



IoT 環境における REST 通信のための CoAP と HTTP の選択指標に関する検討

メタデータ	言語: jpn 出版者: 宮崎大学工学部 公開日: 2020-06-21 キーワード (Ja): キーワード (En): 作成者: 久保田, 真一郎, 溝上, 紘史, 山場, 久昭, 岡崎, 直宣, Kubota, Shin-Ichiro, Mizogami, Hiroshi メールアドレス: 所属:
URL	http://hdl.handle.net/10458/5589

IoT 環境における REST 通信のための CoAP と HTTP の選択指標に関する検討

久保田 真一郎^{a)}・溝上 紘史^{b)}・山場 久昭^{c)}・岡崎 直宣^{d)}

A Study of Effective Indicators to Select the Protocol CoAP or HTTP Communicating with REST in IoT Environment

Shin-Ichiro KUBOTA, Hiroshi MIZOGAMI, Hisaaki YAMABA, Naonobu OKAZAKI

Abstract

The IoT(Internet of Things), which means to connect all of the things around us to the Internet, enables us monitor and control status of things. A huge number of clients communicate with a server using various band-width in IoT environment. The Constrained application protocol(CoAP) has been proposed by IETF so as to use in IoT. However, the difference between the performance of CoAP and HTTP in various communication environment is not shown. This paper shows some indicators in a case where the CoAP works more effectively than HTTP based on a simulation.

Keywords: Internet of Things, Constrained Application Protocol

1. はじめに

近年、身の回りのあらゆるモノに通信機能を持たせインターネットに接続することで、モノの状態監視や自動制御等を行う、Internet of Things(IoT)が注目されている。あらゆるモノに通信機能を持たせるため IoT では、ZigBee のような低コストで低帯域、低消費電力な無線通信規格が用いられることが想定される。また、モノとの情報を交換する方法の1つとして REST による通信が考えられる。文献¹⁾では、M2M 用途において、REST アーキテクチャが適する理由を述べている。REST とは、Web システム作成における設計原則である。しかし、制約のあるようなネットワークにおいて、HTTP による REST 通信を実現しようとした場合、通信におけるオーバーヘッドや TCP の信頼性確保によるパケット数増加などにより、輻輳等の問題が発生する。このため、通信能力に制限のあるような無線通信規格に適したプロトコルの提案が行われており、IETF では Constrained Application Protocol(CoAP)²⁾が提案されている。CoAP は HTTP に比べてヘッダサイズが小さく、軽量である等の特徴を持つプロトコルであり、制約のある通信環境下での使用に適している。

現在のスマートフォンには多数のセンサが搭載され、Bluetooth Low Energy など新たな無線通信規格を搭載した端末もあり、センサ情報の送受信には ZigBee に限らず、スマートフォンのように様々な通信規格を使い情報を送受信する場合が考えられる。文献³⁾においては、ZigBee 端末上での HTTP と CoAP の性能比較が行われている。

しかし、通信規格の多様化に対して、どの程度の通信帯域で CoAP が有効であるかなど、CoAP と HTTP の選択の指標は十分に明確化されていない。

文献⁴⁾では、IoT の適用範囲によっては、情報の生成数が膨大になることが予想されている。本研究では、センサ端末によるセンシングシステム構築の際の HTTP、CoAP、また通信規格の選択の指標を示すことを目的とし、通信帯域や端末数の変化を想定したシミュレーションを行い、HTTP と CoAP の転送効率の比較を行うことで、通信環境ごとの CoAP の有効性の測定、評価を行う。

2. Internet of Things (IoT)

IoT では、身の回りの様々なモノの情報をセンシング、さらに通信機能を持たせることで、そのモノの状態の把握、遠隔操作、さらに自動制御を行い新たなサービスを生み出す。IoT のサービスの対象としては、家庭内の家電等による小規模なネットワークから、屋外やビルなどにおける多数のセンサを利用した大規模なネットワークまで様々である。また現在多くの人が所持しているスマートフォン端

a)情報システム工学科准教授

b)情報システム工学科学部生

c)情報システム工学科助教

d)情報システム工学科教授

末に新たなセンサが内蔵されてきていることや、Bluetooth Low Energy 対応端末の登場などから、スマートフォンはセンシングされた情報を受け取るだけでなく、スマートフォンがセンサ端末として機能するシステムの出現も想定される。

大規模ネットワークでのセンシングや、人が多く集まる地点でのスマートフォンからの情報収集等を想定した場合、サービス規模の拡大に伴いネットワークに参加するクライアントの数も増加していく。ネットワークに参加するクライアント数の増加に対して、適切なネットワークが構築されなければ、ネットワークにおける輻輳等の問題の発生につながってしまい、センシング情報の転送効率が低下してしまうことが考えられる。

これらのことから、IoT サービスに参加する端末数の増加や構築するネットワークの規模に対応して、使用する無線通信規格や、通信方式を考慮することは、IoT サービスの設計において重要な要素になる。

3. Constrained Application Protocol

Web サービスの多様化に伴い、無線センサから得た情報を、インターネットを介した Web サービス上で利用する要望が増えることが予想される。HTTP はインターネット上での Web サービスの実現で利用される代表的な通信プロトコルである。しかし、ZigBee ネットワークのように通信帯域幅が制約された通信環境では、HTTP ヘッダに付加する情報によりオーバーヘッドが大きくなり TCP におけるコネクションの確立、切断の手順によるパケット数の増加のため、センサネットワーク上での通信に HTTP をそのまま用いることが適さない場合がある。

制約のある通信環境下において、HTTP のような REST のアーキテクチャを実現するプロトコルの代替として、IETF より CoAP が提案されている。

CoAP は通信帯域幅が狭く、低速であるなどの制約があるネットワーク上での効率的なデータ転送に適する特徴を持つ。主な特徴としては、以下のような点が挙げられる。

- HTTP に比べ、パケットヘッダサイズが小さい
- UDP での使用を想定
- 非同期通信が可能

また、CoAP はパケットのヘッダとして以下のような情報を付加する。メッセージタイプの **Confirmable**、**Non-confirmable** の設定により、再送制御による信頼性を確保する。

- メッセージタイプ
- GET、POST 等のメッセージコード
- メッセージ ID

CoAP はパケットヘッダのサイズが 4 バイト程度であり、HTTP に比べてヘッダサイズが小さいため、一度に送信できるパケットサイズが制限されている環境下において、パケットの断片化を減少させ、パケット送信エラーの発生機

会を減らす。また、CoAP は UDP 上での使用を想定しており、TCP に比べて通信シーケンスが少ない。図 1 に HTTP、CoAP それぞれを使用した場合の通信シーケンスを示す。

4. 研究目的

IoT によるセンシング情報の応用範囲の広がりに伴い、センシング情報の Web 上での利用が考えられる。インターネット上での Web サービスの実現に有効な通信プロトコルは HTTP であるが、ネットワークにおけるクライアント数や、使用する通信規格による帯域の制限等の要因から、輻輳などの問題回避のためにセンサネットワーク区間でのデータ転送に CoAP 利用の選択が必要となる場合がある。本研究では、プロトコル選択の指標を与えるために、実験による HTTP と CoAP の性能測定を行う。

5. 実験

本章では、HTTP と比較した CoAP の有効性の調査を目的としたシミュレーション実験について述べる。実験では、想定される通信環境として、クライアント数、通信帯域幅の 2 つの要素の変化に対する、HTTP と CoAP の性能を測定し、評価を行う。実験は仮想環境上の Ubuntu14.04 内にて行った。

実験に使用した各プログラムを表 1 に示す。使用したプログラムは、クライアントプログラムでは指定した引数を POST するデータとしてサーバに転送し、サーバプログラムでは、受け取ったデータをサーバ側ファイルへ書き込むようそれぞれ変更を加え、実験を行った。

5.1 実験内容

実験は同一 PC 内のプロセス間通信にて行う。

クライアントプログラムは、シェルスクリプトによる並列実行にて前のプロセスの終了を待たずに次のプロセスを連続で実行していき、それぞれのクライアントがサーバに対して POST リクエストを送信する。クライアント数の制御はシェルスクリプトの for 文のループにより行い、想定するクライアント数に応じてループ回数を設定し、クライアントプロセスの生成を行う。

実験の概要を図 2 に示す。

サーバは、クライアントからリクエストを受け取った場合、サーバ側のテキストファイルに受け取ったデータの書き込みを行う。

通信を行うプロセス間の通信帯域を **tc(traffic control)** によって制限する。

送信を開始し始めてから、通信全体における最後のパケットが送信されるまで計測を行い、パケットをキャプチャする。

測定は 5 回ずつ行い、平均値を使用する。

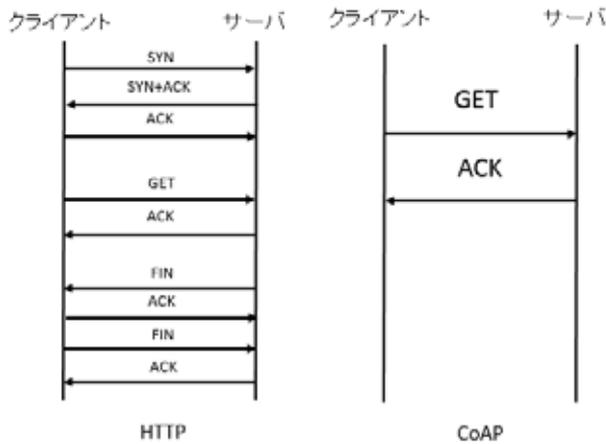


図 1. シーケンス図.

表 1. 使用プログラム.

HTTP	http_post.c ⁵⁾	Apache2.4.7+PHP
CoAP	libcoap-4.0.3 ⁶⁾	Californium-coap-18 ⁷⁾

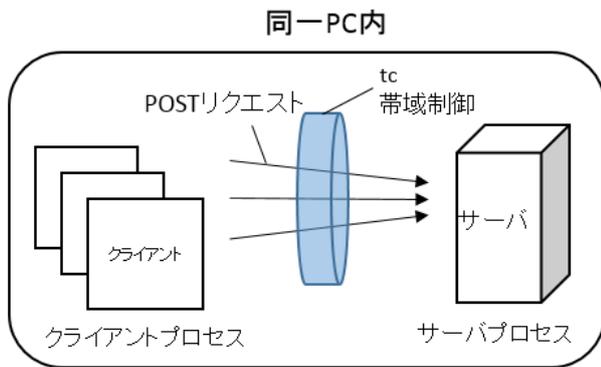


図 2. 実験概要図.

tc による帯域制御では、送信されるデータはデータキューへ入れられる。データキューとは別にトークンが溜まっているバケットが存在し、データを送信する際はそのバケット内のトークンを消費し、送信を行う。バケット内にトークンがなくなった場合、データの転送速度はトークンの補充速度と同量となり、この補充速度の設定を行うことにより通信帯域幅の制限を行う。実験にて想定した通信帯域は、100Mbps、50Mbps、10Mbps、5Mbps、3Mbps、1Mbps、500kbps、250kbps、100kbps、また、クライアント数は100、1000、10000 の場合を想定し実験を行った。

実験によって測定する項目は以下のものである。

総経過時間

スクリプトを実行し、最初の packets が送信されてから通信全体における最後の packets が送信されるまでの経過時間

パケット数の増減率

通信において再送等が1度も発生せず、正常終了した場合の総パケット数を0%とした場合の、総パケット数の増減率

サーバへの書き込み成功率

クライアント数に対する、サーバへのデータ書き込み成功件数の割合

5.2 実験結果

測定結果をそれぞれ図3から図6に示す。

図中において再送制御ありの CoAP を CoAP-CON、再送制御なしの CoAP を CoAP-NON として表記する。

図3はクライアント数100で様々な通信帯域で実験を行った際の総経過時間とサーバへの書き込み成功率を表している。実線が総経過時間、点線が書き込み成功率を示す。HTTP 通信では、帯域幅3Mbpsまで総経過時間の増加は見られなかったが、帯域幅1Mbps以降通信時間が増加し始め、帯域幅が100kbpsの場合、全ての通信が終了するまでに平均7.8秒を要した。また、サーバへのデータ書き込み成功率は、グラフにはないが CoAP の場合は再送制御あり・なしどちらの場合でも、測定した全ての帯域において100%であった。HTTP の場合、帯域幅250kbps程度までは100%であったが、100kbpsになると、データの書き込み率は、平均51.2%まで低下した。このとき、HTTP 通信では、帯域幅5Mbpsから100kbpsにかけて平均1.4%~11.8%程度の総パケット数の増加が測定されていた。

図4はクライアント数が1000と10000それぞれの場合の通信の総経過時間を表している。実線がクライアント数10000の場合、点線がクライアント数1000の場合の結果を示している。クライアント数1000の場合、CoAP 通信では再送制御ありの場合でも大幅な時間の増加は見られなかったが、HTTP 通信では帯域幅3Mbps程度から通信時間が増加を始め、帯域幅が狭くなるにつれてさらに上昇、帯域幅100kbps程度では全ての通信が終了するまでに平均108.2秒を要した。クライアント数10000の場合、CoAP の再送制御ありの通信では、3Mbpsを境に経過時間が上昇していき、帯域幅100kbpsでは、通信が終了するまでに平均60.8秒を要した。また10000クライアントのHTTP 通信では、帯域幅10Mbpsから総経過時間が増加し始め、帯域幅250kbpsの時点では通信終了まで平均285.8秒を要した。しかし、帯域幅250kbpsから100kbpsにかけては、総経過時間は平均286.9秒で、通信時間の大幅な変動は見られなかった。

図5はクライアント数1000の場合のパケット数の増減率とサーバへの書き込み成功率を示している。実線がパケット増減率、点線がサーバへの書き込み成功率を表している。HTTP 通信では、帯域幅3Mbpsまで、パケットの総数が正常終了時より少しずつ増加し、帯域幅3Mbpsの時点で平均14.8%増加していたが、3Mbps以降になると総パケット数が減少を始め、250kbpsから100kbpsにかけてパケットの総数は、250kbpsでは平均-8.0%、100kbpsでは平均

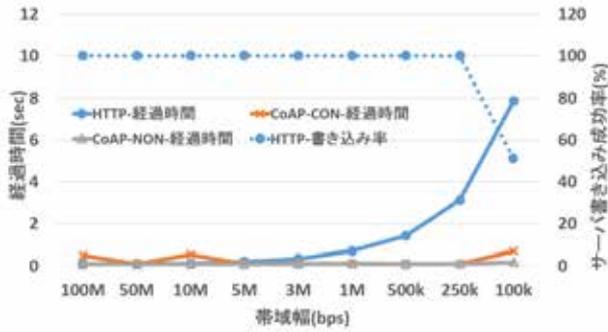


図 3. 100 クライアントの場合の経過時間・サーバ書き込み率.

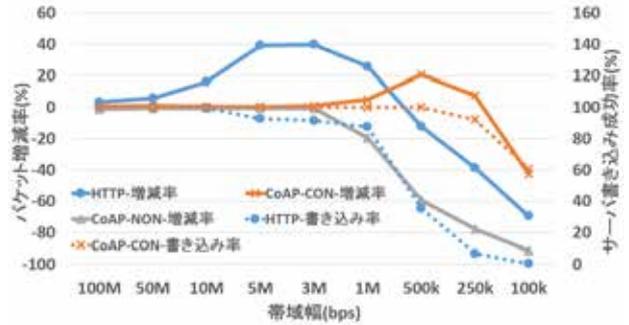


図 6. 10000 クライアントの場合の packets 増減率・サーバ書き込み率.

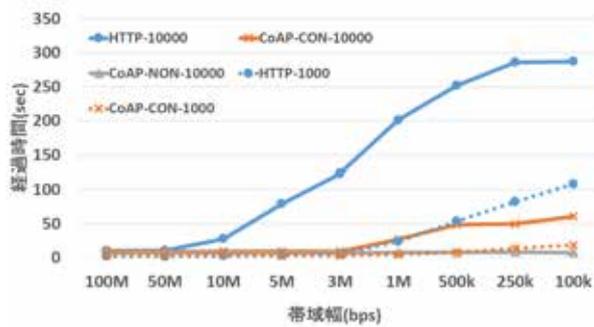


図 4. 10000・1000 クライアントの場合の経過時間.

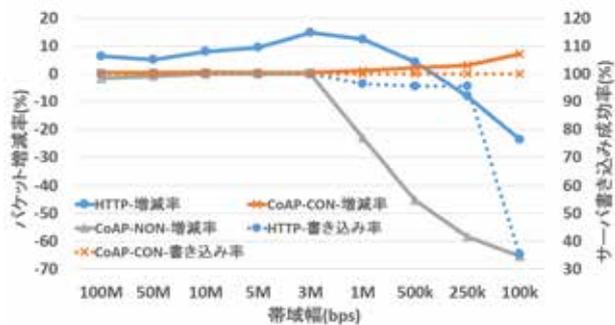


図 5. 1000 クライアントの場合の packets 増減率・サーバ書き込み率.

-23.7%となり、通信の正常終了時のパケット数を下回った。また、サーバへの書き込み成功率は帯域幅 3Mbps までは 100%であったが、帯域幅 1Mbps から 250kbps にかけて、平均 3.7%~4.6%書き込みに失敗し、帯域幅 100kbps では成功率は大幅に減少、書き込み成功率は平均 35.2%になっていた。CoAP の再送制御ありの通信では、帯域幅が 1Mbps 以降では平均 1.1%~7.1%の総パケット数の増加が見られたが、サーバへの書き込み成功率は帯域幅 100kbps の時点でも 100%であった。CoAP の再送制御なしの場合、1Mbps の時点で総パケット数が平均-23.2%と減少をはじめ、総パケット数の減少にほぼ一致するようにサーバへの書き込み成功率も低下していた。

図 6 はクライアント数 10000 の場合の packets 数の増減率とサーバへのデータの書き込み成功率を示している。実

線が packets 数増減率、点線がサーバへの書き込み成功率を表している。HTTP 通信では、帯域幅 10Mbps から総パケット数が増加し、5Mbps では平均 39.1%、3Mbps では平均 39.7%の packets の増加が見られた。その後、帯域の狭まりと共に総パケット数は減少を始め、帯域幅 100kbps 時の総パケット数は、通信の正常終了時に比べ平均-69.5%まで減少した。また、HTTP 通信でのサーバへの書き込み成功率は、帯域幅 50Mbps まで 100%であったが、帯域幅 10Mbps から 1Mbps の間で平均 0.4%~12.5%減少、さらに 1Mbps 以降成功率は大幅に減少し、100kbps 時の書き込み成功率は 0.5%であった。一方、CoAP の再送制御ありの通信の場合、総パケット数は帯域幅 1Mbps 以降増加を始め 500kbps 時に平均 20.7%増加、それ以降 packets 数は減少していき、100kbps 時には通信の正常終了時に比べ平均 42.5%減少していた。また、サーバへの書き込み成功率は 250kbps から減少を始め、100kbps 時には成功率は平均 60.5%であった。CoAP の再送制御なしの通信では、総パケット数、サーバへの書き込み成功率共に 1Mbps 以降から減少し始め、100kbps 時にはサーバへの書き込み成功率は、平均 8.8%まで減少していた。

6. 考察

実験結果より HTTP、CoAP どちらを用いた通信でも、クライアント数の増加、また通信帯域幅の狭まりに伴い通信効率低下することがわかった。帯域幅が十分広く、100Mbps 程度の場合では HTTP と CoAP の間の通信効率に大きな差は見られなかったが、クライアント数 100 の場合帯域幅 500kbps、1000 の場合 1Mbps、10000 の場合 10Mbps を境に HTTP の通信時間が CoAP に比べ増加を始め、またサーバへの書き込み成功率も、クライアント数 100 と 1000 の場合には 100kbps、クライアント数 10000 の場合には 500kbps の時点で、HTTP が CoAP よりも書き込み成功率が大幅に低くなっていた。これは、HTTP が TCP 上で動作しデータ送信の際にコネクションの確立、切断を行い、UDP 上で動作する CoAP よりも輻輳、再送の発生機会が増加しているためだと考えられる。また、TCP では、パケットの送信順の整合性が乱れてしまった場合などに、

RST パケットを送信し通信を終了する。これにより、輻輳等の発生が原因でコネクションの確立が失敗し、通信が終了してしまう割合が増加すると、サーバにデータが到達しなくなるため、CoAP 通信の場合よりもサーバへの書き込み成功率が低くなってしまふ。HTTP では、クライアント数 1000 の場合帯域幅 250kbps 以降、クライアント数 10000 の場合帯域幅 500kbps 以降で総パケット数が正常終了時を下回ったが、これは、途中で通信が終了してしまう割合が増えたためだと考えられる。

また、HTTP、CoAP 共にサーバへの書き込み成功率が 100%であった場合でも、クライアント数 1000 の場合の帯域幅 5Mbps から 1Mbps、またクライアント数 10000 の場合の帯域幅 10Mbps から 1Mbps の際に、HTTP のほうがパケットの増加率が多くなる傾向が見られた。通信端末として、使用できる電力量に限りのあるような無線端末を用いる場合、パケットの送信回数の増加は端末の電力消費の増加につながるため、端末の消費電力量の削減という観点からも HTTP に比べ CoAP の方がより効率よくデータ転送を行えるのではないかと考えられる。

しかし、CoAP を用いた場合でも再送制御を行わなかった場合、クライアント数 1000、10000 ともに帯域幅 1Mbps 以降サーバへの書き込み成功率が大幅に低下してしまう。よって経路上でパケット損失が多発するようなネットワークにおいては、適切な再送制御を行うことが重要である。

7. まとめ

本研究では、シミュレーション実験にて HTTP、CoAP をそれぞれ使用した場合の、通信帯域幅、クライアント数の変化に対する通信効率の比較を行った。実験結果より、クライアント数が 100 でも通信帯域が 100kbps のような制約された環境において、HTTP での通信に対して CoAP での通信が有効性を示すことがわかった。またクライアント数が 10000 などの場合は、通信帯域が数百 kbps 程度の場合に限らず、5Mbps 程度の帯域幅でも、CoAP の使用によりいくつかの観点において効率よくデータの転送ができるということがわかった。今回の実験は、実際の使用を想定したシミュレーションであったが、今後は実際の無線端末を使用した実験を行い、今回の実験結果と同様の結果が得られるのかの調査を行っていきたい。

参考文献

- 1) 藤田隆史, 後藤良則, 小池新 “M2M アーキテクチャと技術課題”, 電気情報通信学会誌. Vol.96, pp.305-312, 2013.
- 2) Z.Shelby, K.Harke, and C.Bormann, “The Constrained Application Protocol (CoAP)”, RFC7252 (June, 2014).
- 3) 山田真弘, 羽田久一, 三次仁, “ZigBee ネットワーク区間を含む HTTP 通信の効率化”, 電気情報通信学会技術研究報告. USN, Vol.111, pp.35-38, 2011.

- 4) 横谷哲也 “IoT/M2M を取り巻く環境と標準化”, 日本信頼性学会誌 : 信頼性 Vol.34, pp.540-545, 2012.
- 5) SourceForge.NET
http://souptonuts.sourceforge.net/code/http_post.c.html.
- 6) libcoap: C-Implementation of CoAP
<http://libcoap.sourceforge.net/>.
- 7) Californium (Cf) CoAP framework in Java
<http://people.inf.ethz.ch/mkovatsc/californium.php>.