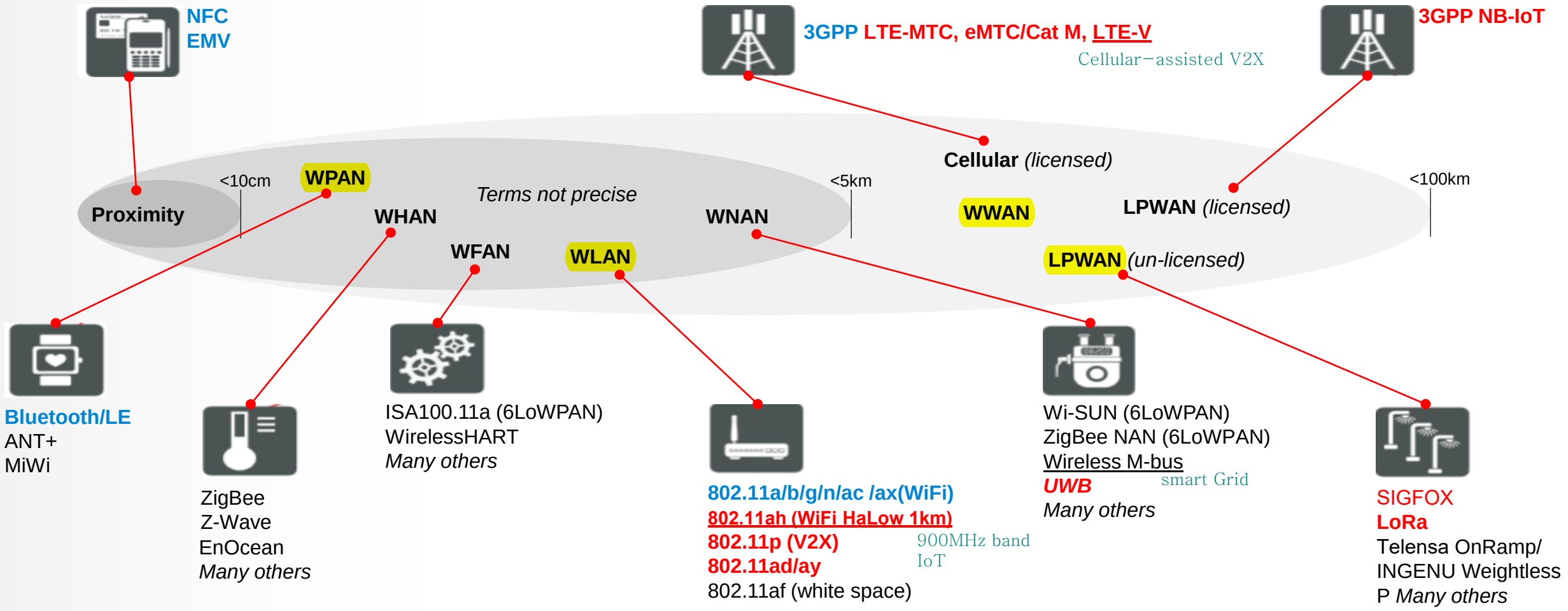
Protocol Stack for IoT

WiCon/IoT Radios

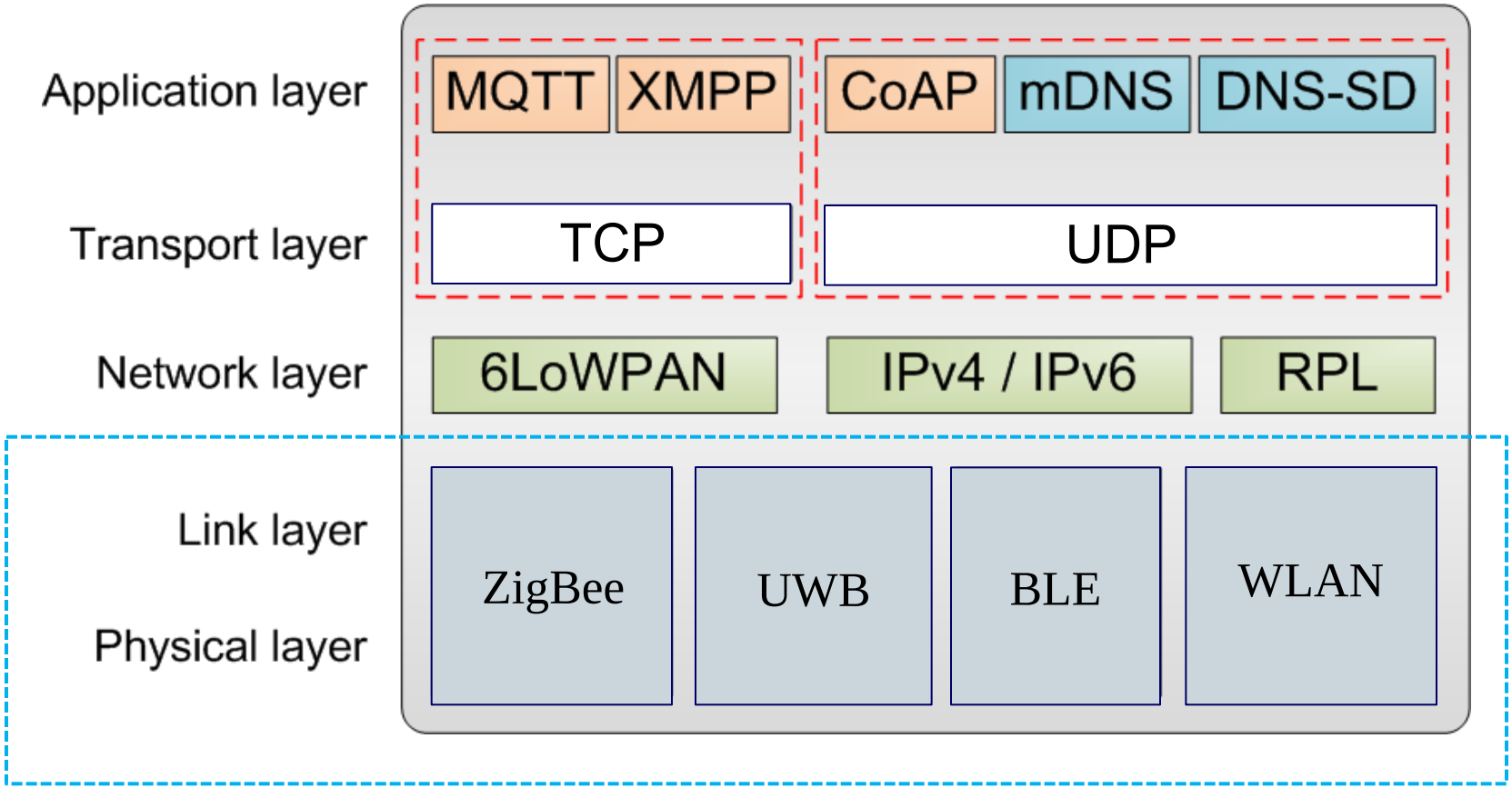
Blue: > billion units/year now
Red: emerging



WPAN: Wireless Personal Area Network
WHAN: Wireless Home Area
WFAN: Wireless Field (or Factory) Area
WLAN: Wireless Local Area
WNAN: Wireless Neighbourhood Area
WWAN: Wireless Wide Area
LPWAN: Low Power Wide Area Network

Trends:
WiFi: Keep evolving. Complementary/competitive for cellular/5G?
IoT: Market still very segmented. Issues for business models are regional. No equivalent business model. Vertical market is key to success.

Protocol stack for Short Range IoT



6LoWPAN

- IPv6 over Low power Wireless Personal Area Networks
  - LoWPAN comprises IEEE 802.15.4 devices or BLE devices which are characterized by short range, low bit rate, low power, and low cost
- Functions defined in 6LoWPAN
  - Header compression
    - To transmit IPv6 packets over an IEEE 802.15.4 or BLE frame
  - Packet fragmentation
    - Upper layer PDUs should be fragmented/reassembled properly in order to be sent/received through LoWPAN
  - Layer 2 forwarding

RPL

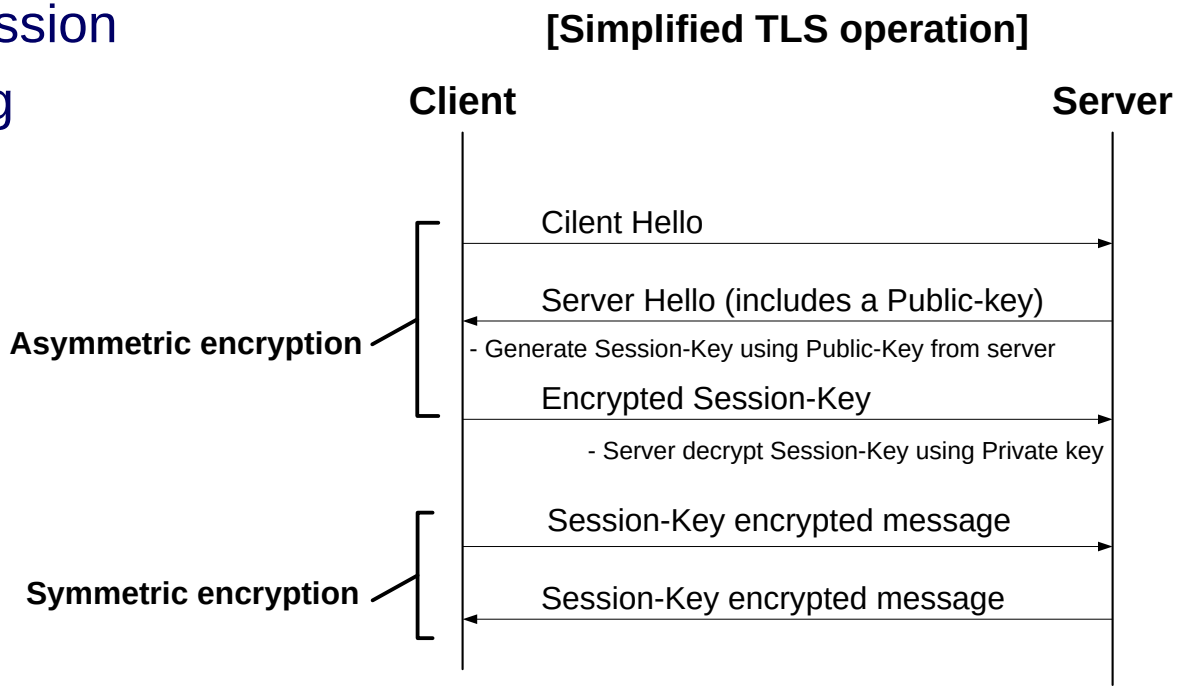
IPv6 Routing Protocol for Low-power and Lossy Networks

- A routing protocol for wireless networks with low power consumption and generally susceptible to packet loss
- Proactive protocol based on distance vector
- operates on IEEE 802.15.4
- Information exchanges between nodes occur based on DAG (directed acyclic graph) computation and ICMPv6
- optimized for a multi-hop mesh network

DTLS (Datagram Transport Layer Security)

❖ TLS provides a secure data communication over TCP

- To support secure communication over UDP (datagram environment) packet loss or reordering must be addressed
- Main functions of DTLS
  - Packet retransmission
  - Packet reordering
  - TLS operation



DNS, DNS-SD (1)

- A protocol for advertisement and discovery of network services and presence information
- based on the Internet protocol suite

- Preliminary: Two types of discovery
  - Advertisement
    - An end device advertises its presence and related information via multicast channel
    - If a control app listens to the multicast channel, it discovers the end device
  - Request search
    - A control app sends search request and end devices response to the request

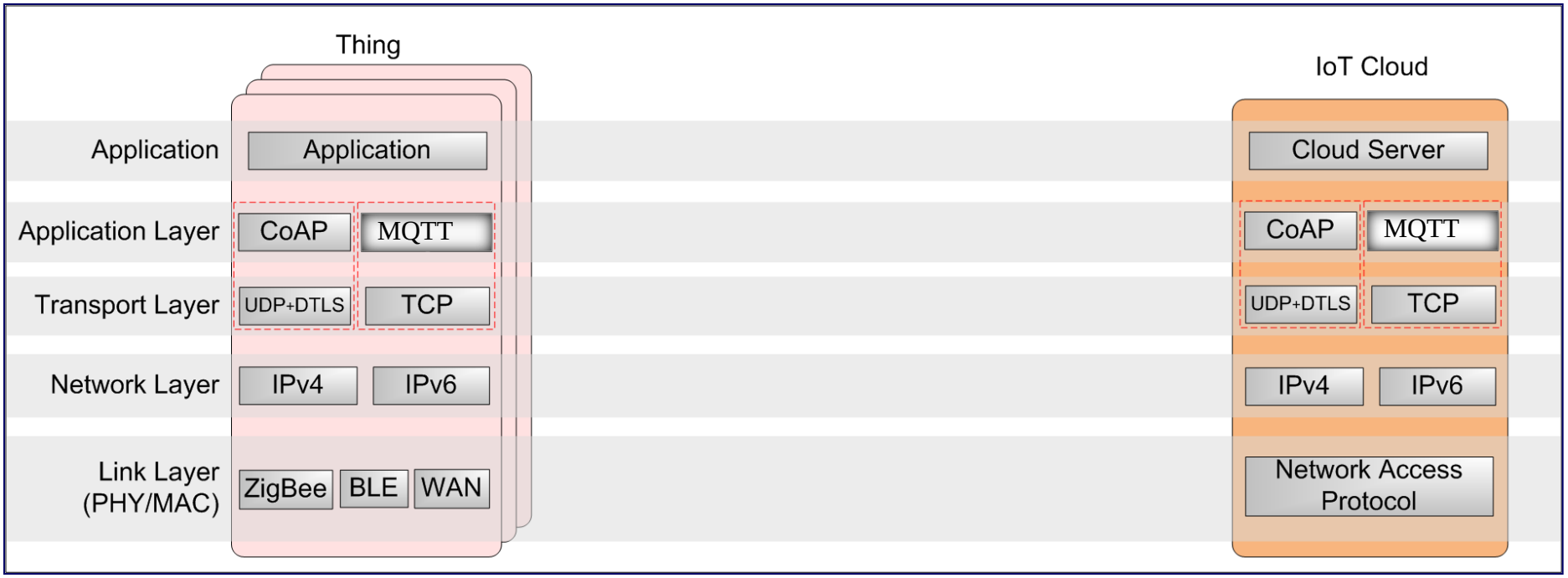
DNS, DNS-SD (2)

- DNS (multicast Domain Name System)
  - It resolves host names to IP addresses within small networks that do not include a local name server
  - It is a zero-configuration service
  - When an mDNS client want to know host name of another client, it transmits a multicast query message
    - This message includes its own host name and IP address
  - Other hosts listen the message and update their mDNS cash

- DNS-SD (Domain Name System – Service Discovery)
  - It can discover another device providing a specific service
    - ❖ e.g., print server or audio device

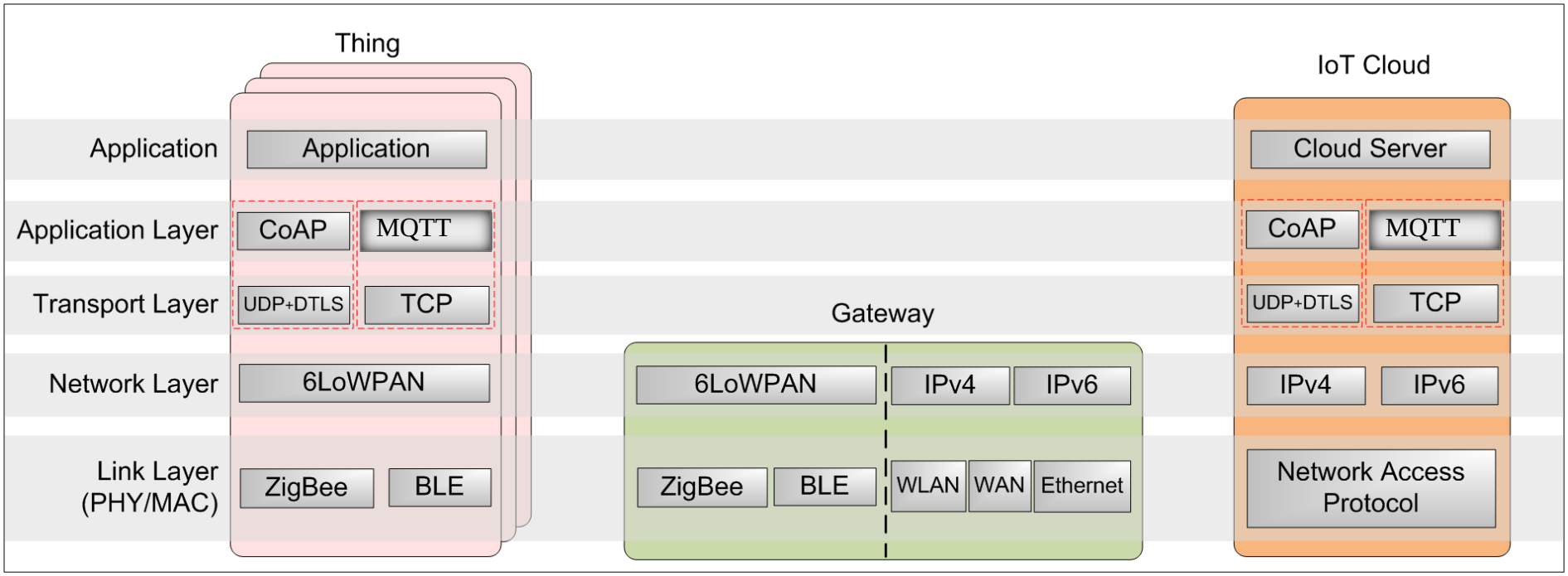
Case 1

- Conventional internet protocols are implemented
- Thing has internet access capability
- M2M gateway is not needed



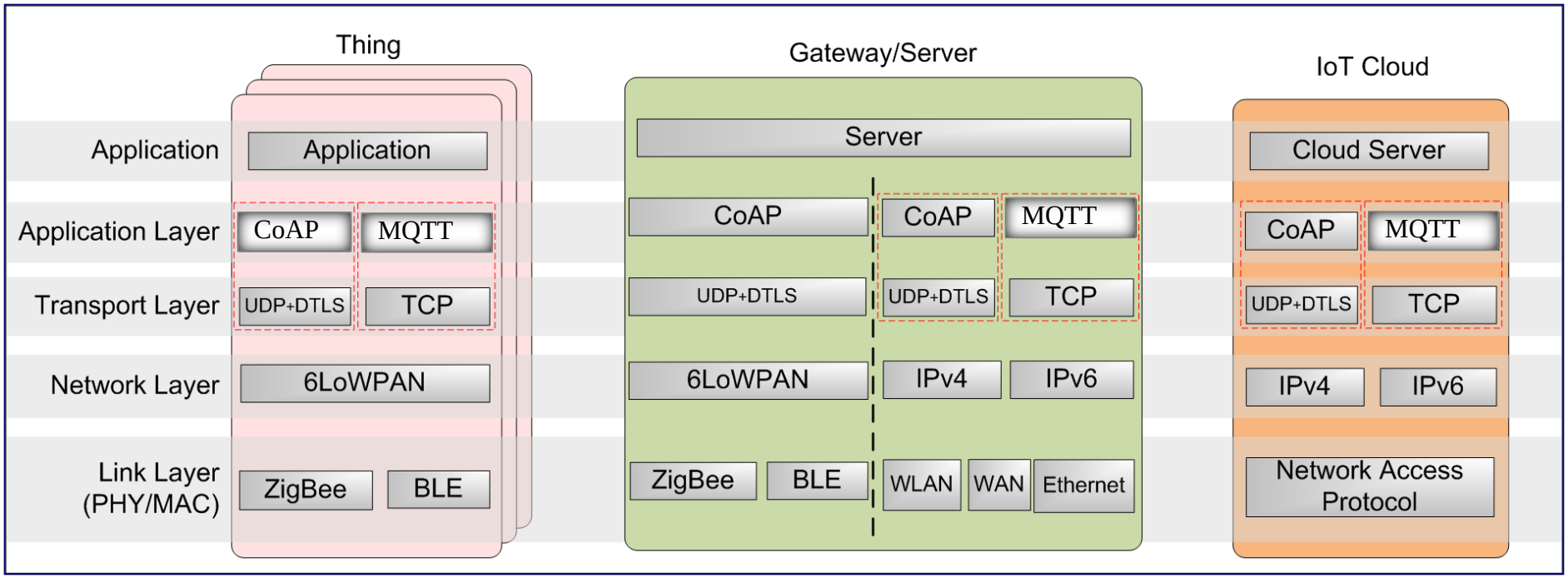
Case 2

- Gateway only forwards network layer packets to IoT cloud



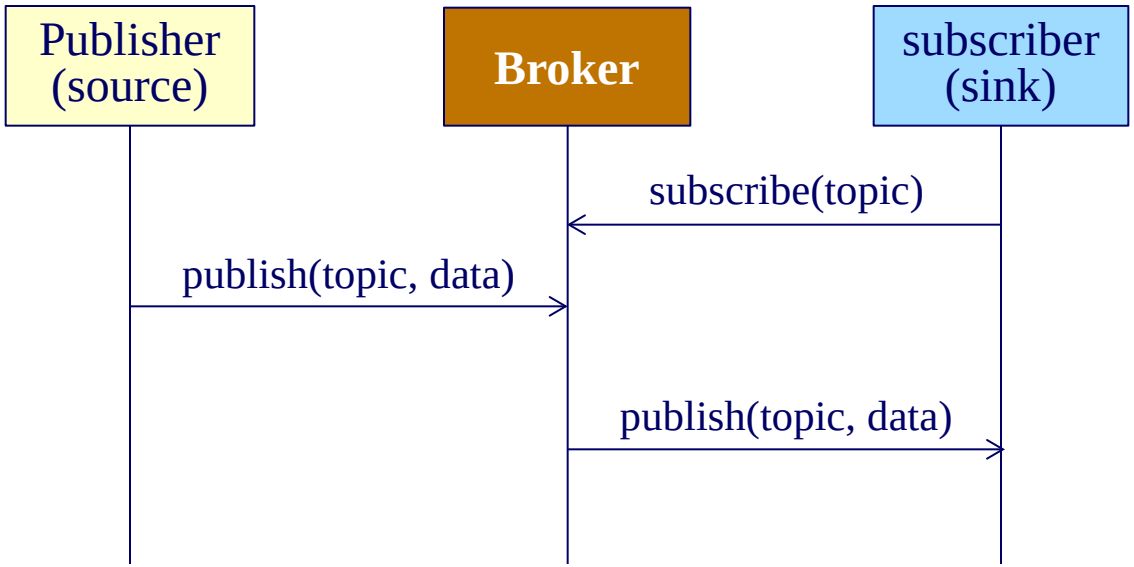
Case 3

- Gateway has the application to communicate with IoT cloud
- Gateway has a role of controlling the network of things



MQTT (Message Queuing Telemetry Transfer)

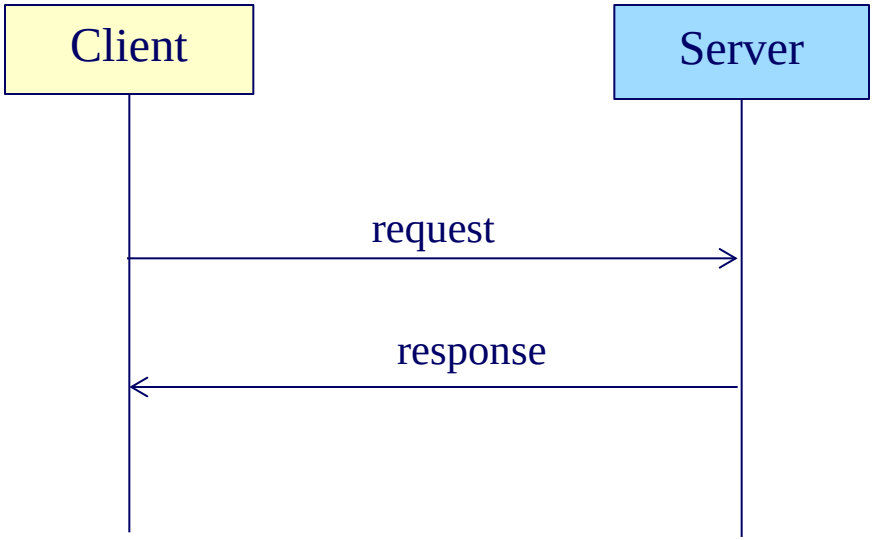
- Lightweight publish/subscribe messaging protocol, designed for M2M connectivity
- Features
  - Data producer publishes a message on a topic and consumer subscribes on the topic
  - Wildcards (+ and #) can be used for simplicity, in descripting a topic
  - Broker (in server) forwards message(s) to corresponding subscribers



CoAP (Constrained Application Protocol)

- A simple application layer protocol that enables web service even for resource-constrained devices (e.g., WSN node)
- Features
  - Translation to HTTP packet can be done easily → Simple integration with web
  - Confirmable: ACK for better reliability over UDP

    Nonconfirmable: No Ack



MQTT vs. CoAP

	MQTT	CoAP
Comm. Model	Publish-Subscribe	Request-Response
Transport Protocol	TCP	UDP
Transport Security	TLS	DTLS
QoS	0: fire and forgot 1: repeat until ack reception 2: exactly once	Confirmable Non-confirmable
RESTful	No	Yes

MQTT vs. CoAP

- Example of messaging

  Assumption: All rooms in building 302 have a thermometer and a hygrometer

Scenario 1:

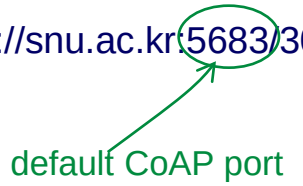
A device want to know the temperature of building 302 room 309

Using MQTT

→ Subscribe this topic: /KOR/Seoul/Gwanak/151-742/302/309/temperature

Using CoAP

→ Request to this URI (using GET method): coap://snu.ac.kr:5683/302/309/temperature



MQTT vs. CoAP

Scenario 2

A device want to know the temperature of all rooms in building 302

Using MQTT

→ Subscribe this topic using wildcard + (single-level wildcard):

/KOR/Seoul/Gwanak/151-742/302/+/temperature

Using CoAP

→ Request to multiple URIs (using GET method):

coap://snu.ac.kr:5683/302/101/temperature

⋮

coap://snu.ac.kr:5683/302/720/temperature

MQTT vs. CoAP

Scenario 3

A device want to know the temperature and humidity of all rooms in building 302

Using MQTT

→ Subscribe this topic using wildcard # (multi-level wildcard):

/KOR/Seoul/Gwanak/151-742/302/#

Using CoAP

→ Request to multiple URIs (using GET method):

coap://snu.ac.kr:5683/302/101/temperature

coap://snu.ac.kr:5683/302/101/humidity

⋮

coap://snu.ac.kr:5683/302/720/temperature

coap://snu.ac.kr:5683/302/720/humidity