Ad Hoc Networks 36 (2016) 49–57

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A formal model and analysis of an IoT protocol

Benjamin Aziz*

School of Computing, University of Portsmouth, Portsmouth PO1 3HE, United Kingdom

ARTICLE INFO

Article history:
Received 17 December 2014
Revised 4 May 2015
Accepted 27 May 2015
Available online 3 June 2015

Keywords:
Formal verification
IoT
MQTT

ABSTRACT

We present a formal model of the MQ Telemetry Transport version 3.1 protocol based on a timed message-passing process algebra. We explain the modelling choices that we made, including pointing out ambiguities in the original protocol specification, and we carry out a static analysis of the formal protocol model, which is based on an approximation of a name-substitution semantics for algebra. The analysis reveals that the protocol behaves correctly as specified against the first two quality of service modes of operation providing at most once and at least once delivery semantics to the subscribers. However, we find that the third and highest quality of service semantics is prone to error and at best ambiguous in certain aspects of its specification. Finally, we suggest an enhancement of this level of QoS for the protocol.

1. Introduction

The Internet of Things (IoT) [1] is a new paradigm with the aim of creating connectivity for "everything" that can carry a minimum of storage and computational power, such that these connected things can collaborate anytime, anywhere and in any form, within applications in various domains such as personal and social, transportation, enterprise businesses and service and utility monitoring [2,3]. Some recent estimates suggest that the number of IoT devices exceeds 30 billion with more than 200 billion intermittent connections [4] generating over 700 billion Euros in revenue by 2020 [5].

This connectivity of IoT devices has been boosted in recent times with the increasing popularity of mobile communications, such as wireless sensor networks and radio frequency identification technologies, and the proliferation of small hardware with minimum computational and storage capabilities. These coupled with the standardisation efforts of Machine-to-Machine (M2M) communication protocols, such as MQTT [6], XMPP [7] and others, meant that the global vision of the IoT is well within reach of industries and their markets. However, such global applications may require, in contexts where criticality is an issue, a minimum degree of reliability in terms of the correctness of the specification of the system as well as its level of assurance with respect to non-functional properties such as security and privacy. Therefore, there is a need for adopting formal analysis techniques to ensure that specifications are as little ambiguous as possible leading to more reliable and robust applications built out of those specifications.

This paper presents a formal model of the MQTT protocol above based on a timed process algebra, called TPi, and then defines an abstract static analysis that approximates the behaviour of processes by limiting the number of copies of input variables and new names that can be captured in the analysis during communications. The analysis is applied manually to the protocol to attempt to understand how robust the behaviour of the protocol is in the different quality of service scenarios mentioned above. The main contribution of the paper therefore is to formalise the specification of the MQTT protocol and to analysis its semantics. Based on this analysis, the paper also makes recommendations for future improvements of the protocol in order to remove current ambiguity in its specification and enhance its dependability.

* Tel.: +44 2392842265; fax: +44 2392842525.
E-mail address: benjamin.aziz@port.ac.uk

http://dx.doi.org/10.1016/j.adhoc.2015.05.013
1570-8705/

1.1. The MQTT Protocol

The MQ Telemetry Transport (MQTT) protocol - version 3.1 [6] is described as a lightweight broker-based publish/subscribe messaging protocol that was designed to allow devices with small processing power and storage, such as those which the IoT is composed of, to communicate over low-bandwidth and unreliable networks. The publish/subscribe message pattern [8], on which MQTT is based, provides for one-to-many message distribution with three varieties of delivery semantics, based on the level of quality of service expected from the protocol.

In the "at most once" case, messages are delivered with the best effort of the underlying communication infrastructure, which is usually IP-based, therefore there is no guarantee that the message will arrive. This protocol, termed the QoS = 0 protocol, is represented by the following flow of messages and actions:

Client → *Server* : **Publish**
Server Action : *Publish message to subscribers*

In the second case of "at least once" semantics, certain mechanisms are incorporated to allow for message duplication, and despite the guarantee of delivering the message, there is no guarantee that duplicates will be supressed. This case is represented by the following flow of messages and actions:

Client → *Server* : **Publish**
Client Action : *Store Message*
Server Actions : *Store Message*,
Publish message to subscribers,
Delete Message
Server → *Client* : **Puback**
Client Action : *Discard Message*

The second message **Puback** represents an acknowledgement of the receipt of the first message, and if **Puback** is lost, then the first message is retransmitted by the client (hence the reason why the message is stored at the client). Once the protocol completes, the client discards the message. This protocol is also known as the QoS = 1 protocol.

Finally, for the last case of "exactly once" delivery semantics, also known as the QoS = 2 protocol, the published message is guaranteed to arrive only once at the subscribers. This is represented by the following flow of messages and actions:

Client → *Server* : **Publish**
Client Action : *Store Message*
Server Actions : *Store Message OR*
Store Message ID,
Publish message to subscribers
Server → *Client* : **Pubrec**
Client → *Server* : **Pubrel**
Server Actions : *Publish message to subscribers*,
Delete Message OR
Delete Message ID
Server → *Client* : **Pubcomp**
Client Action : *Discard Message*

In this protocol, **Pubrec** and **Pubcomp** represent acknowledgement messages from the server, whereas **Pubrel** is an acknowledgement message from the client. The loss of **Pubrec** causes the client to recommence the protocol from its beginning, whereas the loss of **Pubcomp** causes the client to retransmit only the second part of the protocol, which starts at the **Pubrel** message. This additional machinery presumably ensures a single delivery of the published message to the subscribers.

The protocol additionally defines the message structure needed in communications between *client*, i.e. end-devices responsible for generating data from their domain (the data source) and *servers*, which are the system components responsible for collating source data from clients/end-devices and distributing these data to interested subscribers.

1.2. Paper structure

The rest of the paper is organised as follows. In Section 2, we describe related work in current literature and in Section 3, we provide an overview of the TPi process algebra, a timed version of the π-calculus [9]. In Section 4, we develop a model of the MQTT protocol based on TPi, and explain the various modelling options that we adopted. In Section 5, we give an analysis of the protocol in the context of its three versions of the delivery semantics. In Section 6, we discuss the outcome of the analysis and make a recommendation towards the enhancement of the MQTT protocol in the case of QoS = 2. Finally, in Section 7, we conclude the paper and provide directions for future research.

2. Related work

Publish/subscribe is increasingly becoming an important communication paradigm [10], in particular within the domain of sensor device networks and the Internet-of-Things where messages can be communicated with more efficiency and less consumption of the devices' limited computational power. IBM's MQTT-S protocol [11] was one of the first industrially backed lightweight publish/subscribe protocols that was deployed for wireless sensor and actuator networks. This was followed in year 2010 by version 3.1 [6], which is currently undergoing standardisation by the OASIS community.

There has been very little effort in applying formal analysis tools to IoT communication protocols, mainly due to the novelty of such protocols and their very recent arrival at the scene of communication protocols. On the other hand, some work has been done in the area of publish-subscribe protocols in general. An early attempt in [12] was made to model formally publish/subscribe protocols to capture their essential properties such as minimality and completeness, however, without any attempt to incorporate hostile environments within which these protocols may run. One aspect of their model is the use of an incrementing global clock $\mathcal{T}$, similar to our concept of the function $\delta(P)$, which is needed in order to model the passing of time.

In [13], the authors define a formal model of publish/subscribe protocols, within the domain of Grid computing, based on Petri-Nets. Their model offers a mechanism