

# NDNベースのエッジコンピューティング環境における 位置データの計算ノード選択に関する一考察

武政 淳二<sup>†1</sup> 小泉 佑揮<sup>†1</sup> 田上 敦士<sup>†2</sup> 長谷川 亨<sup>†1</sup>

**概要:** ユーザの要求に応じて Internet of Things (IoT) デバイスからデータを収集し計算した結果を応答するオンデマンドなデータ処理サービスでは、IoT デバイスを収容する各エッジルータでのローカルな計算が応答遅延の削減に繋がる。このようなデータは IoT デバイスの現在位置に紐づくため、ユーザは、要求する範囲に位置する IoT デバイスを収容するエッジルータと、要求-応答の通信を必要とする。本稿では、ユーザとエッジルータ間の接続にインターネット網を仮定し、Named Data Networking (NDN) を用いてこの通信を実現する。ただし、NDN ベースのインターネット網では、上述したサービスと他の NDN ベースサービスとの共存が想定されるため、NDN フォワーディングの制約の中での実現が必要である。これに対し、まず、位置の範囲を名前とする名前の最長一致検索により網内のルータが適切なエッジルータを探索可能なルーティング/フォワーディング方式を設計する。次に、NDN の 1 対 1 の要求-応答の通信モデルの制約下で、1 ユーザ対多エッジルータ間の計算の要求-応答通信を RICE を用いて実現する。

## 1. はじめに

ユーザの要求に応じて Internet of Things (IoT) デバイスからデータを収集し、計算した結果を応答するオンデマンドなデータ処理サービスの需要が高まっている。IoT デバイスの生成するデータの数、サイズともに年々急激に増加しており、クラウドの計算基盤へのデータ転送、ひいてはサービス応答の遅延の増加が懸念される。これに対し、IoT デバイスを収容する各エッジルータでローカルに計算を行うエッジコンピューティングの導入が期待されている。

このようなデータは IoT デバイスの現在位置に紐づくため、ユーザは、自身の要求する範囲に位置する IoT デバイスを収容するエッジルータと、要求-応答の通信を必要とする。この通信を IP 網上で実現する場合、エッジルータの計算資源や IoT デバイスの位置を集中管理するブローカで、計算要求の宛先となるエッジルータを解決する必要がある [1]。しかし、宛先解決に要する往復遅延、IoT デバイスの移動に伴う位置収集のオーバーヘッド、ブローカーが単一障害点となるなど、多数の問題を抱えている。

これに対し、Named Data Networking (NDN) [2] を用いて計算の通信を提供する、Named Function [3] は上述した問題を解決することを期待されている。以降では、計算と関

数、という用語を同義に扱う。Named Function では、関数とデータの名前を通信に用いることで、網内のルータが、要求に対し、どのデータをどのように計算すべきか、を認識できる。したがって、網内のルータは、計算要求の中で定義された名前に従い適切なエッジルータを探索し、ブローカーのような宛先解決を行うノードなく、計算の要求を直接転送できる。ここで、本稿では、ユーザとエッジルータ間の接続にインターネット網を、エッジルータと IoT デバイス間の接続に単一ドメインの管理するローカルな網を仮定しており、前者をコア網、後者を IoT ドメインと呼ぶ。

これまで、IoT デバイスから収集したデータの計算を NDN を用いてサポートする研究は IoT ドメインを対象に数多く行われてきた [4], [5], [6]。しかし、既存研究をコア網へ適用する場合、NDN フォワーディングの大幅な変更が必要となり、以下の 2 点が問題となる。1 点目は、ルータの経路表において、階層化された名前の最長一致検索 [2] を改変する点である。NDN において、経路表でのプレフィクスの集約、軽量の検索処理は、名前の最長一致検索により実現されている。一方、位置やデータの種類といったキーワードの集合を名前とし部分集合の検索により IoT デバイスを柔軟に探索する方式 [4] や、階層化された位置の名前に対する前方一致検索により探索対象の IoT デバイスの位置の範囲を調整する方式 [5] が提案されているが、いずれも経路表で高負荷な検索ルールを必要とする。2 点目は、1 要求対 1 応答、つまり、1 要求に対し最大で 1 応答しか

<sup>†1</sup> 現在、大阪大学 大学院情報科学研究科  
Presently with Osaka University

<sup>†2</sup> 現在、KDDI 総合研究所  
Presently with KDDI Research, Inc.

返送しないという、NDN のプル型通信の原則を崩す点である。NDN では、この原則に基づき要求の流量を制御することでネットワークのトラフィックを制御しており [7]、NDN のフロー制御や輻輳制御の方式はこの原則を前提に設計されている。一方、ユーザと探索された複数の IoT デバイスとの間で 1 対多の通信を実現するために、Push 型の Publish/Subscribe を採用する方式 [5], [6] が多く、この原則を崩す必要がある。コア網では、上述したサービスに加え、他の NDN ベースサービスとの共存が想定されるため、上述した 2 つの NDN フォワーディングの原則は必須となる。

そこで、本稿では、1 ユーザ対多エッジルータの間で、位置に基づく計算の要求-応答の通信を、上述した NDN フォワーディングの制約の中で実現する。具体的には、まず、ルータにおいて、ユーザの要求する位置の範囲を名前として持つ計算要求に対し、名前の最長一致検索のみを用いて、適切なエッジルータを探索可能なルーティング/フォワーディング方式を設計する。次に、探索された複数エッジルータからユーザが計算結果を取得する通信を、1 対 1 の要求-応答通信の制約を崩すことなく実現する。

本稿の構成は以下の通りである。2 章で、想定するアプリケーションとシステムモデルを説明後、課題を提起する。3 章で位置ベースのフォワーディング方式を設計し、4 章で 1 ユーザ対多エッジルータ間の要求-応答の通信を RICE をベースに設計する。5 章でシミュレーションにより性能を評価し、6 章で本研究をまとめる。

## 2. 想定シナリオ

本章では、想定するアプリケーションとそれを展開するシステム、および、システムの構築に必要なネットワークの仮定を説明後、本稿で取り組む課題を説明する。

### 2.1 アプリケーション

アプリケーションとして IoT ベースの物体追跡を想定する。具体例として、事故を起こし逃走する自動車 (以降、ターゲットと呼ぶ) の現在位置を警察が特定する場面を想定する。警察は事故の発生位置とターゲットの特徴を含む通報を受けると特定を開始する。警察は、事故の発生位置を中心とする位置の範囲とターゲットの特徴を指定し、画像認識の関数の実行を要求する。計算ノードは要求を受信すると、指定された位置の範囲内に存在するカメラやドライブレコーダーの画像を収集する。その後、計算ノードは、収集した各画像を、ターゲットの特徴とともに画像認識の関数へ入力しターゲットの存在を判定する。最後に、全ての画像に対する実行結果を警察へ返送する。以降では、このようなアプリケーションを位置ベースサービスと呼ぶ。

### 2.2 システムモデル

想定するシステムを図 1 に示す。位置ベースサービスを

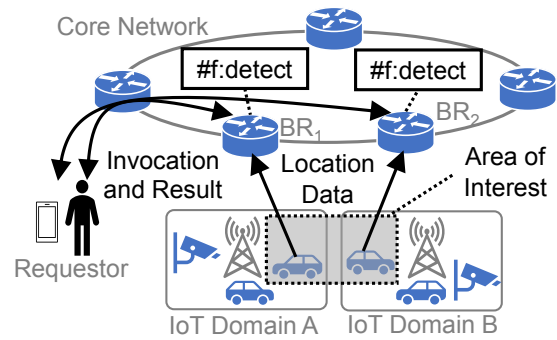


図 1 システムモデル

提供するプロバイダは計算ノードを提供、あるいは他の事業者の提供する計算ノードから計算資源を借用し、関数の実行環境を提供する。計算ノードとして基地局にコロケーションするエッジルータを想定し、各エッジルータが単一ドメインの管理するローカル網 (以降では IoT ドメインと呼ぶ) を収容する。以降ではエッジルータと Border Router (BR) を同義の用語として扱う。BR は、IoT ドメインを経由し IoT デバイスから計算に用いるデータを収集する。以降では、IoT デバイスの生成するデータを位置データと呼ぶ。BR 間、および、ユーザと BR の間はインターネット網 (以降ではコア網と呼ぶ) で接続される。ユーザは、関数の名前、位置の範囲、関数の実行に必要な引数を指定し、計算ノードへ関数の実行を要求する。ここで引数とは、関数の名前と位置の範囲以外で、関数の実行に必要な情報であり、2.1 節の例ではターゲットの特徴のことを指す。

### 2.3 ネットワーク

IoT ドメインは BR と IoT デバイスの間で位置データを転送するために、コア網はユーザと BR の間で関数の実行要求と実行結果を交換するために、それぞれ利用される。

#### 2.3.1 IoT ドメイン

IoT ドメインは単一ドメインの管理するプライベートな網であり、IoT ドメイン毎に固有のネットワークプロトコルを展開することが想定される。したがって、本稿では、特定のネットワークプロトコルを仮定しないが BR がユーザから受信した要求に含まれる位置の範囲を指定し、その範囲内に存在する IoT デバイスから位置データを取得できることを仮定する。例えば、これを実現するアプローチとして、Kurihara らは Publish/Subscribe ベースの位置ルーティング/フォワーディング [5] を提案している。この方式では、まず、IoT デバイスが自身の現在位置を名前とし、BR に Subscribe することで BR を根とする配信木を構築する。また、BR はこの配信木の逆向きの経路として自身の名前を Subscribe する。BR は、位置の範囲を指定し要求を Publish すると、配信木に従い要求が IoT デバイスへ転送される。要求を受信した IoT デバイスは BR の名前を指定し位置データを Publish すると配信木の逆向きの経路を辿り

BR へ転送される。

### 2.3.2 コア網

コア網はグローバルな接続性を提供する網であり、2.1 節のアプリケーション以外にもビデオ配信など NDN ベースのアプリケーションとの共存が想定される。したがって、コア網は NDN [2] に準拠したルーティング/フォワーディングを行うと仮定する。NDN の通信は、要求パケットである Interest、応答パケットである Data の 1 対 1 の双方向通信により実現される。パケットは Unique Resource Identifier (URI) のような階層的な次元の名前を有し、名前に基づき網内で転送される。具体的には、ルータが Interest の経路を保持する Forwarding Information Base (FIB)、Data の経路を保持する Pending Interest Table (PIT) を用いて、以下の通りパケットを転送する。ルータは Interest を受信すると、その名前と受信インタフェイスの組を PIT へ追加し、戻りの Data の経路を構築する。その後、ルータは Interest の名前を FIB に入力し、最長一致、つまり最も長く一致する名前プレフィックスを検索し、対応するインタフェイスへ Interest を転送する。なお、FIB はルーティングプロトコルに従い構築される。一方、ルータが Data を受信すると、その名前を PIT へ入力し、完全一致する名前を検索し、対応するインタフェイスへ Data を転送する。転送後、その名前とインタフェイスの組は PIT から削除される。以降では、経路表と FIB を同義の用語として扱う。

## 2.4 課題

ユーザの要求する位置の範囲は、複数のエッジルータにまたがる。例えば、2.1 節の例で、ターゲットが時速 60km で移動し、警察は事故発生から 1 分後に、ターゲットを特定する計算の要求を開始すると仮定する。このとき、警察は、事故の発生位置を中心とし、半径 1 km の位置の範囲内に自動車が存在すると推測し、この位置の範囲へ計算を要求する。これに対し、計算ノード、つまり、エッジルータのカバレッジは約 150 m であり [5]、ユーザから複数のエッジルータへ計算要求を届け、エッジルータで計算した結果を全てユーザに返送することが求められる。

しかし、ユーザと複数エッジルータの間で、計算の要求と結果を交換するための通信を NDN ベースのコア網で実現するには 2 点の課題がある。第一に、NDN の FIB に従い、計算要求を、その位置の範囲内に存在するエッジルータへ転送する点である。位置の範囲に基づくフォワーディング方式として、 $x$ - $y$  座標を識別子とし 2 次元の空間を検索する方式 [8]、1 次元の空間充填曲線 [9] を識別子とし前方一致検索する方式 [5], [10] が提案されている。しかし、NDN の FIB は次元の名前の最長一致検索を行うものであり、双方の方式を適用することは難しい。

第二に、NDN の PIT に従い、ユーザの送信した 1 つの計算要求に対し、複数の計算結果をユーザへ返送する点で

ある。PIT は、1 つの Interest に対し戻りの経路を生成し 1 つの Data を転送した後に削除する。一方、計算要求を Interest、計算結果を Data とすると、複数の計算ノードが Data を返送しても、1 つの Data しかユーザへ返送されない。これに対し、PIT エントリをタイマー、つまり、一定時間保持し、複数の Data 転送を許容する方式 [11] が提案されているが、1 要求対 1 応答の原則を崩す必要がある。NDN のトラフィック制御はこの原則に基づき、Interest の流量制御により実現されており [7]、グローバルなコア網でこの原則を崩すことは望ましくない。

そこで本稿では、まず、第一の課題に対し、3 章で FIB における名前の最長一致検索の制約下で位置ベースのフォワーディングを設計する。次に、第二の課題に対し、4 章で 1Interest 対 1Data の PIT の下で 1 つの計算要求から複数の計算結果を取得する通信を RICE [12] を用いて実現する。また、ここまでで議論していないが、位置の範囲以外の関数実行に必要な引数、例えば、2.1 節のターゲットの特徴をユーザから計算ノード転送する通信は RICE により実現されており、4 章で併せて説明する。

## 3. 位置ベースルーティング/フォワーディング

### 3.1 要求条件

計算を要求する Interest のフォワーディングは、以下の 3 つの要件に従い設計する。第一に、要求者毎に異なる位置の範囲に対応するため、Interest の名前は、任意のサイズの位置の範囲を表現できる必要がある。第二に、FIB は名前で表現する位置の範囲内に存在するエッジルータへ最長一致検索により Interest を転送する必要がある。第三に、IoT デバイスの移動に対しスケラブルである必要がある。各ルータは、ルーティングプロトコルを用いて IoT デバイスの位置更新のメッセージをルータ間で交換し FIB を構築する。しかし、2.1 節のような自動車のドライブレコーダーを IoT デバイスとすると、位置更新がコア網で頻繁に交換される。コア網の負荷を増加しないよう、この位置更新メッセージ数は少ない必要がある。

本章では、これら 3 つの要件を満たすルーティング/フォワーディング方式を設計する。第一の要件に対し、3.2 節で、位置の範囲を次元の数値列で表現可能な Z 記法 [9] をベースに計算要求の命名規則を設計する。第三の要件に対し、3.3 節で位置の更新を広いエリア単位で行うことで位置更新の頻度を削減しつつ FIB を正しく構築する位置更新方式を設計する。第二の要件に対し、3.4 節で 3.2 節で設計した位置の名前に対し、名前の最長一致検索だけで、所望の位置の範囲に存在する計算ノードへ Interest を届ける FIB の構築方式を設計する。

### 3.2 命名規則

Interest の名前は、位置名と関数名を “/” で結合した文字

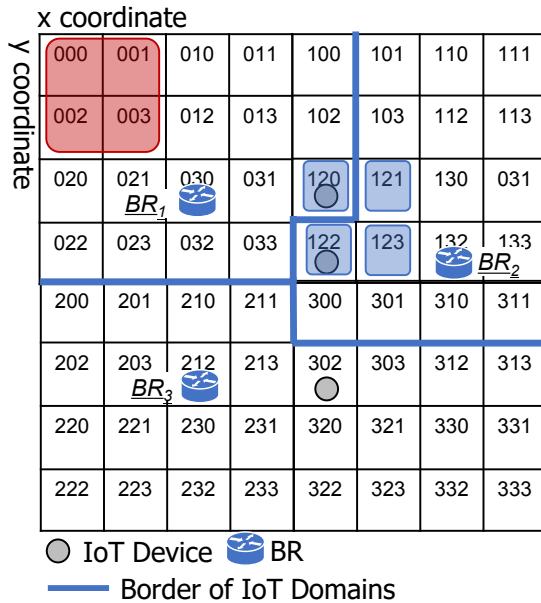


図2 サービスエリアにおける位置の更新

列とする。位置名は位置の範囲を表し、コア網で Interest をどの計算ノードへ転送するかを指定する。関数名は、計算ノードで実行する関数を指定する。

一次元の名前で任意のサイズの位置の範囲を表現するため、2次元座標を1次元の値で表現するZ記法[9]を用いる。Z記法では、対象とする矩形を縦、横に2分割ずつ、計4分割し、分割後の各矩形に対し、Zの順に0から3の値を付与する。この分割を各矩形に対し再帰的に繰り返し、分割毎に0から3の値を付与する。本稿では、位置名として各桁の数値を“/”で結合した文字列を利用する。例えば、図2では、3分割の例を表している。以降では、位置の分割数をNとし、Nの桁数で表現される位置をセル名、N以下の桁数で表現される位置の範囲を位置名と呼ぶ。

Ascigilらの記法[4]に従い、関数名はタグ#f:で指定する。例えば、名前/1/2/#f: detectは、位置の範囲/1/2に存在する計算ノードへ関数detectの実行を要求する。なお、本稿では、全ての計算ノードが所望の関数を実行できることを仮定するが、負荷分散や遅延などの観点で計算ノードの探索が必要な場合は文献[3]の探索方式に従う。

### 3.3 位置の更新

IoTデバイスはGPSなどを用いて定期的に現在位置を取得し、3.2節のZ記法に従いセル名に変換し保持すると仮定する。IoTデバイスは、セル名の単位で現在位置の変更を検知すると、現在と過去のセル名を最近傍のBRに通知する。以降ではこの通知を位置更新メッセージと呼ぶ。BRは位置更新メッセージを受信すると、Named-data Link State Routing (NLSR)[13]などのルーティングプロトコルに従いコア網の他のルータへ広告する。ルータは位置更新メッセージを受信すると、現在のセル名と受信インター

フェイスの組をFIBへ追加し過去のセル名を削除する。

#### 3.3.1 位置プレフィクスへの集約

本稿では、セルよりも広いエリア単位で、コア網で位置を広告する。具体的には、BRはIoTデバイスから位置更新メッセージを受信すると、過去と現在の位置のセル名について、N以下のD桁の位置プレフィクスをそれぞれ計算し、双方の値が異なる場合にのみコア網へ広告する。なお、広告する位置更新メッセージには現在位置、過去位置ともに位置プレフィクスを含むものとする。例えば、N=3桁のセルでD=2とした場合には、図2の赤枠で示すサイズのエリアを移動すると位置更新メッセージが広告される。

#### 3.3.2 ドメイン境界のセルにおける広告

位置プレフィクスによる広告では、IoTドメインの境界のセルを移動する場合に、コア網へ位置の更新が反映されない。例えば、図2で、IoTデバイスがBR1のIoTドメインのセル/1/2/0からBR2のIoTドメインのセル/1/2/1へ移動しても、コア網で位置が広告されない。この場合、位置名/1/2/1へのInterestは、本来BR2へ転送されるはずだが、BR1へ転送されてしまう。

これに対し、BRは各セルについて、セルを管理するIoTドメインを事前に知っていることを仮定し、IoTドメインの境界をまたぐ移動の場合、セル単位でコア網へ位置更新メッセージを広告する。具体的には、BRは位置更新メッセージを受信した場合に、過去の位置と現在の位置のセル名からIoTドメインを取得し、双方が異なる場合、位置更新メッセージをコア網へ広告する。

### 3.4 FIBの構築

3.2節の位置名は、短い桁数であるほど広い位置の範囲を指定する。例えば、図2のように3桁のセルの場合、2桁の位置名/1/2は4つのセル/1/2/0から/1/2/3を位置の範囲として指定する。したがって、ルータのFIBでは、Interestに含まれる位置名に対し、前方一致する全ての位置プレフィクスを検索し、得られた全てのインタフェイス向きへInterestをマルチキャストする必要がある。

しかし、2.3.2節で述べた通り、NDNのFIBは名前の最長一致、つまり、Interestの名前に対し最も長く一致する名前プレフィクスを検索するため、上記の一致を実現できない。これに対し、FIBの一致ルールを前方一致に変更することもできるが、以下の理由より採用しない。第一に、3.3節で設計したようなプレフィクスの集約を適用できないため、位置更新メッセージの広告数が増加する。FIBが前方一致ルールを採用する場合、Interestの名前がFIBのプレフィクスより長い場合に一致しない。例えば、図2の例で、Interestの名前/1/2/2は位置プレフィクス/1/2にマッチしない。そのため、位置プレフィクスへ集約せずセル単位で位置を広告せざるを得ない。第二に、最長一致検索と比べ、前方一致検索の計算量が多い。第三に、他のNDN

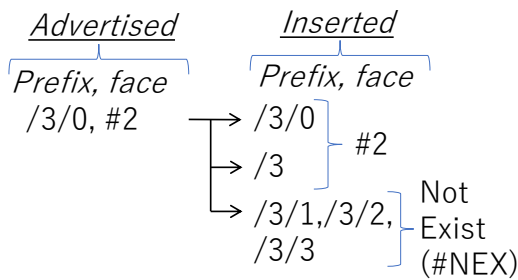


図3 FIBにおける位置プレフィクスの展開

アプリケーションとの互換性が損なわれる、つまり、FIBの最長一致検索を前提とするビデオ配信などのNDNアプリケーションは、前方一致を前提として経路広報などを再設計する必要がある。そのため、本節では、名前の最長一致検索の制約下で上述した前方一致検索を実現するためのFIBを構築する。

### 3.4.1 位置プレフィクスの展開

、位置更新メッセージに含まれる位置プレフィクスに対し、より短い長さの全プレフィクスを計算しFIBに挿入する。まず、ルータは、 $D$  桁の位置プレフィクス  $/z_0/\dots/z_{D-1}$  を有する位置更新メッセージを受信すると、 $1 \leq l \leq D$  を満たす全てのプレフィクス  $/z_0/\dots/z_{l-1}$  (以降、展開プレフィクスと呼ぶ) を計算する。次に、展開プレフィクスと位置更新メッセージの受信インタフェイスの組をFIBへ挿入する。なお、削除時も同様に過去の位置を表す位置プレフィクスから展開プレフィクスを計算しFIBから削除する。図3に、ルータが位置プレフィクス  $/3/0$  の位置更新メッセージをインタフェイス #2 から受信する例を示す。位置プレフィクス  $/3/0$  に対し、 $1 \leq l \leq D$  を満たすプレフィクス  $/3/0$ 、 $/3$  を計算し、インタフェイス #2 とともにFIBへ挿入する。

### 3.4.2 偽陽性の回避

前節の方式では、FIBの検索において、Interestの位置名が、より短いプレフィクスに誤って一致するケースがある。例えば、図3の例では、Interestの位置名  $/3/3$  は、本来FIBに挿入される  $/3/0$  と一致しないべきであるが、展開を適用した場合にはプレフィクス  $/3$  に一致する。この結果、Interestが冗長に転送され、コア網の負荷を増大する。

これを回避するため、プレフィクスを展開する際に、一致すべきでないプレフィクスをFIBへ挿入する。具体的には、まず、位置プレフィクス  $/z_0/\dots/z_{D-1}$  を挿入する際、 $1 \leq l \leq D$ 、 $0 \leq z'_l \leq 3$ 、 $z'_0 = z_0$  を満たす全てのプレフィクス  $z'_0/\dots/z'_{l-1}$  を計算する。次に、計算したプレフィクスと、一致してもInterestを破棄する特殊なインタフェイス #NEXの組をFIBへ追加する。ただし、FIBに同じプレフィクスが存在する場合は挿入しない。図3では、位置プレフィクス  $/3/0$  を挿入する例である。上述した条件を満たすプレフィクス  $/3/1$ 、 $/3/2$ 、 $/3/3$  をFIBへ挿入する。

## 3.5 議論

3.3節の位置更新はネットワーク負荷、3.4節のFIBの構築はフォワーディング性能の観点で、それぞれ、以下のトレードオフを有する。まず、3.3節の位置更新方式は、セルの単位で更新する場合と比べ、位置更新メッセージの数を削減可能であるが、冗長に転送されるInterestの数が増加する。例えば、セル  $/1/2/3$  を位置プレフィクス  $/1/2$  に集約し更新すると、Interestの位置名  $/1/2/2$  が冗長に  $/1/2$  に一致し転送されてしまう。次に、3.4節のFIBの構築方式は、前方一致ルールを採用するFIBと比べ、検索の計算量を削減できるが、展開したプレフィクスの数分、FIBのサイズを増加する。これらのトレードオフは5章で評価する。

## 4. 1対多の関数呼び出し

### 4.1 要求条件

前章では、計算要求を位置の範囲に基づきエッジルータへ転送するルーティング/フォワーディングを設計した。本章では、計算要求に対し複数エッジルータから計算結果を取得するため、RICEを以下の要件に基づき応用する。第一に、1つの計算要求に対し、複数の計算結果を取得できる必要がある。第二に、NDNの1要求対1応答の通信原理を崩さない必要がある。第三に、本サービスをNDNベースのコア網上で実現したとしても、他のNDNベースのサービスは改変することなく展開できる必要がある。すなわち、NDNに対し少なくとも後方互換性を実現する必要がある。

これらの要件に対し、RICE [12] は第二、第三の要件を満たすが、第一の要件を満たすことができない。本章では、まず、RICEの通信シーケンス、RICEにおける制限を説明後、RICEを1ユーザ対多計算ノードへ拡張する。

## 4.2 RICE

### 4.2.1 概要

RICE [12] は、NDN網で、1ユーザ対1計算ノードの間で、関数の実行に必要な引数と実行結果を交換するプロトコルである。RICEでは、引数を、ユーザから計算ノードへプッシュせず、プギーバック、つまり、ユーザからの関数の実行を要求するInterestの受信を起点にサーバがプルすることで、Interestに大容量の情報を含めることを回避している。ここで、引数とは、ユーザの要求する位置の範囲以外に関数の実行に必要な情報であり、2.1節の例ではターゲットの特徴である。また、RICEでは、アプリケーションの長い実行時間に対し、ネットワークのステータス、つまり、NDNルータのPITエントリが長時間残らないよう、関数の実行要求と実行結果の通信を分離している。

RICEの通信は図4のように、3組のInterest、Dataの組をユーザと計算ノード間で交換することで実現される。まず、ユーザがInterest  $I_1$  を送信し関数の実行を要求する。 $I_1$  の名前は3.2節の命名規則に従うとし、例えば  $/1/2/\#f : detect$

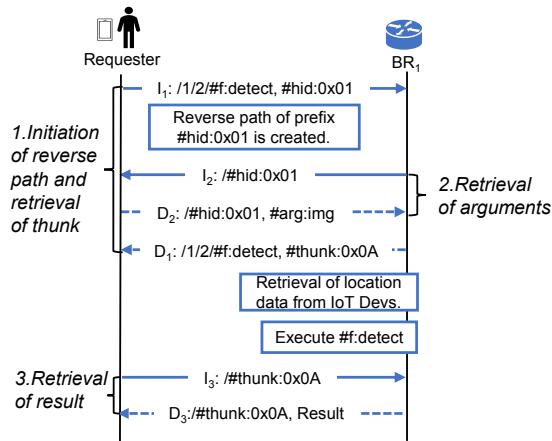


図4 1 ユーザ対1 計算ノード間の RICE の通信シーケンス

である。ただし、 $I_1$  は RICE に必要なオプションのヘッダとして、タグ  $\#hid$  : により計算要求を一意に識別するハンドシェイク ID  $\#hid : 0x01$  を保持する。なお、RICE の処理はこのヘッダをトリガーとするため、通常の NDN のパケットは RICE の処理なく転送され後方互換性が保証される。ルータは  $I_1$  を受信すると、RICE のヘッダを有する場合、 $\#hid : 0x01$  と受信インタフェースの組を経路表へ挿入し、計算ノードが引数を要求するための経路を構築する。計算ノードは  $I_1$  を受信すると、名前  $/\#hid : 0x01$  の Interest  $I_2$  を送信し引数を要求する。ルータは  $/\#hid : 0x01$  の経路に沿って  $I_2$  を転送する。ユーザは  $I_2$  を受信すると名前  $/\#hid : 0x01$  で引数をペイロードにもつ Data  $D_2$  を返送する。計算ノードは  $D_2$  を受信すると実行結果の一意な識別子である  $\text{thunk } \#thunk : 0x0A$  をペイロードに持つ Data  $D_1$  をユーザに返送する。その後、計算ノードは自身の IoT ドメインから位置データを収集し関数を実行する。

最後に、ユーザは  $\text{thunk } \#thunk : 0x0A$  を名前とする Interest  $I_3$  を送信し、計算ノードは実行結果を有する Data  $D_3$  を返送する。 $I_3$  の転送は、RICE [12] やその応用例 [14] と同様に、計算ノードの識別子を Interest の Forwarding Hint フィールド [15] に格納し、この識別子に基づいて行う。なお、全ての Data は NDN の PIT に基づいて転送される。

#### 4.2.2 課題

3 章の方式では、ユーザの送信した  $I_1$  1 つあたり複数の計算ノードへ転送される。例えば、図 2 の例では、 $I_1$  が位置の範囲  $/1/2$  を指定し計算を要求すると、セル  $/1/2/0$  を収容する  $BR_1$ 、セル  $/1/2/2$  を収容する  $BR_2$  へ転送される。この場合、ユーザは  $BR_1$ 、 $BR_2$  から関数の実行結果を取得したいが、一方しか取得できないことが問題となる。

具体的には以下の通りである。実行結果の取得には、 $BR_1$ 、 $BR_2$  から  $\text{thunk}$  を含む  $D_1$  を 1 つずつ計 2 つ返送される必要がある。しかし、対応する  $I_1$  はユーザから 1 つしか送信されないため、ルータの PIT が 1 つの  $D_1$  しか返送しない。つまり、ルータは 2 つの同じ位置名  $/1/2$  を有

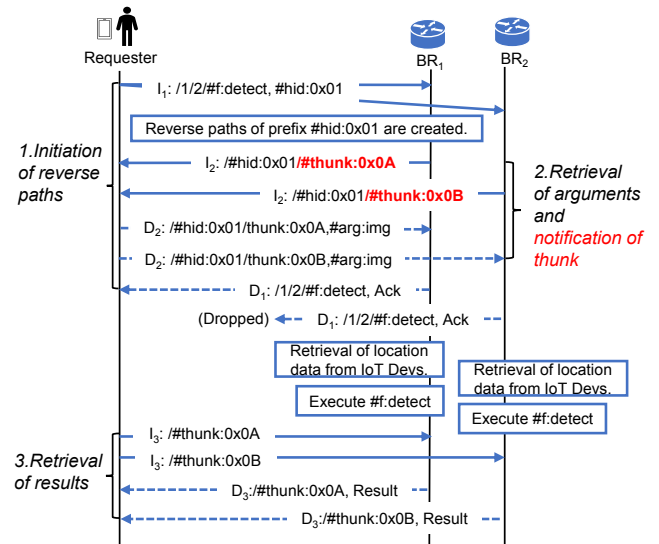


図5 1 ユーザ対多計算ノードの RICE の通信シーケンス

する  $D_1$  を受信するが、先に受信した  $D_1$  を転送した後に PIT エントリを削除するため、後に受信した  $D_1$  を転送しない。この問題を、PIT エントリをタイマーベースで削除する方式 [11] により解決する場合、1 要求対 1 応答の制約を崩す、つまり、要件 2 を満たせない。

#### 4.3 1 ユーザ対多計算ノードへの拡張

$I_1$  と  $I_2$  の通信シーケンスは 1 対 1 の対応関係にないことを利用し、 $\text{thunk}$  を  $I_2$  の名前に含めて配送することで、1 要求対 1 応答の制約下で 1 つの  $I_1$  から複数の  $\text{thunk}$  を取得する。図 5 は 1 ユーザ対 2 計算ノード  $BR_1$ 、 $BR_2$  間の RICE の通信シーケンスである。まず、4.1.1 節と同様に  $I_1$  がユーザから送信され、 $\#hid : 0x01$  の経路がルータで生成された後、 $BR_1$ 、 $BR_2$  へ転送される。次に、 $BR_1$ 、 $BR_2$  は、それぞれ、 $\#hid : 0x01$  の末尾に  $\text{thunk } \#thunk : 0x0A$ 、 $\#thunk : 0x0B$  を結合した名前  $/\#hid : 0x01/\text{thunk} : 0x0A$ 、 $/\#hid : 0x01/\text{thunk} : 0x0B$  の Interest  $I_2$  を送信する。これら 2 つの  $I_2$  の名前はルータの経路表で  $\#hid : 0x01$  に最長一致し、2 つともユーザへ転送され、ユーザは 2 つの  $\text{thunk}$  を受信する。その後、ユーザは 2 つの  $I_2$  に対し引数をペイロードとする  $D_2$  をそれぞれ返送する。最後に、 $BR_1$ 、 $BR_2$  は  $D_2$  を受信すると空のペイロードの  $D_1$  を返送し、 $I_1$  により生成された PIT を削除する。 $BR_1$ 、 $BR_2$  が関数を実行後、4.1.1 節と同様に、ユーザは  $\#thunk : 0x0A$ 、 $\#thunk : 0x0B$  を名前に持つ Interest  $I_3$  を 1 つずつ送信し、2 つの実行結果を  $BR_1$ 、 $BR_2$  から取得する。

4.1 節の要件について、本節の拡張は RICE の  $I_2$  の名前を変更するのみであるため、拡張前の RICE が満たす要件 2、3 を満たすと同時に、要件 1 も満たしている。

## 5. 評価

3.5 節で述べた通り、3 章の設計について、3.3 節の位置

更新の方式で生じるネットワーク負荷、および、3.4節の位置名の前方一致検索を最長一致で実現する方式のフォーワーディング性能をシミュレーションで評価する。

## 5.1 評価シナリオ

位置ベースサービスのサービス範囲として、12km 四方の正方形で囲まれたルクセンブルク市内を想定する。この正方形を再帰的に 8 回 4 分割し、分割した領域に Z 記法を用いた名前付けを行い、各セルは約 40m 四方の正方形となる。携帯基地局の座標データ [16] を元に 73 台の基地局を配置し、基地局をコア網のエッジルータとする。これらエッジルータはコア網を通じて相互接続される。コア網のルータレベルのトポロジは 4 段のツリーであり、4 段目に 73 台のエッジルータ、ルクセンブルクの市内を一樣にカバーできるよう 3 段目に 13 台、2 段目に 4 台のルータ、1 段目にはルクセンブルク市内の中心となるセルに 1 台のルータを設置し、下段のルータは最も地理的に近い上段のルータへ接続される。ユーザは 1 段目のルータを経由し、計算を要求する。エッジルータの配下に存在する IoT デバイスとして、自動車に設置されたカメラを想定する。IoT デバイスの位置は、Simulation Urban MObility を用いたルクセンブルクのトラフィックシナリオ [17] より決定する。

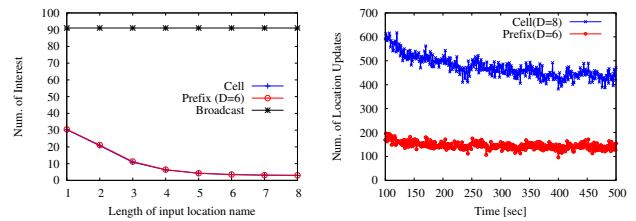
ユーザは、サービス範囲を Z 記法で記載したマップを事前に取得しているものとし、ユーザの要求する位置の範囲は以下のように決定する。まず、マップから中心となるセルを 1 つランダムに選ぶ。次に、8 桁のセルの名前に対し、1 から 8 桁の中からランダムに 1 つ位置名を選び、その位置名で表現される位置の範囲をユーザの要求する位置の範囲とする。ユーザは 1 秒当たり 1 回計算要求を送信し、500 秒のシミュレーションを行う。なお、要求する位置の範囲に IoT デバイスが 1 台も存在しない場合の計算要求はシミュレーションから排除する。

## 5.2 位置更新方式のネットワーク負荷

まず、設計したプレフィクス単位の位置更新方式を、IoT デバイスがセルを移動するたびに位置を更新する方式、IoT デバイスの移動管理を一切せず Interest をブロードキャストする方式 [18] と比較する。以降では、それぞれを Prefix( $D=6$ ) 方式、Cell 方式、Broadcast 方式と呼ぶ。ここで、Prefix( $D=6$ ) 方式では、プレフィクス長  $D$  として最も性能の良い 6 を設定する。Cell 方式は、セル単位で Interest の転送先を制御するため、Interest の総数は最も少ないが、位置更新メッセージの数が多い。一方、Broadcast 方式は、各ルータで Interest が全ての隣接ルータへブロードキャストされるため、Interest の総数が最も多いが、位置の管理をしないため、位置更新メッセージの数は 0 である。

### 5.2.1 Interest 数

図 6a は、各方式を採用した際の、1 回の計算要求につ



(a) コア網における計算要求 1 回あたりの Interest の総数 (b) コア網における 1 秒あたりの位置更新メッセージ数

図 6 Prefix( $D=6$ )、Cell、Broadcast 方式のネットワーク負荷

きコア網を流れる Interest の総数である。ここで、Interest の総数とは、ルータ間を接続する全てのリンクを流れる Interest 数の総和である。なお、計算要求が指定する位置の範囲に応じて、Interest の転送される範囲が変わるため、図 6a では位置の範囲を表す位置名の長さ毎の結果を掲載する。この結果より、以下の 2 つの考察を得た。第一に、Broadcast 方式の Interest の総数が多く、ネットワーク負荷の観点より FIB で IoT デバイスの位置を管理する方式は必要である。第二に、本稿で提案した Prefix( $D=6$ ) 方式の Interest の総数は Cell 方式と同等であり、3.5 節で議論した冗長な Interest の転送は無視できるほど少ない。

### 5.2.2 位置更新メッセージ数

図 6b は、Prefix( $D=6$ ) 方式、Cell 方式について、1 秒あたりにコア網で広告される位置更新メッセージの総数である。なお、本評価では IoT デバイスの位置が 1 秒毎に移動するシミュレーションを行っており、横軸の時刻はシミュレーション開始時刻を 0 とした場合の経過時間を意味する。図 6b には、定常時として時刻 100 から 500 の結果を掲載する。縦軸は各時刻の位置更新メッセージの総数である。本節と前節の結果より、3.3 節で提案したプレフィクス単位の位置更新の方式は、位置更新のメッセージ数を削減でき、冗長な Interest の転送も生じないと考察した。

## 5.3 FIB のフォーワーディング性能

次に、3.4 節の方式で構築した FIB を、前方一致ルールを採用する FIB と比較する。以降では、それぞれを Prefix( $D=6$ ) の FIB、Cell の FIB と呼ぶ。Prefix( $D=6$ ) の FIB は、5.2 節と同様に長さ  $D=6$  のプレフィクス単位で位置を更新し、位置更新メッセージに含まれる位置プレフィクスを 3.4.1 節の方式で展開した全プレフィクスを挿入し、Interest の名前に対し FIB の最長一致検索を行う。一方、Cell の FIB では、3.4 節で述べた通り、セル単位で位置の更新を行うとし、位置更新メッセージに含まれるセル名を挿入し、Interest の名前に対し FIB の前方一致検索を行う。

### 5.3.1 FIB サイズ

図 7a に 2 つの方式の、各時刻における FIB のサイズを示す。ここで、サイズは FIB に格納するプレフィクスの総数である。横軸の時刻は 5.2.2 節と同様にシミュレーション

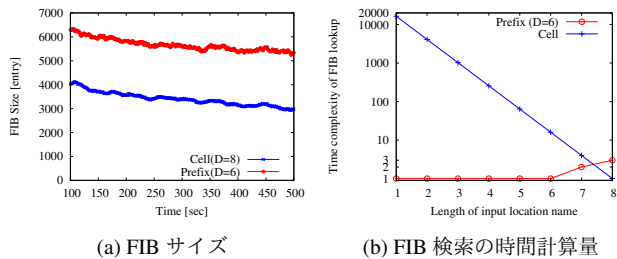


図7 Prefix( $D = 6$ )、Cell の FIB のフォワーディング性能

開始時刻 0 からの経過時間である。縦軸は、各時刻に広告された位置更新の結果構築された FIB のサイズである。なお、最も FIB サイズの大きい 1 段目ルータの結果を掲載している。この結果では、Cell の FIB と比べ、Prefix( $D = 6$ ) の FIB は位置プレフィクスを展開して挿入する分、平均して、FIB のサイズを約  $3 \times 10^3$  プレフィクス増加している。ただし、NDN ベースのコア網の FIB は、ビデオ配信など他のアプリケーションの経路も同時に格納しており、その総数は約  $10^8$  プレフィクスである [19] ことから、この総数に対して、上記の増加は無視できるほど少ないと考察した。

### 5.3.2 検索 1 回あたりの時間計算量

最後に、Prefix( $D = 6$ ) と Cell の FIB の間で、位置名の検索に必要な時間計算量を比較する。ここで、時間計算量とは 1 回の検索で参照するプレフィクスの総数と定義する。セル名の長さ  $N$ 、位置プレフィクスの長さ  $D$  を 8、6 とし、入力する位置名の長さを  $x$  とした場合、Prefix( $D = 6$ ) の FIB の時間計算量は以下の通りである。 $D \leq x$  の場合、長さ  $D$  の位置プレフィクスに一致するため、時間計算量は  $D - x + 1$  である。一方、 $x < D$  の場合、長さ  $x$  の展開プレフィクスに一致するため、時間計算量は 1 となる。一方、Cell の FIB では長さ  $x$  から長さ  $N$  まで、各長さで 0 から 3 の値を接尾辞にもつプレフィクスを全て参照し一致を確認するため、 $x$  によらず計算量は  $4^{(N-x)}$  となる。したがって、位置名の長さ  $x$  と時間計算量の関係は図 7b の通りとなる。この結果より、前方一致する FIB の計算量が大きく、コア網の FIB には最長一致の FIB が望ましいと考察した。

## 6. おわりに

本稿では、1 ユーザ対多エッジルータの間で位置に基づく計算の要求-応答の通信を NDN の制約下で実現した。まず、位置の範囲を Z 記法に基づき次元の階層的な名前前で表し、名前の最長一致検索のみで計算要求をエッジルータへ転送可能なルーティング/フォワーディングを設計した。次に、ユーザからの計算要求とエッジルータからの引数要求の通信が分離される RICE の性質を利用し、計算結果の取得に必要な識別子を引数要求の名前に含めて配送する拡張を RICE に加えることで、1 対 1 の要求-応答通信の制約下で複数エッジルータからの計算結果の取得を実現した。

謝辞：本研究は NICT 受託研究課題 191 によるものである。

## 参考文献

- [1] Grozev, N. and Buyya, R.: Inter-Cloud architectures and application brokering: taxonomy and survey, *Software Practice & Experience*, Vol. 44, pp. 369–390 (2014).
- [2] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., kc claffy, Crowley, P., Papadopoulos, C., Wang, L. and Zhang, B.: Named Data Networking, *ACM SIGCOMM Computer Communication Review*, Vol. 44, No. 3, pp. 66–73 (2014).
- [3] Król, M. and Psaras, I.: NFaaS: named function as a service, *Proceedings of ACM ICN* (2017).
- [4] Ascigil, O., né, S. R., Xylomenos, G., Psaras, I. and Pavlou, G.: A Keyword-based ICN-IoT Platform, *Proceedings of the ACM ICN*, pp. 22–28 (2017).
- [5] Kurihara, Y., Koizumi, Y., Hasegawa, T. and Arumaiturai, M.: Location-Based Forwarding with Multi-Destinations in NDN Networks, *IEICE Transactions on Communications*, Vol. 102.B, pp. 1822–1831 (2019).
- [6] Chen, J., Jahanian, M. and Ramakrishnan, K. K.: Black Ice! Using Information Centric Networks for Timely Vehicular Safety Information Dissemination, *Proceedings of IEEE LAN-MAN* (2017).
- [7] Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B. and Zhang, L.: A Case for Stateful Forwarding Plane, *Computer Communications*, Vol. 36, No. 7, pp. 779–791 (2013).
- [8] Navas, J. C. and Imielinski, T.: GeoCast - Geographic Addressing and Routing, *Proceedings of ACM MOBICOMM*, pp. 66–76 (1997).
- [9] Morton, G. M.: A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing, Technical report, IBM (1966).
- [10] Meijerink, B., Baratchi, M. and Heijenk, G.: Design & analysis of a distributed routing algorithm towards Internet-wide geocast, *Elsevier Computer Communications*, Vol. 13, pp. 201–218 (2019).
- [11] Amadeo, M., Campolo, C. and Molinaro, A.: Multi-Source Data Retrieval in IoT via Named Data Networking, *Proceedings of ACM ICN*, pp. 67–76 (2014).
- [12] Król, M., Habak, K., Oran, D., Kutscher, D. and Psaras, I.: RICE: Remote method invocation in ICN, *Proceedings of ACM ICN* (2018).
- [13] Lehman, V., Hoque, A. K. M. M., Yu, Y., Wang, L., Zhang, B. and Zhang, L.: A Secure Link State Routing Protocol for NDN, *NDN, Technical Report, NDN-0037*, pp. 10470–10482 (2014).
- [14] Król, M., Mastorakis, S., Oran, D. and Kutscher, D.: Compute First Networking: Distributed Computing meets ICN, *Proceedings of ACM ICN*, pp. 67–77 (2019).
- [15] The Named Data Networking project: NDN Packet Format Specification, <https://named-data.net/doc/NDN-packet-spec/0.3/>.
- [16] The National Geopotal of the Grand-duchy of Luxembourg: geoportal.lu, <https://www.geoportal.lu/>.
- [17] Codecá, L., Frank, R., Faye, S. and Engel, T.: Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation, *IEEE Intelligent Transportation Systems Magazine*, Vol. 9, No. 2, pp. 52–63 (2017).
- [18] Wang, J., Wakikawa, R. and Zhang, L.: DMND: Collecting data from mobiles using named data, *Proceedings of IEEE VNC*, pp. 49–56 (2010).
- [19] Song, T., Yuan, H., Crowley, P. and Zhang, B.: Scalable Name-Based Packet Forwarding: From Millions to Billions, *Proceedings of ACM ICN*, pp. 19–28 (2015).