一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CAS2019-100,CS2019-100(2020-02)

集中・分散 IoT データ共有モデルにおけるデータ取得時の遅延評価

髙橋　圭介†　　中里 秀則†　　金井 謙治‡

†早稲田大学基幹理工学部 〒169-8555 東京都新宿区大久保 3-4-1
‡ 早稲田大学理工学術院総合研究所 〒169-8555 東京都新宿区大久保 3-4-1
E-mail: †saru56suke.nb1@akane.waseda.jp , ‡ k.kanai@aoni.waseda.jp

あらまし　本稿では，標準化されている IoT プラットフォームを参考に，IoT データの管理を単一ブローカーでクラウドにて集中する管理する集中型，複数のブローカーをエッジサーバにて分散管理する分散型と定義し，その際の IoT データの要求に対するデータの応答時間（IoT データ応答時間）について，単純な数理モデルから評価を行う．評価実験では，ローカルエッジ間の通信時間，エッジクラウドの通信時間，IoT データのブローカーへのエントリ数，IoT ブローカーへのアクセス負荷をそれぞれ変更させて性能評価を行う．

キーワード　集中型システム，分散型システム，自律分散型システム，IoT プラットフォーム，IoT ブローカー

End-to-End Response Time Evaluations in Centralized and Distributed IoT Data Sharing Model

Keisuke TAKAHASHI†　　Hidenori NAKAZATO†　and　Kenji KANAI‡

†School of Fundamental Science and Engineering, Waseda University
‡ Waseda Research Institute for Science and Engineering

3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan

E-mail: †saru56suke.nb1@akane.waseda.jp, ‡ k.kanai@aoni.waseda.jp

Abstract　In this paper, we model the current standardized IoT platform as a centralized architecture and a distributed architecture to analyze end-to-end IoT data retrieval time by constructing a simple numerical model. In our definition, the centralized/distributed architecture represents IoT data sharing by one/distributed IoT broker(s) that located on the cloud/edge servers. Based on the numerical model, we evaluate end-to-end IoT data retrieval time of these two architectures under various parameters, such as the latency between IoT device and edge server, the latency between edge and cloud servers, number of IoT data entry in the IoT broker(s) and access load of the IoT broker(s).

Keywords　Centralized system, Distributed system, Autonomous distributed system, IoT platform, IoT broker

1. 研究背景

近年，Internet of Things (IoT)の普及に伴い，インターネットに接続される IoT デバイスが急速に増加している．特に，Cisco では，[1]にて，2022 年までに全世界にて接続される IoT デバイス（M2M デバイス）は，2017 年の約 2.4 倍の 146 億台になると予測している．また，IoT サービスにおいても，交通，防犯，行政，決済など，多岐に渡り，これら異分野のサービスが横断的に統合しサービス展開されていくことが予想されている．

このような背景を踏まえ，特に異分野のサービスを横断的に展開するためには，IoT デバイスが生み出すロウデータのみならず，そのデータが処理された後生成される「知見」の各サービス間における共有が重要視されている．さらに，近年の深層学習の急速な発展，普及に伴い，大量の学習データが必要とされていることから，データ共有基盤の需要は今後ますます高くなると言える．

このようなデータ共有基盤は，現在，特にスマートシティ分野を中心に，FIWARE[2]や oneM2M[3]をはじめ IoT プラットフォームという形で様々な標準化が進められており，すでに実証試験が進められている．しかしながら，これら IoT プラットフォームにおけるデータ共有は，あくまで同一の IoT プラットフォームに閉じており，異なる IoT プラットフォーム間でのデータ共有までは想定されていない．

これに対して，本稿の筆者らは，総務省の日欧国際共同研究として「スマートシティアプリケーションに拡

張性と相互運用性をもたらす仮想 IoT・クラウド連携基盤の研究開発（Fed4IoT)」の研究開発[4]を進めている．本 Fed4IoT プロジェクトでは，異なるスマートシティ間，異なる IoT プラットフォーム間におけるデータ共有を目的としている．その際，単純に IoT デバイスが生みだす“データ（あるいは知見）”を共有するのではなく,データが持つメタデータを含めた“セマンティック”も共有することを目的としている.このセマンティックの共有によって，真に異分野サービスにおけるデータ共有を可能としている．

本稿では，本 Fed4IoT が提案しているアーキテクチャ[5]に対して，IoT データの管理を単一ブローカーで集中管理する集中型，複数のブローカーで分散する分散型に対して，その際の要求データに対する応答時間の遅延時間について，単純な数理モデルから分析を行う．その際，クラウド，エッジの通信時間，IoT データのエントリ数等を変化させ，集中型，分散型の応答遅延特性を評価する．また，分散型については，どのブローカーに IoT データを格納しているのか，そのデータディスカバリを解決するための IoT ディスカバリノードを別途配置した場合と IoT ディスカバリを不要とする理想的な場合（情報指向型ネットワークを想定）とで，その応答遅延特性についても比較評価する．

2. 関連研究

2.2. IoT プラットフォーム

1 章で挙げたように，IoT データ共有基盤として，IoT プラットフォームの標準化が進んでいる．ここでは，筆者らの Fed4IoT プロジェクトに対して密接にかかわる FIWARE[2]と oneM2M[3]について紹介する．FIWARE[2, 7]は，高度な IoT データ検索および分散データ管理を可能としている IoT プラットフォームの一つで，データ共有のためのデータアクセスのインターフェース (Next Generation Application Interface(NGSI)-9, 10)およびデータのセマンティックモデル(NGSIv2, NGSI-Linked Data)を標準化している．FIWARE のアーキテクチャとして Context Broker ノードと IoT Discovery ノードが存在する．Context Broker は NGSI-10 のインターフェースを持ち，主に，アプリケーションの要求に対して､センサーデータを転送する機能(ブローカー）を持つ．IoT Discovery は NGSI-9 のインターフェースを持ち，アプリケーションからの IoT データ要求に対して，その IoT データを取得可能とする適切なブローカーを探索する機能（ディスカバリ）を持つ．これらの標準化されたインターフェースによって異分野サービス間における FIWARE 内のデータ共有を可能としている．FIWARE のオープンソース実装として，Orion Context Broker [8]が挙げられる．

oneM2M[3, 9]は，FIWARE と同様に IoT プラットフォームの一つで，水平統合型の IoT プラットフォームに位置する．oneM2M は IoT デバイスやサービス側に位置する AE (Application Entity)と，IoT プラットフォームの共通機能の集まりである CSE (Common Services Entity)が規定されている．特に，CSE はクラウド，エッジ，ゲートウェイなどに配置され，データ管理，デバイス管理，セキリュティなどの 12 の共通機能を提供する．oneM2M では，必ず CSE 同士でデータ交換を行うことで，CSE が FIWARE におけるブローカーとディスカバリ機能を持つと言える．oneM2M のオープンソース実装として，OpenMTC[10], OM2M[11]を初め数多く提供されている．また，[6]では，oneM2M の筆者らの独自実装を試みており，別のオープンソース実装である OM2M とその性能比較を行っている．

3. IoT プラットフォームの IoT データ応答時間モデル

本章では，IoT データ共有を可能とする IoT プラットフォームを想定し，その IoT データ共有システムにおいて，IoT データ要求からその応答時間におけるエンドツーエンド応答時間モデルを検討する．その際，集中型，分散型アーキテクチャに分類し，それぞれの IoT データ応答時間モデルを検討する．なお，Fed4IoT プロジェクトで提案している IoT プラットフォームのアーキテクチャは，分散型アーキテクチャ（自律分散型を含む）を想定している．

3.1. 想定共通ネットワークモデル

想定する共通のネットワークモデルを図 1 に示す．図に示すように，ユーザ端末はローカル側に配置し，エッジサーバとは Wi-Fi，4G，5G といった無線ネットワークで接続されており，エッジネットワークとクラウドサーバ間は，インターネットを介して有線ネットワークで接続されていることを想定する．各ローカルエッジ間，エッジクラウド間の通信遅延をそれぞれ T_{edge}, T_{core} として定義する.図 1 の共通ネットワークモデルに対して，IoT データ転送を担うブローカーノード，IoT データディスカバリを担う IoT ディスカバリノードをマッピングする．また，各ノードでのデータ管理は，データベースで管理することを想定する．以降において，本稿では，IoT ブローカーおよび IoT ディスカバリにおいて，データの検索時間またはブローカーの検索時間としてそれぞれモデル内に項を設定しているが，これらはブローカーあるいはディスカバリの処理時間とみなすことも可能である．なお，本モデルでは,ストリーミングデータは考慮しておらず,1 つの IoT データの要求メッセージに対する応答をモデル

化している．

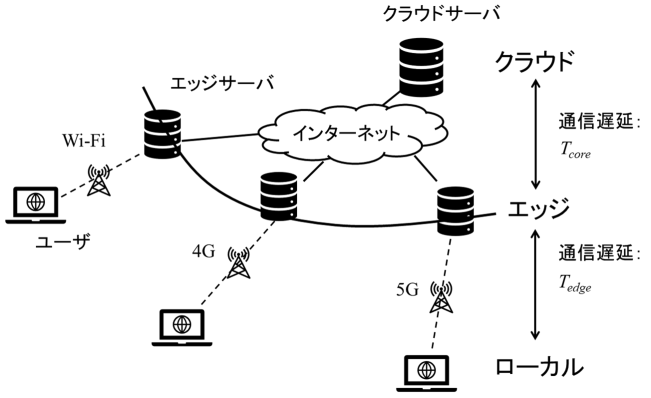

図 1. 想定するネットワークモデル．

3.2. 集中型アーキテクチャにおけるモデル化

まず，集中型アーキテクチャにおける各ノードの配置図およびデータフローを図 2 に示す．図に示すように，エッジサーバはルータとなっており，クラウドサーバが IoT ブローカーおよびディスカバリの機能を持つ．この時，IoT ブローカーが持つデータベースに全 IoT データを格納しているため，IoT データの要求に対して，その検索時間のみがかかる（ブローカーのディスカバリは無し）．そのため，IoT データの取得までのフローは，ユーザの IoT データ要求がエッジサーバを通り（単一の）クラウドサーバまで到達し，データ検索後，必要な IoT データがユーザまで返ることとなる．そのため，集中型アーキテクチャにおける IoT データの要求から応答までのエンドツーエンド遅延時間 $T_{response}$ は次の(1)式で表現できる．

$$T_{response} = T_{edge} + T_{core} + T_{data\_search} \quad (1)$$

ただし，T_{edge}, T_{core} はローカルエッジ間，エッジクラウド間の往復通信遅延，T_{data_search} は，IoT データの検索時間をそれぞれ表す．

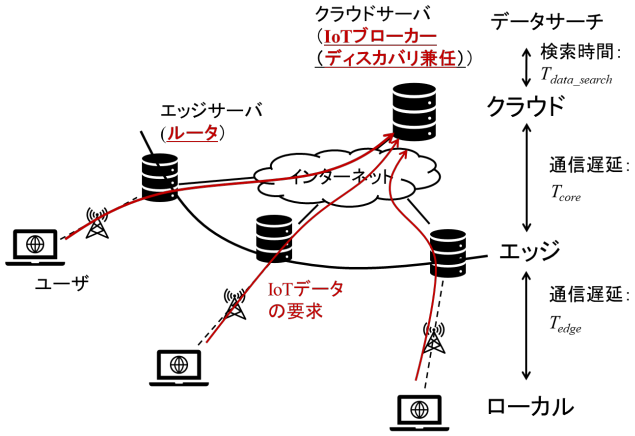

図 2. 集中型アーキテクチャの概要図．

3.3. 分散型アーキテクチャにおけるモデル化

次に，Fed4IoT で想定している分散型アーキテクチャにおける各ノードの配置図およびそのデータフローを図 3 に示す．本アーキテクチャでは，IoT データはエッジクラウドの IoT ブローカーにて分散管理されており，各 IoT ブローカーを管理する IoT ディスカバリノードはクラウドに配備されていることを想定する．前提条件として，ユーザは，IoT データの要求に対して，どの IoT ブローカーに格納されているかは分からないこととする．そのため，IoT データ要求に対して，まずは，IoT ブローカーの探索（ブローカーディスカバリ）が必要となり，その完了後，再度，指定の IoT ブローカーへ IoT データ要求を行うこととなる．つまり，分散型アーキテクチャにおける IoT データの要求から応答までのエンドツーエンド遅延時間 $T_{response}$ は，IoT ブローカーの探索フェーズ $T_{discovery}$ と IoT データの要求フェーズ $T_{request}$ の 2 フェーズの合計となり，次の(2)式にて表現できる．

$$\begin{aligned} T_{response} &= T_{discovery} + T_{request} \\ T_{discovery} &= T_{edge} + T_{core} + T_{broker\_discovery} \\ T_{request} &= T_{edge} + T_{data\_search} \end{aligned} \quad (2)$$

ただし，$T_{broker\_discovery}$ は，IoT ブローカーの探索時間を表す．

なお，分散型においては，集中型アーキテクチャと比較して，一般的にデータの分散管理の効果により，IoT ブローカーにおけるデータの検索時間（T_{data_search}）は大幅に抑えられるが，IoT ブローカーの探索時間が余剰にかかってしまう．

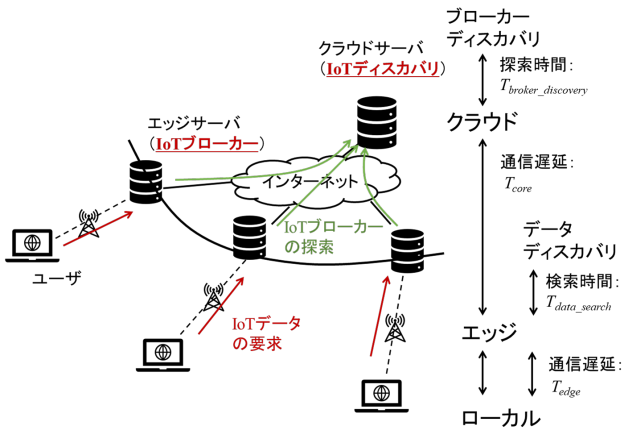

図 3. 分散型アーキテクチャの概要図．

3.4. 自律分散型アーキテクチャにおけるモデル化

最後に，先と同様に Fed4IoT で想定している自律分散型アーキテクチャにおける各ノードの配置図およびそのデータフローを図 4 に示す．図に示すように，本アーキテクチャでは，IoT ブローカーと IoT ディスカ

バリは，各エッジサーバにて実施されることを想定している．特に，IoT ディスカバリは，情報指向型ネットワークで検討されている名前によるルーティングにより，ネットワーク内で自律的に解決されていることを想定する．そのため，分散型アーキテクチャの理想形に位置し，分散型アーキテクチャの応答時間モデル（(2)式）において，IoT ブローカーの探索フェーズは無視できるものとする（$T_{discovery} = 0$）．その結果，自律分散型アーキテクチャにおけるエンドツーエンド遅延時間 $T_{response}$ は，IoT データの要求フェーズ $T_{request}$ のみの次の(3)式にて表現できる．

$$T_{response} = T_{request}$$
$$T_{request} = T_{edge} + T_{data\_search} \quad (3)$$

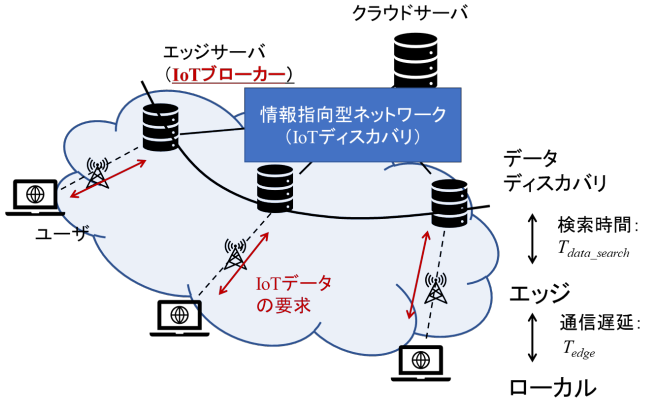

図 4. 自律分散型アーキテクチャの概要図．

以上より，(1), (2), (3)式にて各アーキテクチャにおける IoT データ要求からその応答までの IoT データ応答時間モデルとする．

4. IoT データ応答時間モデルによる評価実験

4.1. データ検索時間の実機評価

前章でまとめた応答時間モデル(1), (2), (3)式を利用した分析を実施するにあたり，特に重要な項であるデータ検索時間について，MySQL を利用した実機計測を行う．データ検索時間は，データベースにおけるデータエントリ数およびデータベースへのアクセス負荷に依存するため，これら 2 つのパラメータを変更して実験を行う．具体的には，データベースの IoT データのエントリ数として，10000 から 100000 個に変化させ，データベースへのアクセス負荷として，MySQL へのクエリ発生数について，平均到着時間を 0.04 ~ 0.1 秒まで変化させたポアソン到着分布に従い発生させた．

各データエントリ数，平均到着時間に対して，連続 10000 回実行し，その平均値を図 5 に示す．なお，図 5 における x 軸のアクセス負荷は，平均到着時間の逆数を示している．図より，アクセス負荷が小さい間は，エントリ数に依らずデータ検索時間はあまり変化していないが，アクセス負荷が大きくなると，エントリ数の増加につれ，データ検索時間が指数的に増加している様子がわかる．これは，アクセス集中に伴うデータ検索による待ち時間が発生するためである．

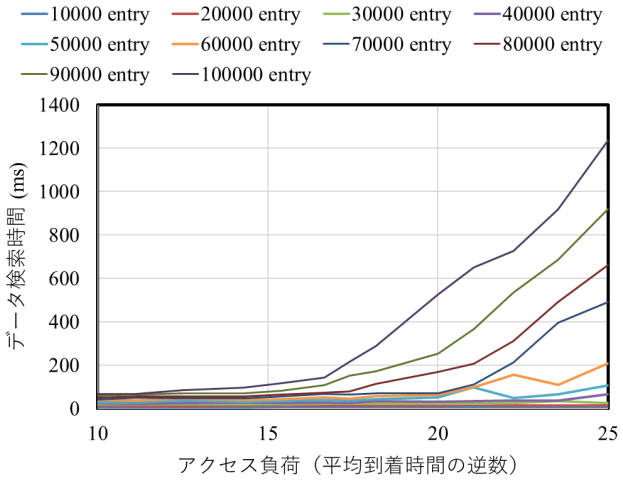

図 5. アクセス負荷に対するデータ検索時間の実機計測結果．

4.2. IoT データ応答時間の評価シナリオ

4.1 節のデータ検索時間の実機計測結果を踏まえ，(1), (2), (3)式による集中型，分散型，自律分散型アーキテクチャにおける IoT データ応答時間の理論分析評価を行う．3 章でモデル化したように，IoT データ応答時間は，通信時間とデータ検索（探索）時間の 2 つの項で成り立つ．通信時間については，IoT ブローカーノードの配置場所に依存するため，ローカルエッジ間，エッジクラウド間の通信時間 T_{edge}, T_{core} をそれぞれ変化した際の評価を行う．また，データ検索時間 T_{data_search} は，4.1 節の実機計測結果に従い，エントリ数が大きい際のアクセス負荷を変化させた場合について，評価を行う．以上，T_{edge}, T_{core}, T_{data_search} をそれぞれ変化させる 3 つの評価シナリオにて，(1), (2), (3)式を利用してそれぞれのアーキテクチャにおける IoT データ応答時間を算出する．なお，分散，自律分散型については，ブローカー数を 3 つと設定し，各ブローカーに格納する IoT データのエントリ数は，一律に集中型の 1/3 とする．

表 1: 各シナリオに設定するパラメータ値

評価シナリオ	T_{edge} (ms)	T_{core} (ms)	Entry 数	アクセス負荷	$T_{broker\_discovery}$ (ms)
①	0~100	150	90000	25	0.2
②	10	0~500	90000	25	0.2
③	10	150	90000	10~25	0.2

4.3. IoT データ応答時間の評価結果

4.3.1. ローカルエッジ間の通信時間 T_{edge} を変化させた場合

図 6 に，ローカルエッジ間の通信時間 T_{edge} を変化させた場合の集中型，分散型，自律分散型における IoT データ応答時間を示す．(1), (2), (3)式からわかるように，ローカルエッジ間の通信時間 T_{edge} は，分散型において，IoT データの要求フェーズおよび IoT ブローカーのディスカバリフェーズにて二重に関わることから，最も影響が大きい．図 6 において，集中型は，IoT データ検索の時間が支配的になることから，最も応答時間が長くなるが，分散型，自律分散型を比較すると，IoT ブローカーのディスカバリフェーズの影響で T_{edge} の時間が増加するにつれ，その差異は大きくなることがわかる．

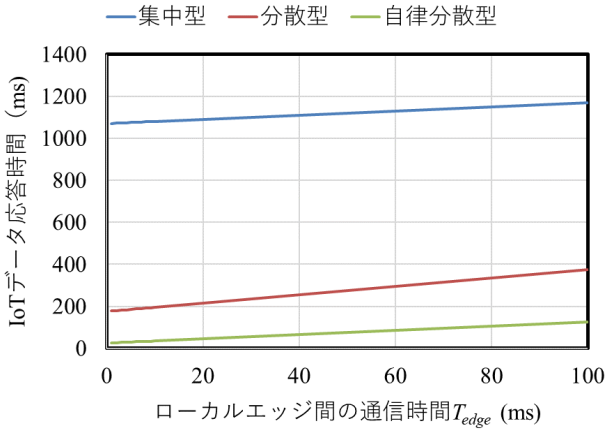

図 6. ローカルエッジ間の通信時間 T_{edge} を変化させた際の IoT データ応答時間の比較結果．

4.3.2. エッジクラウド間の通信時間 T_{core} を変化させた場合

図 7 にエッジクラウド間の通信時間 Tcore を変化させた場合の集中型，分散型，自律分散型における IoT データ応答時間を示す．(1), (2), (3)式より，集中型，分散型において，クラウドへの通信時間が発生するため影響が大きい一方で，自律分散型は，エッジ領域で簡潔するため，クラウドへの通信時間の影響を受けない．その結果，自律分散型において，他の 2 つのアーキテクチャと比較して，IoT データ応答時間のゲインが大きくなることがわかる．

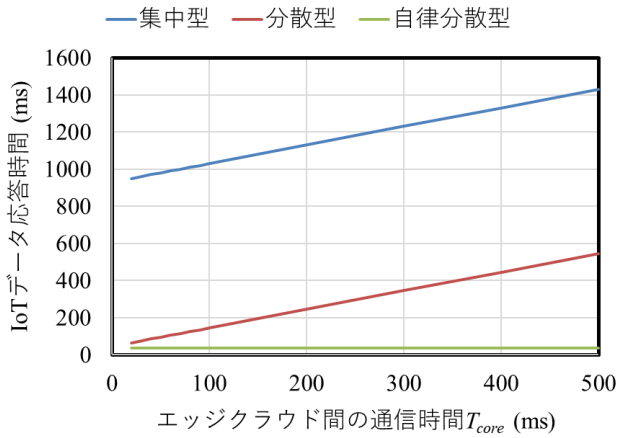

図 7. エッジクラウド間の通信時間 T_{core} を変化させた際の IoT データ応答時間の比較結果．

4.3.3 IoT ブローカーへのアクセス負荷を変化させた場合

最後に，図 8 に IoT ブローカーへのアクセス負荷を変化させた場合の集中型，分散型，自律分散型における IoT データ応答時間を示す．図 5 に示すように，エントリ数が十分に大きい場合，データ検索時間は指数的に増加していく．一方で，分散型の場合，分散させる IoT ブローカー数依存ではあるが，基本的に，分散数が大きいほど，たとえエントリ数が大きかったとしても，データ検索時間の抑制効果は大きい．そのため，集中型と比較すると，分散型，自律分散型の IoT データ応答時間のゲインは大きく，分散型と自律分散型の差異は，IoT ブローカーのディスカバリフェーズにかかる探索時間のみと言える．

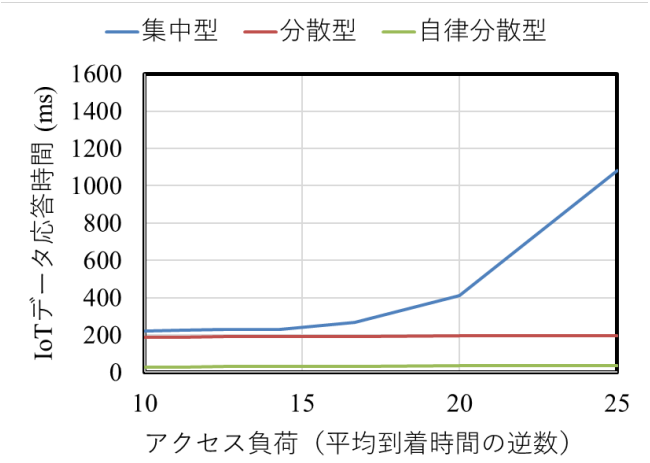

図 8. IoT ブローカーへのアクセス負荷を変化させた際の IoT データ応答時間の比較結果.

以上より，ローカルエッジ間の通信時間 T_{edge}，エッジクラウド間の通信時間 T_{core}，アクセス負荷をそれぞれ変化させた場合の集中型，分散型，自律分散型の IoT データ応答時間を単純なモデルで比較した．これらの結果からわかるように，自律分散型の IoT データ応答時間は，各パラメータの依存度が低く，安定して良好な IoT データ応答時間となることが確認され，他 2 つのアーキテクチャと比較してもそのゲインが大きいことを確認した.

5. まとめと今後の課題

本稿では、現在標準化が進んでいる IoT プラットフォームを想定し，データ共有のためのアーキテクチャとして集中型，分散型，自律分散型の 3 つのモデルを紹介し，それぞれの IoT データ応答時間について，IoT データの要求フェーズと IoT ブローカーのディスカバリフェーズの 2 つに焦点を当てモデル化した．本モデルによる IoT データ応答時間を比較分析し，自律分散型の IoT データ応答時間は，各パラメータの依存度が低く，安定して良好な IoT データ応答時間となることを確認した．今後は，この自律分散型アーキテクチャを実現するために，情報指向型ネットワークに基づくプロトコルを検討していく予定である.

謝辞

本研究成果は，戦略的情報通信研究開発推進事業（国際標準獲得型)「スマートシティアプリケーションに拡張性と相互運用性をもたらす仮想 IoT-クラウド連携基盤の研究開発 (Fed4IoT)」によるものである.

文　献

[1] Cisco Visual Networking Index Forecast and Trends, 2017-2022 White Paper [online]: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] FIWARE [online]: https://www.fiware.org/

[3] oneM2M [online]: http://www.onem2m.org/

[4] 中里秀則，“スマートシティのインターオペラビリティ”, ITU ジャーナル，vol.49，No.9，pp.6-9，2019 年 9 月.

[5] A. Detti, G. Tropea, G. Rossi, J. A. Martinezy, A. F. Skarmetay, and H. Nakazato, “Virtual IoT Systems: Boosting IoT Innovation by Decoupling Things Providers and Applications Developers”, 2019 Global IoT Summit (GIoTS), Jun.2019.

[6] 大西亮吉,笹原将章,佐藤一馬,“oneM2M 規格のオープンソースソフトウェアと自作ソフトウェアの比較”, マルチメディア，分散協調とモバイルシンポジウム 2017 論文集，vol2017,pp1040-1044,2017.

[7] 竹内崇，寺澤和幸，“データ利活用型都市経営を実現する情報プラットフォーム:FIWARE”, NEC 技法，vol.71, No.1,pp45-50，2018.

[8] Orion Context Broker [online]: https://github.com/telefonicaid/fiware-orion

[9] 原田恵，前大道浩之，山崎育生，“水平統合型 IoT プラットフォーム標準規格 oneM2M の最新動向”, NTT 技術ジャーナル，vol30，No.2，pp69-72，2018 年 2 月.

[10] OpenMTC [online]: https://www.openmtc.org/

[11] Eclipse OM2M [online]: https://www.eclipse.org/om2m/