

# コンテンツ指向型ネットワークにおける 再送制御に関する一考察

大西 遼太<sup>1</sup> 米田 孝弘<sup>1</sup> 村本 衛一<sup>1</sup>

<sup>1</sup> パナソニック株式会社 先端研究本部 〒571-8501 大阪府門真市大字門真 1006

E-mail: {ohnishi.ryota, yoneda.takahiro, muramoto.eiichi}@jp.panasonic.com

**あらまし** コンテンツ指向型ネットワークの一種である Named Data Networking (NDN) は従来の IP ベースのアーキテクチャとは異なる特徴を備えており、映像コンテンツの大規模配信などのさまざまなアプリケーションでネットワークの利用効率を飛躍的に向上させる技術として期待されている。しかし、NDN 上でリアルタイム性が求められる通信を実現するためにはアーキテクチャ特有の課題が存在する。本論文では、これらの問題を解決するため NDN 上でのリアルタイム性が求められる通信に適した再送制御を提案する。提案方式はルータ間・ルータ端末間で状態を通知することで、最も低遅延にパケットロスを検知できる。ルータ間での再送する既存方式と比較して、不要な再送要求を 1/3 に削減できること、および数%の状態通知のオーバーヘッドで提案方式が実現可能であることをシミュレーションで確認した。

**キーワード** コンテンツ指向型ネットワーク、Named Data Networking (NDN)、再送制御、リアルタイム通信

## A new retransmission scheme for Named Data Network

Ryota OHNISHI<sup>1</sup> Takahiro YONEDA<sup>1</sup> and Eiichi MURAMOTO<sup>1</sup>

<sup>1</sup>Advanced Research Division, Panasonic Corporation, 1006 Kadoma, Kadoma City, Osaka 571-8501, Japan

**Abstract** Real time data streaming such as large-scale real-time video distribution is one of the most prospective areas where Named Data Networking (NDN) can significantly improve performance and efficiency compare to the current Internet Protocol (IP) based network. However there is no retransmission scheme in NDN which can handle architecture specific issues. We propose new retransmission scheme which detect packet loss based on the inter-router Pending Interest Table (PIT) status exchange. By using this scheme routers can detect packet loss in minimum latency. In comparison with existing scheme with simulator, our method shows that it can reduce useless retransmission request to one third with a few percent of overhead increase.

**Keyword** Information Centric Networking (ICN), Named Data Networking (NDN), Retransmission, Real-Time Communication

### 1. はじめに

Named Data Networking (NDN) および Content Centric Networking (CCN) は 2009 年 Van Jacobson らにより提案された次世代インターネットのためのネットワークアーキテクチャである<sup>1)</sup>。NDN および CCN は Information Centric Networking (ICN) の中で活発に研究開発が行われている技術の一つであり、近年では Internet Research Task Force (IRTF) の Information Centric Networking Research Group (ICNRG) や ITU-T SG13 において活発な議論が行われるなど、研究活動が盛り上がりを見せている<sup>2,3)</sup>。またユースケースについての議論も盛んであり、特に大規模なリアルタイムストリーミングに関しては NDN によって性能や効率の改善が可能であるため、実装についての議論も活発に行われている<sup>4,5)</sup>。しかし、NDN では受信者駆動の通信を行うため、パケットロス発生時に、従来の再送制御方式では遅延の増大や無駄なパケットの送信を引き起こしてしまうという問題点がある。本論文ではこれらの問題点を解決し、NDN をリア

ルタイム性が求められる通信にも適用可能とするための、低遅延かつ効率的な新たな再送制御方式の提案を行う。

本論文の構成は以下の通りである。

まず 2 章で、NDN の技術的特徴とリアルタイムな通信における再送制御の要求について述べる。3 章では NDN 上での再送の関連研究とその課題について述べる。4 章では本論文で提案する再送制御の手法について説明する。5 章ではシミュレータを用いた特性の評価結果を記載し、6 章でまとめる。

### 2. NDN の特徴とリアルタイム通信における再送制御への要求

#### 2.1. NDN の特徴

従来のインターネットは、ホストのロケーションを示す IP アドレスを用いて通信を行うロケーションベースのネットワークアーキテクチャである。一方、NDN はホストのロケーションは使わず、情報を識別する Content Name と呼ばれる ID を使って通信を行う、情報指向型のネットワークアー

キテクチャである。NDN ではネットワーク上の各ルータが Content Store (CS) と呼ばれるキャッシュを備えており、一度転送したコンテンツ(情報)を CS に記録する。そのため、特定のコンテンツに対して大量のアクセスが集中するような環境でも、CS のキャッシュの効果によりネットワークのトラフィックの増大を抑えた効率的な配信を実現できる。これらの利点の実現にあたり、NDN のアーキテクチャでは以下の方針を採用している。

- ・特定のコンテンツの配信を、伝送するパケット単位で受信者が要求・取得することで実現する受信者駆動の通信方式
- ・要求・取得パケットの送受信の状態をルータが記憶・処理するステートフルな転送管理

## 2.2. NDN での通信の概要

NDN では、コンテンツを要求するための Interest パケットとコンテンツを伝送するための Data パケットの 2 種類のパケットを使って通信を行う。NDN におけるノードは、データ受信者であり Interest パケットの発行を行う Consumer、コンテンツを保持し Data パケットを送信する Publisher、Data パケットおよび Interest パケットの転送を行うルータの 3 種類である。これらのノードには、送受信した Interest パケットの入力元インターフェイスと出力先インターフェイスを記憶する Pending Interest Table (PIT)、各 Content Name に関するルーティング情報を記載した Forwarding Information Base (FIB)、一度転送したコンテンツをキャッシュするための CS が備わっている。NDN の通信手順は以下の通りである。

最初に Consumer が取得したいコンテンツの Content Name を Interest パケットに記載してルータに送信する。

Interest パケットを受信したルータは、FIB に従って Interest パケットを転送する。その際、ルータは PIT に Interest パケットに記載された Content Name に関するエントリを、入出力したインターフェイスの情報と共に記録する。また、ルータは PIT に記憶されたエントリと同一の Content Name が記載された Interest パケットを他の Consumer から受信した場合、既にある PIT のエントリに入力インターフェイス情報の追加を行い、それ以上 Interest パケットの転送を行わない。この PIT のエントリは、ルータが Content Name に対応する Data パケットを受信・転送するまで記憶される。

ルータにより転送された Interest パケットが Publisher に到達すると、Publisher は Interest パケットの Content Name に対応する Data パケットをルータに対して送信する。

Data パケットを受信したルータは PIT エントリに記載された Interest パケットの入力インターフ

ェイスに対して Data パケットを転送する。このようにルータが転送処理を繰り返すことで、最終的に Data パケットは Interest パケットを送信した Consumer に到達する。Data パケットは転送される際に、ルータ上の CS に Content Name と紐づく形でキャッシュされる。ルータが Interest パケットを受信した際、その Content Name に対応する Data パケットが CS にキャッシュされている場合には、Interest パケットの転送を止め、CS から Data パケットを取り出して Interest パケットを受け取ったインターフェイスに直接送信する。以上のように、NDN ではルータが PIT を用いてコンテンツのパケットに関する状態管理を行い、CS を用いた Data パケットのキャッシュを行うことで、効率的なコンテンツの配信を実現している。

## 2.3. NDN 上でのリアルタイム配信における再送制御への要求

NDN は効率的なコンテンツの配信を実現できるため、大規模かつリアルタイムな映像配信が有望なユースケースであると我々は考えている。具体的な例としてサッカー等の複数の試合が同時刻に複数の会場で開催されるような大規模なスポーツイベントを挙げる。このようなイベントでより充実した観戦環境を提供するためには、各会場の個々の観客がタブレット等の端末で自会場の任意視点のライブ映像やハイライト映像をリアルタイムに視聴するだけでなく、他会場の映像も同時に視聴可能とするサービスが必要であると考えられる。しかし、このようなリアルタイムな映像配信を実現するためには、リアルタイム性を保ちつつ通信の信頼性を向上させることが重要である。なぜなら、NDN の通信においても無線区間の通信やネットワークの輻輳によってパケットロスが発生する可能性があり、パケットロスを回復しなければ、配信映像に乱れが生じるからである。また、パケットロスに対する回復手段がある場合でも、パケットロスが発生してから Consumer 側で正常にパケット受信が完了するまでの遅延時間が長くなると、映像の再生時刻にパケットの受信が間に合わず再生映像が停止してしまい、ユーザーエクスペリエンスが大幅に低下してしまう。このようなシステムに許容される遅延時間は Consumer の初期バッファ時間に依存するが、我々の想定するユースケースでは頻繁に映像の切替が発生するため、スムーズな視聴のために初期バッファ時間を 500ms 以下とし、映像の切替を 1 秒程度で完了することが必要であると考えられる。一般にパケットロスを回復するために再送制御が用いられるが、本ユースケースでリアルタイムな通信を行うためにはパケットロスの発生を素早く検出し、初期バッファ時間の範囲である 500ms 以下でパケットロス

を回復できる再送制御技術が必要である。NDN は Consumer が送信した Interest パケットに対して、Publisher またはルータから Data パケットが返送される PULL 型のネットワークアーキテクチャである。そのため、パケットロスが発生した場合でも Interest パケットを送信した Consumer、もしくは Interest パケットを中継したルータがパケットロスを検出して、Interest パケットの再送処理を行う必要がある。

しかし、ネットワークが輻輳状態に陥っている際に、不正確なパケットロスの検出で無駄な Interest パケットの再送信が行われると、輻輳状態が悪化してリアルタイム性がさらに毀損される恐れがある。そのため、NDN の再送制御は、パケットロスの検出に関して、素早さだけでなく高い正確性を備えて、無駄な Interest パケットの再送信を行わないようにする必要がある。

### 3. NDN 上の再送制御の関連研究と課題

NDN は PULL 型のネットワークアーキテクチャであることに加え、ルータ上の CS に Data パケットがキャッシュされることや、ルータが送受信したパケットの状態を記憶していることなど、IP での通信とは様々な点で異なった特徴を備えている。そのため、IP の通信で一般的に利用されているパケットロスの検出方式や再送方式を NDN 上でそのまま用いると、NDN の特性を活かせないばかりか性能を悪化させてしまう。この章では、NDN 上の再送制御に関する関連研究と、NDN におけるパケットロス検出および再送制御の課題について述べる。

#### 3.1. パケットロスの検出における課題

従来の研究では、NDN 上でのパケットロスの検出方法として、大きく分けて 3 通りの方式が提案されている。しかしいずれの方法を用いても、誤ってパケットロスを検出して無駄な再送処理が発生する問題や、パケットロスを検出するまでに時間がかかりすぎて再送制御が間に合わなくなってしまいう問題がある。これらの方式と問題点について以下に詳細に説明する。

##### 3.1.1. タイムアウトベースの方式

各 Consumer や各ルータが Interest パケットを送信後、ある閾値時間以内に Data パケットが到着しなかった場合に、パケットロスが発生したと判断する方式である<sup>6,7)</sup>。しかし、NDN では Interest パケットを送信してから Data パケットが到着するまでの遅延時間は、ネットワークの混雑によるキューイング遅延の発生に限らず様々な要因で変化する。たとえば、NDN では Data パケットが Publisher だけでなくネットワーク上のルータの

CS にも分散して存在するため、Consumer が送信した Interest パケットに対して Data パケットを返送するノードが、Publisher と経路上のルータの間でコンテンツごとに変化することも一因である。

タイムアウトベースのパケットロス検出方式は、このような Data パケットの到着遅延が常に揺らぐ環境中でタイムアウトの閾値を決定しなければならないため、閾値を小さくしすぎた場合には、誤ってパケットロスと判断して無駄な再送を引き起こす。一方、閾値を長くしすぎると、実際にパケットロスが発生した際にもパケットロスを検出するまでに時間を要してしまうため、パケットロス発生から Data パケット到着までの遅延を増大させてしまう。リアルタイム性が必要とされる通信では、無駄な再送を発生させず、かつ低遅延な再送を実現する必要があるが、ネットワーク上の Publisher やルータ上の CS に Data パケットが分散している環境で適切なタイムアウト値を決定することは困難である。そのため、タイムアウトベースの方式はリアルタイム性が要求される通信の再送方式としては適していない。

##### 3.1.2. シーケンス番号ベースの方式

IP で用いられているパケットロス検出方式を NDN に拡張したものと、シーケンス番号ベースの方式がある<sup>8)</sup>。この方式では一連の映像コンテンツの Data パケットにシーケンシャルな Content Name が付与されており、Consumer が映像を要求する際には、シーケンス番号順に Content Name に対する Interest パケットを送信すると仮定している。またこのパケットロス検出方式では要求した Data パケットをシーケンシャルに受信できる前提を置いているため、受信した Data パケットのシーケンス番号に抜けが発生した場合にパケットロスと判断する。しかし NDN では、Data パケットの受信順が Interest パケットの送信順に対応しているとは限らない。例えば、より大きなシーケンス番号を持つコンテンツの一部のみがルータの CS にキャッシュされており、残りのコンテンツは Publisher にしか存在していない場合、CS に存在する Data パケットは Publisher にしか存在しない Data パケットよりもより早く届く。つまり大きなシーケンス番号を持つ Data パケットを小さなシーケンス番号を持つ Data パケットよりも早く受信してしまうことで、誤ってパケットロスを検出してしまふ。このため、シーケンス番号ベースのパケットロス検出方式では正確にパケットロスを検出できない。

##### 3.1.3. Nack パケットベースの方式

Interest パケットを受け取ったノードが、対応するコンテンツがまだ存在していない場合や、ネット

ネットワークの輻輳の発生を検知した場合に **Negative Acknowledgement (Nack)** パケットを用いて下流のノードに問題の発生を通知する方式も提案されている<sup>9,10,11)</sup>。これらの NDN 上の **Nack** パケットは IP におけるそれとは異なり、**Data** パケットの送信側に位置するノードが受信側に対して送信する点に特徴がある。

コンテンツが存在していない場合に **Nack** パケットを送信する方式<sup>9)</sup>は、**Interest** パケットに対応する **Data** パケットが **Publisher** に存在していない場合に、明示的なメッセージを **Consumer** に送信する。この方式を用いることで、**Interest** パケットを送信した **Consumer** は、**Data** パケットの到着遅延が発生した際に、遅延の原因が「**Data** パケットが存在しないこと」によるものか「パケットロスが発生したこと」によるものかを判別できるため、パケットロスの誤検出を減少させることができる。しかし、この方式もパケットロスの判定そのものはタイムアウトベースの方式に頼っているため、誤ったタイムアウト値の設定によるパケットロスの誤判定や遅延時間の増大から逃れることはできない。

一方、ネットワークの輻輳を検知した場合に **Nack** パケットを送信する方式<sup>11)</sup>は、各ルータにおいて転送した **Interest** パケットの量から、将来受信される **Data** パケットの量を予測する。そして予想される **Data** パケットの総量が上位リンクの帯域を超えた場合に **Interest** パケットをドロップさせ **Nack** パケットを **Consumer** 側に位置するノードに送信する。これにより **Data** パケットの流量をリンクの帯域以下に抑えることができる。しかし、この方式を用いた場合でも、各 **Interest** パケットに対して受信される **Data** パケットのサイズが一定でない環境や、パケットごとに遅延時間が揺らぐ環境では、ネットワークの帯域を有効に利用できない問題や、パケットロス発生を完全に防げない問題がある。このため、**Nack** パケットを用いた方法でも正確な再送制御を行うことは困難である。

以上のことから、**Nack** パケットを用いる方式でも、2.3 節で説明した再送処理の要件を満たすことはできない。

### 3.2. ルータによる再送制御の課題

IP での通信においては、パケットロスの検出および再送制御は、クライアントもしくはサーバが行う **End to End** の方式が一般的である。一方、NDN では転送経路上のルータが **Contents Name** を記載した **PIT** を保持しているため、ルータが **PIT** の内容からパケットの送受信のステータスを認識可能である。そのため **Consumer** に加えて経路上のルータがパケットロスを検出し、**Interest** パケットの再送を行う等の再送制御を行うことが可能である。

このように NDN では経路上のルータが再送制御を行うことで、**End to End** の再送制御に比較してより早期にパケットロスを検出し再送制御を行うことができる。しかし、ルータによる再送制御を行う場合、タイムアウトによるパケットロス検出などの単純な方式では、パケットロスの誤検出に加えて、不要な **Interest** パケットが送信されるという問題がある。これは、あるリンクで **Data** パケットのロスが発生した際、パケットロスの発生したリンクよりも **Consumer** 側にあるすべてのルータがパケットロスを検出し、ほぼ同時に **Interest** パケットの再送処理を行ってしまうからである。つまり 1 つのパケットロスによって複数の不要な **Interest** パケットの再送が引き起こされてしまい、再送制御自体がネットワークの輻輳を発生させる要因となる。また、不要な **Interest** パケットの送受信は、ルータの処理負荷増大の原因となる。以上のことから、ルータによる再送制御を利用する際には再送処理によって無駄なパケットが発生しないようにするという点にも考慮する必要がある。すなわち、NDN の再送における課題として、「正確かつ素早いパケットロスの検知」と「再送の発生時に無駄な **Interest** パケットを抑制する」という 2 点が挙げられる。

## 4. 提案手法

本章では、3 章で述べた 2 つの課題を同時に解決する再送制御方式を提案する。以降、ネットワーク上のあるノードから見て、**Consumer** が位置する方向を下流（下位）、**Publisher** やコンテンツが位置する方向を上流（上位）と呼ぶ。

NDN の通信において最も効率の良い再送制御を実現するための方法は、パケットロスが発生したリンクに隣接するノードのうち下流側に位置するノードが **Interest** パケットの再送信を行うことである。NDN のノードは **CS** を備えているため、パケットロスが発生したリンクの上流側に位置するノードの **CS** にはロスした **Data** パケットがキャッシュされている。この場合、下流側のノードが **Interest** パケット再送信すると、**Interest** パケットが 1 ホップ転送されるだけで、上位のノードの **CS** からロスした **Data** パケットが返送される。このため最短の時間でロスしたパケットを回復できる。パケットロスが発生したリンクに接するルータ同士が **Interest** パケットの再送によってパケットロスを回復する様子を図 1 に示す。



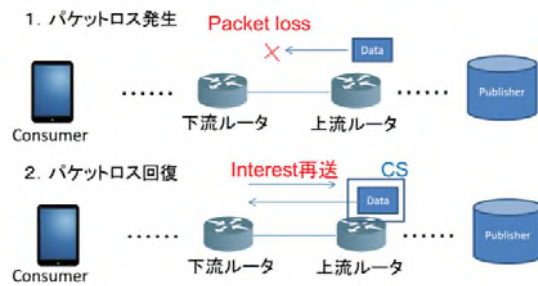


図 1 最も効率の良い再送の例

この方法は再送処理のための Interest パケットの転送が、パケットロスが発生したリンクに接続するノード間に限定されるため、ネットワーク内に無駄な Interest パケットが発生することを抑制できる。以上の考察から提案する再送方式は以下の方針で設計した。

- パケットロスが発生したリンクに接するノード間でホップバイホップにパケットロスを回復する
- 再送要求を行うノードはパケットロスが発生したリンクの直下のノードのみとする

上記の特徴を備えた再送制御を実現するためには、パケットロスの発生したリンクの直下のノードが、正確かつ迅速にパケットロスの検出を行う必要がある。我々はこの課題を解決するため、NDN のノードが PIT に Interest パケット送受信のステートを持つ点に着目した「PITinfo ベースの再送制御方式」を提案する。PITinfo ベースの再送制御方式では、任意のリンクに隣接する上位のノードから下位のノードに対して、PIT の情報を通知する。次に、下位のノードでは通知された PIT の情報と自身の PIT の内容を比較することでパケットロスを正確に検知し、素早く Interest パケットの再送を行う。PIT の比較によりパケットロスを検出する原理については 4.1 節で述べる。また、上記の PIT 情報の通知によって発生する通信オーバーヘッドの圧縮と、従来の NDN アーキテクチャとの親和性を考慮し、本論文の提案手法では PIT 情報の通知に Data パケットを用いる。Data パケットを用いて PIT 情報を通知する方法の詳細およびシーケンスについては、4.2 節で述べる。4.3 節では提案手法のアルゴリズムの Consumer・ルータ・および Publisher での実装について説明する。

#### 4.1. PIT 情報の比較によるパケットロスの検出

本方式におけるパケットロス検出の仕組みについて図 2 を用いて説明する。

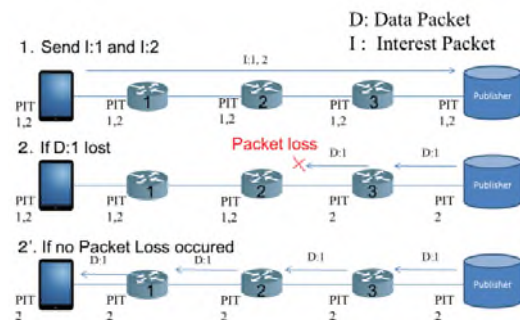


図 2 PIT 情報の比較でパケットロスを検出する例

図 2 は左端に位置する Consumer が、ルータ 1~3 を介して接続された Publisher から Content Name 「1」および「2」のコンテンツ取得するシーケンスを示している。まずシーケンス 1 で Consumer は Content Name 「1」および「2」のコンテンツに対する Interest パケット I:1 および I:2 を送信する。送信された I:1 および I:2 はルータ 1~3 を経由して Publisher に到達する。このとき経路上の Consumer・ルータ・Publisher の PIT ではいずれも Content Name 「1」と「2」のエントリが記載されている。言い換えれば、経路上の各ノードの PIT の状態は一致した状態となる。ここで I:1、I:2 に対応する Data パケットを D:1、D:2 と記載する。

続いてシーケンス 2 で Publisher が I:1 に対応する D:1 を送信する。D:1 を受信したルータ 3 はルータ 2 に Data パケットの転送を行う。このときルータ 3 とルータ 2 の間のリンクでパケットロスが発生し、D:1 がロスしたものとす。ここで、各ノードの PIT の状態を見ると、Publisher およびルータ 3 は D:1 の送信を完了したため Content Name 「1」に関するエントリが消去され「2」に関するエントリのみが残っている。一方 Consumer およびルータ 1・2 は D:1 をまだ受信していないため、PIT に Content Name 「1」および「2」に関するエントリが両方存在している。つまり、D:1 の受信を完了したノードと D:1 の受信していないノードとの間で、PIT の状態に不整合がある状態となる。図 2 のシーケンス 2 の例では、ルータ 2 とルータ 3 の PIT の状態に不整合が生じている。

シーケンス 2' はパケットロスによって D:1 が損失せず、Consumer に届いた場合の図である。パケットロスが発生しなかった場合、すべてのノードの PIT に「2」のエントリのみが存在する状態となるため、各ノードの PIT の状態に不整合は発生しない。

このように、パケットロスの発生は PIT の状態の不整合で検出できる。つまり、シーケンス 2 の状態で、ルータ 2 がルータ 3 の PIT 情報を受取ることができれば、ルータ 2-ルータ 3 間で「1」に関するパケットロスを検出できる。

以上のことから、各ノードの PIT 情報の比較により、パケットロスが発生したリンクの直下のノードが正確にパケットロスの検知できる。

#### 4.2. 隣接ルータへの PIT 情報の通知

次に、下流の隣接ルータに PIT 情報を通知する方法について説明する。

我々は、各ノードが Data パケットを転送する際に、Data パケット内に PIT 情報を記述するフィールドを追加・送信することで下流のノードに PIT 情報を通知する方法を提案する。これは、従来の NDN のアーキテクチャへの親和性と、PIT 情報の通知に係る通信オーバーヘッドについて考慮した結果である。たとえば、他の通知方法として PIT 情報を通知するための制御パケットを導入する方法も考えられるが、本提案と比較して現状の NDN アーキテクチャに対する変更が多い。また提案手法は既存の Data パケットにピギーバックさせる形で PIT 情報の送信を行うため、PIT 情報のためだけにパケットの構成・送信を行う必要がない。

Data パケットに PIT 情報の記載する方法を用いた場合のパケットロス検出のシーケンスを、図 3 を用いて説明する。

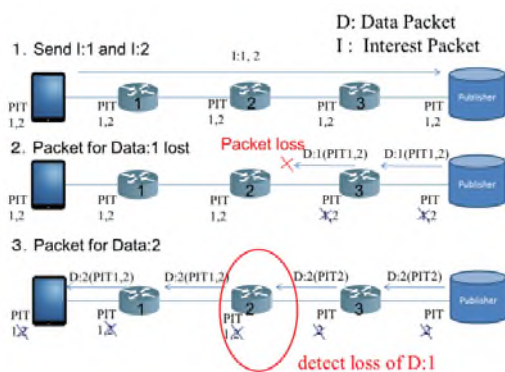


図 3 提案方式におけるパケットロス検出

図 3 のネットワークポロジは図 2 と同様である。シーケンス 1 では図 2 と同様に Consumer が Contents Name 「1」「2」に対する Interest パケット I:1, I:2 を送信し、その結果すべてのノードの PIT に「1」「2」に関するエントリが存在している。

シーケンス 2 で、Publisher が I:1 に対応する D:1 を送信する際に自身の PIT の情報を示すデータ（図中では (PIT1,2) と記載）を追加してルータ 3 に送信する。D:1(PIT1,2)を受信したルータ 3 は、自身の PIT と受信した PIT 情報が一致するため、D:1 に自身の PIT 情報を記載してルータ 2 に対して転送を行う。このときパケットロスが発生し D:1 がロスしたとする。パケットロスの検出は続くシーケンス 3 で行われる。

シーケンス 3 では、Publisher が I:2 に対応する D:2 を送信する。Publisher およびルータ 3 は D:2 を転送する際に自身の PIT データとして (PIT2) を記載して下位のノードに対して Data パケットの転送を行う。ルータ 3 では受信した PIT 情報と自身の PIT の状態が一致するため、再送処理は行わない。一方、ルータ 2 ではこの Data パケット D:2(PIT2)を受信し PIT 情報の比較を行った際に、自身の PIT に存在する「1」に関するエントリが、受信した PIT 情報存在しない。このため「1」のパケットが隣接する上流リンクでロスしたことを検知し、直ちに I:1 の再送を行うことができる。

ルータ 2 やルータ 1 が D:2 を転送する際には、PIT 情報として「1」「2」の両方の情報が記載される。そのためルータ 1 や Consumer が PIT 情報を比較する際は、PIT 情報と自身の PIT の状態が一致するため再送処理は行われない。このため、パケットロスが発生したリンク以外で不要な Interest パケットの送信は行われない。

このように Data パケットに記載する PIT の情報を各ノードが Data パケットを転送するたびに書き換えていくことにより、パケットロスが発生したリンクの直下に位置するノードのみがパケットロスを検出し再送制御を行う。これにより、再送制御を最も効率的な場所で行うことが可能になると共に、再送制御によって発生するパケットの数を最小に抑えることができる。つまり 3 章で述べた課題を解決できる。

#### 4.3. 提案するアルゴリズムの実装

ここまで章で簡単な例を用いて提案方式の原理を説明してきた。本節では Publisher・ルータ・Consumer における提案方式のアルゴリズムの実装について説明する。また以降の説明では、NDN の一般的なシーケンスに関する説明は省略し、提案手法の特徴である Data パケット受信時と Data パケット送信時のアルゴリズムに関してのみ記載する。アルゴリズムの説明の中で用いられる記号はそれぞれ表 1 に示すものを意味している。

##### 4.3.1. Publisher におけるアルゴリズム

Publisher が Data パケットを生成し、インターフェイス「face1」からネットワークへ送信する際のアルゴリズムは以下の通りである。

```

When DATA(name) created and sent to face1:
for each entry in PITself :
    if InFace(entry) == face 1 :
        Append entry to PITinfo in DATA(name)
    Send DATA(name)
    Erase entry(name) from PITself
    
```

Publisher ではパケットロスの検知は行わないが、自身の PIT 情報を Data パケットに記載して送信するので、下位のルータでパケットロスを検知できる。

#### 4.3.2. ルータにおけるアルゴリズム

ルータがインターフェイス「face1」から Data パケットを受信し、インターフェイス「face2」に対して Data パケットを転送する際のアルゴリズムは以下の通りである。

When (DATArecv(name) Received from face1 and will forward to face2 :

```

Extract PITinfo from DATArecv(name)
for each entry in PITself :
  if OutFace(entry) == face1 :
    if entry ∈ PITinfo :
      Reissue Interest of entry
  if InFace(entry) == face2 :
    Append entry to PITinfo in DATAsend(name)
Forward DATAsend(name)
Erase entry(name) from PITself

```

ルータは Data パケットを受信した際に、自身の PIT エントリで face1 から送信したものと、受信した PITinfo を比較する。もし PITinfo に記載のないエントリがあればパケットロスが発生したと判定し再送制御（当該 entry の Interest パケットの再送信）を行う。また Data パケットの転送を行う際には、自身の PIT エントリの中で face2 から受信したものを Data パケットの PITinfo として記載し、転送する。

#### 4.3.3. Consumer におけるアルゴリズム

Consumer が Data パケットをインターフェイス「face1」から受信した際のアルゴリズムは以下の通りである。

When DATA(name) Received from face1 :

```

Extract PITinfo from DATA(name)
for each entry in PITself :
  if OutFace(entry) == face1 :
    if entry ∈ PITinfo :
      Reissue Interest of entry
  Erase entry(name) from PITself

```

Consumer は PIT の転送はおこなわないが、上位のノードから受信した Data パケットに含まれる PITinfo と自身の PIT を比較することにより、自身

表 1 アルゴリズム内の記号の説明

記号	意味
DATA(name)	Content Name「name」に対応する Data パケット
PITinfo	受信した Data パケットに含まれる PIT 情報
PITself	ノード自身の PIT
Entry	PIT 内の個々のエントリ
InFace(entry)	entry に対応する Interest パケットが受信したインターフェイス
OutFace(entry)	entry に対応する Interest パケットを送信したインターフェイス
DATArecv	ルータにおいて受信された上位ノードの PIT 情報を含む Data パケット
DATAsend	ルータが転送する自身の PIT 情報を含む Data パケット

の接しているリンクで発生したパケットロスを正確に検出し、再送制御を行うことができる。

## 5. シミュレーション評価

提案方式のパケットロスに対する有効性を確認するため、シミュレータで従来方式との性能比較を行った。性能比較では、パケットロス環境下において Consumer による Interest パケット送信から Data パケット受信までの遅延時間をどれだけ削減できるかという点と、再送処理を行う際のネットワーク全体の Interest パケットの送信数をどれだけ抑制することができたかという点を中心に評価した。シミュレーション環境には、NS-3 ベースのシミュレータである ndnSIM<sup>12)</sup>を使用した。

### 5.1. トポロジ

本シミュレーションでは大規模映像配信のネットワークモデルを想定し、図 4 のようなツリー型のトポロジを採用した。ネットワーク内のノードの配置は、左側に Consumer「C」と「C'」が計 12 ノードあり、4 ノードずつルータ「R」に接続されている。ルータ「R」は 3 ノードともルータ「R'」に接続されており、「R'」は Publisher「P」に接続されている。またすべてのリンクの片方向遅延は 10ms とした。

今回の評価シナリオでは、すべての Consumer が一斉に Interest パケットの送信を開始するものとした。また Consumer「C」の 9 ノードはシーケンス番号 1~1000、Consumer「C'」はシーケンス番号 1001~2000 のコンテンツに対して 10ms ごとに一定の周期でシーケンシャルに Interest パケットを送信するものとした。加えて、ネットワーク上にコンテンツが分散配置された環境を想定し、ルータ「R」の 3 ノードには事前にシーケンス番号 100~200 と 400~500 の Data パケットを配置した。

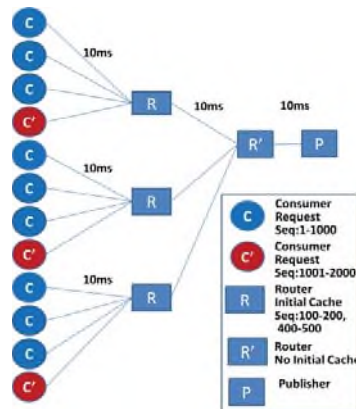


図 4 シミュレーションに用いたネットワークトポロジ

このようにコンテンツが分散配置されているので、Consumer はルータ「R」上にキャッシュされているコンテンツに関しては最短 20ms で取得できる。一方 Publisher 「P」上にしかないコンテンツを取得するには最低 60ms 必要となる。

性能評価を行う際には、ネットワークの各リンクの packet loss rate を 0~10% の間で評価した。

### 5.2. 比較モデル

提案する PITinfo ベースの再送制御方式との比較を行う従来手法のモデルとして、以下の 2 種類のタイムアウトベースの再送モデルを ndnSIM 上に実装し評価を行った。比較方式ではいずれのモデルも、すべてのノードに同一の RTO (Retransmission Time Out) 値を設定した。

#### 5.2.1. Consumer のみが再送を行うモデル

先行研究で一般的に行われている Consumer ベースの再送制御の特性と比較するため、Interest パケットを送信後 RTO 経過で、Consumer のみが再送を行う再送制御モデル(以降 Consumer-Only モデルと記す)を用意した。RTO の値は 20ms~200ms の間で評価した。

#### 5.2.2. ルータと Consumer の両方が再送を行うモデル

Consumer だけでなくルータも再送を行う再送制御モデル (以下 Router-Base モデルと記す) も用意し、提案方式と比較した。Router-Base モデルでも複数の RTO 値で特性を評価した。

### 5.3. 評価結果

#### 5.3.1. パケットロスによるデータ到着遅延

各モデルについて Consumer が Interest パケットを送信してから、対応する Data パケットを受信するまでにかかる遅延時間(Latency)を計測し、Latency の累積確率分布 (CPD : Cumulative

Probability Distribution ) を求めた。この測定により、各モデルが packet loss の発生時にどの程度遅延の発生を抑えてデータの取得を行えるかを評価する。

まずは提案手法について、packet loss rate を 0~10% の間で変化させた場合の Latency を評価した。図 5 はその結果である。図 5 は縦軸が CPD、横軸が Latency となっている。

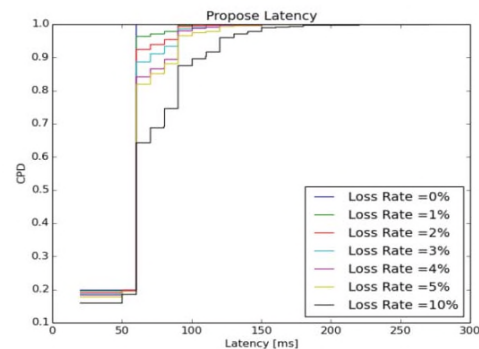


図 5 提案方式の遅延特性

5.1 節で述べたように Data パケットの約 20% が経路上のルータにキャッシュされており、残り Data パケットは Publisher に存在する。そのため、packet loss が 0% の状態では、全体の約 20% の Data パケットが 20ms の Latency で取得され、残り Data パケットが 60ms の Latency で取得される。packet loss rate が高くなるにつれ、Data パケット取得の Latency は大きくなるが、提案手法では、packet loss rate が 10% の場合でも概ね 200ms 以下で取得できている。これは 2.3 節において述べた 500ms 以下で packet loss を回復するという要求を満たしている。

続いて Consumer-Only モデルの評価を行った。図 6 は packet loss rate を 10% に固定し、RTO の値を変化させて取得した特性をまとめたものである。比較のため PITinfo ベースの再送制御の packet loss 10% のときの特性を付す(以降図中に Propose として記す)。

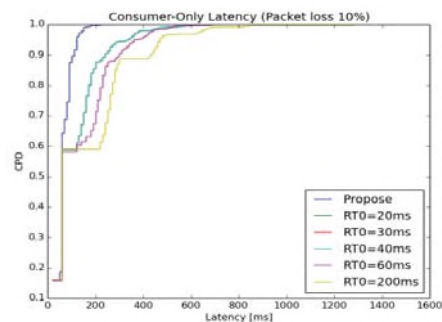


図 6 提案方式と Consumer-Only モデルの遅延特性の比較



Consumer-Only モデルは、RTO が 40ms の場合と 20ms・30ms の場合でほとんど Latency が変わらない。このことからわかるように、Consumer-Only モデルの再送制御では、Interest パケットの再送最も効率的な場所で行うことができないため、RTO をいくら短くしても Latency をある一定以下に抑えることができない。

このように、Consumer-Only の再送方法はパケットロス発生時の Latency 低減に原理的な限界があるため、遅延に敏感なユースケースには不向きである。

続いて Router-Base モデルの Latency の評価を行った。パケットロスを 10%にした際の結果を図 7 に示す。図 7 にも比較のため PITinfo ベースの再送方式のパケットロス 10%時における特性を付す。

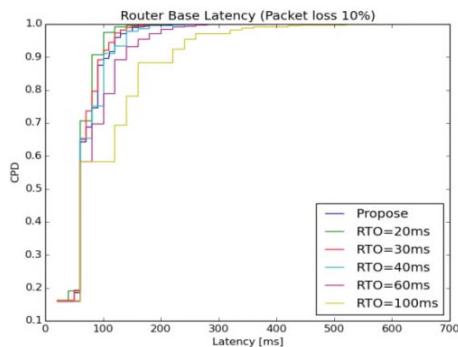


図 7 提案方式と Router-Base モデルの遅延特性の比較

Router-Base モデルでは Consumer-Only モデルと異なり、RTO の値を小さくすることで提案手法と同程度まで Latency を引き下げることができる。今回のトポロジおよびシナリオでは、RTO を 30ms にすると、提案手法とほぼ同等の特性を得ることができている。さらに RTO を 20ms まで小さくすると PITinfo ベースの手法よりもわずかに良い特性を得られる。しかし実際には、3 章でも述べたように、各ネットワークにおいて最適な RTO 値を見つけ出すことは困難である。

### 5.3.2. 再送要求によるパケット数の増加

Consumer やルータに再送制御を実装することにより、再送を行わない場合に比べてネットワーク内で送信される Interest パケットの数が増大することが予想される。

本節ではパケットロス率を 0~10%にした場合に Router-Base モデルと PITinfo ベースの方式で送信 Interest パケット数がどのように増加するかについて評価した。評価の指標としては、すべての Consumer が受け取った合計の Data パケット数で、ネットワーク内の Consumer やルータを含むすべ

てのノードが送信した Interest パケット（再送を含む）の数を除した指標（以下 Interest/Data Ratio と記す）を用いた。この Interest/Data Ratio は値が小さいほど、少ない Interest パケットで Data パケットを取得可能なことを示しているため、効率が良いといえる。結果を図 8 に示す。

Router-Base モデルは、RTO の値を小さくすれば Interest/Data Ratio が大きくなる。PITinfo ベースの方式と同等の遅延特性を示す RTO=30ms の場合、パケットロス率が 0%の場合で 3.8、10%の場合で 4.3 の Interest/Data Ratio となっている。

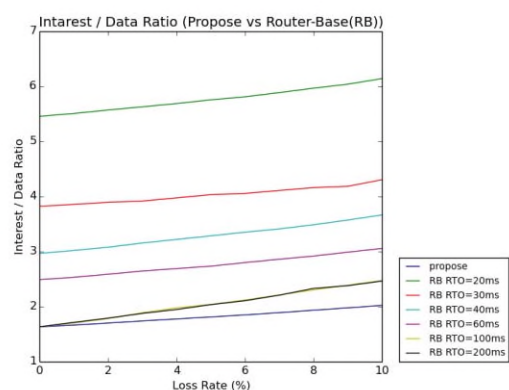


図 8 提案方式と Router-Base モデルの Interest/Data 特性の比較

また RTO=20ms の場合ではパケットロス率が 0%の場合で 5.4、10%の場合で 6.2 の Interest/Data Ratio となっている。逆にある一定の RTO 以上の領域では、RTO の値を大きくしても

Interest/DataRatio が小さくなることがない。今回の評価では、RTO を 100ms よりも大きな値とした場合に、指標への影響はほとんどなくなった。

一方、PITinfo ベースの方式ではどのパケットロス率においても、Router-Base モデルよりも小さな Interest/Data Ratio を実現している。パケットの遅延特性が同等の RTO=30ms のモデルと比較すると、Interest/Data Ratio を全領域で約 1/3 に抑えることができている。これはネットワークを流れるパケットのバイト数（以下通信量と記す）に換算すると、Interest パケットのサイズが Data パケットの 10%であると仮定した場合で 20%程度の通信量の削減にあたる。

以上の結果と 5.3.1 節の結果から単純なルータでの再送モデルでは最適な RTO の値を決定することは困難であり、提案手法と同等の特性を実現する RTO 値を用いた場合でも多量の Interest パケットを送信することがわかる。一方、提案手法では送信される Interest パケットの総量を大幅に抑つつ良好な遅延特性を得られるため、ネットワークの状況の悪化やパケット処理による負荷を増やすことなく、効率的な再送を実現できる。

#### 5.4. PIT 情報の伝送によるオーバーヘッド

PITinfo ベースの再送制御では、Data パケットに PIT 情報を付加するため Data パケット転送時にオーバーヘッドが発生する。ここではオーバーヘッドを検証するため、各リンクの片方向遅延が 5ms~50ms に変化させた場合に 1つの Data パケットに付与される PIT 情報に含まれるエントリの平均の個数を調査した。結果が図 9 である。

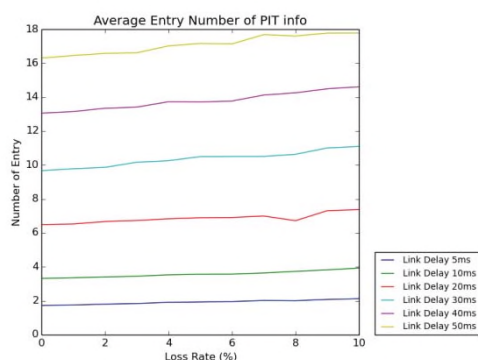


図 9 提案方式における平均 PIT エントリ数

図 9 によるとシミュレーションで用いた各リンクの片方向遅延が 10ms の状況では Data パケットに含まれる PIT 情報のエントリ数は平均 4 個弱である。また各リンクの遅延が 50ms と非常に大きい場合で平均 17 個程度である。PIT 情報が Data パケット内で占めるサイズは、Content Name が長くなった場合や Data パケットのペイロードが小さくなった場合により大きな値となる。しかし映像コンテンツの場合、一つの Data パケットのサイズはネットワークの MTU(Maximum Transfer Unit) と同等の値となることが多い。ここで Data パケットのサイズが 1400byte であるとし、名前のエントリが 1 エントリあたり 8byte であったとすると、リンクごとの遅延が 10ms の場合のオーバーヘッドは約 2.3% 程度、50ms の場合で 9.7% と計算できる。このように PITinfo ベースの再送制御方式にはある程度の通信オーバーヘッドが存在するが、5.3.2 節で述べた不要な Interest の抑制による 20% の通信量削減に比べて十分に小さい。またパケットロス時の遅延発生を抑えられる利点もあるため、提案方式はオーバーヘッドのコストを上回るメリットをもたらしていると考えられる。

#### 6. まとめ

本論文では、NDN 上でパケットロスが発生した際に、データ到着の遅れを最低限に抑える効率的な再送方式として PITinfo ベースの再送制御方式を提案した。提案方式は上流ノードから通知される PIT 情報を使用して正確にパケットロスを検出し、無駄な Interest パケットや Data パケットの送

信を発生させない。シミュレーション評価の結果、従来手法の 1/3 程度の Interest パケットの再送によって、パケットロスの発生時にも最低限の遅延で Data パケットを受信できることを確認した。

今後は、各ルータにおいてコンテンツのフローを識別することにより、より効率的な PITinfo の送信の実現や、実機実装による実用性の検証を行いたい。

#### 謝辞

本論文の執筆に当たり、UCLA の Lixia Zhang 教授からは、様々なご助言を頂いた。またシミュレーション環境の構築や提案方式の実装に当たっては ndnSIM の開発チームからのサポートを頂いた。感謝申し上げたい。

#### 参考文献

- [1] Jacobson, Van, et al. "Networking named content." *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. ACM, 2009.
- [2] <https://irtf.org/icnrg>
- [3] <http://www.itu.int/en/ITU-T/studygroups/2013-2/016/13/Pages/default.aspx>
- [4] <https://tools.ietf.org/html/draft-irtf-icnrg-videos-streaming-03>
- [5] Burke, Jeff. "Video Streaming over Named Data Networking." *E-LETTER*. [http://named-data.net/wp-content/uploads/2013/07/E-Letter-July13\\_page6.pdf](http://named-data.net/wp-content/uploads/2013/07/E-Letter-July13_page6.pdf)
- [6] Carofiglio, G., Gallo, M., & Muscariello, L. (2012, March). ICP: Design and evaluation of an interest control protocol for content-centric networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on* (pp. 304-309). IEEE.
- [7] Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B., & Zhang, L. (2013). A case for stateful forwarding plane. *Computer Communications*, 36(7), 779-791.
- [8] G.-W. Jung, W.-S. Jung, Y. B. Ko, J. H. Kim, and J.-H. Park, "An efficient name-based loss recovery for wireless content centric networking," in *Proceedings of the IEEE Symposium on Computers and Communication (ISCC '14)*, pp. 1-6, Funchal, Madeira, Portugal, June 2014.
- [9] Yi, C., Abraham, J., Afanasyev, A., Wang, L., Zhang, B., & Zhang, L. (2014, September). On the role of routing in Named Data Networking. In *Proceedings of the 1st international conference on Information-centric networking* (pp. 27-36). ACM.
- [10] Wang, Y., Rozhnova, N., Narayanan, A., Oran, D., & Rhee, I. (2013). An improved hop-by-hop interest shaper for congestion control in named data networking. *ACM SIGCOMM Computer Communication Review*, 43(4), 55-60.
- [11] Yi, C., Afanasyev, A., Wang, L., Zhang, B., & Zhang, L. (2012). Adaptive forwarding in named data networking. *ACM SIGCOMM computer communication review*, 42(3), 62-67.
- [12] <http://ndnsim.net/2.0/>