MQTT-S – A Publish/Subscribe Protocol for Wireless Sensor Networks
CMPUT 642 – Winter 2010

Adam Wolfe Gordon
University of Alberta
11 February 2010
Applications on the traditional network want data from the sensor networks.

The two sensor networks might be co-located, but use different technologies.
Subscribers indicate their interest in some kind of data to a broker.
Publishers send their data to the broker.
The broker distributes information to subscribers, based on their indicated interests.
Kinds of Data

Three basic ways of classifying data for subscription purposes:

- **Topic-based**: A text-based or code-based topic is attached to each piece of data, with the list of topics usually being known by all nodes in advance.

- **Type-based**: Type information (e.g. temperature data) is attached to each piece of data.

- **Content-based**: General criteria can be specified in subscriptions (e.g. temperature data from a particular area that has a value of 10°C or higher).
The gateway allows the WSN nodes to interface with the broker, which is on the traditional network.

The broker and the gateway could be integrated into a single component.
Advantages of Publish/Subscribe for WSNs

- Data producers don’t need to know the addresses of their subscribers (they only send to the broker).
- This allows nodes to be replaced without any messages being lost, and without any other nodes knowing or caring.
- Easy to design applications on top of pub/sub architecture, with little limitation on the type of application.
MQTT: Overview

- Message Queueing Telemetry Transport
- A publish/subscribe protocol standard for traditional networks.
- Topic-based with text topics.
  - Topics are hierarchical: e.g. “wsn/sensor/F2/R248/temperature”
  - A subscriber can use wildcards in topics: e.g.
    “wsn/sensor/*/temperature”.
- Three QoS levels
  - Level 0: Best-effort delivery, no guarantees.
  - Level 1: Guaranteed delivery, multiple delivery possible.
  - Level 2: Guaranteed delivery, no multiple delivery.
MQTT: Details

- Every client – publisher or subscriber – sends a CONNECT message to the broker to establish a connection.
- The client must contact the broker periodically to keep its connection alive. It can send a PING message if it doesn’t have any data to send.
- A client can include a will message in its CONNECT, which is published on the client’s behalf if it disconnects unexpectedly (e.g. if a node dies).
- A client sends a SUBSCRIBE message with a topic string to subscribe, and a PUBLISH message with a topic string and message string to publish.
- A client sends an UNSUBSCRIBE message to unsubscribe from a topic.
- All messages are acknowledged by the receiver.
MQTT requires a point-to-point, session-oriented, in-order, auto-segmenting transport (e.g. TCP). Not normally available in WSNs.

Topics, especially hierarchical ones, can be long. The MAC schemes used in WSNs often limit message length (e.g. maximum packet length in ZigBee is 128 bytes).
MQTT-S: MQTT for Wireless Sensor Networks

- Designed to be very similar to MQTT.
- The clients are WSN nodes, which communicate via a gateway to a broker on a traditional network.
- The gateway may just translate messages between MQTT-S and MQTT, so the broker is a normal MQTT broker.
- Does not require a fancy transport: designed to work on any WSN architecture.
MQTT-S Network Requirements

- Point-to-point (multi-hop unicast) communication service.
- One-hop broadcast communication service.
- Unlike MQTT, does not require connection-oriented transport, message segmentation, or in-order delivery.
- Much more reasonable requirements for WSN environment.
The Gateway

- A transparent gateway simply translates each request and passes it on.
- An aggregating gateway uses a single connection to the broker, keeping track of which responses go to which client.
Multiple Gateways

- Nodes near the gateway may be unfairly loaded and die early.
- A gateway could fail.
- MQTT-S supports multiple gateways, with a gateway discovery procedure:
Reduction Message Size

- MQTT CONNECT and PUBLISH can be quite long, maybe too long for ZigBee.
- WSN transports may not support segmentation.
- Solution: Split CONNECT into multiple messages: CONNECT, WILLTOPIC, WILLMSG.
- Solution: Register topics so that the topic string doesn’t have to be sent when publishing.
  - Client registers a topic with the broker, and gets a topic code to use. Topic codes are client-specific.
  - Client then includes the topic code instead of the topic string in PUBLISH messages.
  - Applications can specify pre-defined topics.
Client Implementation

- Implemented on ZigBee and TinyOS.
- Uses platform-specific underlying protocols.
Gateway Implementation

- Written in Java.
- Connects to the WSNs through devices on the serial port, and connects to the broker using MQTT on a traditional network.
- Multiple gateways are supported.
- An aggregating gateway, which the authors argue will scale better than a transparent gateway.
- “Will” functionality doesn’t work.
  - Could be fixed by integrating the Gateway functionality into the Broker.
- QOS not supported.
The paper covers specification version 1.0, which doesn’t support sleeping clients.

In version 1.1:
Future Work

- Integration of the broker and the gateway.
- Testing with large WSNs.
- Evaluation of efficiency, etc.