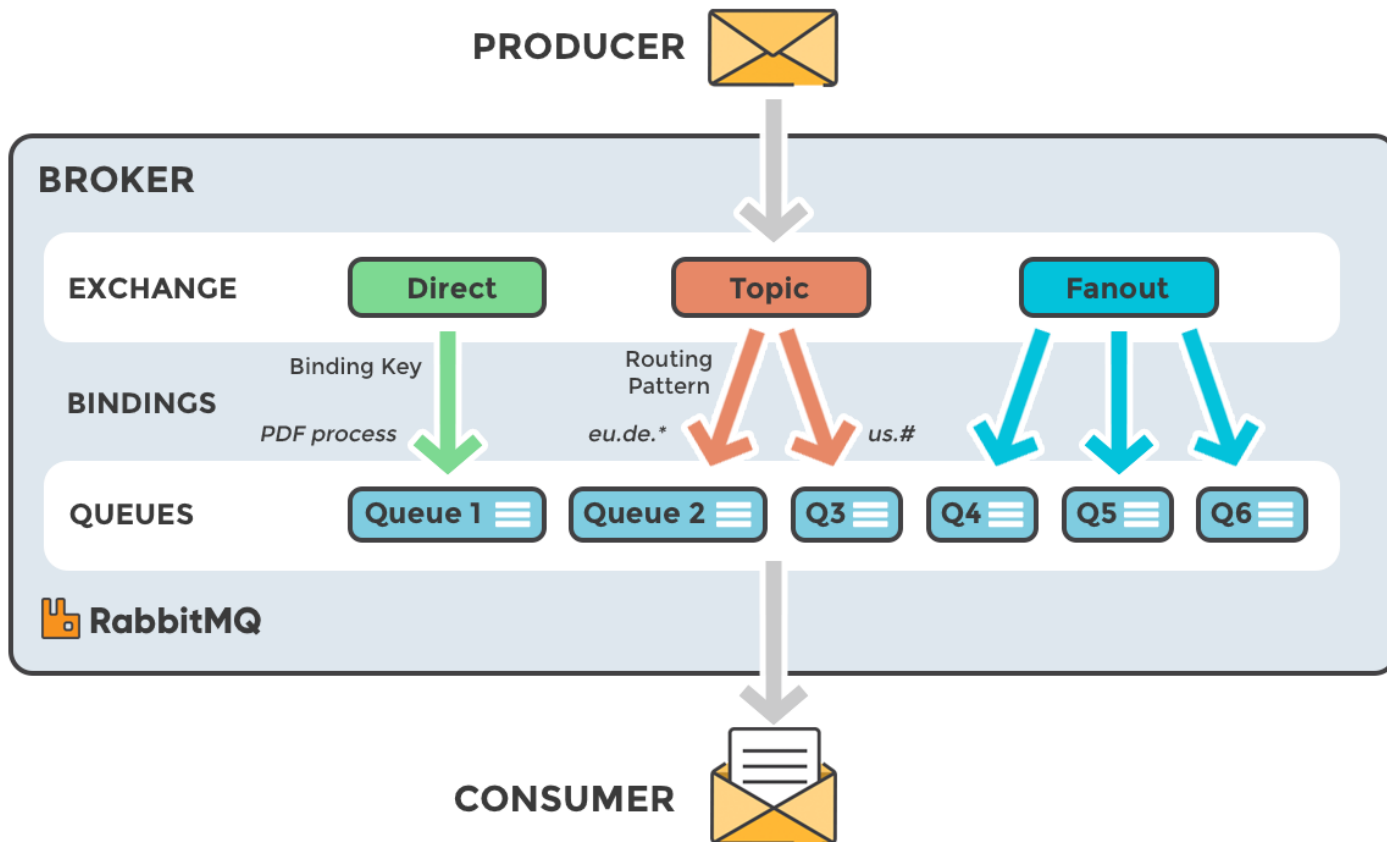
Giao thức AMQP

- RabbitMQ:
 - Luồng gửi nhận message qua RabbitMQ



Giao thức AMQP

- RabbitMQ:
 - 4 loại Exchange: direct, topic, fanout, headers



Giao thức AMQP

- Ví dụ lập trình với RabbitMq

Ví dụ lập trình RabbitMQ

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

import java.nio.charset.StandardCharsets;

public class send {

    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("203.171.20.94");
        factory.setUsername("test");
        factory.setPassword("Navistar@1312");
        try (Connection connection = factory.newConnection();
            Channel channel = connection.createChannel()) {
            channel.queueDeclare(QUEUE_NAME, false, false, false, null);
            String message = "Hello World! Publish message to broker";
            channel.basicPublish("", QUEUE_NAME, null,
                message.getBytes(StandardCharsets.UTF_8));
            System.out.println(" [x] Sent '" + message + "'");
        }
    }
}
```

Sender.java

Ví dụ lập trình RabbitMQ

```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.DeliverCallback;
```

Recv.java

```
public class Recv {
    private final static String QUEUE_NAME = "hello";

    public static void main(String[] argv) throws Exception {
        ConnectionFactory factory = new ConnectionFactory();
        factory.setHost("203.171.20.94");
        factory.setUsername("test");
        factory.setPassword("Navistar@1312");
        Connection connection = factory.newConnection();
        Channel channel = connection.createChannel();

        channel.queueDeclare(QUEUE_NAME, false, false, false, null);
        System.out.println(" [*] Waiting for messages. To exit press CTRL+C");

        DeliverCallback deliverCallback = (consumerTag, delivery) -> {
            String message = new String(delivery.getBody(), "UTF-8");
            System.out.println(" [x] Received '" + message + "'");
        };
        channel.basicConsume(QUEUE_NAME, true, deliverCallback, consumerTag -> { });
    }
}
```

2.4. IoT Cloud Platforms

- Một số nền tảng dịch vụ đám mây trong IoT:
 - IBM BlueMix
 - AWS IoT
 - Google Cloud IoT
 - Azure IoT



<https://www.postscapes.com/internet-of-things-technologies/>