IoT COMMUNICATION PROTOCOLS

Dr.L.S.Jayashree

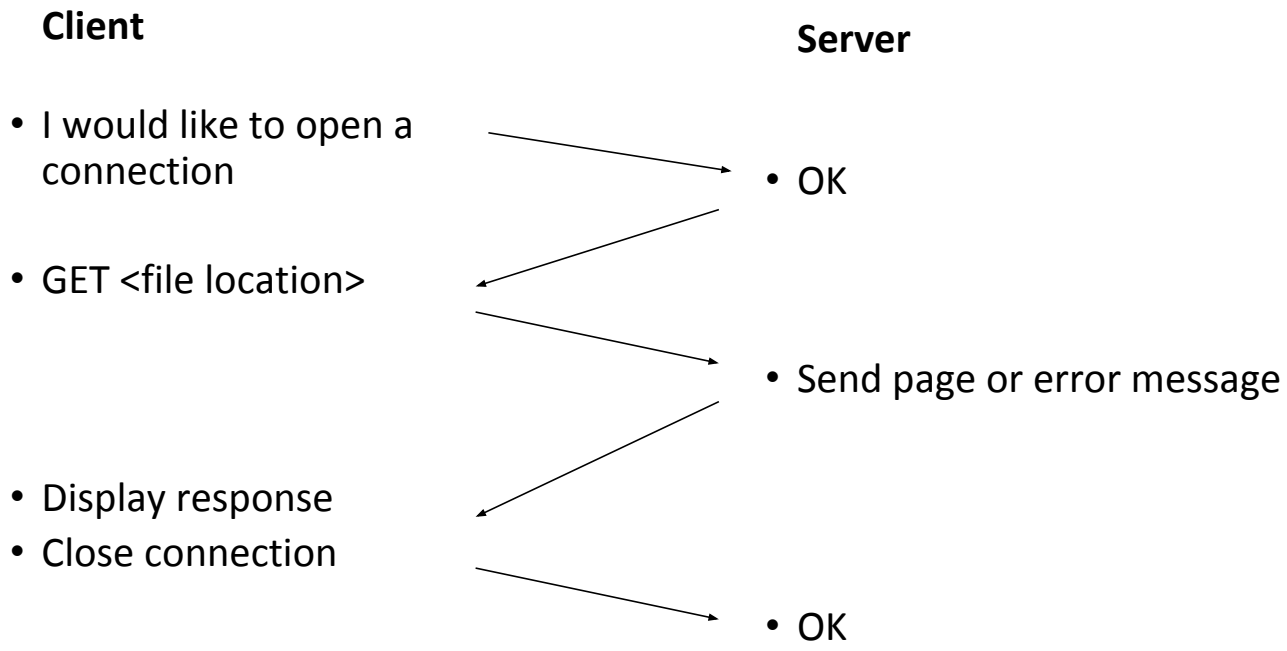
Prof/CSE

PSG CT

HTTP

- synchronous protocol, this means the client waits for the server to send the data.
- a one way street, only the client makes the request.
- one to one protocol, one client request at a time.
- data heavy, filled with headers and rules.

An HTTP conversation



HTTP is the set of rules governing the format and content of the conversation between a Web client and server

GET

The GET method is used to retrieve information from the given server using a given URI.

POST

A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

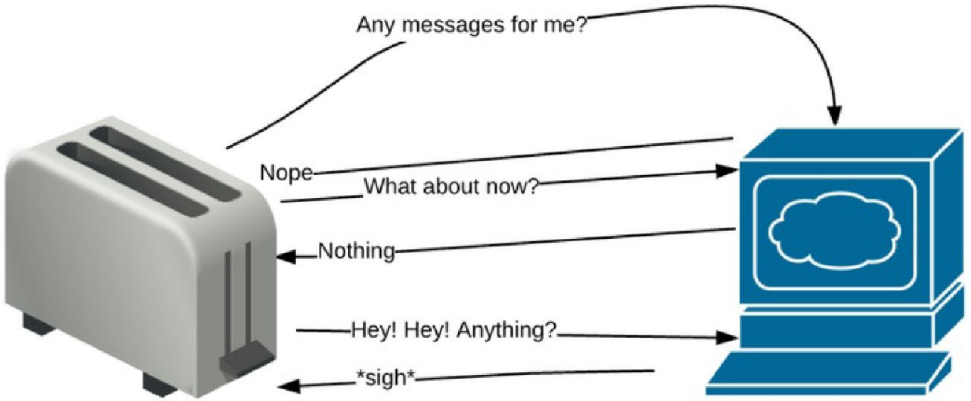
PUT

Replaces all current representations of the target resource with the uploaded content.

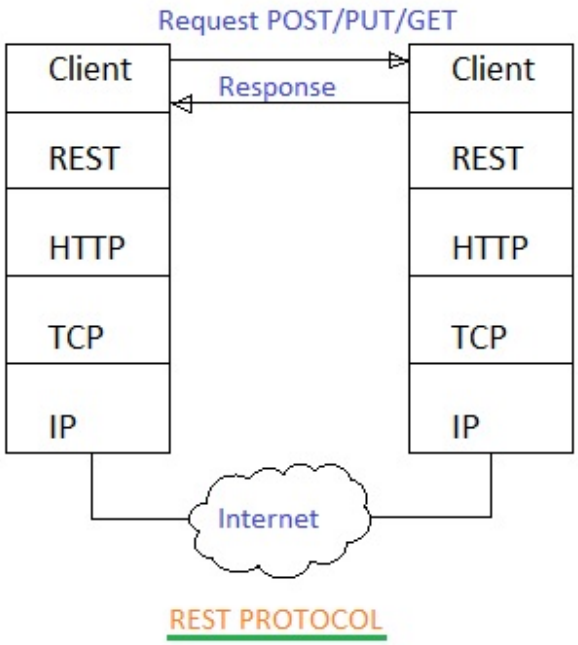
DELETE

Removes all current representations of the target resource given by a URI.

A Typical HTTP Transaction



RESTful Model
Representational State Transfer



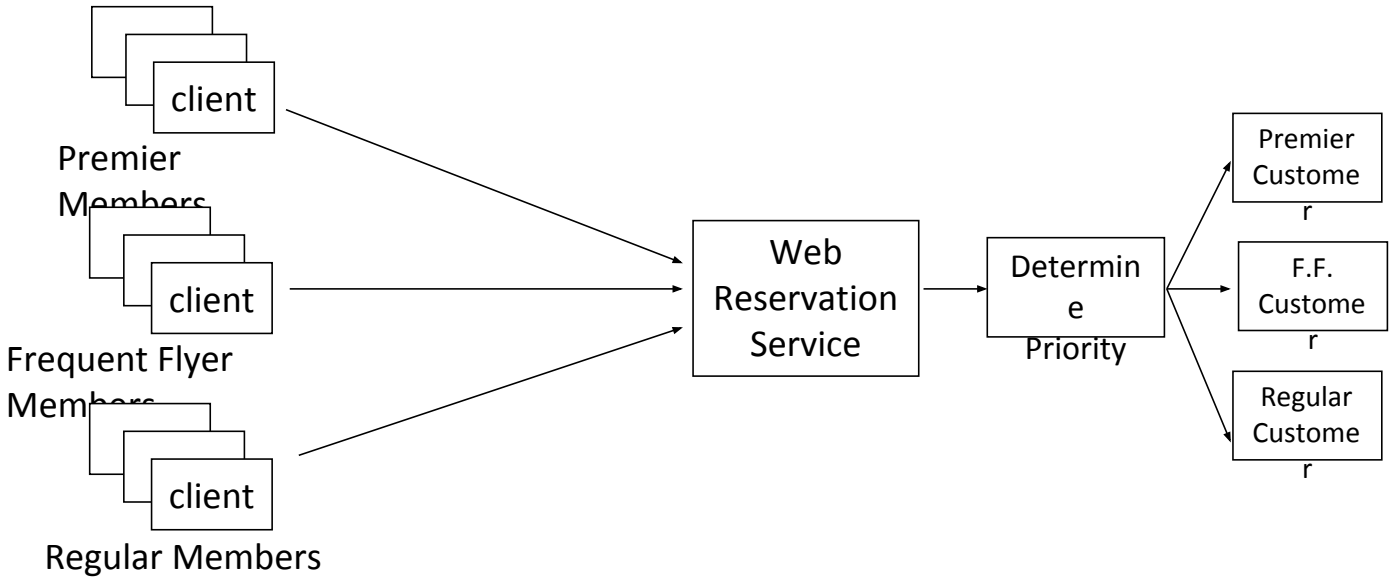
REST PROTOCOL

REST

- Request / Response model – every time you need to do something
  - e.g turn the light bulb on using PUT request
- Tons of requests, lots of data, battery drain, slow response etc
- All things should be exposing their services through a RESTful API
- If things offer RESTful APIs over HTTP, they get a URL and become seamlessly integrated to the World Wide Web

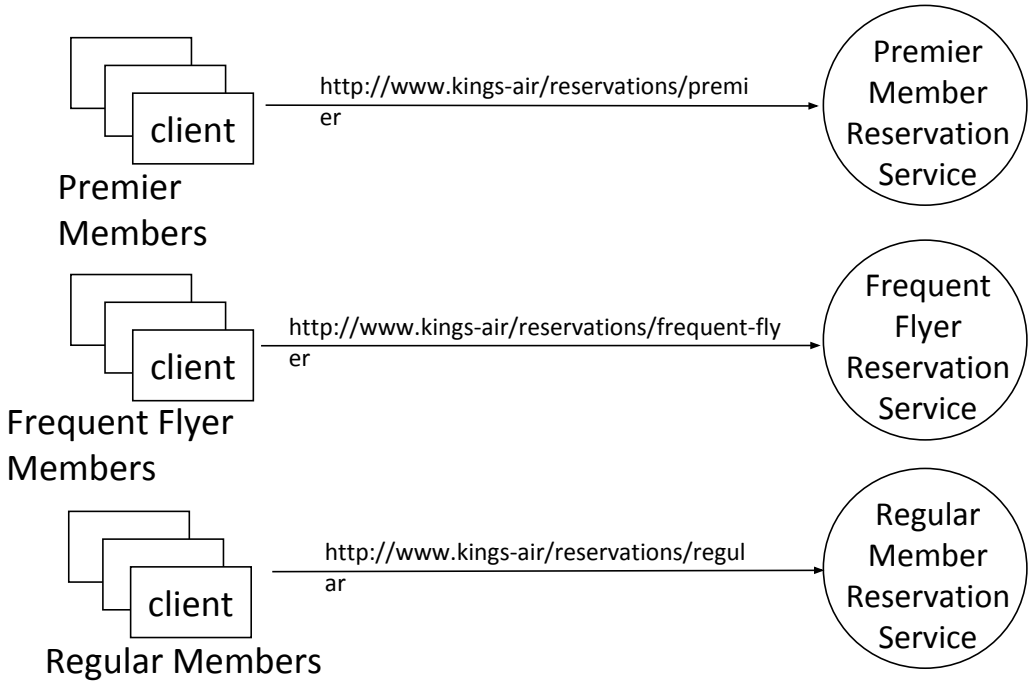
Airline Reservation Example

The airline provides a single URL. The Web service is responsible for examining incoming client requests to determine their priority and process them accordingly.



URLs are Cheap! Use Them! – The REST Design Pattern

The airline provides several URLs - one URL for premier members, a different URL for frequent flyers, and still another for regular customers.



MQTT

What is MQTT ?

- MQTT = MQ Telemetry Transport
- request-response nature of HTTP does not match the event-driven nature of applications- MQTT
- Publish / Subscribe method

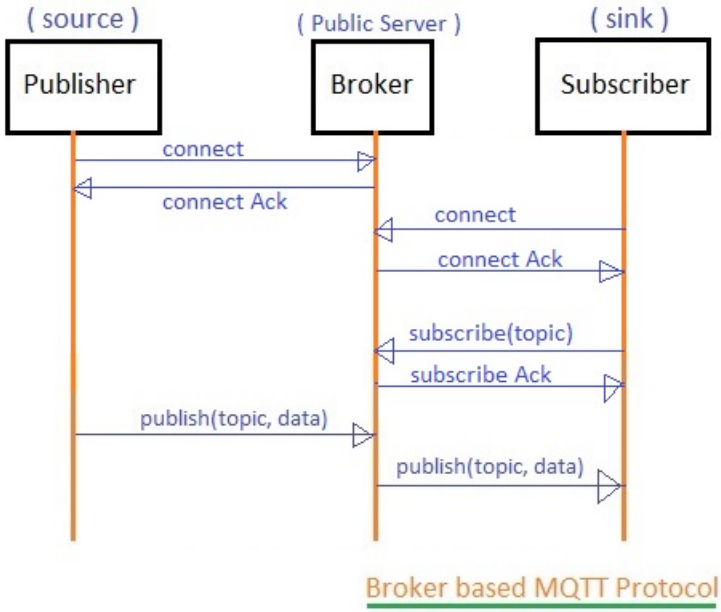
What is MQTT ?

- extremely simple and lightweight messaging protocol
- designed for constrained devices and low-bandwidth, high-latency or unreliable networks.
- The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery

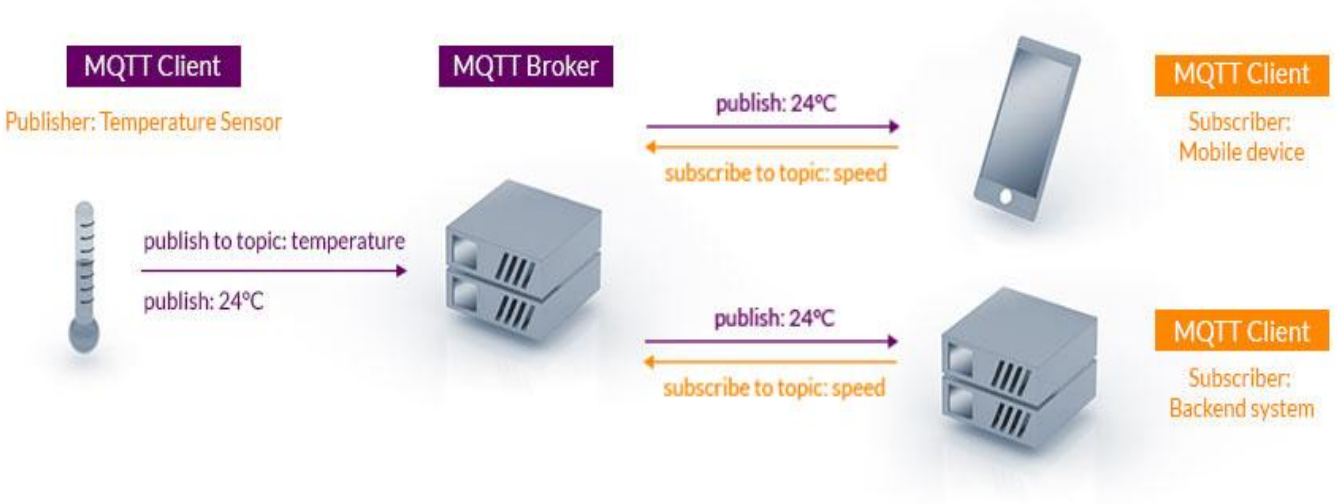
What is MQTT ?

- Lightweight messaging protocol designed for sensors and devices with
  - Flaky network connectivity
  - Low computing power
  - Connections where bandwidth is at a premium
- Protocol specification is open source
- MQTT is nearly 10 years old
  - Mature and evolving

MQTT : MQ Telemetry Transport



MQTT Publish / Subscribe Architecture



Brokers & Clients

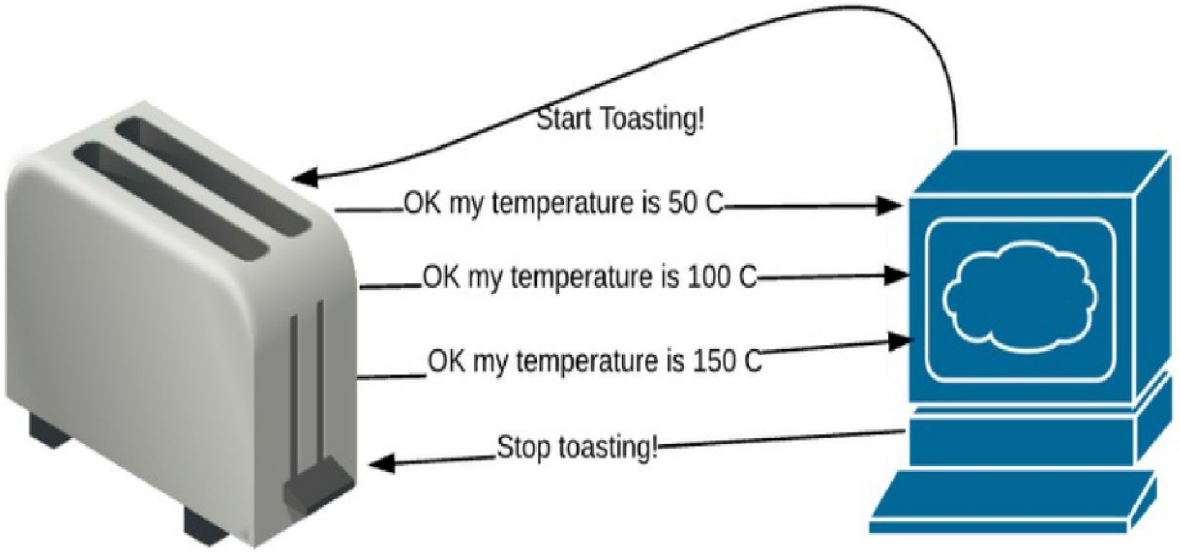
- There’s only one broker, but there can be multiple clients.
- The broker is considered the ‘reliable’ one in the relationship - even if the connection goes out, the broker is expected to be available
- The broker gets all the published messages from the clients and stores them in a database or memory
- The broker then delivers the messages to the subscriber clients.

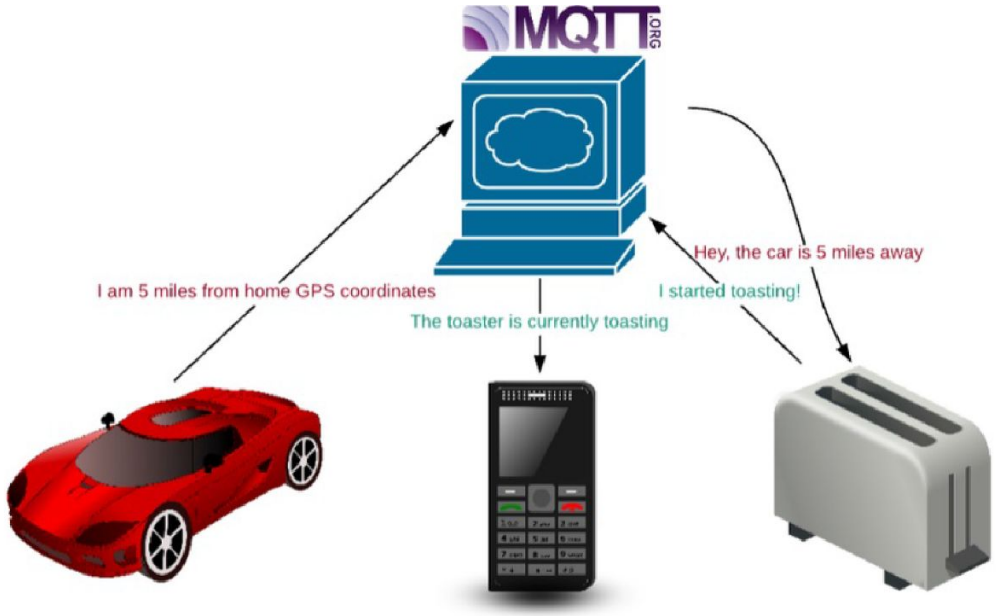
- MQTT is called a lightweight protocol because all its messages have a small code footprint

- Each message consists of a fixed header -- 2 bytes -- an optional variable header, a message payload that is limited to 256 megabytes (MB) of information and a quality of service (QoS) levels

Features of MQTT

- Publish and subscribe to topics
- 3 qualities of service
  - 0 Best effort to deliver a message (Fire and Forget)
    - e.g GPS vehicle tracker
  - 1 Deliver at least once
    - e.g controlling AC
  - 2 Deliver exactly once
    - e.g robotic tractor
- Minimal transport overhead to reduce network traffic
  - As little as 2 bytes


Start Toasting!
OK my temperature is 50 C
OK my temperature is 100 C
OK my temperature is 150 C
Stop toasting!


MQTT.ORG
I am 5 miles from home GPS coordinates
The toaster is currently toasting
Hey, the car is 5 miles away
I started toasting!

MQTT Topics

The third floor main AC would be
sensors/fl3/temperature/AC

The tenth floor boiler would be
sensors/fl10/temperature/boiler

The fourth floor heater would be
sensors/fl4/temperature/heater

Features	MQTT	HTTP
Full Form	Message Queue Telemetry Transport	Hyper Text Transfer Protocol
Design Methodology	The protocol is data centric.	The protocol is document centric.
Architecture	It has publish/subscribe architecture. Here devices can publish any topics and can also subscribe for any topics for any updates.	It has request/response architecture.
Complexity	simple	more complex
Data security	YES	NO, hence HTTPS is used to provide data security.
Upper layer protocol	It runs over UDP.	It runs over TCP.
message size	small, it is binary with 2Byte header.	Large, it is in ASCII format.
Service levels	3	1
Libraries	30KB C, 100KB Java	Large
Port number	1883	80 or 8080
Data distribution	1 to 0/1/N	one to one only

Summary

- you'll likely be using REST whenever you have a transport that has high bandwidth and speed - like Ethernet, when your transport is Internet-connected and talking to online services.

- MQTT is most often used when your transport has battery, bandwidth or packet size restrictions, like WiFi or Cellular.

- CoAP is there when you have very-low bandwidth stateless connections like LoRa, ZigBee or Bluetooth.