Vol.2016-MBL-81 No.4
Vol.2016-ITS-67 No.4
2016/12/7

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

木村 一統[1,a)]　新井 イスマイル[2]　藤川 和利[2]

概要：IoT デバイスはその特徴によって様々な通信環境下での使用が考えられており，通信環境と使用するネットワークプロトコルの組み合わせにより，その通信特性が大きく異る．このため，通信環境に合わせたネットワークプロトコルの選定が重要となる．IoT デバイス向けのアプリケーション層プロトコルに MQTT(MQ Telemetry Transport) と MQTT-SN(MQTT for Sensor Network) がある．本研究ではそれら 2 つのプロトコルについて，通信環境や QoS を変化させた場合に，スループットやデバイスの消費電力がどの程度の影響を受けるのかについて性能比較を行う。

キーワード：プロトコル検証，センサーネットワーク，QoS，IoT，MQTT，MQTT-SN

1. はじめに

近年，小型デバイスの性能向上や省電力化，MVNO(Mobidle Virtual Network Operator) の登場により狭帯域であるが低価格でインターネット通信が行えるようになった．これにより，様々なモノに通信デバイスを組み込み，インターネットに接続しようとする試み，IoT(Internet of Things) が盛んに唱えられている．IoT ではモノに組み込まれたデバイスにセンサーを取り付け，センシングをすることにより様々なデータを取得する．

現在 IoT 機器が通信を行う場合，アプリケーション層のプロトコルには HTTP(Hypertext Transfer Protocol) が多く使われている．しかし，今後の IoT 機器の更なる増加に伴うスケーリングの問題，IoT 機器に求められる省電力性を考えた場合，HTTP のヘッダ長の大きさ，複雑性が問題となる可能性がある．よって，HTTP よりも軽量であり，単純なプロトコルが期待される．また，IoT に使用されるデバイスは組み込み対象の持つ特徴に依る，様々な通信環境下での使用が考えられており，通信環境と使用するネットワークプロトコルの組み合わせにより, その通信特性が大きく異なる．このため，通信環境に合わせたネットワークプロトコルを選定することで，スループットやデバイスの電力効率を向上させることが可能であると考えられる．

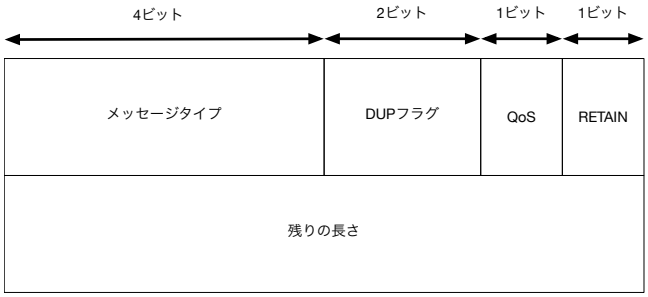

図 1　MQTT ヘッダ

本研究では，IoT デバイス向けのアプリケーション層プロトコルとして注目を集めている MQTT[1] と MQTT-SN[2]，また従来より主なアプリケーション層プロトコルとして用いられてきた HTTP について，通信時のスループットや消費電力，サーバに掛かる負荷，既存の暗号アルゴリズムを適用した時の性能低下の度合いを比較する．本稿では計画までを行ったのでその内容について報告する．

2. 既存技術

2.1 Pub/Sub モデル

MQTT および MQTT-SN はその通信モデルに Pub/Sub モデルを採用している．Pub/Sub モデルでは，データの受信者と送信者が直接通信するサーバ・クライアントモデルとは異なり，データの送信元 (パブリッシャ・クライアント，以下パブリッシャ) とデータの受信先 (サブスクライバ・クライアント，以下サブスクライバ) との間に，ブロー

[1] 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
[2] 奈良先端科学技術大学院大学 総合情報基盤センター
Information Initiative Center, Nara Institute of Science and Technology
[a)] kimura.itto.kd3@is.naist.jp

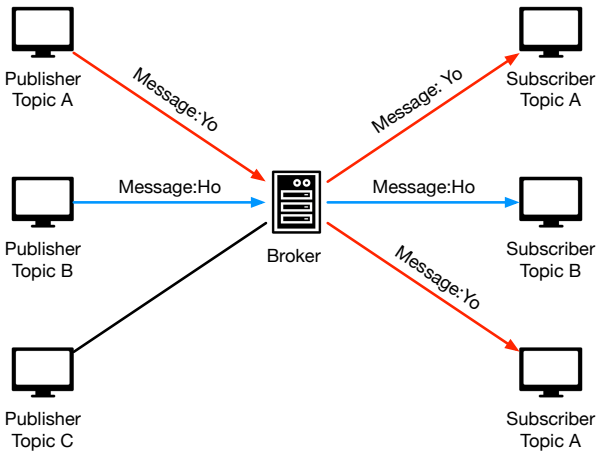

図 2　MQTT アーキテクチャ

表 1　MQTT と MQTT-SN における QoS

QoS	通信方式
QoS 0	最高 1 回届ける
QoS 1	最低 1 回届ける
QoS 2	正確に 1 回届ける

カと呼ばれるサーバを介して通信を行う (以下，パブリッシャとサブスクライバの両方を表す場合にはクライアントと呼ぶ). MQTT，MQTT-SN では通常の Pub/Sub モデルの通信に加え，ブローカがメッセージを保持しておく機能を持つ．これによりパブリッシャがブローカに対して送信したメッセージを，後からブローカに接続したサブスクライバが受信できるという利点もある．

2.2　MQTT

MQTT は 1999 年に IBM により設計されたアプリケーション層のプロトコル [3] であり，トランスポート層のプロトコルには TCP を用いる．プロトコルヘッダは図 1 のようになっており，最小のヘッダ長が 2 オクテットと小さいことが特徴である．また，帯域幅が狭いネットワークやレイテンシの大きいネットワークといった信頼性の低いネットワーク上での使用を目的としている．MQTT では，クライアントとブローカ間において，MQTT による通信が行われる (図 2)。

MQTT では表 1 に示す QoS(Quality of Service) が設定でき，この設定により到達保証の度合いが変化する．

2.3　MQTT-SN

MQTT-SN は，MQTT とほぼ同等の機能を有しているが，トランスポート層のプロトコルには UDP を用いる．プロトコルヘッダは図 3 のようになっており，MQTT と同様に最小のヘッダ長が 2 オクテットと小さいことが特徴である．それに加え，MQTT-SN では，センサー等がスリープ状態に入った場合の動作も定められており，MQTT よ

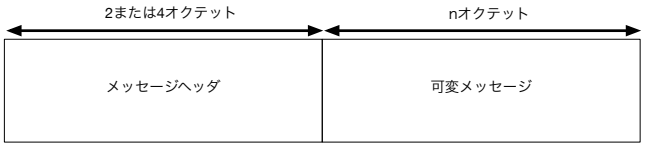

図 3　MQTT-SN ヘッダ

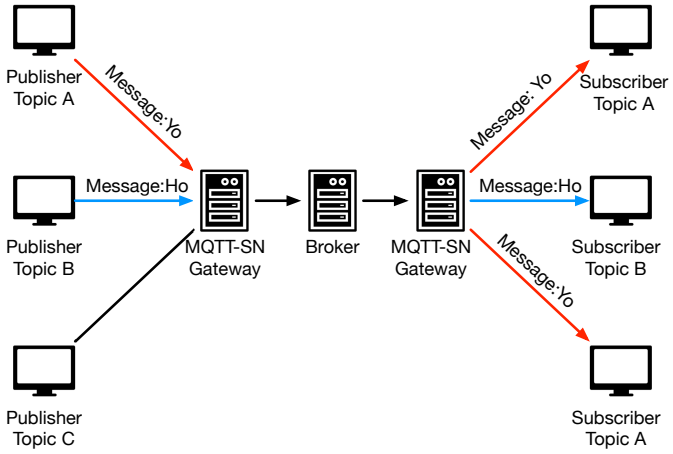

図 4　MQTT-SN アーキテクチャ ゲートウェイ分離型

りも更に，省電力なセンサーデバイスへの適用に特化したプロトコルとなっている．MQTT-SN では，クライアントとブローカの他に，ゲートウェイを設置し，クライアントは MQTT 通信と MQTT-SN 通信を変換するためのゲートウェイを間に挟み，ブローカと通信を行う．また，MQTT-SN ネットワークのアーキテクチャは，ブローカとゲートウェイを一体とするか否かによってゲートウェイ分離型とゲートウェイ一体型の 2 種類に分けられる．ゲートウェイ分離型 (図 4) では，クライアントとゲートウェイ間にて MQTT-SN による通信を行い，ゲートウェイとブローカ間では MQTT による通信を行う．一方，ゲートウェイ一体型 (図 5) では，ゲートウェイとブローカ間の MQTT による通信は行われない．

MQTT-SN でも MQTT と同様，表 1 に示す QoS が設定できる．

本研究では，純粋に MQTT と MQTT-SN の比較を行う目的から，MQTT による通信が行われないゲートウェイ一体型を採用する．

3.　関連研究と課題

MQTT を IoT デバイスへ適用した先行研究として粕谷ら [4] の研究が挙げられる．この研究では建築設備へのセンサーデバイスの適用を目的として，送信メッセージを 10,000 件/分から 1,000,000 件/分まで順次増加させた場合や，QoS の値を変化させた場合のブローカへの負荷検証を行っている．ブローカを動作させるマシンの構築に，Cloudn FLAT[5] を用いて実験を行っているが，もっとも

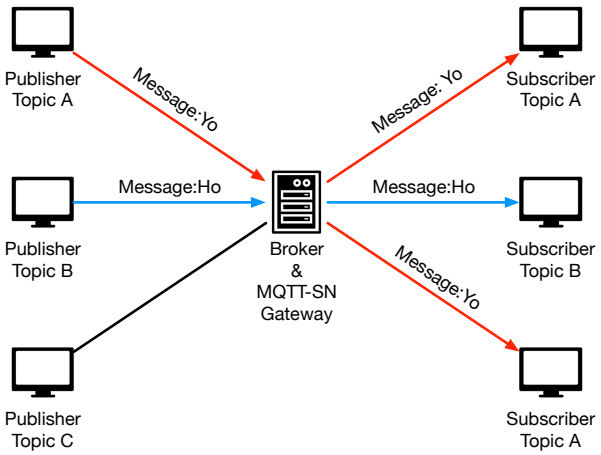

図 5　MQTT-SN アーキテクチャ ゲートウェイ一体型

廉価なプラン (CPU 0.25 コア: 2GHz 程度を 4 台の仮想サーバで共有，メモリ 0.5GB) においても平均 150,000 件/分のメッセージが送受信でき，実用には十分としている．

Shinho ら [6] の研究では，有線接続されたネットワークと無線接続されたネットワークにおいて，QoS の値を変化させた場合のパブリッシャ・サブスクライバ間のネットワーク遅延，パケット到達数についてそれぞれ検証を行っている．この研究では，有線接続，無線接続共に QoS の値が上がるにつれ，パブリッシャ・サブスクライバ間の遅延は大きくなる結果が出ている．また，パケットの到達数に関しては，QoS の値が上がるにつれ高くなり，QoS 2 の場合がもっとも損失率が高くなる結果が出ている．

MQTT 通信を行った場合のデバイスの消費電力に関する先行研究として [7] が挙げられる．この研究では Android 端末上で MQTT 通信を行い，その時の消費電力，時間あたりのメッセージ送受信数を 3G 回線および Wi-Fi 回線において計測するということを QoS の各値について行っている．この研究では，3G 回線よりも Wi-Fi 回線の方が消費電力が小さくなるという結果が出ている．

MQTT と HTTP の暗号化を比較した先行研究として [8] が挙げられる．MQTT に TLS(Transport Layer Security) を適用し [7] と同様の観点から評価を行い，また HTTPS(Hypertext Transfer Protocol Secure) による通信と比較を行っている．この研究では，ほどんどの場合において，HTTPS よりも MQTT の方が消費電力が小さく，時間あたりの受信メッセージ数も MQTT の方が多いという結果が出ている．

以上のように，MQTT については，上記の通りネットワークの負荷検証や消費電力の評価がされている．HTTP は現在，IoT デバイスの通信や Web 通信で最も多く使用されているプロトコルであり，既に多くの研究で評価がされている．

しかし，MQTT-SN については未だ定量的な評価が行われておらず，通信プロトコルに MQTT-SN を採用した例も数少ない．

1.4 で述べた通り，MQTT-SN はよりセンサーデバイスへの適用に特化したプロトコルである．このため，センサーデバイスへの適用を行う場合，MQTT 以上に効率的な通信が行える可能性がある．以上より，通信状況やデバイスの性能によって，MQTT，MQTT-SN，HTTP のいずれのプロトコルを用いるかを適切に選択することができれば，より効率的な通信が可能となると考えられる．

4. 検証方法

本研究では，MQTT，MQTT-SN，HTTP の 3 つのプロトコルについて，ネットワークの帯域幅と RTT の値を変化させ，この時の通信速度，パケットの到達数，デバイスの消費電力，ブローカの CPU 使用率を測定する．また，MQTT，MQTT-SN については QoS の値を変化させる測定する．

HTTP の評価ではサーバ，クライアントの実装にそれぞれ Python 言語の http.server モジュール，urllib モジュールを使い実装し，サーバとクライアント間のネットワークについて通信状態を測定する．

MQTT と MQTT-SN の通信に用いるブローカは，オープンソースのブローカである Mosquitto[9] を用いる．MQTT クライアントはオープンソースのクライアントライブラリ Paho[10] を用いて実装する．MQTT-SN クライアントは汎用的な実装・ライブラリが存在していないため自ら実装する．MQTT と MQTT-SN の評価では，パブリッシャとブローカ間，サブスクライバとブローカ間のネットワークについてそれぞれ通信状態を測定する．ネットワークの帯域幅と RTT の値の設定には Dummynet[11] を用いる．また，通信の暗号化では，MQTT に TLS，MQTT-SN に DTLS を用いて，クライアントとブローカ間の通信を暗号化する．

MQTT-SN はトランスポート層に UDP を用いているため，TCP を用いている MQTT と比較してトランスポート層でのオーバヘッドが小さくなることが予想される．このため，センサーネットワーク等の通信状態が不安定になりやすいネットワークにおいて，MQTT よりも高速にパケットを送受信することが可能と考えられる．また，MQTT-SN はクライアントデバイスがスリープ状態になることも考慮しているため，デバイスのスリープ時間が長いほど電力効率が向上すると考えられる．

5. おわりに

HTTP，MQTT，MQTT-SN について，スループット，消費電力，ブローカにかかる負荷，暗号化した場合の性能低下について検証を行う計画である．先行研究の結果から，HTTP よりも MQTT，MQTT-SN の方が，より IoT デバイスに適用し易いという結果が出ると予測できる．また，

センサーデバイスが頻繁にスリープ状態に入る場合や TCP コネクションによるオーバヘッドが無視できないような環境では MQTT よりも MQTT-SN の方が適用し易いという結果が出ると予測できる.

今後の課題としては，MQTT-SN の評価を行うにあたって，クライアントとゲートウェイを作成する必要があるため，作成のためのライブラリを実装することが必要となる.

謝辞 本研究の一部は，JSPS 科研費 16K00147 の助成によるものである。

参考文献

[1] The MQTT protocol, http://mqtt.org，2016 年 11 月 13 日アクセス
[2] MQTT for Sensor Network (MQTT-SN) Protocol Specification, http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf，2016 年 11 月 13 日アクセス
[3] MQTT，IBM MessageSight のご紹介 - IBM MessageSight for Developers を使って MQTT を体験する，https://www.ibm.com/developerworks/jp/websphere/library/connectivity/ms_mqttintro/，2016 年 11 月 13 日アクセス
[4] 粕谷貴司，近藤正芳，茂手木直也，松岡康友，矢野雅，秋山貴紀，境野哲，貞田洋明，堀越崇，畠山英之. 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 13(4)，1-6，2014-08-19
[5] パブリッククラウドサービスのクラウド・エヌ，http://www.ntt.com/business/services/cloud/iaas/cloudn.html，2016 年 11 月 13 日アクセス
[6] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, Hongtaek Ju, Dept. of Computer Engineering Keimyung University Daegu, Republic of Korea. Correlation Analysis of MQTT Loss and Delay According to QoS Level, The International Conference on Information Networking 2013 (ICOIN)，714-717，2013
[7] Power Profiling: MQTT on Android，http://stephendnicholas.com/posts/power-profiling-mqtt-on-android，2016 年 11 月 13 日アクセス
[8] Power Profiling: HTTPS Long Polling vs. MQTT with SSL, on Android，http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https，2016 年 11 月 13 日アクセス
[9] Mosquitto，https://mosquitto.org/，2016 年 11 月 13 日アクセス
[10] Paho，https://eclipse.org/paho/，2016 年 11 月 13 日アクセス
[11] Dummynet，http://info.iet.unipi.it/~luigi/dummynet/，2016 年 11 月 13 日アクセス