

多種多様な時空間コンテンツ生成のための 異分野データ収集処理手法の提案

長尾健太郎[†] 妙中 雄三^{††} 山本 寛^{†††} 池永 全志^{††††} 塚本 和也^{†††††}

[†] 九州工業大学大学院情報工学府先端情報工学専攻 〒820-8502 福岡県飯塚市川津 680-4

^{††} 奈良先端科学技術大学院大学先端科学技術研究科 〒630-0192 奈良県生駒市高山町 8916-5

^{†††} 立命館大学情報理工学部 〒525-8577 滋賀県草津市野路東 1 丁目 1-1

^{††††} 九州工業大学大学院工学研究院 〒804-8550 福岡県北九州市戸畑区仙水町 1-1

^{†††††} 九州工業大学大学院情報工学研究院 〒820-8502 福岡県飯塚市川津 680-4

E-mail: [†]nagao@infonet.cse.kyutech.ac.jp, ^{††}yuzo@is.naist.jp, ^{†††}hiroyama@fc.ritsumeikan.ac.jp,

^{†††††}ike@ecs.kyutech.ac.jp, ^{†††††}tsukamoto@cse.kyutech.ac.jp

あらまし 我々はこれまで、新しいIoT サービスのための通信基盤として、物理位置を意識したIoT データ連携が可能なGC 情報プラットフォームを提案した。GC 情報プラットフォームでは、地理に基づいて集められたデータから、様々なサービス事業者が必要とするデータを収集・処理することで多種多様な時空間コンテンツを生成・配信することが期待される。本研究では、多種多様な時空間コンテンツ生成を促進することを目的として、ネットワーク上の機器が協調しながら効率的に時空間コンテンツに必要な前処理を実行する手法を提案する。さらに、既存手法と比較して提案手法が高いスケーラビリティと通信品質（データ遅延、パケットロス率）を提供可能なことをシミュレーションによる評価を通じて示す。

キーワード Internet of Things, 時空間データ, データ収集, データ前処理

Proposal of heterogeneous data collection and processing method for various kinds of spatio-temporal contents generation

Kentaro NAGAO[†], Yuzo TAENAKA^{††}, Hiroshi YAMAMOTO^{†††}, Takeshi IKENAGA^{††††}, and
Kazuya TSUKAMOTO^{†††††}

[†] Graduate School of Computer Science and System Engineering, Kyusyu Institute of Technology

^{††} Graduate School of Science and Technology, Nara Institute of Science and Technology

^{†††} College of Information Science and Engineering, Ritsumeikan University

^{††††} Faculty of Engineering, Department of Electrical Engineering and Electronics

^{†††††} Faculty of Computer Science Engineering, Kyusyu Institute of Technology

E-mail: [†]nagao@infonet.cse.kyutech.ac.jp, ^{††}yuzo@is.naist.jp, ^{†††}hiroyama@fc.ritsumeikan.ac.jp,

^{†††††}ike@ecs.kyutech.ac.jp, ^{†††††}tsukamoto@cse.kyutech.ac.jp

Abstract We have proposed Geo-Centric information platform(GCIP), which achieves geo-location aware data collaboration for IoT services. In GCIP, various service providers are expected to collect and process various kinds of data to accelerate the generation of the spatio-temporal contents. In this paper, we proposed new data processing architectures that perform scalable and efficient data collection and preprocessing but also spatio-temporal contents generation. Finally, we showed that our proposed method can provide not only high scalability but also communication quality in terms of data delay and packet loss rate through simulation experiments, compared with the existing method.

Key words Internet of Things, Spatio-temporal Data, Data collection, Data preprocessing

1. 研究背景・課題

近年、スマートシティやスマートホーム等、人々の生活圏内のモノ/ヒト等に関する情報を連携させる IoT サービスが注目されている。その中でも、様々な分野で利用されるデータ（異分野データ）を二次利用することで、新しい IoT サービスを創出することが期待されている [1]。

我々の研究グループでは、地域上に分散する、IoT 機器の種類や、その機器が生成するデータの種類（数値データ、画像データ等）に依存しない IoT データの二次利用を目的として、地理空間を意識したデータ（時空間データ）の収集・処理・配信を行う地理指向情報プラットフォーム（GCIP: Geo-Centric Information Platform）[2] を提案している。GCIP では、地理空間を緯度経度に基づいて階層構造のあるメッシュ形状に分割し（図 1）、メッシュ構造に対応したネットワークを構成する。GCIP ではメッシュ毎に、2 種類のエッジサーバ（データ蓄積サーバと異分野連携サーバ）を配置する（図 2）。データ蓄積サーバには、メッシュに対応した地理空間範囲内の全ての時空間データが収集され、異分野連携サーバはデータ蓄積サーバから選択的に時空間データを取得、組み合わせることで地理空間を意識した新たなコンテンツ（時空間コンテンツ）を生成する。

時空間コンテンツを生成するには、各データのメタデータ（形式、種類、精度、送信位置等）を把握する必要があるが、GCIP で流通されるデータ形式は様々であり、メタデータは事前にデータに設定されていない場合が多い。そのため、不足しているメタデータを補完するための手段が必要となる。またメッシュ内の機器は、メッシュに対応する地域内の交換局、基地局内に設置するため、各種サーバを配備できる空間に限られる。そのためストレージ、ネットワーク、計算資源に関する物理的な制約がある。この時、データ蓄積サーバに保存している大量の時空間データを複数の異分野連携サーバに送信すると、メッシュ内全体のネットワーク/ストレージ資源を逼迫する。そこで、異分野連携サーバにデータ送信の際は、プラットフォーム側でデータサイズを削減し、ネットワーク上に流れるトラフィック、異分野連携サーバに保存されるデータサイズを抑える必要がある。

本稿では、メッシュ毎に機器が協調しながら、時空間コンテンツ生成に必要なデータ削減を行うことで、メッシュ内の計算資源を有効に活用可能な時空間データ収集・処理手法を提案する。2 節では、GCIP の説明と、データ収集トポロジに関する説明を行い、GCIP 上のデータ収集・処理要件を整理する。3 節では、提案手法を紹介し、4 節で比較手法、評価指標を説明する。5 節でシミュレーション結果に関する考察を行い、6 節でまとめと今後の課題について議論する。

2. GCIP におけるデータ収集・処理要件

2.1 物理位置調和型通信

我々は、図 1 のメッシュコードを IPv6 アドレス中に含めることで、物理位置に基づく通信経路制御を実現する物理位置調和型通信技術を提案した [2][3][4]。これを用いることで、メッシュ内のデータ収集者は、データ収集時に物理位置を意識することが可能となる。

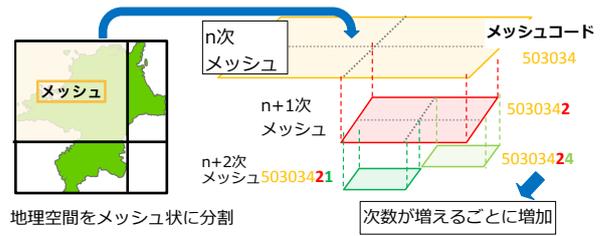


図 1 GCIP のネットワーク構造

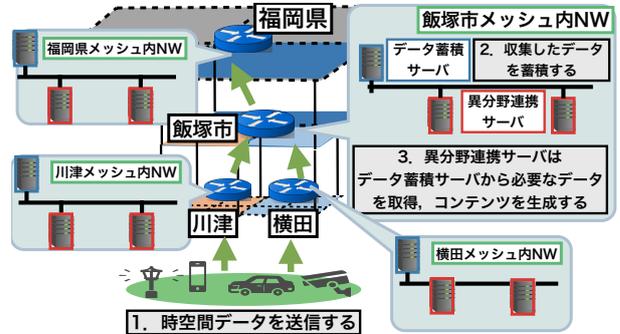


図 2 GCIP の想定環境

2.2 GCIP におけるデータ収集形態

メッシュ内の機器は、交換局、基地局、AP 等、複数の管理ネットワークが物理的に近接する地点に設置され、それぞれの収集範囲に応じたメッシュに所属する。メッシュの次数が増えるほど地理空間の分割粒度が細くなり、交換局は 2~4 次メッシュ (10km~2.5km 四方)、基地局は 5~9 次メッシュ (1.2km~78m 四方)、AP は 10 次メッシュ (39m 四方) をカバーする。各メッシュ内のデータ蓄積サーバは、自治体などのその地域の管理者（地域管理者）が設置することを想定する。異分野連携サーバは、データ蓄積サーバで保存された時空間データを活用しサービスを提供する事業者が設置することを想定する。本研究では GCIP 上のメッシュ内には、データ蓄積サーバは 1 台、異分野連携サーバは複数台存在すると想定する [6]。

2.3 GCIP におけるデータ収集課題

GCIP でのデータ収集・処理には以下の 3 つの要件がある。

要件 1: 時空間コンテンツの増加に適応するアーキテクチャ

GCIP 環境では、データ蓄積サーバが様々な事業者に対してデータ送信を行う。事業者は、収集データが増えるに連れ、様々な時空間コンテンツを生成する。そのため GCIP 環境では、地域管理者が、時空間コンテンツの増加に伴うデータの取得依頼の増加に柔軟に対応可能な収集・処理アーキテクチャであることが求められる。

要件 2: メッシュ内の効率的なデータ削減前処理方法

メッシュ内の機器は、交換局、基地局、AP 付近に設置されるため、設置可能なスペースに物理的な制限がある。そのため、メッシュ内の機器はメッシュの規模に応じて、ネットワーク/ストレージ/計算資源に制約を持つ。一方で、メッシュ上には地理範囲上の全ての IoT データが収集されるため、大量のデータトラフィックが発生する。この時、データ蓄積サーバで保存している収集データを複数の異分野連携サーバに送信すると、メッシュ内全体のネットワーク/ストレージ資源を逼迫する。そこで、収集データを異分野連携サーバに送信の際は、計算資源の制約の中でプラットフォーム側でデータサイズを削減する前処理を実行し、ネットワーク上に流れるトラフィック、異分野連

携サーバに保存されるデータサイズを抑える必要がある。尚、前処理はデータの利用目的に応じて異なるため、異分野連携サーバから受け取る前処理内容に応じて実施する必要がある。

要件 3: 収集データの二次利用に必要な前処理の実行

GCIP で収集されるデータ (収集データ) は、多様な IoT 機器から送信されるため、様々なデータ形式であることが予測される。事業者が収集データを 2 次利用するには、データの種類や精度、送信機器情報、送信時の詳細な位置情報等のメタデータ/コンテキストが必要であるが、収集したデータにはそれらの情報が含まれていないことが考えられる。そこで、不足しているメタデータ/コンテキストを補完する手段が必要となる。この時、GCIP は時空間データを二次利用するため、一次利用先に影響の無い手法が求められる。更に、収集データを保管するデータ蓄積サーバは、不足しているメタデータを IoT 機器から取得し、収集データと紐付ける必要がある。

3. 提案手法

本稿では、上記要件を満たすために、複数の異分野連携サーバに共通する前処理を先行して実施することで、データ収集・処理効率を向上させる手法を新たに提案する。

3.1 データ収集・処理アーキテクチャ

本稿では、要件 1 を満たすために、GCIP 上のデータ収集・処理を行うために、各メッシュ内に設置されるメッシュルータとデータ蓄積サーバ、異分野連携サーバのそれぞれの間での通信には、Pub/Sub モデルを活用する (図 3)。Publisher にはメッシュルータ、Broker にはデータ蓄積サーバと要件 2 の前処理を実行する前処理サーバからなる複数の機器、Subscriber にはサービス事業者を想定する。データ収集に Pub/Sub モデルを利用することでメッシュルータと異分野連携サーバの通信が疎結合となる。これにより Subscriber である異分野連携サーバの増加に対して Publisher が送信先の追加や変更をする必要がなくなり、メッシュルータの処理負荷を抑えることが出来るため要件 1 を満足する。

前処理サーバはデータ蓄積サーバによって、メッシュ内に存在する計算資源に余裕のある他異分野連携サーバの中から選定される。要件 2 の前処理を複数のサーバに分散して実行させることにより、1 台あたりの処理負荷を抑える事が出来る。更に、複数の異分野連携サーバに共通する前処理を事前に一括して処理することで、ネットワーク/ストレージ資源の逼迫を抑えることができ、要件 2 を満足する。

要件 3 の前処理は、収集データを保管するデータ蓄積サーバが行う。収集データに欠損するメタデータは、提供者 (IoT 機器) から収集データに関する詳細なメタデータを提供してもらい、収集データのメタデータを補完する前処理をデータ蓄積サーバで実行する事で要件 3 を満足する。

3.2 データ削減前処理指定方法

提案手法では、Subscriber は取得したいデータの種類 (トピック) と前処理内容を指定し、Broker に向けて Subscription を送信する。Broker は、データ蓄積サーバに保存されたデータの中から、Subscription 要求に指定されたトピックを抽出し、前処理を実行後、Subscriber にデータを送信する。Subscription には、異分野連携サーバが要求するデータの種類、条件、前処理を詳細に指定する必要があるため、Subscription model

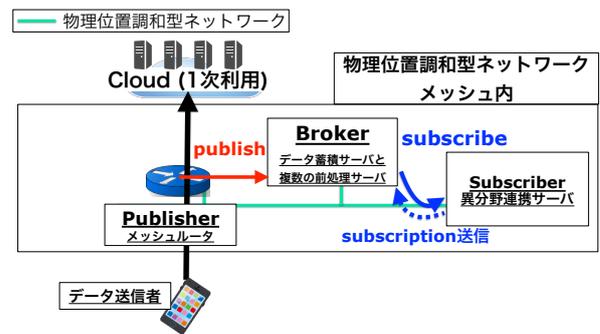


図 3 GCIP 収集・処理要件を満たす通信モデル

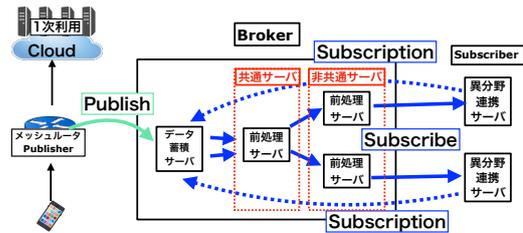


図 4 STEP1 のデータ転送経路

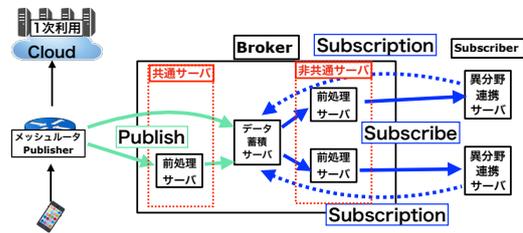


図 5 STEP2 のデータ転送経路

には Content-Based model [7] を使用することを想定する。Subscription メッセージには、トピックと属性 (機器情報、制度、送信環境、送信時間等)、前処理 (データ削減前処理) が指定され、データ蓄積サーバに向けて送信される。

3.3 データ削減前処理の最適化手法

提案手法では、下記の 2 つの STEP で、Broker 上のデータ削減順序を最適化させ、要件 2 に寄与する。ここで、複数の異分野連携サーバに共通する前処理を実行する前処理サーバを共通サーバ、そうでない前処理サーバを非共通サーバとする。

3.3.1 STEP1

STEP1 のデータの転送経路を図 4 に示す。要件 2 の前処理は Subscriber から指定されるため、STEP1 では、全ての前処理サーバはデータ蓄積サーバ - 異分野連携サーバ間に配置される。データ蓄積サーバは、異分野連携サーバからの Subscription を受け取った後、依頼された前処理を実行可能な前処理サーバを選択し、異分野連携サーバまでの経路を生成する。その後、Subscribe メッセージに、要求されたデータと異分野連携サーバまでの経路を含め、前処理サーバに向けて転送する、Subscribe を受け取った前処理サーバは、Subscribe メッセージに含まれる前処理を実行した後に、記載されている経路に従って次の前処理サーバ、または異分野連携サーバに向けて送信される。

3.3.2 STEP2

STEP2 のデータの転送経路を図 5 に示す。STEP2 では、STEP1 で取得した前処理依頼に関する情報を基に、前処理順序を最適化する。データ蓄積サーバには、メッシュ内の全ての Subscriber からの Subscription の情報を蓄積することが出来るため、複数の Subscriber に向けて共通して実行されてい

る前処理を割り出し、共通サーバを判定できる。その後共通サーバの処理順序を変更し、データ蓄積サーバに保存される前 (Publisher 側の通信) に前処理をさせるよう制御する。これにより、STEP1 では重複していた前処理を 1 回で実行でき、データ蓄積サーバに保存されるデータサイズも削減できる。

3.4 メタデータ/コンテキスト取得方法

要件 3 を満足するためには、GCIP で取得したデータに欠損しているメタデータ/コンテキストを取得する方法が必要となる。メタデータ/コンテキストを含める方法として、アプリケーションペイロード、もしくはアプリケーションヘッダ中に含める方法があるが、これは一次利用先のアプリケーションに影響がある可能性があり、要件 3 を満たせない。更に、前項の STEP2 では、Publisher 側に前処理サーバを配置するため、Publisher であるメッシュルータが、到着したデータによって転送先のサーバを制御し、Publish する必要がある。

そこで本稿では、物理位置調和型アドレス内にトピックを含める事で、Publisher がデータの内容を把握する方法を用いる。Publisher は IoT 機器が送信したデータを複製後、トピックを付与した上で Broker に向けて Publish する。Broker は Publish されたデータを受信し、不足しているメタデータの補完を行い、データ蓄積サーバに保存する。しかし、この方法だけではメタデータに関する詳細な情報を含めることは出来ないため、本手法に加えて、メタデータの詳細情報を IoT 機器から提供してもらう方法も組み合わせる。尚、収集データと、提供してもらう詳細なメタデータを紐付けるために、GCIP 内のデータフローを管理するフロー ID 管理サーバを設置する。IoT 機器は、送信したデータのフロー ID をフロー ID 管理サーバから受け取り、提供するメタデータに含めた後、データ蓄積サーバに向けて送信する。データ蓄積サーバは、受信したメタデータに含まれるフロー ID に対応した収集データに対してメタデータの補完を行う。

4. 性能評価

本章では、提案手法が時空間コンテンツをスケーラブルかつ効率的に生成可能な手法であることを示す。尚、本稿では「スケーラブル」とは、Broker に依頼される前処理量やメッシュルータに到着するデータトラフィックの増加に対して、異分野連携サーバへのデータ送信通信品質 (通信遅延、パケットロス率) の劣化割合が小さいということを指す。尚、提案手法は前処理順序の最適化を行った場合 (STEP2) で評価する。

4.1 比較手法

手法 1: データ蓄積サーバ 1 台で前処理を実行する手法

Publisher であるメッシュルータは、到着した全てのデータをデータ蓄積サーバに向けて Publish する。データ蓄積サーバは、Publish メッセージとそのデータに関する詳細なメタデータを IoT 機器から受信し、不足するメタデータを補完する。Subscriber 側では、提案手法と同様にデータ蓄積サーバに向けてデータのトピック、属性、前処理を含んだ Subscription 要求を送信する。データ蓄積サーバは、蓄積されたデータの中からトピックと属性に合致しているデータに対して、指定された前処理を実行し Subscriber に送信する。手法 1 では、データ削減前処理を 1 台で実行し、前処理依頼が集中するため処理負荷に関するスケーラビリティに問題点がある。

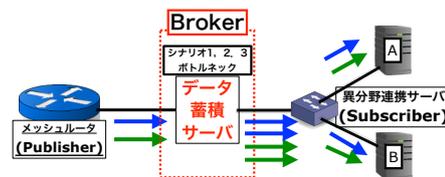


図 6 手法 1 のトポロジ

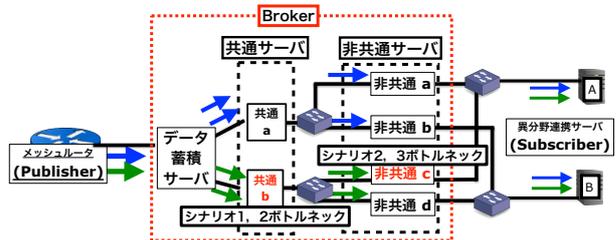


図 7 手法 2 のトポロジ

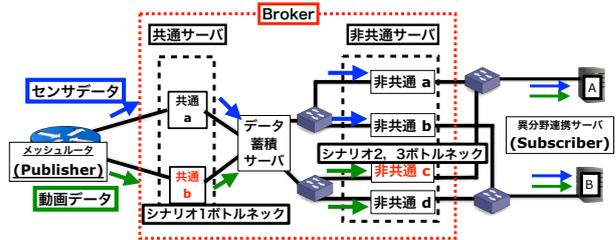


図 8 提案手法のトポロジ

手法 2: 共通する前処理を一括処理しない手法

手法 2 は、提案手法の STEP1 の形態から、STEP 2 の前処理順序の最適化を行わない通信モデルである。この手法では、異分野連携サーバが増加し、それらが同じ前処理を依頼している場合、前処理サーバに処理負荷が集中する可能性がある。

4.2 通信性能に関する定量評価

本節では、提案手法と比較手法に関する通信性能、スケーラビリティについての評価を、NS3 を用いたシミュレーションによる定量評価により行う。

4.2.1 シミュレーション環境

各手法は図 6~8 に示すような論理ネットワークを構築し、Publisher から、ストリーミング動画データとセンサデータの 2 種類のフローを流す。Subscriber となる異分野連携サーバは 2 台存在し、2 種類のデータを依頼する。手法 2 及び提案手法では、それぞれの前処理を実行する前処理サーバを計 6 台配置する。前処理は、パケット単位ではなく、アプリケーションデータ単位で実行されるため、データ送信頻度の単位は data/sec とする。ストリーミング動画データの 1data は 1 フレームの動画データを想定する。また、前処理サーバは、Subscriber によって依頼された前処理の発生条件を満たすと、処理イベントを発行する。本稿では、前処理を実行する機器はシングルコア CPU での実行を想定するため、FIFO に従って順に処理イベントを実行していく。手法 1 や手法 2 では、データの到着に対して異分野連携サーバ A, B に対する処理イベントが同時に発生する場合がある。この場合、発生した処理イベントは並列に処理され、発生したイベントの処理遅延の総和だけ時間がかかる事を想定する。

4.2.2 評価指標と検証シナリオ

定量評価の評価指標には、システム許容量、通信遅延、パケットロス率 (PLR: Packet Loss Rate) を用いる。システム許容量は「Broker 内のいずれかのサーバの平均利用率が 1 となったときのトラフィック」と定義する。本稿では、表 1 で示す 2 種類の

データの送信割合を保ちながらトラフィックを増加させ、システム許容量を測定する。また、システム許容量のトラフィックが流れる時、平均利用率 = 1 となっている機器を「ボトルネック」と定義する。次に、通信遅延は、Subscriber S に向けて送信されるデータフロー $flow$ について、「最初に処理イベントが発生してから、 S が前処理後のデータを受信するまでの時間」を評価し、 $D_{(S,flow)}$ と表現する。以降、トポロジ上の異分野連携サーバ A, B を A, B 、動画データ、センサデータに関するフローを $movie, sensor$ と表現する。また、パケットロス率は、「サーバの処理負荷が上昇し、受信バッファが溢れることによるパケットロス」を想定し、 $D_{(S,flow)}$ と同様に $PLR_{(S,flow)}$ と表現する。シミュレーション上では、各機器についての前処理バッファに最大バッファサイズ (30 に設定) を設け、最大バッファサイズ時に受信データをロスさせる事でパケットロスを再現する。また、上記の通信環境で表 2 に示すように、前処理の処理遅延を変更させることで各手法のボトルネックを変化させ、以下の 3 つのシナリオを評価する。更に、提案手法での共通前処理の一括処理による、通信性能の向上率を評価するために手法 1、手法 2 に対する提案手法の改善率を求める。改善率は式 1 により算出し、通信遅延とパケットロス率についての改善率をそれぞれ評価する。

$$ImprovementRate = \frac{CompMethod - PropMethod}{CompMethod} \quad (1)$$

($CompMethod = Method_1 \text{ or } Method_2$)

- シナリオ 1: 共通サーバのみがボトルネック
- シナリオ 2: 共通サーバ、非共通サーバがボトルネック
- シナリオ 3: 非共通サーバのみがボトルネック

システム許容量/通信遅延/パケットロス率は、トポロジを流れるトラフィックと異分野連携サーバから依頼された前処理に依存して変化する。そこで本稿では、各手法のトポロジに流すトラフィックについては以下の 2 つの通信環境を想定する。

- 通信環境 1: システム許容量内のトラフィックの通信性能
- 通信環境 2: 背景トラフィック加えた環境における通信性能

通信環境 1 では各前処理サーバの平均利用率は 1 以下となり、パケットロスが発生する程の通信負荷は発生しないため、通信遅延のみを評価する。通信環境 2 では実環境で発生するパケットロスを考慮するため、表 3 に示すパスト的に発生する背景トラフィックを加える。背景トラフィックは、データ蓄積サーバ → 各前処理サーバ (物理的には異分野連携サーバ) に向けて送信されるデータ伝送を想定する。手法 1 では、データ蓄積サーバは、背景トラフィックが発生するとデータ伝送に関する処理イベントが発生し、前処理バッファを逼迫する。手法 2、提案手法では、異分野連携サーバへのデータ受信・解析処理を想定する。各手法の背景トラフィックはトポロジ全体での負荷が同じになるように調整し、手法 2 と提案手法の各前処理サーバの解析処理負荷は手法 1 の 1/6 とする。また提案手法は、共通サーバのボトルネックを解消することで通信性能を向上する。そこで、評価には提案手法でボトルネックを回避している経路 $D_{(B,movie)}$ 、 $PLR_{(B,movie)}$ に着目する。

4.2.3 システム許容量評価結果

各手法のシナリオ別のシステム許容量を表 4 に示す。手法 1

表 1 送信データ

データ種別	1data 当たりのサイズ	データ送信頻度
動画データ	20 kByte	30 data/sec
センサデータ	300 Byte	1000 data/sec

表 2 それぞれのシナリオにおける前処理遅延 [ms]

前処理	シナリオ 1	シナリオ 2	シナリオ 3
共通 a (注2)	0.5	0.5	0.5
共通 b	7.5	5	3
非共通 a	1	1	2
非共通 b	0.5	0.5	1
非共通 c	7.5	10	9
非共通 d	2.5	5	5

表 3 機器 1 台当たり到着する背景トラフィック

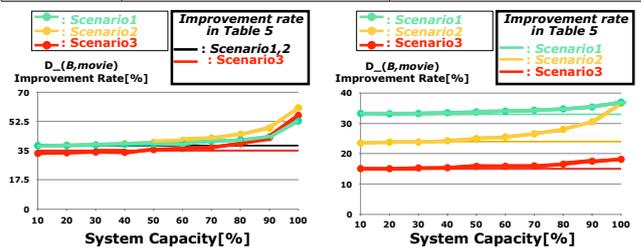
手法	処理遅延 [ms]	データ送信頻度 [data/sec]	発生時間 [ms]
手法 1	1	6000	100
手法 2	1	1000	100
提案手法	1	1000	100

表 4 システム許容量 [data/sec]

	シナリオ 1		シナリオ 2		シナリオ 3	
	sencer	movie	sencer	movie	sencer	movie
手法 1	1000	30	1000	30	1000	30
手法 2	2200	66	3300	99	2200	66
提案手法	4400	132	3300	99	2200	66

表 5 通信遅延と伝送遅延から算出した改善率

	手法 1 に対する改善率 [%]	手法 2 に対する改善率 [%]
シナリオ 1	38	33
シナリオ 2	38	24
シナリオ 3	35	15



(a) 手法 1 と提案手法の比較 (b) 手法 2 と提案手法の比較

図 9 比較手法の $D_{(B,movie)}$ 改善率 (通信環境 1)

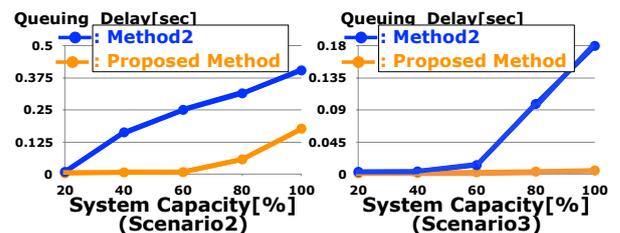
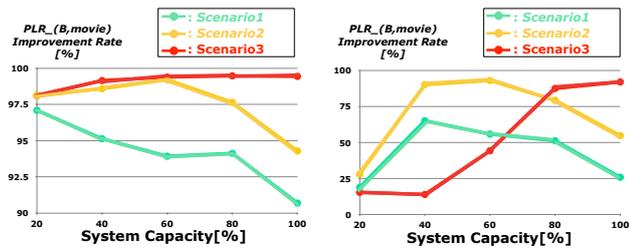


図 10 手法 2 と提案手法のキューイング遅延 (通信環境 1)

は、全ての前処理を 1 台で集中して実行しているためシナリオに依らず、最も低いシステム許容量を示す。手法 2 で共通サーバがボトルネックとなるシナリオ 1 に着目すると、提案手法では前処理の共通化により共通サーバの負荷を半減でき、システム許容量を 2 倍に改善できている。しかし、提案手法では非共通サーバの処理性能を改善できないため、手法 2 と比較した時非共通サーバをボトルネックに持つ他のシナリオでは、提案手法を用いてもシステム許容量を改善出来ていないことが分かる。

4.2.4 通信環境 1 の通信性能評価

通信環境 1 では各前処理サーバの平均利用率が 1 以下の環境のため、処理遅延、伝搬遅延が $D_{(B,movie)}$ の大半を占める。



(a) 手法 1 と提案手法の比較 (b) 手法 2 と提案手法の比較
 図 11 比較手法の $D_{(B,movie)}$ の改善率 (通信環境 2)

そこで、表 5 に各手法の処理遅延と通信遅延から算出したシナリオ別の比較手法の改善率を示す。改善率による評価結果より、全てのシナリオで $D_{(B,movie)}$ を改善している事が示される。以上より提案手法は、通信環境 1 においてシナリオや比較手法に依らず、通信遅延を改善できることを示した。

次に、図 9 に、各シナリオにおける比較手法の $D_{(B,movie)}$ を提案手法の改善率を示す。通信結果から、どのシナリオでもトラフィックが少ない場合、表 5 で示した改善率に近い値を示している。シナリオ 1 はシステム許容量が手法 2 の 2 倍の値を持つにも関わらず、図 9 (b) で最も高い改善率を示している。シナリオ 2 に着目すると、トラフィックが増加するにつれて改善率が向上している。これは、シナリオ 2 において提案手法が前処理最適化により経路中のボトルネックを解消していることに起因する。図 10 にシナリオ 1、シナリオ 2 における手法 2 及び提案手法のキューイング遅延を示す。ボトルネックを経由しない経路は、システム許容量近いトラフィックを流した場合もキューイング遅延が増加しない。シナリオ 1 において提案手法は、経路中のボトルネックを解消しないため手法 2 と同様にキューイング遅延が増加しているが、シナリオ 2 では提案手法のキューイング遅延が増加しないため改善率が向上する。

4.2.5 通信環境 2 の通信性能評価

手法 2 に対する提案手法の改善率を図 11 に示す。図 11 (a)、(b) も通信環境 1 と同様に、どのシナリオの場合も提案手法により改善している。シナリオ 1、2 に着目すると、図 11 (a)、(b) 共にトラフィックが増加するにつれ改善率が低下している。また、図 11(b) では一度システム許容量の 20~60% の範囲で改善率が向上しているが、高負荷になるにつれ低下している。一方、シナリオ 3 に着目すると、図 11 (a) ではトラフィックを増加させた場合も改善率が 97% 以上の値を保っており、図 11(b) では、トラフィックが増加するにつれ、改善率が向上している。これらの原因は、トラフィックの増加に伴うキューイング遅延の増加である。通信環境 2 では、動画データとセンサデータに加えて背景トラフィックを流すため、システム許容量を超えるトラフィックが発生する。故にキューイング遅延は増加し、 $D_{(B,movie)}$ の大半を占める。図 12 に、シナリオ 2 とシナリオ 3 における手法 2 と提案手法のキューイング遅延を示す。シナリオ 2 では、提案手法でボトルネックを経由しないものの、システム許容量を大きく超えるトラフィックが発生する場合はキューイング遅延は増加していく。そのため、システム許容量の 20~60% の場合は提案手法のキューイング遅延は上がらないが、システム許容量の 80% のトラフィックを流した時、キューイング遅延は増加する。故にトラフィックが増加するに連れ改善率は下がる。また、シナリオ 3 では、提案手法の共通前処理の最適化により手法 2

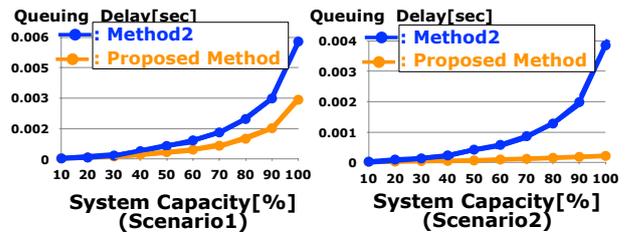
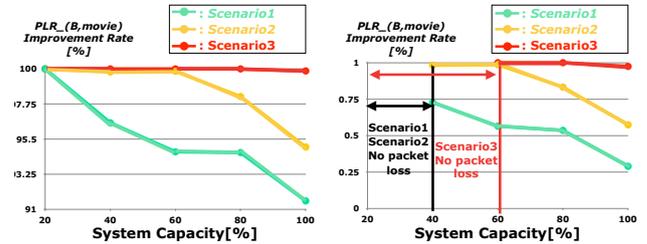


図 12 手法 2 と提案手法のキューイング遅延 (通信環境 2)



(a) 手法 1 (b) 手法 2

図 13 比較手法に対する提案手法の $PLR_{(B,movie)}$ の改善率

よりも多くのトラフィックに対するキューイング遅延の増加耐性を持つため、システム許容量の 100% のトラフィックを流した場合でも提案手法のキューイング遅延は増加しない。

最後に図 13 にシナリオ毎の $PLR_{(B,movie)}$ の改善率を示す。この結果も $D_{(B,movie)}$ と同様に、シナリオ 3 が最もよい改善率を示し、他のシナリオではトラフィックが増加するにつれて改善率が低下していることが示されている。この原因は、提案手法のキューイング遅延が上昇したことでバッファサイズが最大に達し、パケットロスが発生し始めたためである。

5. まとめと今後の検討

本稿では、GCIP 環境におけるスケーラブルなデータ収集方法を提案し、比較手法より低遅延かつ低パケットロス率で事業者に向けてデータが配送されることを示した。今後はメッシュ内の計算性能、計算負荷等の状況も考慮して最適な前処理サーバを検索する手法を検討し、実環境における通信性能、システム許容量の評価を行う。

謝辞 本研究成果は、JSPS 科研費 JP18H03234、及び国立研究開発法人情報通信研究機構の委託研究により得られたものです。

文 献

- [1] スマート IoT 推進フォーラム異分野連携プロジェクト, “異分野データ連携 H28 年度技術報告書 データでつなぐ人・モノ・コト スマート IoT 推進フォーラム異分野データ連携プロジェクト,” 2017 年 3 月.
- [2] 永田晃, 中村勝一他, “物理位置調和型アドレスに基づく Geo-Centric Networking の検討,” 2018 年電子情報通信学会総合大会, pp.51, B-6-51, 2018 年 3 月.
- [3] (株) NTT ドコモ “オープン i エリア説明書第 5.7 版,” 2012 年 4 月.
- [4] 田村暉, “位置情報に基づく IP アドレスを決定する技術,” 特許番号第 6074829 号, 2017 年 1 月 20 日.
- [5] 中村芳美, 中村勝一他, “近接空間内 IoT データのスクラム連携の実効性検証実験,” 2018 年電子情報通信学会総合大会, 2018 年 3 月.
- [6] 長尾健太郎, 塚本和也他, “異種サービス連携のための実空間連動型データ管理アーキテクチャの提案,” 2018 年ネットワークシステム研究会, 2018 年 7 月.
- [7] Salvador GarcilAa, Sergio RamilAñez-Gallego, “Big data preprocessing: methods and prospects”, Big Data Analytics, December 2016