



Internet of Things (IoT)

DAY 4

Getting devices talking: MQTT, telemetry and commands

August 2 to 6, 2026

KGSP Summer Enrichment Program 2026
KAUST Academy, KAUST

Salah Abdeljabar
salah.abdeljabar@kaust.edu.sa



On Today's Agenda

1. **IoT communication:** publish/subscribe & the broker
2. **MQTT:** the standard protocol for IoT
3. **Telemetry:** sending sensor data to the cloud
4. **Commands:** controlling devices from the cloud

IoT Communication

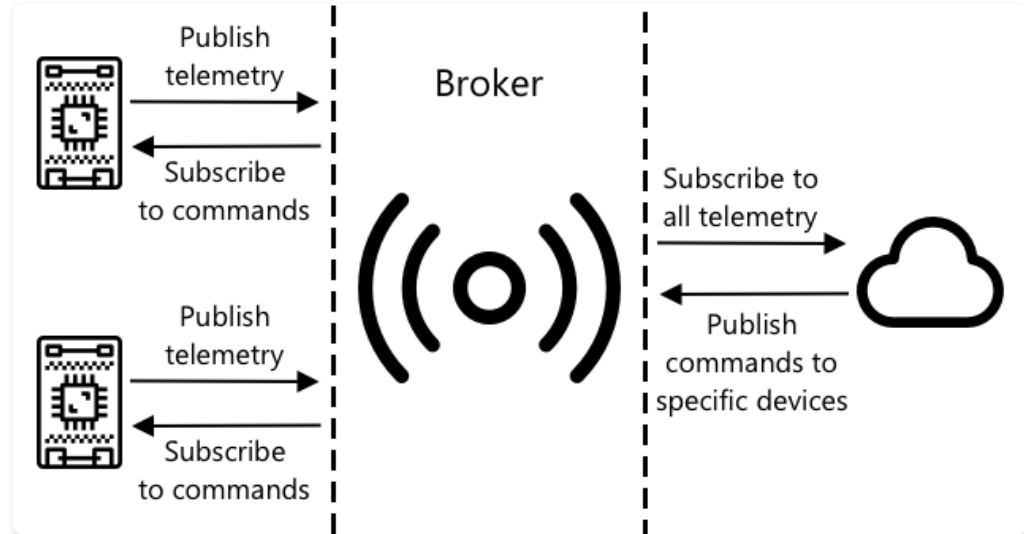


How devices and the cloud **talk**

Publish / Subscribe

Most IoT messaging uses **pub/sub** through a **broker**:

- Devices **publish** telemetry to topics
- Devices **subscribe** to command topics
- The **broker** routes messages between them
- Cloud services subscribe to telemetry & publish commands



Popular Protocols

MQTT

Lightweight, designed for IoT, pub/sub. (*Our focus.*)

AMQP

Robust enterprise messaging.

HTTP / HTTPS

The web standard: simple, but higher overhead.

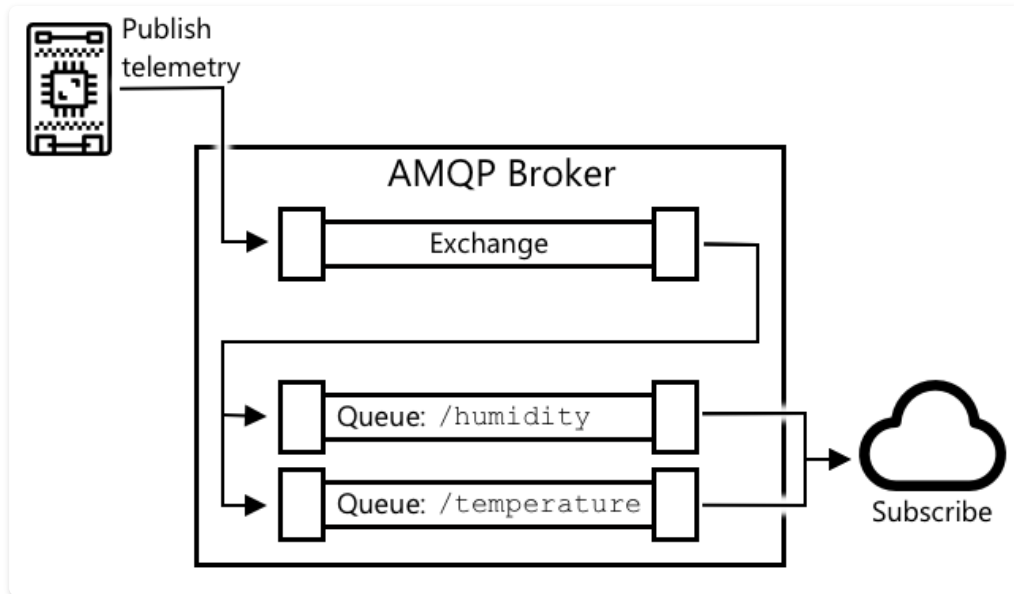
Important

MQTT is the most popular protocol for IoT: small, efficient, and built for constrained devices.

A Closer Look at AMQP

AMQP splits the broker into an **exchange** and **queues**:

- The device publishes into the **exchange**
- The exchange **routes** each message into one or more queues
- Subscribers read from the queues, and a message waits until someone takes it
- Heavier than MQTT, but you get **guaranteed delivery** and richer routing



MQTT hands a message to whoever is listening now. AMQP holds it until it is collected.

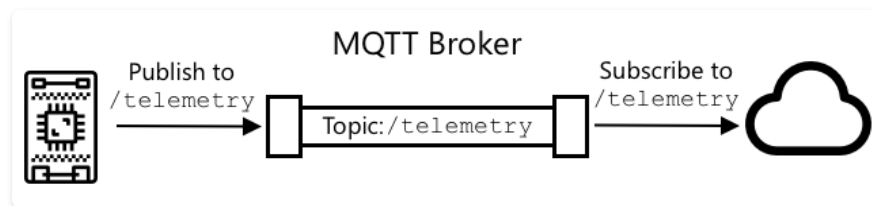
MQTT

The standard messaging protocol for IoT

One Broker, Many Clients

MQTT was designed in **1999** to monitor oil pipelines, later open-sourced by IBM.

- All clients connect to a single **broker**
- Messages route by named **topics**, not to clients directly
- Topics are **hierarchical**: `/device-id/telemetry`
- **Wildcards**: subscribe to `/telemetry/*`



Reliability Features

Quality of Service (QoS)

- **0**: at most once (fire & forget)
- **1**: at least once (acknowledged)
- **2**: exactly once (two-step handshake)

Try it free

Public test broker: test.mosquitto.org
(no account needed, but not private!).

More guarantees

- **Retained**: broker keeps the last value
- **Keep-alive**: checks the connection is live
- **TLS** + username/password or certificates

Telemetry

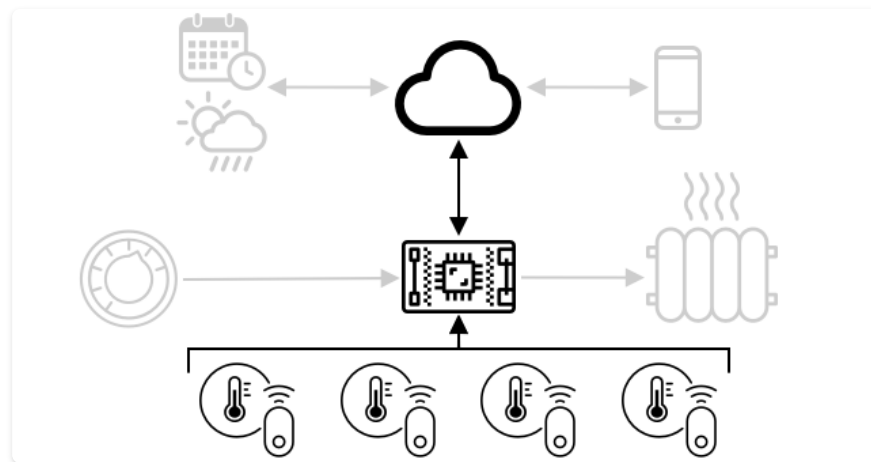
Sending sensor data from device to cloud

What is Telemetry?

From the Greek for "*measuring remotely*": gathering sensor data and sending it to the cloud, usually as **JSON**.

```
{ "light": 300, "temperature": 22.5 }
```

The first telemetry device (1874) sent Mont Blanc weather to Paris over wires.



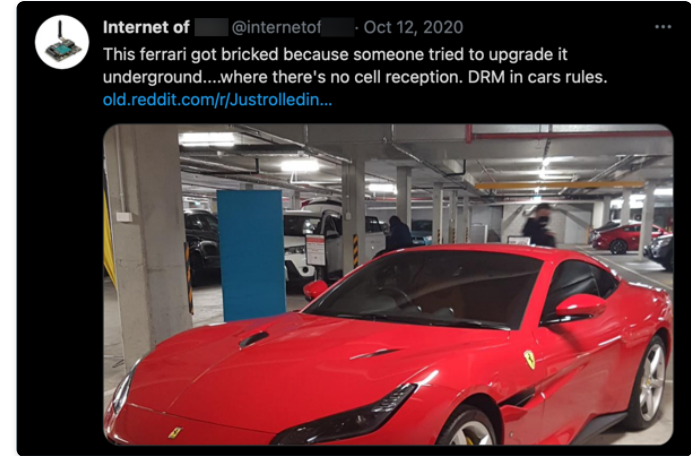
Frequency & Connectivity

📄 How often to send?

Too often → wastes power, bandwidth, and money. **Too rarely** → you miss events. Match the rate to the job.

⚠️ Caution

Connections drop. Smart devices should still work **offline**: a thermostat keeps deciding locally.



No signal, no cloud.

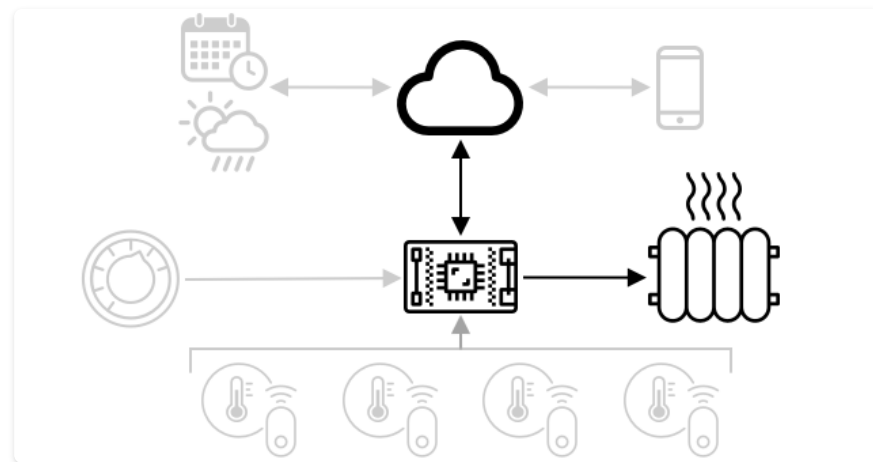
Commands

Controlling devices from the  cloud

Cloud → Device

A **command** is a message from the cloud telling a device to act: drive an actuator, reboot, or send extra data.

- Server checks telemetry → publishes to `/<id>/commands`
- Device subscribes and acts
- e.g. `light < 300` → `{ "led_on": true }` → LED on



When the Device is Offline

What if a command can't be delivered? **It depends on the command.**

① Latest wins → discard

A thermostat **setpoint**: a newer command overrides the old one, so drop the stale ones.

① Order matters → queue

A **robot arm** ("up", then "grab"): commands must be **queued and replayed** in sequence.

Meme check

The "bricked car", trapped underground with no signal: always design for lost connectivity!