IoT Data Exchange Protocols and associated software development

How does IoT devices communicate

Module Contents

- IoT Protocol Stack
- Communication Models & Patterns
- OS for IoT devices – Contiki -os
- RESTful Architecture for IoT
- IoT Protocols - HTTP, CoAP & MQTT

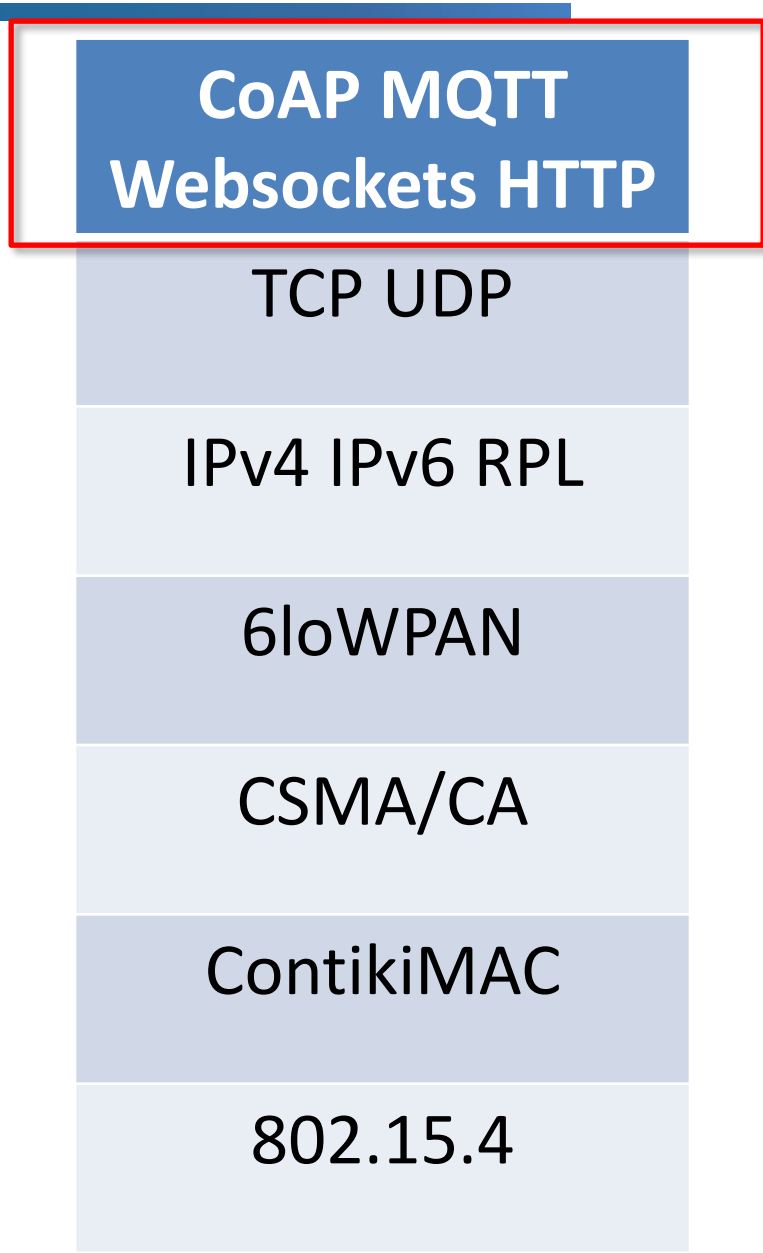
IoT Protocol stack for constrained devices

Typical IoT stack for IEEE 802.15.4 Motes

Application	CoAP MQTT Websockets HTTP
Transport	TCP UDP
Network, Routing	IPv4 IPv6 RPL
Adaptation	6loWPAN
MAC	CSMA/CA
Duty Cycle	ContikiMAC
Radio	802.15.4

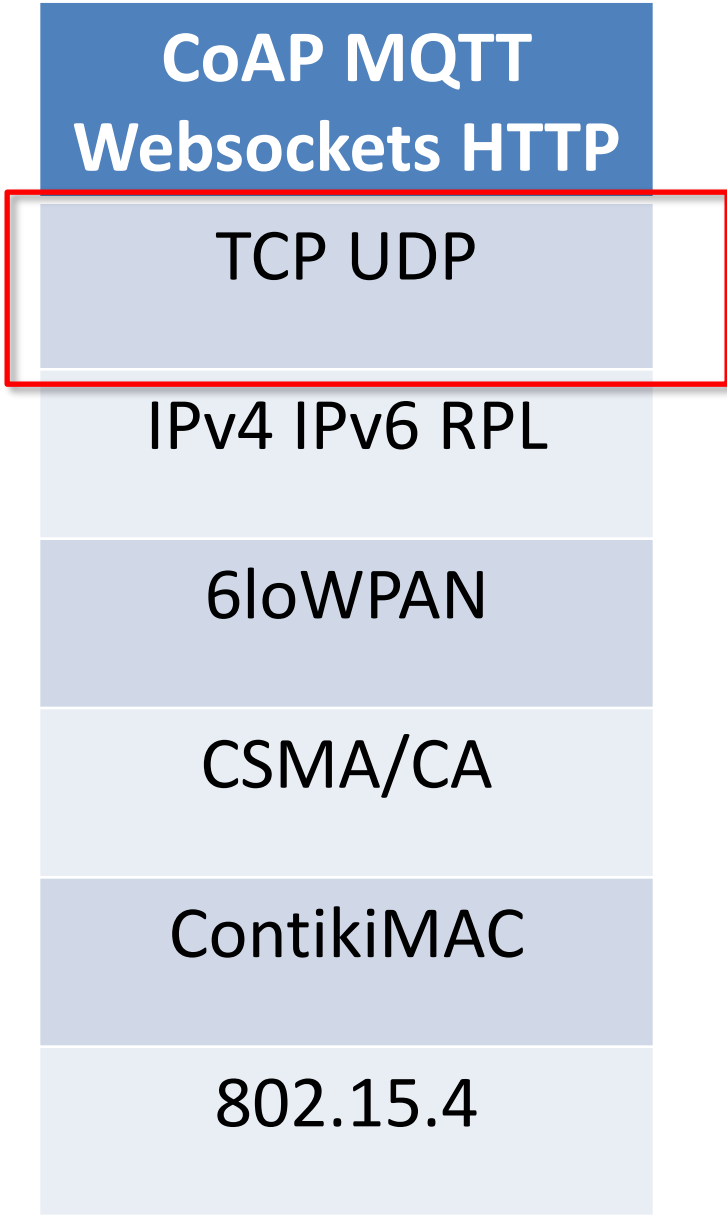
Application Layer

- Responsible for data formatting
- Ensures that data is sent/received in application optimal schemes
- Applications are built on the top of application layer protocols
- IoT application layer headers are lightweight



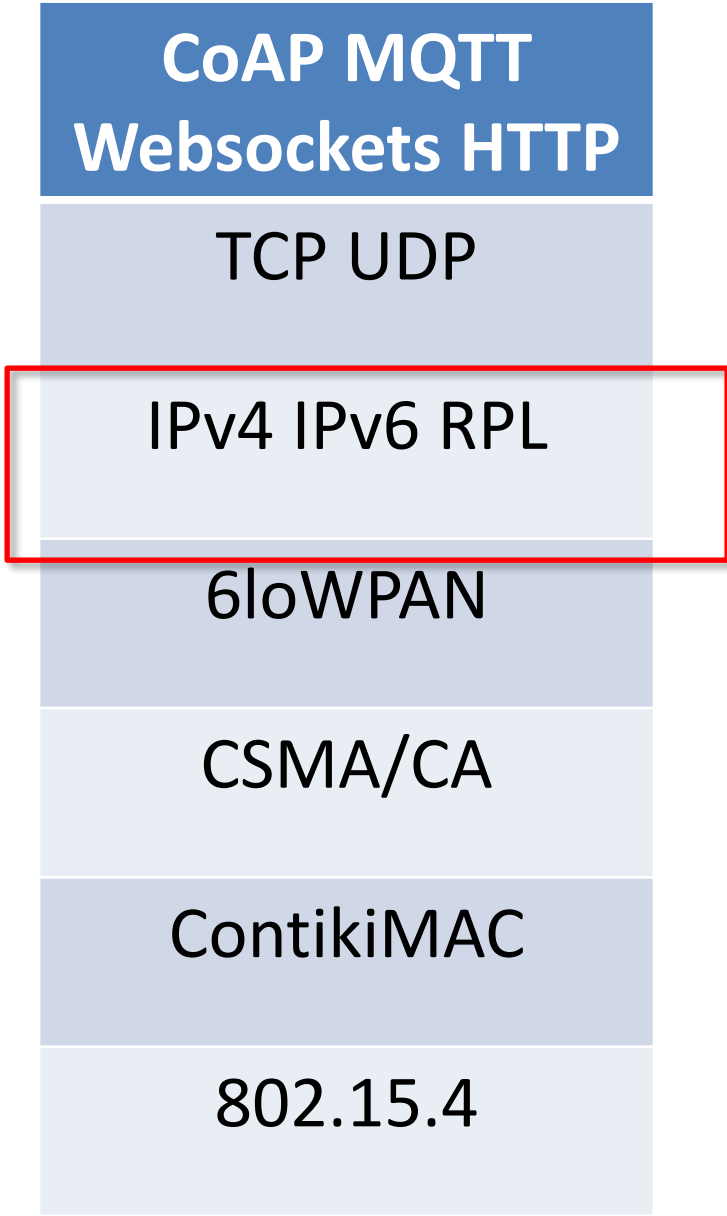
Transport Layer

- Allows multiple applications to run on devices by having dedicated channel numbers
- Maintains sessions between applications on the devices
- In-order delivery of packets
- Optional secured communication using TLS (transport layer security) on TCP and DTLS (Datagram transport layer security) on UDP



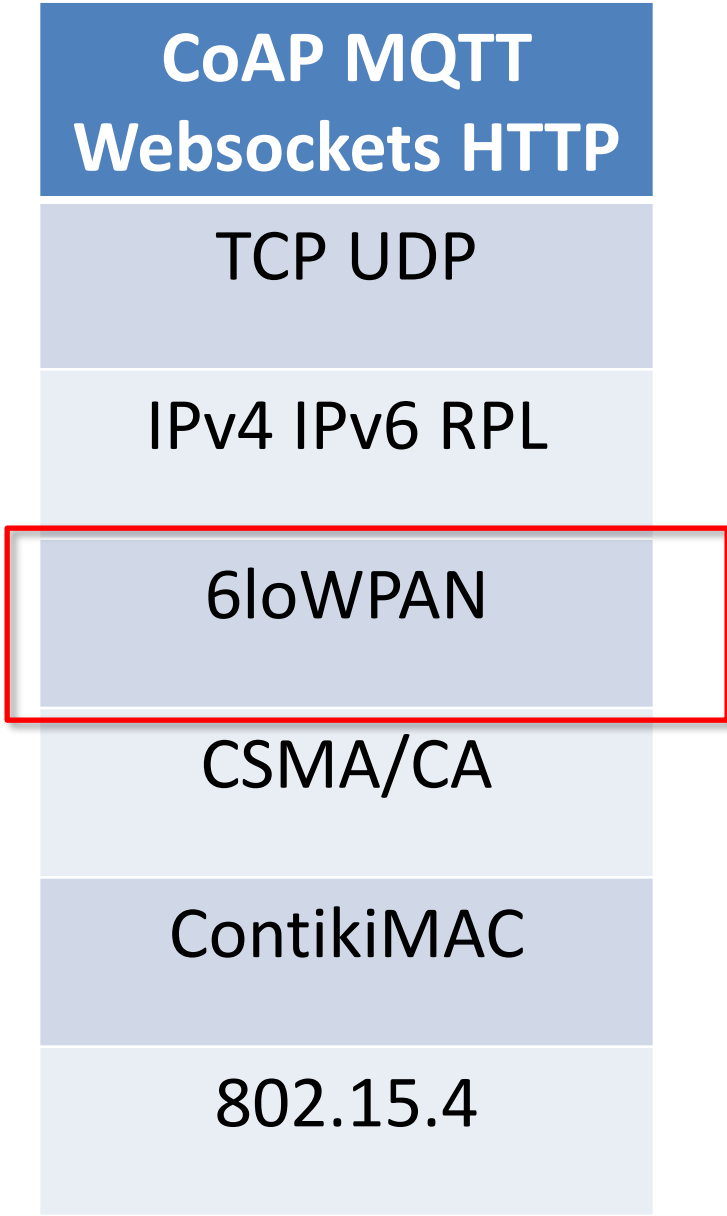
Network Layer

- Device addressing – IPv6
- Routing of data through multiple hops if required
- Routing algorithms optimized for power consumption in IoT



Adaptation Layer

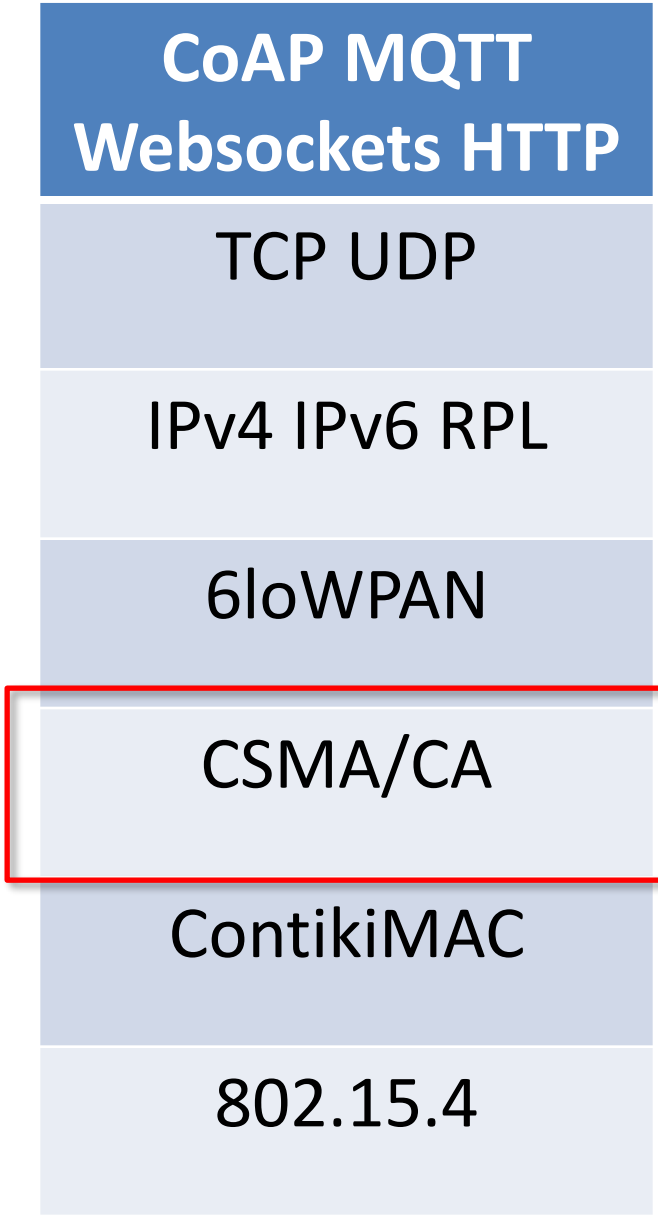
- Transmits IPv6 datagrams over IEEE 802.15.4 Link layer frames and vice-versa
- Header compression for IPv6 and UDP
- Data Fragmentation and reassembly



MAC Layer

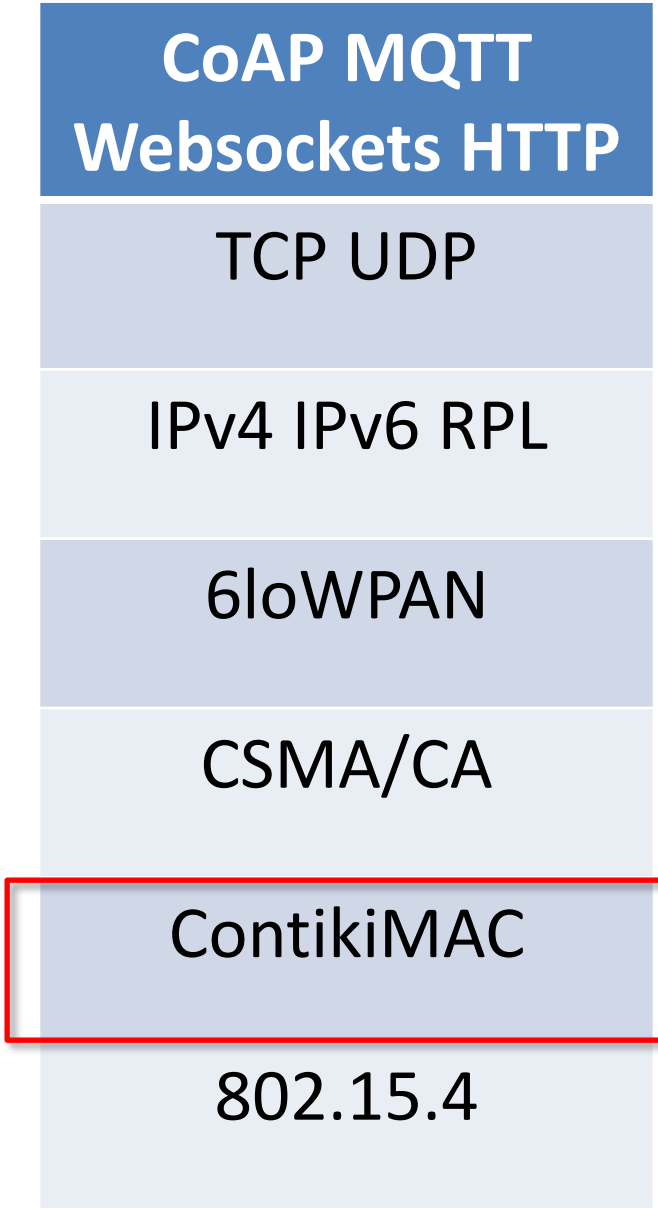
- Reliable link between two devices by detecting and correcting errors that may happen in physical layer

- Protocol used : CSMA/CA



Radio Duty Cycle layer

- Used for reducing power consumption of radio transreceiver
- Algorithm decides when the radio is On/Off – Contiki MAC
- Need to optimize for Radio being On when receiving data from others

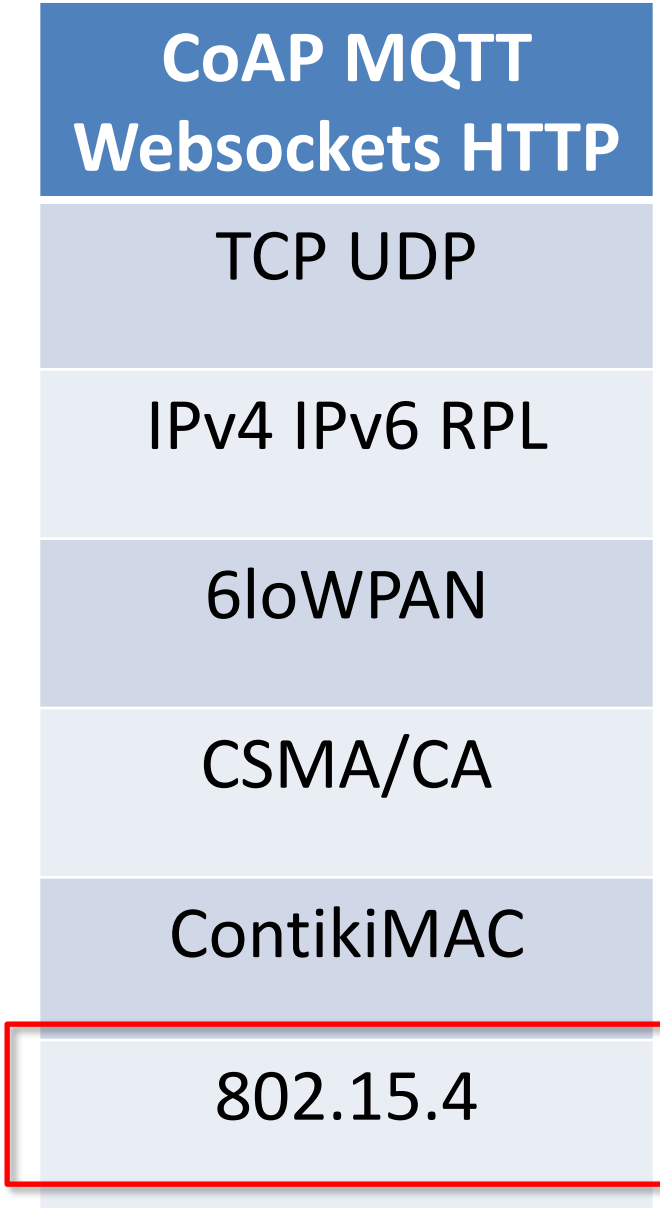


Physical layer

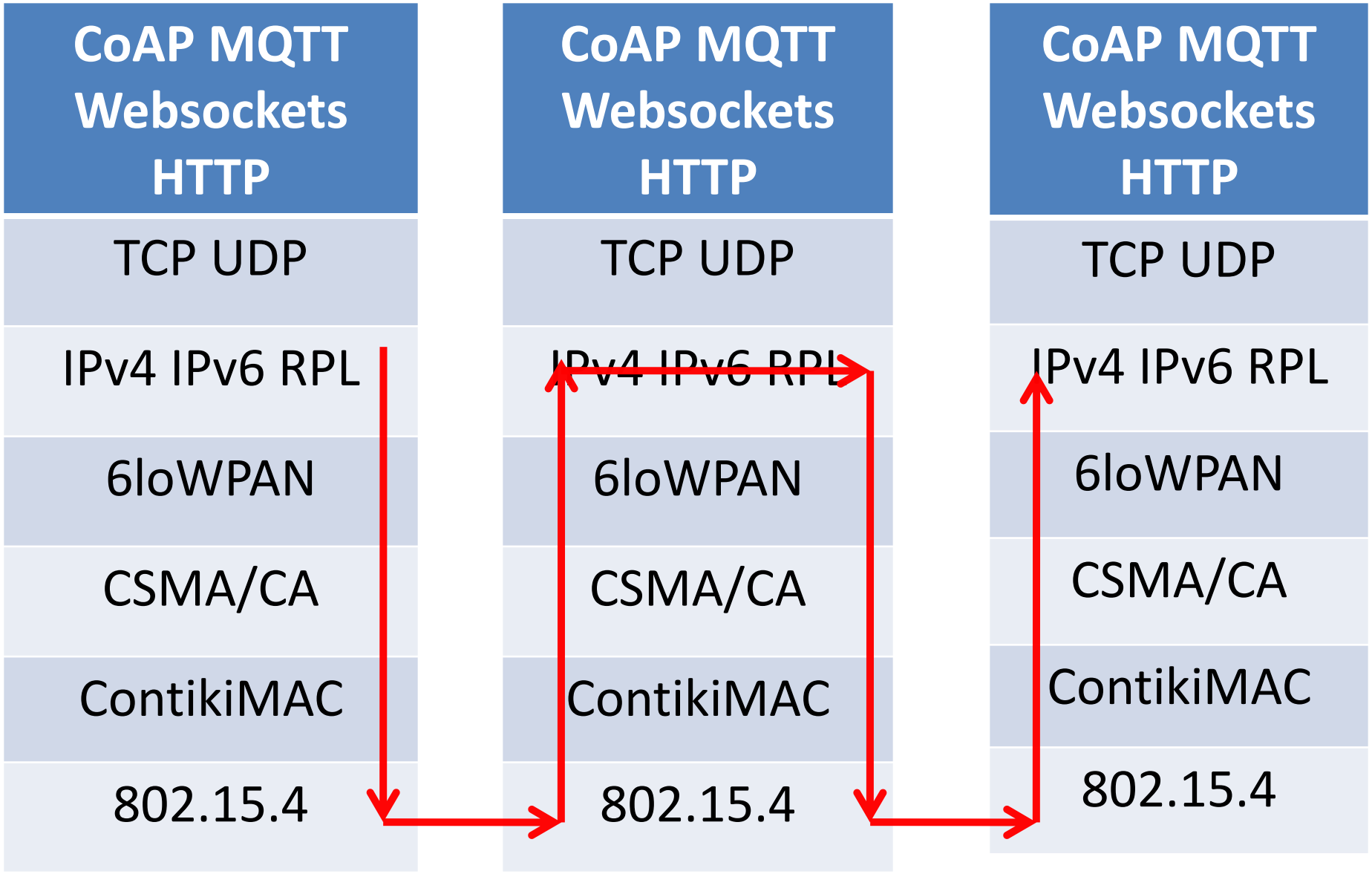
- Radio layer transmission and reception of signals in the frequency band of 2.4 GHz and Sub 1 – GHz.

- IEEE 802.15.4 – 2006 , IEEE 802.15.4e Standards

- Bit timings, synchronization, Modulation – demodulations and encodings



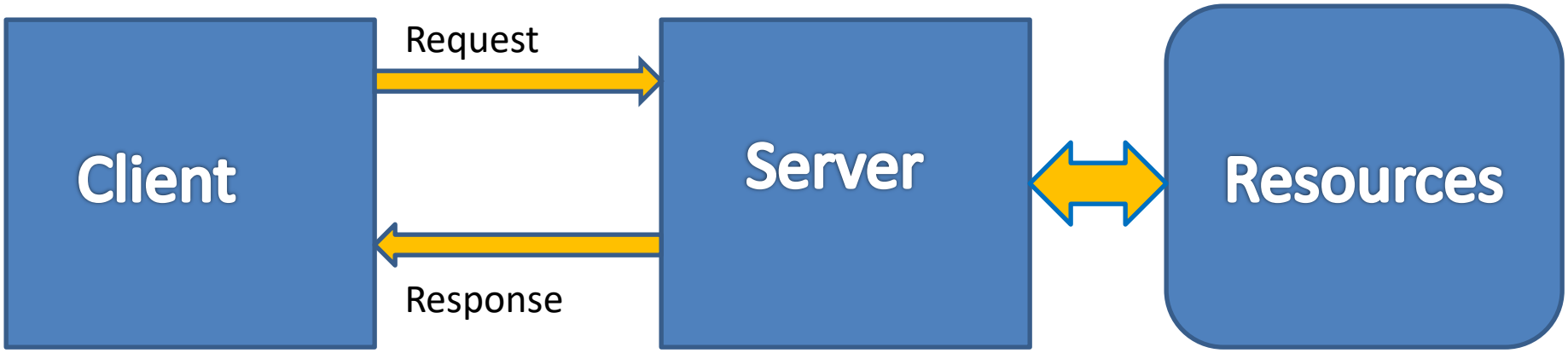
Routing Path



Communication Models in IoT

- Request – Response
- Publish – Subscribe
- Push – Pull
- Exclusive Pair

Request – Response Model

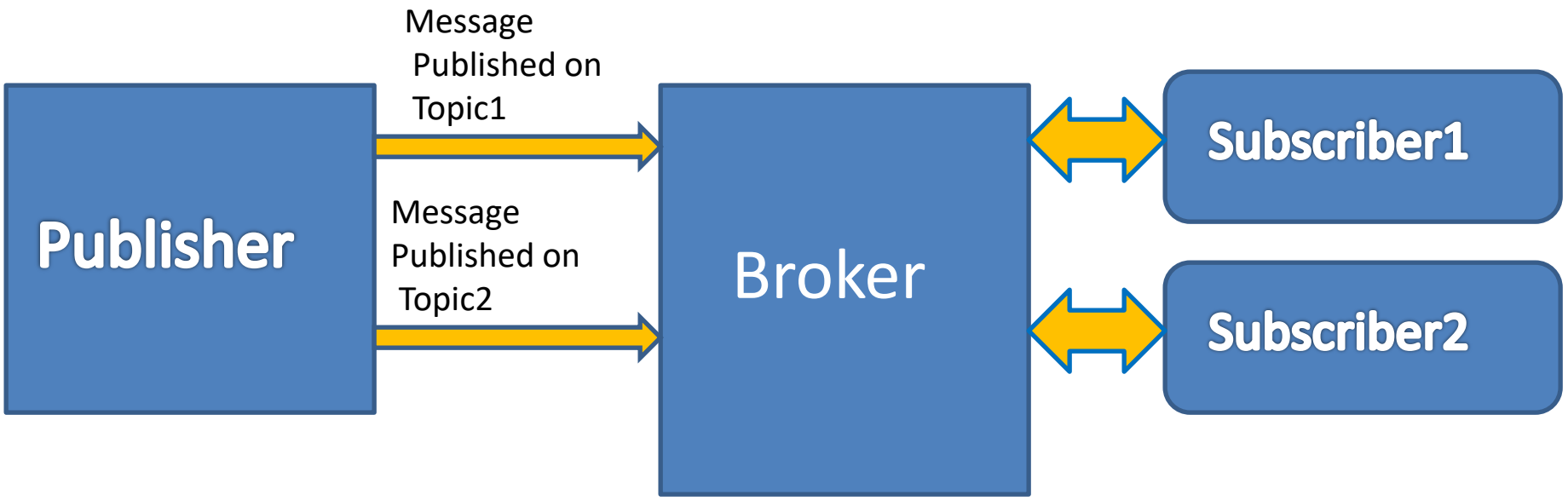


- First Client makes a request to server
- Server then responds to clients request
- Server maintains the resources
- Based on the request from Client, Server may retrieve resource representations and sends the data client in the appropriate format

Examples of Protocols based on Request – Response

- HTTP (Hyper text transfer protocol)
- CoAP (Constrained Application Protocol)

Publish – Subscribe Model

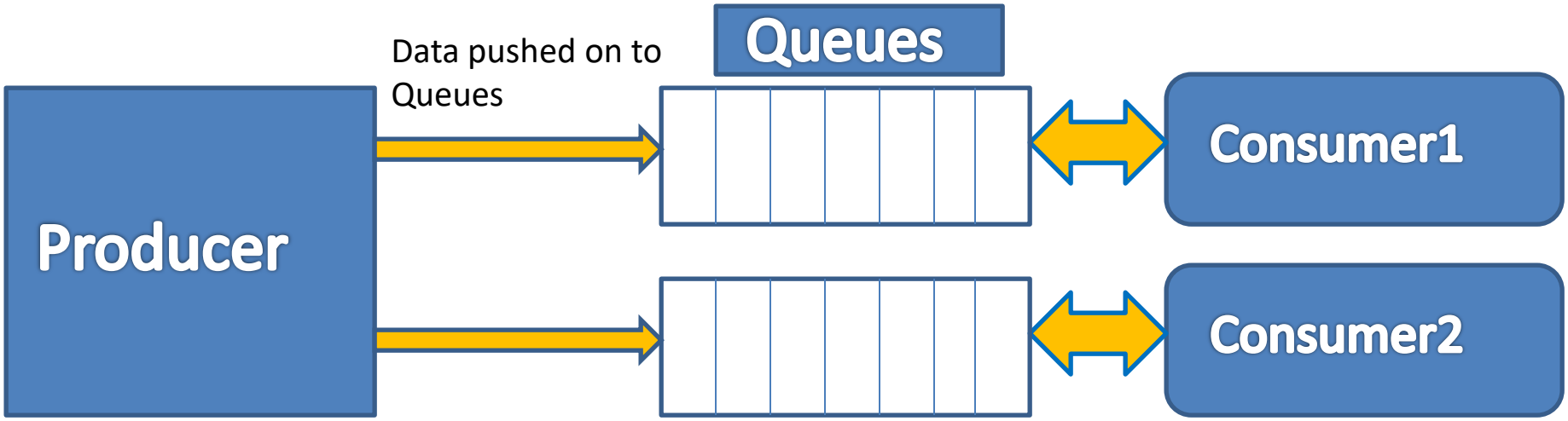


- Consists of Publisher, Subscriber and Broker
- Publishers publishes data onto broker on a given topic
- Subscriber need to subscribe onto particular topic on Broker
- Whenever data arrives on the Broker on a topic, Broker in turn notifies all the subscribers who have subscribed on that topic
- Decouples Publishers & Subscribers

Examples of Protocols based on Publish – Subscribe

- MQTT (Message queue telemetry transport)
- DDS (Data distribution service)

Push – Pull Model

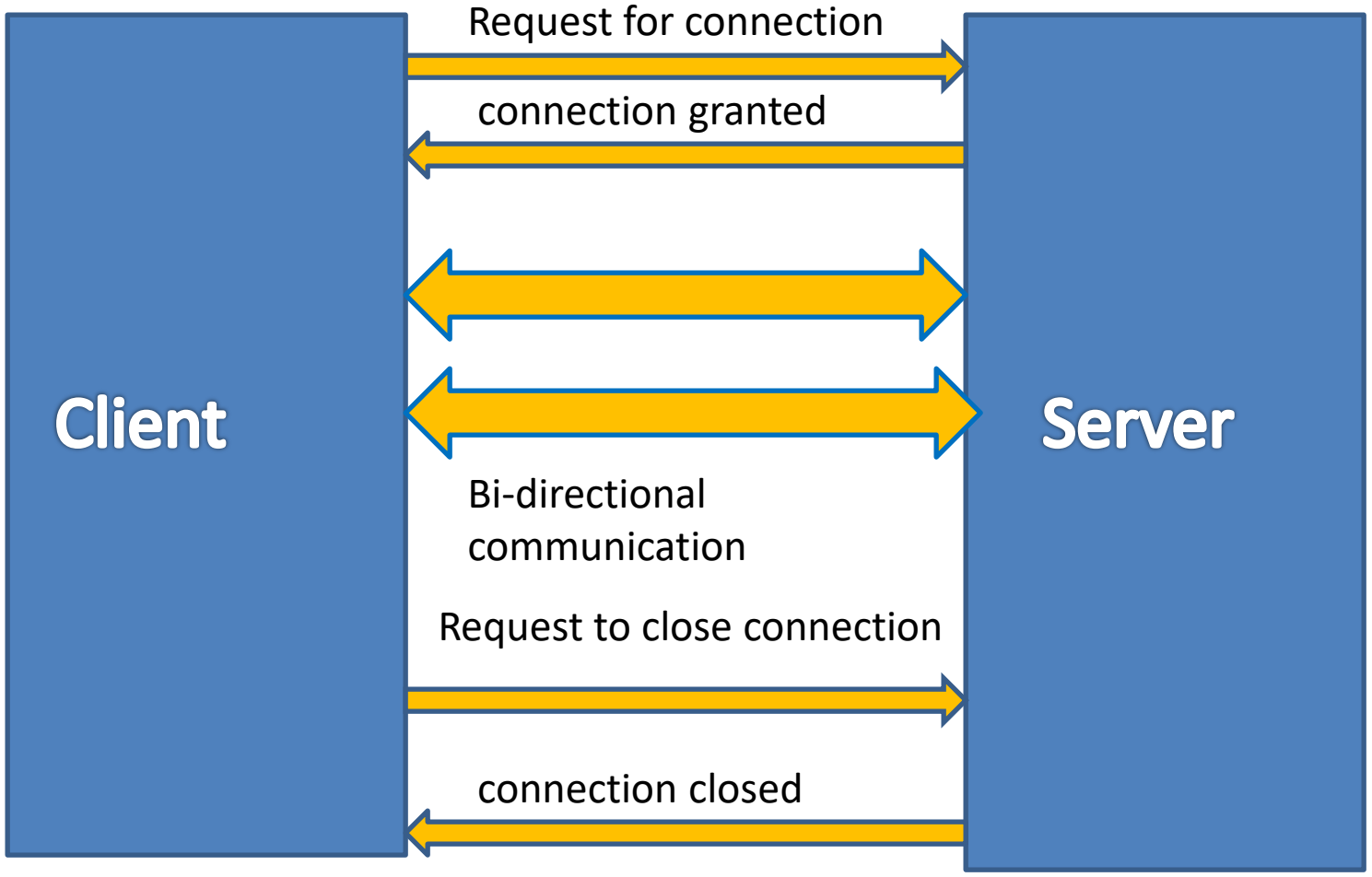


- Producers push data onto queues
- Consumers pull data from queues
- Queue kind of data structure is maintained
- Decouples Producers and Consumers

Examples of Protocols based on Push – Pull

- AMQP (Advanced message queuing protocol)

Exclusive Pair Model



Exclusive Pair Communication Model

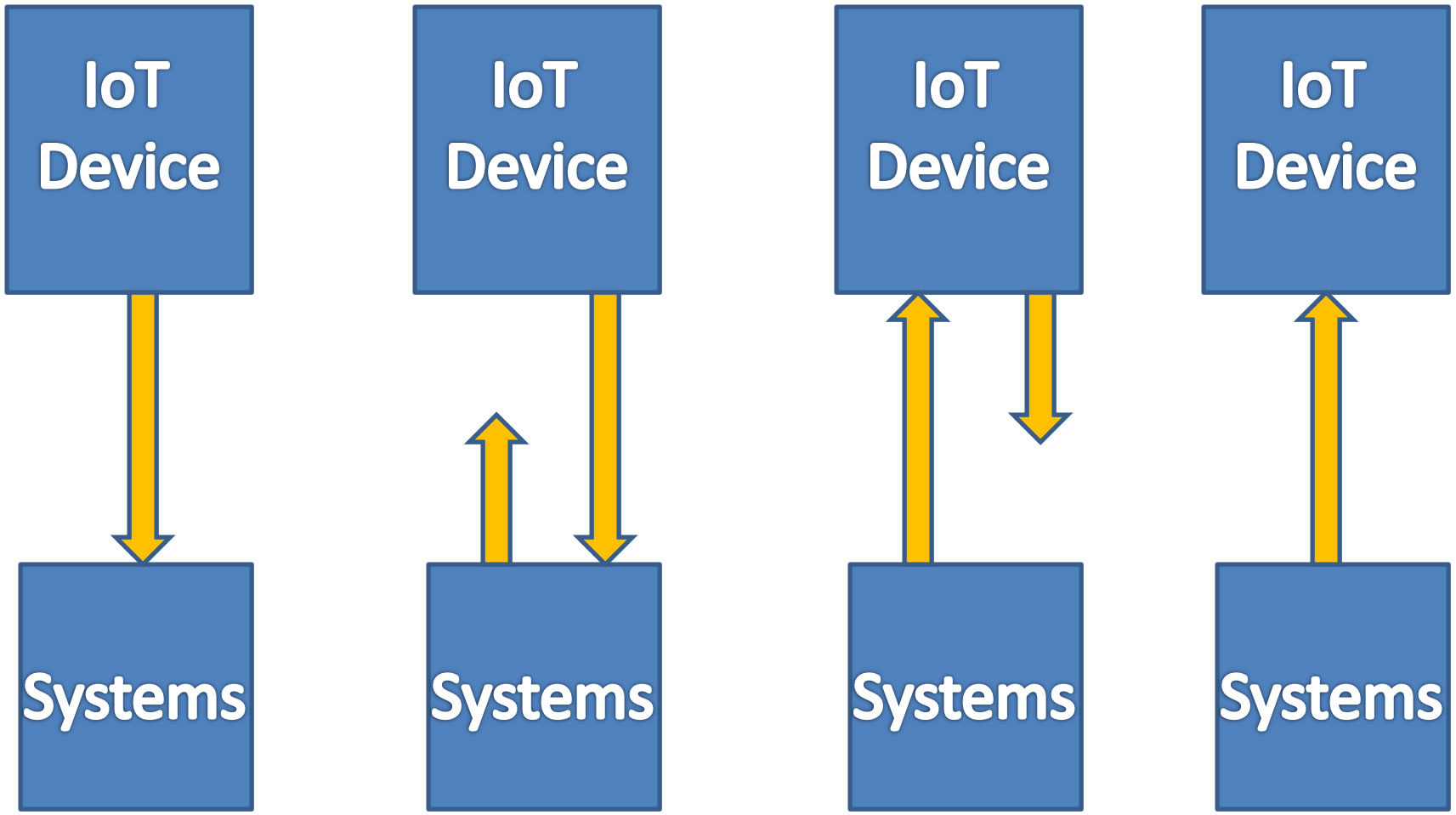
- Bi-directional, full duplex communication after the initial handshake

- Once the connection is established, it remains open and allows full duplex communication between client and server until the connection closes

Example of Protocol based on Exclusive Pair

- Websockets : Websocket protocol allows full-duplex communication over a single socket connection between server and client machines

Communication Patterns in IoT



Telemetry Inquiries Commands Notifications