

非同期型な ICN サービスファンクションチェイニングの 提案とその実機検証

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、オンデマンドな IoT デバイス仮想化を実現する ThingVisor Factory の主要な機能である情報指向ネットワークを利用したサービスファンクションチェイニング (ICN-SFC) について紹介する。その際、ICN-SFC について、先行研究として提案されている同期型の ICN-SFC と筆者らの提案である非同期型の ICN-SFC について、実装例も含めて示す。ICN-SFC の性能評価として、研究室内に構築した環境下において、オープンソースの Python ベースの ICN ソフトウェアを利用し、ICN-SFC の性能検証および非同期型 ICN-SFC の有効性を実機評価する。

キーワード Things as a Service, ICN, サービスファンクションチェイニング, 仮想化技術

Proposal of Asynchronous ICN Service Function Chaining and its Verification in Real Environment

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 introduce Information Centric Networking-based Service Function Chaining (ICN-SFC) that is one of major components of “ThingVisor Factory” platform to realize “Things as a Service (and IoT device virtualization.” In ICN-SFC, we introduce synchronous ICN-SFC as a previous work and asynchronous ICN-SFC as our proposal including their implementation ways. In addition, we implement ICN-SFC by using an open source Python-based ICN software and evaluate the performance of ICN-SFC in real environment.

Keywords Things as a Service, ICN, Service Function Chaining, Virtualization technologies

1. はじめに

近年、Internet of Things (IoT) の普及により、IoT サービスのビジネス化について注目が高まっている。特に、深層学習を含む AI 技術の急速な発展に伴い、IoT サービスが生み出すデータ（ログデータや知見）がビジネス面においても非常に重要視されており、IoT データ収集を容易とする IoT プラットフォームの導入が広く進んでいる。

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められているが、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。前述したように、IoT データの重要性の高まりから、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ことから、サイロ化が進められている[1]。

このような背景の中、異分野領域のサービスを総合的に展開することを目指しているスマートシティにおいては、IoT プラットフォームの垂直統合によるサイロ化ではなく、水平統合するアプローチが望まれており、アプリケーション、プラットフォーム、デバイスと各領域において連携、相互運用することが求められている。

筆者らはこれまで 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、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]

に取り組んできている。Fed4IoT では、異なる IoT プラットフォーム間に相互運用、連携を提供する仮想 IoT システム (VirIoT) [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」や異なる IoT プラットフォームが生成する IoT データを共有する「コンテキスト情報共有技術」が挙げられる。

本稿ではこれら要素技術の中でも IoT デバイス仮想化技術に焦点を当て、オンデマンドな IoT デバイス仮想化を実現する ThingVisor Factory[4]の主要な機能を紹介する。さらに、その要素技術である情報指向ネットワークを利用したサービスファンクションチェイニング (ICN-SFC) について、先行研究として提案されている同期型の ICN-SFC と ThingVisor Factory のアーキテクチャを活用した非同期型の ICN-SFC についても紹介する。また、研究室内に構築した環境下において、オープンソースの Python ベースの ICN ソフトウェアを利用し、ICN-SFC の性能検証および非同期型 ICN-SFC の有効性を実機評価する。

2. 関連研究

2.1. 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、[8]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[5]や割りリソース量を最適化する最適割当問題[9]、最適リソーススケールリング技術[10]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[11]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[12]にて SFC のアーキテクチャを、IETF RFC8300[13]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)[14]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) [15]や Named Function Networking (NFN) [16]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。[17]の提案の延長として[18, 19]にてプロトタイプ実装が検討されており、また、[17]の類似研究として ICN-FC [20]などが挙げられる。また、[21]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[6]や ICN ベースのサービスファンクションチェイニング[7]を提案している。特に、[6]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。[7]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[21]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、[6, 7]とは異なり、NICT が開発している ICN のソフトウェア実装の一つである Cefore[22, 23]および Python ライブラリである Cefpyco[23]を利用して ICN-SFC を実現する。なお、Cefore は Content Centric Networking [14]をベースにしている。

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

本章では、筆者らが開発している Thing as a Service を実現する ThingVisor Factory について紹介する。

3.1. ThingVisor: Thing Hypervisor [3]

図 1 に Thing Hypervisor のコンセプトを示す。Thing Hypervisor (ThingVisor) は、コンピュータ仮想化技術の一つとして知られている Hypervisor 型仮想化のコンセプトを IoT デバイスへ適応したものに位置付けられる。図に示しているように、IoT アプリケーションでは、IoT デバイスが生成するローデータよりもその処理結果が重要視され、また、異なる IoT アプリケーション間においても IoT デバイスが生成するローデータの使われ方は多様である。そのため、IoT アプリケーションにとっては、物理 IoT デバイスそのものが重要では

なく、IoT デバイスが生成、処理されたデータが重要と言える。そこで Thing Hypervisor (ThingVisor) は、アプリケーション側から見て、物理 IoT デバイスからのデータ取得や処理を隠蔽し、仮想的な IoT デバイスを生成する仕組みを提供する。ThingVisor によって、IoT アプリケーションからは、自身の用途に適合した IoT デバイスが仮想的に見えており、この仮想 IoT デバイ스에接続するだけで良くなり、アプリケーション開発の簡便化、さらには、運用性の向上が期待できる。

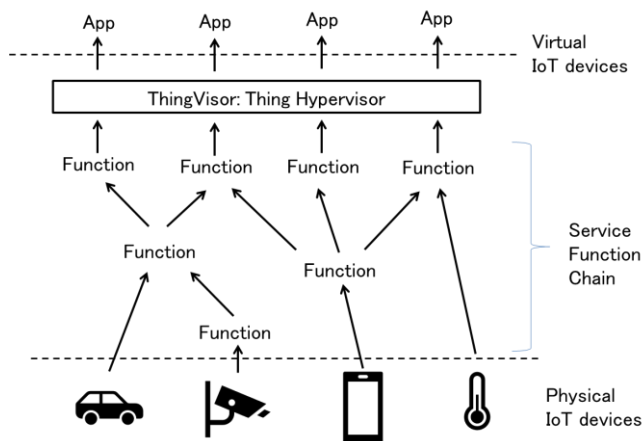


図 1. Thing Hypervisor (ThingVisor) のコンセプト。

一方で、多様な IoT アプリケーションの要件を満足するためには、ThingVisor を要求に合わせて (自動) 生成する仕組みが必要なるが、ThingVisor Factory は、オンデマンドに ThingVisor を生成するプラットフォームに位置する。また、仮想 IoT デバイスとして IoT データの取得や処理については、ICN-SFC により実現するものとする。

3.2. ThingVisor Factory

[4]で紹介した ThingVisor Factory の機能拡張について説明する。図 2 に ThingVisor Factory の主要な機能を示す。図に示すように、ThingVisor Factory は、主要な機能として、Service Designer, Service Image Factory, ThingVisor Factory Controller, Service Chain Manager, Service Deploy Manager の機能を持つ。Service Designer は、NodeRED[24]の拡張版の位置づけで、GUI を持つアプリケーション開発者とのインターフェース機能を提供する。アプリケーション開発者は、Service Designer の GUI を通して、自身のアプリケーションを開発し、Service Designer は、開発されたアプリケーションに従い、JSON 形式にてサービスファンクションチェーンの定義ファイルを生成する。この時、提案する Service Designer では、NodeRED と異なりサービスファンクション間を繋げるネットワークプロトコル (例: Pub/Sub, ICN) を指定することができる。次に、

Service Image Factory は、開発者によって定義されたサービスファンクションチェーンに従い、必要に応じてサービスファンクションの Docker Image を生成し、レポジトリ (例: Docker Hub) へ登録を行う。この際、Service Designer を通して、アプリケーション開発者自身でサービスファンクションの実装も可能とする。次に、ThingVisor Factory Controller は、ThingVisor の機能を実行する API を提供する。また、前述したように、ThingVisor Factory は、ThingVisor を要求に合わせて動的に作成するプラットフォームとなり、Fed4IoT の提案する仮想 IoT システム (VirIoT) との連携も意識している。そのため、VirIoT の Master Controller とは、ThingVisor Factory Controller を介して VirIoT 側の API を呼び出すものとする。次に Service Deploy Manager は、必要となるサービスファンクションやサービスファンクションの候補先の通信資源、計算資源の情報に従い、サービスファンクションのネットワーク内の (ICN) ノードへの配備先を決定する。実際の配備は、Kubernetes を利用することを想定する。最後に、Service Chain Manager は Service Deploy Manager によって導出されたサービスファンクションの配置位置に従い、それぞれサービス間の初期経路設定やネットワーク変動に応じた動的な経路制御機能を提供する。

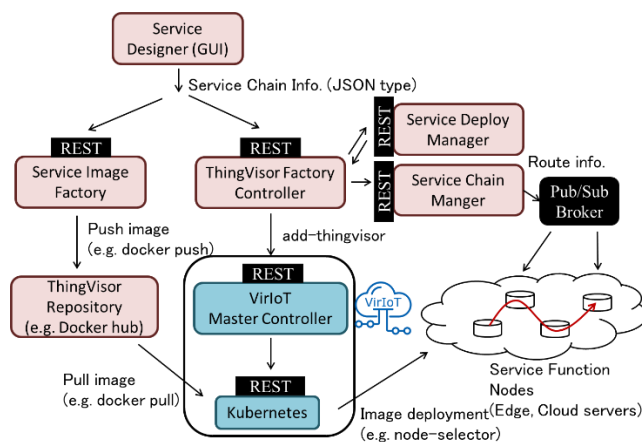


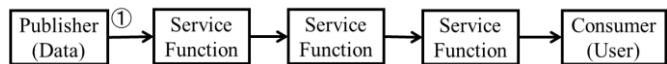
図 2. ThingVisor Factory の主要な機能構成図。

4. ICN サービスファンクションチェーンニング

4.1. 従来の ICN-SFC (同期型)

ICN でサービスファンクション実現するためには、従来の P2P 型のサービスファンクションの仕組みをそのまま適用することは難しい。これは、ICN プロトコルは、基本的にリクエストレスポンス型のプロトコルを採用していることに起因する。図 3 に示すように、従来の P2P や Pub/Sub 型では、処理の対象となるデータを持つ Publisher (IoT デバイス) からデータを PUSH 配信することが可能である。一方、リクエストレスポンス型の ICN では、Consumer からのリクエスト

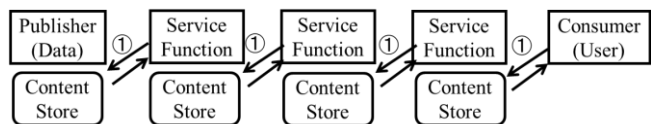
(Interest) メッセージが起因になり、処理をするべき Service Function を一度経由して、要求データ元に届いた後、Interest が辿った経路を逆に戻る形でサービスファンクションチェイニングが実行される。そのため、各ノードが同期して通信することから同期型 ICN-SFC と言える。そのため、サービスファンクションを重ねるごとに、遅延が増大することとなる。



(a) P2P, Pub/Sub 型



(b) 従来の ICN 型 (同期型)



(c) 提案する ICN 型 (非同期型)

図 3. サービスファンクションチェイニングのデータフロー図

4.2. 提案する ICN-SFC (非同期型)

同期型 ICN-SFC の前提として、各サービスファンクションノードが送信する Interest メッセージが Consumer から発行されない限り分からない、という点にある。これに対して、ThingVisor Factory プラットフォームにおいては、Service Chain Manager の機能によって、各サービスファンクションノードが発行する Interest メッセージを把握することができるため、各サービスファンクションノードは非同期に実行することが可能となる。さらに、図 3 に示すように、ICN の Content Store (キャッシュ機能) を有効活用することで、サービスファンクションノードは自身のキャッシュへコンテンツを発行するのみで、他のサービスファンクションノードから独立して動作可能となる。このように、各サービスファンクションノードが非同期に動作することから非同期型 ICN-SFC と言える。

4.3. ICN-SFC の実装例

最後に、ICN-SFC のソフトウェア実装例を図 4 に示す。図に示すように、ICN でサービスファンクションチェイニングを実装する際、大きく 4 つの状態に分割することができる。

1. リクエスト待ち状態: Service Function を実行するための Interest メッセージを待ち受ける状態と定義する。

2. コンテンツの取得状態: サービスファンクション要求を受け取った後、必要となるデータを取得する状態と定義する。
3. サービスファンクション呼び出し状態: サービスファンクションを呼び出し、処理結果を待ち受けている状態と定義する。
4. 処理結果のパブリッシュ状態: サービスファンクションによって処理された結果を Content Store へパブリッシュする状態と定義する。

以上の 4 状態に基づき、ICN-SFC の実装を行う。なお、同期型と非同期型の違いは、リクエスト待ち状態に入るか入らないかの違いとなる。非同期型の場合は、リクエスト待ち状態には入らずに、コンテンツの取得状態に入ることが可能とする。

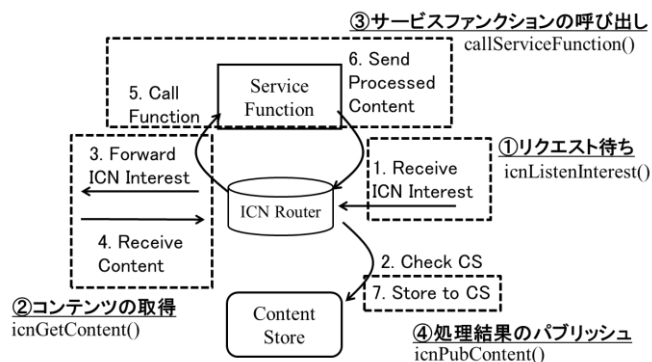


図 4. ICN-SFC 実現のための 4 状態の実装例

5. ICN-SFC の性能評価

5.1. 実機環境と評価シナリオ

本章では、提案している ICN-SFC の性能評価を行う。図 5 に示すように、研究室内のローカルネットワーク上に 5 台のベアボーン PC を利用し、ICN ルータおよびサービスファンクションをソフトウェア実装した。ICN のソフトウェアとしては、NICT が開発している Cefore, Cefpyco を利用して ICN-SFC を実装した。サービスファンクションとして、カメラ、物体検知、顔検知、人物カウントの 4 つを想定している。ただし、単純化のため、各サービスファンクションはダミーとし、各サービスファンクションで乱数にて値を発生させ、サービスファンクションの処理時間はほぼ無視できるものとし、ICN-SFC のプロトコル面に焦点を当てて評価することとする。各サービスファンクションは、図 5 に示すような Interest 名で呼び出し可能な状態としている。

評価シナリオとして、Consumer (ユーザ) は、カメラの画像から物体検知、顔検知、人物カウントを要求するものとする。そのため、Consumer 側から Interest メッセージとして、「ccn:/counter/face/yolo/camera」を

送信する。同期型 ICN-SFC の場合、4.1 節で説明したように、送信された Interest メッセージは、人物カウント、顔検知、物体検知、カメラへ到達しカメラの画像データを取得し、逆順のノードをたどって処理を実行しつつ Consumer へ最終の結果が帰ることとなる。

一方、非同期型は、それぞれの ICN ノードは、自身が処理に必要とするデータを把握しているため、それぞれが適切な Interest メッセージを送信し必要なデータを取得することが可能となり、処理結果については、自身の Content Store へキャッシュする。そのため、Consumer から送信された Interest メッセージは、自分カウントの Content Store へ 1 ホップで到達し、キャッシュされているデータを取得する。

以上の流れにおいて、各サービスファンクションにおけるデータ取得にかかる遅延時間を評価する。その際、ノード間の往復遅延時間の影響を検証するために、tc コマンドにより、10, 20, 40, 80ms の往復遅延を発生させている。

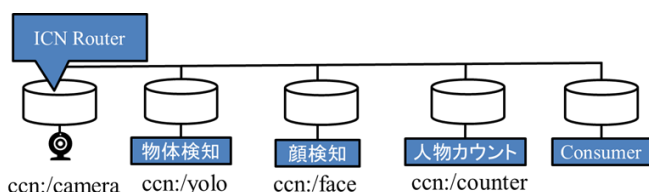
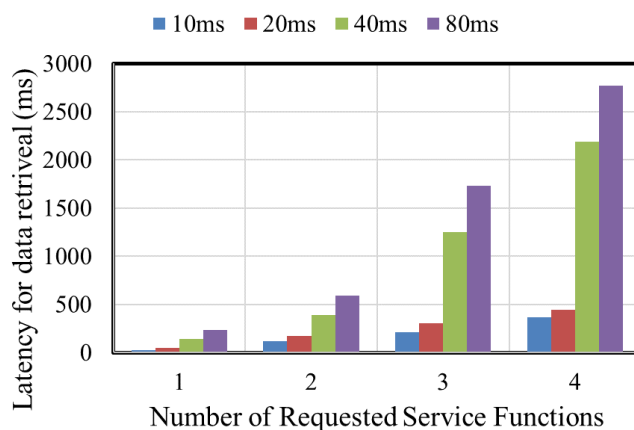


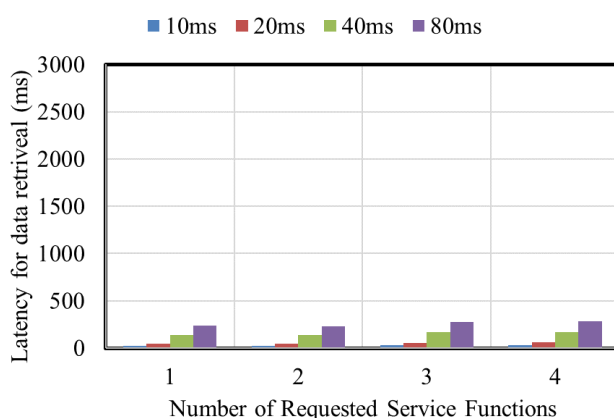
図 5. 研究室内に構築した実験環境の概要。

5.2. 評価結果

図 6 に、同期型と非同期型 ICN-SFC の遅延特性の評価結果を示す。図 6 に示すように、同期型の場合、Interest の受信無しにはサービスファンクションの起動が行われないため、ICN ノード間の往復遅延時間の影響を大きく受ける結果となった。また、この傾向については、要求するサービスファンクション数が増加するにつれ、その影響はさらに大きい。一方で、非同期型の場合は、それぞれのサービスファンクションが非同期に実行されるため、同期型と異なり、要求されるサービスファンクション数が増加したとしても、通信遅延の増加は抑えられる結果となる。以上により、単純化な環境下において、ICN-SFC の検証、また、非同期型 ICN-SFC の有効性を確認した。



(a) 同期型 ICN-SFC



(b) 非同期型 ICN-SFC

図 6. 同期型、非同期型 ICN-SFC の遅延特性の比較結果。

6. まとめと今後の予定

本稿では、オンデマンドな IoT デバイス仮想化を実現する ThingVisor Factory の主要な機能を紹介するとともに、その要素技術である情報指向ネットワークを利用したサービスファンクションチェイニング (ICN-SFC) について紹介した。ICN-SFC について、先行研究として提案されている同期型の ICN-SFC と ThingVisor Factory のアーキテクチャを活用した非同期型の ICN-SFC について、また、その実装例についても示した。ICN-SFC の性能評価として、研究室内に構築した環境下において、オープンソースの Python ベースの ICN ソフトウェアを利用し、ICN-SFC の性能検証および非同期型 ICN-SFC の有効性を実機評価した。

今後は、より大規模な環境下において、引き続き ICN-SFC の性能検証を行うとともに、ICN プロトコルの改善を行っていく。さらに、ネットワークが変動する環境下において、自律的に最適化なサービスファンクションチェイニングを実現するような自律運用法についても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] 金井, 中里, 金光, “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”, 電子情報通信学会 信学技報, vol. 119, no. 424, CS2019-101, pp. 19-24, 2020 年 2 月.
- [5] 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.
- [6] 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.
- [7] 吉井, 中里, “NDN におけるファンクション・チェイニングの実装”, 電子情報通信学会 信学技報 CS2018-39, pp. 133-138, 2018 年 7 月.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] IETF RFC 7665 “Service Function Chaining (SFC) architecture,” Oct. 2015.
- [13] IETF RFC 8300 “Network Service Header (NSH),” Jan. 2018.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] L. Liu, Y. Peng, M. Bahrami, 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.
- [20] 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.
- [21] Named Data Networking Project [online]:
<https://named-data.net/>
- [22] H. Asaeda, A. Ooka, K. Matsuzono, R. Li, “Cefore: Software Platform Enabling Content-Centric Networking and Beyond,” IEICE Trans on Comm. Vo. E102 B, No. 9, pp.1792-1803, 2019.
- [23] Cefore [online]: <https://cefore.net/>
- [24] NodeRED [online]: <https://nodered.org/>