

平成29年度 修士論文

NDNにおける拡散現象を利用した Interest パケット送出  
レート削減法

Diffusion Type Interest Packet Sending Rate Reduction in  
Named Data Network

指導教授 中里秀則 教授

2018年1月30日

早稲田大学大学院 基幹理工学研究科 情報理工・情報通信専攻

5116F023-0

海堀弘太

Kota Kaihori



# 目次

第 1 章 序論	1
第 2 章 NDN	3
2.1 NDN の概要	3
2.2 コンテンツ識別子	3
2.3 NDN のパケット	3
2.3.1 Interest パケット	3
2.3.2 Data パケット	4
2.4 ルータのデータ構造	4
2.4.1 Content Store(CS)	4
2.4.2 Forwarding Information Base(FIB)	4
2.4.3 Pending Interest Table(PIT)	4
2.5 NDN ルータ内におけるパケットの処理	4
2.5.1 Interest パケットの処理	4
2.5.2 Data パケットの処理	5
第 3 章 拡散型フロー制御	7
3.1 拡散現象	7
3.2 拡散型フロー制御の動作	8
3.3 ネットワークの境界でのパケット送出レート決定法	9
3.3.1 ネットワーク入り口の場合	10
3.3.2 ネットワークの出口の場合	10
第 4 章 提案手法	11
4.1 拡散型フロー制御の拡張	11
4.2 複数経路のための拡散型フロー制御	11
4.3 NDN への拡散型フロー制御の適応	13
第 5 章 シミュレーション	17
5.1 直線型ネットワーク	17
5.1.1 実験結果	17
5.1.2 考察	21
5.2 複数経路型ネットワーク	21
5.2.1 実験結果	21
5.2.2 考察	23
第 6 章 結論	25



## 第1章 序論

インターネットはあらゆる場面で利用され、現代社会の存在そのものを支えていると言っても過言ではないほどの重要な技術である。インターネットの基本技術である TCP/IP に代表される IP ネットワーク技術は 1970 年代から 1980 年代に作られたものに基づいており、容易に接続できること、および構築、運用の費用が小さいことから 1990 年代から普及が大きく進み、情報化社会を支える最も重要な技術のうちのひとつとなっている。インターネットが開発される際に、通信形態として電話のような特定の二者間を接続する End-to-End の形式が採用された。このようなネットワーク上の位置情報を用いた通信をロケーション指向通信と呼ぶ。しかし、近年においてインターネットの利用形態は、通信端末の普及に伴い、特定の相手と接続する通信形態よりも、情報(コンテンツ)の取得を目的とした通信形態が主なものとなっており、動画や音楽などコンテンツを取得する際に発生するトラフィックがネットワークトラフィックの多くを占めている。多くのユーザはコンテンツを要求するためにインターネットを使うが、その際、特定の相手と通信することには頓着していない。今のネットワークはあくまで End-to-End 通信の基本技術をそのままに対処療法的な機能拡張を続けているものに過ぎないため、根本的に通信速度やネットワーク利用効率を最適化しているとは言えない。したがって、将来の通信サービスを支える上で、通信形態に合わせた基本的なアーキテクチャから設計しなおされたネットワーク技術の開発が迫られている。そこでコンテンツ取得に対応したネットワークアーキテクチャとしてコンテンツ指向ネットワーク (CCN/ICN:Contents/Information Centric Networking) が提唱された。

CCN/ICN の大きな特徴として、コンテンツの名前を用いたルーティングと、ネットワーク内でのキャッシングの二つが挙げられる。CCN/ICN では、IP 通信のネットワークアドレスのようなロケーション情報ではなく、コンテンツに割り当てられた名前を用いて通信が行われる。ユーザは名前でコンテンツを要求し、ルータはその名前に基づいてルーティングを行う。また、ルータはコンテンツが運ばれてくる際、コンテンツのキャッシュを行っており、要求されたコンテンツがルータでキャッシュされていた場合は、コンテンツ源まで要求パケットをフォワーディングせず、その時点でキャッシュからコンテンツを取り出してユーザへと送り返す。

また、CCN/ICN には既存研究 [1][2][3] がいくつか存在し、それらは分散型アーキテクチャと集中型アーキテクチャの 2 つに大きく分類できる。分散型のアーキテクチャの代表例としては、本論文でも扱う、アメリカ合衆国の CCN/ICN のプロジェクトである NDN(Named Data Networking)[4] が挙げられる。NDN ではユーザ(コンシューマ)は、コンテンツの提供者(プロデューサ)の IP アドレスの代わりにコンテンツの名前を用いる。また、NDN では Interest パケットと Data パケットの 2 種類のパケットが扱われる。コンシューマはコンテンツを要求するために Interest パケットにコンテンツの名前を設定し、プロデューサに向けて送信する。プロデューサは受信したコンシューマからの Interest パケットへの応答として、Data パケットをプロデューサに向けて送り返す。Interest パケット一つにつき、一つの Data パケットが要求されることになる。また、Data パケットは Interest パケットの通った経路の逆を辿ってコンシューマまで配送される。そして、Data パケットが配送されるその際、経路上の NDN ルータはその Data パケットをキャッシュする。他のコンシューマが同じコンテンツの Interest パケットを送信したときにルータに目的のコンテンツがキャッシュされていた場合はそのルータからコンシューマへ Data パケットが送信される。一方、集中型アーキテクチャの代表例としては欧州のプロジェクトである PURSIT が挙げられる。PURSIT ではランデブーシステムと呼ばれるコンテンツ名管理システムがネットワーク全体のコンテンツ名とその存在位置を管理している。ユーザがらコンテンツ要求を受け取るとそのコンテンツの登録の有無を調べ、一致する記

録があれば、そのコンテンツの提供者（またはそれをキャッシュしているルータ）にユーザへの配信経路を通知することにより、その情報に基づいてコンテンツを送信する。本論文ではNDNを扱う。

もし、NDN ルータに Interest パケットもしくは Data パケットがその処理能力を超えて届いた場合、遅延やパケットロスが発生することが予測される。この問題を回避するため、NDN において輻輳制御メカニズムについての研究が既にいくつか行われている。NDN における輻輳制御へのアプローチは End-to-End 方式のウィンドウ制御と Hop-by-Hop 方式のレート制御の二つに大きく分類できる [5]。前者は従来の TCP によく似た仕組みである。この手法では、終端ノードであるコンシューマが Interest パケットとそれに対応する Data パケットが返ってくるまでの Round Trip Time(RTT) に基づいて、送信する Interest パケットのウィンドウサイズを決定する。これらの例として Interest Control Protocol(ICP)[6]、Information Centric Transport Protocol(ICTP)[7]、Content Centric TCP(CCTCP)[8] が挙げられる。これらの方式では RTT を輻輳検知に用いている。しかし、NDN において、中間ルータにおけるキャッシュの機能によって、RTT が大きく変わってしまうので輻輳を正しく検知するのが困難である。したがって、NDN においては End-to-End 方式のウィンドウ制御は適していないと言える [9]。一方、Hop-by-Hop 方式のレート制御ではコンシューマとルータはリンク帯域幅のようなローカルな情報に基づいて Interest の送信レートの制御を行う [10][11][12]。つまり、Hop-by-Hop 方式のレート制御は RTT に大きく依存することがないので End-to-End 方式のウィンドウ制御よりも NDN において適していると考えられる [9]。

また、現在の IP 通信においても、将来を見越して現在よりもさらに高速なネットワークを考慮した研究が行われている。ネットワークの高速化によってノードでのパケットの処理速度が高速化すると、End-to-End の遅延はノードでの処理速度よりもリンク伝搬遅延がその割合を多く占めるようになる。したがって、ネットワーク制御も高速で行われる必要がある。制御が高速で行われることで、ネットワーク状態も短い時間で大きく変化するが、光速によって決まるリンク伝搬遅延により、高速なネットワークであるほど広範囲なリアルタイムなネットワークの情報把握は困難になる。そのため、自律分散制御に基づくフロー制御機構が必要になる [13]。また高速なネットワークでは RTT の遅延もネットワークの性能に大きな影響を占めることが考えられる。そのためには End-to-End ではなく局所情報に基づいた Hop-by-Hop 方式に基づいた自律分散型のフロー制御が適している。その例として拡散型フロー制御 [14] が挙げられる。この制御はあるフローに沿った経路上のノード内パケット数を平滑化することで輻輳ノードでのバッファ溢れによるパケットロスを防ぐ効果を狙っている。

拡散型フロー制御はこれまで複数フローに対する研究 [15] は行われてきたが、複数経路に対する制御方式は提案されていなかった。また前述の通り NDN においては、Hop-by-Hop 方式のレート制御が適している。本論文では NDN に適した形にした上で複数経路においても適用できる拡散型フロー制御を提案する。更に、提案した制御方式の効果を NDN ネットワークシミュレータである ndnSIM のバージョン 1.0[16] を用いて評価する。

本論文は以下の様に構成される。第2章では NDN の動作の概要について説明する。第3章では拡散型フロー制御について説明する。第4章では拡散型フロー制御の複数経路への拡張および NDN への適応への提案を行う。第5章では提案手法のシミュレータによる評価及び考察を行う。最後に第6章で結論を述べる。

## 第2章 NDN

本論文で扱う NDN について説明する。

### 2.1 NDN の概要

NDN はデータ (コンテンツ) を要求するコンシューマ、プロデューサからの要求に対してコンテンツを提供するプロデューサ、そしてそれらの中継役となるルータによって構成されている。NDN は IP 通信における IP アドレスの代わりにコンテンツ名 (コンテンツ識別子) を使用する。また、パケットには Interest パケットと Data パケットの 2 種類が存在する。そして、IP 通信におけるインターフェイスのことを NDN では Face と呼ぶ。以下では、NDN におけるコンテンツ名、パケット、NDN ルータのデータ構造、そして NDN でのルータにおける Interest パケットと Data パケットの処理を説明する。

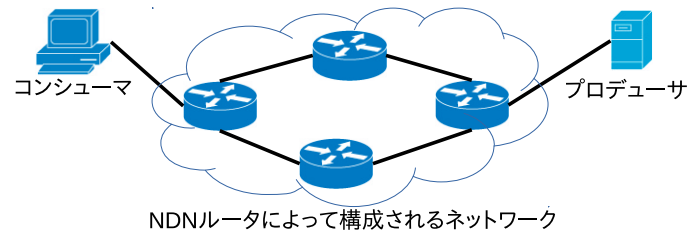


図 2.1: NDN ネットワークのモデル

### 2.2 コンテンツ識別子

コンテンツ識別子はネットワーク内で唯一の名前を与えられ、その構造は URL のような階層構造である。

### 2.3 NDN のパケット

NDN には Interest パケット、Data パケットの 2 種類のパケットが存在する。それぞれ、要求パケットと応答パケットの役割を果たしている。

#### 2.3.1 Interest パケット

Interest パケットはコンシューマがコンテンツを要求する際にネットワークへ送られるパケットである。Interest パケットを受けとったルータは、もし目的のデータを持っていればコンシューマに対して Data パケットを返送する。もし該当するコンテンツをキャッシュしていない場合は持っている可能性のあるルータがある Face へと Interest パケットを送る。Interest パケットが主に持っている情報は、コンシューマの要求するコンテンツ識別子である。

### 2.3.2 Data パケット

コンシューマの要求である Interest パケットに対して、プロデューサが応答として返すパケットが Data パケットである。Data パケットはコンシューマが要求するコンテンツそのものでもある。Data パケットが主に持っている情報はコンテンツ識別子とコンテンツそのものである。

## 2.4 ルータのデータ構造

NDN でのルータは Content Store(CS)、Forwarding Information Base(FIB)、Pending Interest Table(PIT) の 3 つの特徴的なデータ構造を持っている。

### 2.4.1 Content Store(CS)

CS にはその一つ一つのエントリにコンテンツ識別子とそのコンテンツが格納されている。つまり、CS はどのコンテンツをキャッシュをしているのかを記憶しているテーブルである。CS に格納されているコンテンツに該当する Interest パケットを受信した場合、ルータはキャッシュされているコンテンツを取り出して、返送する。

### 2.4.2 Forwarding Information Base(FIB)

FIB にはその一つ一つのエントリにコンテンツ識別子の上層部分 (Name Prefix) と Interest パケットを送信する Face が格納されている。FIB は送る Face の数は必ずしも 1 つとは限らないことを除けば、IP 通信におけるルーティングテーブルとほとんど同一のものである。ルータは FIB に従って、受信した Interest パケットを該当するコンテンツの存在する可能性のある Face へと送信する。

### 2.4.3 Pending Interest Table(PIT)

PIT にはその一つ一つのエントリにコンテンツ識別子、Interest を受信したインタフェース名が格納されている。そして、PIT は現在 Interest パケットを送信したが、該当するコンテンツがまだ返ってきていない、未解決の Interest パケットについて、その Interest パケットが到着した Face と共に記憶しているテーブルである。Data パケットはこれをルータからルータへと辿ることでコンシューマへと送り返される。Data パケットを送った際に、該当した PIT のエントリは消去される。また、PIT に一致するコンテンツ名を持つ Interest パケットを受信した場合は、その Interest パケットを転送しない。

## 2.5 NDN ルータ内におけるパケットの処理

### 2.5.1 Interest パケットの処理

コンシューマは要求するコンテンツのコンテンツ識別子を格納した Interest パケットを作成し、ネットワークへと送信する。Interest パケットを受け取った NDN ルータの Interest パケットの処理方法について図 2.2 と以下に示す。

Interest パケットがルータに届いたとき、まず、NDN ルータは Contents Store に格納されているルータ内でキャッシュしているコンテンツのコンテンツ識別子を検索する。もし同一のコンテンツ識別子が見つければ、そのコンテンツをキャッシュから取り出し、Data パケットに格納、そして Interest パケットが到着した Face に送り返す。この際に到着した Interest パケットは破棄する。キャッシュに該当するコンテンツが存在しなかった場合は PIT の確認へと処理を移る。



次にルータは PIT の中から、到着した Interest パケットのコンテンツ識別子があるか検索する。一致する PIT のエントリが存在した場合は、そのエントリの Interest パケットの到着 Face のリストに、今回の Interest パケットが到着した Face を追加し、その Interest パケットを破棄する。また、既に到着 Face のリストに Interest パケットが到着した Face があれば、PIT のエントリには何も追加せずにこの Interest パケットを破棄する。PIT のエントリにも該当するコンテンツ識別子がなかった場合は FIB の確認へと処理を移る。

最後にルータは Interest パケットのコンテンツ識別子を FIB の Name Prefix について最長一致で検索する。該当するエントリがあれば、その FIB のエントリの送出 Face へと Interest パケットを送り出し、到着した Interest パケットについての新しい PIT のエントリを作成する。FIB の Name Prefix のリストの中に該当するものがなかった場合は、この Interest パケットを破棄するか、否定応答を返して、ルータは Interest パケットに対する処理を終了する。

この動作により、最終的に Interest パケットがプロデューサに届いた場合、プロデューサは該当するコンテンツを含む Data パケットを作成して、Interest パケットが到着した Face へ送信する。

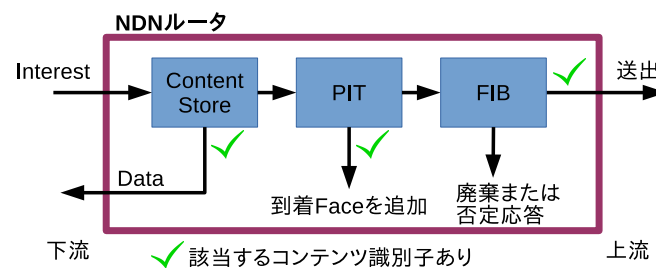


図 2.2: NDN ルータ内での Interest パケットの処理

### 2.5.2 Data パケットの処理

Data パケットを受け取った NDN ルータの Data パケットの処理方法を図 2.3 に示す。

ルータに Data パケットが到着したとき、はじめに PIT の保持するコンテンツ識別子のエントリの中に、到着した Data パケットのコンテンツ識別子と一致するものがあるか検索する。存在しなかった場合は以前に配送された Data パケットによってコンテンツ配送は既に終了したとみなされ、Data パケットは破棄される。該当する PIT エントリが見つかった場合はルータにこのコンテンツをキャッシュし、Content Store にこの Data パケットのコンテンツとコンテンツ識別子が記憶される。そして、PIT エントリに記録されている Interest パケットの到着 Face へ Data パケットを送信する。最後に、この PIT エントリは解決されたので破棄される。

以上のように、Data パケットは Interest パケットが PIT に残した到着 Face を辿っていくことで、まるで Interest パケットの足跡をたどるようにコンシューマまで配送されていく。

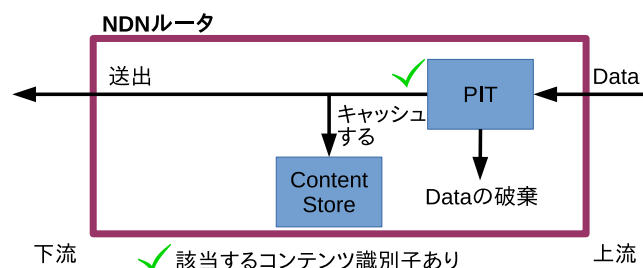


図 2.3: NDN ルータ内での Data パケットの処理



## 第3章 拡散型フロー制御

NDN はその通信特性上 End-to-End 方式によるウィンドウ制御が適していない。一方、NDN におけるパケット通信は各ルータが自律動作することによって行われるため、Hop-by-Hop 方式のルータが自律動作を行うタイプのレート制御機構との親和性が高い。

自律分散型フロー制御の既存研究の 1 つに、拡散型フロー制御が提唱されている。拡散型フロー制御とは各ノードが自身の知りうるローカルな情報のみに基づいて自律的に制御を行い、ネットワーク全体の状況を知ることはないが、その結果、ネットワーク全体の状態を望ましい方向に誘導することを狙いとした制御システムである。具体的には輻輳に陥ることを回避、および輻輳状態から回復するように動作することを狙いとしている。本章では提案手法の元になったこの拡散型フロー制御について説明する。

### 3.1 拡散現象

ローカルな情報に基づいて個々が自律動作することで、全体として秩序ある状態を生み出すことができることができる根拠として、拡散型フロー制御の概念の例となる、熱拡散現象について説明する。この概念図を図 3.1 に示す。

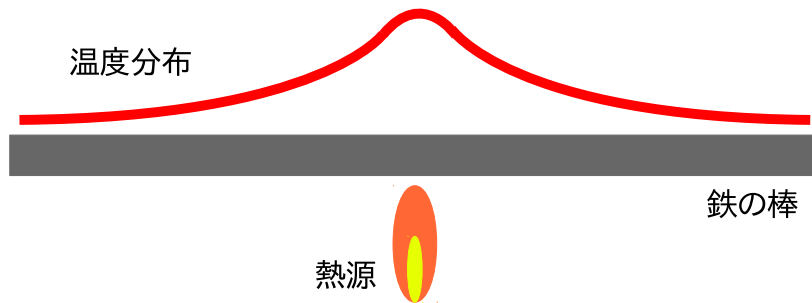


図 3.1: 熱拡散現象の概念図

冷えた鉄の棒を考え、その鉄の棒の一点を一瞬だけ熱した場合を考える。熱は時間が経過するとともに鉄の棒全体に拡散し、温度分布は平滑化していく。時刻  $t$  での鉄の棒上の微小区間  $(x, x + dx]$  にある熱量を  $n(x, t)dx$  とする。すなわち、 $n(x, t)$  は時刻  $t$  における鉄の棒の上の位置  $x$  における熱量を表している。ところで拡散現象では、鉄の棒内の熱の流れ  $J(x, t)$  が

$$J(x, t) = -\kappa \frac{\partial n(x, t)}{\partial x} \quad (3.1)$$

となる。ここで  $\kappa$  は正の定数である。熱が鉄の棒の外には逃げないとする、鉄の棒の各点での温度変化は熱の移動によるものに限られるので、連続の式が成立する。

$$\frac{\partial n(x, t)}{\partial x} = -\kappa \frac{\partial J(x, t)}{\partial x} \quad (3.2)$$

式 (3.1) を式 (3.2) に代入すると、以下の方程式が得られる。

$$\frac{\partial n(x, t)}{\partial x} = \kappa \frac{\partial^2 n(x, t)}{\partial x^2} \quad (3.3)$$

この式は一般によく知られた拡散方程式である。この方程式は初期条件として瞬間的に鉄の棒を熱して  $\delta$  関数的な温度分布を与えると、その後は時間とともに正規分布にしたがって、熱が拡散して温度が均一化していくような解を持つことが知られている。

この仕組みを、自律分散制御のシステムに対応させて考えると次のようなことが言える。

- 鉄の棒の各点は、鉄の棒全体 (システム全体) に関する情報は持っておらず、最初から決められたルールに則って隣の点と相互作用する。
- システム全体の情報を知っている存在はいない。また、離れた点からの直接的な作用もない。
- このような状況であるにもかかわらず、システム全体では偏微分方程式の解で与えられるような全体として秩序ある振る舞いをしている。

このことから、ローカルな状態の情報だけに基づいて個々が自律して動作するシステムが全体として秩序ある状態へと導くことができることを示唆している。これを根拠に拡散型フロー制御は、熱拡散におけるある一点における熱を、ノードにおけるパケット数に置き換え、ネットワーク全体でのパケット量の平滑化を目指して設計されている。

### 3.2 拡散型フロー制御の動作

拡散型フロー制御の基本的な動作について説明する。

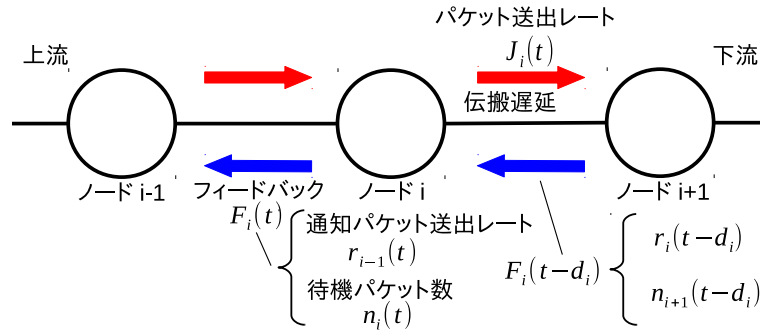


図 3.2: 拡散型フロー制御の動作のモデル

図 3.2 のような、一直線に連結した単一経路のノード  $i (i = 1, 2, \dots, N)$  を考え、ノード  $i$  と下流ノード  $i+1$  の間の伝搬速度を  $d_i$  とする。また、時刻  $t$  におけるノード  $i$  から下流ノード  $i+1$  へのパケット送出レート  $J_i(t)$  を自分自身が知りうるローカル情報に基づいて決定し、パケット転送を行う。また、上流ノード  $i-1$  に向けて、自身であるノード  $i$  の情報を通知する。このノード  $i$  からのフィードバック情報を  $F_i(t)$  とする。ノード  $i$  でのパケット送出レート  $J_i(t)$  の決定は、下流ノード  $i+1$  へのパケット送出レート  $J_i(t)$  は、

$$J_i = \max(0, \min(L_i(t), J'_i)) \quad (3.4)$$

$$J'_i(t) = r_i(t - d_i) - D_i(n_{i+1}(t - d_i) - n_i(t)) \quad (3.5)$$

によって定義する。ここで  $n_i(t)$  は時刻  $t$  でノード  $i$  に存在するパケット数、 $r_i(t-d_i)$  と  $n_{i+1}(t-d_i)$  は下流ノード  $i+1$  から、伝搬遅延  $d_i$  で伝わるフィードバック情報  $F_{i+1}(t-d_i)$  に含まれる情報である。 $r_i(t-d_i)$  と  $n_{i+1}(t-d_i)$  は時刻  $t$  において、それぞれノード  $i$  が下流ノード  $i+1$  から伝搬遅延  $d_i$  で受け取った、下流ノード  $i+1$  が指定するパケット送出レートと、下流ノード  $i+1$  に存在するパケット数である。 $L_i(t)$  はノード  $i$  から下流ノード  $i+1$  に向かうリンクの利用可能な帯域幅である。

また、ここで  $D_i$  は

$$D_i = \frac{D}{d_i} \quad (3.6)$$

と決める。ここで  $D$  は定数であり、 $D_i$  を拡散係数と呼ぶ。

また、ノード  $i$  が送出するフィードバック情報  $F_i(t)$  は

$$F_i(t) = (r_{i-1}(t), n_i(t)) \quad (3.7)$$

$$r_{i-1}(t) = J_i(t) \quad (3.8)$$

である。これを上流ノード  $i-1$  に向けて送出する。

次にこのレート決定規則の物理的な意味を説明する。 $J_i(t)$  によるネットワーク全体の性能を調べるため、ノード番号  $i$  を  $x$  とし、連続極限をとって  $d_i \rightarrow 0$  とする。このとき、 $J' \rightarrow J'(x, t)$ ,  $r_i(t-d_i) \rightarrow r(x, t)$ ,  $n_i(t) \rightarrow n(x, t)$  とすると式 (3.5) は以下の形で書くことができる。

$$J'(x, t) = r(x, t) - \kappa \frac{\partial n(x, t)}{\partial x} \quad (3.9)$$

ここで  $\kappa > 0$  は離散のときの  $D$  に対応する定数である。ところで、一次元における連続の式は

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0 \quad (3.10)$$

で、表される。 $x$  は流体の座標、 $\rho$  は  $x$  での流体の密度、 $u$  は  $x$  での流体の流速、 $\rho u$  は  $x$  での流体の流出量を意味する。これらを今回のそれぞれのパラメータに対応させると

$$\frac{\partial n(x, t)}{\partial t} + \frac{\partial J'(x, t)}{\partial x} = 0 \quad (3.11)$$

と書くことができ、この式 (3.11) を式 (3.9) を用いると、パケット密度  $n(x, t)$  の時間発展は

$$\frac{\partial n(x, t)}{\partial t} = -\frac{\partial r(x, t)}{\partial t} + \kappa \frac{\partial^2 n(x, t)}{\partial x^2} \quad (3.12)$$

と書くことができる。ここで一般に物理量  $C$  の  $(x, t)$  で表される 1 次元空間の拡散方程式は拡散係数  $\kappa$  によって

$$\frac{\partial C}{\partial t} = \kappa \frac{\partial^2 C}{\partial x^2} \quad (3.13)$$

と表される。この式 (3.13) と式 (3.12) と比較することにより、式 (3.12) が拡散方程式と同じ形をしていることが分かる。

### 3.3 ネットワークの境界でのパケット送出レート決定法

ネットワークの境界、すなわちネットワークの入り口と出口では拡散酸型フロー制御を備えた上流ノードと下流ノードが存在しない。したがって式 (3.5) で計算することができない。したがって、別にパケット送出レート決定法を決める必要がある [17]。

### 3.3.1 ネットワーク入り口の場合

ネットワークのパケット送出レートは、ノード  $i = 1$  からのフィードバック情報  $F_1(t)$  によって届けられるパケット送出レート  $r_0(t - d_0)$  によって制限する。ノード  $i = 1$  からフィードバック情報として送信される  $r_0(t)$  は、ネットワークの外のノード及びエンドホストのバッファに存在するパケット数を 0 と仮定して式 (3.5) によってパケット送出レート  $J'_0(t)$  を計算した結果を通知することで、ネットワーク外からのパケット送出レートとネットワーク内の拡散型フロー制御のパケット送出レートとの乖離を小さくすることができる。すなわちノード  $i = 1$  で計算する情報  $r_0(t)$  は式 (3.8) の代わりに

$$r_0(t) \triangleq J'_0(t + d_0) = J_i(t) - D_0 n_1(t) \quad (3.14)$$

とすることで、ノード  $i = 1$  から取得できる情報のみで、 $r_0(t)$  を決定することができる。

### 3.3.2 ネットワークの出口の場合

ネットワークの外へパケットが転送されるノードでのパケット送出レートを決める上で、ネットワーク外のノードまたはホストは拡散型フロー制御のシステムをそなえてないためフィードバック情報が存在しない。そのため、ネットワーク内のノード数を  $N$  個としたとき、終端ノードであるノード  $i = N$  のパケット送出レートはリンクの利用可能な帯域幅そのものとする。すなわち、終端ノードのパケット送出レート  $J_N(t)$  は

$$J_N(t) = L_N(t) \quad (3.15)$$

となる。

## 第4章 提案手法

### 4.1 拡散型フロー制御の拡張

拡散型フロー制御の既存研究は基本的に一次元の直線経路型のネットワークモデルについての評価がされているものが主である。実際のネットワークの形により近い、枝分かれしたネットワークに拡散型フロー制御を採用することを考えた際に、一次元の直線経路との状況の違いが生じることから新たに、パケット送出レートの決定法とフィードバック情報の2つを改良する必要がある。

前者の理由は、一次元の直線経路では一つのノードに対して下流ノードから送られてくるフィードバック情報は一つであったのに対して、複数経路では一つのノードに複数の下流ノードからのフィードバック情報が送られてくる状況が生じるので、複数のフィードバックを受け取った際にそれぞれのフィードバック情報に応じた、各経路のパケット送出レートを決定する方法が必要であるからである。

後者の理由は、一次元の直線経路では一つのノードに対して上流ノードに送るフィードバック情報は一つであったのに対して、複数経路では一つのノードから複数の上流ノードへそれぞれに適したフィードバック情報を送る必要があるからである。

これらを改良するために必要な情報を示す。拡散型フロー制御では、送信先の経路が混雑しているまたはリンク帯域幅が小さいなどの理由により混雑しているノード内に存在するパケット(待機パケット)数が増えることを前提としている。これは到着パケット数が送信パケット数を超えるときを考えれば明らかである。そして、複数経路でも同様に混雑している経路に送られる待機パケット数は増加するはずである。ところで輻輳制御の観点から、混雑している経路に送る待機パケット数が減るような機構が適しているはずである。したがって、直線経路の設計では待機パケットの送信先経路を考慮する必要がなかったが、複数経路の拡散型フロー制御では送信経路ごとの待機パケット数を知る必要がある。同様に、ノードにおける各送信経路のパケット送出レートも全て知る必要がある。

複数のフィードバックを生成するにあたって、下流ノードに応じたフィードバックの生成規則が必要である。前章で説明したようにフィードバック情報を生成するときに搭載するデータは、そのルータでのパケット送出レートと待機パケット数である。上流ノードから送られてくるパケットは直線経路であれば、パケットを受け取ったノードから送られる経路は(送り返すことを考えなければ)ただ一つに決まる。しかし、複数経路である場合は受け取ったパケットの送信経路が常に同じとは限らない。パケットが到着したインターフェイスが同じでも送信先の経路は違う場合が生じる。輻輳制御の観点から、待機パケット数が増加しているときには、混雑要因となるパケットを多く送っている経路からのパケット流入量をより強く制限するような機構が適していると考えられる。そのためには、各経路からきたパケットの送信先も知る必要がある。

### 4.2 複数経路のための拡散型フロー制御

4.1によって明らかになった必要な情報を用いて、パケット送出レートの決定法、およびそれに伴うフィードバック情報の決定法を変更し、拡散型フロー制御を拡張する。

図4.1のような、ノードに複数の上流ノードと下流ノードが接続しているネットワークモデルを考える。なお、上流ノードと下流ノードには、それぞれ一つずつノードが接続していることとする。ここで、ノード  $(i, 1)(i = 1, 2, \dots, N)$  について考える。ノード  $(i, 1)$  と下流ノード  $(i+1, q)(q = 1, 2, \dots, Q)$

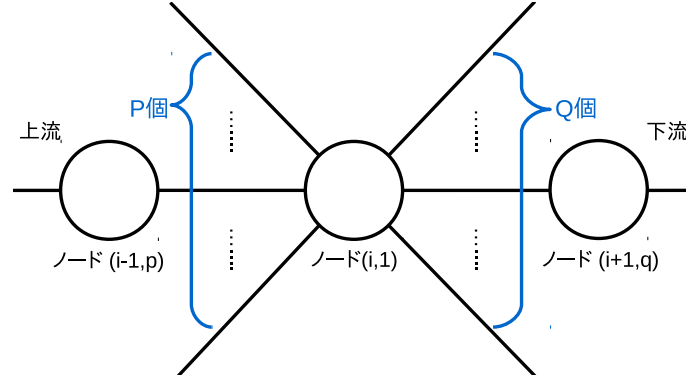


図 4.1: 複数経路における拡散型フロー制御の動作のモデル

との間の伝搬遅延  $d_i^q$  をとする。また、ノード  $(i,1)$  は時刻  $t$  におけるノード  $(i,1)$  から下流ノード  $(i+1,q)$  へのパケット送出レート  $J_i^q(t)$  をノード  $(i,1)$  が知りうる局所情報に基づいて決定し、パケット転送を行う。また、ノード  $(i,1)$  は上流ノード  $(i-1,p)$  ( $p = 1, 2, \dots, P$ ) に向けて自身の情報をフィードバックする。このノード  $(i,1)$  からのフィードバック情報を  $F_i^p(t)$  とする。パケット送出レート  $J_i^q(t)$  の決定は下流ノード  $(i+1,q)$  からのフィードバック情報  $F_{i+1}^q(t - d_i^q)$  が到着する度に行う。

時刻  $t$ 、ノード  $(i,1)$  における、ノード  $(i,1)$  から下流ノード  $(i+1,q)$  へのパケット送出レート  $J_i^q(t)$  は、

$$J_i^q = \max(0, \min(L_i^q(t), J_i^{\prime q}(t))) \quad (4.1)$$

$$J_i^{\prime q}(t) = r_i^q(t - d_i^q) - D_i^q (n_{i+1}^q(t - d_i^q) - n_i^q(t)) \quad (4.2)$$

と定義する。ここで式 (4.2) において、 $n_i^q$  は時刻  $t$  でノード  $(i,1)$  の待機パケット数のうち、下流ノード  $(i+1,q)$  へ送られる待機パケット数である。また  $r_i^q(t - d_i^q)$  と  $n_{i+1}^q(t - d_i^q)$  は下流ノード  $(i+1,q)$  から伝搬遅延  $d_i^q$  で伝わるフィードバック情報  $F_{i+1}^q(t - d_i^q)$  に含まれる情報であり、 $r_i^q(t - d_i^q)$  は下流ノード  $(i+1,q)$  が指定するパケット送出レート、 $n_{i+1}^q(t - d_i^q)$  は時刻  $t$  においてノード  $(i,1)$  が知ることができる、下流ノード  $(i+1,q)$  の待機パケットを意味する。 $L_i^q(t)$  はノード  $(i,1)$  から下流ノード  $(i+1,q)$  に向かうリンクの利用可能な帯域幅である。また、 $D_i^q$  は

$$D_i^q = \frac{D}{d_i^q} \quad (4.3)$$

とする。 $D$  は定数である。 $D_i^q$  は拡散係数と呼ぶ。また、ノード  $(i,1)$  が上流ノード  $(i-1,p)$  へ送出するフィードバック情報  $F_i^p(t)$  は

$$F_i^p(t) = (r_{i-1}^p(t), n_i^1(t)) \quad (4.4)$$

ここで上流ノード  $(i-1,p)$  に対する指定レート  $r_{i-1}^p(t)$  の決定法について説明する。フィードバック情報を送った後、再びフィードバック情報を送るまでの間に上流ノード  $(i-1,p)$  からノード  $(i,1)$  へ送られてきたパケットの総数を  $\Omega_i^p$  とする。 $\Omega_i^p$  のうち、下流ノード  $(i+1,q)$  へ転送されるパケットの数を  $\omega_i^p(q)$  とすると、 $\Omega_i^p$  は、

$$\Omega_i^p = \sum_{q=1}^Q \omega_i^p(q) \quad (4.5)$$



と表すことができる。各上流ノード  $(i-1, p)$  について転送されるパケットがどの下流ノードへいくつ転送されるかをこの  $\omega_i^p(q)$  で、上流ノード  $(i-1, p)$  へ送るフィードバック情報として送るパケット送出レート  $r_{i-1}^p(t)$  と待機パケット数  $n_i^1(t)$  へ重み付けをすることで、各上流ノード  $(i-1, p)$  からの転送パケットおよび送出レートが各下流ノード  $(i+1, q)$  への待機パケット数にどの程度影響を与えているのかを知ることができる。したがって、 $r_{i-1}^p(t)$  と  $n_i^1(t)$  は  $q = 1, 2, \dots, Q$  において、以下のように決めることができる。

$$r_{i-1}^p(t) = \begin{cases} \sum_{q=1}^Q \left( \frac{\omega_i^p(q)}{\Omega_i^p} J_i^q(t) \right) & (\Omega_i^p > 0) \\ \max(J_i^q(t)) & (\Omega_i^p = 0) \end{cases} \quad (4.6)$$

$$n_i^1(t) = \begin{cases} \sum_{q=1}^Q \left( \frac{\omega_i^p(q)}{\Omega_i^p} n_i^q(t) \right) & (\Omega_i^p > 0) \\ 0 & (\Omega_i^p = 0) \end{cases} \quad (4.7)$$

これらを保持するフィードバック情報  $F_i^P(t)$  を  $P$  個の各上流ノード  $(i-1, p)(p = 1, 2, \dots, P)$  へ向けて送出する。

### 4.3 NDN への拡散型フロー制御の適応

複数経路における拡散型フロー制御方式を実現するために必要な局所情報は 4.2 で示した。そしてこの方式を NDN の Interest パケットのパケット送出レートに適応する。今回、要求パケットである Interest パケットの送出レートを制御することで間接的に応答パケットである Data パケットの流量を制御することを目指す。NDN において拡散型フロー制御方式を実現するために必要な局所情報は、各 Face からの Interest パケット送出レート、各 Face における転送待機中の Interest パケット数、そして重み付けに用いる Interest パケット数の情報である。

各 Face からの Interest パケット送出レートはルータで二種類のパケットを区別していることから、転送時に計測することによって取得は容易であると考えられる。

各 Face における転送待機中の Interest パケット数は、提案手法では代わりに下流ノードとの PIT サイズ差を用いる。ここで、PIT サイズとは、ある NDN ルータ内の PIT のエントリ数を意味する。そしてこれに関して PIT の改良が必要になるので、先に説明をする。

通常の NDN ルータの PIT は 2.4.3 で説明したように、複数個の PIT エントリで構成されており、一つ一つの PIT エントリにはコンテンツ識別子、Interest を受信した Face 名 (Interest 到着 Face) が保持されている。この PIT エントリが保持する情報に新たに Interest を送信した Face 名 (Interest 送出 Face) を追加する。そして、PIT エントリ内の Interest 到着 Face の中にある特定の Face に関する PIT エントリの集合をある Face に関する InPIT とし、そのエントリ数を InPIT サイズとする。同様に PIT エントリ内の Interest 送出 Face の中にある特定の Face に関する PIT エントリの集合をある Face に関する OutPIT とし、そのエントリ数を OutPIT サイズとする。そして今回、ある特定の Face についての OutPIT サイズとその Face と接続している下流ノードの InPIT サイズ差をある特定の Face の待機 Interest パケット数の差の代わりとして用いる。ここで、ネットワークの途中でボトルネックとなる帯域幅の小さいリンクがある直線上のネットワークを考える。この状況で待機パケットに関して考えると図 4.2 のようにボトルネックに接続しているノードに特に多く滞留することが予想できる。また PIT についても考えると図 4.3 のようにコンシューマからボトルネックに接続しているノードまでのノードの PIT サイズが大きくなることが予想できる。ここで先ほどの OutPIT サイズと下流ノードの InPIT サイズの差を考えるとボトルネックに接続しているノードの OutPIT サイズは大きくなるが、その先の下流ノードには Data パケットが滞りなく届くので下流ノードの InPIT サイズは小さくなり、これら二つの差は待機パケット数と同じく大きくなる。一方、上流ノード

ドはボトルネックから Data パケットが帰ってこない、OutPIT サイズも InPIT サイズも大きくなるが、差は大きくなりにくいことが予想できる。したがって、図 4.4 のようにこの PIT サイズ差はボトルネックリンクに接続するノードでのみ大きくなり、前章で述べた待機パケット数と同じような振る舞いをするのが推測されるので、今回待機パケット数の代わりに PIT サイズ差を用いる手法を採用した。

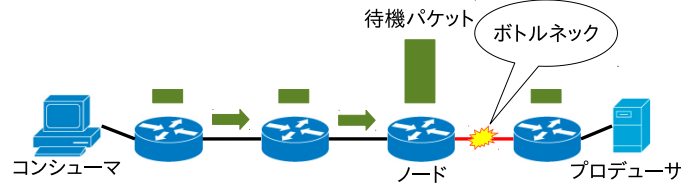


図 4.2: ボトルネックリンクを持つ直線経路ネットワーク内の待機パケット数の分布

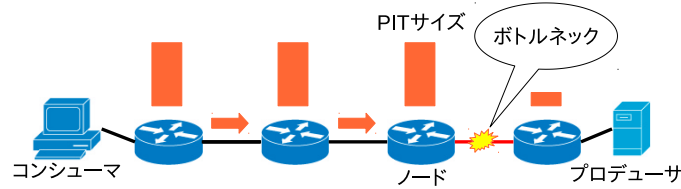


図 4.3: ボトルネックリンクを持つ直線経路ネットワーク内の PIT サイズの分布

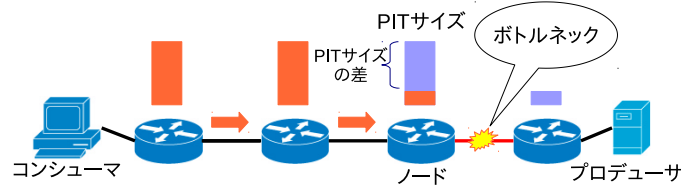


図 4.4: ボトルネックリンクを持つ直線経路ネットワーク内の PIT サイズの差の分布

4.2 における重み付けに用いる情報、 $\Omega_i^p$  と  $\omega_i^p(q)$  はそれぞれ、図 4.5 のようにある特定 Face における InPIT サイズとその Face に到着した Interest がノードから送出するときに作られる、各 Interest 送出 Face の OutPIT サイズを指す。

また、NDN において応答パケットとして送られる Data パケットはフィードバックパケットと同じ方向に流れるパケットであるので、Data パケットにフィードバック情報を付加して、フィードバックパケットとしても扱う。

したがって、Data パケットに付加されるフィードバック情報は、各 Face からの Interest パケット送出レートに重み付けを施したもの、各 Face における OutPIT サイズと隣接上流ノードとの InPIT のサイズの差に重み付けを施したもの、そして Data 送出 Face における InPIT サイズの 3 つの情報となる。

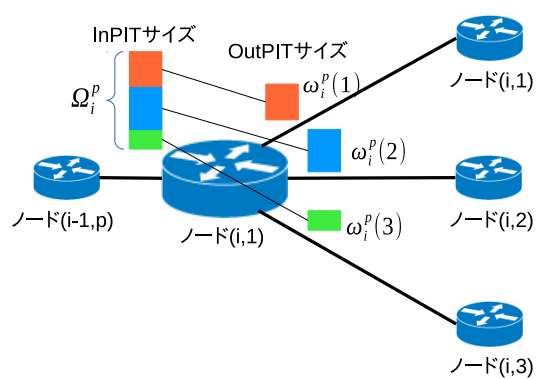


図 4.5: 重み付け情報



## 第5章 シミュレーション

本章では、NDNにおける拡散型フロー制御の特性を調べるために、コンテンツ指向ネットワークのシミュレータである ndnSIM のバージョン 1.0 を用いて、輻輳制御を特に行わない NDN ネットワークとの比較を直線型と複数経路型の二つのネットワークでシミュレーションを行った。Data パケットのサイズは 1250 バイトとした。ちなみに、ndnSIM はレート制御にリーキーバケットを使用しており、バケットの容量となる Interest パケット送出レートの最大値  $LimitMax$  は、

$$LimitMax[\text{パケット/秒}] = \frac{\text{リンク帯域幅} [\text{バイト/秒}]}{\text{Data パケットサイズ} [\text{バイト}] + \text{Interest パケットサイズ} [\text{バイト}]} \quad (5.1)$$

で決定される。また、直線型ネットワークでは 10 秒間、複数経路型ネットワークでは 15 秒間、シミュレーションを行った。性能の評価には、直線型ネットワークのモデルでは Interest 送出レートと PIT のサイズと上流ノードとの PIT サイズ差を、複数経路型のネットワークのモデルでは Interest 送出レートと PIT のサイズをそれぞれ 0.1 秒間隔で計測した。

また、拡散型フロー制御のパラメータは  $D = 0.1$  とした。

### 5.1 直線型ネットワーク

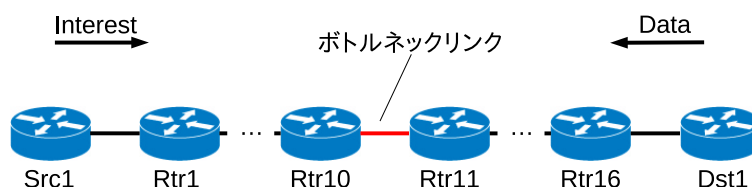


図 5.1: 直線型ネットワークのモデル

図 5.1 のような 18 ノードからなる直線型の NDN ネットワークを考える。Src1 はコンシューマ、Dst1 はプロデューサである。ノード間のリンク伝搬遅延は 5ms であり、時刻  $t = 0$  から Src1 は Dst1 に向けて Interest パケットを毎秒 200 パケットで送信し、コンテンツを要求する。Rtr10 と Rtr11 の間はボトルネックリンクとなっており、リンク帯域幅は 1Mbps、その他のリンクの帯域幅は 10Mbps である。したがって、式 (5.1) から、Interest パケット送出レートの最大値はボトルネックリンクでは毎秒 100 パケット、その他のリンクでは毎秒 1000 パケットとなる。

#### 5.1.1 実験結果

シミュレーションを通して Data パケット廃棄は、 $t = 0$  から  $t = 10$  までに、制御なしでは Rtr11 で  $t = 0.7$  から計 22 回発生し、うち 6 回は同時に 2 個の Data パケットが廃棄された。一方、提案手法では Data パケットの廃棄は Rtr11 で  $t = 1.8, 3.2, 6.5, 7.3, 9.3, 9.7$  のときに単一の Data パケット廃棄が計 6 回起きた。図 5.2a、図 5.3a より、制御なしでは Src1 からボトルネックリンクまでのリンク Interest パケット送出レートは毎秒 200 パケットとなっていることがわかる。さらにボトルネッ

クリンクでは式 (5.1) による Interest 転送量上限である毎秒 100 パケットが送られ、プロデューサまではそのレートで送られていることがわかる。一方、図 5.2b、図 5.3b より、提案手法では  $t = 0.2$  または  $0.3$  までは Interest 送出レートが一時的に毎秒 200 パケット近くまで増加したが、その後は全てのノードのリンクで毎秒 100 パケット付近の値になっていることがわかる。ちなみにどちらも Dst 1 には上流ノードがないので毎秒 0 パケットとなる。次に PIT のサイズに注目する。制御なしでは提案手法に比べて多くの Interest パケットが送出されるが、それに対して Rtr10 と Rtr11 間のボトルネックリンクのために Data パケットが返ってくる量が少ない。したがって、図 5.4a、図 5.5a のように Src1 から Rtr10 までの PIT のサイズが大きくなる。Rtr11 から Dst1 までのリンクにおいては Interest パケット送出量に対してリンク帯域幅が十分にあるために Data パケットはすぐに返ってくる。そのため、Rtr11 から Dst1 までのリンクにおける PIT のサイズはほぼゼロである。Rtr10 で PIT のサイズが 20 付近より大きくならないのはボトルネックリンクのリンク帯域幅により設定された Interest パケット送出レートの上限値よりも Rtr10 で受信した Interest パケットが大きいので、Interest パケットをリンクから送信し PIT エントリを作成する前に廃棄しているためである。制御なしでは Interest パケットの廃棄は  $t = 0.3$  以降起き続けた。一方、提案手法では図 5.4b、図 5.5b のように Src1 から Rtr10 までの PIT のサイズは 25 前後で安定している。Interest パケットの廃棄は  $t = 0.3$  でいくつかのノードで一度起きただけで後は起きていない。これは  $t = 0.3$  から  $0.4$  にかけて大きく制限された Interest 送出レートに対して送られてきた Interest パケットが多かったために廃棄されたものである。そして、制御なしでは図 5.7a より、上流ノードとの PIT のサイズ差が Rtr9 でさらに大きくなってしまっているが、提案手法では図 5.5b より、Rtr10 以外では大きくなることはなく、提案手法の通り、PIT のサイズの差が小さくなっていることが確認された。

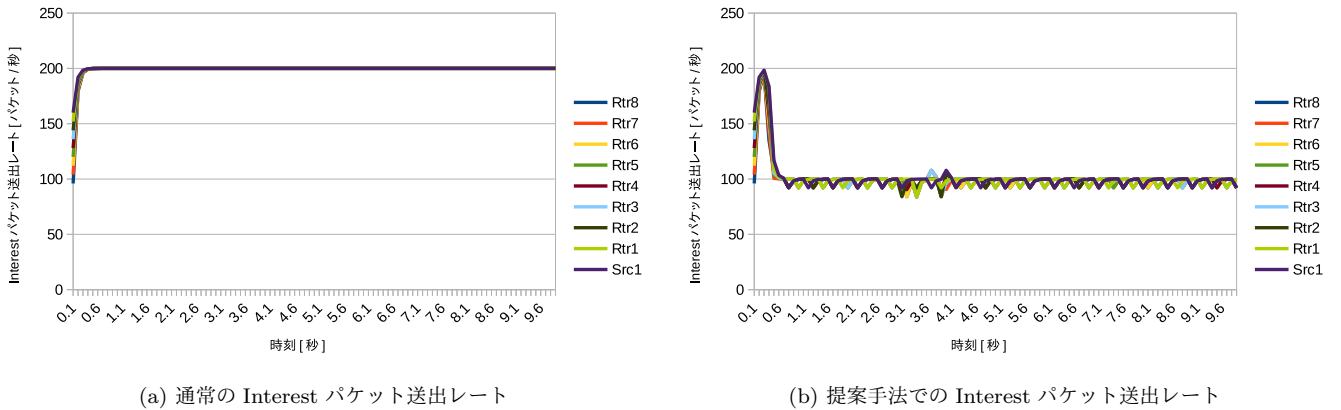
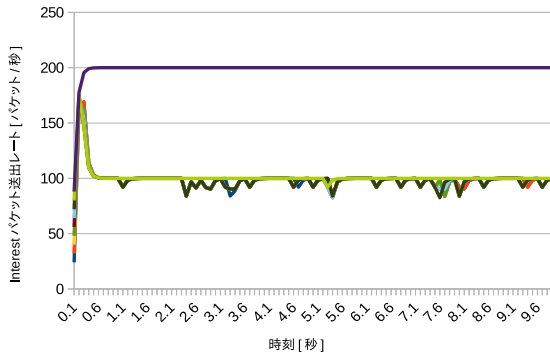
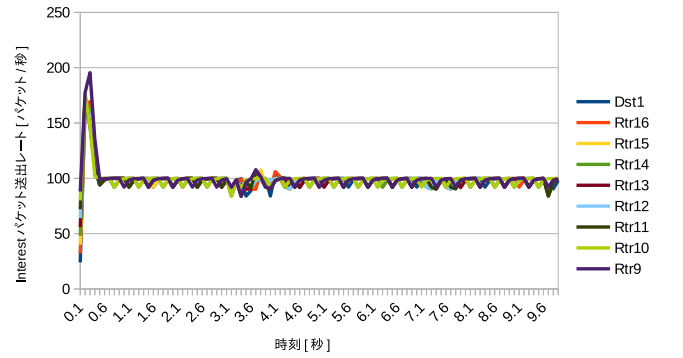


図 5.2: 直線型ネットワークにおける Src1 から Rtr8 までのノードでの Interest パケット送出レートの経時変化

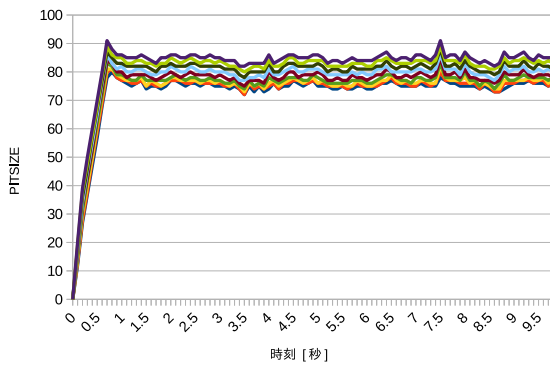


(a) 通常の Interest パケット送出レート

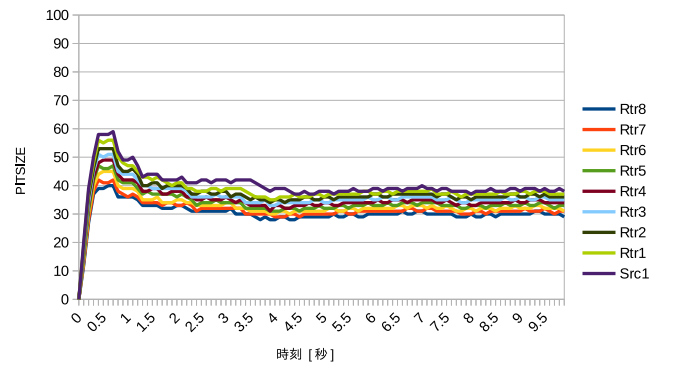


(b) 提案手法での Interest パケット送出レート

図 5.3: 直線型ネットワークにおける Rtr8 から Dst1 までのノードでの Interest パケット送出レートの経時変化

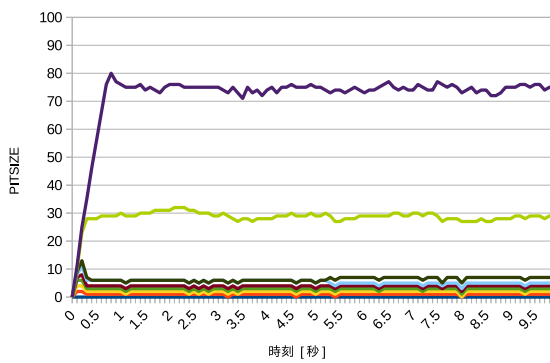


(a) 通常のノードでの PIT サイズ

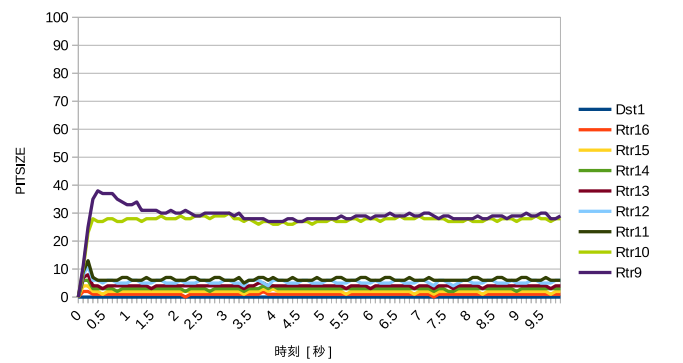


(b) 提案手法での PIT サイズ

図 5.4: 直線型ネットワークにおける Src1 から Rtr8 までのノードでの PIT サイズの経時変化

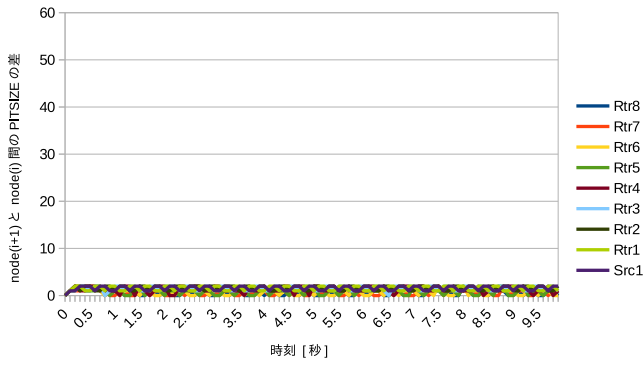


(a) 通常のノードでの PIT サイズ

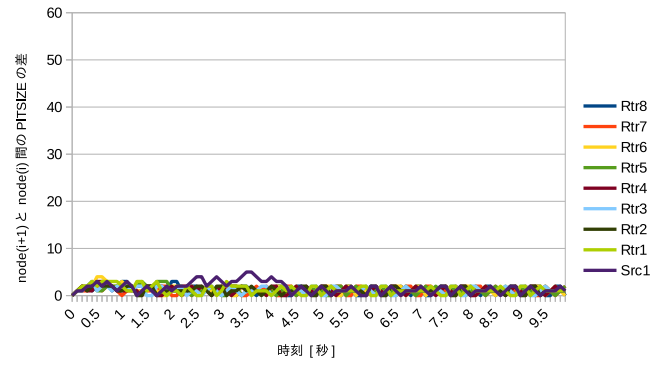


(b) 提案手法でのノードでの PIT サイズ

図 5.5: 直線型ネットワークにおける Rtr8 から Dst1 までのノードでの PIT サイズの経時変化

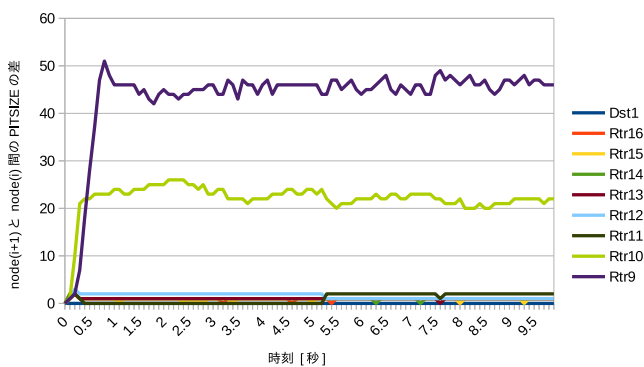


(a) 通常の上流ノードとの PIT サイズの差

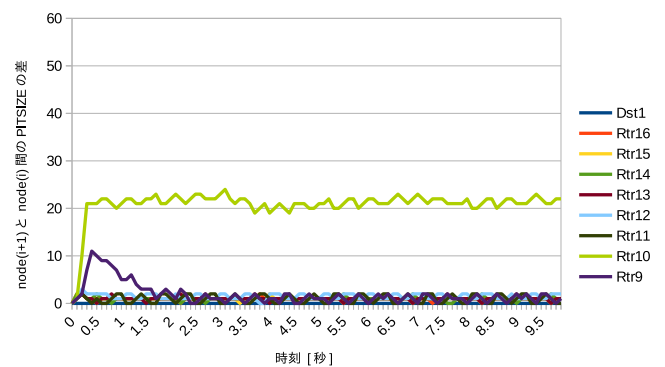


(b) 提案手法での上流ノードとの PIT サイズの差

図 5.6: 直線型ネットワークにおける Src1 から Rtr8 までのノードでの上流ノードとの PIT サイズの差の経時変化



(a) 通常の上流ノードとの PIT サイズの差



(b) 提案手法での上流ノードとの PIT サイズの差

図 5.7: 直線型ネットワークにおける Rtr8 から Dst1 までのノードでの上流ノードとの PIT サイズの差の経時変化



### 5.1.2 考察

実験結果から NDN におけるノード間の PIT サイズの差を減らすように Interest パケット送出レートを制御する提案手法によって、制御なしでは多数の Data パケット廃棄が発生する原因である過剰量の Interest パケット送出レートを過剰に削減することなく、適切に制限できたことがわかる。その結果、Data パケット廃棄回数を大きく減らすことができた。したがって、本来待機パケット数を利用していた拡散型フロー制御を、NDN におけるノードの上流ノード度との PIT のサイズの差を利用することで同様の効果を得ることができた。

## 5.2 複数経路型ネットワーク

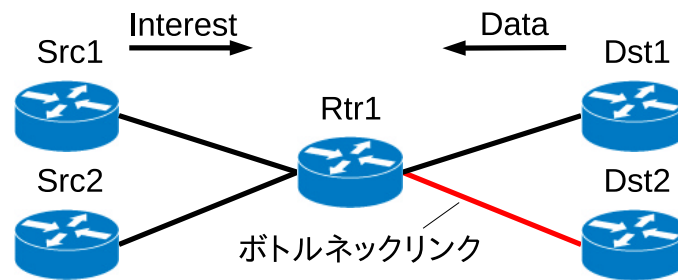


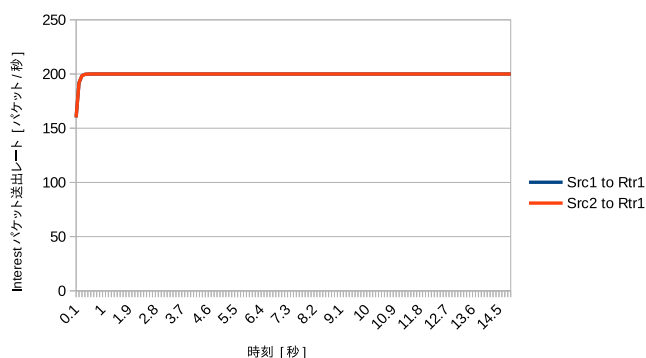
図 5.8: 複数経路型ネットワークのモデル

図 5.8 のような 5 ノードからなる複数経路型の NDN ネットワークを考える。Src1 と Src2 はコンシューマ、Dst1 と Dst2 はプロデューサである。ノード間のリンク伝搬遅延は 5ms であり、時刻  $t = 0$  から Src1 は Dst1 に向けて Interest パケットを毎秒 40 パケット、Dst2 に向けて毎秒 160 パケットで計毎秒 200 パケット送信し、Src2 は Dst1 に向けて Interest パケットを毎秒 160 パケット、Dst2 に向けて毎秒 40 パケットで計毎秒 200 パケット送信する。Rtr1 と Dst2 の間はボトルネックリンクとなっており、リンク帯域幅は 1Mbps、その他のリンクの帯域幅は 10Mbps である。したがって、直線型ネットワークと同様に式 (5.1) から、Interest パケット送出レートの最大値はボトルネックリンクでは毎秒 100 パケット、その他のリンクでは毎秒 1000 パケットとなる。

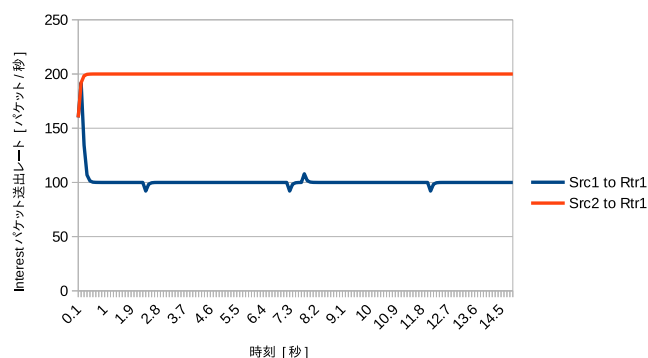
### 5.2.1 実験結果

シミュレーションを通して Data パケット廃棄は、 $t = 0$  から  $t = 15$  までに、制御なしでは Dst2 で  $t = 0.4$  から計 41 回起き、うち 1 回は同時に 2 個の Data パケット廃棄が発生した。提案手法では Data パケット廃棄は発生しなかった。制御なしでは、図 5.9a より、Src 1、Src 2 の Interest パケット送出レートはそれぞれ毎秒 200 パケットである。提案手法では、図 5.9b より、Src2 の Interest パケット送出レートは毎秒 200 パケットであるが、Src1 の Interest パケット送出レートは開始直後に毎秒 100 パケットになっている。さらに、制御なしでは図 5.10a より、Dst1 への Interest パケット送出レートは毎秒 150 パケット、Dst2 への Interest パケット送出レートは毎秒 100 パケットになっている。前者が Src1、Src2 から送られてくる Dst1 への Interest パケット送出レートの合計の毎秒 200 パケットではないのは、Src1 と Src2 がいくつか Dst1 へ同じコンテンツ識別子の Interest パケットを要求しているために、その Interest パケットに関する PIT エントリの Interest 到着 Face のリストに後からその Interest パケットが到着した Face を追加するかまたは Rtr1 のキャッシュからコンテンツの返送が行われるので、Dst1 へ Interest パケットを送出する必要がないからである。Dst2 への Interest パケット送出レートでも同じことが起きるが、毎秒 160 パケットはボトルネックリンクの Interest パケット送出レートの最大値である毎秒 100 パケットよりも多いので上限の毎秒 100 パケッ

トに制限されている。提案手法では図 5.10b のようにボトルネックリンクである Dst2 への Interest パケット送出レートがほぼ毎秒 80 パケットになっている。これは、図 5.9b でわかるように Src1 の Interest パケット送出レートが制限され、Src1 内で Interest パケットが廃棄された影響である。どの Interest パケットが廃棄されるかのパターンはシミュレータが決めており、周期的に約 1.5 秒間は Dst1 と Dst2 への Interest パケットはそれぞれ毎秒 20 パケットと毎秒 80 パケットの割合で廃棄しており、その後の約 0.5 秒間は Dst1 への Interest パケットはほとんど廃棄せず、Dst2 への Interest パケットを毎秒 60 パケットの割合で廃棄するのを繰り返していた。PIT のサイズについては、図 5.11a のように制御なしでは Interest パケットを過剰に送っているため、Src2 での PIT のサイズが Rtr1 よりも大きくなっている。Rtr1 での PIT サイズが 20 付近より大きくならないのは、Interest パケットが過剰に Src1 から送られてくるため、到着した Interest パケットを処理せずに廃棄しているからである。一方、提案手法では、ノードは自身が由来している上流ノードとの PIT サイズ差が小さくなるように Interest 送出レートを制限するので、図 5.11b のようにボトルネックリンクに接続している Dst2 へ多く Interest パケットを送り PIT のサイズが大きくなっている原因の Src1 の PIT サイズとその上流ノードである Rtr1 で Dst2 へ送ることのできた PIT サイズがほぼ同じ大きさであり、つまり、PIT サイズの差を小さくすることができている。図 5.9b について述べたように、Dst1 への Interest パケット送出レートは周期的に毎秒 80 パケットと毎秒 60 パケットを繰り返しており、制御なしの毎秒 100 パケットに比べて少ないため、PIT のサイズそのものも小さくなっている。

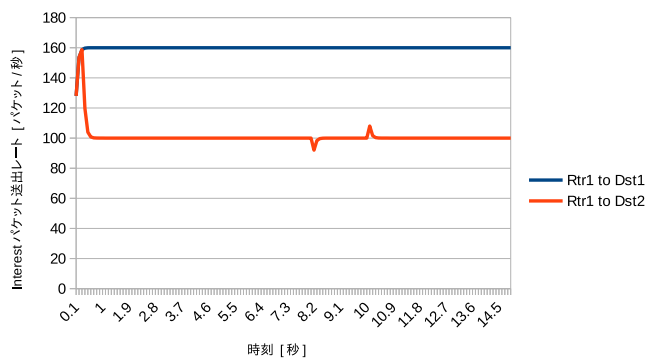


(a) 通常の Interest パケット送出レート

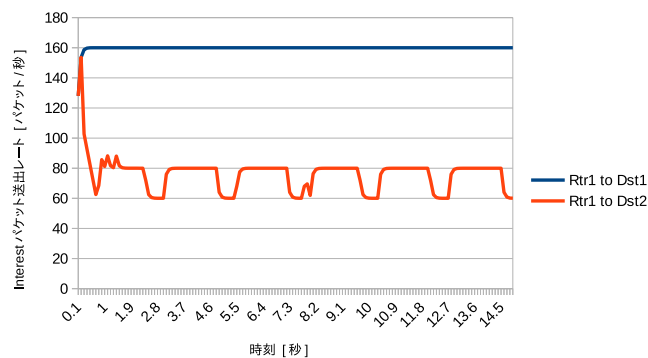


(b) 提案手法での Interest パケット送出レート

図 5.9: 複数経路型ネットワークにおける Src1 と Src2 から Rtr1 への Interest パケット送出レートの経時変化



(a) 通常の Interest パケット送出レート



(b) 提案手法での Interest パケット送出レート

図 5.10: 複数経路型ネットワークにおける Rtr1 から Dst1 へと Dst2 への Interest パケット送出レートの経時変化

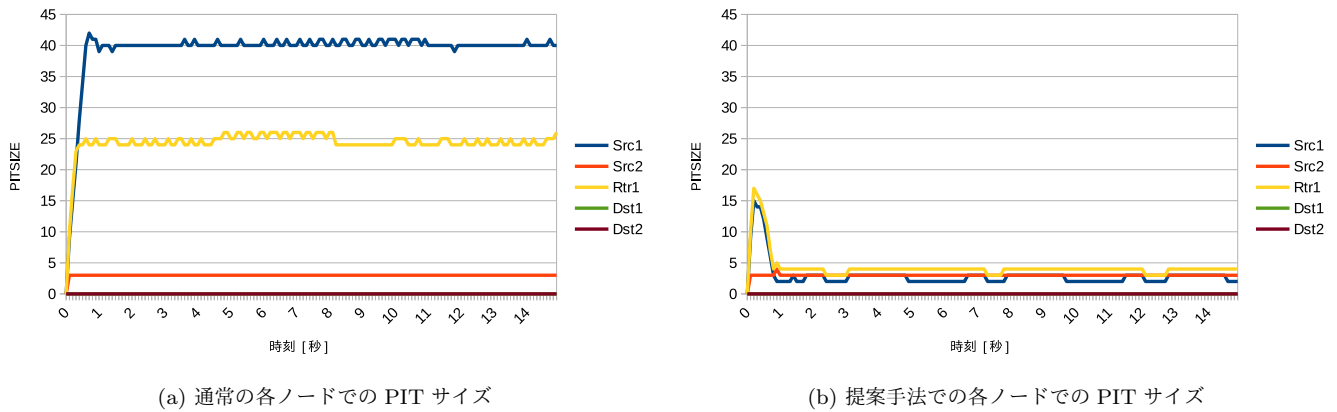


図 5.11: 複数経路型ネットワークの PIT サイズの経時変化

### 5.2.2 考察

実験結果からボトルネックリンクへ過剰量の Interest パケットを送ることで制御なしでは Data パケット廃棄を起こす原因となっている Src1 の Interest パケット送出レートだけを提案手法では制限することができている。OutPIT と InPIT を利用することで拡散型フロー制御を複数経路に拡張するための局所情報を得ることができ、適切なノードに対する Interest パケット送出レートの削減ができることがわかった。さらに、同じネットワークトポロジーで IP であれば Src1 と Src2 から同じコンテンツに対して送られた要求パケットはそのまま転送されるべきだが、PIT やキャッシュのおかげで必要以上の要求パケットがプロデューサへ送られないことで未然に輻輳を防ぐ仕組みになっていることがわかった。また、コンシューマにおいて上流ノードから指定された Interest パケット送出レートが自身の Interest パケット作成レートより低くなることで内部で発生する Interest パケットの廃棄を 2 種類の Interest パケットを作成している場合、こういった割合でそれらを廃棄するかで PIT のサイズに影響を与え、その結果、Interest 送出レートにも影響が及ぶことがわかった。一方、待機パケットを扱った場合では一時的な混雑により Rtr1 にあたるノードに多くたまることになるパケットが混雑が解消したあとは待機パケット数が減ることでパケット転送レートを回復することができるが、PIT サイズの差を扱うため、一時的な混雑が解消しても一度減ってしまった Interest 転送レートに対しての PIT サイズに変化が起きない。そのため輻輳制御としての性能を向上するために、混雑が解消した後に Interest 転送レートを回復させる方法が必要である。



## 第6章 結論

本研究では拡散型フロー制御の複数経路への拡張のための手法を NDN ルータにおける PIT のサイズの差に適応させることで NDN における輻輳制御のための Interest パケット送出レートの削減手法の提案を行った。実験から、Data パケットの受信レートに対する過剰な Interest 送出によって大きくなった PIT のサイズに関して、上流ノードとの PIT サイズの差が小さくなるように Interest パケット送出レートを制限することで Data パケットの廃棄回数を削減することができた。また、拡散型フロー制御を複数経路に拡張することで、混雑原因となるパケットを多量に送出しているノードに対してのみ強くパケット転送レートを制限するようなフィードバック情報及びパケット転送レートを削減する手法を提案、実験から NDN における Interest パケットにおいてその効果を確認することができた。しかし、ネットワークの混雑の状況の変化に対して、Interest 送出レートをより削減することは可能だが、が混雑解消後リンクの利用率を回復するように Interest 送出レートを回復するまでには至らなかった。結論として、拡散型フロー制御を NDN ルータにおける PIT の特性を利用することで複数経路においても利用可能な形に拡張した Interest 送出レートの制御を行うことによって NDN の性能を向上できることを示した。



## 参考文献

- [1] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys Tutorials*, Vol. 16, No. 2, pp. 1024–1049, Second 2014.
- [2] Athanasios V. Vasilakos, Zhe Li, Gwendal Simon, and Wei You. Information centric network: Research challenges and opportunities. *Journal of Network and Computer Applications*, Vol. 52, pp. 1 – 10, 2015.
- [3] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu. Transport control strategies in named data networking: A survey. *IEEE Communications Surveys Tutorials*, Vol. 18, No. 3, pp. 2052–2083, thirdquarter 2016.
- [4] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pp. 1–12, New York, NY, USA, 2009. ACM.
- [5] Yongmao Ren, Jun Li, Shanshan Shi, Lingling Li, Guodong Wang, and Beichuan Zhang. Congestion control in named data networking – a survey. *Computer Communications*, Vol. 86, pp. 1 – 11, 2016.
- [6] G. Carofiglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networking. In *2012 Proceedings IEEE INFOCOM Workshops*, pp. 304–309, March 2012.
- [7] Stefano Salsano, Andrea Detti, Matteo Cancellieri, Matteo Pomposini, and Nicola Blefari-Melazzi. Transport-layer issues in information centric networks. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pp. 19–24, New York, NY, USA, 2012. ACM.
- [8] L. Saino, C. Cocora, and G. Pavlou. Cetcp: A scalable receiver-driven congestion control protocol for content centric networking. In *2013 IEEE International Conference on Communications (ICC)*, pp. 3775–3780, June 2013.
- [9] N. Rozhnova and S. Fdida. An extended hop-by-hop interest shaping mechanism for content-centric networking. In *2014 IEEE Global Communications Conference*, pp. 1–7, Dec 2014.
- [10] N. Rozhnova and S. Fdida. An effective hop-by-hop interest shaping mechanism for ccn communications. In *2012 Proceedings IEEE INFOCOM Workshops*, pp. 322–327, March 2012.
- [11] Cheng Yi, Alexander Afanasyev, Ilya Moiseenko, Lan Wang, Beichuan Zhang, and Lixia Zhang. A case for stateful forwarding plane. *Computer Communications*, Vol. 36, No. 7, pp. 779 – 791, 2013.

- [12] Yaogong Wang, Natalya Rozhnova, Ashok Narayanan, David Oran, and Injong Rhee. An improved hop-by-hop interest shaper for congestion control in named data networking. *SIGCOMM Comput. Commun. Rev.*, Vol. 43, No. 4, pp. 55–60, August 2013.
- [13] C. Takano and M. Aida. Stability and adaptability of autonomous decentralized flow control in high-speed networks. In *The Sixth International Symposium on Autonomous Decentralized Systems, 2003. ISADS 2003.*, pp. 311–318, April 2003.
- [14] Masaki Aida and Chisa Takano. Principle of diffusion-type autonomous decentralized flow control. *Proceedings of the Society Conference of IEICE*, p. 194, 2003.
- [15] Masaki AIDA, Chisa TAKANO, and Kaori MURANAKA. Diffusion-type flow control scheme for multiple flows. *IEICE technical report. Information networks*, Vol. 104, No. 120, pp. 19–24, jun 2004.
- [16] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.
- [17] 会田雅樹, 高野知佐. 拡散型フロー制御のネットワーク境界におけるレート決定法の検討. 電子情報通信学会技術研究報告 : 信学技報, Vol. 103, No. 445, pp. 25–30, jun 2003.



## 謝辞

研究を行うに際し、早稲田大学大学院基幹理工学研究科の中里秀則教授にはご指導を頂きました。深く感謝いたします。またシミュレータ実装面で助けて頂いた李さん、上智大学大学院理工学研究科の加藤亮彦さん、そして同研究室の皆様にも感謝いたします。