Formal verification of the implementation of the MQTT protocol in IoT devices

Kristiyan Mladenov

University of Amsterdam
Faculty of Physics, Mathematics and Informatics
MSc System and Network Engineering
Research Project 2

July 3, 2017

Introduction

- Mirai botnet producing one of the largest DDoS attacks ever.
- We can also talk about botnet "wars".
- Compromise due to human error.

IoT testing

- Rapid7 IoT Security Testing Methodology
- OWASP IoT Top 10
- IoT Inspector (SEC Technologies)

IoT testing

- Rapid7 IoT Security Testing Methodology
- OWASP IoT Top 10
- IoT Inspector (SEC Technologies)

What would happen if we dig deeper?

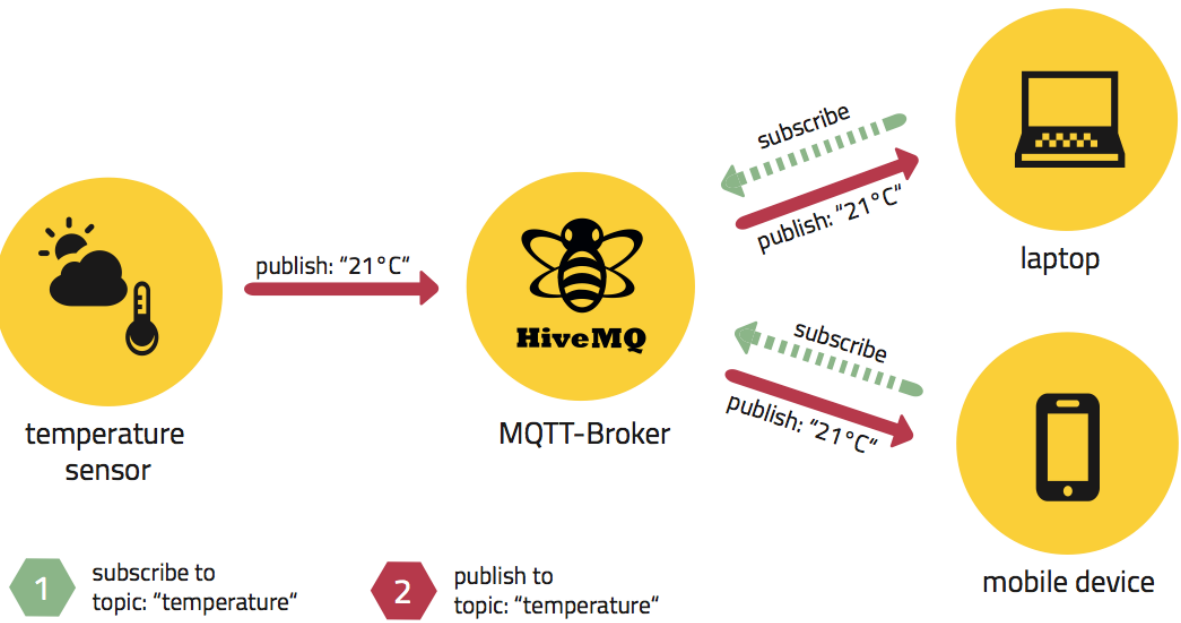
- One of the main goals of the IoT devices is to exchange data using some message exchange mechanism.
- How can we assure a proper protocol implementation?
- Could we make sure that it is correct in a more formal way?

Protocol of choice

MQTT

Message Queue Telemetry Transport

- Designed for message transfer with small code footprint and limited bandwidth in mind.
- First version was available in 1999. Version 3.1.1 is standardised by OASIS (2014) and ISO (2016).

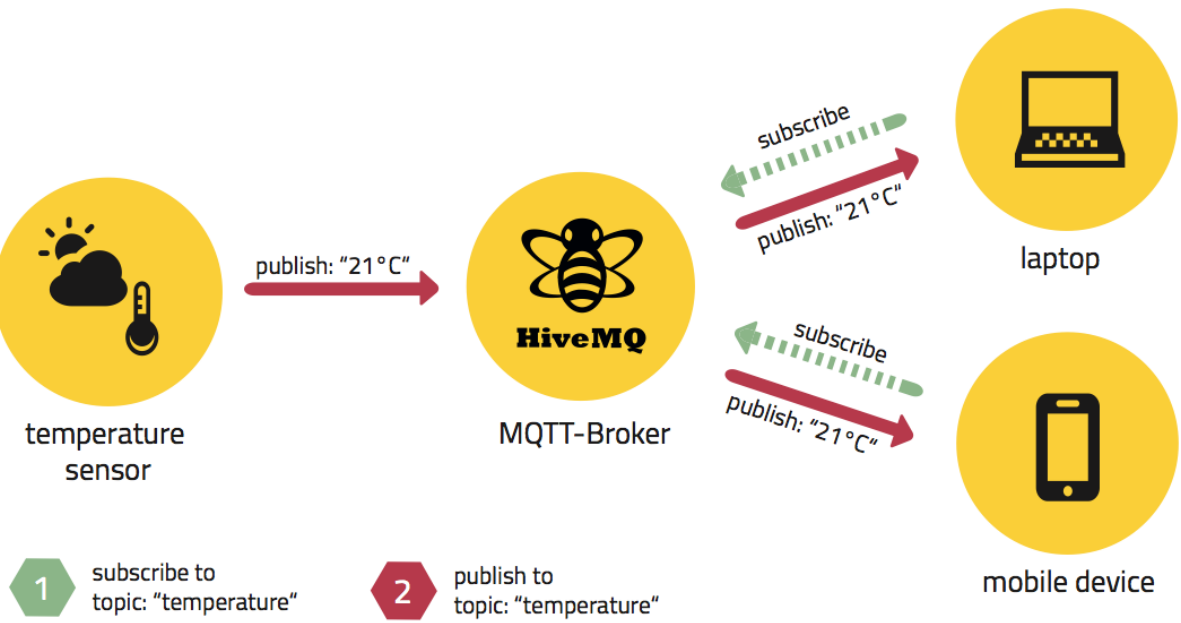


Protocol of choice

MQTT

Message Queue Telemetry Transport

- Designed for message transfer with small code footprint and limited bandwidth in mind.
- First version was available in 1999. Version 3.1.1 is standardised by OASIS (2014) and ISO (2016).
- Publish/Subscribe communication mechanism similar to IRC.
- Adds the concept of Last Will and QoS.



MQTT use cases

MQTT is implemented in:

- The backend of The Things Network (LoRa)
- AWS IoT, Google Cloud IoT

MQTT use cases

MQTT is implemented in:

- The backend of The Things Network (LoRa)
- AWS IoT, Google Cloud IoT

Applications that use MQTT

- Fitness trackers, Medical equipment, ATM machines
- Implemented by Deutsche Bahn (DB)
- Facebook Messenger (Unconfirmed)

Research Question

Can the MQTT protocol implementation in IoT devices be verified formally?

Subquestions

- What methods can be used to formally assess the implementation of a communication protocol?
- Using the chosen formal testing methods, does the MQTT implementation in certain selected IoT devices adhere to the standard?

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework.

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.

There is a tool for every approach

Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.

There is a tool for every approach

- The testing to follow is focused on Eclipse Titan.

MQTT Packet Structure

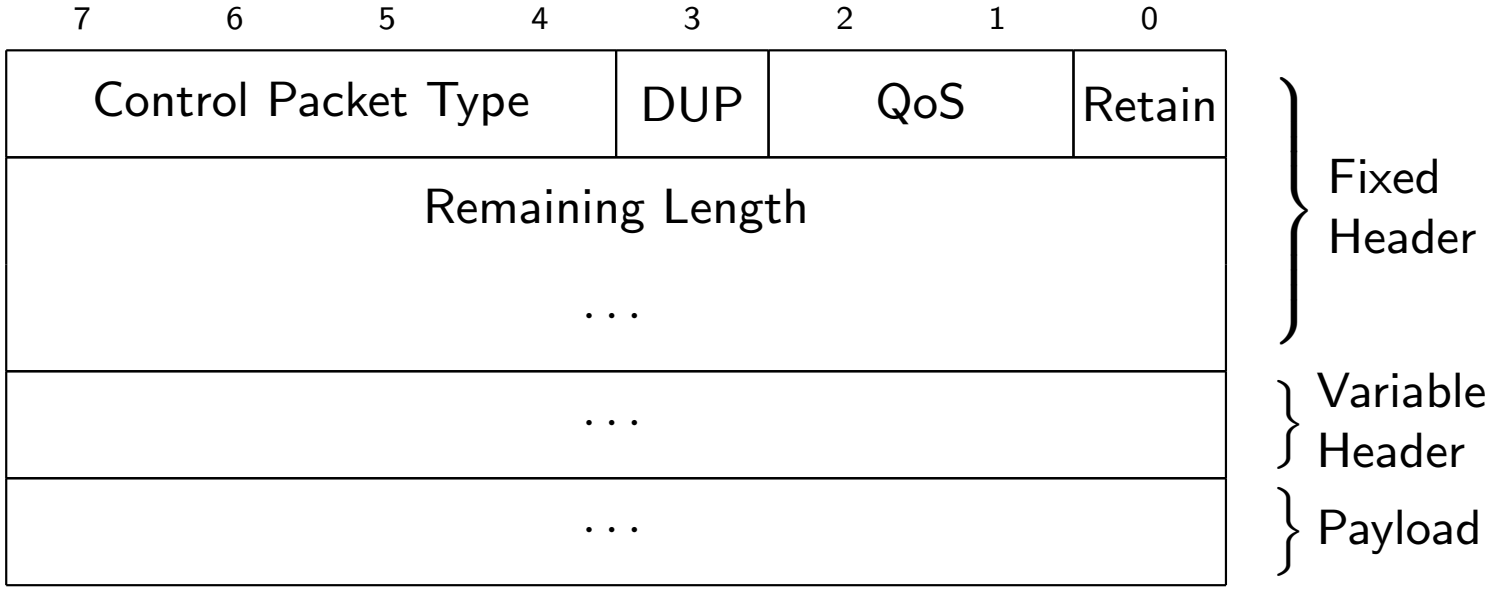


Figure: MQTT Packet structure

Example test

[MQTT-2.3.1-1]

SUBSCRIBE, UNSUBSCRIBE, and PUBLISH (in cases where QoS>0) Control Packets MUST contain a non-zero 16-bit Packet Identifier.

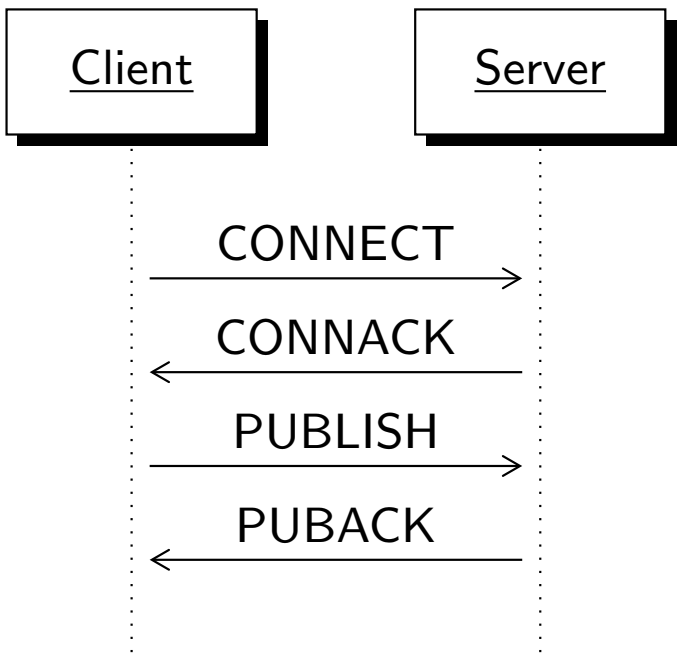


Figure: Publish with Packet ID 0

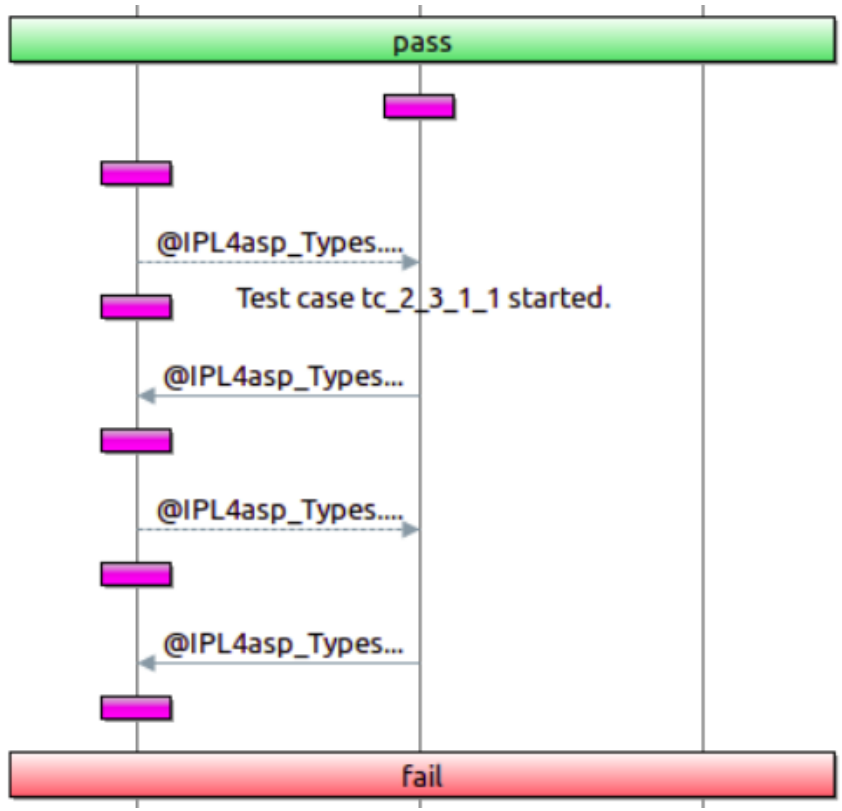


Figure: Test execution flowgraph

Room for improvement

Writing is nature's way of letting you know how sloppy your thinking is.[1]

[1] Dick Guidon

Room for improvement

Writing is nature's way of letting you know how sloppy your thinking is.[1]

- Translating a specification from natural to formal language is prone to errors.
- How can we safely come up with new values for the tests?
- If the specification is defined in a formal language, testing might be easier.

[1] Dick Guidon

Intermezzo

The Die Hard challenge[2]

- You have two buckets
  - 3 litres
  - 5 litres
- You have an infinite amount of water.
- You can waste as much water as you want.
- How do you fill the large bucket with exactly 4 litres?

[2] `https://github.com/tlaplus/Examples/tree/master/specifications/DieHard`

Intermezzo approach (enter TLA+)

Intermezzo approach (enter TLA+)

$$\text{EXTENDS } \mathit{Integers}$$

$$\text{VARIABLES } \mathit{small},\ \mathit{big}$$

$$\begin{aligned} \mathit{TypeOK} \triangleq\ & \land\ \mathit{small} \in 0\,..\,3 \\ & \land\ \mathit{big}\ \ \in 0\,..\,5 \end{aligned}$$

$$\begin{aligned} \mathit{Init} \triangleq\ & \land\ \mathit{big}\ \ = 0 \\ & \land\ \mathit{small} = 0 \end{aligned}$$

$$\begin{aligned} \mathit{FillSmall} \triangleq\ & \land\ \mathit{small}' = 3 \\ & \land\ \mathit{big}'\ \ = \mathit{big} \end{aligned}$$

$$\begin{aligned} \mathit{FillBig} \triangleq\ & \land\ \mathit{big}'\ \ = 5 \\ & \land\ \mathit{small}' = \mathit{small} \end{aligned}$$

$$\begin{aligned} \mathit{EmptySmall} \triangleq\ & \land\ \mathit{small}' = 0 \\ & \land\ \mathit{big}'\ \ = \mathit{big} \end{aligned}$$

$$\begin{aligned} \mathit{EmptyBig} \triangleq\ & \land\ \mathit{big}'\ \ = 0 \\ & \land\ \mathit{small}' = \mathit{small} \end{aligned}$$

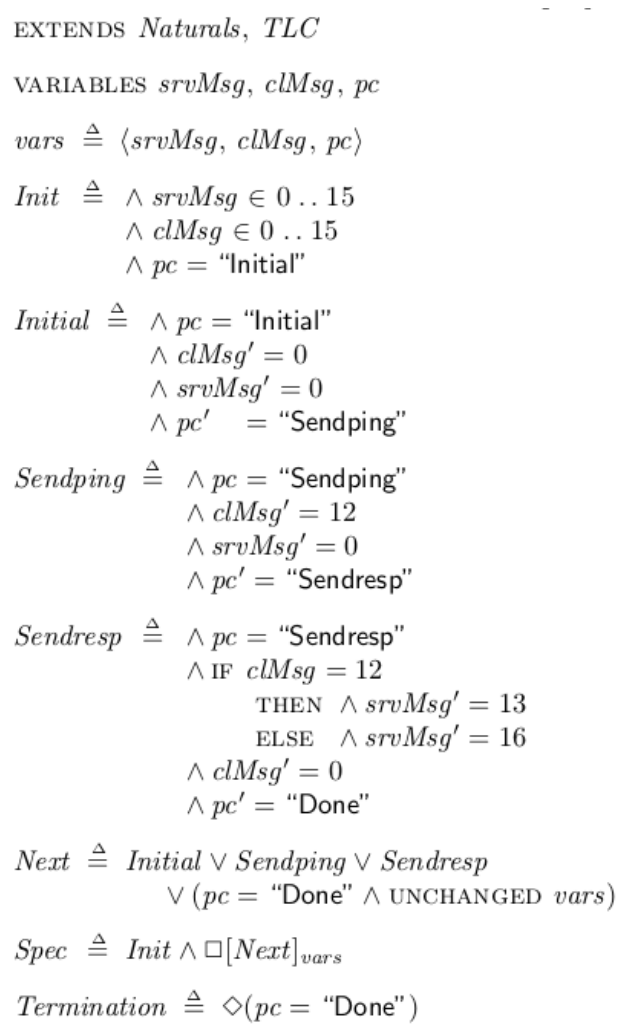
$$\begin{aligned} \mathit{SmallToBig} \triangleq\ & \text{IF } \mathit{big} + \mathit{small} \leq 5 \\ & \text{THEN } \land\ \mathit{big}'\ \ = \mathit{big} + \mathit{small} \\ & \qquad\ \land\ \mathit{small}' = 0 \\ & \text{ELSE } \land\ \mathit{big}'\ \ = 5 \\ & \qquad\ \land\ \mathit{small}' = \mathit{small} - (5 - \mathit{big}) \end{aligned}$$

$$\begin{aligned} \mathit{BigToSmall} \triangleq\ & \text{IF } \mathit{big} + \mathit{small} \leq 3 \\ & \text{THEN } \land\ \mathit{big}'\ \ = 0 \\ & \qquad\ \land\ \mathit{small}' = \mathit{big} + \mathit{small} \\ & \text{ELSE } \land\ \mathit{big}'\ \ = \mathit{small} - (3 - \mathit{big}) \\ & \qquad\ \land\ \mathit{small}' = 3 \end{aligned}$$

$$\begin{aligned} \mathit{Next} \triangleq\ & \lor\ \mathit{FillSmall} \\ & \lor\ \mathit{FillBig} \\ & \lor\ \mathit{EmptySmall} \\ & \lor\ \mathit{EmptyBig} \\ & \lor\ \mathit{SmallToBig} \\ & \lor\ \mathit{BigToSmall} \end{aligned}$$

TLA+ model of a simple MQTT keepalive

- Define different invariant in the TLA+ model checker.
- Observe the behaviour of the model; relax constraints if necessary.
- Map the observed behaviour in terms of TTCN-3 tests.
- The problem of translating natural to formal language is still not solved.



$$
\begin{array}{l}
\text{EXTENDS } \mathit{Naturals},\ \mathit{TLC} \\
\\
\text{VARIABLES } \mathit{srvMsg},\ \mathit{clMsg},\ \mathit{pc} \\
\\
\mathit{vars} \triangleq \langle \mathit{srvMsg},\ \mathit{clMsg},\ \mathit{pc} \rangle \\
\\
\mathit{Init} \triangleq \begin{array}{l} \land\ \mathit{srvMsg} \in 0\,..\,15 \\ \land\ \mathit{clMsg} \in 0\,..\,15 \\ \land\ \mathit{pc} = \text{“Initial”} \end{array} \\
\\
\mathit{Initial} \triangleq \begin{array}{l} \land\ \mathit{pc} = \text{“Initial”} \\ \land\ \mathit{clMsg}' = 0 \\ \land\ \mathit{srvMsg}' = 0 \\ \land\ \mathit{pc}' \quad = \text{“Sendping”} \end{array} \\
\\
\mathit{Sendping} \triangleq \begin{array}{l} \land\ \mathit{pc} = \text{“Sendping”} \\ \land\ \mathit{clMsg}' = 12 \\ \land\ \mathit{srvMsg}' = 0 \\ \land\ \mathit{pc}' = \text{“Sendresp”} \end{array} \\
\\
\mathit{Sendresp} \triangleq \begin{array}{l} \land\ \mathit{pc} = \text{“Sendresp”} \\ \land\ \text{IF } \mathit{clMsg} = 12 \\ \qquad \text{THEN } \land\ \mathit{srvMsg}' = 13 \\ \qquad \text{ELSE } \land\ \mathit{srvMsg}' = 16 \\ \land\ \mathit{clMsg}' = 0 \\ \land\ \mathit{pc}' = \text{“Done”} \end{array} \\
\\
\mathit{Next} \triangleq \mathit{Initial} \lor \mathit{Sendping} \lor \mathit{Sendresp} \\
\qquad \lor (\mathit{pc} = \text{“Done”} \land \text{UNCHANGED } \mathit{vars}) \\
\\
\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}} \\
\\
\mathit{Termination} \triangleq \Diamond(\mathit{pc} = \text{“Done”})
\end{array}
$$

Figure: TLA+ simplified keepalive

Results

What follows is a list of the normative requirements and how do the tested implementations conform to them.

Normative Requirements

	2.2.2	2.3.1-1	3.1.0-1a	3.1.0-1b	3.1.0-2	3.1.2-2	3.1.2-24	3.1.3-8	3.3.1-4	3.6.1-1	3.8.1-1	3.8.3-4	3.12.4-1
Mosquitto	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Emqtt	X	X	✓	✓	✓	X	✓	✓	✓	X	X	✓	✓
RabbitMQ	X	X	✓	✓	✓	X	✓	✓	✓	X	✓	✓	✓

Results

What follows is a list of the normative requirements and how do the tested implementations conform to them.

Normative Requirements

	2.2.2	2.3.1-1	3.1.0-1a	3.1.0-1b	3.1.0-2	3.1.2-2	3.1.2-24	3.1.3-8	3.3.1-4	3.6.1-1	3.8.1-1	3.8.3-4	3.12.4-1
Mosquitto	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Emqtt	X	X	✓	✓	✓	X	✓	✓	✓	X	X	✓	✓
RabbitMQ	X	X	✓	✓	✓	X	✓	✓	✓	X	✓	✓	✓

Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.

Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.
- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.

Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.
- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.
- Those inconsistencies can be used to fingerprint and identify implementations.

Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.
- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.
- Those inconsistencies can be used to fingerprint and identify implementations.

As a side note, adhering to the standard does not mean that a device is secure, especially in the cases of bad protocol design.

Future work

- Building a complete TLA+ model could be able to identify additional behavioural differences between different implementations.
- The output derived from the TLA+ model might be used for fuzzing.
- It could also help in identifying deficiencies in the protocol design itself, rendering all implementations vulnerable.

Questions?

Share your thoughts?

References

- Image depicting the interaction between the MQTT Client and Server taken from:
  `http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe`
- Representative solution to the Die Hard problem taken from:
  `https://github.com/tlaplus/Examples/tree/master/specifications/DieHard`