

M2M ネットワークにおける MQTT を用いた調停機構

丸山 真治[†]立命館大学情報理工学部[†]山本 修平[‡]立命館大学大学院情報理工学研究科[‡]高田 秀志[†]立命館大学情報理工学部[†]

1 はじめに

近年, Machine-to-Machine (M2M) ネットワークへの注目が高まり様々な M2M デバイスが普及している. M2M デバイスは多端末間で情報を共有するプラットフォームで使用されることが多く, 多端末で協調作業を支援する環境も想定される.

そのような協調作業環境では, データの同期や一貫性, 通信遅延, 順序付けなどの調停問題を解決する必要がある. この調停問題を解決する方法として, ネットワークに参加している端末の中から調停役の端末を用意する方法が考えられるが, 従来の調停機構は, 必ずしも M2M ネットワークに最適化されているとはいえない.

M2M ネットワークでは低消費電力が求められ, また, 調停役離脱の際に継続してアプリケーションを実行するにはできる限り早く調停役を再配置することが必要とされる. そのため, 本研究では, シンプルで軽量かつ汎用的な通信プロトコルである MQ Telemetry Transport (MQTT) を M2M ネットワークに適用し, 消費電力の低減や再配置時間の最適化が行われた調停機構を構築する.

2 研究背景

2.1 関連研究

協調作業アプリケーションにおける調停役の再配置は, リーダ選択問題と捉えることができる. リーダ選択問題は, 複数のノードから1つのノードをリーダーとして選ぶ問題であり, 多くの研究がなされている [1]. P2P ネットワークにおける調停機構の研究では, 動的に最適な調停役を決定するプラットフォームを構築している [2]. しかし, これらは低消費電力が求められる M2M ネットワークにおいては必ずしも最適化されているとは言えない.

2.2 MQTT

本研究で使用する通信プロトコルである MQTT について説明する. 先に述べたように MQTT は M2M ネットワークの実現に適したシンプルで軽量のプロトコルであり, 消費電力を抑えられ, HTTP と比較して処理速度に優れるといった特徴を持つ.

通信には publish/subscribe モデルを採用している. クライアントはメッセージを送信する publisher とメッセージを受信する subscriber としての役割を持ち, MQTT broker と呼ばれるサーバを介して通信を行う. publisher は特定の topic を指定してサーバにメッセージを送信 (publish) し, subscriber はその topic に対して受信することをあらかじめ申し込んでおく (subscribe) と, 送信されたメッセージが配信される.

また, MQTT には本研究で使用する特徴的な機能である will 機能が存在する. これはサーバとクライアントとの通信が意図せず切断された場合に, サーバが will 機能により指定された topic とメッセージを subscriber に送信する機能である.

2.3 研究の位置付け

M2M ネットワークに適した調停機構を確立するため, MQTT を用いた調停機構を構築し, その有効性を評価する. M2M ネットワークにおける調停機構では, 低消費電力であること, 最適な調停役を選択しできる限り早く再配置を行うことを考慮すべきである. 本研究では, MQTT を用いることで低消費電力を実現し, MQTT の will 機能を利用することで再配置時間を最適化する.

3 調停機構

3.1 フェーズ遷移

本研究で構築する調停機構には, どの端末が調停役にふさわしいかを定める選択フェーズと, 調停役が選択された後に任意のアプリケーションを実行するアプリケーションフェーズの2つのフェーズがある.

選択フェーズにおいて調停役が決定した場合にアプリケーションフェーズへの遷移が発生する. また, アプリケーションフェーズにおいて調停役が離脱し, その離脱を検知した場合に選択フェーズへの遷移が発生する.

本研究では, アプリケーションフェーズにおける具体的なアプリケーションを考慮しない. アプリケーションフェーズへ移行した場合, 任意のアプリケーションが実行されると想定する.

3.2 リーダ選択

調停役選択には端末と server の間の Round Trip Time (RTT) を考慮する.

WiFi 環境のような不安定なネットワーク環境ではメッセージの遅延が生じる可能性が高い. 調停役は他の端末と比べて処理負荷が高いため, server とのやりとりが低遅延で済む端末を選ぶべきである. そこで server との間の RTT が最も小さい端末を調停役とする.

RTT の測定は一定時間間隔で行い, 平均 RTT を算出して利用する.

3.3 再配置手順

調停役の再配置の手順を図1を用いて説明する. ここで, 本研究では, サーバには再配置処理のための実装を加えず, メッセージ配信のためのみに利用することとする. そのため MQTT broker は故障しないことを前提とする.

図1(a)はアプリケーションフェーズにおいて各端末が調停役とその調停役に従うフォロウに分かれてアプリケーションを実行していることを示している. 各端末は一定時間ごとに自身の IP アドレスと平均 RTT を, IP アドレスと平均 RTT 登録用の topic に publish し, そのメッセージが配信された端末は IP アドレスと

Arbitration mechanism using MQTT in M2M network

[†]Shinji MARUYAMA [‡]Shuhei HASHIMOTO [†]Hideyuki TAKADA[†]College of Information Science and Engineering, Ritsumeikan University[‡]Graduate School of Information Science and Engineering, Ritsumeikan University

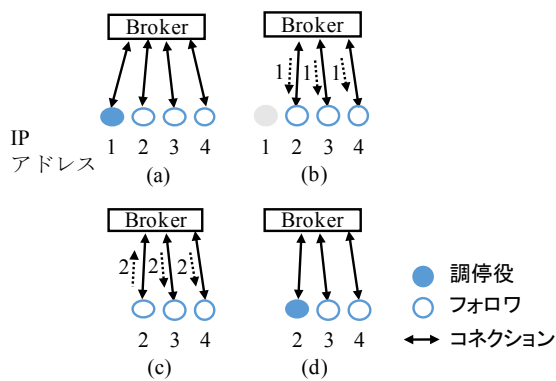


図 1: 調停役再配置手順

平均 RTT を紐づけて登録する．平均 RTT はメッセージが送られてくるたびに更新される．また，調停役となった端末は自身の IP アドレスを調停役 IP アドレス登録用の topic に publish し，そのメッセージが配信された端末は調停役の IP アドレスとして登録する．ここで，各端末は各 topic を subscribe している．そのため各端末は全端末の IP アドレスとそれに紐づく平均 RTT，またどの端末が調停役であるかを知っている．

各端末は Broker server に接続する際，will 機能を利用し，自身の IP アドレスを will メッセージとして will トピックに登録する．アプリケーションフェーズにおいていずれかの端末がネットワークから離脱した場合，図 1 (b) のように Broker server から will メッセージが各端末に配信される．もし配信された will メッセージが調停役の IP アドレスであれば，調停役が離脱したことを検知し，選択フェーズへ移行する．

選択フェーズにおいては，各端末で登録された IP アドレスに紐づけられた平均 RTT を比較し，自身の平均 RTT が最小であれば図 1 (c) のように自身の IP アドレスを調停役 IP アドレス登録用の topic に publish する．そのメッセージが配信された端末は調停役の IP アドレスを更新し，アプリケーションフェーズへ遷移する (図 1 (d)) ．

4 評価実験

4.1 実装

MQTT broker には mosquitto を使用し，MacOS で稼働させる．クライアント端末には Arduino として利用可能な ESP8266EX を搭載した WiFi モジュールである ESPr Developer を使用する．

調停役が離脱してから各端末が調停役離脱を検知するまでの時間は，Broker server に設定するタイムアウトに依存する．そのタイムアウトを 3000msec に設定する．平均 RTT は，各端末が自身の RTT 登録用の topic を用意し，その topic に対して publish してから自身に配信されるまでの時間を 1000msec ごとに計測することで求める．

4.2 実験

全ての端末がアプリケーションフェーズに移行してから調停役を離脱させ，残った端末から新たに調停役を再配置させる実験を行う．調停役の離脱はクライアント端末の電源を落とすことで実現する．

実験では各端末ごとの再配置時間と，それに伴う送受信パケット数の計測を行う．調停役離脱から離脱検知までの時間は，先に述べたように設定するタイムアウトに依存するため，離脱検知から調停役選択までの時間を再配置時間として扱う．計測には Wireshark を使用する．調停役を含む 4 端末を使用し，計測は 5 回を行い，その平均を実験結果として示す．

4.3 結果

新たに調停役となる端末にとっての再配置時間とそのフォロワにとっての再配置時間に分けて計測結果を示す．新たな調停役にとっての平均再配置時間は 10.1628msec，平均パケット数は 4 パケット，フォロワにとっての平均再配置時間は 6.4444msec，平均パケット数は 1 パケットであった．

4.4 考察

定期的に各端末の IP アドレスと平均 RTT を更新しているため，調停役から will メッセージが配信された場合，瞬時に自身の平均 RTT が最小か判断し，最小であれば自身の IP アドレスを publish する．そのメッセージが配信されると，各端末は再配置が完了したと判断できる．新たに調停役となる端末は自身の IP アドレスを publish する分だけフォロワよりも再配置までの工程が多くなるため，再配置時間がフォロワよりも長くなったと考えられる．

平均するとフォロワの再配置時間の方が短い，新たに調停役となる端末よりも先にそのフォロワとなる端末が現在の調停役の離脱を検知した場合は，フォロワの再配置時間の方が長くなった．これは，新たに調停役となる端末よりも先に現在の調停役の離脱を検知すると，その分新たな調停役の publish メッセージが配信されるまでに待ち時間が発生するためだと考えられる．

5 おわりに

本論文では，M2M ネットワークにおいて調停役を必要とする協調作業環境に対して MQTT を用いた調停機構を提案した．

低消費電力が求められる M2M ネットワークに最適化した調停機構を構築するため，軽量のプロトコルである MQTT を利用した．

評価実験から，MQTT の will 機能を利用することで再配置時間を数 msec ~ 数十 msec 程度になることが分かった．

今後の課題として，比較プロトコルを用意して同様の実験を行い，M2M ネットワークの調停機構における MQTT の有効性を示すことがあげられる．

参考文献

- [1] Sudarshan Vasudevan, Jim Kurose, and Don Towsley. Design and analysis of a leader election algorithm for mobile ad hoc networks. In *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, pp. 350–360. IEEE, 2004.
- [2] 吉永浩和, 土屋健, 小柳恵一ほか. Peer-to-peer 環境における協調作業アプリケーションのための適応型調停プラットフォーム. 情報処理学会論文誌, Vol. 46, No. 8, pp. 2162–2174, 2005.