IoT Communications - Part 3

Dr Priyanka Bagade, IITK

CS698T, Lecture 6

Content Copyrights

IoT Network Protocol Stack

Application Layer (HTTP, MQTT, CoAP, AMQP)

Transport Layer (TCP, UDP)

Network/Internet Layer (RPL – routing protocol)

Adaptation Layer (6LoWPAN)

Physical Layer (IEEE 802.1.5.4, RFID, NFC)

IoT Network Protocol Stack

The resource constrained IoT devices can’t use the robust and flexible IP stack as it is.

Communication through non-IP protocols:

BLE, RFID, NFC etc

Limited range, only useful in PAN

IoT protocols for long range: LoRaWAN, Sigfox, WiFi, 6LoWPAN etc

Physical and MAC Layer

- IEEE 802.15.4 protocol – for low power embedded devices
- Defines standards and protocols for the physical layer and MAC layer
- Low power (1 mW as 1% as compared to WiFi), low cost and short range communication
- Enables multihop to cover longer distances
- Small packet size 127 bytes, communication rate 250kbps
- Supports short 16 bit link address to reduce header size, communication overhead and memory requirements.

Adaptation Layer

- 6LoWPAN – IPv6 over low power wireless communication standard
- Enables communication using IPv6 over IEEE 802.15.4 protocol
- Defines adaptation layer between the 802.15.4 link layer and the network layer
- 6LoWPAN can communicate with all other devices on the internet through a gateway (wifi or ethernet)
- IPv6 headers are not small to fit 127byte message size of 802.15.4 standard. Following optimizations are done to reduce the overhead
  - Header compression – some fields shared across packets
  - Fragmentation – message fragmentation
  - Link Layer forwarding – mesh under routing uses short address from link layer instead of network layer

Network Layer

- Responsible for routing the packets from transport layer
- Uses open routing protocol, RPL, based on distance vectors
- Builds a destination oriented directed acyclic graph
- Object function/constraints used to create the best path
  - Prefer encrypted links
  - Avoid battery powered devices
  - Minimize latency
  - Expected number of packets that need to be received

Transport Layer

- TCP or UDP
- TCP highly reliable but large overhead of connection oriented protocol
- UDP – preferred choice for IoT due to connectionless protocol

Application Layer

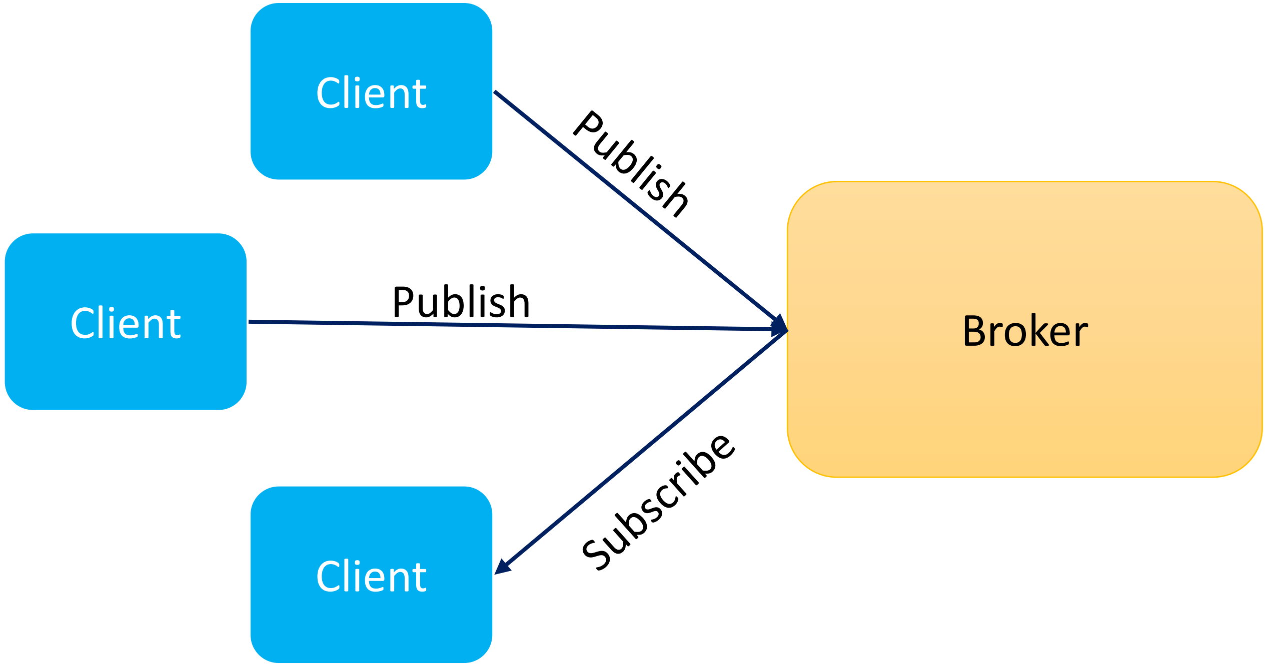
- Data formatting and presentation
- HTTP
  - a typical internet application layer
  - Not suitable for resource constrained IoT devices due to
    - Memory overhead
    - Large bandwidth requirement

HTTP (Hyper Text Transport Protocol)

- Text based robust protocol
- Based on REST (Representational State Transfer) architecture
- Client and server communicate through URI (Universal Resource Identifier) instead of header
- Not suitable for IoT devices
  - High bandwidth requirement – verbose and large parsing overhead
  - Resource usage
  - High delay
  - Use of TCP as a default transport protocol

MQTT (Message Queueing Telemetry Transport)

- Developed by IBM as a client/server protocol in late 1990's
- Message based publish/subscribe protocol that uses TCP as a transport layer protocol
- Header size: 2byte and small payload size maximum 256MB
- Broker
  - Authenticates clients
  - Coordinates subscribers
- Client
  - Subscriber/Publisher



MQTT

- Advantages
  - Lightweight protocol
- Disadvantages
  - All types of IoT devices do not support TCP. MQTT uses TCP protocol to communicate
  - Uses texts for topic names that leads to increase in overhead
- MQTT-SN
  - Optimized for wireless sensors networks (WSNs) for low power consumption
    - Use of IDs in topic instead of names
    - Topics are preregistered
    - Only required information gets sent
    - Messages get buffered and sent only when the device is in wake-up state

Data Distribution Service (DDS)

- A platform independent middleware developed by Open Management Group (OMG)
- Uses publish/subscriber broker less architecture
- Allows many to many communications
- Security – uses SSL and DLTS connections
- Reliability – supports wide variety of QoS mechanisms

CoAP (Constrained Application Protocol)

- Session based M2M communication protocol
- Uses binary data format, EXI (Efficient XML Interchanges) – memory efficient than XML/HTML
- Uses UDP as a transport layer protocol with request/response architecture
- Features
  - Header compression, resource discovery, autoconfiguration, asynchronous message exchange, congestion control and support for multicast messages
- Communicates using UDP – connectionless protocol
  - Uses conformable messages for reliable data transmission
  - Response gets piggybacked in the acknowledgement
- Uses DTLS (Datagram Transport Layer Security) for security purposes
- HTTP replacement for IoT protocols

CoAP

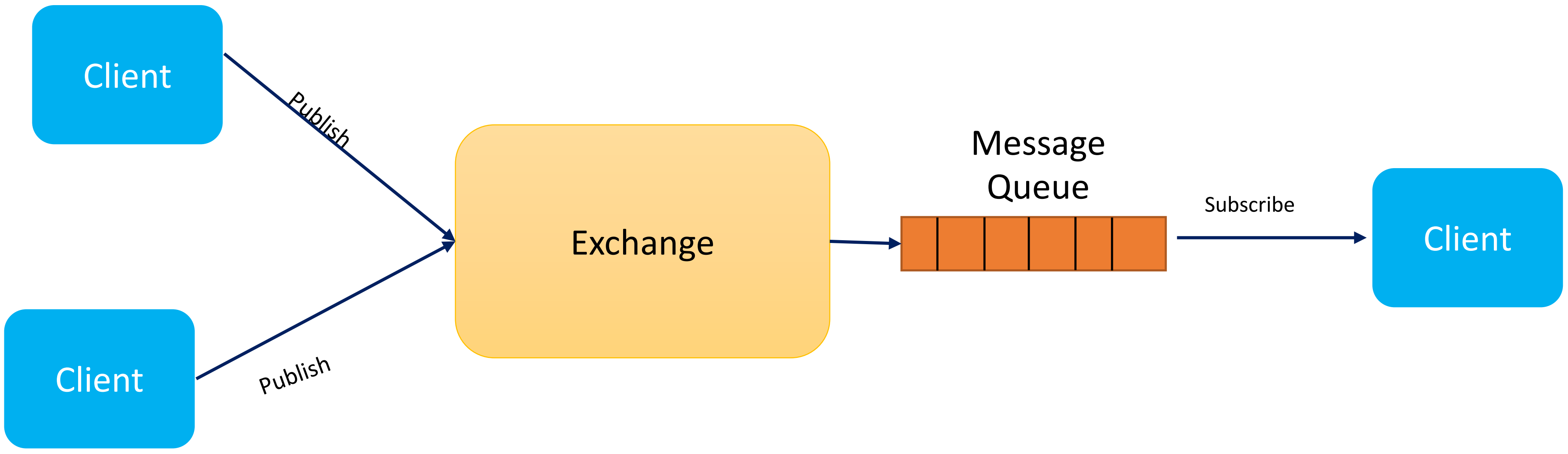
- CoAP protocol layers
  - Lower layer: message sublayer
  - Upper layer: request/response sublayer
- Message Sublayer
  - Types of messages
    - CON (confirmed), NON (non-confirmable), ACK (acknowledgment), RST (reset)
  - Communication models
    - Reliable Message Transport
    - Unreliable Message Transport
- Request/response sublayer
  - Implements RESTful similar to HTTP with GET, PUT, POST or DELETE methods

XMPP (Extensible Messaging and Presence Protocol)

- Originally designed for text messaging and exchange messages in applications
- Uses both publish/subscribe and request/response method of communication
- A text-based protocol with XML data format
- Uses TCP as a default transport layer protocol
- Security
  - Uses TLS methods to ensure privacy and integrity of the data.
  - Uses security, authentication, privacy and access control
- Suitability for IoT devices
  - Less delay in data transmission
  - Overhead of headers and tag formats – lead to more power consumption in communication

AMQP (Advanced Message Queueing Protocol)

- Designed to handle smart and reliable business transaction
- Uses publisher/subscriber architecture
- Uses TCP as a default transport layer protocol
- 3 levels of QoS: at least once, at most once and exactly once



Comparison of IoT Application Layer Protocols[2]

Table 2: Comparative of Communication Protocols for IoT Systems: HTTP, MQTT, DDS, XMPP, AMQP and CoAP

Characteristics	HTTP	MQTT	DDS	XMPP	AMQP	CoAP
Architecture	Client/Server	Client/Broker	broker-less	Client/Server	Client/Brokeror Client/Server	Request/Response or Publish/Subscribe
Abstraction	Request/Response	Publish/Subscribe	Broker-less Publish/ subscribe	Request/Response Publish/ Subscribe	Publish/Subscribe or Request/Response	Request/Response or Publish/Subscribe
Header Size	Undefined	2 Byte	-	no Header uses XML Stanza	8 Byte	4 Byte
Message Size	Heavyweight	lightweight	-	lightweight	Lightweight	yes
Cache and Proxy Support	Yes	Partial	yes	yes	Yes	yes
Quality of Service (QoS)/ Reliability	Limited (via Transport Protocol - TCP)	QoS 0 - At most once (Fire-and-Forget), QoS 1 - At least once, QoS 2 - Exactly	23 policies: Security, reliability, durability, priority etc.	No support for QoS	Settle Format (similar to At most once) or Unsettle Format (similar to At least once)	Confirmable Message (similar to At most once) or Non-confirmable Message (similar to At least once)
Transport Protocol	TCP	TCP (MQTT-SN can use UDP)	UDP	TCP	TCP, SCTP	UDP
Energy consumption	requires highest power/energy consumed by HTTP was much larger than with MQTT	MQTT was more energy efficient	----	Increase power consumption	requires slightly higher power	CoAP is more efficient in terms of energy
Security	TLS/SSL	TLS/SSLHas the lowest level	TLS/SSL, DTLS	TLS/SSL	TLS/SSL, IPSec, SASL Strongest security	DTLS, IPSec guarantee authentication, integrity and encryption
Connectivity	One -to-one	one-to-one, one-to-many and many-to-many	peer-to-peer communicat ion one-to-one, one-to-many, many-to-many, and many-to-one	One -to-one	point-to-point	one to one and many to many communications
Latency	involves largest latency, HTTP has highest latency than all others	MQTT has lowest latency than HTTP	Low latency	Low latency	AMQP has lowest latency than MQTT	CoAP has lowest latency than all others
Bandwidth consumption	involves largest bandwidth	consumes higher bandwidth	Low	Low	High consumption of bandwidth	involves lowest bandwidth
Encoding Format	Text	Binary	Binary	Text	Binary	Binary
Standards	IETF and W3C	OASIS, Eclipse Foundations	OMG	IETF	OASIS, ISO/IEC	IETF, Eclipse Foundation
Applications	Web	Home automation, Enterprise level applications	Medical Imaging, Military Systems,	Instant Messaging, Group chat, Gaming, Vehicle Tracking	Business Messaging, and in Banking Industry	Smart homes, smart grid and Building automations

References and Reading Material

1. Pallavi Sethi, Smruti R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications", *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, 25 pages, 2017. https://doi.org/10.1155/2017/9324035
2. Sidna, Jeddou, Baina Amine, Najid Abdallah, and Hassan El Alami. "Analysis and evaluation of communication Protocols for IoT Applications." In *Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications*, pp. 1-6. 2020.

Questions?