

Things as a Service を実現する ThingVisor Factory と IP/ICN 間の Things 共有アーキテクチャの提案

金井 謙治¹⁾ 中里 秀則²⁾ 金光 永煥¹⁾³⁾

1) 早稲田理工学術院総合研究所 〒169-8555 東京都新宿区大久保 3-4-1

2) 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

3) 東京工科大学 〒192-0982 東京都八王子市片倉町 1404-1

E-mail: 1) k.kanai@aoni.waseda.jp, 2) nakazato@waseda.jp

あらまし 本稿では、IoT サービスをユーザ自身で開発し、そのサービス化 (Things as a Service) を実現するプラットフォームである「ThingVisor Factory」を紹介する。ThingVisor Factory では、GUI ベースのインターフェースを介してデザインされた IoT サービスを、サービスファンクションチェイニングによってネットワーク内に効率的に実現する機能を提供する。さらに、この IoT サービスを IP、ICN プロトコル間でも効率的に共有するために、相互のプロトコルバインディングを行う IP/ICN Proxy の紹介を行うとともに、IP/ICN 間の Things 共有アーキテクチャの初期検討を行った。

キーワード Things as a Service, IoT プラットフォーム, ICN, サービスファンクションチェイニング

ThingVisor Factory for Things as a Service and IP and ICN-based Things Sharing Architecture

Kenji KANAI[†] Hidenori NAKAZATO[‡] and Hirohide KANEMITSU[‡]

1) Waseda Research Institute for Science and Engineering (WISE), 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-0072, Japan

2) Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-0072, Japan

3) Tokyo University of Technology, 1404-1 Katakura-machi, Hachioji, Tokyo, 192-0982, Japan

E-mail: 1) k.kanai@aoni.waseda.jp, 2) nakazato@waseda.jp

Abstract In this paper, we propose “ThingVisor Factory” platform to realize “Things as a Service,” and this platform provides users the functionalities of designing, creating, deploying and sharing their own IoT services. Since the platform adopts service function chaining technology to define the IoT service as a virtual IoT service (or Virtual Thing), the platform can efficiently manage network and computing resources. In addition, to share the Virtual Things over different communication protocols, such as IP and Information Centric Networking, we propose IP and ICN-based Things Sharing Architecture that gives a functionality of IP and ICN protocol binding.

Keywords Things as a Service, IoT platform, ICN, Service Function Chaining

1. はじめに

近年、Internet of Things (IoT)の普及により、IoT のビジネス化、サービス化について、広く取り組まれている。特に、IoT サービスの多様化に伴い、IoT サービスが生み出すデータ（ログデータや知見）の共有の重要性が高まっている。その背景を受けて、異分野サービス間で効率的なデータ共有を可能とする IoT プラットフォーム技術について注目を浴びており、標準化や実用化が進められている。

IoT プラットフォームの機能として、主に IoT デバイスの生成するデータの収集、蓄積、共有を担ってお

り、データ収集、蓄積については、IoT ブローカー技術が、データ共有については、IoT ディスカバリ技術がそれぞれ位置付けられる。このブローカーとディスカバリを通して、様々なサービスを横断的に連携することを可能にしている。

現状、IoT プラットフォームは、様々なプラットフォームや標準化団体によって開発が進んでいるが、必ずしも水平統合型のアプローチで進んでいるとは言えない。プラットフォームにとっては、IoT デバイスからアプリケーションまで垂直統合することで、サービスプロバイダーやユーザを抱き込むことができるため都合がよく、サイロ化が進められている一方で、

異なる IoT プラットフォーム間で相互運用することは想定されていない[1].

この主な要因として、[2]で報告したように、IoT プラットフォーム技術においてデータ収集、蓄積を担う IoT ブローカーでは、各プラットフォームにおいて固有のデータフォーマットを定義しており、このフォーマットに従い、デバイス管理、セッション管理、データ管理を行っている。そのため、データフォーマットが固有化されていることから、異なる IoT プラットフォーム間で、IoT ブローカーを介したデータ共有や連携は困難と言える。

このような背景の中、多様なアプリケーション領域の集合体であるスマートシティでは、アプリケーション技術、IoT プラットフォーム技術、IoT デバイスの相互運用、連携が重要視されており、IoT プラットフォームを横断した相互運用技術やその基盤が求められている。

そこで、サイロ化された IoT プラットフォームに対して拡張性と相互運用性を実現させることを目的に、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 (Fed4IoT)」の研究開発[2, 3, 4]を進めている。本 Fed4IoT では、サイロ化された IoT プラットフォームが抱き込む IoT デバイス領域、IoT ブローカー領域、IoT アプリケーション領域を仮想的に切り離し、異なる IoT プラットフォームをまたいで、相互に連携を可能とする仮想 IoT システム[3]を提案している。その要素技術として、Things as a Service を実現する「IoT デバイス仮想化技術」、(仮想) IoT デバイスが生み出す“Things”をデバイスが持つセマンティック (メタ情報) を保持したまま共有する「情報共有サービス」が挙げられる。

筆者らは、これら要素技術に関連して、これまで IP、Pub/Sub ベース、情報指向型ネットワーク (Information Centric Networking (ICN)) ベースのサービスファンクションチェイニングに基づく Things as a Service の提案を行ってきた[4, 5, 6]。本稿では、これらの先行研究を踏まえ、サービスファンクションチェイニングを利用した Things as a Service を実現するプラットフォームである「ThingVisor Factory」を検討する。さらに、ThingVisor Factory を通して実現される IoT サービスに対して、IP および ICN プロトコルで提供されたことを想定し、異種プロトコル間でも効率的にサービス共有を実現する IP/ICN Proxy の検討、また、IP/ICN 間の Things 共有アーキテクチャの初期検討を行う。

2. 関連研究

2.1. IoT プラットフォーム

まず、Fed4IoT が想定している IoT プラットフォームである oneM2M[7, 8]と FIWARE[9]について、以下に、それぞれ概説する。

2.1.1. oneM2M

oneM2M[7, 8]は、水平統合型の IoT プラットフォームの標準化の一つで、主に、ETSI, ARIB, TTC が協力している。oneM2M では、共通アーキテクチャと通信方式の二つを標準化している。アーキテクチャは、非常に単純化されており、アプリケーションを表す Application Entity (AE) と共通のサービス機能群を表す Common Service Entity (CSE) がそれぞれ通信を行うモデルとなっている。特に、AE 同士が繋がることはなく、必ず通信には CSE を介するよう規定されており、CSE がアプリケーションの管理、デバイスの管理、セッションの管理等の機能を担う。これらの機能 (データモデル) は、リソースとして定義されており、このリソース操作のための REST API が提供されている。通信方式として、HTTP の他に、CoAP や MQTT が想定されている。また、リソースは木構造の階層構造で定義されており、XML や JSON で記述されている。oneM2M のオープンソース実装として、OpenMTC [10], OM2M[11]が挙げられる。

2.1.2. FIWARE

FIWARE[9]は、NECをはじめ多くの欧州の大手企業によって開発されている IoT プラットフォームの一つである。FIWARE では、アプリケーション領域の横断型のデータ利活用を目的に、各アプリケーション領域が生成するデータは、IoT ブローカー (コンテキストブローカー) を介してデータの流通を行い、IoT ディスカバリによって、デバイス、セッション、データの管理等を行う。アプリケーションと IoT ブローカー間は、NGSI-10 と呼ばれるプロトコルによって定義されており、IoT ブローカーと IoT ディスカバリ間は、NGSI-9 と呼ばれるプロトコルによって定義されている。また FIWARE では、FIWARE 内に流れるデータについて、そのデータモデルの標準化を行っており、この標準化されたデータモデルと共通プロトコルによって高度なデータ検索を可能としている。各 NGSI プロトコルは、REST によって API が定義されており、その通信には主に HTTP が利用されている。FIWARE のオープンソース実装として、Orion Context Broker[12]が挙げられる。

2.2. サービスファンクションチェイニング

これまで非常に多くのサービスファンクションチェイニングの研究はなされており、ここではその関連

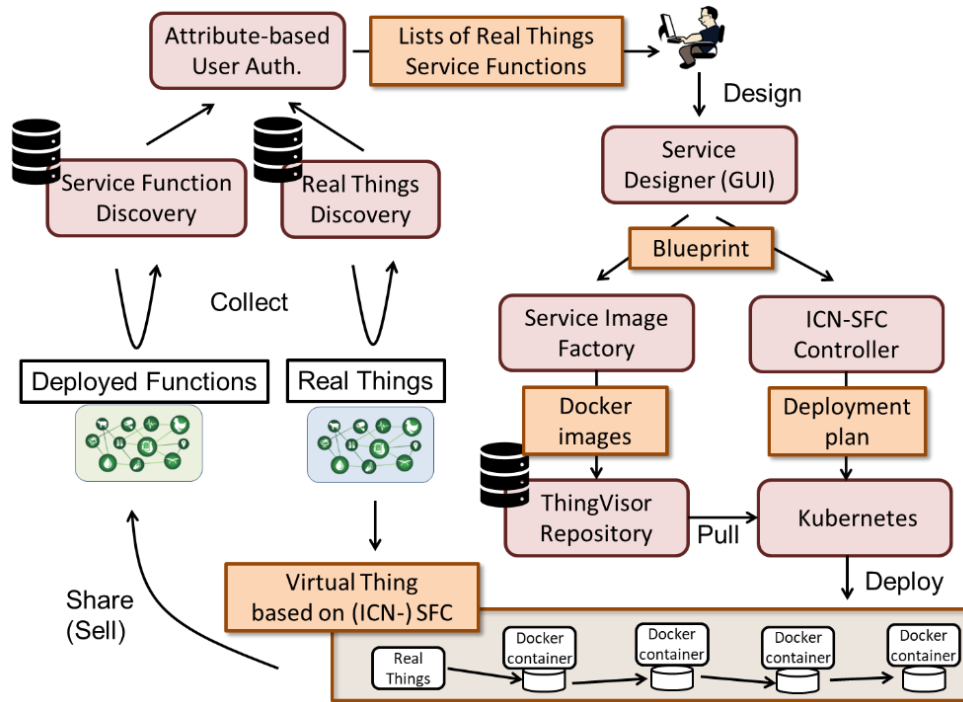


図 1. Things as a Service を実現する ThingVisor Factory の概要図.

技術についてまとめる.

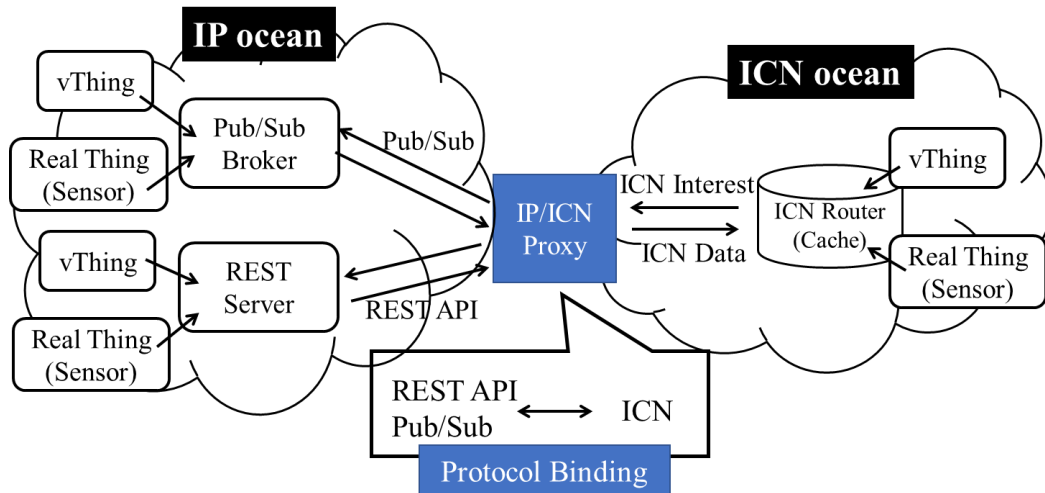
サービスファンクションチェイニング (Service Function Chaining (SFC)) は、ネットワーク仮想化技術 (Software Defined Network (SDN)) の一つとされ、類似技術として、ネットワーク機能仮想化技術 (Network Function Virtualization (NFV)) が挙げられる. SFC や NFV は、ネットワークのサーバ上にソフトウェアで(ネットワーク)サービスを実現する技術で、SFC は、この仮想化された NFV をサーバ内、サーバ間でネットワークを介して連結していきサービスを実現する技術である. 近年では、[13]で解説しているように、エッジコンピューティングとも絡め、NFV のエッジ・クラウドへの最適配備問題[4]や割り当り資源量を最適化する最適割当り問題[14]、最適リソーススケール技術[15]などが非常に盛んに行われている. このアルゴリズムとして、近年急速に発展している深層学習ベースの手法[16]も登場してきている.

SFC を実現するうえで、これら NFV をネットワーク内で結合する際のプロトコルについても重要である. 現在、IP 通信を利用するものは、パケット内に NFV を識別するための Network Service Header (NSH) タグを付与することで、そのタグを持つルータへ転送する方式が検討されており、IETF RFC7665[17]にて SFC のアーキテクチャを、IETF RFC8300[18]にて NSH タグがそれぞれ標準化されている.

これに対して、IP アドレスと NSH タグの代わりに、名前を利用して SFC のルーティングを解決する情報指

向型ネットワーク (ICN/CCN/NDN)[19]に基づく SFC も多く検討されている. 代表的なものとして、Function-Centric Service Chaining (FCSC) [20]や Named Function Networking (NFN) [21]が挙げられる. いずれも ICN に基づく Interest パケットに SFC におけるサービスファンクションの実行順を定義し、名前ベースでルーティングを解決している. [20]の提案の延長として[22, 23]にてプロトタイプ実装が検討されており、また、[20]の類似研究として ICN-FC [24]などが挙げられる. また、[25]にて ICN ベースの NFV のプロトタイプ実装を行っている.

これらの先行研究を踏まえ、筆者らも Pub/Sub メッセージモデルに基づく Topic ベースのサービスファンクションチェイニング[2]や ICN ベースのサービスファンクションチェイニング[6]を提案している. 特に、[2]では、本来 ICN のリクエストレスポンス型ではなく、IP ベースの Push 型をベースに Pub/Sub のトピックを活用することで、ルーティングを解決し、また、効率的にサービスファンクションが生み出すデータやサービスファンクションそのものの共有を実現している. [6]では、ICN-FC をベースに、ICN のオープンソース実装の一つである Named Data Networking[26]が提供するライブラリ群である ndn-cxx, Named Forward Demon をカスタマイズし、そのプロトタイプ実装を行っている. 本稿では、これら ICN-SFC に基づく Things as a Service を実現するプラットフォームである ThingVisor Factory を紹介するとともに、IP と ICN を橋渡しする



<https://www.hogehoge.org/v2/ResourceID> ↔ [icn://ThingID \(or ThingType\)](#)

[kafka://hogehogeThing/Function](#) ↔ [icn://Fuction/hogehogeThing](#)

図 3. IP と ICN のプロトコルマッピングを実現する IP/ICN Proxy の概要図.

「IP/ICN Proxy」を介した情報共有アーキテクチャの提案を行う.

GUI ベースのデータフロープログラミングに従ってサービスの開発を行う. 図 2 に示すように, IoT サービス開発として, 利用可能な Things (IoT データやオープンデータ) を選択し, それに対して, キャンバスにてドラッグアンドドロップで IoT サービスを「デザイン」していくことを想定している. 各ファンクションを矢印で「チェイン」させていくことで, IoT サービスの実現を可能とする. このように, 開発した IoT サービスは, あくまで論理的なサービスの結合しか示しておらず, Service Image Factory にて, サービスファンクションのコンテナイメージを作成する (ここでは Docker イメージを想定). 同時に, 関連研究で述べたように, 各サービスファンクションをネットワーク上に展開するにあたり, その配備プランやリソース割当を実施する必要がある. これについては, 各サービスファンクションが持つ機能要件に従い, (ICN-)SFC Controller によってその最適配備プランおよびネットワークのルーティングを解決する. これらのステップを踏み, 最終的に, ネットワーク内にサービスのイメージを展開し, SFC によって, 自分自身の IoT サービス (Things) を仮想的に実現する. このような仮想 IoT サービス (Virtual Thing (vThing)) やそのコンポーネントであるサービスファンクションを他のユーザと共有することで, Things as a Service を実現する. この共有の際に, ユーザ属性によって公開範囲を制御することを想定する.

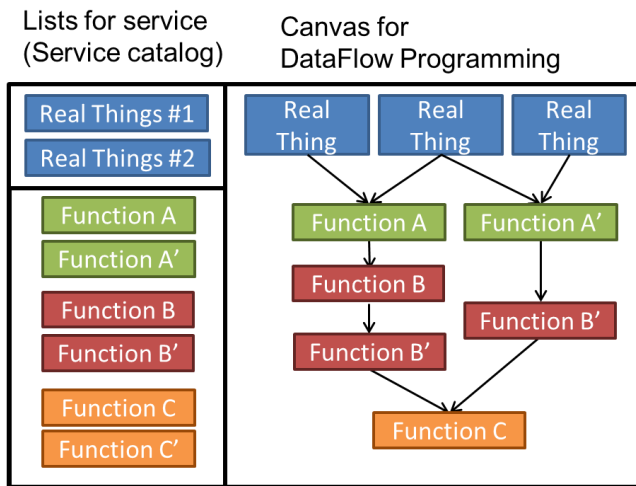


図 2. Service Designer の GUI イメージ図.

3. Things as a Service を実現する ThingVisor Factory

IoT サービスが普及している中, IoT サービスをユーザ自身でデザインする要求が増えている. パブリックにされている IoT デバイス (あるいはオープンデータ) から, 自身の “Things” をデザインしサービスする仕組みとして, ThingVisor Factory を提案する. ThingVisor Factory の概要図を図 1 に示す. 図 1 に示すように, IoT サービス開発者は, Service Designer (図 2) を通して, 自分自身の IoT サービスを開発する, Service Designer のインターフェースは, NodeRED[27] に代表される

4. IP/ICN 間の Things 共有アーキテクチャ

先に述べた ThingVisor Factory において, 仮想 IoT

サービス (vThing) を生成するためのサービスファンクショナルチェーンの通信プロトコルとして、関連研究で述べたように、IP ベースと ICN ベースが考えられる。通信プロトコルの選択制は、あくまで IoT サービス開発者に委ねる、あるいは、システムで自動的に決定した場合、IP ベースの vThing と ICN ベースの vThing の共有は必須と言える。IP ベースの場合は、vThing の流通は、IoT ブローカーや REST サーバを介して行われ、ICN ベースの場合は、vThing の流通は、主に ICN ルータのキャッシュ (Content Store) を介して行われることとなる。これらの中でデータ共有を行うためには、相互のプロトコル変換が必要となる。この機能を提供するノード (あるいは機能そのもの) を IP/ICN Proxy と定義し、IP/ICN Proxy によってプロトコルバインディングを行う。その概要図を図 3 に示す。例えば、IP 側が REST API で提供されている場合、REST API と ICN Interest メッセージとのマッピング、また、IP 側が Pub/Sub ベースの Topic 名で提供されている場合、Topic 名と ICN Interest 名とのマッピングを提供することで、それぞれのプロトコル変換が可能となる。ただし、このマッピングにおいて、お互いの命名規則が明確に定義されていることを前提とする。

さらに、IP/ICN Proxy が分散配備されている場合、IP 側にとっては、要求メッセージに対して、どの IP/ICN Proxy へリクエストを送信すればよいか、そのエンドポイント (アクセス先) を知る必要がある。これについては、FIWARE 等で検討されているように、エンドポイントを Thing Discovery ノードによって管理することを想定している (図 4)。ただし、ICN 側から提供する IoT データ (vThing) 全てを Thing Discovery ノードで登録すると、大量のデータを管理する必要があるため、例えば、ICN の名前スキーム (あるいは FIB 情報) に基づく効率的かつ自律的なデータの分散管理スキームが必須であると言える。

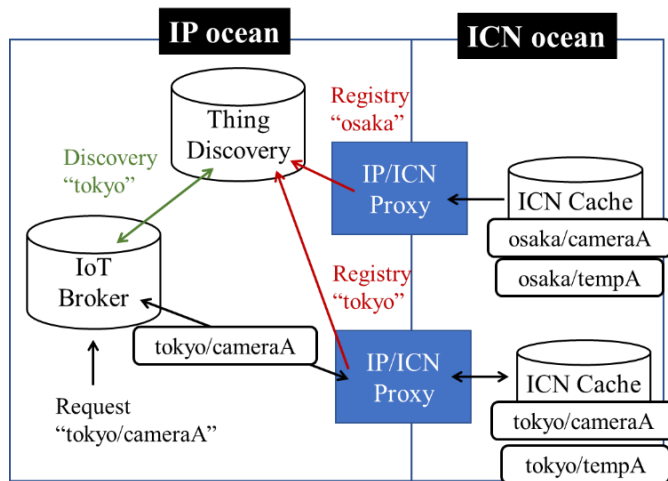


図 4. Thing Discovery による IP/ICN Proxy 探索の例。

5. まとめと今後の予定

本稿では、IoT サービスをユーザ自身で開発し、それをサービス化 (Things as a Service) を実現するプラットフォームである「ThingVisor Factory」を紹介した。ThingVisor Factory では、GUI ベースのインターフェースを介してデザインされた IoT サービスを、サービスファンクショナルチェーンによってネットワーク内に効率的に実現する機能を提供する。さらに、この IoT サービスを IP, ICN プロトコル間でも効率的に共有するために、相互のプロトコルバインディングを行う IP/ICN Proxy の紹介を行うとともに、IP/ICN 間のデータ共有アーキテクチャの初期検討を行った。

今後は、プロトコルバインディングのスキームを具体化し、また、効率的かつ自律的なデータの分散管理スキームを検討する。さらに、ThingVisor Factory のプロトタイプ実装を進めるとともに、FIWARE や oneM2M といった実際の IoT プラットフォームとの連携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)」の支援を受けている。

文 献

- [1] 長尾, 亀津, “ビジネス価値につながる IoT の実現 -最適な IoT プラットフォームをどう選択するか-”, 野村総合研究所, IT Solutions Frontier, 2018 年 2 月. [online]: https://www.nri.com/-/media/Corporate/jp/Files/PDF/knowledge/publication/it_solution/2018/02/ITSF180205.pdf?la=ja-JP&hash=F0ADC26F649E66BC83E7816B4BE9A06B7326124F
- [2] 金井, 吉田, 金光, 中里, 横谷, 向井, 中村, 上杉, “Things as a Service を実現する Fed4IoT プラットフォームの研究開発”, 電子情報通信学会 信学技報 CS2019-69, pp.39-44, 2019 年 11 月.
- [3] A. Detti, G. Tropea, G. Rossi, J. A. Martinez, A. F. Skarmeta, H. Nakazato, “Virtual IoT Systems: Boosting IoT Innovation by Decoupling Things Providers and Application Developers”, Global IoT Summit, 2019.
- [4] Hidehiro Kanemitsu, Kenji Kanai, Jiro Katto and Hidenori Nakazato, “A Function Clustering Algorithm for Resource Utilization in Service Function Chaining,” IEEE CLOUD 2019, July 2019.
- [5] Keigo Ogawa, Hibiki Sekine, Kenji Kanai, Kenichi Nakamura, Hidehiro Kanemitsu, Jiro Katto and Hidenori Nakazato, “Performance Evaluations of IoT Device Virtualization for Efficient Resource Utilization,” 2019 Global IoT Summit, Jun.2019.
- [6] 吉井, 中里, “NDN におけるファンクション・チェーンの実装”, 電子情報通信学会 信学技報

- CS2018-39, pp. 133-138, 2018 年 7 月.
- [7] 山崎, “M2M 技術のアーキテクチャ及び oneM2M 標準化動向”, 通信ソサエティマガジン No.29, 2014.
- [8] 原田, 前大道, 山崎, “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 2018 年 2 月.
- [9] 竹内, 寺澤, “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 技報, Vol.71, 2018 年 11 月, [online]:
<https://jpn.nec.com/techrep/journal/g18/n01/pdf/180110.pdf>
- [10] OpenMTC [online]: <https://www.openmtc.org/>
- [11] Eclipse OM2M [online]:
<https://www.eclipse.org/om2m/>
- [12] FIRARE-ORION [online]:
<https://fiware-orion.readthedocs.io/en/master/>
- [13] V. P. Kafle, Y. Fukushima, P. M. Julia, T. Miyazawa, and H. Harai, “Adaptive Virtual Network Slices for Diverse IoT Services,” IEEE Comm. Standards Magazine, vol.2, Issue.4, pp.33-41, Dec. 2018.
- [14] P. Wang, C. Yao, Z. Zheng, et al., “Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems”, IEEE Internet of Things Journal, vol.6, Issue.2, pp.2872-2884, April. 2019.
- [15] V. P. Kafle, Y. Fukushima, P. M. Julia, and H. Harai, “Scalable Directory Service for IoT Applications,” IEEE Comm. Standards Magazine, vol.1, Issue.3, pp.58-65, Sep. 2017.
- [16] T. Hirayama, T. Miyazawa, M. Jibiki, and V. P. Kafle, “Service Function Migration Scheduling based on Encoder-Decoder Recurrent Neural Network,” IEEE NetSoft 2019, Jun. 2019.
- [17] IETF RFC 7665 “Service Function Chaining (SFC) architecture,” Oct. 2015.
- [18] IETF RFC 8300 “Network Service Header (NSH),” Jan. 2018.
- [19] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” ACM CoNEXT, Dec. 2009, pp. 1-12.
- [20] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin, “An information centric network for computing the distribution of computations,” ACM ICN 2014, pp.137-146, Sep.2014.
- [21] M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, “Exploiting icn for flexible management of software-defined networks,” ACM ICN 2014, pp. 107-116, Sep.2014.
- [22] M. Arumaithurai, J. Chen, E. Maiti, X. Fu, and K. K. Ramakrishnan, “Prototype of an ICN Based Approach for Flexible Service Chaining in SDN,” IEEE INFOCOM Workshop, May. 2015.
- [23] S. G. Kulkarni, M. Arumaithurai, A. Tasiopoulos, Y. Psaras, K. K. Ramakrishnan, X. Fu, and G. Pavlou, “Named Enhanced SDN Framework for Service Function Chaining of Elastic Network Functions,” IEEE INFOCOM Workshop, Apr. 2016.
- [24] L. Liu, Y. Peng, M. Bahrani, S. Mnatsakanyan, G. Qu, Z. Ye, H. Guo, “ICN-FC: An Information-Centric Networking Based Framework for Efficient Functional Chaining,” IEEE ICC 2017, May 2017.
- [25] L. Bracciale, P. Loreti, A. Detti, R. Paolillo, and N. B. Melazzi, “Lightweight Named Object: An ICN-Based Abstraction for IoT Device Programming and Management,” IEEE Internet of Things Journal, vol.6, no.3, Jun. 2019.
- [26] Named Data Networking Project [online]:
<https://named-data.net/>
- [27] NodeRED [online]: <https://nodered.org/>