

5G SA 環境での IoT システムのための通信性能の調査

伊藤 千紗 (指導教員：小口 正人)

1 はじめに

IoT (Internet of Things) 機器から収集された様々なセンサデータをクラウドで収集、解析し、活用することが期待されている。しかし、モバイル環境にある IoT データの収集では、自動運転車や倉庫内自走ロボットなどの各種サービスで要求される通信スループットや通信遅延を維持することが必要である。これらの通信スループットや通信遅延の性能要求を満たすには、現在のモバイル通信技術である LTE では難しく、性能向上が課題となる。高性能なモバイル通信技術として、5G の活用が期待されているが、新しい通信であるため IoT への実用化は十分にされておらず、IoT に利用するための性能特性については明らかではない。

本研究では、5G SA (Stand Alone) 環境での IoT データ通信性能を調査する。5G SA とは、4G 用のコア装置を流用する NSA (Non Stand Alone) に対して、5G 専用のコア装置と基地局装置を使用するものである [1]。国立情報学研究所内に設置された 5G SA 環境で、IoT 用通信ライブラリを提供する SINETStream を用いて基本性能を調査する。

2 関連技術

2.1 5G SA

5G とは、第 5 世代移動通信システムの略で LTE(4G) をさらに進化させた通信規格である。最高伝送速度は 10Gbps とされており、LTE(4G) の帯域幅よりもさらに広い帯域幅である。5G のメリットは高速大容量で、これによって大容量のデータを迅速に収集・転送が可能になる。また、超低遅延であることから遠隔でのリアルタイム制御の可能性が広がるうえに、多数同時接続が可能のためセンサ数を格段に増やすこともでき、今後 IoT 機器との接続など社会発展の基盤としても大きく期待されている。現在商用で利用されている 5G NSA は、4G 用のコア装置である EPC (Evolved Packet Core) を流用し、4G 基地局と連動し 4G と 5G の電波を併せて使用している。そのため 4G に一度接続した後に 5G 接続となり、4G と 5G の両方の電波を利用しないと 5G 通信できない仕組みとなっていた。これに対して 5G SA とは、5G 専用のコア装置を導入して 5G 専用の基地局装置を使用するものである。4G 接続をせずにはじめから 5G 接続となるため、5G 通信開始までの時間が短くなるだけでなく、5G の電波のみを利用して 5G 通信ができるようになるため、広帯域な周波数帯を組み合わせた通信も可能となる。

2.2 SINETStream

SINETStream[2][3] とは、国立情報学研究所で開発された広域ネットワークを介してデータを欠損なく確実に収集・解析するための機能を提供しているソフトウェアパッケージである。SINETStream は、IoT 通信に適した Publisher/Subscriber 型の通信モデルを採用している。MQTT[4] 準拠のブローカや Apache Kafka[5] のような標準的なメッセージブローカに対応した共通 API を提供しており、具体的には、(1) センサ等から収集されるデータをクラウドや大学などに設置されたサーバへ書き込む (2) サーバに収集されたデータを解析プログラムに読み込むといった機能を提供している。また、通信やデータの暗号化、センサ等のデバイスの認証を行う機能も含んでいるため、安全な広域データ収集を実現できる。本研究では SINETStream を用いた MQTT および Kafka の通信の性能調査を行う。

2.3 perftool

perftool[6] とは SINETStream を介してデータの送受信を行い、そのメトリクス情報を出力するプログラムである。クライアント端末で Publisher と Subscriber のプログラムを実行し、サーバで起動しているブローカにメッセージを送信し、ブローカで収集されたメッセージをクライアント端末の Subscriber で取得するまでの時間を測定する。メトリクス情報の例としては、1 秒あたりに受信したメッセージ数、1 秒あたりに受信したメッセージ量、1 秒あたりのエラー数であるエラーレートがある。

3 実験

3.1 実験概要

NTT docomo の 5G SA 環境と LTE 環境でそれぞれスループットを測定する。Raspberry Pi 4 Model B を Wi-Fi STA-TION SH-52B のモバイルルータに USB 接続して測定を行う。測定には perftool を使用し、メトリクス情報のひとつである reader_msg_bytes_rate に着目する。ブローカは Kafka と MQTT ブローカの Mosquitto[7] を用いる。また、データ暗号化はしないものと AES アルゴリズムで暗号化したものの 2 通り、データサイズは 1024byte, 10240byte, 102400byte の 3 通りで比較する。

5G SA の性能について予備実験を行った。実験環境を表 1 に示す。モバイルルータと端末を USB で接続し、Google

表 1: 予備実験環境

Raspberry Pi	
OS	Raspbian GNU / Linux 11
CPU	ARMv7 Processor rev 3 (v7l)
Main Memory	4GB
macOS	
OS	macOS Big Sur バージョン 11.1
CPU	1.6 GHz デュアルコア Intel Core i5
Main Memory	8 GB 2133 MHz LPDDR3

表 2: 速度テスト結果

(Mbps)	LTE(4G)		5G SA	
	Download	Upload	Download	Upload
Raspberry Pi	22.4	8.54	208	56.0
macOS	16.9	2.71	436.4	51.9

のインターネット速度テストで測定を行う。3回の測定結果の平均値を表2に示す。Raspberry PiとmacOSのどちらの端末においてもLTEと比べて5G SAの方が性能が良く、Upload性能に比べてDownload性能の方が良いことが確認できた。また、端末性能の差により5G SAのDownload性能が顕著に異なることも確認できた。

3.2 実験結果

5G SAとLTE環境でデータ暗号化の有無で比較した結果を図1に示す。横軸はoutput_countで計測回数を表し、縦軸はreader_msg_count.rateで受信メッセージレートを表している。実験では、100メッセージの送受信における平均通信スループットを10回計測している。暗号化無しが赤色でplain、有りが緑色でaes、サイズは1024byteの結果を表している。また、実線が5G SA、破線がLTEである。左側がKafkaブローカを経由したもの、右側がMQTTブローカを経由したものである。5G SAとLTEで比較すると、どちらも5G SAの方が性能が良いことが確認できる。暗号化の有無での比較では、どちらのブローカを経由した場合も暗号化による性能の差は見られないことが確認できた。

データサイズで比較したものを図2に示す。縦軸横軸は図1と同様で暗号化なしの結果を示す。赤色が1024byte、緑色が10240byte、青色が102400byteで実線が5G SA、破線がLTEである。どのデータサイズにおいても5G SAの方が性能が良く、データサイズが大きくなるほど性能の差が大きいたことが読み取れる。

4 まとめと今後の課題

IoT用通信ライブラリSINETStreamを用いて5G SA環境のスループットの測定を行い、従来のLTEと比べて高い通信スループットを示すことを確認した。また、データ暗号化による性能の差はあまりないこと、データサイズが大きくなるほど5G SAとLTEの性能の差は大きくなることも確認できた。今後の課題は、圧縮やTLSなど様々なパラメータで測

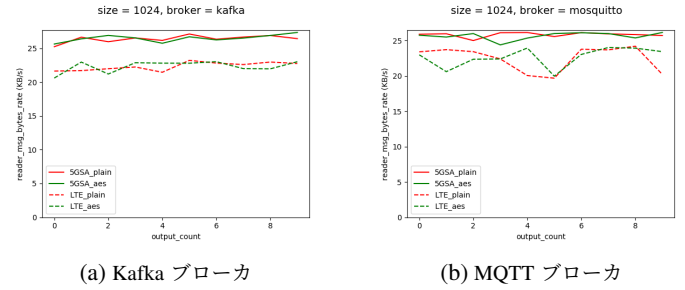


図 1: 暗号化比較

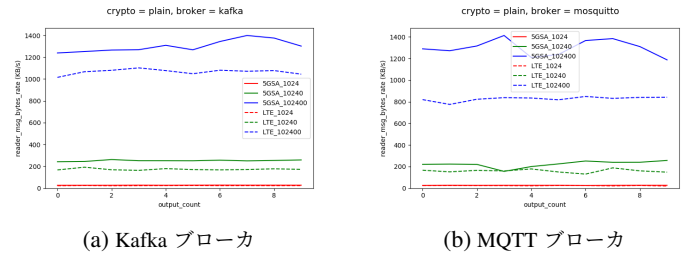


図 2: データサイズ比較

定し、性能特性を調査していきたいと考えている。また、クライアントを増やしてマルチスレッドで実験を行っていく。

謝辞

5G SA環境の利用では、NTTコミュニケーションズ様にご協力いただきました。また、本研究はJSPS科研費JP19H04089の助成を受けたものです。

本研究成果はデータ活用社会創成プラットフォームmdxを利用して得られたものです。

参考文献

- [1] docomo. 5G. <https://www.docomo.ne.jp/area/5g/>.
- [2] SINETStream. <https://sinetstream.net/>.
- [3] Atsuko Takafusa, Jingtao Sun, Ikki Fujiwara, Hiroshi Yoshida, Kento Aida, and Calton Pu. SINETStream: Enabling Research IoT Applications with Portability, Security and Performance Requirements. *In Proc. COMPSAC 2021*, pp. 482–492, 2021.
- [4] MQTT (Message Queue Telemetry Transport). <https://mqtt.org/>.
- [5] Apache Kafka. <https://kafka.apache.org/>.
- [6] perftool. <https://github.com/nii-gakunin-cloud/sinetstream/tree/main/python/sample/perftool>.
- [7] Mosquitto. <https://mosquitto.org/>.