

修士論文

バンディットアルゴリズムを用いて ICN 上の 通信経路変化に対応するパケット転送戦略の 提案と評価

平成 31 年 1 月 30 日提出

指導教員 相田 仁 教授

工学系研究科 電気系工学専攻

37-176493 山内 智晴

概要

Information Centric Network (ICN) は、ネットワーク上の通信において宛先の情報を“通信相手となるホスト”ではなく、“取得したい情報”によって指定する新しいタイプのアーキテクチャである。これに対して、既存の IP アドレスを用いるネットワークアーキテクチャを Host Centric Network (HCN) と呼ぶことがある。通信の目的がある特定のコンテンツの取得である場合、取得するコンテンツが正当なものでありさえすれば、“通信相手のホストが誰であるか”ということはそれほど重要ではない。この点において、ICN では同一のコンテンツをネットワーク上にキャッシュとして分散配置し、通信相手を指定せずにキャッシュの中のどれか一つを取得する、というやり方を取る。キャッシュが分散配置されていることによって、HCN に比べて遅延が低減されたり、障害時の冗長性が増したりすることが期待されている。

一方で、通信相手が固定されなくなることやネットワークトポロジと関連づけられた情報である IP アドレスのような符号を用いないことで、パケット転送における経路選択が非常に困難なものとなる点が ICN における課題の一つである。コンテンツが分散配置されている上、新たにキャッシュされたり既存のキャッシュが削除されたりすることでその配置自体が変化していく。ネットワーク上のあらゆる場所にコンテンツが存在しうするため、IP アドレスをベースとしたルーティングの様にある程度ネットワークトポロジと関連付けた経路選択をすることも困難である。

本研究では、これらの問題に対応しつつ、コンテンツ配置の変化や回線状況の変化に伴って必要に応じて選択する経路を変化させることのできる通信経路選択の手法を提案する。複数の経路間で経路の切り替えを決断するためには、切り替え前後の両方の経路について通信状態がわかっている必要がある。ICN 上で経路の通信状態を把握するためには、実際にその経路を利用する必要があるため、経路の切り替えに備えて利用していない経路に関しても通信状態だけは把握しておく必要があるということになる。この点について本研究では、パケット転送における経路選択と多腕バンディット問題との間に類似点があることを指摘し、多腕バンディット問題を解くのに用いられるバンディットアルゴリズムを参考にした手法を提案している。

本研究の目的は、経路選択手法を提案した上でシミュレーションによる動作の確認を行うこと、また、既存手法との比較を行いその優位性を評価することにある。実際に ICN をシミュレートできるネットワークシミュレータ上で提案手法を実装し、複数の状況を想定したシナリオでシミュレーションを行った。得られた結果を基に既存手法である Bestroute Strategy との比較を行い、提案手法が通信状況の変化が多い状況下でも適切な経路選択を行いつつ通信を継続できることを確認した。また、通信速度という観点からは既存手法と比べてより良い通信状況を維持できることも確認された。

目次

1	はじめに	1
1.1	本研究の目的	1
1.2	本論文の構成	1
2	Information Centric Network	2
2.1	本章の概要	2
2.2	ICN の概説	2
2.3	Named Data Network について	3
2.4	NDN 上での通信フロー	4
2.5	ICN が抱える課題	8
3	ICN におけるパケット転送戦略	9
3.1	本章の概要	9
3.2	既存のパケット転送戦略とその分類	9
3.3	ICN 上でのパケット転送戦略における課題	18
3.4	多腕バンディット問題とその解法	20
3.5	多腕バンディット問題とパケットフォワーディングの類似点	22
4	提案手法	24
4.1	本章の概要	24
4.2	バンディットアルゴリズムを参考にしたフォワーディング戦略の提案	24
4.3	初期化に伴う問題と解決策	27
5	提案手法の評価	30
5.1	本章の概要	30
5.2	シミュレーション設定	30
5.3	提案手法の動作の評価とパラメータの与える影響	32
5.4	安定した通信状況のシナリオにおける評価	36
5.5	通信状況の変化を伴うシナリオにおける評価	37
5.6	総合的な評価	39
6	おわりに	40
	謝辞	41
	参考文献	42
	発表文献	44

1 はじめに

1.1 本研究の目的

本研究では、次世代のネットワークアーキテクチャとして研究がなされている Information Centric Network (ICN) 上のパケット転送における経路選択手法を扱う。ICN は、通信の宛先を“通信相手となるホスト”ではなく“目的とするコンテンツ”によって指定する。取得するコンテンツが正当なものでありさえすれば、そのコンテンツを取得する先のホストが誰であるかというのはそれほど重要なことではない。そこで ICN では、コンテンツをキャッシュとしてネットワーク上に分散配置し、通信相手のホストは指定しないことでネットワーク上のキャッシュのどれか一つを取得する、というやり方をとる。分散配置されたキャッシュの中から適切なものを選択すれば、低遅延での通信が可能であると期待されている。また、コンテンツが分散配置されていることは障害時の冗長性にも寄与すると考えられている。

しかし、この恩恵を受けるためには、キャッシュ配置の変化や障害の発生に合わせて、動的に通信経路を変化させるようなパケット転送におけるルーティング戦略が必要不可欠である。加えて、宛先となるホストの情報を持たないパケットをやりとりするという ICN のアーキテクチャそれ自体が、ルーティングを困難なものとしている。

通信経路の切り替えを決断するためには、切り替え前後の両方の経路について通信状況がわかっている必要がある。これを実現するためには、使用していない経路に関しても通信状況を取得するための“探索”を行うようなルーティング手法が必要である。

本研究では、このような条件を満たすルーティング手法の提案を行い、既存手法との比較を通じてその優位性を明らかにすることを目的とする。

1.2 本論文の構成

本論文は 6 章からなる。前半部の第 1 章から第 3 章までは提案手法に至るまでの前提となる事項についての解説である。第 1 章は本章であり、研究全体の概要、目的と本論文の構成を明らかにしている。第 2 章では ICN というネットワークアーキテクチャ全体に関する解説を行う。第 3 章ではこれを踏まえて、本研究の主たるテーマである ICN 上でのパケット転送に着眼点を絞りその課題を明らかにする。また、パケット転送と多腕バンディット問題との関連性についてもここで触れる。

後半部の第 4 章から第 6 章まででは本研究での提案手法を示し、その評価を行う。第 4 章では、前半部で触れた課題を踏まえて本研究における提案手法を提示する。第 5 章では実際にシミュレータ上に実装した提案手法を用いて実験を行い、提案手法の性能を確認するとともに、既存手法との比較を行う。第 6 章は本論文全体のまとめである。

2 Information Centric Network

2.1 本章の概要

本章では本研究の主要なテーマである ICN についての解説を行う。はじめに 2.2 節で ICN の設計上のコンセプトや既存ネットワークアーキテクチャに対する優位性、期待される用途等について概観する。その後、2.3 節では ICN の具体的な実装の提案の一つであり、シミュレータなどの研究環境が充実している Named Data Network (NDN) と呼ばれるアーキテクチャに簡単に触れる。これを受けて、2.4 節で ICN 上での通信の行われ方の詳細を NDN での実装を例に確認する。最後に 2.5 節では ICN が抱えている課題を明らかにする。

2.2 ICN の概説

今日、情報通信技術の進歩によってユーザが自ら作成したコンテンツをソーシャルネットワークサービスなどを利用して他者と共有することが容易になってきている。それに伴いネットワーク上を流れるトラフィックの急激な増加に対する対応がインターネットの抱える課題の一つとなっている。一方で現在のインターネットで用いられている TCP/IP プロトコルスイートではコンテンツの送受信を行うために IP アドレスに基づいて相手ホストとの接続を確立するという手順が必要である。TCP に限らず現在の IP ネットワーク上で動くプロトコルでは IP アドレス、すなわちホスト端末を基準に通信相手を選択する。このような文脈では既存の IP アドレスベースのネットワークシステムを Host-Centric Network (HCN) と呼ぶが、これは今日のコンテンツ中心的なアプリケーションに適しているとは言い難いと指摘されている [1]。

ネットワーク上での通信は多くの場合“取得したいデータの要求”と“要求されたデータの応答”から成っている。HCN において、これらの通信はパケットヘッダに含まれる IP アドレスという形で事前に宛先を指定された状態で始まる。IP アドレスは個々のホストに紐づけられたものであるから、HCN 上での通信はホストに紐づけられたものであるということが出来る。加えて、IP アドレスの割り当てはネットワークトポロジとも関連がある。これらを利用することで、HCN でのパケットルーティングは効率よく行われている。

一方で、通信相手のホストがどれであるか、というのは本質的な問題ではなく、実際にユーザが入手したいのはコンテンツであって、特定のサーバに接続するというのは本来必須のアクションではない（コンテンツが存在する場所を意識する必要はない）という考え方がある。単一のホストを通信相手とするよりも、むしろ複数の通信経路、通信相手が利用できることが負荷分散、通信遅延低減などに貢献することが多く、通信相手が固定されてしまうことはある種のロスであると言える。

そこで、HCN とは対照的に目的とするコンテンツを直接探せるようなネットワークシステムが考えられている。これを ICN と総称する。HCN における IP アドレスに対応する識別子として、ICN ではコンテンツ名を用いる。コンテンツ名を用いることで通信相手のホストを意識することなく、目的のコンテンツが存在する任意の場所からコンテンツを入手することが可能となる。これと合わせて、ネットワーク上でコンテンツをキャッシュするノードを多数配置し、オリジナルのコンテンツデータのキャッシュを保存できるようにする。これによってネットワークアーキテクチャレベルでのコンテンツの分散配置が可能になる。

このような技術を用いることで、現代のインターネットで課題とされている多くの問題に対して改善が可能であると見込まれている。具体的には、ICN の導入によって

- 通信に対する応答速度の向上
- 移動体通信に対する適応性の向上
- マルチパス通信に対する適応性の向上
- サーバ運用コストの低減
- サーバに対するトラフィック負荷の低減
- ホストに依存するセキュリティ脆弱性の低減

などの効果が期待されている [2][3]. ICN はこれらのメリットを実現するための設計上の課題を解決すべく研究が進められている段階である.

実用化, という観点では ICN をポストインターネットアーキテクチャとして捉える見方も存在はしているが, ICN のアーキテクチャは HCN のそれとは大きく異なっている部分が多い. それゆえ, 例えば既存のインターネットを全面的に ICN のアーキテクチャに置き換えるといった作業には困難が予想される. その代わりに, IP ネットワークの一部分を ICN アーキテクチャに置き換え, 既存の IP ネットワークとの間に透過性を持たせて IP ベースのアプリケーションも ICN 上で動作可能にする IP over ICN [4] や TCP/ICN [5] といった技術も提案されている.

また, ICN は特にモバイル環境をはじめとする多ノード環境や移動体間の通信で有効であると考えられている. 例えば, 近年ではコネクテッドカーなどによる車両からのデータ収集と共有というアイデアが注目されているが, ICN をこのような自動車間無線通信ネットワーク (Vehicular Ad-hoc Network, VANET) に用いるという提案がなされている. このような環境においては各端末に固定的な IP アドレスを, ネットワークトポロジと紐づけて割り振り管理することは困難である. また, あるノードが他のノードからデータを取得する場合にも, どのノードが所望のデータを持っているか分からなければこれは不可能である. 結局, クラウドサーバやエッジサーバを設置してそこでデータを管理することになるが, このような通信に関わるオーバーヘッドやサーバへの負荷は無視できない問題である. 加えて, サーバとの通信に既存の携帯電話回線などを利用しようとする場合, 多数のノードが生み出す膨大な応報量がインフラにもたらす負荷が深刻な問題となってくる. ICN のアーキテクチャを導入し, ノード間の直接通信によってデータをやり取りすることができれば, これらの問題の解決に繋がると考えられている.

2.3 Named Data Network について

Jacobson らが提唱した Content Centric Network (CCN) 及びその発展系の NDN [2] は ICN の実装の 1 つであり, ネットワークシミュレータ ns3^{*1}で動作するシミュレーション環境 NDNsim^{*2}が用意されているなど研究対象としてもよく扱われるものとなっている. なお, NDN の他にも DONA [6], NetInf [7] といった実装も提案されている.

本研究では, 第 5 章で提案手法を実際に実装してシミュレーションを行う際に NDNsim を用いるのをはじめとして, NDN を ICN の実装の代表例として扱っていく. また, 次節以降で説明していく ICN 上での通信フロー, その他 ICN の具体的な実装・動作に関する事項は, 特別に断りがなければ NDN でのものである.

^{*1} <https://www.nsnam.org>

^{*2} <http://ndnsim.net/current/>

2.4 NDN 上での通信フロー

2.4.1 全体像

本章では NDN 上での通信の流れについて、必要に応じて HCN でのそれと比較しながら解説する。先述の通り、ここで解説する実装方式については NDN のものだが、概念としては ICN に共通である。

ICN 上では、通信は必ず“ユーザによるコンテンツの要求”とそれに対する“コンテンツ所有者の応答”の組からなる。以下では、コンテンツ発行、要求と応答までを順に従って概説した後、通信におけるパケットのやり取りで重要な役割を担うルータの動作について確認する。

2.4.2 コンテンツの発行

まず初めに、コンテンツ提供者によるコンテンツの発行を考える。2.2 節で書いた通り、ICN では通信の宛先としてコンテンツ名を用いる。従ってコンテンツの発行者はネットワーク上においてユニークなコンテンツ名を設定する必要がある。これはいわば、HCN においてコンテンツにユニークなドメイン名、ファイル名を与えることに相当する。ネットワーク上でユニークであるという条件を満たすコンテンツ名設定の仕方については一つの研究テーマとされているところである。例えば、NDN では後述するルーティング等での利便性も考慮して既存ネットワークのドメイン名のように階層型の命名を行うことが検討されている。生成されたコンテンツは、最初の段階ではコンテンツ提供者のサーバに保存される。その後コンテンツは徐々に拡散していくことになるが、この大元のサーバをそのコンテンツに対するオリジンサーバと呼ぶ。

2.4.3 コンテンツの要求

続いて、発行されたコンテンツをユーザが要求する段階を考える。ICN ではコンテンツを要求するユーザ側のノードを *consumer* と呼ぶ。コンテンツを取得する際、*consumer* はコンテンツ要求パケットに取得したいコンテンツ名の情報を載せてネットワーク上に送信する。このコンテンツ要求パケットのことを *Interest* と呼んでいる。*Interest* は、要求するコンテンツ名を情報として持つ非常に小さなサイズのパケットである。言い換えると、*Interest* パケットには送信元ノード、送信先ノードなどの情報が含まれていない。そもそも、純粋な ICN においてはネットワーク層レベルで任意のノードを一意に識別するようなアドレスに当たるものは割り振られないためである。一般的に IP ネットワークでやり取りされるパケットは送信元や送信先の IP アドレスを含んでいるため、それとは根本的に異なったものである。*Interest* にはこの他、タイムアウトを判定するための生存時間や、個々の *Interest* を区別するために与えられたランダムな値である *Nonce* などが含まれている。

2.4.4 Interest の転送

consumer が送信した *Interest* は、既存の HCN 同様経路上にあるルータを経由して目的のノードへと転送されていく。すなわち、*Interest* パケットを受信したルータはそのデータを適切な経路に再度転送するという処理を行う必要がある。このようにパケットを転送していくことをパケットフォワーディング、特に適切な経路を決定することをパケットのルーティングと呼ぶ。HCN では、ネットワーク上を流れるあらゆるパケットが宛先 IP アドレスの情報を持っていることから、ルータはこれと自身の持つルーティングテーブルに保存されたデータを参照して、IP アドレスごとに適切な経路にパケットを転送することでルーティングを実現している。さらに言えば、IP アドレスがネットワークポロジにある程度対応した階層構造に基づいて割り振られていることから、ルーティ

ングテーブルの構築も比較的容易に行うことができる。ルーティングテーブルには通常、宛先 IP アドレスごとの送信経路の候補と、それらに対するスコアが保存されている。ルータは使用可能な経路候補のスコアを参考に経路を選択して転送を行う。

一方 ICN では、Interest パケットが持つ情報は要求するコンテンツ名だけである。更に、前述の通りコンテンツはオリジンサーバのみならずネットワーク上に分散配置されていることが前提となっている。従って、ルータの動作は、理想的には「その時点で自身から最もアクセスしやすい位置にある目的のコンテンツを持つノードに向けてパケットを転送する」というものでなければならない（ネットワーク上の複数の取得先から一つを選択しなければならない）。なお、“アクセスしやすい”と書いたルーティングの基準としては「目的のコンテンツを持つノードまでのホップ数」あるいは「コンテンツデータを得られるまでにかかる時間」が多く用いられる。Interest を送信してからデータを実際に受信するまでの時間を Round Trip Time (RTT) と呼び、RTT が小さくなる転送先が良い転送先であると考えるのが後者である。それに対し前者は、ホップ数が同じであればその中での通信遅延の差は無視される。よって、厳密に最小の時間で通信を完結させることはできない場合もあるが、事前にネットワークトポロジがわかっている状態であればスコアリングがしやすい、というメリットを持つ。また、これらに加えて通信の安定性や経路の信頼性が考慮されることもある。これらの基準によってスコアリングされた経路の候補は HCN のルーティングテーブルに当たるテーブルに保存されており、HCN と同様そのテーブルを参照しつつ転送を行う。

本稿で主要なテーマとなるのはこのルータのパケットルーティング処理の部分である。この処理は Interest パケットが実際に要求するデータを持っているノード（オリジンサーバ、又はコンテンツデータをキャッシュしたノード）にたどり着くまで、経由するルータごとに繰り返される。ICN において、各ルータは受け取った Interest の次転送先のみを決定し、それ以上先の経路に関与することはない。

2.4.5 データの返送

Interest パケットを受けとったノードが、その Interest パケットが要求しているコンテンツのデータを自身で保持している場合はそのデータを返送する処理に入る。ICN において、この役割を担うことになったノードは producer と呼ばれる。producer は返送するコンテンツデータを含むパケットを作成し、ネットワーク上に送信する。このパケットは Data と呼ばれる。

HCN の場合、ある要求パケットに対応する応答パケットは、要求パケットと同じように宛先ノードの情報を持ち、要求パケットが転送されてきたのと同じように経路上のルータによって転送されつつ元のノードにたどり着く。パケットの構造やルーティングのされ方、という観点で見たとき、要求パケットと応答パケットは基本的に同じものであると言って良い。

それと異なり、ICN の Interest パケットには送信元ノードに関する情報が一切含まれていないため、必然的に Data はその目的地となるべき consumer の情報を一切持つことができない。従って、Data は Interest とは異なった方法によりネットワーク上でルーティングされ、consumer に送り返されることになる。具体的には、Data は Interest が転送されてきた経路を逆に辿ることによって consumer ノードまで導かれることになっている。これを実現するために、ネットワーク上のあらゆるルータは自身が転送した全ての Interest について、自身のどのネットワークインターフェースから受信したかを（それに加えてどのネットワークインターフェースに送信したかも）保存している。こうしておくことで、Data を受信したルータが、“それに対応する Interest を受信した方向”に向けて Data を転送する、という処理を繰り返すだけで、Data は consumer ノードに到達することができる。また、ルータは Data を転送する際にその Data パケットに含まれるコンテンツのデータをキャッシュすることができる。

これによって先述のコンテンツキャッシュの分散配置が進められる。コンテンツをキャッシュしているルータは、オリジンサーバに変わって producer の役割を果たすことが可能となる。このために、ICN におけるルータは HCN のルータよりも機能的に複雑なものであり、これを明確に表すために ICN のルータのことをキャッシュルータと呼ぶこともある (ただし本稿では今後も単にルータと呼ぶ)。どのような場合にコンテンツをキャッシュするか (あるいはキャッシュしたコンテンツをいつ削除するか)、という判断はキャッシングの問題として、ICN における研究課題の一つとされている。

2.4.6 Nack の発行

ICN 上での通信が原則として consumer の発行する Interest とそれに対して応答する Producer の発行する Data の組からなっていることはここまで説明した通りである。ICN 上でこれらの他にやり取りされるパケットとして、ここでは Nack に触れておく。

Nack は HCN でも用いられるもので、Ack (確認応答) に対して “要求に応じられない” 意図を示すために使われる。NDN には通常 Ack の概念はないが、要求に応答できなかったりなんらかのエラーによって Interest を転送できなかったことを示すために Nack が用いられる。具体的に、例えば次のような場合にルータは Interest を転送せず、代わりに Nack を発行して一つ前のルータに送り返す。

- 利用可能な経路候補がテーブルに登録されていない場合
- Interest の生存時間を過ぎている場合
- 既に一度転送したはずの Interest を再度受信した場合

ルータがある Interest を転送した結果、それに対応して Nack を受信した場合、Data の代わりとしてその Nack を転送する。これによって、Nack もまた consumer まで届くことができる。ただし、Nack を受信したのとは別の経路から Data が送られてくる可能性がある場合 (複数の経路に Interest を転送していた場合に発生しうる) には、Nack の転送は一旦保留しその別の経路からの応答を待つ。Data の送られてくる可能性がある全ての経路から Nack を受信した場合には Nack の転送を行う。

2.4.7 NDN 上でのルータの動作

本節では、ここまで述べてきたパケット通信における ICN の動作を実現するために、ルータが行なっている処理を詳しく見ていく。ルータの動作の全体像は概ね図.1 に示したようなものとなっている。

NDN のルータは、その動作に必要なデータを 3 つのテーブルに分けて保持している。それぞれ、Forwarding Information Base (FIB)、Pending Interest Table (PIT)、Content Store (CS) と呼ばれている。

FIB に含まれる情報は、コンテンツ名とそれに対応する転送経路候補、各経路候補に対するスコアである。Interest を受信したルータはその Interest が要求するコンテンツ名を確認し、それが FIB に保存されているかどうかを検索する。FIB に該当するエントリが存在すれば、それによって Interest パケットを転送すべき経路がわかる。すなわち FIB は HCN におけるルーティングテーブルのような役割を果たす。HCN で、ルーティングテーブルのエントリと宛先 IP アドレスのマッチングを行う際にはいわゆるロングストマッチの手法をとるが、NDN における FIB のエントリ検索に関してもこれは同じである。NDN では階層化されたコンテンツ名を付けることが期待されており、実際に FIB を活用してルーティングを行う際にはロングストプレフィックスマッチによってエントリ検索を行う。

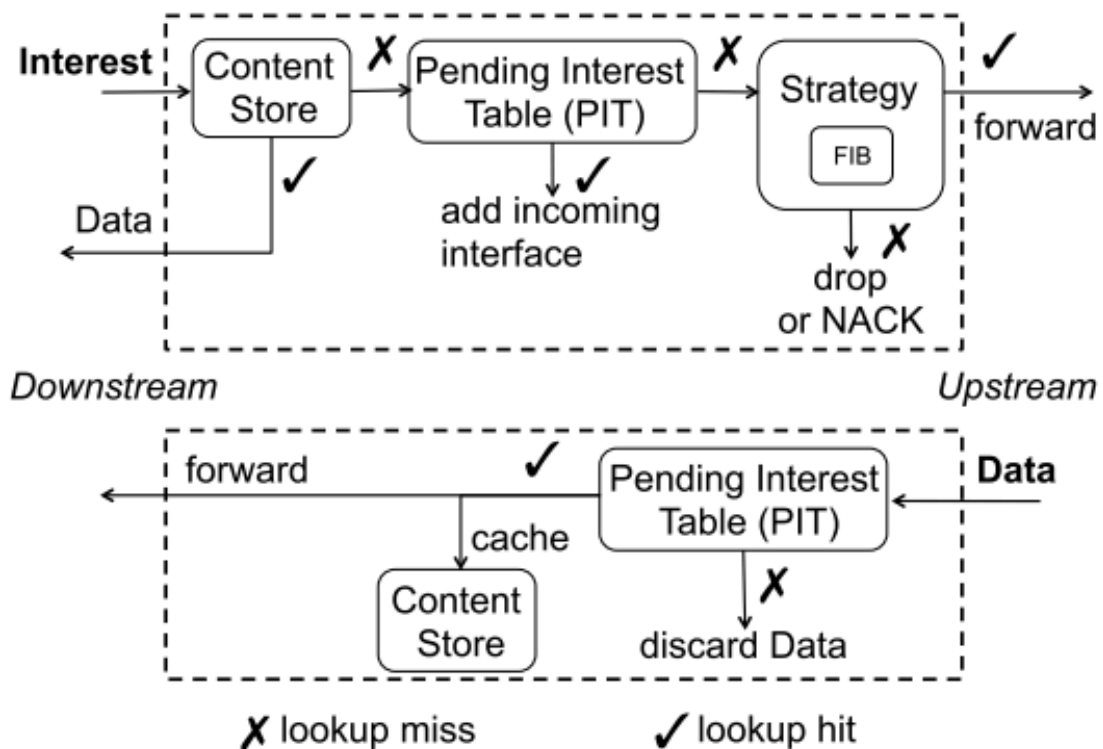


図 1. NDN のキャッシュルータが持つ 3 つのテーブルとその使用順序 ([8] より引用)

ICN における特徴として、一つのコンテンツ名に対して複数の転送先を登録できる、という点が挙げられる。ここでいう転送先は、実際にはフェースという概念によって管理されている。フェースはインターフェースの拡張的概念であり、同一マシン上で動くアプリケーション同士のデータ交換にも用いられることからこのように呼ばれている。ここでは、FIB の持つ転送経路候補のリストのことを転送先フェースリストと呼ぶ。

PIT には、そのルータが既に転送し、まだ対応する Data を受け取っていない Interest の情報が保存されている。これは先述した通り、Data を転送する際に用いる情報である。転送した各 Interest に対し、その Interest をどの Face から受信してどの Face に転送したか、という情報が紐づけられている。ルータは Interest を転送する際に PIT にこれらの情報を保存しておき、Data を受信した際には PIT を参照してその Data の処理を決定する。Data に対応する Interest の情報が保存されていれば Data はその情報に基づいて転送され、PIT 上の該当する情報は必要がなくなるので一定時間後に削除される。PIT に該当する Interest の情報がない場合その Data を転送する必要はないので Data が破棄される。PIT は冗長なパケットの転送を防ぐ役割を果たしている。PIT がない場合個々の Interest や Data が consumer ノードに関する情報を持たねばならなくなる上、宛先は別の consumer だが途中で同一経路で転送されるような Data があってもそれぞれを別のパケットとして用意しなければならない。また、ルータは Interest を転送する際にも一旦 PIT の情報を確認する。同一のコンテンツ名を指定する Interest が既に転送されていれば、その Interest を転送するのは冗長であり改めて転送する必要はない。既に転送済の Interest に対して Data が送り返されてくることが期待されるので、そのようにして得られた Data パケットを後から来た Interest の送信元に対しても転送すれば良いからである。

CS には、そのノードがキャッシュしているコンテンツの情報が保存されている。Interest を受信したルータは

CS を検索し、要求されているコンテンツがそこに登録されていれば自身が **producer** となって **Data** を発行し応答する。このテーブルの情報はルータがコンテンツをキャッシュしたり、キャッシュしていたコンテンツを削除したりする際に更新される。

これらのテーブルの検索スピードや使用するメモリサイズの改善は NDN 全体の性能向上に繋がるため、一つの研究課題となっている。例えば、PIT に関してマッピング・ブルームフィルタを用いることによってメモリ消費の削減を図った [9] のような研究が行われている。

2.5 ICN が抱える課題

ICN は現在も研究途上にあるネットワークアーキテクチャであり、実現のために解決すべき課題が指摘されている。それらの研究課題は大きく以下のように区分される。

- Naming
- Caching
- Routing
- Security

Naming は、あらゆるコンテンツにネットワーク上で一意なコンテンツ名を与えなければならない、という点に起因する研究課題のグループである。ここには、単に一意であることだけではなく、コンテンツ名を階層化することによってルーティングの助けとするなどの課題も含まれる。

Caching は、ルータが受信した **Data** に含まれるコンテンツデータをキャッシュするか否かの戦略に関する研究課題である。現実にはメモリの制約等によりキャッシュできるデータには限りがあるため、データをキャッシュするか否か、およびキャッシュ済みのデータを破棄するか否かを決定することは必要不可欠である。ICN はコンテンツの分散配置をうまく進める事によって通信遅延の低減を期待するアーキテクチャであるため、このようなキャッシング戦略はネットワーク全体の性能に深く関わるものである。

Routing は、Interest の転送に際していかに最善の経路を選択するか、という戦略に関する研究課題である。HCN と異なり、パケット自身が宛先ノードを指定していないことや、コンテンツを持ち **producer** となりうるノードがキャッシュの変化に伴い一定ではない可能性があることなどから、ICN 独自のルーティングアルゴリズムを導入する必要があると考えられる。ルーティングはネットワーク全体の通信性能に関わるが、特に通信の信頼性に大きな影響を与える部分である。本研究もこの分野に属する。

Security はコンテンツの信頼性や通信内容の秘匿性を保つ事に関する研究課題のグループである。HCN の場合ではコンテンツを提供するホストの正当性をデジタル署名などを通じて保証する事で、間接的にそのホストが提供するコンテンツの正当性を保証していた。一方 ICN では、**consumer** から見て **producer** がどのノードなのか担保されないことなどから、このような手法は使うことができず、取得したコンテンツ自身がその正当性や信頼性を保証できるような仕組みが求められる。

3 ICN におけるパケット転送戦略

3.1 本章の概要

本章では、ICN におけるパケット転送戦略の設計における考え方について述べていく。3.2 節では既存のパケット転送戦略を分類した上で、各手法がどのように設計されているかを見ていく。これを通じて、既存手法が抱える課題や、経路選択のために用いている評価尺度の分析を行い、提案手法への検討とする。それら既存手法、並びに 2 章で触れた ICN の動作を踏まえつつ、3.3 節では本稿で提案するパケット転送戦略が解決すべき課題を明らかにする。3.4 節では、提案手法に深く関わる多腕バンディットアルゴリズムについての紹介を行っている。これを踏まえて 3.5 節では、ルーティングにおける課題と多腕バンディット問題との類似点を指摘し、次章で扱う提案手法への導入とする。

3.2 既存のパケット転送戦略とその分類

3.2.1 分類の全体像

ICN におけるルーティング技術は様々な観点から分類されている [1]。まず、大きな分類として有線の固定ネットワーク網向けのものと無線ネットワーク向けのものとに分けることができる。さらにそれぞれについて目的に応じて細かな分類をすることができる。

文献 [1] は無線ネットワーク向けのルーティング技術に関する調査であるが、その中でもさらに

- Proactive 型
- Reactive 型
- Opportunistic 型

という分類がなされている。

Proactive 型ルーティングは、コンテンツの生成と同時にそのコンテンツの名前や位置の情報をネットワークに拡散する。これによってルーティングに必要な経路情報を確実にネットワーク上に拡散させることができる。この手法は事前にルーティングが完了していることから通信応答そのものは早くなるが、常時通信のオーバーヘッドが発生していること、特にモバイル端末間においては端末のバッテリー消費が無視できなくなることなどから特に無線環境の ICN 向けとしてはあまり研究が行われていない。

Reactive 型ルーティングは、Consumer からの Interest パケットを受信するとそれに応じてルータがなんらかの処理あるいは通信を行いルーティングをしていくような手法である。通信量と通信速度のバランスが良く多数の研究がなされている。また、このルーティング方式にはさらにルーティング時に考慮する情報の種類に応じて

Neighbor-aware 型 近隣ノードの情報を利用する

Distance-aware 型 周辺ノードとの距離を考慮する

Geo-aware 型 コンテンツやノードの位置情報を利用する

などのサブカテゴリに分類がされている。

Opportunistic 型ルーティングは移動体間の無線通信に用いられる方式で、Interest パケットを受信してもルータ

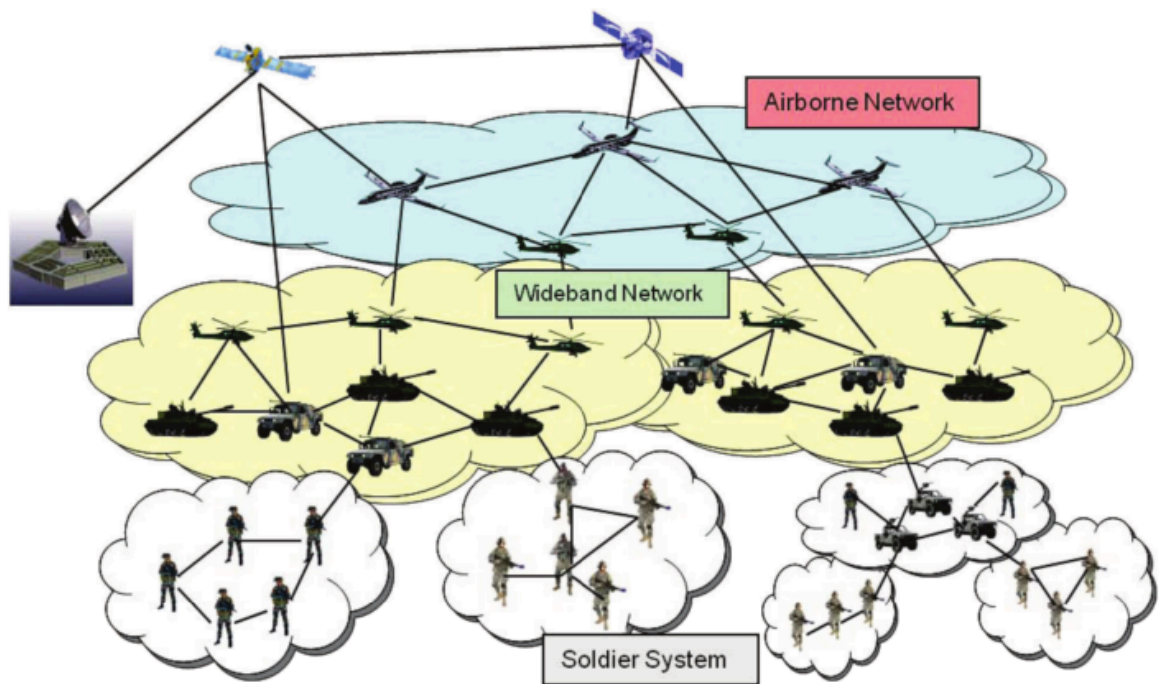


図 2. 災害現場や戦場を想定した階層型ネットワークの概念図 ([11] より引用)

自身からルーティング用の通信は行わない。ノードの移動に伴い自身の通信圏内に対象データが発見されたらそれを入手する，というような手法がここに分類される。この方法は通信のオーバーヘッドが極めて小さいが，通信速度については期待できない。

3.2.2 無線環境向けルーティング手法

無線環境向けのルーティング手法は，対象となる環境やアプリケーションを具体的に想定した上で個別に設計されたものが多い。また，ICN を適用することが有効だと考えられている多ノード間における移動体通信を想定した手法が多い。以下ではそのいくつかを紹介する。

MANET CCN

文献 [11] で提案されている MANET CCN は災害現場，あるいは戦場などの環境を想定した無線通信向けルーティングプロトコルである。NDN に近いアーキテクチャを採用しているが，ノードの持つテーブルの構造は NDN のそれとは幾分異なる。このルーティング方式では災害現場及び戦場におけるネットワーク構成を階層型と仮定した上で，その末端部分においてモバイル端末間がアドホック通信を行うことを想定し，それに適したルーティングプロトコルを設計している。例えば戦場では図 2 に示すような階層構造が想定されている。各階層に当てはまるノードの種類として，階層構造の末端部にはレスキュー隊員あるいは兵士が持つモバイル端末，中層部にはそれらを束ねる自動車等に搭載された端末，上層部には例えば本部と地方とを結ぶ衛星通信可能な端末などが当てはまる。それぞれの層を繋ぐノードはゲートウェイと呼ばれる。想定されるシナリオの特性から，MANET CCN では通信の信頼性が最優先に求められていると考えられる。

MANET CCN においては 2 種類のデータが定義される。1 つは一般的なデータ（以下 Data）であり，もう一つ

は地域及び時間に強く依存するようなデータ（以下 Temporal/Spatial Content）である。これには例えばある地域の気象情報のようなものが含まれる。MANET CCN はこの Temporal/Spatial Content について通常の Data よりも迅速に必要な地域に送信することを考慮したプロトコルとなっている。具体的には、Temporal/Spatial Content については生成と同時にコンテンツの拡散を行う。この時には端末の位置情報を利用するなどして、適切な方向に拡散させていく。これはルーティングというよりはコンテンツキャッシュの拡散であるが、結果としてルーティングに資することになる。

MANET CCN では各ノードは、ルーティングに使うために以下の 3 種類のテーブルを保持している。

Content Repository 他のノードに配信するためのコンテンツのキャッシュを保管しておくための領域である。ただし端末の種類によってこのキャッシュのサイズは異なる場合がある。例えば、末端部のモバイル端末ではこの領域のサイズは極めて小さく、受信バッファ程度のサイズしか持たないケースが想定される。NDN における CS と同じ機能を持つ。

Interest Table 自身が転送した Interest パケットの情報を保存しておくためのテーブルである。NDN における PIT と同じ機能を持つ。

Meta-Data Registry ファイルタイプ、作成日時、サイズ、作成者、説明などからなるメタデータを保管しておくテーブルである。Data パケットの名前はこれらの情報を結合することによって生成され、ここに保管されている情報は Interest パケット転送時の判断に使われる。また、登録されるデータ量が多くなるため高速で検索できるハッシュテーブルとすることが望ましい。

各ノードにおける実際のルーティングは、次のような手順で行われる。

まず、Data と Temporal/Spatial Content いずれの場合も生成時にそのコンテンツのメタデータを作成し、周囲のノードに通知する。同時に、先述の通り Temporal/Spatial Content についてはデータそのものの拡散を開始する。このデータ拡散は位置情報を利用して行くとされているが、その具体的な手法については触れられていない。通知を受信したノードは、そのメタデータを Meta-Data Registry に保管する。

データを受信したいノードは Interest パケットを周囲のノードに送信する。Interest パケットを受信したノードはまず自身の Content Repository を検索する。Interest で要求されているデータが登録されている場合はそのデータを返送することとする。データ返送については後に述べる。Content repository にない場合は Interest パケットのルーティングを行わなければならない。まず Interest Table を検索する。ここに同じ Interest の情報が格納されている場合はその Interest は破棄する。ただし、新しく受け取った Interest の送信元が元々登録されている情報と異なる場合は今回の Interest の送信元も Interest Table に追加する。最後に、Meta-Data Registry を検索する。Meta-Data Registry に登録されているのは付近のノードに存在するコンテンツのメタデータのみであるから、もし Meta-Data Registry に該当するデータが存在しない場合は Interest パケットを上位層へのゲートウェイにのみ転送する。該当するデータがヒットした場合は周囲のノード全てに対して Interest パケットをブロードキャストで転送する。Interest パケットの転送時には自身の Interest Table に転送した Interest の情報を記録する。

データの返送段階においてはパケット衝突防止のため TCP/IP のスリーウェイハンドシェイクにも似たやり方をとる。この概要を図 3 に示す。Data パケットを受信したノードは、その Data を要求する Interest をすでに自分が受信していたかどうかを Interest Table を検索することで確認する。もし、該当する情報が存在しなかった場合、受信した Data パケットはその場で破棄される。登録されていた場合はそれに従って返送先を決定するが、この時即座にデータの転送を開始するのではなく初めにリクエストパケットと呼ばれるパケットを送信する。リクエ

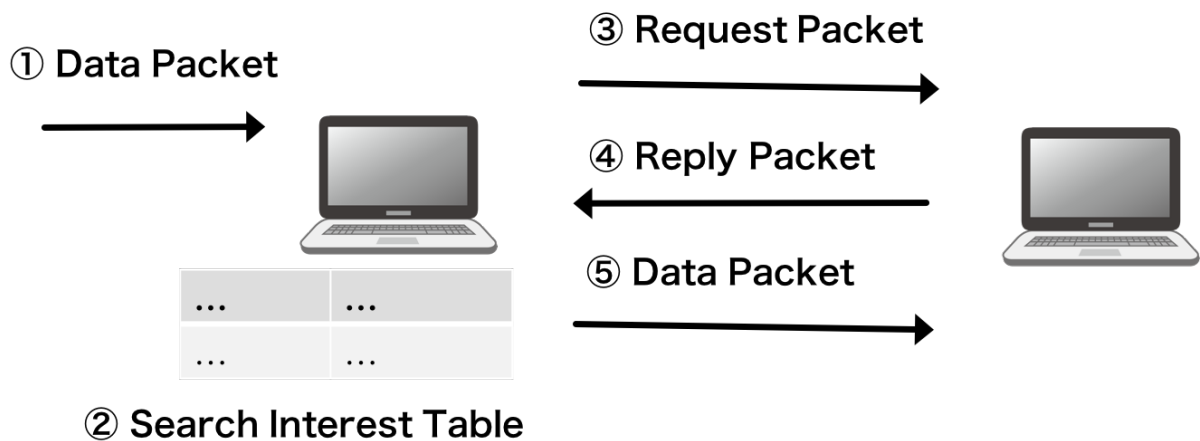


図 3. Data の転送時に行われるパケット衝突防止のための通信の模式図

ストパケットを受信したノードはデータの受信を望むならリプライパケットを返送し、それを受けて初めて Data パケットの送信が行われる。これによって複数ノードからの Data パケットの衝突を防止することができる。また、Data パケットの転送時には Content Repository に受信したデータのキャッシュを保存する。この時、Content Repository の容量次第ではデータの置き換えを行う必要があるが、これには LRU 方式あるいは LFU 方式のアルゴリズムを用いることが示唆されている。

MANET CCN は、単純なブロードキャストの繰り返しであるという点で最も単純なルーティング手法であると捉えることができる。実際に文献 [11] に示された実験結果では、極めて単純なネットワーク環境下においては、P2P 型のデータ交換プロトコルに比べて高速に全てのノードへのデータ転送を完了できるということが示されている。また、ネットワークの維持に必要なコントロールパケット（この方式で言えばメタデータの拡散などが当てはまる）の量も少なく済むことが示されている。しかしながら、単純なブロードキャストの繰り返しでは Interest 拡散のロスが非常に大きくなる。したがって、ネットワークの利用効率を考えた場合この手法は優れたやり方であるとは言えない。ネットワーク利用効率を犠牲に通信の信頼性を高めた手法であると言える。

NAVIGO

NAVIGO [8] は、NDN をベースとした自動車間通信（V-NDN）向けに設計されたルーティングアルゴリズムである。V-NDN は

- 移動するノード（自動車）と固定ノード（路側機）が存在する
- 各ノードには GPS 機器や無線 LAN ユニットなどが搭載されている

といった特徴を持つ自動車間、および自動車・クラウド間通信のためのネットワークである。ノードに GPS 機器が搭載されているという特徴を利用して、自ノードの位置情報を利用してルーティングを行う手法となっており、Geo-aware 型に分類される。

NDN ベースであるため、ノードの持つデータ構造等のアーキテクチャは NDN のものと同様である。NAVIGO は位置情報を扱うために次のような機能を備えている。

まず、NAVIGO において扱うデータは地域に依存するもの（例えば、特定の地域の渋滞情報）と依存しないも

のに分けられる。前者のようなデータについては、その地域の情報をコンテンツ名と関連づける。後者のようなデータについては、そのデータの提供者が存在する地域をコンテンツ名に関連づける。これによって、データの種類に関わらずコンテンツ名と地域情報を関連づけることができる。以下ではコンテンツ名が地域情報と関連づけられていることを前提にルーティングを行っていく。

Interest を受信したノードが、その Interest に含まれるコンテンツ名に関連づけられた位置情報について何の前提知識も持っていない場合、そのパケットはひとまず転送可能な全方向に向けて転送される。この Interest に対して、該当するデータを持つ別のノードが応答する場合そのノードは応答の Data に自身の位置情報を付加する。これによって、Data の返送経路上のノードはそのコンテンツ名に対応する位置情報を得ることができる。これ以降、同じコンテンツに対する Interest を受信した場合はこの位置情報を利用してルーティングを行うことが可能になる。位置情報を把握しているコンテンツに対するルーティングは、Link Adaptation Layer (LAL) と呼ばれる機構を介して行う。LAL は、Interest が経由してきた直前のノードの位置情報、自身の位置情報、目的地の位置情報の3つを利用する。V-NDN では直前のノードの位置情報は Interest が所持しているものとする（これは NDN の原則に従い、ノードそのものを識別する情報ではなくあくまでも位置情報であることに注意）。LAL によって「直前ノードから目的エリアまでの転送コスト」と「自ノードから目的エリアまでの転送コスト」が計算され、後者の方が前者よりも低い場合（これは自ノードの方が目的エリアに近いことを意味する）のみ Interest パケットの転送を行うように制御される。

また、NDN の設計上 FIB にはコンテンツ名と Face の対応関係を保存することしかできず、位置情報を保存しておくことができない。そこで、NAVIGO では GeoFace と呼ばれる概念を導入し、LAL が GeoFace を位置情報に解釈し直す、というステップを踏むことによってこの問題を解決している。

全体としてのルーティング制御 (Strategy) は次のようになる。

- 受信した Interest を FIB 上で検索する。マッチするエントリーがない場合ノードは探索フェイズに入り、Interest パケットは無条件でブロードキャスト転送される。
- FIB 上にマッチするエントリーがあり、対応する GeoFace が1つだけ登録されている場合は確率 p で登録されている GeoFace 情報が用いられる。確率 $1 - p$ で登録されている GeoFace は無視され、探索フェイズの時と同じ振る舞いをする。
- FIB 上にマッチするエントリーがあり、対応する GeoFace が2つ以上登録されている場合はそれらの中からラウンドロビン法によって1つの GeoFace を選択し、その情報を用いる。
- 用いる GeoFace 情報が決まったら、その情報が LAL に送られて Interest の転送を行うかどうか判断される。

なお、処理の中で確率 $1 - p$ で登録されている GeoFace を無視しているのは1つのコンテンツ送信元エリアに固執せず、複数のコンテンツ入手先を探すことでトラフィックのバランスを取ることが見込めるためとされる。

Interest の転送後、事前に定めた一定の時間 T 以内に目的の Data が受信できなかった場合、FIB 上のコンテンツ名と GeoFace の対応は削除される。これは、コンテンツ保持ノードの移動などによりコンテンツに対応付けられた位置情報が無効になったと考えられ、その経路ではもはや該当の Data は取得できないと判断されるためである。

NAVIGO は、NDN の特徴でありルーティング上の課題でもある“コンテンツ配置が変化する”という問題点、および自動車間ネットワークの特徴である“各ノードが絶えず移動を続ける”といった点に地理的情報を用いることで対処した手法であると言える。すなわち、コンテンツ配置が変化するとは言っても、地域に依存するコンテン

ツで最新のもの（例えばある地域における最新の交通情報）は該当する地域で生成、保持されている可能性が高いのであるから、該当の地域を宛先としてルーティングを行うことで目的のコンテンツが取得できる可能性が高いという考えに基づいている。一方で、これはコンテンツが地域に依存するようなアプリケーションでのみ利用できるルーティング手法である。

また、この手法は冗長な、ないしは不要な Interest の拡散を行わないことにも配慮されている点にも注目できる。ルーティング手法を検討するにあたって、Interest や Data の冗長性は信頼性を高めることにもつながるが、その一方でネットワークに対する負荷となる事は避けられない。本手法は、地理情報を用いて明らかに転送する必要のない Interest を転送しないようにすることによって、効率的に Interest の削減を行うことができています。

RUFS

RUFS [12] は NAVIGO 同様自動車間通信を目的としたルーティングアルゴリズムである。NAVIGO が NDN のアーキテクチャを完全に踏襲していたのと対照的に、RUFS は NDN の PIT を Neighbors Satisfied List (NSL) というテーブルに置き換えることで設計されている。NSL は隣接ノードのコンテンツ受信状況をまとめたものであり、隣接ノード情報をルーティングに用いていることから Neighbor-aware 型に分類される。

NSL を作成するために、各ノードはまず Recent Satisfied List (RSL) と呼ばれるテーブルを作成する。RSL には受信したコンテンツごとに受信にかかった時間とコンテンツ送信者から自分自身までのホップ数がまとめられており、加えて自身の現在の走行速度と Interest Satisfaction Rate (ISR) という値が保存される。ISR は次の式 (1) のようにして計算される。

$$ISR = \frac{\text{Number of Satisfied Contents(NSC)}}{\text{Number of Requested Contents(NRC)}} \quad (1)$$

ここで、NRC はそのノードが入手しようとしたコンテンツの数（転送した Interest の数）、NSC はそのうち実際に入手できたコンテンツの数である。RSL に含まれる各要素は、後にルーティングを行う際に各経路のスコアを計算するために用いる評価尺度となる。

各ノードは作成した RSL を隣接ノードと定期的に交換し、受信した隣接ノードの RSL を元に図 4 のように NSL を作成する。実際のデータを転送するにあたっては NSL に含まれる各要素を考慮して最適な隣接ノードの選択を行いたい。そこで、作成した NSL を元に以下に示す計算によって隣接ノード V_i に対する評価 E_i を計算する。

まず NSL から、受信しようとするコンテンツの受信実績がある隣接ノードを抽出しそれらを V_i とする。NSL が持つ各プロパティ（受信速度、ホップ数など）を P_j とおいて、ノード V_i に対するプロパティ P_j の値を a_{ij} とする。 P_j のスケールはプロパティごとに異なるため、次の式 (2) によって a_{ij} は正規化され u_{ij} となる。

$$u_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^m a_{ij}^2}} \quad (2)$$

さらに、式 (3) によって各プロパティに対する重み付けを行う。単純に行う場合、 w_j の値は一様で良い。

$$v_{ij} = w_j u_{ij} \quad (3)$$

続いてこれらの値から、各プロパティ P_j に対する I^+ 及び I^- と呼ばれるベクトルを計算する。これは各プロパティに対する最大スコア及び最少スコアの一覧であり

$$I_j^+ = \max v_{ij} \quad (4)$$

$$I_j^- = \min v_{ij} \quad (5)$$

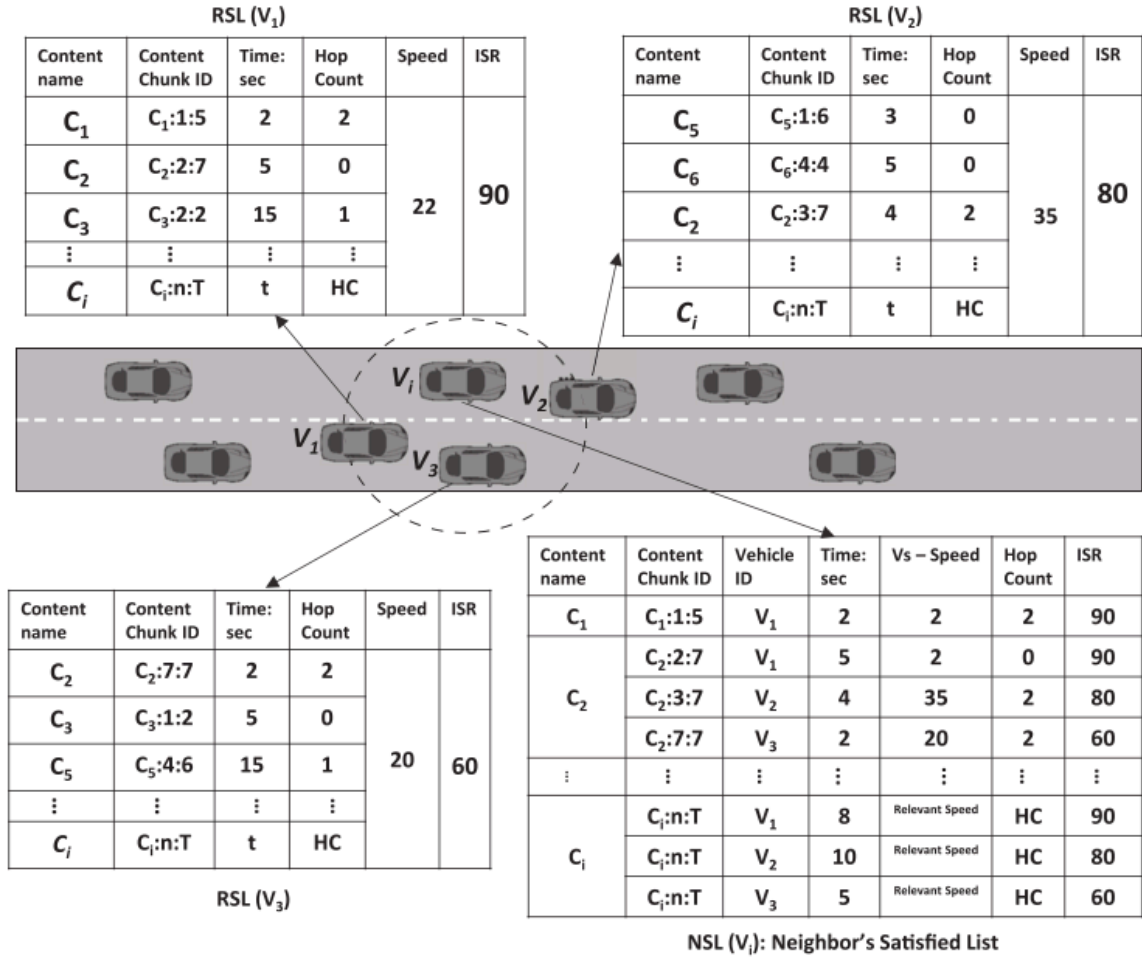


図 4. RUFs において、各ノードは隣接ノードと RSL の交換を行い、それによって NSL を作成する ([12] より引用)

によってそれぞれ定められる。これを元に、各ノードに対する評価 V_i^+ と V_i^- が次の式 (6), (7) のように決まる。

$$V_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - I_j^+)^2} \quad (6)$$

$$V_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - I_j^-)^2} \quad (7)$$

最終的な各隣接ノードの評価はこれらを合わせた

$$E_i = \frac{V_i^-}{V_i^- + V_i^+} \quad (8)$$

によって得られ、 E_i が最大のノードが Interest の転送先ノードとして選ばれる。

Interest の転送に際しては転送先ノードとして選ばれたノード情報がパケットに付加され、転送先として選ばれていないノードがこの Interest パケットを受信したとしても即座に破棄する。

RSL の定期的な交換が行われるため、この手法は動的にルーティングスコアを更新していく手法である。また、経路のスコアを求める際に ISR や走行速度といった情報を複合的に用いていることも特徴的である。

LFBL

Distance-aware 型に分類されるルーティングプロトコルである LFBL [13, 14] では、各ノードが Distance Table (DT) と呼ばれるネットワーク内のコンテンツとの距離表を維持する。また、Data のヘッダにはそのパケットが送信されてからの距離の情報（例えば経由したホップ数）が含まれており、Data パケットを受信あるいは転送したノードは自身と該当コンテンツとの距離情報を得ることができる。また、Interest は送信元ノードが持つ DT に登録された該当コンテンツまでの距離情報をヘッダに保持しており、Interest を受信したノードはこのヘッダ情報と自身の DT に登録された距離情報を比較することによって Interest の転送を行うか否かを決定することができる。また、パケット転送前に待機時間を設け、他のノードが同じ Interest の転送を行っていないかを確認してから転送することによってパケットロスを減らしている。また、LFBL をベースとして、NDN における PIT のようなテーブルを加えることによって通信効率を向上させた CHANET [15] のような手法が提案されている。

Quadrant 分割による Interest 拡散

文献 [16] では、位置情報を利用して Interest を空間的に効率よく拡散させる手法を提案している。この手法では、各ノードは自身の周囲を 90 度ずつ 4 方位に区切り、各方位に属する隣接ノードのうち最も遠いもの 4 つだけを選択して Interest を転送する。これによってネットワーク全体の Interest の増加と各 Interest の経由ホップ数の増加を抑えつつ広範囲に Interest を拡散させることが可能である。ただし、本手法も広い意味ではフラッディングを行なっていることに変わりはないため、特定の経路 / ノードに選択的に Interest を転送するような手法に比べればネットワーク全体に対する Interest の負荷は大きくなる。

3.2.3 有線環境向けルーティング手法

有線環境向けのルーティング手法は、無線環境向けのものと比べると汎用的に設計されているものが多い。以下では、それらの中から NDN 用に設計されたルーティング手法についていくつかの例を挙げる。

フラッディング

フラッディングは、最も原始的なルーティング手法の一つであると言える。原理的には有線環境に限るものではないが、無線環境で議論されることが少ないためこちらに分類した。この手法では、受信した Interest は明らかに転送されるべきではないフェースを除いて全てのフェースに転送される。明らかに転送されるべきでないフェースとは、Interest を受信したフェース (上流方向のフェースと呼ぶ)、物理的に接続されていないフェースなどのことを指す。従って、Interest は生存時間が尽きない限り最終的にはネットワーク上のあらゆるノードに到達することとなる。

この手法は、Interest が確実に要求するコンテンツデータを持つノードに到達するという確実性には優れている。また、複数のノードにコンテンツデータがあるときは全てのノードが応答しようとするため、結果として Data 応答の冗長性が確保できるとも言える。しかし、これは言い換えればネットワークに大量の無駄なパケットを流しているのと同じことであり、ネットワーク負荷という観点で言えば望ましくない。ネットワークでやり取りされるパケットの量が多くなればすぐに回線の輻輳やルータの処理能力の限界によって遅延が増大すると考えられる。

bestroute strategy

フラッディングがネットワーク負荷の面からは好ましくないため、ほとんどのルーティング手法は 1 つあるいは複数の経路を選択してその経路のみに Interest を転送するという動作をするように設計される。bestroute strategy[17] は、そのようなルーティング手法の中では最もシンプルなもので、Interest は原則として一つのフェースにのみ転送される。フェースの通信状況は RED, GREEN, YELLOW の 3 状態で管理され、FIB の各エントリ内に存在している転送先フェースリストに状態を随時保存している。この 3 状態は、次のように区分される [18]。

GREEN 現に通信が行われ、成功している

YELLOW 過去に通信が行われ成功しているが十分に長い時間使用されていない、または利用されたことがない

RED 機能していない

加えて、それぞれの転送先フェースは事前に決定されたルーティング優先度を持つ。ルーティング優先度としてはホップ数を基準に用いることが多い。これは事前に、ネットワークトポロジと各コンテンツに対するオリジンサーバは既知のものとした上で、オリジンサーバまでのホップ数として定められる。

bestroute strategy において利用されるフェースは、FIB に登録されている転送先フェースリストの中で、状態が RED でないフェースのうちルーティング優先度が最も高いものである。また、ルータ自身が能動的に Interest の再送を行うことはないが、上流方向のフェースから Interest を再度受信した場合や最初に Interest を受信したとは別のフェースから同一の Interest を再度受信した場合には再送を行う。この時のフェースの選択ルールは先に述べたものと同じであるが、その Interest に対して既に利用したフェースは候補から除かれる。従って、FIB に登録された転送先フェースリストが n 個あれば n 回までの再送に対応するということである。

全てのフェースに対して再送を行なってもなお Data が得られない場合は上流方向のフェースに対して Data が取得できないことを示す Nack パケットを Data の代わりに送信する。

この手法では、1 回の Interest 受信に対して転送先を 1 つしか用いないため、フラッディングするのに比べてネットワークを流れるパケットの数は極めて少なく済む。その一方で、回線の輻輳や他ノードの状態変化に伴う経路変更に弱いという欠点を抱えている。通信状況が RED のフェースは機能していないフェースを指すのであって、たとえ遅延が大きくなったり、輻輳によるパケットロスがあつたりしてもそれは YELLOW に分類されるに過ぎない。しかしこの手法では GREEN と YELLOW のフェースを区別していないことから、YELLOW に分類されたフェースも等しくルーティング先の候補として捉えられる。そして、ルーティング先の決定は事前にルーティングテーブルに登録された優先度に基づいて行われるため、YELLOW のフェースの優先度が高ければルータは YELLOW のフェースを利用し続けようとする。もちろん、YELLOW のフェースを利用した結果何らかの事情で Data が得られず、Interest の再送要求があつた場合には別のフェースを利用することから、結果としての Data の入手可能性は高いが、ネットワーク負荷や遅延という観点から言えばまだまだ改善の余地がある。

asf strategy

asf strategy は [19] で提案されたルーティング手法であり、プロービングを積極的に活用しているところに特徴がある。asf strategy では、ルータが Data を受信するたびに、その Data の取得にかかった RTT を計測している。計測された RTT はそれまでの計測結果と合わせた重み付き平均値 (SRTT) として、FIB に記録される。Interest 受信時の転送先決定はこの SRTT が最も小さいフェースを選択するというやり方で行われる。ただし、直近の

Interest 転送時に Data を取得できないままタイムアウトを起こしているフェースは転送先の候補からは除外される。

このような転送だけでは、SRTT が最小のフェースの RTT の情報ばかりが集まり、使用されないそれ以外のフェースの RTT 情報を集めることができない。そこで、asf strategy では定期的にプロービングを行うことによってこの問題を解決している。ルータは一定間隔ごとにプロービングのための Interest を発行し送信する。この Interest は完全に RTT の計測だけを目的としたもので、Data は転送されてくるがそれは活用されない。プロービングは一度に 1 つのフェースのみを対象として行われ、SRTT の小さいフェースほど選択される確率の高くなるような選択方法によって対象が決定される。

この手法はリアルタイムで計測された SRTT の値をルーティングに用いていること、プロービングを行う機能を持つことによって、回線輻輳や切断などのトラブル発生時に速やかに使用する経路を切り替えることができることされている。ただし、速やかにとは言ってもそれはプロービングの間隔に依存する。プロービングのための Interest とそれに対する応答の Data はネットワークにとっては consumer と producer の間での通信とは全く関係のない負荷である。加えて、プロービングは FIB のエントリの一つ一つについて行われるため、頻繁に利用されるエントリもほとんど使用されないエントリも、等しくプロービングによる負荷をネットワークにかけることになる。従って、高頻度なプロービングはネットワークへの負荷を増大させることにも繋がる。この手法によって、プロービングを適切な頻度で行うことが経路選択上有効であることが明らかにされた一方で、プロービングによるネットワークへの負荷については検討の余地がある。

3.3 ICN 上でのパケット転送戦略における課題

前節ではこれまでに提案されてきた様々なルーティング戦略について述べた。これらを踏まえた上で、本節ではルーティング手法に求められる要件やその評価尺度について明らかにしていく。

既に繰り返し述べている通り、ICN 特有の問題として、コンテンツを所持するノード (オリジンサーバおよびキャッシュルータ) が変化することがある。それゆえ、これまで通信していたノードが目的のコンテンツを破棄してしまったり、これまで全く通信していなかったノードが新たに目的のコンテンツのキャッシュを取得した、といった状況が発生し、最適な通信経路が変化することがネットワーク障害等に関わらずありうる。前者のような状況では当然通信遅延や安定性に大きな悪影響があるため、速やかに代替の通信相手を発見する必要がある。後者のような場合においても、新たに目的のコンテンツをキャッシュしたノードがこれまでの通信相手よりも低遅延で通信できる位置にあるのなら、速やかに通信相手を切り替えることが望ましい。ルータはこのような常時変化する最適経路に可能な限り追従してパケット転送先を決定すべきであり、それを実現できるルーティング戦略が求められていると指摘されている [18]。前節で挙げたルーティング戦略の中の多くは、何らかの形で経路情報を調査し、それまでに使っていた経路でデータが取得できなくなった場合には (あるいは、優秀なものであれば取得できないまでいかないうちに、遅延が増大した時点で) 使用する経路を切り替える仕組みを持っている。これはネットワークポロジの安定しない無線アドホック環境下で用いられるルーティング手法ではもちろんのこと、ICN の場合には有線環境下で用いられるルーティング手法においても重要な要素である。

前節で挙げた多くの手法は、通信遅延時間や通信の安定性などを考慮して各経路に対する評価を算出し、経路を評価していた。最もシンプルなものとしては、bestroute strategy で用いられていたような、通信が成立しているか否かだけを管理するクラス分類による評価がある。ただし、このような単純なクラス分類は通信の信頼性、安定性を管理するものではあるが、通信遅延の小ささまでは反映できない。そこで、これに加えて管理の容易なコンテン

ッ取得に要するホップ数を加えて用いるのが、最もシンプルな経路に対する評価方法であった。これをもう少し詳細に行いたい場合には、ルータが RTT を計測して用いることになる。これらに加えて、具体的な使用状況を想定して提案されたルーティング手法の中には、地理情報や自ノードの移動速度といった、対象とする環境で経路評価に有効であると考えられる尺度を用いているものも存在していた。

RTT を計測して用いるという方法は、通信遅延を小さくするという目的を直接達成しうる方法であり非常に有効である。しかし、これを利用しようとする場合、RTT の取得は実際にその経路を利用して初めて可能であることが問題となってくる。複数経路の RTT を取得するためにはそれぞれの経路を利用してみなければならず、それは最適な経路のみを利用し続けることとは両立し得ないためである。これに対処するためには、asf strategy のようにプロービングを有効活用するというやり方がある。Interest をコピーし、最適と判断した以外の経路にも RTT の取得だけを目的に転送する方法である。ただし、asf strategy の解説で触れた通り、プロービングを行う場合プロービングそれ自体はネットワークに対する負荷としても作用してしまう。経路の通信状況をリアルタイムで監視し、経路状況の変化に速やかに対応するためにはプロービングは頻繁に行われることが望ましい。一方で過剰なプロービングはネットワーク上に冗長な Interest と Data を多数もたらし、負荷の増大や (本来優先して届けられるべき) 他の Data の遅延に繋がらう。従って、ルーティングのためのプロービングを行う場合はその制御も含めてルーティング手法として考慮する必要がある。特に、ICN においては Interest に比べて Data の方がパケットサイズが遥かに大きくなるため、Data を過剰に生成しないことが重要となってくると言える。

また、単純な通信遅延の低減のみならず、場合によってはネットワークのロードバランシング機能を担うこともルーティングの役割だといえる。特に複数経路が利用可能であると判断される場合に、少しでも良いと判断できる方を使い続けるのか、通信量などに応じて複数経路を使い分けるべきなのかも検討が必要な部分である。

以上をまとめると、ルーティング戦略において考慮する必要がある要素は以下のようなものであると言える。

- コンテンツ配置の変化や障害発生時に速やかに経路を変更することが可能であるかどうか (可能であることが望ましい)
- 通信経路評価に RTT を用いるやり方は有効だが、どのようにして複数経路の RTT を取得するか
- プロービング等による、ネットワークへの負荷はどれほどになるか (小さいほど望ましい)
- 複数経路を活用したロードバランシングがルーティングの機能として実現可能かどうか (実現可能なルーティング手法であればより有用である)

なお、各ルータが経路の状況を監視して、その結果によって通信経路を切り替える手法の他に、ルータ同士が連携して経路変化に対応する手法も考えられる。例えば、自身の持つ CS の情報 (どのコンテンツを保持しているか) や FIB の情報 (どの Name に対応する経路を把握しているか) を周辺のルータと共有することで、コンテンツキャッシュの配置の変化に素早く対応したり、経路情報を素早く拡散させたりすることが可能である。一方でこれには、プロービングと同様にこれらのデータ共有がネットワークに対する負荷となる懸念や、悪意を持ったノードによる攻撃を可能にするといった懸念も存在する。ルータ間での共有ではなくコンテンツキャッシュの配置を一元管理するサーバを用意するという選択肢もあるが、コンテンツの分散配置というメリットを薄れさせる可能性がある。これらの理由から、本研究においてはルータ間での連携は行わず、1 台のルータで完結するようなルーティング手法を検討することとする。

3.4 多腕バンディット問題とその解法

3.4.1 本節の概要

本節では、一旦 ICN から離れ、多腕バンディット問題と呼ばれる種類の問題、およびそれを解決するためのアルゴリズムであり本研究の提案手法にも深く関わってくるバンディットアルゴリズムについての解説を行う。まずはじめに多腕バンディット問題の概略について、次にバンディットアルゴリズムの概略についてそれぞれ述べていく。

3.4.2 多腕バンディット問題の概説

多腕バンディット問題は“探索”と“活用”と呼ばれる2つの行動のトレードオフをモデル化した問題であるとして説明されることが多い。バンディット問題の名の通り、この問題は通常バンディットマシン (スロットマシン) を題材として説明される。多腕バンディット問題とは以下のような問題である。

多腕バンディット問題

プレイヤーの目の前に n 台のスロットマシンがある。それぞれのスロットマシンをプレイすると、プレイヤーはマシンごとに異なった何らかの確率的分布に従う報酬を得ることができる。ただし、プレイヤーはそれぞれのスロットマシンの報酬の分布に関する知識は持っておらず、それらの情報は実際にスロットマシンを回した結果として統計的にのみ得ることができる。この条件の下で、プレイヤーがスロットマシンを繰り返しプレイした時に得られる報酬を最大化するためにはどうすれば良いか。

さらに細かく、多腕バンディット問題はスロットマシンの与える報酬のモデル化のやり方によって、割引定式化、確率的定式化、敵対的定式化の3つに分類されている。これら3種の定式化手法の中でも本稿では確率的定式化について扱う。バンディット問題の確率的定式化では、スロットマシンをプレイして得られる報酬は現在のものも未来のものも全く同等の価値があるものとして扱う。従って、スロットマシンをプレイする回数が事前に想定できない場合に適した定式化である。また、この定式化では各スロットマシンの報酬は一定の確率分布に従って得られるものとし、分布は既知であるが分布の各種パラメータは未知のものであるとすることが多い。

多腕バンディット問題において課題となるのは、初めに述べたとおり“探索”と“活用”のトレードオフである。定義に記した通り、多腕バンディット問題ではプレイヤーはスロットマシンを実際にプレイすることによってのみそのスロットマシンの報酬に関する情報を得ることができる。また、そのようにして得られるのは“ある確率分布に従って無作為に取り出した1回分の報酬”であるから、得られた情報からそのスロットマシンの報酬が従う確率分布を推定するには相応の回数そのスロットマシンをプレイすることが求められる。

ここで“すでに十分な回数プレイしていて、今のところ最も良い報酬を得られると期待できるスロットマシンをプレイして報酬を得たい”という考えと“まだ十分な回数プレイしておらず、報酬の推定値がまだ信頼できないスロットマシンをプレイして情報を得たい”という考えとの間でジレンマが生じる。短期的な報酬を大きくすることを考えるだけならば前者のような行動が優先されるべきであるが、長期的に見れば後者のような行動を挟んでいくことが結果として総報酬の増加につながる可能性があるためである。前者のような行動を“活用”，後者のような行動を“探索”と呼んでいる。“活用”をすればするほど真に報酬の大きい可能性のあるスロットマシンの“探索”は行えなくなっていくというトレードオフが重要である。

なお、多腕バンディット問題には理論的な限界があることが知られている。プレイヤーが n 回目までスロットマシン

ンをプレイした時点で、 n 回のうち“実際には最適でないスロットマシン (期待報酬が最大でないスロットマシン) を選択してしまう回数”を $O(\log n)$ 回以下に抑えることが可能である。逆に、これよりも小さく抑えることを保証することはできない。このような選択を行なった場合に、 n 回目のプレイにおいて最適でないスロットマシンを選択してしまう確率は $1/n$ 程度となる。しかしながら、この理論限界を達成するアルゴリズムは常に複雑なため実用には適さないと考えられており、理論限界よりも性能は低下するがより実用的な手法が数多く提案されている。

3.4.3 バンディットアルゴリズム

以下では実際に提案され、用いられているバンディットアルゴリズムについて説明していく。多腕バンディット問題を“解く”ためのアルゴリズムが多腕バンディットアルゴリズムである。ここまで述べたとおり、多腕バンディット問題の理論限界を達成するアルゴリズムは実用には適さないことから、シンプルさと性能をトレードオフにかけける形で様々なバンディットアルゴリズムが提案されている。

ϵ -greedy 法

最も簡単なバンディットアルゴリズムの例が ϵ -greedy 法である。このアルゴリズムでは、プレイヤが探索を行うか活用を行うかはパラメータ ϵ によって決定される。 ϵ は 0 から 1 の範囲をとる実数値であり、プレイヤが活用を行う確率を表している。すなわち、プレイヤが各ラウンドに行動を決定する際には確率 ϵ で現時点での期待報酬が最も高いマシンをプレイし、確率 $1 - \epsilon$ でそれ以外のマシンからランダムで 1 台を選択してプレイする。

この手法は非常にシンプルであるが、それ故に欠点も多い。プレイ回数が増加し、各スロットマシンの報酬に関する統計データが十分に信頼の置けるものになってきたとしてもプレイヤは序盤と変わらない確率 ϵ で探索を行ってしまう。実際には統計データに信頼が置けるようになってくればそれだけ探索を減らし活用を行うべきであり、これは非効率であると言える。これを解消するために、パラメータ ϵ をプレイ回数に応じて変化させる方法が考えられている。

Softmax 法

Softmax 法では、各マシンをプレイする確率 p を以下のような式によって各マシンの期待報酬から計算する。

$$p = \frac{\exp \frac{E_i}{\tau}}{\sum_i \exp \frac{E_i}{\tau}} \quad (9)$$

ここで、 E は各マシンの期待報酬を示す。 τ は探索と活用の比率を決定するパラメータであり、 $\tau \rightarrow \infty$ の時は一様ランダムに各マシンを選択するようになり、 $\tau \rightarrow 0$ の時には期待報酬の最も高いマシンだけをプレイするようになる。この方法を用いると期待報酬がどのような値を取っていてもそれを確率に変換することができる。しかしこの手法も各マシンの期待報酬の信頼度を考慮したものとはなっていない。例えば、100 回プレイした結果期待報酬が 10 であるマシンと、1000 回プレイした結果期待報酬が 10 であるマシンは同じ確率に変換されることになってしまう。

UCB アルゴリズム

UCB アルゴリズム [10] では、それぞれのマシンの期待報酬とそれまでのプレイ回数を用いて、各マシンに以下のような UCB スコアを与える。

$$UCBScore_i = E_i(n) + \epsilon \sqrt{\frac{\log(n)}{C_i(n)}} \quad (10)$$

ここで、 E_i はその時点で推定されるマシン i の期待報酬、 n はその時点での全マシンの合計プレイ回数、 $C_i(n)$ はその時点でのマシン i のプレイ回数である。また、 ε は探索と活用の割合を決めるパラメータであり、 ε が大きいほど探索の割合が高くなる。第 1 項がベースとなる各マシンの期待報酬で、第 2 項は各マシンをプレイした回数によって判断される「各マシンの期待報酬の信頼度」に基づく補正を加える項である。この手法は収束が遅いという欠点はあるものの、最終的には理論限界とほぼ同等 (理論限界 $O(\log n)$ の係数部分が若干悪くなる) の精度が得られることが示されている。

3.5 多腕バンディット問題とパケットフォワーディングの類似点

本節では、3.3 節で述べた諸問題を解決しつつ経路選択を行うという ICN 上でのルーティングの問題と、3.4 節で述べた多腕バンディット問題との間の類似点を指摘し、多腕バンディット問題の解決に用いられるバンディットアルゴリズムをルーティングの問題に適用することについて述べていく。

遅延時間を小さくするようなルーティングを行いたい場合、RTT を計測して用いるのが望ましいことは前節で述べた。最適経路の変更に対応するためには、ルーティングに“探索”と“活用”のフェイズを持たせることが必要である。“探索”のフェイズでは、ルータはあえてその時点で最適であると考えている経路とは異なる経路にパケットを転送する。これは当然本来のルーティングの目的には反する行為であるが、これによって最適経路以外の経路についても通信にかかる時間を確認することができ、データのキャッシュによる RTT の変化が生じた場合にこれを検知することができる。一方“活用”のフェイズではルーティングの原則に従い、最適と考えられる経路にパケットを転送する。

“探索”と“活用”をバランスよく行うことができれば、最適経路以外の経路の情報も適宜取得しつつそれに基づいた最新の最適経路情報に基づいたルーティングが可能になる。この時“探索”と“活用”のバランスは非常に重要な問題であり、“探索”が多過ぎれば最適でない経路に送られるパケットの多い非効率的なルーティングとなってしまう。一方“活用”が多過ぎれば最適経路の変化に十分対応できないルーティングとなってしまう、ICN には適さないものとなってしまう。ここで述べているような、一定の確率で“探索”を行うようなルーティング手法を確率的ルーティングと呼ぶことがある。ICN ルーティングの分野ではこのような確率的ルーティング手法についても一定の研究がなされており、例えば文献 [20], [21] に示されるような手法が挙げられる。

“探索”と“活用”のフェイズをバランスよく行わなければならないという点において、ICN のルーティングと多腕バンディット問題とは共通点のある課題であると考えられる。ルーティングにおける転送先の候補は多腕バンディット問題におけるスロットマシンに対応し、各転送先の評価指標 (例えば RTT) はスロットマシンの期待報酬に対応する。経路を選択し Interest を転送することはスロットマシンを選択しプレイすることにあたる。このように対応づけると、ICN における最適な転送先の決定は多腕バンディット問題におけるスロットマシンの選択と同種の方法によって行うことができる。

一方、ICN のルーティングと多腕バンディット問題の間には無視できない相違点も存在している。例えば、バンディット問題におけるスロットマシンが常に同一の期待報酬を持つものであったのに対し各転送経路の評価はキャッシングなどによって時間的に変化していくという点が挙げられる。また、バンディット問題ではスロットマシンを回した瞬間にその行為に対する報酬が明らかになっていたが、ルーティングにおいて報酬が明らかになるのは返送されてくる Data を受け取った瞬間であり、Interest を転送するという行為を行った時点からはタイムラグが生じている。本稿では前者を「時間依存性の問題」、後者を「報酬遅延の問題」と呼ぶことにする。

これらの相違点を考えてもなおルーティングは多腕バンディット問題と同じ枠組みで捉えられると考え、実際に

バンディットアルゴリズムを ICN 上でのルーティングに活用した研究として文献 [22] がある。この研究では、代表的バンディットアルゴリズムである ϵ -greedy 法およびその改良型と UCB アルゴリズムをそのまま ICN 上でのルーティングに適用している。ルーティングに適用するにあたって何らの変更を加えていないにも関わらず、バンディットアルゴリズムを用いたルーティング手法は高い精度で最適な経路にパケットを転送することが示されている。検討された 3 つの手法のうち特に UCB アルゴリズムについては次のような式によって各経路に対する UCB スコアを定義していた。

$$\text{UCBscore}_i(t) = \overline{\text{RTT}}_i(t) - \epsilon \sqrt{\frac{\log(C_{\text{total}}(t))}{C_i(t)}} \quad (11)$$

上式によれば、時刻 t の時点での経路 i に対する UCB スコアはその経路に対する時刻 t 以前に判明している平均往復遅延時間 $\overline{\text{RTT}}_i(t)$ と、同じく時刻 t 以前に行われたパケット転送回数の合計 $C_{\text{total}}(t)$ および経路 i に対するパケット転送回数 $C_i(t)$ から計算される。元々の UCB アルゴリズムにおけるマシンの報酬を経路の RTT に、マシンをプレイするという行為をパケットを転送するという行為にそれぞれ当てはめた、単純な UCB アルゴリズムの応用である。マシンの期待報酬の場合と異なり、RTT は小さいほど良いものであるため第 2 項の符号が逆転しているとともに、経路選択の際は UCB スコアが最も小さいものを選択することになる。

しかしこの研究は、「報酬遅延の問題」に主眼を置いたものであったため、「時間依存性の問題」についての検証が行われておらず、またそれを考慮した設計ともなっていない。ICN のルーティングにおいてコンテンツのキャッシングという ICN 特有の機能が引き起こす「時間依存性の問題」を無視するべきではなく、バンディットアルゴリズムをベースとしたルーティング手法がこれに対応しうるかどうかは明らかにする必要がある。

4 提案手法

4.1 本章の概要

本章では、本研究における提案手法について述べる。4.2 節において提案手法の詳細を示している。4.3 節では、提案手法の設計において課題となった部分である、FIB の初期化についての検討を行う。

4.2 バンディットアルゴリズムを参考にしたフォワーディング戦略の提案

4.2.1 提案手法の設計思想

本研究では、NDN における転送先フェースの決定問題を多腕バンディット問題に類似した問題として捉え、多腕バンディット問題を解くためのアルゴリズムの一つである UCB アルゴリズムを用いて解決することを提案する。UCB アルゴリズムを NDN におけるルーティングに適用する場合、ルーティングにおいて考慮される要素は“RTT”と“フェースを利用した回数”となる。ルーティングは原則として RTT の小さい経路を優先的に選択するものであるが、UCB アルゴリズムの考え方により、使用頻度の少ないフェースは多いフェースに比べてルーティング優先度が高まるようにする。これはプロービングを目的とした機能である。プロービング専用の Interest を発行する代わりに普段の転送先決定の中にプロービングに相当する要素を加えることで、ネットワークに余分な負荷をかけることなくプロービングを行うことを試みている。とはいえ、プロービングを行う場合でも、あくまでも一定の RTT で通信が行えることが期待できる経路を選択するので、本来の通信に大きな遅延が出るという影響は避けられると考えられる。また、使用回数に依存してプロービングを行えば使用されない FIB エントリの情報取得のためにプロービングだけが行われるというような事態も回避できる。

本提案手法では、先行研究と異なり、RTT やフェースの利用回数といった情報には時間による重みをかけた平均値を用いる。これは、先述した時間依存性の問題に対応するためのものであり、“**直近の情報であればあるほど、現在のネットワークの通信状況と近い可能性が高く、信用できる**”という仮定に基づいている。これら RTT とフェースの利用回数に基づいて UCB スコアを算出し、それが小さいフェースを転送先として用いるようなアルゴリズムとする。時間重み付き平均値を用いることによってこれは厳密には UCB アルゴリズムとは異なるものなのだが、便宜上ここでは UCB スコアと呼ぶことにする。

本提案手法では、先行手法でも用いられているように、原則としては RTT の小さい経路を優先的に選択するルーティングを行う。また、一度の Interest 受信に対し、通常利用する転送先フェースは 1 つとし、フラッディング時のようなネットワーク負荷を軽減する。

加えて、提案手法では転送先としての使用回数が少ないフェースのスコアを小さくするようなスコア算出式を用いる。これによって、一定回数使用されなかったフェースは使用されていたフェースに比べてルーティング優先度が高まることになる。これは、プロービングを目的とした機能である。ただし、プロービング専用の Interest を発行するのではなく consumer から受信した Interest を用いてプロービングを行なっていることになる。プロービングに利用される Interest に関していえば最善のルーティングではなくなる可能性もあるが、ネットワーク全体として過剰な負荷をかけずにプロービングを行うことが可能である。加えて、使用された回数に依存してプロービングが行われることになるため、使用されない FIB エントリの RTT 情報の維持のために無用なプロービングを行うこともなく、必要な情報だけを取得することが期待される。

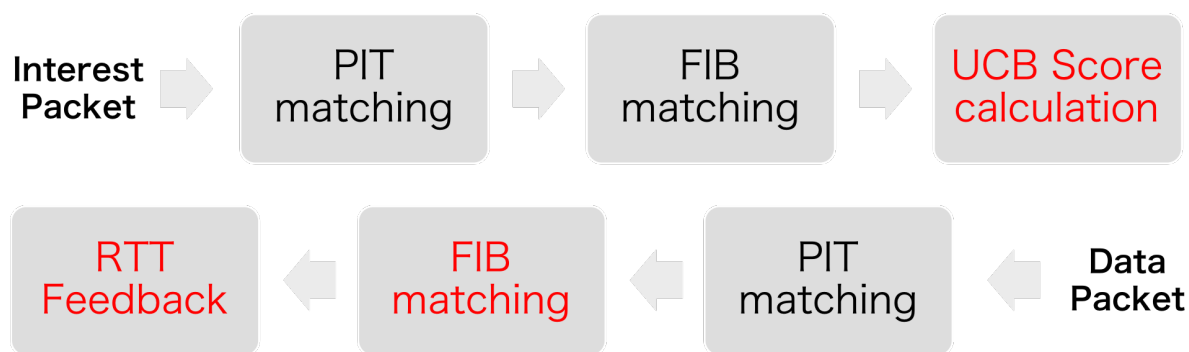


図 5. 提案手法を用いるルータの動作

Interest Prefix	NextHop List	Total TWUse Count
/prefix/sample0/	(NextHop List object)	58.18
/prefix/sample1/		

Face No.	TWRTT	TWRTT denominator	TW Use Count	Last Send Time
258	89000	27.84	27.84	20000000ns
259	89000	30.34	30.34	22000000ns

図 6. 提案手法で用いる FIB の構造の概念図. Name のプレフィックスごとにエントリが作成され, 各エントリはさらに転送先フェース候補のリストを持っている.

Interest Name	Timer	Nonce List	OutRecord	InRecord(Face)								
/prefix/sample0/	25000000ns	(64bit)	(OutRecord Object)	256,257								
/prefix/sample1/												
			<table><tr><th>OutFace</th><th>Last Nonce</th><th>Last Renewed</th></tr><tr><td>258</td><td>(64bit)</td><td>20000000ns</td></tr><tr><td>259</td><td>(64bit)</td><td>22000000ns</td></tr></table>	OutFace	Last Nonce	Last Renewed	258	(64bit)	20000000ns	259	(64bit)	22000000ns
OutFace	Last Nonce	Last Renewed										
258	(64bit)	20000000ns										
259	(64bit)	22000000ns										

図 7. 提案手法で用いる PIT の構造の概念図. 転送した Interest の Name ごとにエントリが作成され, 各エントリは転送に用いたフェースのリストを持っている.

提案手法に基づくルータの動作の概略を図.5 に示す. 図中で黒字で表現されている部分は一般にどのようなルーティング戦略を用いるルータでも行われる処理, 赤字の部分は提案手法に特有の処理である.

また, 提案手法を用いるルータでは, FIB および PIT の構成は図.6, 図.7 のようなものとする. 以下, 提案手法を用いる時のルータの動作を示す.

4.2.2 Data 受信時

まず初めに、ルータが Data を受信した場合の動作について述べる。Data 受信時、まず初めに参照されるのは PIT である。PIT に Data のコンテンツ名とマッチするエントリが存在しない場合、その Data は破棄され処理は終了する。マッチするエントリが存在する場合、エントリ内の InRecord フィールドが参照される。ここには、その Interest を受信したフェースの情報が記録されている。InRecord に記録されている全てのフェースに Data を転送し PIT エントリを削除すれば、Data 転送処理そのものは完了する。また、必要に応じて Data は CS にキャッシュされる。

続いて、提案手法のための RTT 等の情報の取得フェーズに入る。参照されるのは FIB である。Data に含まれるコンテンツ名と FIB に保存されている Interest Prefix とを比較していく。Data に対応する Interest を転送していた (PIT にマッチするエントリがあった) のであれば、ここでは必ずマッチするエントリがあるはずであり、それは Interest 転送時に用いられたエントリと同じものになるはずである。マッチしたエントリ内の NextHop List のうち、Data を受信したフェースに対応する情報を取得する。この情報に対して、RTT 等のフィードバックを行うことになる。

フィードバックされる情報はパケットの RTT に基づき、時間で重みをかけた平均値となる Time Weightd RTT (TWRTT) と呼んでいる情報である。TWRTT は、それまでに受信した全 Data から得られる RTT から以下のように求められる。

$$TWRTT_i(t) = \frac{\sum_{\tau=0}^{t=t_1} r^{t_1-\tau} RTT_i(\tau) T_i(\tau)}{\sum_{\tau=0}^{t=t_1} r^{t_1-\tau} T_i(\tau)} \quad (12)$$

ただし、 i は対象としているフェース番号であり、時刻 t 以前で最後にフェース i にパケットを転送した時刻を t_1 とする。また、 $RTT_i(\tau)$ は時刻 τ に経路 i から Data パケットを受信していた場合はそのパケットの RTT であり、 $T(\tau)$ は時刻 τ に Data パケットを受信していた場合は 1、そうでない場合は 0 を返す関数とする。さらに、 r は単位時間あたりの指数移動平均の減衰率である。これによって、直近の RTT 情報を重視するような評価指標を得ることができる。ただし、実際のルータの処理として毎回式 (12) のような計算を行うのは非常に負荷が大きい。そこで実際には以下のような漸化式に従って Data パケットを受信する毎に TWRTT を更新していくことでも式 (12) と同様の TWRTT を算出することができる。なお以下の式では見やすさを考慮して i の添え字を省略している。

$$\text{denom}_n = r^{f_{\text{data}}} \text{denom}_{n-1} + 1 \quad (13)$$

$$TWRTT_n = \frac{r^{f_{\text{data}}} \text{denom}_{n-1} \overline{TWRTT}_{n-1} + RTT}{\text{denom}_n} \quad (14)$$

式 (13), (14) の t_{data} は前回の Data パケット受信時からの経過時間であり、RTT はその時受信した Data パケットの RTT を示す。このように漸化式として逐次計算を行っていくことで、ルータの計算コストを抑えることができる。これらの計算に必要な情報は全て FIB から取得できる。

また、フェースの使用回数に関する情報も更新する。フェースの使用回数も時間重みをかけた値となっており、Time Weighted Use Count (TWC) と呼んでいる。TWC は以下のように算出する。

$$TWC_i = \sum_{\tau=0}^{t=t_1} r^{t_1-\tau} T_i(\tau) \quad (15)$$

また、漸化式として計算することも可能で、Interest パケットを転送するたびに

$$TWC_n = r^{f_{\text{interest}}} TWC_{n-1} + 1 \quad (16)$$

として求められる (TWRTT の漸化式同様 i の表記は省略した). t_{interest} は前回の Interest パケット転送時からの経過時間である.

これらの算出値で FIB エントリの情報を更新し, 最後に FIB エントリの Total TWUse Count フィールドの情報を更新する. ここに保存されている値は, そのエントリに含まれる全ての転送先フェースの TWC の合計である. これで, Data 受信時の処理は終了となる.

4.2.3 Interest 受信時

続いて Interest 受信時の動作について述べる. Interest 受信時にはまず CS でのマッチングが行われ, マッチしなかった Interest のみがルーティング対象として扱われる. 初めに利用されるのは PIT である. 受信した Interest の Interest 名とマッチするエントリがある場合受信したものと同一名の Interest をすでに一度は送信済みということになる. この場合はさらに PIT の該当エントリの Nonce フィールドを確認する. 受信した Interest の Nonce 値と同じ値が PIT にも保存されているのなら, その Interest はループしていると思われ, 破棄される. Nonce が一致しない場合はその Interest は再送されたものであるか, 別の consumer から送信されたものであるかのどちらかである. これらの場合, Interest は再送信として処理が継続される. PIT にマッチするエントリがない場合, 新たに PIT エントリを作成して処理を継続する.

続いて FIB とのマッチングが行われる. FIB に受信した Interest 名とマッチするエントリがない場合は転送先の候補を取得することができないため Interest の転送は行わず, Interest を受信したフェースに対して Nack を送信することで転送が行えなかったことを通知する. マッチするエントリがあった場合は, そのエントリに含まれる NextHop List を取得する. ここには転送先フェース番号とそれぞれに対応する TWRTT などの UCB スコア算出用のデータが記録されている. これらを用いて, 各フェースに対する UCB スコアを計算していく.

ここで, TWC は TWRTT と異なり, Interest パケットの転送時に最新の情報に更新することが可能である. よって Interest パケットの転送時には経路 i に対する UCB スコアを直前の Data パケット受信時に計算された TWRTT の値と, その時点で経路 i に Interest パケットを送ったと仮定した場合の TWC の値 (これを preTWC と呼ぶこととする) によって計算する. preTWC の採用に伴い Total TWUse Count の値も変更され, 式中では $\text{preTWC}_{\text{total}}$ として表現されている. これらは UCB スコア算出のために計算される一時的なもので, 実際の FIB の更新は Data の受信時にのみ行う.

提案手法では UCB スコアを

$$\text{UCBscore}_i(t) = \overline{\text{TWRTT}}_i(t) - \varepsilon \sqrt{\frac{\log(\text{preTWC}_{\text{total}}(t))}{\text{preTWC}_i(t)}} \quad (17)$$

のようにして計算する. 式の第 1 項は RTT を扱っており, 第 2 項がフェースの利用回数によるスコアの補正となっている. ε は第 2 項の重みを表す係数である.

NextHop List に含まれる全てのフェースに対して上記 UCB スコアを計算し, スコアが最少だったフェースを転送先として Interest を転送する. Interest が再送されてきたものである場合は, すでに使用済みのフェースは候補から除外して計算を行う.

4.3 初期化に伴う問題と解決策

本節では提案手法における, FIB の各経路スコアの初期化の手法について述べる. 経路スコアの初期化とは, FIB のエントリ作成時 (本研究においては, 後述するシミュレーションの開始時) に, 各経路のスコアをどのよう

に与えるか、という問題である。例えば *bestroute strategy* において、FIB に登録される経路スコアとはホップ数であるから、(先述の通りネットワークポロジとオリジンサーバが既知という条件のもとでは) 初期値として直ちに正確なホップ数を設定することが可能である。

提案手法では、FIB に登録される経路スコアは RTT とその経路の利用回数である。このうち、経路の利用回数の初期値は明らかに 0 であるが、RTT はホップ数とは異なりネットワークポロジが既知であったとしてもそこから直ちに与えられるものではない。そこで、何らかの形で各経路スコアの初期化を効率よく行う手法を検討する必要がある。経路スコアの初期化を行う際の状況として、大きく以下の 2 つの状況が考えられる。

- (1) ネットワークを構成する大多数の他ノードはすでに経路スコアの初期化を終えている
- (2) ネットワークを構成する大多数の他ノードも同時に経路スコアの初期化を行っている

(1) に該当する場合とは、例えば、ネットワークに新たなノードを追加した時や、何らかの理由であるノードの FIB を初期化することになった時が該当する。(2) に該当する場合とは、ネットワーク上に新たなコンテンツが誕生した場合などである。ただし、実運用においては、HCN でのデフォルトルートに当たる、“/” に対する経路情報が登録されていると考えられる (ロングストプレフィックスマッチでは少なくともこれは全ての prefix にマッチする) ので、新規コンテンツ生成時に直ちに初期化が必要になることはない。必要になれば FIB にエントリを追加することは考えられるが、全ノードが一斉にエントリを追加するのでなければ (1) に該当することになるため、ほとんどの場合これを心配する必要はないと言える。しかしながら、本研究で行うシミュレーションの初期状態は (2) に該当するため、提案手法では (2) の状態にも対応できるような初期化手法を用いることを検討した。

初期化の処理方法に関していえば、(1) の方が (2) に比べて簡単であると考えられる。(1) では、それぞれの経路に対して 1 度ずつ Interest の転送を行うことで全ての経路の情報を取得することができると予想される。この 1 度ずつの転送は、フラッディングという形では行わないことが望ましい。NDN の各ルータには、同一の宛先 Name を持つ Interest を複数受信した場合には PIT の情報を参照することでそれを検知し、冗長な Interest の転送を防ぐ機能がある。特に、個々の Interest が持つ Nonce までもが同一であった場合、それは同一の consumer が同一のタイミングで発行した Interest であることになるから、即座に Nack が返される。こうなると、目的であった RTT の取得ができない。フラッディング等の手法によって複数経路の RTT を同時に取得しようとした場合このような問題が起こりうることから、既存手法でも全経路の RTT を取得したい時にはラウンドロビン法を用いて順に取得していくものが多い。

一方、(2) においては初期化に必要な Interest の転送回数は飛躍的に増えることが予想される。各経路に Interest を転送したところで、その先の経路が未だ確定していない状況では有効な RTT を取得できないためである。これについては、予備実験中にも確認された。ここで、各経路に対して有効な RTT を取得するために必要な Interest の転送回数について、次のような考察ができる。

ネットワークポロジが単純なツリー構造であると仮定する。ツリーの根となるノードが consumer、葉となる全てのノードが producer となるものとし、ツリーを構成する全てのノードにおいて経路スコアの初期化が必要であるとすると、ここでは、ツリーの根となる consumer ノードがある一つの経路に対する有効な RTT を取得するために必要な Interest の転送回数を考える。簡単のため、ツリーの各内部ノードは各々 n 個の個ノードを持つものと仮定する。

このような仮定を置いた時、葉ノードの親であるノードは各経路に対して 1 回ずつ、合計で n 回の Interest 転送によって、自身から先の全経路に対する有効な RTT を取得することができる。更にその親であるノードは、前述

のように1つの子ノードに対しまず n 回 Interest を転送することによって、その子ノードの経路情報の初期化を済ませることができる。その上で更にもう1回 Interest を転送することで、その経路に対する有効な RTT を取得することができる。これを n 個の各経路に対して行う必要があるため、このノードが全経路に対して有効な RTT を取得するのに必要な Interest の転送回数は $n^2 + n$ 回となる。このように求めていくと、単純なツリー構造の場合でも、必要な Interest の転送回数は経由するホップ数に応じて指数オーダーで増えていくことがわかる。

実際のネットワークポロジは必ずしも階層型ではなく、メッシュ状のネットワークであることが予想されるが、その場合にはこの計算ほど容易には必要な転送回数を求めることができない。このことから、(2) のような状況の初期化において(1) 同様にラウンドロビン法を用いることは、初期化に必要な Interest 転送回数が多くなりすぎる上、未確定になることから望ましくない。一方、ツリー構造を仮定する場合にはフラッディングによる転送が有効で、必要な Interest 転送回数は経由するホップ数に比例する程度で抑えることができるが、ネットワークポロジがメッシュ状であれば、フラッディングは前述の通り冗長な Interest を生み出すことになるため有効に機能しない。(2) のような状況の初期化で全ノードがフラッディングを用いた場合、NDN の動作原理上は consumer と producer 間の最短経路では必ず Interest と Data の往復が成立するが、それ以外の経路は冗長な Interest として処理され有効な RTT が取得できない可能性が排除できない。

そこで本手法においては、前節に示した通常の経路選択手法に加えて、初期化を目的としたフラッディングを利用するための、次のような経路選択ルールを用いることとした。

- 初回転送時には全経路に対するフラッディングを行う
- Interest に対する応答として Nack を受信した経路に対しては、次に同一 Name に対する Interest 転送を行う際に **UCB スコアによって選択された経路とは別に再度転送を行う**

この方法では、初回のフラッディングによって、少なくとも最短経路の RTT を正確に取得することを期待している。これが取得できていれば、その後の通信はその経路を用いて成立させることができる。一方、それ以外の経路については初回のフラッディングだけでは取得できない可能性もある。その中でも特に Nack が返されてきたことによって RTT を取得できなかった経路のみを繰り返し調査することになる。初回のフラッディング以後、Interest が転送される経路は減っていくことが期待できる。フラッディングでは冗長な Interest となってしまう経路でも、転送経路が減ることによって、冗長な Interest として処理されることなく RTT を取得できるようになることを想定している。正確ではなくとも一度 RTT を取得してしまえば、提案手法の性質上、その後はそれらの経路にも UCB アルゴリズムに基づいて適当な頻度で Interest が転送され、RTT の修正を行っていくことが可能である。

5 提案手法の評価

5.1 本章の概要

本章では、前章で提案した手法について、実際にシミュレータ上に実装した上で、シミュレーションを通じて既存手法との比較を含めた評価を行う。シミュレーション全体の設定は 5.2 節で示す。シミュレーションは 3 種類に分類される。まず第一のシミュレーションでは、提案手法の動作を概観するとともに、パラメータが動作にどのような影響を及ぼすかを評価する。その結果および考察は 5.3 節に示す。第二のシミュレーションでは、安定した通信状況下における通信性能について、既存手法との比較を行う。その結果および考察は 5.4 節に示す。最後に、第三のシミュレーションとして、経路状況が不安定な場合を想定したシナリオにおいて、既存手法と提案手法の比較を行う。その結果および考察は 5.5 節に示す。

5.2 シミュレーション設定

5.2.1 シミュレーション環境

本研究におけるシミュレーションには、ネットワークシミュレータ ns3 の拡張として用意されている、NDN 環境でのネットワークシミュレータ NDNsim を用いる。

5.2.2 実験用ネットワークトポロジ

本研究におけるシミュレーションは、図 8 に示すネットワークトポロジを用いて行った。図中の楕円形に示したのがルータとして機能する各ノードであり、加えて consumer, producer としての役割を持たせることができる。リンクの脇に示された数字はそのリンクにおける通信遅延時間で、単位はミリ秒である。各リンクの通信帯域は 1Mbps としている。この値は、後述する過剰なトラフィックによる負荷を与える場合を除いて、シミュレーション中の通信を行うのに十分な帯域幅である。

5.2.3 consumer アプリケーション

consumer となるノードには consumer アプリケーションを搭載する。consumer アプリケーションは、事前に prefix を指定すると一定の時間間隔ごとに Interest を発行しコンテンツの要求を行う。発行される Interest の Name は事前に指定された prefix を用いて “/< prefix >/<シーケンス番号>” という形で指定される。このため、同一名称のコンテンツは interest の再送を除けば原則として一度しか要求されない。本シミュレーションにあたって、consumer アプリケーションには “/prefix/main” という prefix を設定し、Interest の要求頻度は 200 ミリ秒に 1 回とした。

consumer アプリケーションは、Interest を発行してから一定時間（再送タイムアウト時間）の間応答が得られない場合には Interest の再送を行う。本来、この再送タイムアウト時間はそれまでの通信の RTT などから算出された値を用いるのが一般的であるが、今回は簡単のため常に一定値の 1 秒とした。

本シミュレーションでは、ネットワーク上に 4 台の consumer となるノードを配置する。consumer は WASEDA, URJC, CSU, UCI とする。

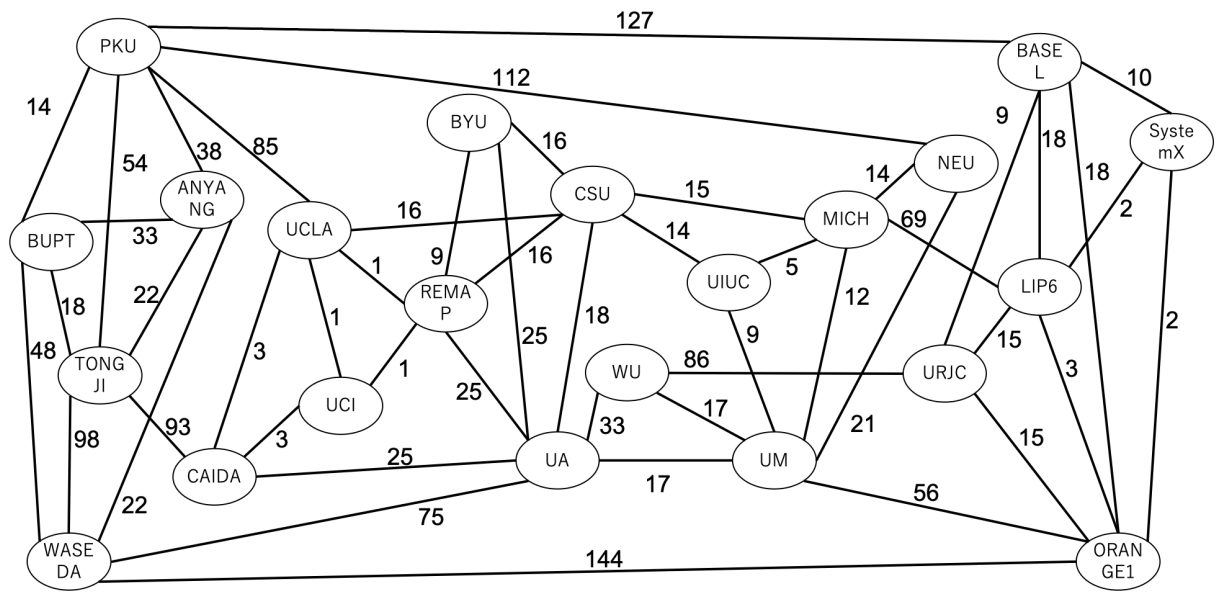


図 8. シミュレーションに用いたネットワークポロジ

5.2.4 producer アプリケーション

オリジンサーバとなるノードには producer アプリケーションを搭載する。producer アプリケーションは、事前に prefix を指定すると、該当する prefix を持つ Interest を受信した時に無条件でそれに応答する。応答する Data のサイズは一律で 1000 バイトとしている。

本シミュレーションでは、後述するようにコンテンツキャッシュを無効とした上で、ネットワーク上に 1 台のみオリジンサーバとなるノードを配置する。故に producer となるノードはネットワーク上で 1 台のみである。オリジンサーバは NEU とする。

5.2.5 ルータとしての動作

今回のシミュレーションでは、ルータに各種のルーティング戦略を搭載してその動作を見る。使用するのは提案手法の他に既存手法である bestroute strategy である。ルータには、使用するルーティング戦略とそれを適用する prefix を指定することができる。必要であれば、prefix ごとに異なるルーティング戦略を適用することも可能ということである。

また、全シミュレーションを通じてコンテンツキャッシュの機能は無効としている。

5.2.6 提案手法の設定

提案手法の詳細は 4.2 節で示した通りである。提案手法中には UCB スコアを求める際に第二項の重みを定める係数 r と、時間重みを示す係数 ε の 2 種類の係数が存在していた。このうち、 ε は $1.0 * 10^8$ と設定している。 r に関しては後述するが、検証の一つとしてこの値が提案手法の動作に及ぼす影響を調べている。また、提案手法中で RTT はミリ秒の単位で扱っている。

提案手法における FIB 情報の初期化方法は 4.3 節で示したものをを用いるため、提案手法を用いる場合は FIB の転送先候補となる隣接ノードへのフェース情報だけを初期化し、スコアの初期値は与えないこととする。

5.2.7 bestroute strategy の設定

bestroute strategy を用いる場合、FIB の初期化はシミュレーション開始前に事前に行なっている。FIB には全ての隣接ノードへのフェース情報と、その経路を利用した時のオリジンサーバへの最短ホップ数がシミュレーション開始前に事前に登録される。これによって、bestroute strategy に関しては初期化の問題はない。

5.2.8 不安定なネットワーク状況の再現

本研究における提案手法は、経路変化への対応を目的としたものである。この点についての比較を行うために、シミュレーション中で経路変化を必要とするような状況が発生させる。今回のシミュレーションでは、一部のリンクに通信が集中してしまうことを想定した処理を行うこととする。この処理を行う場合は、シミュレーション中の経過時刻で 100 秒から 140 秒の間の 40 秒間、UA-UM, PKU-NEU, CSU-UICU, CSU-MICH の 4 本のリンクを挟むノード間に 1.6Mbps 相当のトラフィックが発生させる。先述の通りリンク帯域が 1Mbps であることから、このトラフィックが発生している間はこれらのリンクを使用した通信は非常に不安定である。このトラフィックは全て “/prefix/cross/<シーケンス番号>” という Name を持つ Interest の発行と、それに対する応答という形で生成している。本文中では、このような処理を行うシミュレーション設定を“通信状況の変化を伴うシナリオ”と呼び、このような処理を行わない場合は“安定した通信状況のシナリオ”と呼ぶ。

5.2.9 測定対象

本シミュレーションでは、各 consumer ノードが発行した各 Interest に対して、応答の Data が得られるまでの時間を RTT として測定する。より具体的には、ある Name に対する Interest を最初に発行してから、実際にその Data を受け取るまでの時間を測定している。また、これとは別に、Interest 再送が発生した場合はその回数を記録している。シミュレーション時間は 200 秒間とした。従って、各 consumer は 1000Interest を送信する。

5.3 提案手法の動作の評価とパラメータの与える影響

5.3.1 本シミュレーション設定の目的と詳細

まずはじめに、提案手法におけるパラメータの検討を行った。本設定でのシミュレーションの目的は、提案手法の動作を概観するとともに、UCB スコアの算出式に用いられるパラメータ r の値が提案手法の動作に与える影響を確認することである。なお、consumer は 4 台存在していたが、今後の分析に当たっては特に WASEDA ノードにおける RTT を見ていく。これ以外の consumer についても、同様の傾向が見られる結果となる事は確認している。

本設定において、ルーティング戦略は提案手法のみを用いる。パラメータ r の値として 0.9, 0.99, 0.999 の 3 通りを用意した。それぞれの設定において通常の通信状況のシナリオと通信状況の変化を伴うシナリオのそれぞれでシミュレーションを行い、パラメータ r の値による提案手法の動作の変化を確認した。

5.3.2 シミュレーション結果および考察

まず、安定した通信状況のシナリオにおけるシミュレーションの結果を図 9～図 11 に、通信状況の変化を伴うシナリオにおけるシミュレーションの結果を図 12～図 14 にそれぞれ示す。また、各シミュレーションにおける 1000 個の Interest の平均 RTT と、再送が発生した Interest の数を表 1 に示す。

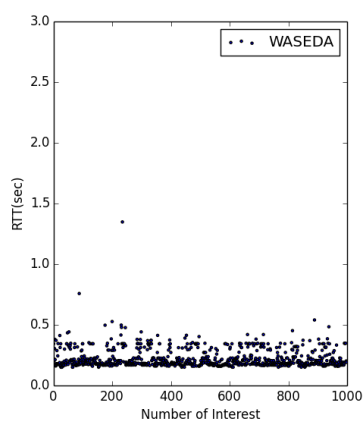


図 9. 安定した通信状況のシナリオ, 提案手法 $r = 0.999$, WASEDA-NEU 間における RTT の推移

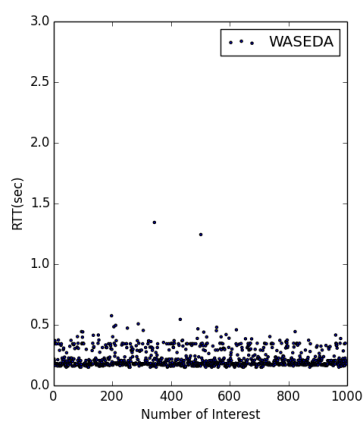


図 10. 安定した通信状況のシナリオ, 提案手法 $r = 0.99$, WASEDA-NEU における RTT の推移

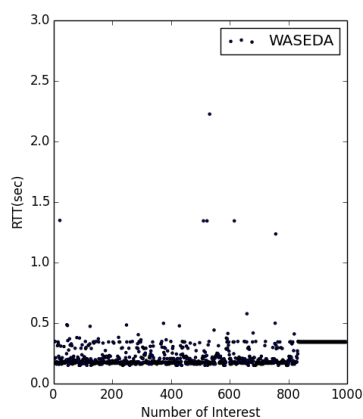


図 11. 安定した通信状況のシナリオ, 提案手法 $r = 0.9$, WASEDA-NEU における RTT の推移

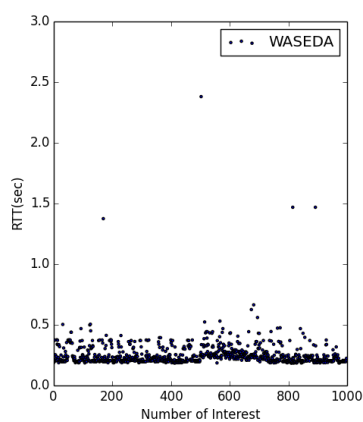


図 12. 通信状況の変化を伴うシナリオ, 提案手法 $r = 0.999$, WASEDA-NEU における RTT の推移

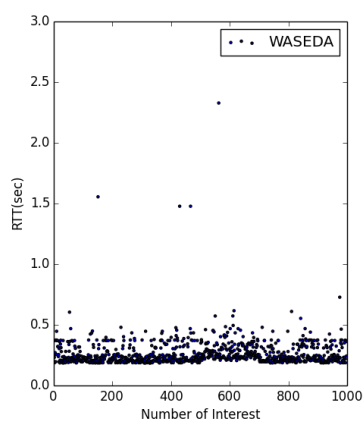


図 13. 通信状況の変化を伴うシナリオ, 提案手法 $r = 0.99$, WASEDA-NEU における RTT の推移

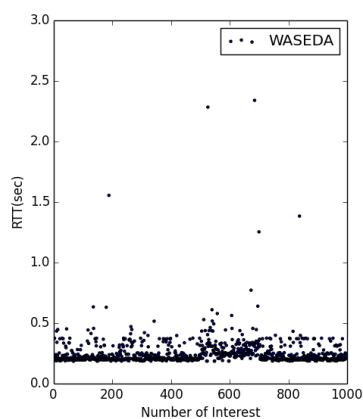


図 14. 通信状況の変化を伴うシナリオ, 提案手法 $r = 0.9$, WASEDA-NEU における RTT の推移

パラメータ	安定した通信状況のシナリオ	通信状況の変化を伴うシナリオ
$r = 0.999$	221 (1)	254 (4)
$r = 0.99$	225 (2)	264 (4)
$r = 0.9$	239 (6)	254 (5)

表 1. 提案手法の各パラメータにおける WASEDA-NEU 間の RTT および再送発生回数. RTT の単位はミリ秒でありカッコ内の値が再送回数である.

少なくとも、安定した通信状況のシナリオではどのパラメータにおいても平均 230 ミリ秒程度、通信状況の変化を伴うシナリオでも平均 260 ミリ秒程度の RTT で通信が成立していることがわかる。また、この中には“探索”を行った Interest も多数含まれているにも関わらず、応答の Data が得られず再送となってしまった Interest は全体から見れば非常に少ないこともわかる。

ネットワークトポロジから求められる WASEDA-NEU 間の最小遅延経路での遅延時間は 226 ミリ秒であるため、これらの結果からは提案手法の WASEDA-NEU 間はほぼ理想的な形で通信を行なっていることがと言える。一部の結果では、この 226 ミリ秒という値を下回る結果も見られるが、これは設定した 4 台の各 consumer から producer までの最小遅延経路が、途中から重複していることに起因すると考えられる。

例えば、WASEDA-NEU 間の最小遅延経路は WASEDA-UA-UM-NEU と結ぶ経路である。これに対して URJC-NEU 間の最小遅延経路は URJC-ORANGE1-UM-NEU と結ぶ経路である。この例では、UM-NEU の部分において 2 つのノードからの最小遅延経路が重なっている。このような場合において、WASEDA と URJC が同時に同じ Name に対する Interest を発行すると、経路の重なり始める UM においては先に到着する Interest（この場合は URJC からのものが WASEDA からのものより 21 ミリ秒早く到着する）を受け取った時点で Interest の転送を行うことになる。すると、後から到着する Interest から見れば Interest が到着する前に Interest の転送が行われたことになり、擬似的に通信時間の短縮がなされることになる。この場合、URJC からの Interest を転送してから 42 ミリ秒後には応答の Data が到着するが、それは WASEDA からの Interest が到着してからでは 21 ミリ秒後となる。すなわち、WASEDA-NEU 間の通信で見た場合にはこの差分の 21 ミリ秒が短縮されることになる。この影響により、WASEDA-NEU 間の通信時間はトポロジから求められる理想的な遅延時間を下回ることがありうるのである。

NDN 上での通信では、このように同一の Data を要求する通信の経路は重なっている方が望ましいと考えられる。同一の経路を利用している部分では幾つの Interest が届こうとも、応答の Data は 1 つにまとめることができ、ネットワーク帯域負荷の削減に寄与する上に、ここで述べたような遅延低減の効果も生みだされるからである。

この現象は、本シミュレーションのようにコンテンツキャッシュが無効となっている場合、経路の重なり始めるルータに最初の Interest が到着してから、それに対する応答の Data が到着するまでの間に同じ Name を指定する別の Interest が到着する場合に起こることになる。もしコンテンツキャッシュを有効にしている場合にはこの傾向はより顕著となることが予想される。コンテンツキャッシュが作られていれば、前文に下線部を引いて示した、この現象の恩恵を受けられる限度となる時刻が、“コンテンツキャッシュが削除されるまで”に大幅に延長されることになるからである。本結果からは、コンテンツキャッシュが無効という限られた環境下とはいえ、提案手法がこの恩恵を活用したルーティングを行なっていることが確認できる。

続いて、提案手法が通信状況の変化にも対応できていることを確認しておく。通信状況の変化を伴うシナリオにおいて、大量のトラフィックを発生させている時間帯は 100 秒～140 秒の間の 40 秒間であるが、この影響を受ける Interest は概ね 500～700 番目に発行されるものである。そこで、図 12～図 14 の該当する部分に注目する。

いずれの図においても、該当する部分でその前後に比べて僅かではあるが RTT が増加している様子が見て取れる。これが、通信状況の変化による影響である。通信状況の変化に伴い、速やかに別経路での通信に移行していると言える。これは、提案手法が事前に複数経路の RTT を探索していることによる恩恵である。また、クロストラフィックの生成が終了した後速やかに RTT が元の値に戻っている点も望ましい動作であると言える。

しかし、提案手法を用いた場合でも、特に通信状況の変化が発生した直後には、Interest の再送が起りやすい傾向にあることもわかる。Interest の再送はこの経路変化のタイミングと、シミュレーション開始直後の初期化の段階に発生しやすく、いずれも経路探索が積極的に行われるような状況である。探索を行おうとすると再送を完全に防ぐことは困難であるが、必要以上には発生しないことが望ましく、まして専用のプローブパケットを用いない提案手法では、再送は直ちに RTT の増加という形でユーザーに影響を与えてしまうことからなおさらである。この点について、本研究では課題として残すこととなるが、再送の間隔および頻度をどのように設定するのが望ましいのか、RTT の増大に与える影響とネットワーク負荷への影響の双方を考慮しつつ検討する必要があると考えられる。

最後に、パラメータ r の値による提案手法の動作の変化について見る。表 1 に示した結果は、僅かな変化ではあるが r の値が小さくなるほど平均 RTT が増加していく傾向を示している。また、図 11 においては、シミュレーションの終盤、RTT が 400 ミリ秒程度で一定になっている。これは、UCB スコアに基づく経路選択が、最適ではない経路に収束してしまったことを示唆している。

経路の収束自体は、通信状況に変化がない場合には十分起こりうることである。むしろ、通信状況に変化がないのであれば、最適な経路は常に一つに定まるので、経路は収束した方が望ましい。しかし実際にはそのようなことはなから、UCB スコアに基づき一定の頻度で探索を行うように設計したのが本提案手法である。にも関わらず最適ではない経路に収束してしまい、探索が行われないというのは提案手法がうまく機能していないことを示唆するものである。

この問題はパラメータ $r = 0.9$ の場合にのみ発生しており、他の場合には発生していないことがわかる。問題の原因は、大きすぎる時間減衰率によって、UCB スコア第二項の値が第一項の RTT の経路ごとの差異を無視できるほどに大きくなってしまっているためだと推測される。

本シミュレーションの結果を総合的に見ると、僅かな差ではあるものの、今回の設定の中ではパラメータ $r = 0.999$ の場合が最も安定し、かつ高速に通信を行なっていると言える。以後のシミュレーション設定において、既存手法と比較する場合には、この $r = 0.999$ を設定として用いることとする。

5.4 安定した通信状況のシナリオにおける評価

5.4.1 本シミュレーション設定の目的と詳細

前節では、提案手法の動作を見るためにシミュレーションを行った。本節では、まずは安定した通信状況のシナリオにおいて、提案手法と既存手法との比較を行う。本節のシミュレーションでは、ルーティング戦略として提案手法、bestroute strategy の 2 つを用いる。それぞれについて、安定した通信状況のシナリオでの試験を行い、その結果を比較していく。前節で述べたとおり、提案手法のパラメータは $r = 0.999$ を用いる。

5.4.2 シミュレーション結果

提案手法を用いた場合の結果については、結果として前節と同一のシミュレーション設定となることから、図 9 がそれとなる。bestroute strategy を用いた場合の結果については図 15 に示す。また、平均 RTT および再送回数

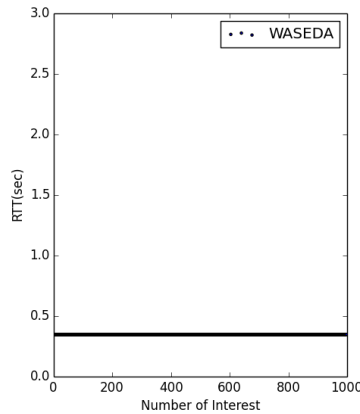


図 15. 安定した通信状況のシナリオ，bestroute strategy，WASEDA-NEU 間における RTT の推移

手法	平均 RTT[msec]	再送発生回数 [回]
提案手法	221	1
bestroute strategy	348	0

表 2. 安定した通信状況における，提案手法と既存手法における WASEDA-NEU 間の RTT および再送発生回数.

は表 2 に示す.

bestroute strategy では，WASEDA-NEU 間の平均 RTT が 348 ミリ秒と，提案手法に比べてかなり大きくなってしまっている．bestroute strategy は図 15 からわかるように，シミュレーションの開始から終了まで一貫して同じ経路を使用し続けており，RTT も一定値を維持している．この通信に利用されている経路は WASEDA-BUPT-PKU-NEU というものであった．図 8 に示した遅延時間から算出したこの経路の合計の遅延時間は往復でちょうど 348 ミリ秒となることから，実験結果にも合致している．前節で述べたとおり，これは WASEDA-NEU 間の最小遅延経路ではない．にも関わらずこの経路を選択してしまっていることが，平均 RTT に大きな差がついてしまっている要因である．加えて，前節で述べたような経路の重複による遅延時間短縮の恩恵も受けられていない．

bestroute strategy がこのような経路を選択してしまっているのは，経路のスコアリングにホップ数のみを用いているためである．提案手法が主として用いた経路 WASEDA-UA-UM-NEU も，bestroute strategy が用いた経路も，WASEDA-NEU 間のホップ数は 3 である．したがって，bestroute strategy はこれらを同一の順位に位置付けることしかできない．このような場合の動作は FIB のエントリ順，という非常に曖昧なものとなっているため，bestroute strategy は最適経路の選択に失敗してしまっているのだと言える．逆に，可能であるならば提案手法のように RTT を用いた経路評価を行うべきであることがこの結果からは明らかである．

5.5 通信状況の変化を伴うシナリオにおける評価

5.5.1 本シミュレーション設定の目的と詳細

本節では，通信状況の変化を伴うシナリオにおいて，提案手法と既存手法との比較を行う．本節のシミュレーションでは，ルーティング戦略として提案手法，bestroute strategy の 2 つを用いる．それぞれについて，通信状況の変化を伴うシナリオでのシミュレーションを行い，その結果を比較していく．先に述べたとおり，提案手法のパ

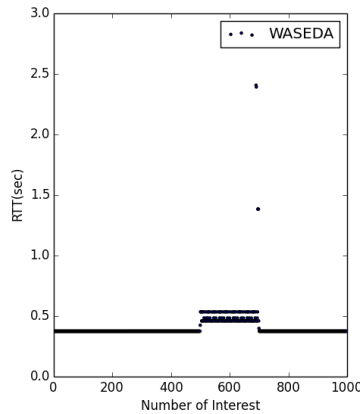


図 16. 通信状況の変化を伴うシナリオ，bestroute strategy，WASEDA-NEU 間における RTT の推移

手法	平均 RTT[msec]	再送発生回数 [回]
提案手法	234	2
bestroute strategy	1458	78

表 3. 通信状況の変化を伴う場合の，提案手法と既存手法における WASEDA-NEU 間の RTT および再送発生回数.

ラメータは $r = 0.999$ を用いる.

5.5.2 シミュレーション結果

提案手法を用いた場合の結果については，結果として 5.3 節と同一のシミュレーション設定となることから，図 12 がそれとなる．bestroute strategy を用いた場合の結果については図 16 に示す．ただし，一部後述する理由で非常に大きな RTT を取ってしまうパケットが存在していたため，図 16 では RTT が 3 秒を超えるデータは表示していない．また，平均 RTT および再送回数は表 3 に示す．

前節で指摘したのと同様，bestroute strategy の平均 RTT が非常に大きなものとなっている．これは，bestroute strategy が通信状況に応じた経路変化を実現できていないためである．bestroute strategy では経路が明らかに使用不可能にならない限り，経路の変更を行わないためである．本シミュレーションのような状況ではなく，実験結果からもこの動作は読み取れている．回線の輻輳が発生している状況下でも，bestroute strategy はその経路を使用し続けようとする．その結果として，多くのパケットが複数回の再送をおこなわなければならない状況となっていた．もっとも，bestroute strategy では再送されてきたパケットに関しては初回とは別の経路を用いて転送するので，最悪でも数度の再送を行えば通信を成立させることは可能である．本シミュレーションでは再送タイムアウトを 1 秒に固定してしまっているが，この設定次第ではもう少し改善の余地はある．

これを踏まえて，経路変化への対応という観点では提案手法は bestroute strategy に比べて非常に優れていると言える．特に，本シミュレーション設定の，帯域の輻輳に伴う RTT 増大のような状況においては，提案手法が擬似的な輻輳制御アルゴリズムとして機能する可能性や，いわゆるロードバランサとしての機能を持つことができていると言える．

5.6 総合的な評価

本節では3つのシミュレーションを通じた、提案手法の総合的な評価を行う。3つのシミュレーションの結果から提案手法は、低遅延での通信を実現すること、および通信状況の変化に合わせて通信経路を変化させること、といった観点において、既存手法である `bestroute strategy` よりも優れた性能を持つと考えられる。提案手法が経路スコアの評価に `RTT` を用いていることがこのような結果をもたらした最大の理由である。それに加えて、探索により `RTT` 情報を頻繁に更新して保持しておくことが、通信経路の変化に対応するために貢献していると言える。また、専用のプローブパケットなどによる過剰なネットワーク負荷をもたらすことなく経路探索を行えることも示された。これに伴う再送発生は限定的なものであり、本来の通信に用いられるパケットのみでも十分な経路探索を行うことが可能であると結論できる。

6 おわりに

本研究では、通信の宛先としてコンテンツを用いることを特徴とする新しいタイプのネットワークアーキテクチャである ICN に着目し、そのルーティング手法の提案を行った。

ICN にはコンテンツのキャッシュによる分散配置という特徴があったため、キャッシュ配置の変化に伴う通信経路の変化が頻繁に発生することが予想された。加えて、通信の宛先がホストで指定されないことから、障害発生時等にもコンテンツ取得先ホストを変更することで通信を継続することができる。これらを実現し、その恩恵を受けるためには、通信状況の変化に適応して使用する経路を変化させることが可能なルーティング手法が必要であった。

使用する経路を変化させるためには、事前に複数の経路の評価を行なっておく必要がある。本研究ではこのようなパケット転送経路の探索を伴うパケット転送と、多腕バンディット問題との類似性を指摘した。また、既存の ICN 向けのルーティング手法における、通信状況の変化に応じた経路選択のための経路の動的なスコアリングの必要性やそれに用いる指標、その取得方法、スコアの更新方法等を分析した。これらを踏まえて、多腕バンディット問題を解くためのアルゴリズムである UCB アルゴリズムを参考としたルーティング手法を提案した。

提案手法を実際にシミュレータ上で実装し、シミュレーションを通じた評価を行なった。評価の結果、提案手法では経路探索専用のパケットを用いなくとも、経路変化に対応するための探索を行うことができていることを明らかにした。また、探索を行うこと自体が通信遅延という点では不利であるにも関わらず、探索を行い各経路の正確な RTT を取得したことで、全体としては既存手法よりも低遅延での通信を実現できることがわかった。

提案手法には、今後の課題となるべき点も数多く存在している。コンテンツキャッシュが行われる環境下では、本稿で想定していたような状況以上に提案手法が既存手法に比べて有効に機能することが期待されるが、これを実際にシミュレーションを通じて評価することができていない。シミュレーション条件に関しては、本研究では極めて単純化されていたため、より実践的な条件に変化させていく余地は十分にある。また、提案手法のパラメータに関する分析も行ったが、これも実際の使用環境に応じて変化する可能性が高いと考えており、より具体的な検討が必要である。

謝辞

まず初めに、2年間にわたり熱心なご指導をいただいた指導教官の相田仁教授にこの場を借りて多大な感謝を申し上げます。研究内容に関する指導から、研究活動全般にわたる助言まで、大変お世話になりました。

また、秘書の元岡みさ子氏、技術専門職員の千葉新吾氏には日々の研究室生活において様々な手助けをいただきました。元助教である古宇田フミ子氏は研究内容に関して積極的な質問を頂き、それによって多くの改善点を見出すことができました。

最後に、日頃から研究室での仕事をこなしてしてくれた同期の長田知明君を始め、研究室の先輩、後輩の皆様にも感謝いたします。

参考文献

- [1] Xuan Liu, Zhuo Li, Peng Yang, and Yongqiang Dong. Information-centric mobile ad hoc networks and content routing: A survey. *Ad Hoc Networks*, Vol. 58, pp. 255–268, 2016.
- [2] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Brannard. Networking Named Content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09*, p. 1, 2009.
- [3] 朝枝 仁. 情報指向ネットワーク技術：データ通信路からコンテンツ共有. NICT オープンハウス, 2015.
- [4] Dirk Trossen, Martin J. Reed, Janne Riihijarvi, Michael Georgiades, Nikos Fotiou, and George Xylomenos. IP over ICN - The better IP? In *2015 European Conference on Networks and Communications (EuCNC)*, pp. 413–417. IEEE, jun 2015.
- [5] Ilya Moiseenko and Dave Oran. TCP/ICN: Carrying TCP over Content Centric and Named Data Networks. In *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking - ACM-ICN '16*, pp. 112–121, 2016.
- [6] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A Data-Oriented (and Beyond) Network Architecture. *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '07*, Vol. 37, No. 4, pp. 181–192, 2007.
- [7] SAIL Project, available at <http://www.sail-project.eu/> (accessed 2017-07-10).
- [8] Giulio Grassi, Davide Pesavento, Giovanni Pau, Lixia Zhang, and Serge Fdida. Navigo: Interest forwarding by geolocations in vehicular Named Data Networking. In *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–10. IEEE, jun 2015.
- [9] Zhuo Li, Kaihua Liu, Yang Zhao, and Yongtao Ma. MaPIT: An Enhanced Pending Interest Table for NDN With Mapping Bloom Filter. *IEEE Communications Letters*, Vol. 18, No. 11, pp. 1915–1918, nov 2014.
- [10] Peter Auer and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem*. *Machine Learning*, Vol. 47, pp. 235–256, 2002.
- [11] Soon Y. Oh, Davide Lau, and Mario Gerla. Content Centric Networking in tactical and emergency MANETs. In *2010 IFIP Wireless Days*, pp. 1–5. IEEE, oct 2010.
- [12] Syed Hassan Ahmed, Safdar Hussain Bouk, and Dongkyun Kim. RUFS: RobUst Forwarder Selection in Vehicular Content-Centric Networks. *IEEE Communications Letters*, Vol. 19, No. 9, pp. 1616–1619, sep 2015.
- [13] Michael Meisel and Vasileios Pappas. Listen First, Broadcast Later: Topology-Agnostic Forwarding under High Dynamics. *Annual conference of international technology alliance in network and information science*, p. 8, 2010.
- [14] Michael Meisel, Vasileios Pappas, and Lixia Zhang. Ad Hoc Networking via Named Data. In *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture - MobiArch '10*, p. 3, 2010.
- [15] Marica Amadeo and Antonella Molinaro. CHANET: A content-centric architecture for IEEE 802.11 MANETs. In *2011 International Conference on the Network of the Future*, pp. 122–127. IEEE, nov 2011.

- [16] You Lu, Biao Zhou, Lung-Chih Tung, Mario Gerla, Ashwin Ramesh, and Lohith Nagaraja. Energy-Efficient Content Retrieval in Mobile Cloud. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing - MCC '13*, p. 21, 2013.
- [17] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. *NDN*, 2012.
- [18] Cheng Yi, Alexander Afanasyev, Lan Wang, Beichuan Zhang, and Lixia Zhang. Adaptive Forwarding in Named Data Networking. *ACM SIGCOMM Computer Communication Review*, Vol. 42, No. 3, pp. 62–67, 2012.
- [19] Vince Lehman, Ashlesh Gawande, Beichuan Zhang, Lixia Zhang, Rodrigo Aldecoa, Dmitri Krioukov, and Lan Wang. An Experimental Investigation of Hyperbolic Routing with a Smart Forwarding Plane in NDN. *NDN Technical Report*, 2016.
- [20] Kai Lei, Jie Yuan, and Jiawei Wang. MDPF: An probabilistic forwarding strategy based on maximizing deviation method. *2015 IEEE Global Communications Conference, GLOBECOM 2015*, 2015.
- [21] Lirui Gong, Jiawei Wang, Xiang Zhang, and Kai Lei. Intelligent Forwarding Strategy Based on Online Machine Learning in Named Data Networking. *Trustcom/BigDataSE/ISPA*, pp. 1288–1294, 2016.
- [22] Konstantin Avrachenkov and Peter Jacko. CCN Interest Forwarding Strategy as Multi-Armed Bandit Model with Delays. *Network Games, Control and ...*, 2012.

発表文献

山内 智晴・相田 仁，バンディットアルゴリズムを用いた ICN パケット転送戦略の評価，2019 年電子情報通信学会総合大会，2019 年 3 月（発表予定）