

7Y-04

モバイル環境における IoT システムのための圧縮処理を用いた通信性能評価

伊藤 千紗† 竹房 あつ子†† 中田 秀基††† 小口 正人†
 †お茶の水女子大学 ††国立情報学研究所 †††産業技術総合研究所

1 はじめに

IoT 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、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できるかが課題となる。IoT 通信では小規模データが大量に送信されるため、通信環境に合わせた転送方法を検討する必要がある。本研究では、IoT 用通信ライブラリを提供する SINETStream[1][2] を用いて、IoT データ通信の性能評価を行う。既発表研究で通信スループットを高めるにはマルチスレッド通信やデータ圧縮が有効であることを示した [3]。本稿ではモバイル環境で、送受信データ特性や圧縮率に焦点を当てた圧縮処理の有効性を調査し、性能向上のための検討を行う。

2 実験

LTE 環境で IoT 通信の往復通信時間を測定する。LTE はモバイル SINET[4] の Softbank の LTE 環境を用いる。SINETStream では Publisher は Writer, Subscriber は Reader と呼ぶ。また、本実験ではブローカとして MQTT[5] ブローカの Eclipse Mosquitto v. 1.6.9 を用いる。実験環境を表 1 に示す。クライアントは Raspberry Pi, サーバはデータ活用社会創成プラットフォーム mdx の VM を用いる。クライアント端末から柏にある mdx VM の MQTT ブローカに対して通信する。モバイルルータとクライアント端末は USB で接続する。また、モバイルルータは Aterm MR51FN を用いる。データ圧縮アルゴリズムには gzip と zstd[6] を用いる。zstd は gzip とほぼ同じ圧縮率だが、より高速に圧縮と解凍が行えるように設計されている圧縮アルゴリズムである。測定には perftool[7] を使用する。perftool とは SINETStream を介しての MQTT ブローカとの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。本実験では、指定したデータサイズのデータを Publisher が 100 個送信し、Subscriber が受信するまでを 10 セット行うことを、perftool 実行 1 回分と定義する。そして perftool 実行 1 回分の実行時間を測定し、累積送信データ量を実行時間で割ったものをスループットとする。

2.1 データ特性による比較

LTE 環境で送受信データ特性を変えて比較を行う。データサイズは 10KB, 100KB の 2 通りであり、圧縮アルゴリズムは gzip を使用する。送受信データは、perftool ではデフォルトが乱数だが、本実験では乱数のデータ “rand”, 圧縮が効きやすいように 0 で埋めたデータ “0”, JSON 形式

Evaluation of IoT Communication Performance using compression in a Mobile Environment

†Chisa Ito ††Atsuko Takehisa †††Hidemoto Nakada †Masato Oguchi

†Ochanomizu University

††National Institute of Informatics

†††National Institute of Advanced Industrial Science and Technology (AIST)

表 1: 実験環境

クライアント : Raspberry Pi	
機種名	Raspberry Pi 4 Computer Model B 4GB RAM
OS	Raspbian GNU / Linux 11
CPU	ARMv7 Processor rev 3 (v7l), 4 コア, 2065 MHz
Main Memory	4GB
サーバ : mdx VM	
OS	Ubuntu 20.04.5 LTS
仮想 CPU コア数	16
メモリ	24.19GB

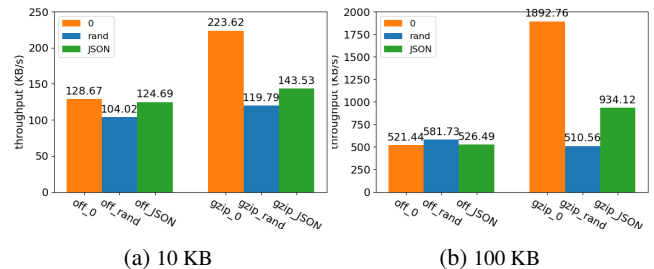


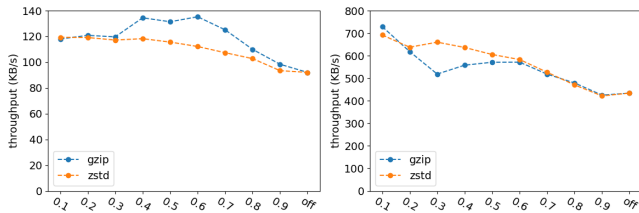
図 1: データ特性の違いによる通信スループットの比較

で書かれた実際のセンサ情報を読み込んだデータ “JSON” の 3 通りで測定する。なお、JSON のデータサイズは厳密には 11KB, 112KB だが、ここでは 10KB, 100KB と近似して比較する。

通信スループットを図 1 に示す。横軸の “off” は圧縮無, “gzip” は gzip アルゴリズムで圧縮を表しており、縦軸はスループットである。グラフから、乱数データにおいてはデータ圧縮の効果がみられないことがわかった。これは乱数データは圧縮率が悪く、データ圧縮自体に時間がかかってしまうため性能が低下していると考えられる。0 埋めデータと JSON データでは、データ圧縮を行ったものは性能向上がみられ、特に 0 埋めデータの性能向上が著しかった。また、データサイズが大きくなるほど性能差が大きくなった。実験から、送受信するデータサイズやデータ特性に応じた圧縮処理の適用が重要であることがわかった。

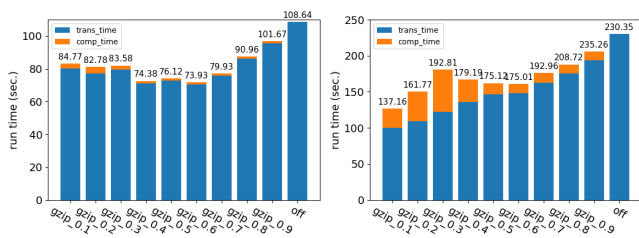
2.2 圧縮率による比較

LTE 環境で圧縮率を変えて比較を行う。データサイズは 10KB, 100KB の 2 通りである。圧縮アルゴリズムは gzip, zstd を用いる。また本実験では、指定した圧縮率目標値に近くなるデータを生成するプログラムを作成し、送受信データを生成する。圧縮率目標値は、0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 の 9 通りで比較する。ここで、圧縮率 0.1 とは圧縮によりデータサイズが 0.1 倍になることを目標としている。

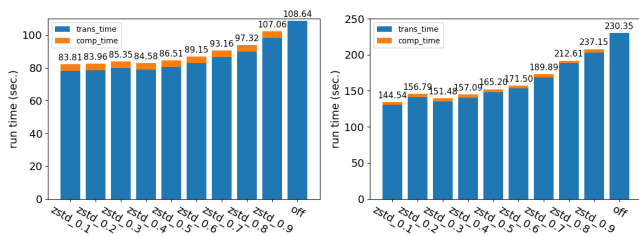


(a) 10 KB (b) 100 KB

図 2: 圧縮率の違いによる通信スループットの比較



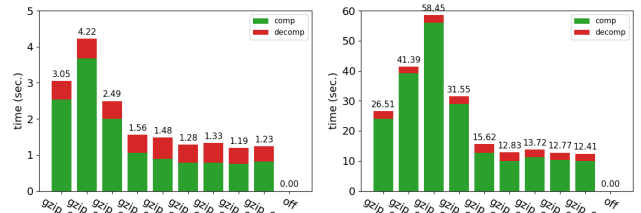
(a) gzip, 10 KB (b) gzip, 100 KB



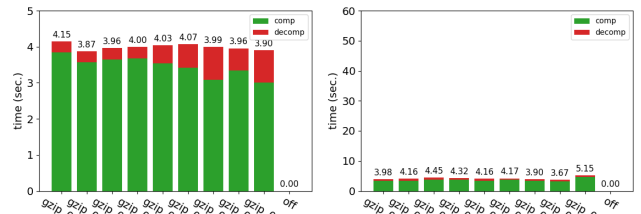
(c) zstd, 10 KB (d) zstd, 100 KB

図 3: 圧縮率の違いによる圧縮解凍・転送時間の比較

通信スループットを図 2 に示す。青色のグラフが gzip アルゴリズムで圧縮、橙色のグラフが zstd アルゴリズムで圧縮した結果である。横軸の数字は圧縮率目標値を表しており、縦軸はスループットである。結果から gzip と zstd 両方で、高圧縮であるほど性能が良く、データサイズが大きいほど性能差が顕著であることが確認できた。さらに、圧縮解凍・転送時間を比較したものを図 3, に示す。橙色が圧縮解凍時間で perftool 実行時間のうちデータ圧縮とデータ解凍にかかる時間のみを測定したもの、青色が転送時間で全体の実行時間から圧縮時間を引いたものとした。また、圧縮解凍時間のうち、圧縮時間と解凍時間を比較したものを図 4 に示す。緑色がデータ圧縮時間、赤色がデータ解凍時間を表している。図 4 から、gzip と zstd いずれもデータ圧縮率と圧縮・解凍時間にあまり相関がみられないことがわかる。gzip は、10KB で圧縮時間の割合はかなり小さく、データサイズが大きくなると圧縮時間の割合が大きくなり、圧縮率 0.3 付近で圧縮時間が長くなることが読み取れる。zstd は、データサイズに関係なく圧縮時間はほぼ一定であることがわかる。よって、データ圧縮を行う際に、データサイズが小さいときは gzip、大きいときは zstd のようにアルゴリズムを適切に選択することが重要だと考えられる。また、圧縮時間が全体の実行時間に与える影響は小さいこともわかった。実験から、効率的に通信するにはデータの特性により適切なアルゴリズムを選択することが重要であること、圧縮時間が全体の実行時間に与える影響はわずかであるため、実行時間の観点においても圧縮処理が有効であることがわかった。



(a) gzip, 10 KB (b) gzip, 100 KB



(c) zstd, 10 KB (d) zstd, 100 KB

図 4: 圧縮率の違いによる圧縮・解凍時間の比較

3 まとめと今後の課題

IoT 用通信ライブラリ SINETStream を用いてモバイル環境の通信性能の測定を行い、圧縮処理の有効性を調査した。その結果、高圧縮であるほど性能が高くデータサイズが大きいほど性能差が顕著であること、圧縮時間と圧縮率に相関はないこと、圧縮時間が全体の実行時間に与える影響はわずかであること、適切なアルゴリズムの選択や、データサイズ・特性に応じた圧縮処理の適用が性能向上に効果的であることが確認できた。今後は、効果的な圧縮処理の適用について引き続き調査を行っていく。また、通信環境に合わせて IoT 通信の転送効率を高める手法を検討する。

謝辞

本研究は JSPS 科研費 JP19H04089, および 2023 年度国立情報学研究所公募型共同研究 (2023-23S0205) の助成を受けたものです。また、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] SINETStream. <https://sinetstream.net/>.
- [2] 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.
- [3] 伊藤千紗, 竹房あつ子, 中田秀基, 小口正人. 5G 環境での IoT システムのためのモバイル通信性能の評価. マルチメディア, 分散, 協調とモバイルシンポジウム 2023 論文集, Vol. 2023, pp. 367–374, 06 2023.
- [4] モバイル SINET. <https://www.sinet.ad.jp/wadci>.
- [5] MQTT (Message Queue Telemetry Transport). <https://mqtt.org/>.
- [6] Zstandard. <https://facebook.github.io/zstd/>.
- [7] perftool. <https://github.com/nii-gakunin-cloud/sinetstream/tree/main/python/sample/perftool>.