IoT時代の軽量プロトコル
MQTTのRubyへの適応

2016年2月3日
JOPF　アプリケーション部会
株式会社日立ソリューションズ　三好　秀徳
富士通株式会社　原　嘉彦

目次

1. 通信状況の変革
2. HTTPプロトコルの問題点
3. 何故非HTTPプロトコルなのか？
4. 非HTTPプロトコル例の紹介
5. 非HTTPプロトコル例・MQTT
6. MQTTのビジネス適応例
7. MQTT適用を支援するプロダクト（Broker）
8. MQTT適用を支援するプロダクト（MQTTクライアント）
9. RubyのMQTT適応に向けた動向
10. 次のステップへ向けて

付録: 参考情報

1. 通信状況の変遷

- IoT時代が到来
  - 送受信を行う機器数の変化
    - ネットワークにつながるすべての機器がデータの送受信者となりうる
    - すべての機器が、データ発信者かつ受信者となりうる
  - 送受信されるデータサイズの変化
    - 送信データが数Byteですむほど極小（例: 温度センサが発信する温度情報）
  - 送受信されるデータ数の変化
    - 従来とは桁違いのデータ数がネットワーク上を飛び交う

2. HTTPプロトコルの問題点

- 通信効率
  - サーバ・クライアント方式
  - 同期型通信
  - 実現は容易だが、速度面での性能は低い
  - IoT時代の情報発信機器数は飛躍的に増大するため、通信速度の向上が必須

- データサイズ
  - 通信の際データに付随するプロトコルヘッダサイズが最小でも50Byte
  - 通信データが巨大であれば50Byteの付属物は誤差のうち
  - IoT時代のデータサイズは極めて小さいため、50Byteもの付属物がつくことは、通信性能上好ましくない

3. 何故非HTTPプロトコルなのか？

- HTTPプロトコルより高速な通信ができる必要がある
- HTTP通信より小さなサイズのパケット（ヘッダ＋データ）で通信できる必要がある

4. 非HTTPプロトコル例の紹介

- プロトコル例とHTTPの問題に対する改善点

	MQTT	CoAP	AMQP	HTTP（参考）
コネクションの開始／終了処理	恒久的	コンテンツを構成する一連のデータの送信開始・受信終了時に実施	恒久的	データの送受信1件ごと
パケットヘッダ最小サイズ	2Byte	4Byte	8Byte	50Byte

- 他の主な特徴
  - 非同期型
  - Publish/Subscribeモデルの採用

5. 非HTTPプロトコル例・MQTT

- プロトコル例とHTTPの問題に対する改善点

	MQTT	CoAP	AMQP	HTTP（参考）
コネクションの開始／終了処理	恒久的	コンテンツを構成する一連のデータの送信開始・受信終了時に実施	恒久的	データの送受信1件ごと
パケットヘッダ最小サイズ	2Byte	4Byte	8Byte	50Byte

- 他の主な特徴
  - 非同期型
  - Publish/Subscribeモデルの採用

6. MQTTのビジネス適応例

企業のシステムへの適用例

- Facebook Messanger
- Connected Vehicle
- 北九州スマートコミュニティー創造事業

街全体を対象としたメッセージングシステムを実現

街中の工場のような生産施設や自動車のような交通機関に設置されたセンサ、各家庭内の家電等とのメッセージ交換を実現。
巨大な分析/制御システムを構築。

世界中の人々が発するメッセージの送受信システムを実現。

世界規模のメッセージングシステムを実現

7. MQTT適用を支援するプロダクト(Broker)

- サービス
  - AWS IoT Message Broker
  - Microsoft Azure IoT Hub
  - IoT Foundation for IBM Bluemix
  - 国内各クラウドサービスでもMQTTのサービスを提供しているものあり
- 企業製品(ハード)
  - MessageSight（ IBM ）
- 企業製品(ソフト)
  - Akane（株式会社時雨堂）
- OSS製品(Broker)
  - Mosquitto(http://mosquitto.org/)
  - RabbitMQ（ http://www.rabbitmq.com/)
  - Paho（http://www.eclipse.org/paho/)
  - Active MQ Apollo（ https://activemq.apache.org/apollo/)

8. MQTT適用を支援するプロダクト(Client)

Japan OSS Promotion Forum

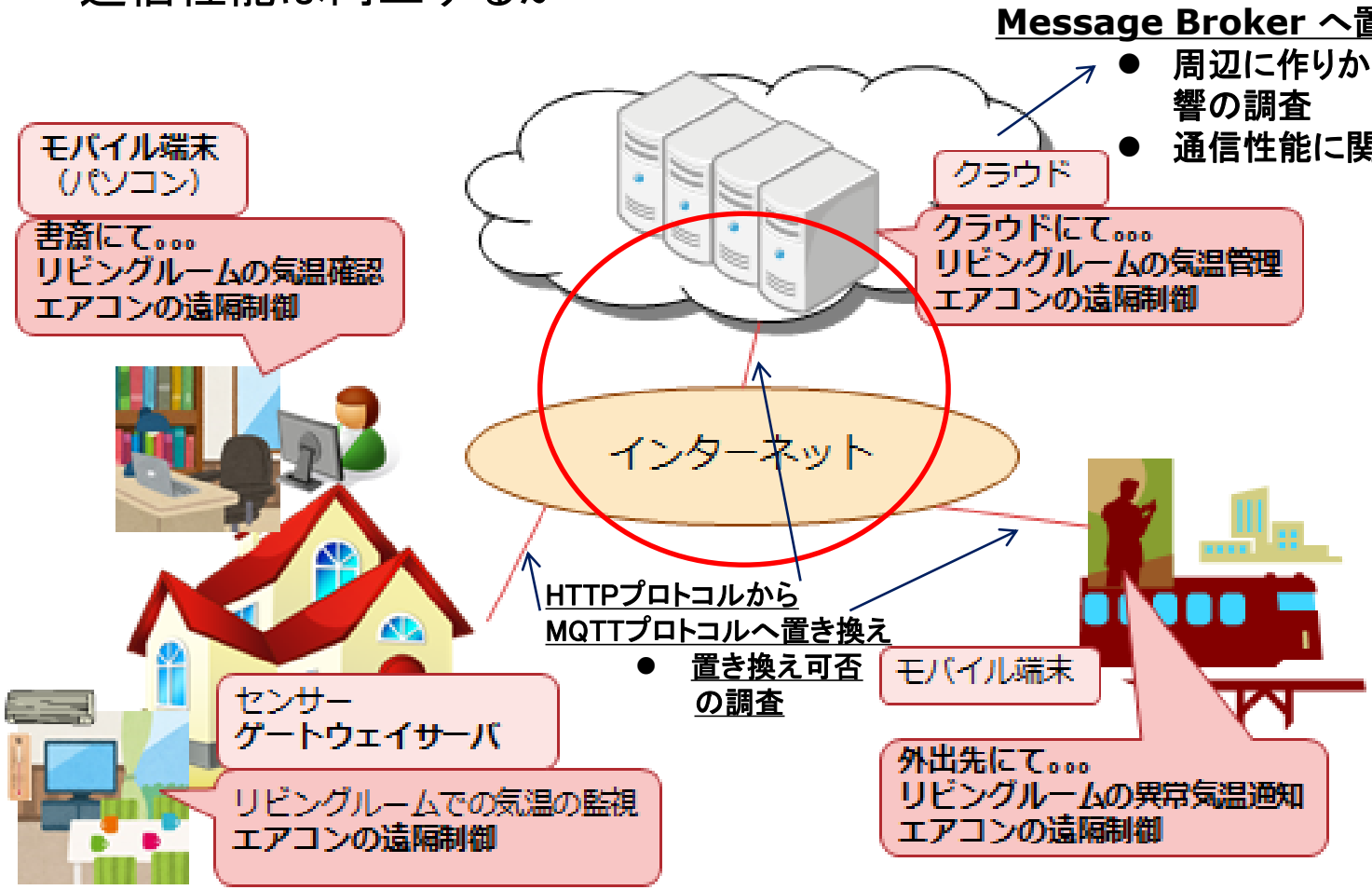
- OSS製品(Client)
  - Paho(C / C++ / Java / JavaScript / Lua / Python)
    (http://www.eclipse.org/paho/)
  - ruby-mqtt(Ruby) (https://github.com/njh/ruby-mqtt)
  - mruby-mqtt(mruby)(https://github.com/hiroeorz/mruby-mqtt)
  - Node.red(node.js)(http://nodered.org/)

9. RubyのMQTT適応に向けた動向

- 性能向上
  - mruby開発チームによる性能検証、機能改善の動き
    - MQTTとHTTPでの時間（レイテンシ）の比較等を実測中

10.次のステップへ向けて

- アプリケーション部会作成IoT実証モデルを使った実証
  - HTTP通信ありきで設計されていたシステムのMQTT対応は可能か
  - 通信性能は向上するか



付録：参考情報

- 通信プロトコルの仕様
  - MQTT V3.1 プロトコル仕様
    (http://public.dhe.ibm.com/software/dw/jp/websphere/wmq/mqtt31_spec/mqtt-v3r1_ja.pdf)
  - The Constrained Application Protocol (CoAP) (http://tools.ietf.org/html/rfc7252)
  - OASIS Advanced Message Queuing Protocol(AMQP) Version 1.0
    (http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf)
- 事例
  - Building Facebook Messenger
    (https://www.facebook.com/notes/facebook-engineering/building-facebook-messenger/10150259350998920/)
  - Connected Vehicle(コネクテッド・ビークル)に関する紹介記事(zdnet)
    (http://japan.zdnet.com/extra/ibm_iot_201504/35062776/)
  - MQTT、IBM MessageSightのご紹介
    (https://www.ibm.com/developerworks/jp/websphere/library/connectivity/ms_mqttintro/)
  - 北九州スマートコミュニティ創造事業 (http://www.tokugikon.jp/gikonshi/265/265tokusyut5.pdf)
  - 北九州スマートコミュニティ創造事業の実証成果について(http://www.nedo.go.jp/content/100750431.pdf)
- サービス
  - Message Broker for AWS IoT (http://docs.aws.amazon.com/iot/latest/developerguide/iot-message-broker.html)
  - Microsoft Azure IoT Hub プレビュー (https://azure.microsoft.com/ja-jp/services/iot-hub/)
  - Internet of Things Foundation (http://www-03.ibm.com/software/products/ja/internet-of-things-foundation)
  - クラウド 機能・サービス(MQTT(β)) | ニフティクラウド (http://cloud.nifty.com/service/mqtt.htm)
  - Nシリーズ MQTT(β) | ビットアイル・エクイニクス(http://www.bit-isle.jp/service/cloud/n-series/spec/mqtt.html)
  - MQTT as a Service sango - 株式会社時雨堂 (https://sango.shiguredo.jp/)
  - Milkcocoa | リアルタイムアプリ・IoTやるならMilkcocoa (https://mlkcca.com/)
- 製品
  - Akane (http://akane.shiguredo.jp/)