MQTT Interop Test Day Report

March 17, 2014

Burlingame, California

Hosted by:

Eclipse IoT Working Group
iot.eclipse.org

Report Prepared by:

Ian Skerrett
Eclipse Foundation

Introduction and Summary

The first MQTT Interop Test Day was held March 17, 2014 in Burlingame, CA. A total of 15 different products and organizations participated in the event, resulting in 65 different test pairing between the different products.

The purpose of the first test day was to provide feedback and validation on the proposed OASIS MQTT specification and to provide initial feedback to the participants on their implementation of the MQTT 3.1.1 specification. Overall, the participants declared the event to be a success.

Introduction to MQTT

MQTT is a messaging protocol that has been developed specifically for the Internet of Things. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over occasional dial-up connections with healthcare providers, and in a range of home automation and small device scenarios.

MQTT was initially developed by IBM and Eurotech. In November 2012, IBM announced it would contribute the client implementation of MQTT to the Eclipse Paho open source project. In March 2013, IBM, Eurotech and others began the standardization process of MQTT within the OASIS standards group. The OASIS Technical Committee has released a draft specification of MQTT 3.1.1. The completion of the standardization process is expected in the Fall 2014, resulting in the MQTT 3.1.1 specification being published.

Participants in the Test Day

The following companies and products participated in the testing day:

- Eurotech
- 2lemetry Platform
- Sierra Wireless AirVantage
- ClearBlade
- HiveMQ
- WSO2 ESB and WSO2 Message Broker
- JBoss A-MQ, Apache ActiveMQ, Apache Camel
- Litmus Automation Loop
- Xively
- IBM Node-RED

- Software AG Universal Messaging
- Eclipse Kura
- Eclipse Paho
- IBM MessageSight
- RabbitMQ
- OpenPicus

Testing Approach

Each product was identified as being a MQTT client or MQTT broker. A schedule was developed to allow for testing between the different MQTT clients and brokers. Each testing pair were allocated approximately 30 minutes to complete the test. Most products participated in at least 5-6 test pairings.

Each participating product were provided advanced access to test clients and test brokers. The code for this test clients and brokers is available at https://wiki.eclipse.org/Interop_Testing_Plan

Test Plan
Each test pairing were asked to complete the following tests:

1. Connect to an MQTT server using the MQTT 3.1.1 standard (connect packet has protocol name MQTT and version 4)
2. Send and receive messages with all three QoS levels (0, 1 and 2).
3. Receive messages queued up for a non-cleansession client on reconnect. Each QoS should be sent and received. Queueing up QoS 0 messages is optional.
4. Set and receive retained messages at each QoS. Retained messages are set by publishing with the retained flag set. They are received by subscribing to a matching topic filter.
5. Unsetting retained messages. Sending publish with zero length payload and retained flag should remove any existing retained message for that topic.
6. Setting and receiving will messages. Ending the session (for example by closing the socket) without sending a disconnect should cause a will message to be sent by the server.
7. More than one subscription matching a publication topic. A server may send one message per matching subscription each at the QoS corresponding to the subscription, or one message for the publication at the highest QoS.
8. Keepalive processing. When no other messages are being sent/received, PINGs should be. For example, set the keepalive interval to 10 seconds, connect and send no messages. The session should be kept alive by PINGs from the client.

Optional Tests
1. 0-length clientid. The server may accept 0-length clientids. If so, the connect must be with cleansession True. Connecting with cleansession=False should be rejected.
2. $topics. Subscribing to a wildcard such as # should not return any topics starting with $. This is to enable $ topics to be used for server specific purposes.
3. Subscribe failure. Subscribe may now return 0x80 if it has failed for any reason. That reason would be server specific.

Test Results

Overall, 65 test pairings were reported. A total of 55% of these test pairings were considered successful. This is considered a good result considering the maturity of the specification, the variety of implementations of the 3.1.1 specification and the fact this was the first time for doing the interoperability testing.

Detailed test results will not be reported since the purpose of the session was to not certify an implementation. The findings of each test pairing will be used by each participant to ensure future conformance to the OASIS MQTT specification. Overall, the results will be used to validate the MQTT specification and standardization.

Quotes from Participants

2lemetry

"As a company that relies on interoperability with MQTT, the Eclipse Interop Test Day provided a great way to validate our services with other clients," said Chris Chiappone, CTO of 2lemetry. "Eclipse also did a great job creating a conformance test suite prior to the event, which helped us successfully pre-test client and server compliance."

Xively

"We believe that standardization and interoperability are critical to accelerating innovation in today's early-stage Internet of Things market," said Mario Finocchiaro, Director of Xively Business Development for LogMeIn. "The inaugural MQTT interoperability event proved to be a great opportunity for collaboration between emerging IoT leaders, while showing promise for encouraging and empowering broader commercial IoT adoption. Congratulations to Eclipse for bringing it all together."

ClearBlade

"The MQTT Interop Day was extremely valuable in bringing together both specification and implementation leaders, and reconfirming our objectives as a unified community. The ClearBlade enterprise platform was built on the foundation and belief that strong open standards and interoperation create enterprise value, and that is achieved with this level of industry collaboration."

Red Hat

"The Internet-of-Things (IoT) will require open standards to make it a reality. That is why Red Hat JBoss A-MQ, an open source enterprise message broker, supports open protocol standards such as AMQP 1.0 and MQTT. Red Hat has demonstrated its commitment to the MQTT standard by successfully taking part in the MQTT Interoperability Test Day at EclipseCon this year. JBoss A-MQ supports MQTT across TCP/TLS and WebSockets, and allows for an MQTT client to interoperate with other clients connected to JBoss A-MQ, regardless of the protocol that client is using." - Rob Davies, director of software engineering, Red Hat

Eurotech

"It has been really exciting to see so many companies investing in high quality MQTT implementations. The interoperability test has been a huge success and is a testament to how quickly the community is growing around this IoT protocol. We're looking forward to completing the MQTT standardization process within the OASIS standards body and continuing the comprehensive activities within the IoT Working Group of the Eclipse Foundation. With Eurotech there at the very beginning of the protocol definition, it's gratifying to see acceptance building so quickly now as MQTT helps to enable the Internet of Things."

IBM

"The mix of MQTT technologies represented, ranging from open source to commercial, and the successes achieved at the event demonstrate the vibrancy of the MQTT ecosystem and means we will be well positioned as we move toward a ratified standard. Interoperability of technology in this space is vital for the Internet of Things ecosystem." - Dave Locke, IBM senior inventor and co-lead of the Eclipse Paho project.

Conclusion

The MQTT Interop Test Day was considered an overall success in achieving the goals and purpose. The consensus of the group was to run a future Test Day later in 2014 to validate the final MQTT 3.1.1 specification.