Advanced Message Queuing Protocol (AMQP) as a Communication Protocol Standard for IoT

Seminar: Communication Systems, Talk No. 5
Author: Maciej Lebiedz
IoT Protocols

- Infrastructure
- Identification
- Communication / Transport
- Discovery
- Data Protocols
- Device Management
- Semantic
- Multi-layer Frameworks
Data Protocols

• Message Queuing Telemetry Transport (MQTT)
• Constrained Application Protocol (CoAP)
• Extensible Messaging and Presence Protocol (XMPP)
• Advanced Message Queuing Protocol (AMQP)
Development Reasons for AMQP

• Need for an open standard for asynchronous messaging
• Only proprietary protocols or protocols with serious limitations
• Interoperability at enterprise-scale required by many different big actors in multiple different industries
  ➢ Formation of AMQP working group
  ➢ Iteratively developing the protocol based on feedback from the industry and developers
  ➢ Focus on reliability, scalability, performance and manageability
Main Characteristics

• Symmetric
• Multiplexed
• Secure
• Compact
• Reliable
• Binary
AMQP Model

Main elements:

• Containers, Nodes - endpoints
• Connections, Channels, Sessions, Links - routes
• Frames, Messages - payloads
Containers and nodes

Container:
• Communicating application

Node:
• Addressable entity within container
• Organized in any way
• Can be a consumer, producer, queue, relay or other entity

Picture source:
Connections and channels

Connection:
• Between two containers
• Usually layered over TCP
• Provides reliably ordered sequence of frames

Channel:
• Within connection
• Independent path through which messages can be sent
• Two channels can be bound together to form a session
Sessions

- Within connection, between containers
- Binds two unidirectional channels
- Provides a window-based flow control model
- There can be multiple concurrent sessions within a connection
Links

• Between nodes, within session
• Unidirectional message transfer route
• Named
• There can be multiple links within session
• Links can be recovered on a new connection in case connection breaks
Frames

- Transported over channel
- Ordered sequence
- Maximum size negotiated
- Type describing AMQP operation
• Containers connect to other containers
• Connections manage transfer capacity - frame size, channel count
• Sessions formed over pairs of one-directional channels
• Connections and sessions are ephemeral – if connection or session collapses, it cannot be recovered and must be setup again
Messages

- Transported over link
- Message format is extensible
- Bare message is immutable, cannot be changed by any intermediary between sender and receiver
- Annotated message can be changed by intermediaries
Message Transfer

- Unidirectional, named links formed over sessions
- Flow management through link credits
- In case of collapsed connection or session link can be recovered
Questions

• Can lack of competition from open-source standards be bad for AMQP?
Questions

• Complexity of AMQP – pro or con?
Questions

• How important is community for IT projects, can AMQP grow without community behind it?
The End

Thank you for your attention! 😊