

IoT 向けメッセージングプロトコルの スループット評価

森敬祐¹ 古屋保² 下園幸一² 升屋正人²

概要：軽量性や省電力性が求められる IoT で用いられる通信プロトコルに MQTT がある。MQTT は、HTTP など他のプロトコルに比べてヘッダ長が短く、伝送に際してデータ以外に消費される帯域が少ない。このためネットワーク帯域幅が小さい場合により多くのデータを伝送でき、高いスループットを実現できる。しかし、安全な通信を実現するため暗号化を組み込んだ MQTTS ではスループットの低下が予想される。そこで本研究では、複数の条件下で MQTT と MQTTS の伝送データ量と伝送時間を収集し、スループットの評価を行った。その結果、予想に反して MQTTS の方が MQTT より高いスループットとなることが分かった。

キーワード：MQTT, MQTTS, スループット, QoS

Throughput evaluation of messaging protocols for IoT

KEISUKE MORI¹ TAMOTU FURUYA²
KOICHI SIMOZONO² MASATO MASUYA²

Abstract: MQTT is a communication protocol used in the IoT, which requires lightweight and power-saving features, and has a shorter header length than other protocols such as HTTP. This allows more data to be transmitted when network bandwidth is small, resulting in higher throughput. However, MQTTS, which incorporates encryption for secure communication, is expected to have lower throughput. In this study, we evaluated the throughput of MQTT and MQTTS by collecting the amount of data and transmission time under multiple conditions. The results showed that, contrary to expectations, MQTTS has a higher throughput than MQTT.

Keywords: MQTT, MQTTS, Throughput, QoS

1. はじめに

アプリケーション層のプロトコルである MQTT (MQ Telemetry Transport) は軽量の通信に向いていることから、IoT 機器に用いられる。同じく IoT 機器に使用されるアプリケーション層のプロトコルである HTTP (Hyper Text Transfer Protocol) はヘッダ長や通信の複雑性により電力消費の増加や通信速度の低下が問題となるため、最低 2 バイトのヘッダ長での通信や双方向通信をすることができる MQTT プロトコルが注目されている。

この MQTT に平文で送られるメッセージを TLS で暗号化することにより、通信内容の盗聴や改ざんを防ぐことができるようにしたプロトコルが MQTTS である。TLS で暗号化した場合は暗号化しない場合に比べて複雑性が増し、ス

ループットが低下することが予想される。また、メッセージ長が大きくなるほどパケットサイズも大きくなり、より処理に時間がかかり、スループットはより低下するはずである。

実環境での MQTT の性能検証[1]や MQTT と MQTT-SN のスループットの評価[2,3]を行った研究の例はあるが、MQTTS の評価は行われていない。そこで本研究では MQTT と MQTTS のスループットを測定し、その差を明らかにすることにした。

¹ 鹿児島大学大学院理工学研究科
Graduate School of Science and Engineering, Kagoshima University
² 鹿児島大学情報基盤統括センター

Center for Management of Information Technologies, Kagoshima University

2. MQTT/MQTTs について

2.1 Pub/Sub アーキテクチャ

Pub/Sub 型の通信は、データ送信者である Publisher, 受信者である Subscriber, そしてそれらの通信を仲介する Broker から構成される。Publisher は Broker に接続し、Topic を指定してメッセージを送信する。Subscriber は Broker に一度接続すると、常時接続し続け、Subscriber は Subscribe (購読) している Topic が Broker にある場合、メッセージを Broker から受け取ることができる。これにより Publisher と Subscriber は直接、通信を行わず、仲介者である Broker を経由してメッセージの配信/購読をすることができる。

2.2 MQTT

MQTT は 1990 年代に開発されたアプリケーション層プロトコルであり、通信にはトランスポート層の TCP が使われる。プロトコルのヘッダは図 1 のようになり、最小ヘッダは 2 バイトである。これは最小 50 バイトの HTTP プロトコルに比べ、かなり小さい。そのため MQTT はデータサイズが小さくなり、通信品質が悪い場合でもパケットロスを減らすことができる。また MQTT 通信は Publisher と Broker 間、または Subscriber と Broker 間で行われ、その通信形態は図 2 に示す通りである。



図 1 MQTT プロトコルヘッダ

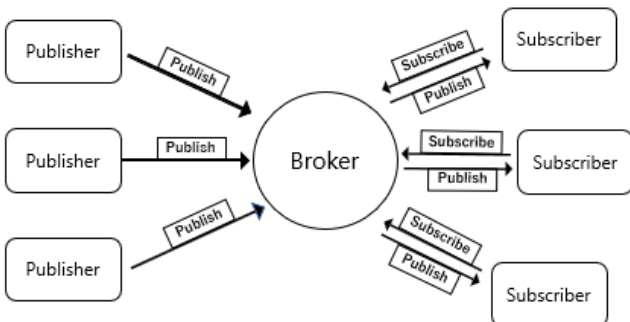


図 2 MQTT アーキテクチャ

2.3 MQTTs

MQTTs は TLS 上で MQTT を実現したアプリケーション層プロトコルである。MQTTs では通信内容が Publisher と Broker 間、Subscriber と Broker 間で暗号化される。本研究では TLS のバージョンに TLS1.2 を用いた。

2.4 QoS 機能

QoS (Quality of Service) とはネットワーク上で提供する通信サービス品質のことで、特定の通信で利用する帯域幅の保証や通信優先度の設定をするために利用される技術である。MQTT における QoS のレベルは表 1 に示すように 3 種ある。QoS 0 は最高 1 回、QoS 1 は 1 回以上、QoS 2 は必ず 1 回メッセージを届ける仕組みとなっている。QoS レベルが上がるにつれ、通信の到達保障は高くなるが、確認応答 (Ack) の頻度が上がるため通信量が増加する。

表 1 MQTT における QoS 機能

QoS レベル	到達保証	到達回数	通信量
QoS 0	なし	最高 1 回	低

QoS 1	あり	1 回以上	中
QoS 2	あり	必ず 1 回	高

3. 測定方法

3.1 MQTT と MQTTs 通信の測定環境と測定方法

MQTT と MQTTs 通信の測定のため Publisher, Subscriber, Broker を用意した。Publisher と Broker 間、Subscriber と Broker 間の通信は IoT デバイスにおける通信を想定して無線 LAN アクセスポイントを用いて行った。測定環境を図 3 に示す。測定に際してはまず事前準備として Broker を立ち上げたのちに Subscriber から指定した Topic を Broker に登録した。次にパケットを収集するために Broker で Wireshark を実行した。そして Publisher で Topic を指定し、メッセージを Broker へと送信した。メッセージを受信した Broker は指定された Topic に従って、Subscriber にメッセージを送信する。測定環境の条件として、メッセージ長及び Publisher・Subscriber 数、QoS レベルを変更した。それぞれの値は、メッセージ長は 2 バイトと 2000 バイト、Publisher・Subscriber 数は 1 と 4、QoS レベルは QoS 0、QoS 1、QoS 2 とした。

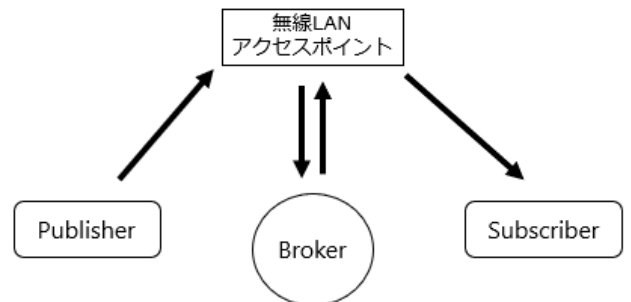


図 3 MQTT と MQTTs の測定環境

3.2 使用環境

本研究で使用した機器は Publisher と Subscriber では PC を使用し、Broker では Raspberry Pi、無線 LAN アクセスポイントでは IEEE802.11g を使用した。それぞれの性能を表 2、表 3、表 4 に示す。MQTT と MQTTs の測定で使用した機器は同じである。

表 2 Broker である Raspberry Pi 4 の性能

CPU	1.5GHz クアッドコア Cortex-A72
メモリ	8GB
OS	Raspberry Pi OS

表 3 Publisher と Subscriber である PC の性能

CPU	Intel Core i5-3320M 2.60GHz
メモリ	4GB
NIC	ELECOM WDC-150SU2MBK
OS	Windows10

表 4 無線 LAN アクセスポイントの仕様

規格	IEEE802.11g
周波数帯	2.4GHz 帯
通信方式	OFDM 方式/MIMO/DS-SS
最大伝送速度	54Mbps

3.3 スループットの計算方法

本研究では Publisher から送られた 10 回のメッセージが Broker を経由して Subscriber に到達するまでに要する時間で、Publisher と Broker 間、Subscriber と Broker 間で伝送されたすべての伝送データ量を除することによりスループットを求めた。なお、Publisher が連続してメッセージを送る場合には、その間隔を 0.3 秒空けた。

4. 測定結果

MQTT と MQTTS のスループットを測定した結果を図 4～7 に示す。棒グラフは左から順に QoS レベルを QoS 0, QoS 1, QoS 2 としたときのスループットを表している。Publisher が 10 回のメッセージを送ったときの全ての伝送時間と伝送データ量を 5 回測定し、それぞれの平均値を求め、伝送データ量の平均を伝送時間の平均で除することによりスループットを求めた。

図 4 と図 5 からわかるように、メッセージ長が 2 バイトと 2000 バイト、いずれの場合も QoS レベルや Publisher・Subscriber 数に関わらず MQTTS の方が高いスループットとなった。

図 6 と図 7 からわかるように、Publisher・Subscriber 数が 4 の QoS0 の場合を除いて、メッセージ長が 2 バイトと 2000 バイト、いずれの場合も QoS レベルやパブリッシャ・サブスクライバ数に関わらず MQTTS の方が高いスループットとなった。

また、図 4 と図 6、図 5 と図 7 からわかるように、伝送データ量が大きい方が高いスループットになった。

さらに、プロトコルやメッセージ長、Publisher・Subscriber 数に関わらず QoS 0, QoS 1, QoS 2 の順番にスループットが高くなっていくことがわかった。

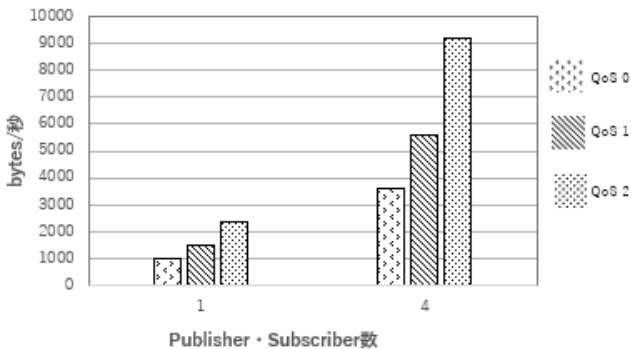


図 4 MQTT で 2 バイトのデータを伝送したときのスループット

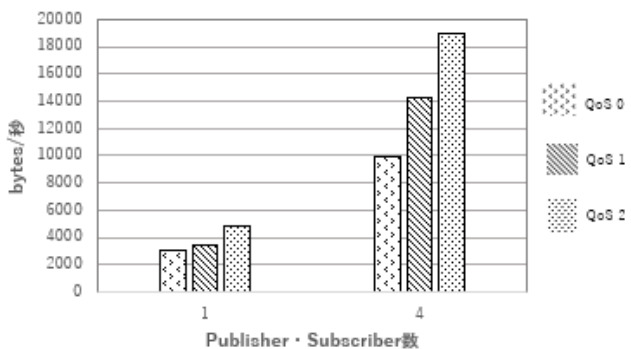


図 5 MQTTS で 2 バイトのデータを伝送したときのスループット

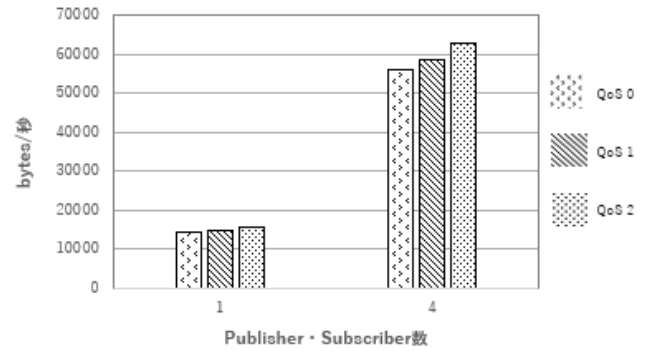


図 6 MQTT で 2000 バイトのデータを伝送したときのスループット

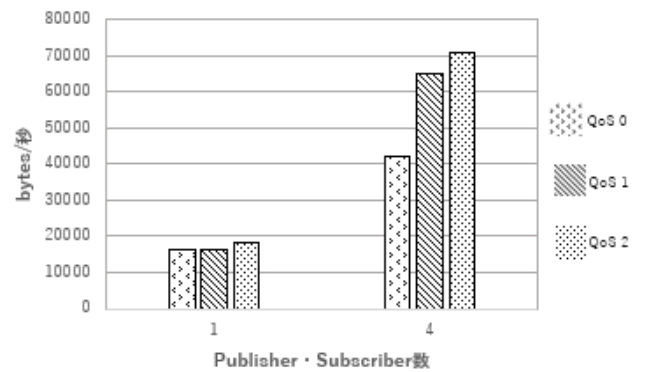


図 7 MQTTS で 2000 バイトのデータを伝送したときのスループット

5. 考察

5.1 暗号化の有無に注目したスループット比較

MQTT より MQTTS の方が高いスループットになった。これは MQTTS が MQTT に比べて伝送データ量が大幅に増加しているにもかかわらず、伝送に要する時間は少しの増加であったからである。1 つのメッセージを含むパケットに注目すると、暗号化したときのデータは暗号化していないときと比べ、29 バイトしか増加しておらず、伝送時間もほとんど変わらなかった。一方、全ての伝送データの packets の数は MQTTS の方が 10 パケット多かった。これはメッセージ長を含むパケットを伝送する前に TLS のハンドシェイクプロトコルにより共有鍵の生成や証明書の交換を行うためである。これらの伝送に要した時間は 0.04 秒であるのに対し、伝送データ量は 6539 バイト増えている。このことから MQTT より MQTTS の方が単位時間当たりの伝送量が大きく、高いスループットになったと考えられる。

5.2 QoS の違いに注目したスループット比較

QoS の違いによるスループットに注目すると、プロトコルやメッセージ長、Publisher・Subscriber の全ての条件において QoS 0, QoS 1, QoS 2 と順番に高いスループットになった。これは QoS レベルが高くなるにつれ、全ての伝送データ量が増えているにもかかわらず、伝送に要する時間は少しの増加であったからである。1 つのメッセージが Publisher から Subscriber に到着するまでの通信において、QoS 0 の場合の通信回数は 4 回であった。QoS 1 の場合、QoS 0 の通信の時と比べ、パケット数は 3 パケット多くなっていた。それらの多くなっているパケットの処理に要した合計時間は 0.04 秒であった。最後に、QoS 2 の場合は QoS 1 と比べてパケット数は 6 パケット多くなっていた。これらの

伝送に要した合計時間は 0.04 秒であり、QoS 1 は QoS 0 に比べ伝送データ量が 1470 バイト増加し、QoS 2 は QoS 1 に比べ、伝送データ量が 2860 バイト増加していた。このことから QoS 0, QoS 1, QoS 2 の順番にスループットが高くなったと考えられる。

5.3 メッセージ長の違いによるスループット比較

メッセージ長が 2000 バイトでの通信の方が 2 バイトのときよりも高いスループットになった。理由はメッセージ長が 2000 バイトでの通信の方が伝送データ量は多いにも関わらず、伝送に要する時間は少ししか増加しなかったからである。2000 バイトの通信では TCP フラグメンテーションにより、パケットが分割して送られるため、2 バイトの通信と比べて 1 パケットだけ多くの通信を送る必要がある。具体的には 2000 バイトのメッセージ長を送信した場合、MSS (Maximum Segment Size) の 1460 バイトと 550 バイトに分けられる。メッセージ長が 2000 バイトの通信では 1 回多くのパケットを送るが、その通信に要した時間は 0.0003 秒であり、全ての伝送データ量は 41551 バイト増えている。このことから MQTT と MQTTS の両方において、2000 バイトの通信の方が 2 バイトのそれよりも高いスループットになったと考えられる。

6. まとめ

MQTT と MQTTS のスループットを比較した結果、暗号化されている MQTTS の方が高いスループットとなった。次にメッセージ長が 2 バイトと 2000 バイトの違いに注目した結果、MQTT と MQTTS のどちらにおいても大きいメッセージ長の方が高いスループットとなった。最後に QoS の違いによるスループットに注目した結果、メッセージ長、Publisher・Subscriber 数の全ての条件において QoS 0, QoS 1, QoS 2 の順にスループットが高くなった。理由はいずれも伝送データ量の増加に対し、伝送に要する時間の増加が小さかったからである。

ただし、伝送するデータにのみ注目すると、伝送時間がわずかではあるが増加する MQTTS のスループットは MQTT に比べて低い。

参考文献

- [1] 粕谷貴司, 近藤正芳, 茂手木直也ほか: スマートシティのための MQTT プラットフォームの検証, 情報科学技術フォーラム講演論文集, Vol.13, No.4, pp.1-6 (2014).
- [2] 木村一統, 新井イスマイル, 藤川和利: IoT デバイス向けアプリケーション層プロトコルの性能比較計画, 研究報告モバイルコンピューティングとパーベシブシステム (MBL), Vol.2016-MBL-81, No.4, pp.1-4 (2016).
- [3] 木村一統, 新井イスマイル, 藤川和利: MQTT-SN の実装評価, マルチメディア, 分散協調とモバイルシンポジウム 2017 論文集, Vol.2017, pp.949-954 (2017).