

ICNにおけるIn-Network Processingのための複数機能 ルーティング手法の一検討

速水 祐作[†] 永田 晃^{††} 山本 幹[†]

[†] 関西大学大学院理工学研究科 〒564-8680 大阪府吹田市山手町 3-3-35

^{††} 株式会社 iD 〒812-0011 福岡県福岡市博多区博多駅前 1-7-22

E-mail: †{k911112,yama-m}@kansai-u.ac.jp, ††a-nagata@intelligent-design.co.jp

あらまし IoT (Internet of Things) の普及に伴いエッジコンピューティングへの期待が高まり、その実現手段として情報指向ネットワーク (ICN: Information-Centric Networking) 技術の適用可能性や、データ取得だけではなくネットワーク内でのデータ処理 (In-Network Processing) を ICN 技術を用いて実現する研究が注目されている。本稿では、ICN において複数の異なるデータ処理機能を提供する In-Network Processing を実現するため、複数のデータ処理機能へのルーティング手法を提案する。

キーワード 情報指向ネットワーク, ルーティング, In-Network Processing, 機能チェイニング

Multi-Function Routing for In-Network Processing in Information-Centric Networking

Yusaku HAYAMIZU[†], Akira NAGATA^{††}, and Miki YAMAMOTO[†]

[†] Kansai University 3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680 Japan

^{††} iD Corporation 1-7-22, Hakata-ekimae, Hakata-ku, Fukuoka, 812-0011 Japan

E-mail: †{k911112,yama-m}@kansai-u.ac.jp, ††a-nagata@intelligent-design.co.jp

Abstract ICN (Information-Centric Networking) which is originally innovated for efficient content distribution, is now discussed to be applied to edge computing in IoT (Internet of Things) environment. In this paper, we focus on more flexible network processing environment, In-Network Processing, which is realized with ICN architecture. Our proposal, a novel routing method for In-Network Processing, efficiently forwards content requests to group of multiple functions widely distributed in a whole network.

Key words Information-Centric Networking, Routing, In-Network Processing, Function Chaining

1. ま え が き

IoT (Internet of Things) においては、今までネットワークに接続されていなかったモノからデータが収集され、大量のトラフィックが発生する。一般的に、これらのデータはクラウドにおいて処理されるが、ネットワーク内に流通するデータトラフィック量の増加が懸念されることから、エッジコンピューティングの適用が注目されている [1]~[3]。エッジコンピューティングは、センサなどが存在するネットワークの末端、つまり、ネットワークエッジにおいてデータを処理するという考え方である。これにより、ネットワーク内に流入するデータ量の減少が期待できる。加えて、エッジに存在するユーザはクラウドを経由せずに処理されたデータ取得完了するため、応答時間の短縮が見込める。

しかし、ネットワークのエッジ全てにサーバなどの計算資

源を配置する事は、コストの面で現実的ではない。そのため、エッジではなく、それよりも多少コアネットワーク側、つまり、モノが存在する「エッジ」と、クラウドが存在する「コア」の間に介在するネットワーク内のノードが分散的にデータを処理する方法が考えられる [2]。このような構造を採る事で、処理機能を提供するノードの数を削減し、かつ、ネットワークに流入するデータトラフィックを低減する In-Network Processing が実現できる。

一方、In-Network Processing を実現する方法として、NFN (Named-Function Networking) [4] が提案されている。このアーキテクチャでは、データ取得のアーキテクチャである CCN/NDN (Content-Centric Networking/Named-Data Networking) を拡張し、データ処理もネットワーク、つまり、ルータが提供するための枠組みを提案している。データのみならず処理機能にも名前に基づく探索を適用する事により、ユーザは

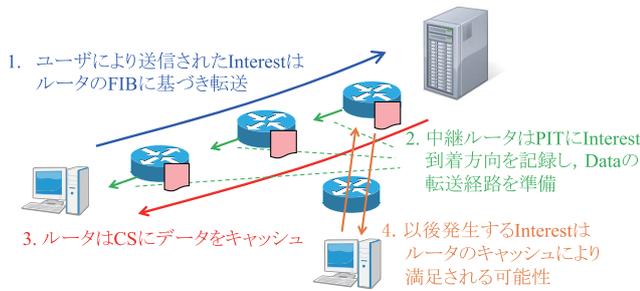


図1 CCN/NDNのパケット転送.

所望するデータと処理機能の名前を指定するだけで、データを自身が所望する形に加工された上で取得することができる。このアーキテクチャにおけるルータはNFNルータと呼ばれ、アプリケーションを実行し、従来の名前に基づくデータ転送に加えてデータ処理機能を提供する。NFNは、NFNルータが提供元からアプリケーションをダウンロードしてデータを処理するモデルを基本的に採用しているため、ネットワーク内で機能が利用される度に機能の実行場所が変動する。この場合、ネットワーク内の機能配置が流動的になるため、機能配置の状態遷移と経路設定の時間粒度を考慮すると、適切な機能への経路を制御することは困難である。

他方で、NFNのように機能を任意の場所からダウンロードするような、容易に機能の実行場所を変更可能な形態ではなく、機能の管理ポリシーやルータの計算資源の都合で、特定のルータでのみ機能が実行可能である形態も考えられる。この場合、ユーザの要求を特定の機能を実行可能なルータを経由させた上で、データが存在する場所までルーティングする必要がある。また、ユーザが指定する機能を提供可能なルータがネットワーク中に複数存在し、ユーザが取得するデータを取得可能なノードがキャッシュも含めて複数存在する場合、ネットワーク中の機能をどのように経由し、どのデータを取得するのがユーザにとって最適であるかを考慮する必要がある。それに加え、ユーザが指定する処理が複数の異なる機能で構成され、その処理の適用順序に制約があるとき、この経路決定問題はより困難なものとなる。

本研究では、将来ネットワーク全体が連携してユーザが所望するデータと機能を提供するため、必要な処理機能を群として捉えるIn-Network Processingを提案する。そして、個々の機能ではなく必要な機能群へのルーティングを実現する課題に取り組む。機能を群として捉えるため、複数機能の局在性を考慮したルーティング手法を提案する。そして、シミュレーションにより本提案手法の特性を分析する。

2. 背景技術

2.1 情報指向ネットワーク (ICN: Information-Centric Networking)

情報指向ネットワークを代表するアーキテクチャにCCN/NDN (Content-Centric Networking/Named-Data Networking) が存在する [5] [6]。CCN/NDNはユーザ駆動型の通

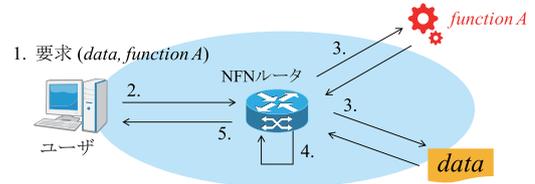


図2 NFNのデータ取得・処理の例.

信モデルを採用しており、データ取得はユーザの要求 (Interest) 送信から開始される (図1)。ユーザによって送信された Interest は中継ルータのルーティングテーブルである FIB (Forwarding Information Base) に基づいてサーバまで転送される。各中継ルータは、PIT (Pending Interest Table) に Interest が到着したインタフェースを保存する。サーバに Interest が到達すると、対応するデータ (Data) が PIT を辿り、Interest の逆経路で転送される。Interest の転送経路選択によりネットワーク内で発生するトラヒックパターンが変化するため、Interest の経路制御はCCN/NDNにおいて重要な技術課題の一つである。

2.2 ICNにおけるIn-Network Processing

ICNにおいてIn-Network Processingを実現するアーキテクチャとして、NFN (Named-Function Networking) が提案されている [4]。このアーキテクチャは、通常のCCN/NDNのデータ取得に加えて、名前に基づくデータ処理を実現するフレームワークである。図2にNFNにおけるデータ取得と処理の流れの一例を示す。

1. ユーザはあるデータ (data) を処理 (function A) して欲しいという要求をネットワークに送信する。2. 送信された要求はアプリケーションを実行できるNFNルータまで転送される。3. NFNルータは要求を分割し、適切なデータと処理機能を取得する。4. NFNルータはデータに処理を施し、5. そのデータをユーザへと返信する。これらの手順により、ユーザは所望するデータと機能の名前を指定する事で、データを取得すると同時にそのデータの処理をネットワークに委任する事ができる。ユーザの要求する処理が複数機能から構成される事を想定した場合、機能を提供する複数のルータへの経路制御が重要となる。しかし、NFNではこれらの複数機能を考慮した経路制御に関する検討がされていない。

一方、ICNにおいて、アプリケーション、つまり、機能の実行場所に関する検討が報告されている [7]。この文献では、データのサイズによりアプリケーションを実行するのに適した場所が変化する事が述べられている。しかし、この方式では、データ転送経路決定後に経路上の適切なルータにアプリケーションを移動して実行する形態を前提としており、機能の実行場所に制限がある。かつ、それらの機能がネットワーク内に散在する場合の経路決定問題は対象とされていない。

また、文献 [8] [9] では、CCN/NDNを拡張しIn-Network Processingを実現する複数機能のチェイニング (Function Chaining) が検討されている。この方式では、FIBにおいて次に実行する機能を探索する際に、同一ホップ数の転送先が複数存在すれば、その後に経由すべき機能が多く登録されている転

送先を選択するなどの簡単な経路制御ポリシーが検討されている。しかし、この手法では次に探索したいある機能に対して同一ホップ数で到達できる範囲内でしか選択の余地がなく、その効果は限定的となる可能性が高い。

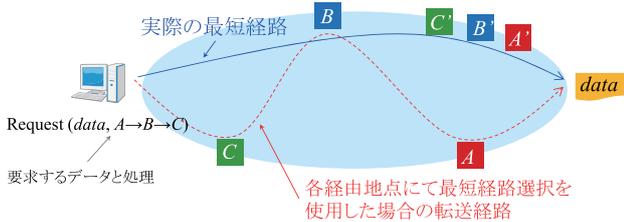


図3 複数機能への経路制御が抱える課題。

2.3 複数機能への経路制御における課題

本節では、複数機能への経路制御が抱える技術課題について説明する。通常の最短経路選択は単一の機能を宛先とするため、複数の機能を經由する際に、ある機能から他の機能に順々に最短経路選択する場合、それが全体最適最短経路でない可能性がある。図3に問題となる場合の例を示す。

ユーザが $A \rightarrow B \rightarrow C$ という順に処理された $data$ を要求する状況を想定する。1. 節で述べたように、機能の実行場所を自在に変更できない状況において CCN/NDN の枠組みを拡張してこれを実現する場合、ユーザが送信した Interest は $C \rightarrow B \rightarrow A$ という順に機能を經由する必要がある。この時、それぞれの機能に対して最短経路選択をする場合の Interest の転送経路は破線のようなになる。しかしながら、経由すべき全ての機能を考慮した $data$ までの最短経路は実線のようなになる。このように、経由すべき機能毎に単純に最短経路を選択すると、機能の配置によっては全体として最短ではない経路が選択される可能性がある。

この問題を防ぐためには、経由すべき機能の全ての組み合わせに対して経路を計算する方法が考えられる。しかし、通常の最短経路選択は単一の機能を宛先とするため経路計算が容易であるが、複数機能を宛先とした場合は、その複数機能の数に応じて計算が複雑化する。加えて、ルータは、利用される機能の組み合わせにつき一つの FIB エントリが必要になるため、全ての機能の組み合わせに対して FIB エントリを保持することはスケーラビリティの観点から現実的ではない。そのため、経路計算の複雑性が低く、スケーラビリティのある経路制御が必要になる。

3. ICN における In-Network Processing のための複数機能に対する経路制御

本節では、ICN において In-Network Processing を実現するための複数機能への経路制御方法について説明する。まず 3.1 節において、本研究が想定するシナリオについて説明する。そして、3.2 節において本稿の提案手法である複数機能の局在性を考慮したルーティング (LAR: Localization-Aware Routing) について説明する。

3.1 IoT 環境における ICN を用いた分散処理モデル

図4に本研究が想定するIoTのためのICNを用いた分散処理モデルについて説明する。このモデルでは、クラウドやユーザがエッジからデータを取得する際にデータを取得するだけでなく、In-Network Processing 層にてデータが事前処理される。図5に In-Network Processing 層における処理の手順について説明する。まず、ユーザ、もしくはクラウド内のアプリケーションが、あるデータと必要な処理内容の名前が記載された Interest を送信する。Interest はルータにより処理に必要な複数の機能へと順に転送される。各機能に到達する際に Interest の名前は変換され、経由した処理機能の情報が取り除かれる。全ての機能を經由した後、Interest はデータまで転送される。データは PIT を辿り機能へと転送され、各機能において処理が施される。全ての処理が完了すると、ユーザの元へデータが返送される。図5に示す挙動を実現するための具体的な CCN/NDN プロトコル拡張は今後の検討課題であり、本稿の対象外であるが、実現手段の一つとして、NFN [4] のような方法を参考にすることなどが考えられる。

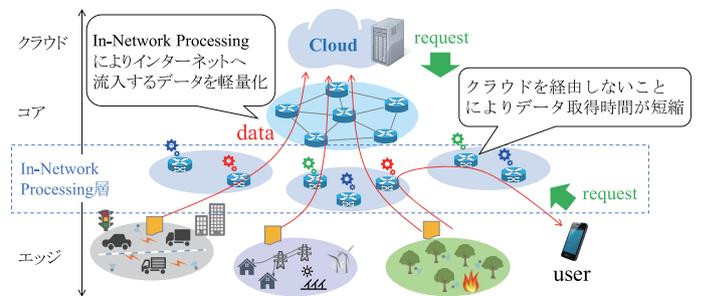


図4 IoT における ICN を用いた In-Network Processing の適用例。

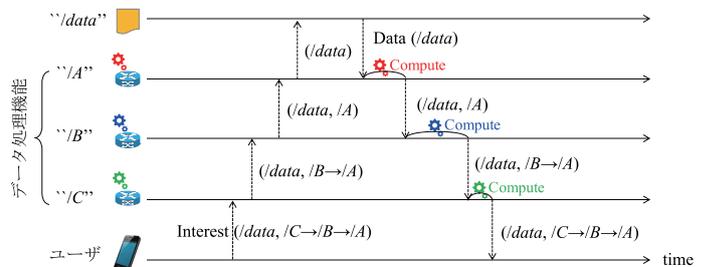


図5 In-Network Processing 層の動作。

以上の動作により、ネットワークに流入するトラフィック量を低減できる。また、ユーザがデータを取得する際、クラウドの機能を一切使用せずにネットワーク内で処理が完了する可能性があるため、ユーザに対する応答時間の短縮が見込める。クラウドにおける処理が必要なデータはクラウドまで転送され、In-Network Processing 層で処理が満足できるデータはクラウドまで転送されずにネットワーク内で処理される。これらの協調した動作により、コアネットワークの負荷を低減し、かつ、ユーザへのレスポンスを高速化するネットワーク内分散処理を実現する。

表 1 記号表

記号	定義
\mathcal{V}	ノードの集合
\mathcal{C}	データサーバの集合
\mathcal{S}_c	データ c を提供するサーバの集合
\mathcal{F}	機能の集合
$\mathcal{F}_{\mathcal{CH}}$	要求が経由する必要がある未訪問の機能の集合
\mathcal{R}_f	機能 f を提供するルータの集合
\mathcal{N}_v	ノード v の隣接ノードの集合
$d_{v,w}$	ノード v, w 間の距離 (ホップ数)
q_v	ノード v のクオリティ

3.2 複数機能の局在性を考慮したルーチング

本節では、本稿の提案方式である複数機能の局在性を考慮したルーチング (LAR: Localization-Aware Routing) について説明する。先述の通り、複数の機能を利用する際に問題となるのが経路計算の複雑性とルータにおけるルーティングテーブルのスケラビリティである。これらの問題に対応するため、本方式では PBR (Potential-Based Routing) [10] を拡張し、複数機能の局在性を考慮したルーチングを実現する。異なる種類の複数の機能が局在、つまり、局所的に偏って存在している箇所へと要求を積極的に誘導することにより、データ転送経路長を低減する経路制御を目標とする。そして、最後に本提案手法の応用として、複数機能への経由順序制約に対応する方法について説明する。まず、本稿でのポテンシャルの定義と、そのポテンシャルを経路制御に利用する方法を示した後、本方式の動作例を用いて説明する。表 1 に本稿で使用する記号をまとめる。ノード v におけるデータ c に関するポテンシャルの定義を以下に示す。なお、本稿では簡単のため、ネットワーク内に同様のデータが一つしか存在しない場合を限定して考える。

$$\forall v \in \mathcal{V}, \forall c \in \mathcal{C};$$

$$\phi_v^c = \begin{cases} -q_s, & \text{if } v = s \in \mathcal{S}_c, \\ -\frac{q_s}{d_{v,s}+1}, & \text{otherwise.} \end{cases} \quad (1)$$

式 (1) において、 q_s はデータ c を保持するサーバ $s \in \mathcal{S}_c$ のクオリティ、つまり、サーバ $s \in \mathcal{S}_c$ がデータ c に関する要求を引き込む力 (引力) を表す。 $d_{v,s}$ はノード v, s 間の距離 (ホップ数など) を表す。次に、ノード v における機能 f のポテンシャルの定義を以下に示す。本稿では、同じ機能を提供するノードがネットワークに複数存在する状況を想定する。

$$\forall v \in \mathcal{V}, \forall f \in \mathcal{F};$$

$$\phi_v^f = \begin{cases} -q_r, & \text{if } v = r \in \mathcal{R}_f, \\ -\max_{r \in \mathcal{R}_f} \left\{ \frac{q_r}{d_{v,r}+1} \right\}, & \text{otherwise.} \end{cases} \quad (2)$$

式 (2) において、 q_r はデータのポテンシャルと同様に、機能 f を保持するノード r のクオリティを表す。 $d_{v,r}$ はノード v, r 間の距離 (ホップ数など) を表す。このように、データや機能の保持ノードから距離が大きくなるにつれて徐々にポテンシャルが小さくなるように定義することで、ユーザからの要求をデータや機能へ引き込むポテンシャルの勾配を形成する。

先述のポテンシャルはルータにおける経路選択に使用される。式 (3), (4) にルータにおけるインタフェース選択方法を表す式を示す。

$$n_{next} = \operatorname{argmax}_{\mathcal{N}_v} P_{v \rightarrow n}, \quad (3)$$

$$P_{v \rightarrow n} = \begin{cases} \sum_{f \in \mathcal{F}_{\mathcal{CH}}} (\phi_v^f - \phi_n^f), & \text{if } \mathcal{F}_{\mathcal{CH}} \neq \emptyset, \\ \phi_v^c - \phi_n^c, & \text{otherwise.} \end{cases} \quad (4)$$

任意のルータ v において、ノード v の隣接ノード集合 \mathcal{N}_v の中から要求の転送先である n_{next} を決定する。式 (4) に表すように、 $\mathcal{F}_{\mathcal{CH}} \neq \emptyset$ である場合、経由する必要がある機能のポテンシャルを合成する。そして、その合成ポテンシャルの勾配が一番大きい隣接ノード n へと要求を転送する。要求が経由する必要がある機能に到達すると、その機能は $\mathcal{F}_{\mathcal{CH}}$ から取り除かれる。これを $\mathcal{F}_{\mathcal{CH}} = \emptyset$ になるまで繰り返し機能を訪問する。その後、経由すべき機能が無くなった場合、要求は宛先のデータに関するポテンシャルに基づいて転送される。

図 6 に簡易的なトポロジを用いた提案ルーチング手法の動作例を示す。ルータ 2 以外の全ルータがそれぞれ機能を実行している状況を想定する。機能を実行する各ルータから広告されるポテンシャル $\{\phi_v^f\}$ は、図 6 の上のグラフにおける実線になる。今回、各機能のクオリティ q_v は等しいと仮定する。一般的なポテンシャルの定義では、図の (A) に当たる実線はネットワーク内の機能 A のポテンシャルの総和、つまり破線を全て合計した値となるべきであるが、本稿では、式 (2) に示すようにポテンシャルの最小値のみをポテンシャルとして使用する。これは、ルータにおける経路選択の際、あるインタフェース方向に、同種の機能が複数集中して存在することによるポテンシャルの減少を抑制するためである。これにより、異なる種類の機能が複数局在する方向へ積極的に要求を転送する。

ここで、LAR の経路選択動作について説明する。ルータ 2 に機能 $C \rightarrow B \rightarrow A$ を経由する必要がある要求が到着した場合を考える。初期状態における合成ポテンシャルは図 6 の下のグラフにおける実線になる。そのため、合成ポテンシャルの勾配に従い、要求はルータ 3 の方向へ転送される。要求がルータ 3 に到着すると、ルータ 3 において経由すべき機能の内の一つである機能 C が実行されているため、機能 C を未訪問の機能集合 $\mathcal{F}_{\mathcal{CH}}$ から取り除く。それにより、合成ポテンシャルは図 6 の下のグラフにおける破線のようにになる。再び、要求は合成ポテンシャルの勾配に従いルータ 4 の方向へ転送される。ルータ 4 では機能 B が実行されているため、 $\mathcal{F}_{\mathcal{CH}}$ から機能 B が取

り除かれ、合成ポテンシャルは図6の下のグラフにおける鎖線になる。要求は、同様にしてルータ5において機能Aを経由した後、データが存在する方向へデータのポテンシャルに従い転送される。

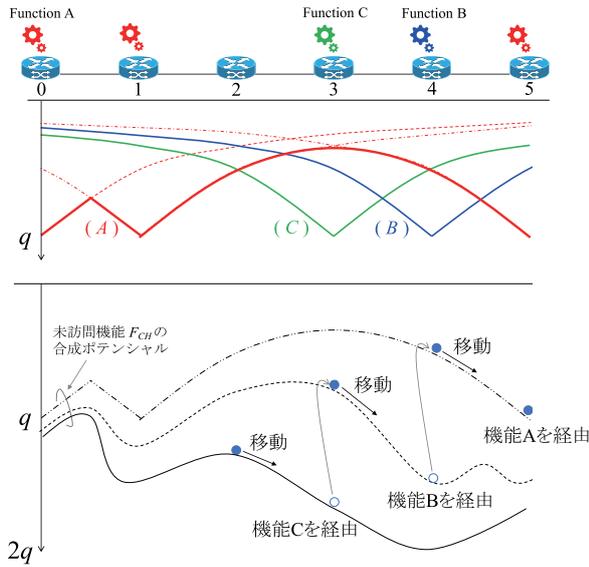


図6 ポテンシャルルーティングの動作例.

次に、機能の経由順序制約のためのポテンシャルの応用方法について説明する。複数の機能を経由する際にその経由順序に制約が有る場合が存在する。本方式では、この制約を満足するためにポテンシャルの勾配を活用する。ユーザは要求送信時に、優先機能 f_p の情報を要求に記載する。ネットワーク内の各ルータはそれを認識し、優先機能に対するポテンシャルの勾配を常に監視する。そして、その機能に対するポテンシャルの勾配が負になる時、つまり、 $\phi_v^{f_p} - \phi_n^{f_p} < 0$ である場合、要求が転送される方向が優先機能から遠ざかると判断し、ポテンシャルに基づくルーティングから機能の経由順序を考慮したルーティングへとポリシーを切り替える。以上の動作により、機能の経由順序を考慮した経路選択を行う。

機能とデータに関するポテンシャルを適宜使用することで、要求が経由する必要がある複数機能を全て考慮した経路選択が可能となる。また、ポテンシャルの勾配を活用することにより、機能の順序制約を満足することができる。加えて、ルータが持つルーティングテーブルのステートは、各エントリに対してポテンシャルの値のみを保持するだけで良いため、通常のルーティングコストと同様に実装が可能である。

4. 性能評価

3.2節で提案したルーティング手法は、(i) 機能ポテンシャル合成による所望の機能群の局在性の考慮と、(ii) 優先機能のポテンシャル勾配監視による機能の経由順序制約の考慮の2つの要素から為る。本稿では、まず前者 (i) のみを対象として、本方式の基本的な有効性の検証と、課題の抽出を目的にシミュレーションによる評価を行う。実験には、ns-3 [11] をベースとした CCN/NDN のオープンソースシミュレータである ndnSIM

2.1 [12] を使用する。なお、後者 (ii) は今後の課題とする。

4.1 評価モデル

図7にシミュレーションに用いたトポロジを示す。4つのネットワークの内、 NW_2 に producer (データ) が、 NW_4 に consumer (ユーザ) が存在する。シミュレーション条件を表2に示す。ネットワーク内に3種類の異なる機能A, B, Cが存在し、同一種類の機能を実行可能なルータが2台ずつ配備された状況を想定する^(注1)。つまり、機能を実行可能なルータがネットワーク内に6台配備されている。consumerがデータ取得とデータ処理を要求し、その処理機能としてA, B, Cの3種類全てを指定するシナリオを考える。ランダム生成した1000通りの機能配置パターンの中から特徴的な結果を選択して分析する。以下に示す2通りのルーティング手法 (a), (b) を用いて比較評価を行った。いずれも、Interestにより要求された機能の全てを順不同で経由した後、producerへとルーティングする手法である。

(a) NFF (Nearest-Function First)

経由すべき機能毎に単純に最短経路選択を行うルーティング手法。

(b) LAR (Localization-Aware Routing)

機能ポテンシャルの合成により経由すべき機能群の局在性を考慮するルーティング手法。提案手法の内の (i) にあたる。

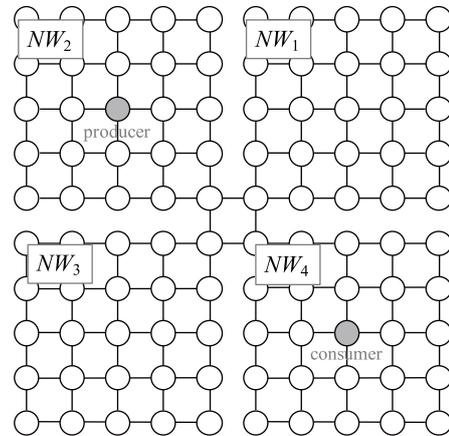


図7 評価トポロジ.

表2 シミュレーション条件

ノード数	100 [node]
データ数	1 [content]
機能数	3 [function]
q_s, q_r	1.0
伝搬遅延	10 [msec]
リンク帯域	1 [Gbps]

(注1): 提供する機能としては全く同一であるが、便宜上、ネットワーク内で実行される機能を区別するため、A, A' のように表記する。

4.2 実験結果

本節では、ランダムな機能配置の中から特徴的な配置3つを選択し、その結果について考察する。

- ケース 1: 複数機能を考慮する効果が有る場合

図 8 にルーチングに複数機能を考慮することによる利益がある場合の結果を示す。ケース 1 では、 NW_4 において consumer が Interest を送信して最初に機能 A への経路を選択する際、NFF は単一機能のみを考慮するため、 NW_4 に存在する機能 A, A' の内、ここでは A' が選択されている。それに対し、LAR は所望の機能を提供可能なルータが同在する方向、つまり、 NW_4 の下部の方へとポテンシャルに従い要求を機能 A の方へ転送する。結果、複数の機能を効率的に経路することができ、producer に到達するまでにネットワークを迂回する量が減少する。このように機能を經由する毎に最短経路を選択する場合と比較して、経路すべき複数の機能の局在性を考慮することで、データ転送経路長の改善が見込める。

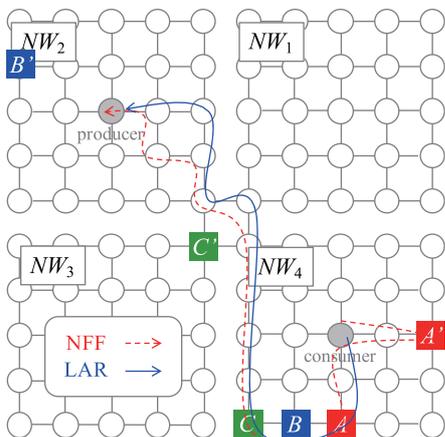


図 8 ケース 1: 複数機能を考慮する効果が有る場合.

- ケース 2: 複数機能を考慮する効果が顕著に有る場合

図 9 に LAR による経路選択の効果がより顕著に現れる場合を示す。両方式ともに、consumer が送信した要求は、まず NW_1 に存在する機能 C へ転送される。次に經由する機能を決定する際、NFF では NW_1 に存在する最近隣の機能 B へと要求を転送する。その結果、NFF は未訪問の機能 A を經由するためにネットワークを往來することになり、そのデータ転送経路長が大きく増加する。それに対して、LAR は複数機能、この場合未訪問の機能である A と B、を考慮した経路を選択するため、 NW_2 に存在する B' を選択する。それに対して LAR は複数の機能を考慮することにより、ネットワーク内を迂回しないデータ取得が可能となる。

- ケース 3: 複数機能を考慮する効果が無い場合

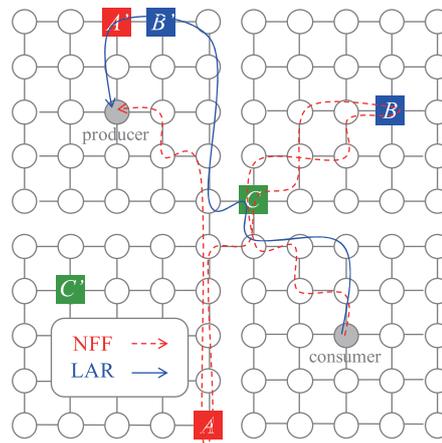


図 9 ケース 2: 複数機能を考慮する効果が顕著に有る場合.

図 10 に NFF と LAR の選択経路に有意な差が見られない場合の結果を示す。この原因は、consumer-producer 間の最短経路上に經由すべき全ての複数機能が偏って配置されているためである。NFF で単一機能に対する最短経路を順に選択しても、LAR で複数の機能を考慮したルーチングをしても、consumer-producer 間の全体最適な最短経路を經由する。ゆえに、NFF と LAR の選択経路に有意な差が見られない。

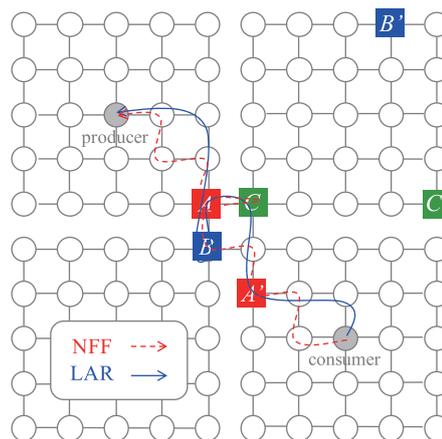


図 10 ケース 3: 複数機能を考慮する効果が無い場合.

5. まとめ

本稿では、IoT 環境におけるエッジコンピューティングを実現する方法として、In-Network Processing に着目し、ICN を用いた In-Network Processing の分散処理モデルを提案した。課題として、複数のデータ処理機能への経路制御を取り上げ、機能の局在性に着目した経路制御を提案した。評価においては、処理機能の順序制約が無い場合を対象とし、最近隣の機能から順に經由する経路制御方式と比較して、複数の機能の局在性を考慮することで改善が見られることを示した。今後の課題として、機能の順序制約を考慮した経路制御の評価や異なるモデルにおける本経路制御方式の評価が挙げられる。

謝辞 本研究の一部は、JSPS 科研費 26280034 により得られたものである。

文 献

- [1] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *IEEE Pervasive Computing’09*, vol. 8, no. 4, pp. 2-11, Oct.-Dec. 2009.
- [2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” in *Proc. ACM MCC’12*, Helsinki, Finland, Aug. 2012, pp. 13-15.
- [3] “FG IMT-2020: Report on Standards Gap Analysis,” TD 208 (PLEN/13), ITU-T SG13, 2015.
- [4] C. Tschudin and M. Sifalakis, “Named Functions and Cached Computations,” in *Proc. IEEE CCNC’14*, Las Vegas, NV, USA, Jan. 2014, pp. 851-857.
- [5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” in *Proc. ACM CoNEXT’09*, Rome, Italy, Dec. 2009, pp. 1-12.
- [6] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, “Named Data Networking (NDN) Project,” PARC Technical Report 2010-003, Oct. 2010.
- [7] 栗田敏彦, 佐藤出, 福田健一, “In-network processing におけるアプリケーション実行場所の検討,” 電子情報通信学会通信ソサイエティ大会講演論文集, BT-4-4, Sept. 2016.
- [8] 福田健一, 伊藤章, “データやアプリケーションを繋ぐ情報指向ネットワーク技術の開発,” 電子情報通信学会通信ソサイエティ大会講演論文集, BT-1-4, Sept. 2016.
- [9] L. Liu, L. Xie, M. Bahrami, Y. Peng, A. Ito, S. Mnatsakanyan, G. Qu, Z. Ye, and H. Guo, “Demonstration of a Functional Chaining System Enabled by Named-Data Networking,” in *Proc. ACM ICN’16*, Kyoto, Japan, Sept. 2016, pp. 227-228.
- [10] A. Basu, A. Lin, and S. Ramanathan, “Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm,” in *Proc. ACM SIGCOMM’03*, Karlsruhe, Germany, Aug. 2003, pp. 37-48.
- [11] ns-3, network simulator 3. <http://www.nsnam.org/>
- [12] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2.0: A new version of the NDN simulator for NS-3,” NDN, Technical Report NDN-0028, 2015.