INTERNET OF THINGS

IoT Application Protocols

Professor: Dr.Siavash Khorsandi

Lecturer: Jaber Babaki

Amirkabir University of Technology

Fall 2020

CONTENTS

- Introduction
- Physical and Link Layers Protocols (IoT Access Technologies)
- Application Layer Protocols

Mostly adopted from Chapters 4, 5, and 6 of IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing, Cisco press, 2017

CONTENTS

Mostly adopted from Chapters 4, 5, and 6 of IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing, Cisco press, 2017

WEB PROTOCOL FOR THE INTERNET OF THINGS

- Let's take a closer look at the way HTTP works
  - The protocol is used to interact with web resources (e. g. web pages or forms) on a server.
  - Several methods are available for this purpose.
    - Most inquiries are made to request data ("GET"),
    - Other inquiries data can be transferred, e. g. when an order form has to be filled out in the online shop (method "POST").

- In principle, HTTP methods can also be used for interaction with devices:
  - Getting data from the device ("GET /temperature"), or
  - Control things/actuators ("POST /fan/control").

- Most developers are already familiar with HTTP web services. So, why not just have IoT devices connect to web services?

Application protocols for the Internet of Things, online available, https://www.wespeakiot.com/application-protocols-for-the-

HTTP: NOT NECESSARILY SUITABLE FOR SMALL DEVICES

- There are several reasons why HTTP is not well suited to interacting with constrained devices:
  - HTTP is a heavy weight protocol with many headers and rules
    - HTTP would have a large header (up to several megabytes) which is not sensible for IoT usage with small packets
  - HTTP is a 1-1 protocol.
    - The client makes a request, and the server responds.
    - It is difficult and expensive to broadcast a message to all devices on the network, which is a common use case in IoT applications.

Application protocols for the Internet of Things, online available, https://www.wespeakiot.com/application-protocols-for-the-

IOT PROTOCOL STACK- APPLICATION LAYER PROTOCOLS

- Two of the most popular protocols are CoAP and MQTT.
- Before introducing CoAP and MQTT, we review two data exchange models
- Two Communication (Data Exchange) Models:
  - Request and Response
  - Publish and Subscribe

* IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing, Cisco press, 2017

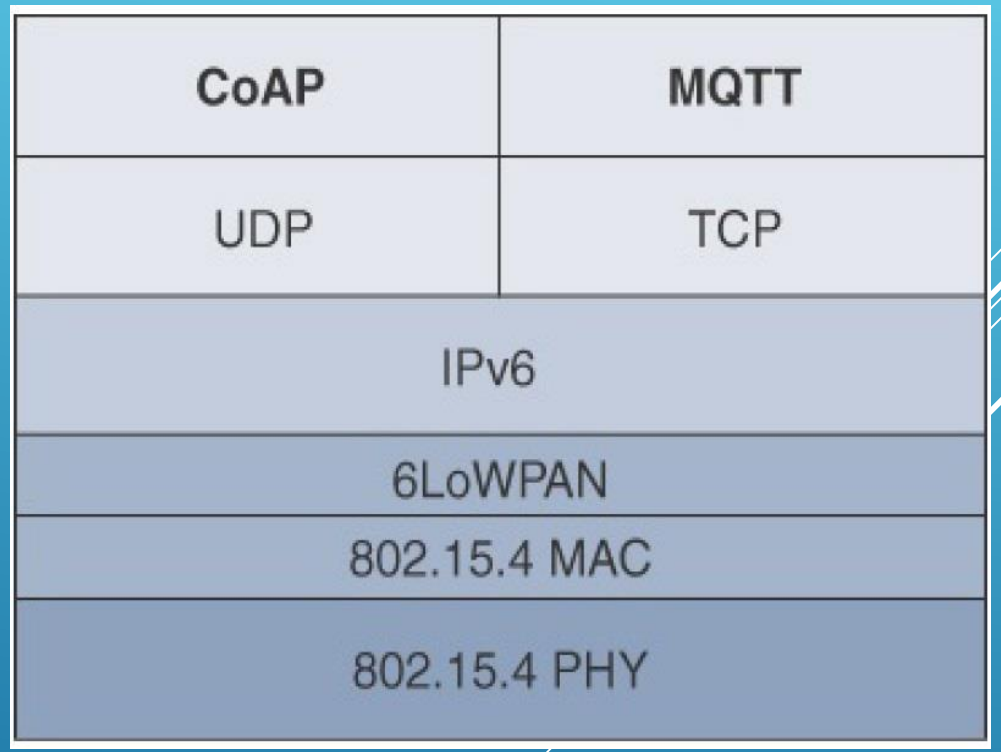
REQUEST-RESPONSE VS. PUBLISH-SUBSCRIBE

- Publish and Subscribe
  - A central source called a broker (also sometimes called a server) which receives and distributes all data.
  - Pub-sub clients can publish data to the broker or subscribe to get data from it—or both.
  - Clients that publish data send it only when the data changes (report by exception, or RBE).
  - Clients that subscribe to data automatically receive it from the broker/server, but again, only when it changes.
  - The broker does not store data; it simply moves it from publishers to subscribers.
    - When data comes in from a publisher, the broker promptly sends it off to any client subscribed to that data.

Request-response vs. publish-subscribe, part 1: What's the diff? , **online available,** https://blog.opto22.com/optoblog/request-

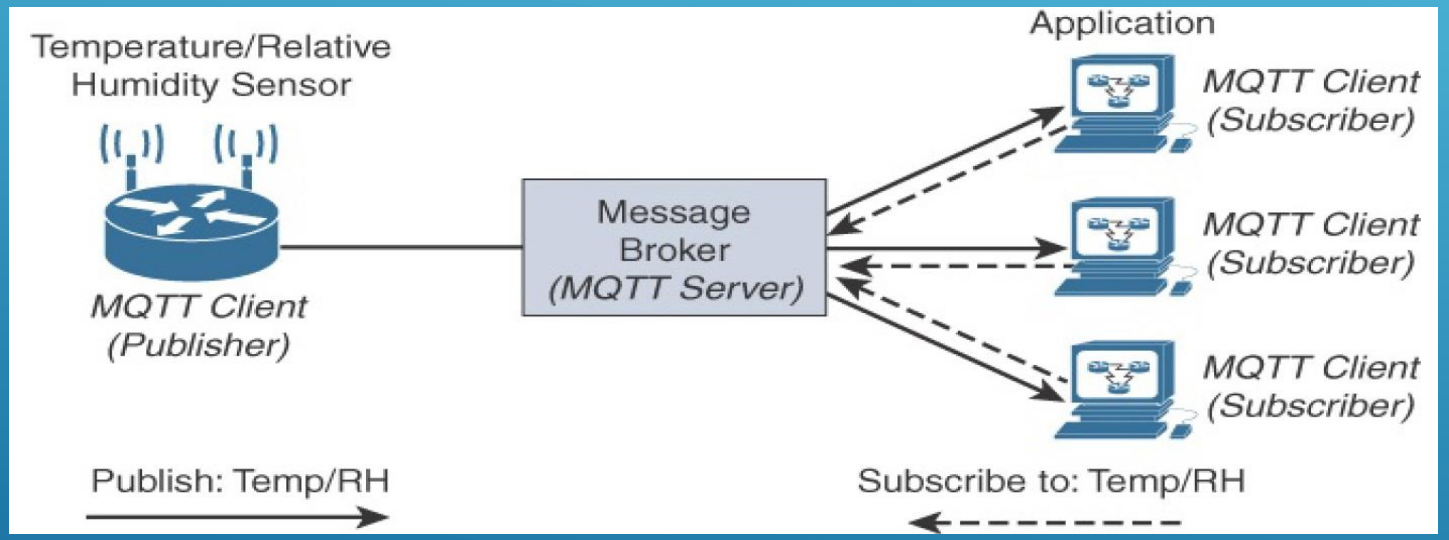
IOT APPLICATION LAYER PROTOCOLS

- Constrained Application Protocol (CoAP)
  - Uses both client-server and pub-Sub methods
  - UDP-based
- Message Queuing Telemetry Transport (MQTT)
  - Based on pub-sub
  - TCP-based



IOT APPLICATION LAYER PROTOCOLS-MQTT

- Application Layer Protocols- MQTT
  - MQTT Publish/Subscribe Framework



  - Is like Twitter

* IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Thing, Cisco press, 2017

IOT APPLICATION LAYER PROTOCOLS-TERMINOLOGY OF MQTT

- **Client**:
  - Clients can be persistent or transient.
    - Persistent clients maintain a session with the broker
    - transient clients are not tracked by the broker.
  - Clients often connect to the broker through libraries and SDKs.
    - There are over a dozen libraries available for C, C++, Go, Java, C#, PHP, Python, Node.js, and Arduino.

Get to Know MQTT: The Messaging Protocol for the Internet of Things, online available,

IOT APPLICATION LAYER PROTOCOLS-TERMINOLOGY OF MQTT

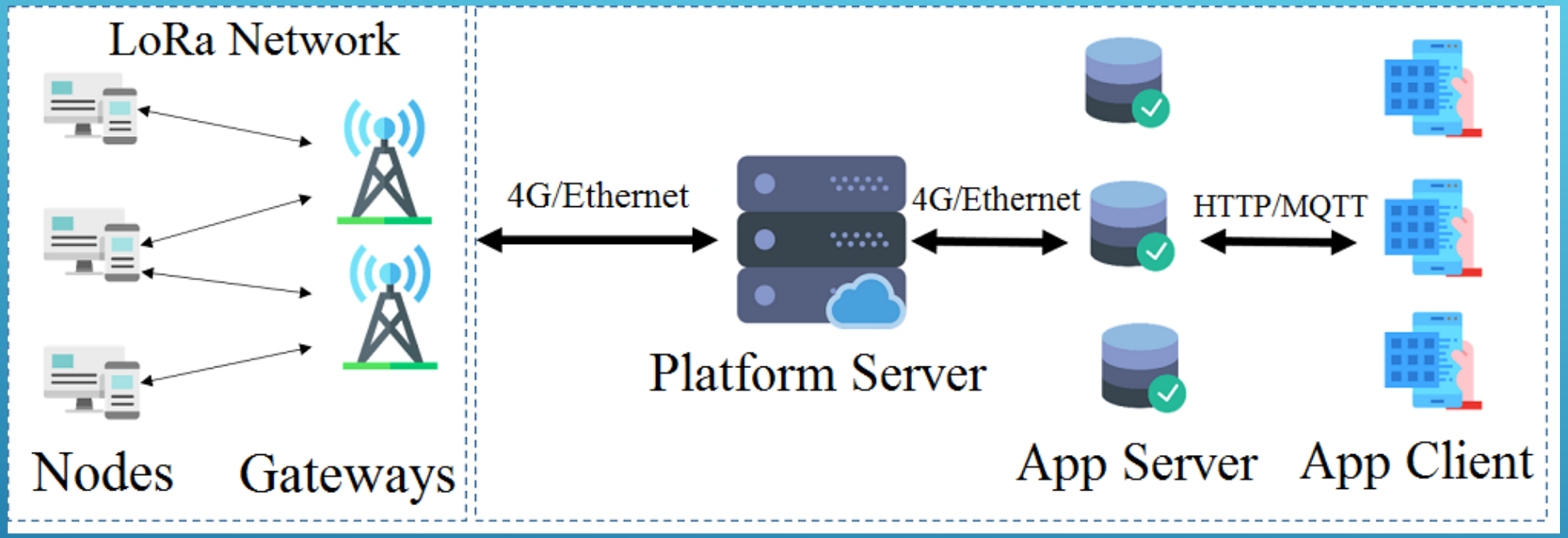
- **Broker**
  - The broker is the software that receives all the messages from the publishing clients and sends them to the subscribing clients.
  - It holds the connection to persistent clients.
  - Since the broker can become the bottleneck or result in a single point of failure, it is often clustered for scalability and reliability.
    - It is up to the implementers to decide how to create a scalable broker layer.
  - MQTT brokers include
    - Commercial: HiveMQ, Xively, AWS IoT, and Loop,
    - an open source: Mosquitto.

Get to Know MQTT: The Messaging Protocol for the Internet of Things, online available,

IOT APPLICATION LAYER PROTOCOLS-TERMINOLOGY OF MQTT

- **Topic**
  - A topic in MQTT is an endpoint to that the clients connect.
  - It acts as the central distribution hub for publishing and subscribing messages.
  - A topic is a well-known location for the publisher and subscriber.
  - Topics are simple, hierarchical strings, delimited by a forward slash.
    - For example, building1/room1/temperature and building1/room1/humidity are valid topic names. Subscribers can choose to subscribe to a specific topic or all the subtopics through wildcards.
    - A subscription to building1/+/temperature will automatically subscribe to the temperature topic of all the rooms in the building.
    - Similarly, building1/#/ will match all the topics available under building1. Refer to the MQTT specification for more details on the wildcards.

Get to Know MQTT: The Messaging Protocol for the Internet of Things, online available,


LoRa Network
4G/Ethernet
4G/Ethernet
HTTP/MQTT
Platform Server
Nodes
Gateways
App Server
App Client

spreading factor which has a value between $7-12$

transmit power which has a value of $[2, 5, 8, 11, 14]$ dBm

bandwidth ($[125, 250, 500]$ KHz).

Bit rate?

Time on air?

Range?

Energy consumption?

Transmission Bit Rate:

$$R_b = SF \times \frac{BW}{2^{SF}} \times CR \quad \text{(bps)}$$

BW =125 kHz
CR =4/5

SF	R_b	Ttx (1 Byte)	Ttx (8 Byte)	Ttx (20 Byte)
7	5.47 Kb/s	1.463 ms	11.703 ms	29.26 ms
8	3.12 Kb/s	2.56 ms	20.48 ms	51.2 ms
9	1.76 Kb/s	4.55 ms	36.40 ms	91.02 ms
10	0.98 Kb/s	8.19 ms	65.54 ms	163.84 ms
11	0.54 Kb/s	14.89 ms	119.15 ms	297.89 ms
12	0.29 Kb/s	27.3 ms	218.45 ms	546.13 ms

$$T_{sym} = \frac{2^{SF}}{BW}$$

LoRa Receiver Sensitivity : -

$$S = -174 + 10 \log_{10} BW + NF + SNR$$

Where,

BW = Bandwidth in KHz

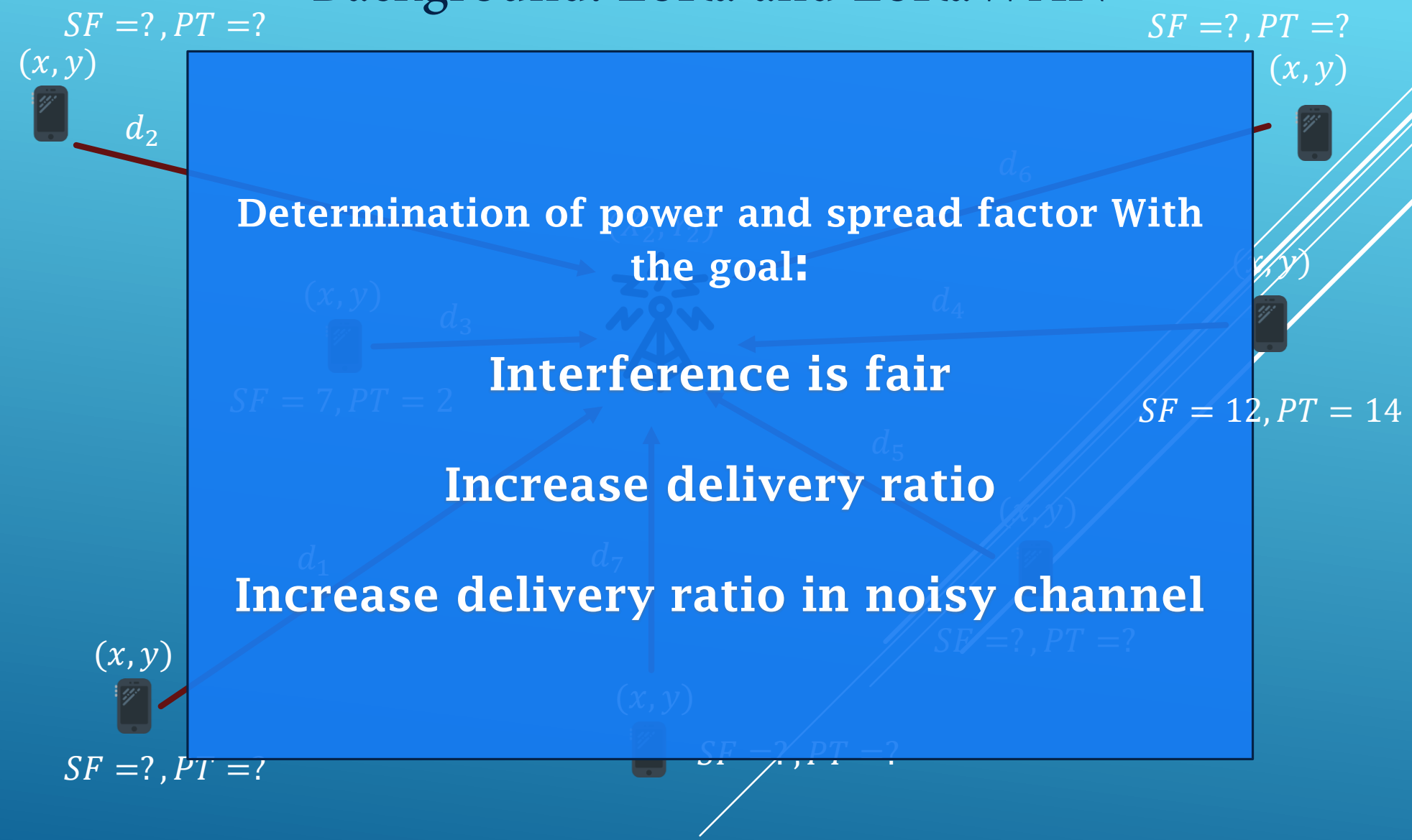
NF = Noise Figure of Receiver in dB

SNR = Signal to Noise Ratio from Table as per SF (Spreading Factor) entered in the calculator

S = Sensitivity in dBm

Spreading factor	Sensitivity [dBm] 125 kHz	Sensitivity [dBm] 250 kHz	Sensitivity [dBm] 500 kHz
SF7	-124	-122	-116
SF8	-127	-125	-119
SF9	-130	-128	-122
SF10	-133	-130	-125
SF11	-135	-132	-128
SF12	-137	-135	-129

Background: LoRa and LoRaWAN



Determination of power and spread factor With the goal:

Interference is fair

Increase delivery ratio

Increase delivery ratio in noisy channel