# 優先度を考慮した送信制御が可能な P-MQTT の開発と評価

内山仁 [1]　　峰野博史 [2]

**概要：** Internet of Things（IoT）は農業，環境モニタリング，セキュリティ監視など広い分野で活用され近年さらに期待が高まりつつある．全世界の M2M 接続の増加傾向から IoT の成長は見て取ることができ，2022 年の M2M 接続は 2017 年の約 2.4 倍まで増加すると予想されている．それに対して，IoT 向けのモバイルネットワークサービスも登場しているが，通信品質が保証されていない場合がある．IoT 向けの通信プロトコルとしては小容量かつ高頻度の通信に適した MQTT プロトコルの使用が注目されている．しかし，MQTT ではメッセージの優先順位に関する機能は保証されておらず，データの特性と関係なくデータ送信が行われる．そのため，通信帯域や速度の品質が低いネットワークを用いたアプリケーションでは，データ利活用の際に重要性の低いデータ送信で帯域を圧迫し，重要なデータ送信の遅延や信頼性の低下が予想される．そこで本研究では，コンテンツごとに定められた優先度の情報をもとに送信順制御が可能な MQTT として P-MQTT（Priority based MQTT）を提案する．ここで優先度は IoT デバイスが生成するデータの特性（遅延耐性など）に応じて決定される．提案手法を詳細設計，実装を行い優先度の異なるデータが混在する環境で評価したところ，通信速度に制限のあるネットワークで P-MQTT は効果を発揮し，優先度の高いデータ送信の遅延時間を 6%程度削減できた．また，通信速度が低くなることでより効果的であることがわかった．

# Implementation and Evaluation P-MQTT for Priority Control

JIN UCHIYAMA [1]　HIROSHI MINENO [2]

## 1. 論文フォーマットについて

Internet of Things（IoT）[1]は農業，食品加工業，環境モニタリング，セキュリティ監視など広い分野で活用され[2]近年さらに期待が高まりつつある．全世界の M2M 接続の増加傾向から IoT 成長は見て取ることができ，2022 年の M2M 接続は 2017 年の約 2.4 倍まで増加すると予想されている[3]．また，IoT デバイスによって収集されたデータを分析，活用する IoT アプリケーションも同様に様々な分野において増加傾向にある．モバイルネットワーク上の通信サービスはスマートフォン向けが主であったが，近年は IoT 向けサービス[4]も登場し，サービス形態が多様化しつつある．これらの IoT 向けサービスは帯域保証されていない場合があり，データ通信に大幅な遅延や欠落の生じる可能性がある．そのため，遠隔地のデータをサーバに収集し，リアルタイムに分析するようなシステムでは，アップリンクのデータ欠落や遅延によるアプリケーションの信頼性や可用性の低下が懸念される．

近年，IoT 向けの通信手法として様々な手法が提案されている．特に通信環境が不安定な状況で用いられる IoT 向け通信手法の一つに MQTT がある．MQTT は少量かつ高頻度のデータ通信に適した通信手法であり，IoT システム運用のための通信に広く用いられる．しかし，MQTT はメッセージの優先順位が考慮されないという欠点がある[5]．そのため遅延耐性[6]や利活用目的の異なるデータが混在した場合でもデータの特性は加味されず，データ利活用の際に重要性の低いデータ送信で帯域を圧迫し，重要なデータ送信の遅延や信頼性の低下が予想される．そこで本研究ではデータごとに異なる遅延耐性に着目し，データ利活用を行うアプリケーションの要求に応じた送信順制御が可能な MQTT プロトコルを提案する．

以下，本論文の構成を述べる．第 2 章で関連研究について整理し，第 3 章で提案手法である優先度を考慮した送信順制御機能つき MQTT について述べる．第 4 章で評価結果，第 5 章でまとめと今後の課題を述べる．

## 2. 関連研究

### 2.1 IoT 向け通信手法

IoT はインターネットを介した相互通信，他のデバイスやサービスと通信することを可能にする識別機能，検知機能，ネットワーキング機能，処理機能などを含んだ考え方ととらえることができる．IoT では，システムを管理するために IoT 機器が別の機器に命令を配信する場合があるため，異なるデバイス間のメッセージ送信が重要となる[7]．これまで，デバイス間通信を維持するために，HTTP がデータの遠隔監視および分析に使用されてきた．しかし，HTTP はヘッダサイズが大きいため通信のオーバーヘッドが大きく，システムのデータ送信に用いる帯域幅を圧迫するなどの問題があった．そのため，狭い帯域幅や制限のあ

1 静岡大学大学院総合科学技術研究科
Graduate School of Integrated Science and Technology, Shizuoka University
2 静岡大学学術院情報学領域
College of Informatics, Academic Institute, Shizuoka University

ⓒ2019 Information Processing Society of Japan

るネットワークを有効利用するために MQTT[8]，AMQP[9]，CoAP[10]などのプロトコルが IoT における通信手段として用いられ，中でも軽量かつ信頼性の高い MQTT が最も広く使用される．

MQTT は IoT 向けの通信手法の 1 つである．MQTT プロトコルは制御用のヘッダが小さく，メッセージに定められた QoS レベルに応じた到達保証機能が存在する．軽量かつ信頼性の高い MQTT プロトコルは，不安定なネットワークに接続されている機器などの制約のある環境に適しており[11]，多くの IoT ガジェットやインスタントメッセージ配信システムの一部として利用されてきた．しかし MQTT はメッセージの優先順位という機能をサポートしていない．例えば火災警報システムから収集されたデータなどは急を要するデータであり，温度や圧力センサから得られたデータより早く送信処理を行う必要がある．緊急のメッセージ以外であっても，データの利用までに時間的制約が存在するデータは他のデータより優先的に送信するのが望ましい．このような事例からメッセージごとの優先順位の考慮が必要であるといえる．優先度を考慮した改良版 MQTT としては p-MQTT[12]がすでに提案されている．しかし，p-MQTT には優先度の段階に制限があり，多様な環境が想定される IoT アプリケーションへの十分な対応ができているとはいえない．

### 2.2 Message Queuing Telemetry Transport（MQTT）[8]

MQTT は 1999 年に IBM が標準化した Publish/Subscribe メッセージングモデルのプロトコルであり，サーバからデータの通信を開始するプッシュ型プロトコルである[7]．MQTT は，遅延が大きく狭帯域のネットワークにおいて，正確なデータ送信を行うことを目的としている．MQTT を用いる利点としては，接続の一時的な切断に強い点や帯域の利用効率が良く軽量である点が挙げられ，信頼性や通信速度の低いネットワークでの利用に有効である．

MQTT のアーキテクチャを図 1 に示す．MQTT はメッセージを送信する Publisher と受信する Subscriber，それらを仲介する Broker の 3 つの要素で構成されている．MQTT で

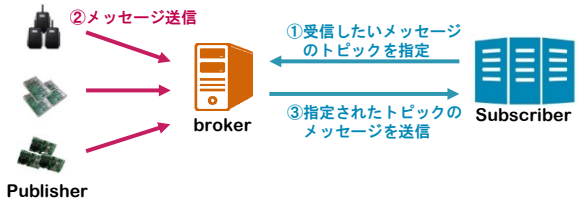


図 1　MQTT アーキテクチャ概要

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| メッセージタイプ | | | | DUP | QoSレベル | | RETAIN |
| メッセージ長 | | | | | | | |
| 可変長ヘッダ(一部メッセージのみ) | | | | | | | |
| メッセージペイロード(一部メッセージのみ) | | | | | | | |

図 2　MQTT のメッセージフォーマット

は，トピックと呼ばれるメッセージに付属する件名を表す文字列を用いて，Subscriber が受信したいメッセージを Broker に伝達する．ここで，トピックは 1 つのメッセージに対して複数指定することが可能である．Broker は Publisher からメッセージを受信した後，トピックの情報からメッセージを要求する Subscriber へ送信を行う．送信先の存在しないメッセージに関してはトピックごとに最新のメッセージのみ保持することが可能であり，新たに送信先となる Subscriber からの接続があった際に送信される．

MQTT のメッセージフォーマットを図 2 に示す．MQTT のメッセージは，全てのメッセージに存在する制御用の固定長ヘッダ，いくつかのメッセージに存在する可変長ヘッダと可変長のメッセージペイロードで構成されている．固定長ヘッダはメッセージのタイプを表すフィールド，QoS レベルを含めた各種フラグ，残りのメッセージ長を示すフィールドが含まれている．固定長ヘッダは 2bytes という大きさで軽量でありオーバーヘッドが小さい．また，3 段階の QoS レベルに基づいた到達保証が可能でありメッセージ送信の正確さも備えている．しかし，メッセージの送信順が保証されず，優先順位の機能が存在しない．そのためメッセージの優先度を考慮してメッセージの送信順を変更することはできない．緊急のメッセージが予想される環境や送信可能なメッセージの量に制限がある環境では，メッセージの特性に応じた送信順制御が望まれる．そのため，MQTT においても，優先度を考慮した通信制御を検討する必要がある．

## 3. 提案手法

### 3.1 提案手法の概要

本稿では MQTT の機能に加えて，優先度に基づいた送信

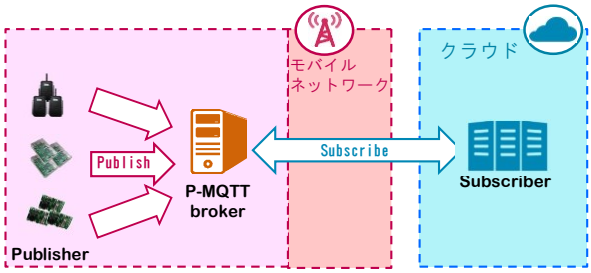


図 3　リモートモニタリング IoT システムにおける P-MQTT 適用例

ⓒ2019 Information Processing Society of Japan

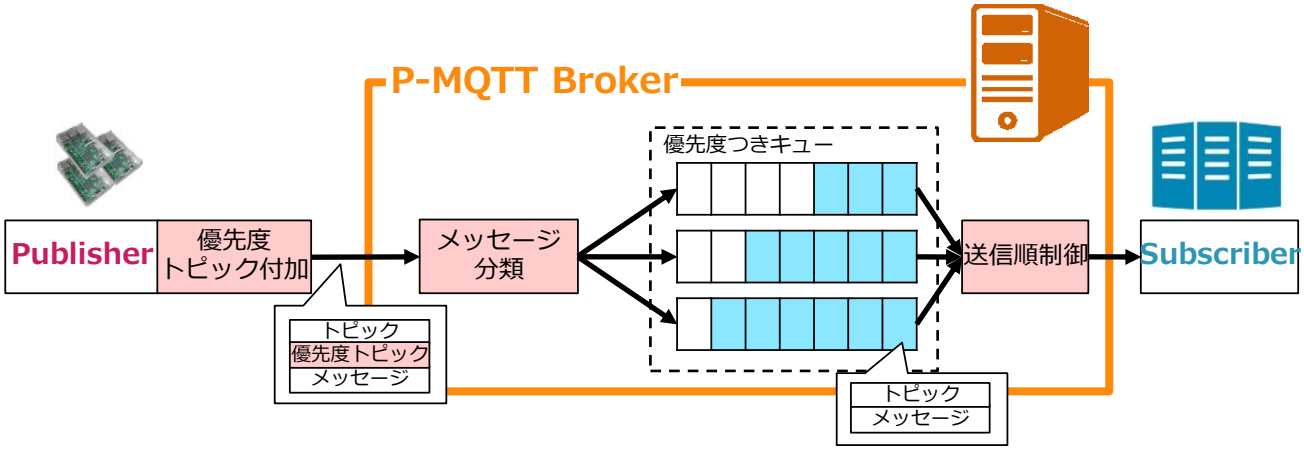


図 4　P-MQTT 優先度制御フロー

順制御機能をもつ P-MQTT（Priority based MQTT）を提案する．P-MQTT は従来の MQTT と同様に Publisher と Subscriber の 2 種類のクライアントと Broker サーバの 3 要素で構成する．P-MQTT では MQTT に存在するトピックに優先度情報を含めることで publisher の送信データの優先度を Broker サーバへ伝達する． Broker サーバは優先度の情報をもとに送信順の制御を行い，優先度が高く設定されたメッセージから順に送信処理を行う．メッセージの優先度を考慮した送信処理を行うことで急を要するメッセージへの対応や，データ利用目的に応じた帯域利用が可能となる．

P-MQTT ではモバイルネットワークと IoT デバイスを用いて遠隔地の情報をサーバへアップロード，分析するリモートモニタリング IoT システム[13]への運用を想定している．リモートモニタリング IoT システムの例として農場の環境データのセンシングと分析を行う施設園芸環境向けシステム[14]などが存在する．施設園芸環境向けシステムでは温湿度や光量などのセンサデータやセンサ端末の状態を示すデータなどが送信対象として挙げられる．リモートモニタリング IoT システムに P-MQTT を用いる場合のシステム全体像を図 3 に示す．Broker サーバを Publisher と同一のネットワークに配置し，IoT デバイスが取得したデータのアップロードを制御することで低品質なネットワークを用いたシステムの性能向上を実現する．3.2 節では，P-MQTT で行う優先度制御について詳細を述べる．

### 3.2 P-MQTT における優先度制御

P-MQTT の優先度制御のフローを図 4 に示す．P-MQTT では Publisher で行われるトピックの機能を利用した優先度情報の付加に加えて，Broker で行われる優先度情報に基づいた受信メッセージの分類及び優先度つきキューへの格納処理，送信順制御の 3 つの段階に分けて行う．

優先度情報の付加では Publisher が優先度トピックをトピックの末尾に付け加えることで行う．この処理は MQTT の機能を利用しているため制御用のヘッダを変更する必要がない．そのため P-MQTT を利用してアプリケーションを

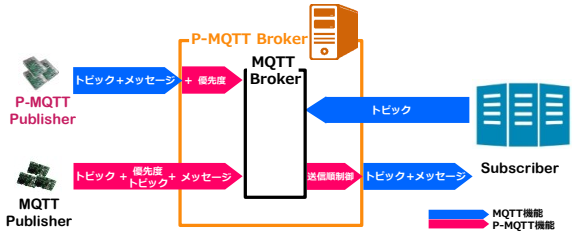


図 5　P-MQTT 全体アーキテクチャ

作成する場合には，トピックの変更のみで P-MQTT の機能を利用できる．また，MQTT 用の Publisher と P-MQTT 用の Publisher が混在している場合でも運用可能である．優先度トピックは，優先度トピック識別用文字列と優先度の段階を示す数字を組み合わせた文字列である．優先度の段階に制限がなく利用者が用途に合わせて設定可能とした．

受信メッセージの分類ではメッセージを受信した Broker が優先度トピックを含めたトピックに関してフィルタリングを行い，受信したメッセージと優先度の情報を分離する．P-MQTT Broker は Publisher からメッセージを受信後，MQTT 同様にメッセージタイプの取得や QoS レベルの取得を行う．また，トピックから送信先である Subscriber を決定する．送信先の決定後，Publisher から受信したメッセージのうち送信先が存在するメッセージにのみ優先度トピックに基づいた分類とキューへの格納を行う．トピックをもとにしたメッセージの分類後に優先度制御を行うことで制御対象のメッセージを削減可能であり，優先度制御のオーバーヘッドや優先度付きキューのサイズも小さくできる．P-MQTT 用の Publisher と MQTT の Publisher が混在している場合に関しては優先度トピックの解析の際に対応を行う．優先度トピックが付加されているメッセージは，優先度トピックの末尾の数値に応じた優先度として判断し，優先度トピックを分離した後に対応したキューへ格納を行う．MQTT の Publisher から受信したメッセージに関しては優先度トピックが付加されていないことから，MQTT からの

ⓒ2019 Information Processing Society of Japan

メッセージを受信したと判断し，Broker に事前に設定した特定の優先度を設定しキューへの格納を行う．P-MQTT では優先度トピックがオーバーヘッドとなることが懸念される．しかし，Broker での優先度キューへのメッセージ格納時に優先度トピックの取り外しが行われているため，通信品質が低くオーバーヘッドの影響が懸念される，Broker と Subscriber の通信には影響を与えない．

送信順制御では優先度つきキューに格納されたメッセージを取り出し送信を行う処理である．優先度の高いキューから順にメッセージと送信先などの情報を取り出し送信を行うことで送信順の制御を行う．

## 4. 実装・評価実験

### 4.1 実験環境

P-MQTT について優先度制御機能の有効性を評価するために，実機上に実装を行い実験と評価を行った．Publisher, Subscriber 及び Broker 用の端末には RaspberryPi3 を使用した．使用端末の OS は全て Raspbian9.6 とした．P-MQTT 用 Broker の実装及び比較対象である MQTT 用 Broker には Eclipse Mosquitto[15]を使用した．Mosquitto はバージョン 3.1 以降の MQTT 機能を提供するオープンソースの Broker である．実験に使用する Publisher と Subscriber は，Eclipse Paho[16]が提供するオープンソースライブラリを用いて python で実装を行った．

実験におけるネットワーク構成を図 6 に示す．実験では Broker サーバと Publisher 及び Subscriber の各クライアントをそれぞれ別の端末で動作させ，同一の有線ネットワーク上に配置した．提案手法で述べたように今回は Broker サーバと Subscriber が遠隔地にあることを想定している．そのため，TC コマンドを用いて通信を制限し，条件の違うネットワーク環境を再現した．Publisher と Broker 間の通信環境であるネットワーク A は有線で繋がれたローカルネットワーク，Broker と Subscriber 間の通信環境であるネットワーク B はモバイル回線を用いたアップリンクを想定し実験用のネットワーク環境とした．Publisher は同一の端末上にマルチスレッドで動作させ，Broker サーバに対して同時に複数の接続を確立し，メッセージを送信する．本実験ではメッセージが Publisher から送信されてからネットワーク A, Broker，ネットワーク B を介して Subscriber で受信されるまでの時間を評価指標とし，MQTT と P-MQTT の性能を比較した．

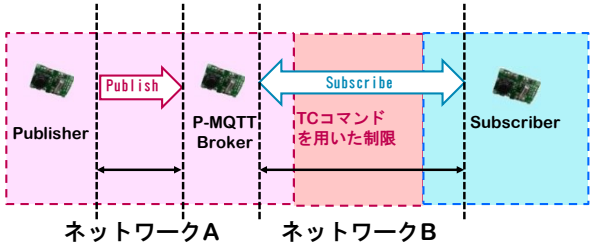


図 6 実験ネットワーク構成

表 1 基礎評価実験諸元

| 項目 | 値 |
|---|---|
| Publisher 数 | 200(各優先度 100 ずつ) |
| メッセージサイズ | 120Byte(全優先度共通) |
| 送信回数 | 1(全優先度共通) |
| 優先度レベル | 2 種(高/低) |

表 2 基礎評価実験通信環境一覧

| | ネットワーク A | | ネットワーク B | |
|---|---|---|---|---|
| | 通信速度 | 伝送遅延 | 通信速度 | 伝送遅延 |
| 条件 1 | 94Mbps | 0.65ms | 500kbps | 60ms |
| 条件 2 | 94Mbps | 0.65ms | 100kbps | 100ms |
| 条件 3 | 94Mbps | 0.65ms | 50kbps | 100ms |
| 条件 4 | 94Mbps | 0.65ms | 30kbps | 100ms |

### 4.2 基礎実験・評価

P-MQTT の優先度制御の効果を確認するための実験を行った．実験では優先度の異なるメッセージの送信を同時に行い，優先度に応じて送信制御が行われることを確かめる．実験におけるメッセージの条件一覧を表 1 に示す．通信帯域に制限を加え条件の異なる複数の環境で実験を行うことで，データアップロードに用いるネットワークの負荷が異なるとき，P-MQTT 適用の効果にどのような変化があらわれるかを検証する．ネットワーク A は有線接続されたローカルネットワークを想定しているため通信速度が高く，遅延が小さい．ネットワーク B はモバイル回線を用いたアップリンクを想定しているため通信速度が低く，遅延が大きい．本実験では優先度を 2 種とし，他のデータ送信よりも優先して送信されるべきデータが P-MQTT の導入とともに設定された状況を想定する．メッセージのサイズは時間情報といくつかのセンサデータがまとめられたデータを想定し 120byte とした．異なる優先度をもつ Publisher が 100 台ずつ存在し，同時に Broker へのアクセスと 1 回のデータ送信を行う．実験を行ったネットワーク環境の条件については表 2 に示す．評価指標である遅延時間は Publisher 送

ⓒ2019 Information Processing Society of Japan

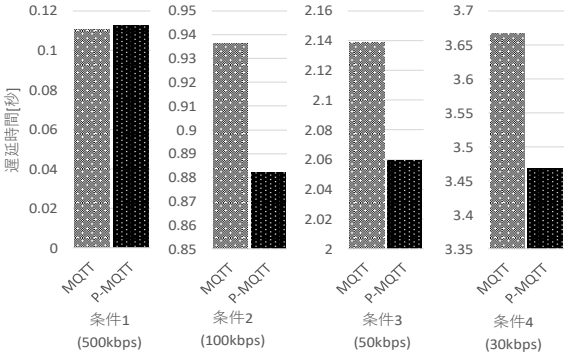


図 7　条件別高優先度データの送信時間比較

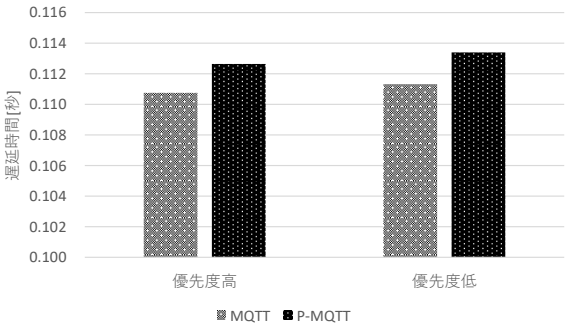


図 8　条件 1(500kbps)での実験結果

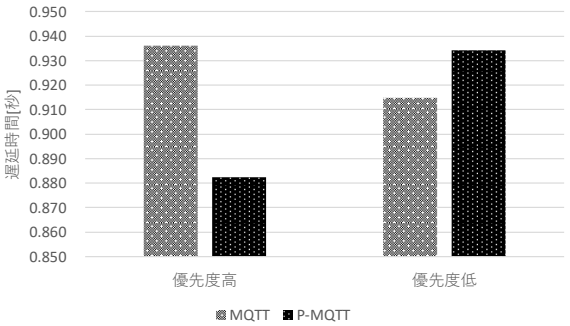


図 9　条件 2(100kbps)での実験結果

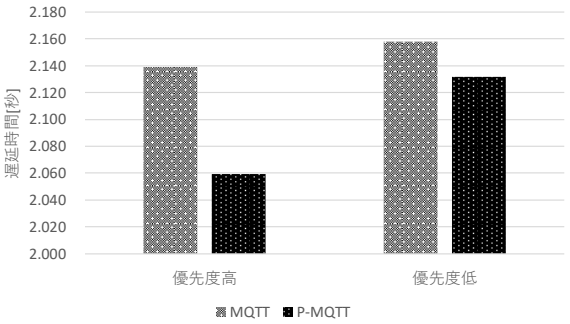


図 10　条件 3(50kbps)での実験結果

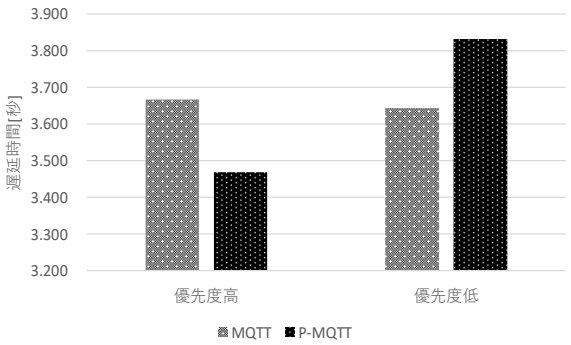


図 11　条件 4(30kbps)での実験結果

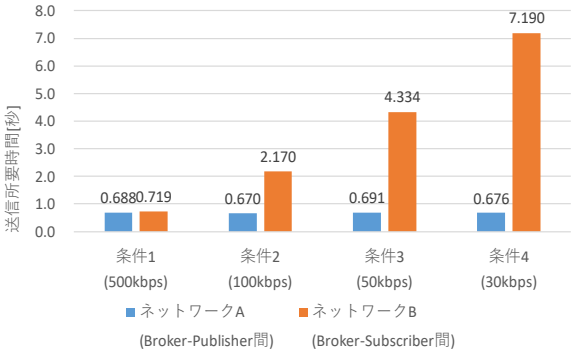


図 12　条件ごとの区間別別送信所要時間の比較

信されたメッセージが Subscriber に届くまでの時間であり，P-MQTT を適用することで優先度が高いメッセージの遅延時間減少を確認できれば優先度制御の効果があるといえる．

条件別の遅延時間について高優先度データの送信時間を図 7 に，条件 1 から条件 4 までの実験結果をそれぞれ図 8，図 9，図 10，図 11 に示す．条件 1 では P-MQTT を適用した場合に高優先度のメッセージ送信における遅延時間は MQTT よりも増加している．また図 8 を見ると，高優先度メッセージの遅延時間も低優先度メッセージの遅延時間と大きな差異はなく，優先度制御の効果があらわれていない．しかし，優先度制御を行うことで増加する遅延時間は 5ms 以下であり，P-MQTT は優先度制御が有効でない状況であっても MQTT と同程度の性能で通信が行えることがわかる．100kbps 以下の狭い帯域を用いた条件 1, 条件 2, 条件 3 の場合では高優先度メッセージ送信の遅延時間は MQTT に比べて低下しており，P-MQTT 適用の効果があるといえる．図 7 での 100kbps 以下の帯域幅での実験結果を見ると，どの結果でも同様に 4%から 6%程度遅延時間が減少している．しかし，帯域幅が狭くなるにつれてメッセージ 1 つあたりの送信時間は長くなっているため，P-MQTT を適用することで削減される遅延時間も長くなっており，P-MQTT が通信速度低下とともにより効果を発揮することがわかる．通信速度と P-MQTT 適用効果の関係性について更に詳細な分析を行った．条件別にネットワーク A でのメ

ⓒ2019 Information Processing Society of Japan

ッセージ送信に要する時間と，ネットワーク B でのメッセージ送信に要する時間の比較を行う．比較結果を図 12 に示す．メッセージ送信に要する時間の指標となる通信所要時間は，最初のメッセージの受信から最後のメッセージである 100 個目のメッセージを受信するまでの時間である．条件 1 では Publisher から Broker へのメッセージ送信に要する時間と Broker から Subscriber へのメッセージ送信に要する時間は 0.03 秒しか差がない．よって，ネットワーク A とネットワーク B はほぼ同数のメッセージを同じ時間内で通信可能であるといえる．Broker では Publisher から受信したメッセージを Subscriber へ送信するが，送信にかかる時間が変わらない場合は Broker にメッセージが蓄積されることはなく Publisher から受信したメッセージは Broker への到着順に Subscriber へ送信が行われることになる．そのため優先度制御の効果はあらわれず，P-MQTT では Publisher のメッセージを受信してから Subscriber へ送信するまでに優先度制御が行われるため，遅延時間がわずかに増加していると考えられる． 条件 1 以外では Publisher から Broker へのメッセージ送信に要する時間に比べて Broker から Subscriber へのメッセージ送信に要する時間が長くなっている．これはネットワーク B が Publisher から受信したメッセージ送信に十分な通信速度でないからである．そのため，Publisher から送信されたメッセージは一度 Broker で待機した後に Subscriber へ送信を行う必要がある．P-MQTT では Broker に保存されているメッセージに関して優先度つきキューを用いた送信順制御を行うため，Broker にメッセージがたくわえる場合に効果があらわれると考えられる．

P-MQTT においてネットワーク A とネットワーク B それぞれの遅延時間について分析を行った．条件 1 での比較結果を図 13 に，条件 2 での比較結果を図 14 に示す． 送信所要時間の結果より，条件 1 での実験では Broker へメッセージが蓄積されない．そのため Broker へ届いたメッセージは届いたタイミングと関係なく常に一定の時間で Subscriber へ届けられる．条件 2 での実験では，Broker1 つのメッセージを Subscriber へ送信するまでに，Publisher から複数のメッセージを受信するため，時間経過とともに Broker へメッセージが蓄積される．そのため Broker への到着が遅れたデータほど早く到着したデータの送信を待つことになり Subscriber に届くまでの時間が長くなる．100kbps の比較結果を示した図 14 からは優先度制御の効果を見て取ることができ，高優先度のデータ送信が同タイミングで到着した低優先度のデータよりも早く送信処理が行われている事がわかる．P-MQTT では Publisher からのメッセージ受信処理に対して Subscriber への送信処理に遅れが発生するような環境で効果を発揮し，高い優先度に設定されたメッセージの送信処理を低い優先度のメッセージの送信処理よりも先に行うことで遅延時間の削減を行うことができる．また Publisher-Broker 間と Broker-Subscriber 間の通信速度の差が大きいほど遅延時間を大きく減少できるといえる．

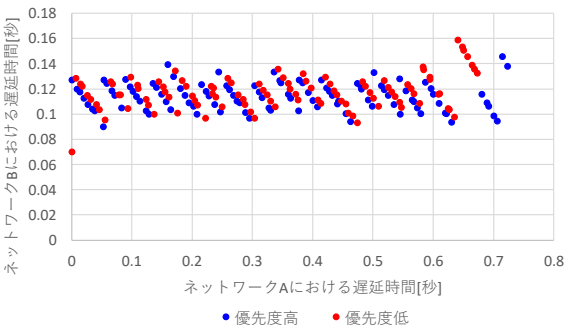


図 13　区間別遅延時間の比較（500kbps）

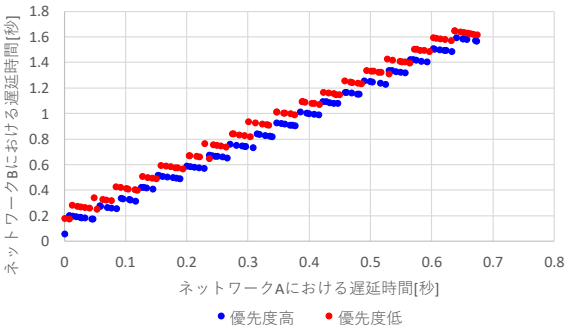


図 14　区間別遅延時間の比較（100kbps）

## 5. おわりに

本研究では，遠隔地の情報を収集し分析するリモートモニタリング IoT システムにおいて，低品質なネットワークを用いた場合でもシステムの信頼性や可用性を向上させるため，送信データの優先度に着目した送信順制御が可能な MQTT プロトコルである P-MQTT の提案をした．更に P-MQTT の評価を行うため MQTT の機能を実現したオープンソースである mosquitto と Eclipse Paho を利用し実装を行った．実装した P-MQTT は優先度に応じて通信処理を行っており，通信速度の低いネットワークにおいて高い優先度のメッセージは遅延時間が MQTT に比べて減少していた．遅延時間は約 6%削減でき，メッセージの送信時間が長くなる低速なネットワークほど P-MQTT は効果的であった．今後は優先度の設定値や送信メッセージの条件を複雑にした場合，mosquitto の同時送信処理数などを変更した場合の性能評価を実施する予定である．


## 謝辞

本研究は科学研究費補助金基盤研究 B（26280028）の支援を受け実施されたものである．


ⓒ2019 Information Processing Society of Japan

## 参考文献


[1] Towards a Definition of the Internet of Things (IoT), IEEE,<http://iot.ieee.org/images/>,2016 files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf> (参照 2018-10-08)

[2] Li Da Xu., et al. “Internet of Things in Industries: A Survey”, IEEE Transactions on Industrial Informatics 10(4):2233-2243,2014.

[3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022. Technical report, Cisco, 2018.

[4] SOLACOM,SORACOM INC,<https://soracom.jp/>(参照 2018-10-08).

[5] Dipa S , Ashwin M,“A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT)”, INTERNATIONAL CONFERENCE ON TELECOMMUNICATION, POWER ANALYSIS AND COMPUTING TECHNIQUES ,2017

[6] Daniela M, Daniele T, Giovanni E,“A Unified Urban Mobile Cloud Computing Offloading Mechanism for Smart Cities.” IEEE Communications Magazine, vol. 55, no. 3, 2017, pp. 30–37.2017.

[7] Jie Lin, Wei Yu, Nan Zhang ; Xinyu Yang ; Hanlin Zhang ; Wei Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” IEEE Internet of Things Journal,no. 99, pp. 1–1, 2017.

[8] MQ Telemetry Transport, http://mqtt.org.

[9] Kramer, Joshua Advanced “Message Queuing Protocol (AMQP)” LinuxJournal 187, Houston, TX, 2009.

[10] Z Shelby, K Hartke, C Bormann,“The constrained application protocol (CoAP)." 2014.

[11] V Lampkin, WT Leong, L Olivera, S Rawat, et al.“Building smarter planet solutions with mqtt and ibm websphere mq telemetry” IBM Redbooks. 2012

[12] Kim, Y., et al. “Message Queue Telemetry Transport Broker with Priority Support for Emergency Events in Internet of Things. “ Sensors and Materials, 30(8), 1715.,2018

[13] Gubbi, et al. “Internet of Things (IoT): A vision, architectural elements, and future directions.” Future Generation Computer Systems 29.7 pp.1645-1660, 2013.

[14] 柴田瞬, 古都 哲生, 峰野 博, “Optical Flow を用いた植物水分ストレス推定手法の検討, ” 第 78 回情報処理学会全国大会, pp.2-335－2-336,2016.

[15] Eclipse Mosquitto,<https://mosquitto.org/>,(参照 2019-4-10).

[16] Eclipse Paho - MQTT and MQTT-SN software,<https://www.eclipse.org/paho/>,(参照 2019-4-10).


ⓒ2019 Information Processing Society of Japan