

# 並列パスアクティブ測定によるパケット廃棄率推定

村井 啓之介<sup>†</sup> 渡部 康平<sup>††</sup> 中川 健治<sup>††</sup>

<sup>†</sup> <sup>††</sup> 長岡技術科学大学 大学院工学研究科 〒940-2188 新潟県長岡市上富岡町 1603-1

E-mail: <sup>†</sup>nmurai@kashiwa.nagaokaut.ac.jp, <sup>††</sup>{k\_watabe,nakagawa}@vos.nagaokaut.ac.jp

あらまし 本稿では、ネットワーク上の複数のパスに対して並列に end-to-end のアクティブ測定を行い、パケット廃棄率を推定する手法を提案する。多くの測定場面では、複数のプローブフローにより複数のパスが並列に測定される。しかし、1 つのパス上の end-to-end メトリックを測定する場合、従来の方法ではパスに沿った 1 つのプローブフローの情報しか利用されない。一般に並列測定ではプローブフロー同士で共通なパスが存在する。提案法では、共通パスにおいて同一の輻輳を経験したプローブフロー同士で廃棄パケット情報を相互に利用し、パケット廃棄率の推定を行う。また、廃棄パケット情報を共有可能なフローをツリー状に探索することで、すべてのプローブフローの情報を活用する。NS-3 を用いたシミュレーションを行い、提案法と従来法によりパケット廃棄率を推定し RMSE を計算した。提案法は、従来法よりも RMSE を平均 57% 低減させることを確認した。

キーワード アクティブ測定, パケット廃棄率, 並列測定, バースト遅延

## Estimation of Packet Drop Rate by Parallel Path Active Measurement

Norinosuke MURAI<sup>†</sup>, Kohei WATABE<sup>††</sup>, and Kenji NAKAGAWA<sup>††</sup>

<sup>†</sup> <sup>††</sup>Graduate School of Engineering, Nagaoka University of Technology,  
Kamitomiokamachi 1603-1, Nagaoka, Niigata, 940-2188 Japan

E-mail: <sup>†</sup>nmurai@kashiwa.nagaokaut.ac.jp, <sup>††</sup>{k\_watabe,nakagawa}@vos.nagaokaut.ac.jp

**Abstract** In this paper, we propose a method to estimate packet drop rate by performing end-to-end multi-path active measurement on the network. In many measurement situations, multiple paths are measured in parallel with multiple probe flows. However, information of only one probe flow of the multiple probe flows is utilized to measure an end-to-end metric on a path in conventional active measurement. Generally, the path of the probe flow has a common path in the parallel measurement. In the proposed method, the packet drop rate is estimated by using the drop packet information mutually among probe flows that experienced the same congestion in the common path. Simulation using NS-3 was performed, and packet drop rate and RMSE were calculated by the proposed method and conventional method. The proposed method reduces the RMSE on average by comparison with the conventional method.

**Key words** Active Measurement, Packet Drop Rate, Parallel Measurement, Burst Delay

### 1. はじめに

ネットワークサービスが提供される際には、インターネットサービスプロバイダ (Internet Service Provider; ISP) などのサービス提供者とユーザの間で、サービス品質に関する契約 (Service Level Agreements; SLAs) が結ばれる。ISP にとってネットワークの特性を正確に把握することは、サービスの信頼度の確保及びユーザの定着化を図る上で重要であり、SLAs 遵守のためには end-to-end メトリックに対する正確な測定手段が必要とされる。

end-to-end メトリック測定の一般的な手法として、プロー

ブパケットをネットワークへ送信し測定を行うアクティブ測定がある。アクティブ測定では計測精度とプローブレートとの間にトレードオフが存在するため、様々なメトリックをプローブレートや計測パスを増やさず詳細に測ることが重要である。

アクティブ測定では、プローブパケットの送出時刻と到着時刻の差を求め、ネットワーク内部で生じた遅延を得る。ネットワーク内部で生ずる遅延は、主に伝搬遅延・転送遅延・処理遅延・キューイング遅延の 4 つに分類される。伝搬遅延・転送遅延・処理遅延はネットワークのトラヒック量によらずパスや機器によってほぼ一定の遅延であるが、キューイング遅延はトラヒック量に依存して変化する。ノードの処理能力を超える量の

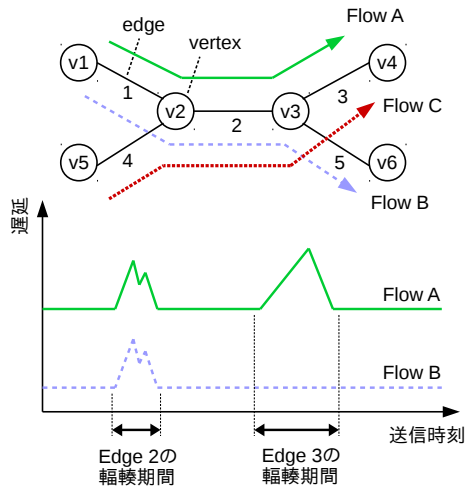


図 1 複数のプローブフローの測定における遅延

パケットがごく短時間の内に到着すると、バースト遅延と呼ばれる瞬間的に大きなキューイング遅延が生じ、更にパケットが到着してキューが溢れるとパケット廃棄が生じる。パケット廃棄はリアルタイムアプリケーションの品質に悪影響を及ぼすことが知られている。アクティブ測定では、測定期間中の廃棄されたプローブパケット数を測ることでネットワークの安定性や信頼性の指標となるパケット廃棄率が得られる。測定期間中に送信した全プローブパケット  $P$  個のうち廃棄されたプローブパケットが  $N$  個だとすれば、パケット廃棄率は  $N/P$  によって求められる。アクティブ測定を用いてパケット廃棄のような発生が稀な事象を高精度に測定する際には、高いプローブレートが要求されるため、ネットワーク負荷が増加する。

ISP がアクティブ測定を用いて計測する際には、図 1 のような複数のプローブフローによって複数のパスを並列して同時間帯に測定することが多い。しかし、1 つのパスに対する廃棄率測定には、そのパスに沿って流れる 1 つのプローブフローから得られた情報しか活用していない。

我々はこれまで、ネットワーク内部で特定のキューに複数のフローが通る場合、各フローは同様のキューイング遅延を経験する性質を利用して、プローブレートを上げることなくフローにおける遅延時系列を高精度に推定する手法を提案してきた [1]。図 1 のように並列にアクティブ測定を行なっている場合、中央の Edge 2 で遅延が生じた際には Flow A, B, C それぞれのプローブパケットは同程度の遅延を経験する。したがって、Flow A における測定結果は Flow B, C の遅延及び廃棄情報を補助的に利用することで精度向上を図ることが可能となる。Flow B, C においても同様である。

本稿では、この手法をベースに、利用可能なすべてのパスのプローブ情報を活用する方法を提案すると共に、パケット廃棄率推定に拡張する方法を示す。提案法では、各プローブフローで観測されたパケット廃棄に関する情報を、遅延時系列の情報を利用して相互変換する。プローブフロー間におけるサンプルの変換を繰り返すことで、パケット廃棄が発生しているキューを通るすべてのプローブフローに関する情報を利用可能にし、

プローブレートを上げることなく、高精度なパケット廃棄率推定を実現する。

本稿の構成は以下の通りである。まず、第 2 章において、本研究におけるネットワークモデルについて説明する。次に、第 3 章では、先行研究によって提案された手法の概要を述べ、第 4 章において、提案法の具体的手順について説明を行う。第 5 章では、シミュレーションを用いた提案法の評価をする。最後に、第 6 章で本研究のまとめと、今後の研究の方向性について述べる。

## 2. ネットワークモデルと仮定

本研究で扱うネットワークは有向グラフで表される。有向グラフの edge は、物理的/仮想的なリンクとリンクの両端のネットワークインタフェースを表す。なお、ネットワークインタフェースには入出力パケットキューが含まれる。vertex は、フォワーディング処理などネットワークインタフェース以外の部分を表す。パスは、vertex と edge の列として定義される。パケットは、送信元から宛先までパスに沿って配送される。パスは頻繁に変更されないため、パスは測定期間 (通常数分以内) で安定しているとする。

パケットは、主にパス上の edge で遅延と廃棄を経験する。現代のネットワークでは伝搬遅延とキューイング遅延の 2 つが支配的であることから、推定する遅延の原因は伝搬遅延とキューイング遅延を想定している [2]。キューイング遅延はある程度短い期間内で発生・終息し、非発生期間よりも短期間、つまり輻輳リンクが疎であると仮定する。パケット廃棄の発生は、主にトラヒックの集中によりネットワークインタフェースに割り当てられたキューが溢れることにより発生するとする。したがって、キューイング遅延とパケット廃棄は高い相関を持つ。また、通常のネットワークでは緊急時等を想定し、特定のパケットを優先的に処理するプロトコルが組み込まれていることがあるが、モデル簡略化の観点からこれを考慮しないものとする。測定技術の適用領域の広さから、本稿では、パケット廃棄率推定のためにネットワークトポロジの情報を使わずに推定を行う。

## 3. 複数パスの並列測定による遅延計測技術

本稿で提案する廃棄率推定手法は、並列フローのアクティブ測定を利用した遅延推定技術 [1] をベースとしている。文献 [1] で提案された遅延推定技術では、複数フローでアクティブ測定を並行して行うことを前提に、各フローで観測された遅延の情報を互いに変換することにより、高精度化を図っている。本章では、文献 [1] の遅延推定技術の概要について述べる。

### 3.1 キューイング遅延過程の重複

共通の送信元と edge を持つ複数のパス上のキューイング遅延過程  $\hat{\chi}_A(t)$  は、輻輳発生期間において互いに一致する場合がある。Flow A のプローブパケットが経験する遅延は、Flow A のパスに送信された仮想パケットによって経験する仮想遅延過程  $\chi_A(t)$  のサンプルとみなすことができ、サンプルは  $X_A = \{(t_A^i, x_A^i); i = 1, \dots, m\}$  のように集合として表現することができる。ここで、 $t_A^i, x_A^i$  は、Flow A の  $i$  番目のプロー

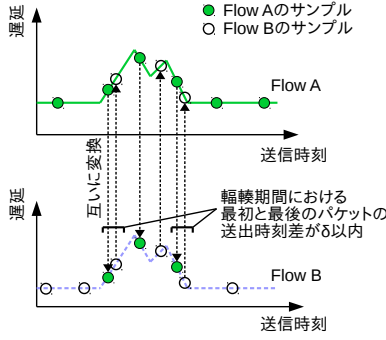


図2 2つのフローで同時に開始・終了した輻輳期間のサンプルの変換

ブパケットの送信時刻およびそのプローブパケットによって測定された遅延である。  $x_A^i$  は  $\chi_A(t_A^i)$  に対応する。仮想遅延過程  $\chi_A(t)$  は伝搬遅延  $\bar{\chi}_A(t)$  とキューイング遅延  $\hat{\chi}_A(t)$  の和である。ここで、輻輳期間を、大きな遅延がその期間に含まれ、キューイング遅延がゼロでない期間と定義すると、輻輳リンクが疎であるとの仮定から、輻輳期間中のキューイング遅延の増加は、高い確率で単一の edge の輻輳のみに由来する。このとき、図1に示すように、共通の送信元と輻輳原因となる edge を持つパス上のキューイング遅延  $\hat{\chi}_A(t)$  は一致する。

### 3.2 遅延サンプルの変換

複数のプローブフローが経験したキューイング遅延過程が一致しているとき、プローブパケットによる遅延のサンプルは互いに交換することができる。文献[1]の手法では、遅延サンプルの変換を実現するために、まず、伝搬遅延と輻輳期間をサンプルから推定する。伝搬遅延  $\bar{\chi}_A(t)$  については、伝搬遅延が一定であると仮定して、Flow A のパス上の伝搬遅延を最小値  $\bar{x}_A \equiv \min_{1 \leq i \leq m} x_A^i$  で推定することができる。一方、Flow A の  $j$  番目の輻輳期間は、  $\bar{x}_A + x_{th}$  より大きい  $x_A^i$  の  $j$  番目の連続したサンプルの集合として推定される。ここで、閾値  $x_{th}$  は提案法の制御パラメータである。次に、各フローの輻輳期間の開始・終了時刻を比較し、図2にあるように開始・終了時刻が共に  $\delta$  以内に入る輻輳期間のサンプルを互いに交換する。Flow B の  $k$  番目の輻輳期間から A へ変換を行うとすると、  $X_{B,k}$  の各サンプル  $(t_B^i, x_B^i)$  が、  $(t_B^i, x_B^i - \bar{x}_B + \bar{x}_A)$  へ変換され、Flow A のサンプルに統合される。

上記で議論した仮想遅延過程  $\chi_A(t)$  は、時刻  $t$  に送信されたプローブパケットが経験する遅延の過程であったが、同様の議論は時刻  $t$  に受信されたプローブパケットが経験する遅延の過程  $\psi_A(t)$  についても展開できる。受信時刻を基準とした仮想遅延過程は、  $\chi_A(t) = \psi_A(t + \chi_A(t))$  と定義され、キューイング遅延過程は  $\hat{\psi}_A(t) = \psi_A(t) - \bar{\chi}_A$  となる。  $\chi_A(t)$  と  $\psi_A(t)$  は互いに交換することが可能であるため、宛先が同じパスのサンプルも変換により統合することが可能である。

### 3.3 クラスタリングによるサンプルの除外

ある edge に由来する輻輳期間中に、別の edge に由来する輻輳が開始・終了した場合、3.2 節で示した方法では、不適切なサンプルを変換してしまう可能性がある。図1において、Edge 3 によって開始された輻輳期間が、Edge 2 によって引き起こさ

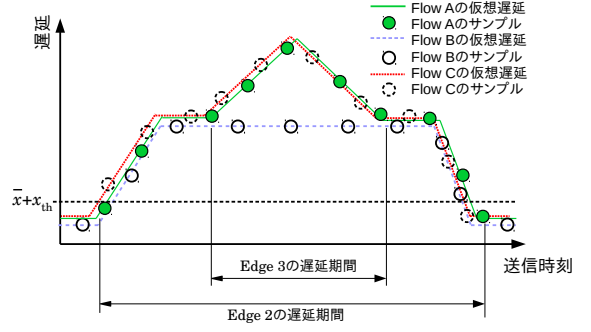


図3 輻輳した edge が複数の場合に発生する不適切なサンプルの変換

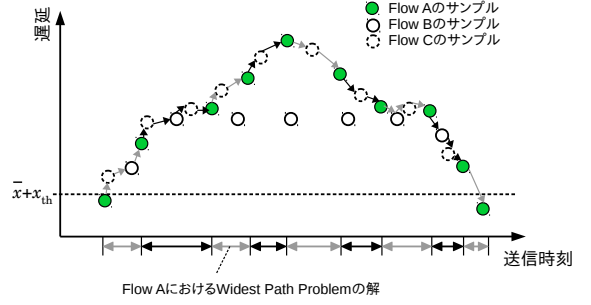


図4 WidestPathProblem の解による Flow A のすべてのサンプルを介した経路

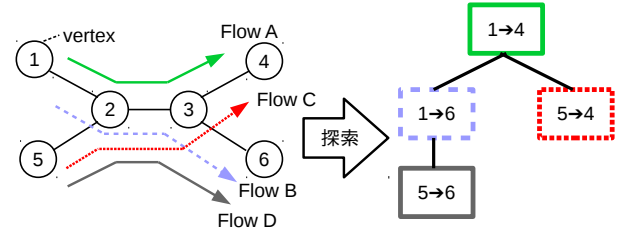


図5 送受信点異なる2つのフローが変換可能な例

れた輻輳期間中に発生している場合、図3のように輻輳の開始・終了時刻は  $\delta$  以内になっているため、3.2 節で示した方法では、サンプルが変換されてしまう。しかし、実際には、Flow A と B の仮想遅延過程は一致しておらず、Flow B のサンプルは Flow A のサンプルに変換するべきではない。一方、Flow C については、仮想遅延過程が Flow A と一致するため、Flow C についてのみ変換がなされるべきである。

文献[1]では、不適切なサンプルを除去するために、機械学習のクラスタリング技術[3]を利用する。図3の例では、Flow A および C は同一クラス内にあり、Flow B は別のクラス内に分類される。クラスタリングを適用するために、まず、図4のように、サンプル間の距離の逆数をコストに Widest Path Problem を解き、各フローのサンプルをすべて経由する経路を算出する。次に各経路を  $n$  次元ベクトルに変換して、Minimum Entropy Clustering [7] によりクラスタリングを実行する。クラスタリングの結果から、同じクラスタのフローから変換されたサンプルのみを残し、サンプルを削除する。

## 4. 提案法

文献[1]では、送信元または宛先が同じプローブフロー同士を

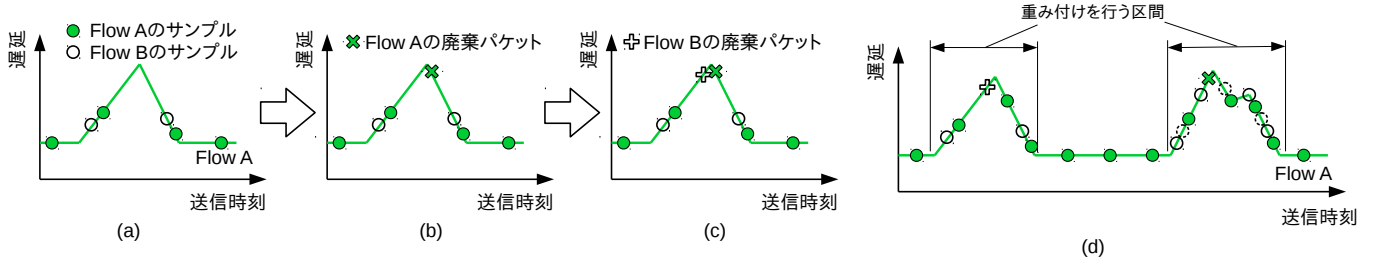


図 6 (a),(b),(c) サンプルへの廃棄パケット追加プロセス. (d) サンプルへの重み付けを行う対象となる区間

互いに変換することで遅延時系列を高精度に推定する手法を提案しているが、本稿では、この手法をすべてのプローブフローの情報を活用可能にすると共に、パケット廃棄率の推定に拡張する。送信元も宛先も異なるプローブフローの情報もすべて活用することにより、非常に低い確率で発生するパケット廃棄を的確に観測可能にしている。

#### 4.1 送信元と宛先が共に異なるプローブフローによるサンプルの変換手法

本稿で提案する手法では、各プローブフローから得られたサンプルを繰り返し変換することにより、送信元も宛先も異なるプローブフローの情報を活用可能にする。たとえ送信元も宛先も異なるプローブフローであっても、パスの一部が共有されていれば、元のプローブフローが経験したキューイング遅延の情報を含んでいる。ここで、図 5(左) のような 4 つのフローが存在する場合を考える。Flow A に注目すると Flow B や C は送信元または宛先が同じであるため、文献 [1] の手法を用いて、Flow B から Flow A にサンプルを変換することが可能である。一方で、Flow D は Flow A と送信元と宛先が共に異なるため、サンプルの変換は互いにできない。しかし、Flow A と D はパスの一部を共有しているため、同じキューイング遅延を経験している可能性があり、2 つのフローで互いにサンプルを変換することで、さらなる精度向上が実現可能である。

送信元も宛先も異なるプローブフローによるサンプルの変換を可能にするために、各プローブフローの各輻輳期間において変換可能なプローブフローを探索する手法を考える。3.2 節では送信元または宛先が同じであれば、互いにサンプルの変換を行うことができる。そこで、サンプルが変換可能なプローブフローを発見した場合、直ちに交換は行わず、その変換可能なプローブフローのサンプルに対して変換可能な別のプローブフローの探索を行う。このようにすることで、互いに変換可能なプローブフローを連鎖的に探索し、変換が繰り返され、複数のプローブフローを経由して、送信元も宛先も異なるプローブフローのサンプルを変換することが可能になる。例えば、図 5(左) のように、Flow A と B は送信元が同じであり、Flow A の  $i$  番目の輻輳期間のサンプル  $X_{A,i}$  と Flow B の  $j$  番目の輻輳期間のサンプル  $X_{B,j}$  は互いに変換可能であるとする。一方、Flow B は、Flow D と宛先が同じであり、Flow B のサンプル  $X_{B,j}$  と Flow D の  $k$  番目の輻輳期間のサンプル  $X_{D,k}$  が互いに変換できるとする。この場合、まず、Flow D のサンプルを Flow B に変換すると、Flow B のサンプル  $X_{B,j}$  に Flow D のサンプ

ル  $X_{D,k}$  が追加され、Flow B の  $j$  番目の輻輳期間のサンプルが  $X_{B,j} \cup X_{D,k}$  となる。次に、Flow B を Flow A に変換すると、Flow A の  $i$  番目の輻輳期間のサンプル  $X_{A,i}$  に Flow B の  $j$  番目の輻輳期間のサンプル  $X_{B,j} \cup X_{D,k}$  が追加されることになり、Flow B を経由することで、Flow D のサンプルの情報が Flow A の計測に活用可能になる。

すべてのプローブフローのすべての輻輳期間に対して、それぞれの輻輳期間のサンプルに変換可能な他プローブフローの輻輳期間のサンプルが存在するかどうか探索し、相互変換可能なサンプルの連鎖を表すツリーを生成する。この探索では、まず対象の輻輳期間のサンプルに対して、変換可能な他フローの輻輳期間のサンプルを探索する。探索により発見した輻輳期間に到達フラグがない場合は、発見した輻輳期間に到達フラグを立て、ツリーの葉を伸ばした上で、発見した輻輳期間を対象の輻輳期間として再度探索する。発見した輻輳に到達フラグがある場合は何も行わない。こうして到達フラグのない輻輳期間が発見されなくなるまで探索を行う。以上の手順により、対象の輻輳期間に対して変換可能なフローのツリーが作成される。例えば図 5(左) のネットワークに対して探索を行うと、図 5(右) のツリーが作成される。ツリーの各ノードは、どのフローのどの輻輳期間が親ノードに対して変換可能であるかの情報を持つ。このツリーにおいて、ノードの深さが深いフローの輻輳期間から順に変換を行っていくことで、すべてのフローに関する変換可能なサンプルの情報を活用することができる。

#### 4.2 パケット廃棄率推定

文献 [1] の手法では、プローブフローの遅延に関する情報を基に、サンプルを相互に変換することで、遅延時系列の高精度な推定を実現していたが、本稿で提案する手法では、遅延の情報を基に変換を行いつつ、パケット廃棄率の推定精度を高精度化する。特定のキューに対して複数のフローが流入する場合、各フローは同様のキューイング遅延を経験し、通過フロー全てが同様の確率で廃棄事象を経験する。プローブフローのサンプルを変換する際に、変換を行うプローブフローが経験した廃棄パケットの情報を含めることにより、廃棄パケットの観測数を擬似的に増加させることが可能である。

提案手法では、プローブフローにより観測された遅延のサンプルに、パケット廃棄を観測したサンプルを加える。前述したとおり、本稿で想定するパケット廃棄は、トラヒックの集中によるキュー溢れであり、輻輳が発生し遅延が増大した場合のみが発生する。送信したプローブパケットと受信したプローブパ



ケットのシーケンス番号を照合し、受信プローブパケット側に存在していないものを探索することで検出することが可能である。文献[1]の手法では、Flow A の  $j$  番目の輻輳期間に含まれる遅延に関するサンプルを  $X_{A,j}$  と定義していた。提案法では、これに紐付けて、Flow A の  $j$  番目の輻輳期間内に送信されたプローブパケットのうち、廃棄されたパケットの個数  $P_{A,j}^{\text{org}}$  を廃棄サンプル数として記録する。提案法では、以下に示す手順により、プローブフローの各輻輳期間のサンプルに紐付いた廃棄サンプル数を追加していく。まず、文献[1]のアルゴリズムに基づいて、プローブフローの各輻輳期間ごとに変換可能なプローブフローのサンプルを変換する。ここで、変換先を Flow A の  $j$  番目の輻輳期間とし、変換元を Flow B の  $k$  番目の輻輳期間とすると、変換時に、変換先に紐付いた廃棄サンプル数  $P_{A,j}^{\text{org}}$  に変換元に紐付いた廃棄サンプル数  $P_{B,k}^{\text{org}}$  を加算する。図 6(a) から (c) の例では、Flow A のサンプル  $X_A$  に対して、Flow B で生じた廃棄サンプルを追加することで、観測された廃棄サンプル数が 2 つに増加している。以上の手順を経て、図 6(c) のように他のフローで生じた廃棄サンプルを含むことが可能となる。このような変換を行うと、パケット廃棄率が非常に低く、かつ、プローブパケットの送信間隔が大きい場合に、たとえ自フローのプローブパケットが廃棄事象を経験できなかったとしても、他フローのプローブパケットが廃棄事象を経験していれば、廃棄率を正しく推定することができるようになる。

相互変換可能なフロー数は輻輳ごとに差異があることから、輻輳ごとに変換によって増加するサンプル数に偏りが生じてしまう。そのため、変換後の全サンプル数と廃棄サンプル数の比からパケット廃棄率を推定すると真値から大きく離れた値になってしまう。パケット廃棄率を推定する際には偏りを補正するため、図 6(d) のような輻輳期間内のサンプルに対して、サンプル数が変換前と同じになるように重み付けを行う。重み付け後の Flow A における  $j$  番目の輻輳期間中の非廃棄サンプル数  $P_{A,j}$  と廃棄サンプル数  $D_{A,j}$  は次式のように定義される。

$$P_{A,j} = (P_{A,j}^{\text{org}} + P_{A,j}^{\text{add}}) \frac{P_{A,j}^{\text{org}}}{P_{A,j}^{\text{org}} + P_{A,j}^{\text{add}}}, \quad (1)$$

$$D_{A,j} = (D_{A,j}^{\text{org}} + D_{A,j}^{\text{add}}) \frac{P_{A,j}^{\text{org}}}{P_{A,j}^{\text{org}} + P_{A,j}^{\text{add}}}. \quad (2)$$

$P_{A,j}^{\text{org}}$  は変換前のサンプル数、 $P_{A,j}^{\text{add}}$  は変換によって加わったサンプル数を表す。また、 $D_{A,j}^{\text{org}}$  は変換前に検出された廃棄サンプル数、 $D_{A,j}^{\text{add}}$  は変換によって加わった廃棄サンプル数を表す。重み付け後、次式によりパケット廃棄率を算出する。

$$\text{DropRate} = \frac{\sum_{j'=0}^j D_{A,j'}}{P^{\text{st}} + \sum_{j'=0}^j P_{A,j'}}. \quad (3)$$

$P^{\text{st}}$  は非輻輳時のサンプル数を表す。

## 5. シミュレーション

NS-3 [8] によるシミュレーションを実行し、アクティブ測定

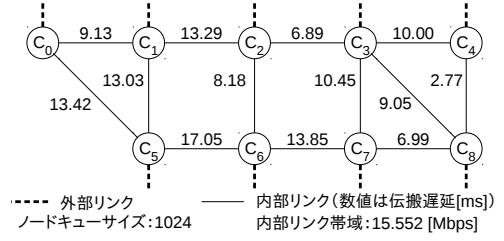


図 7 シミュレーショントポロジ

表 1 シミュレーション諸元

プローブパケット トラフィック	パケットサイズ パケット間隔	74 Byte 0.2 s
ユーザ トラフィック (定常)	パケットサイズ 送出レート パケット間隔	600 Byte 388.8 kbps 指数分布
ユーザ トラフィック (バースト)	パケットサイズ 送出レート バーストタイム アイドルタイム	500 Byte 10000 kbps 平均 1.0 s の指数分布 平均 100.0 s の指数分布

による並列フローのサンプルに含まれる廃棄サンプル数が増加し、パケット廃棄率が推定できていることを確認する。

### 5.1 シミュレーション条件

シミュレーションは図 7 に示すような Internet2 [9] を模したトポロジで行なった。内部リンクと共に記されている数値は内部リンクが持つ伝搬遅延の大きさを示し、Internet2 のノード間の距離に比例した数値となっている。

トポロジ内では、9 つのノードすべての対の間を結ぶ 72 本のパスが存在する。各パスでは表 1 に示すような 3 種類のトラフィックが流れている。プローブパケットは周期的に送信が行われるが、フロー毎で位相が異なっておりランダムである。シミュレーション時間は 1005 [s] とし、5 [s] から 1005 [s] までに送信されたパケットを用いて測定を行う。シミュレーションはプローブパケットの送信時刻の位相を変えて 10 回行う。

### 5.2 シミュレーション結果

ユーザの定常トラフィックにより得られるデータをユーザの経験したパケット廃棄率の真値として提案法の評価を行う。ツリー探索による送信元と宛先が共に異なるプローブフローによるサンプルの変換を用いた場合と、用いない場合に加えて、通常のアクティブ測定による推定値を比較した。なお、図中において、パケット廃棄率の真値が 0 となったフローは除いている。また、ユーザの経験したパケット廃棄率は最大で約  $1.7 \times 10^{-3}$  だった。

図 8 にパケット廃棄率の推定値の RMSE (Root Mean Squared Errors) を比較した結果を示す。ツリーによる変換を行なった提案法は従来法に対して、RMSE を最大 83% 低減することがフロー ID 5-4 より確認でき、低減率を平均すると RMSE を約 57% 低減することが確認された。また、ツリーによる変換を行わずとも提案法は、フロー ID 4-2 において RMSE を最大 71% 低減しており、平均して RMSE を約 31% 低減す

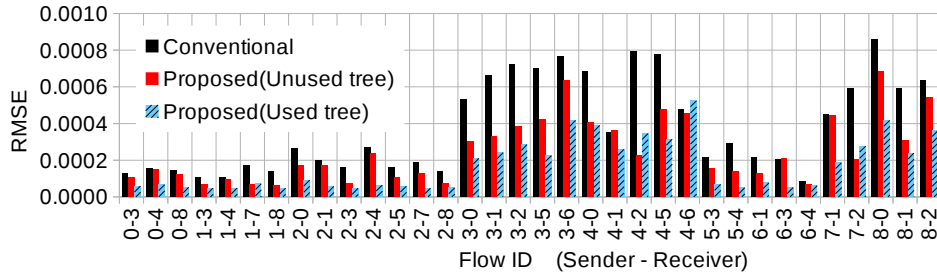


図 8 従来法と提案法によるパケット廃棄率の RMSE

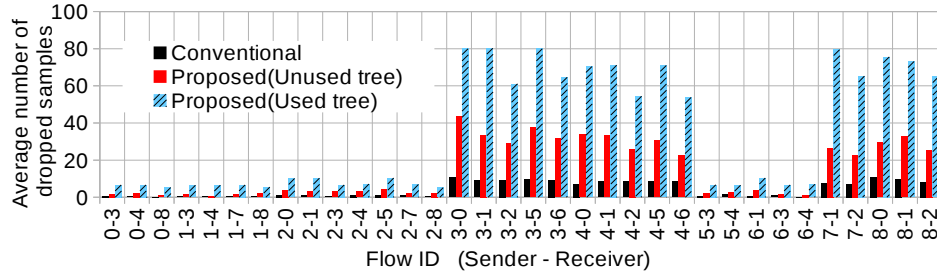


図 9 従来法と提案法が観測した廃棄パケット数の比較

る。図 9 に示すフローごとの平均廃棄サンプル数からも分かる通り、精度の向上効果は、観測された廃棄サンプルの飛躍的な増加によるものである。今回のシミュレーション中に送信されるプローブパケット数は 5000 個であるため、通常のアクティブ測定では、原理的に 0.0002 以下の確率を推定することはできない。提案手法では、廃棄サンプルを増やすことにより、この問題を克服することに成功している。

## 6. おわりに

本稿では、ネットワーク上の複数のパスに対して並列に end-to-end のアクティブ測定を行い、同一の輻輳を経験したプローブフロー同士で廃棄パケット情報を互いに共有し、共有された廃棄パケットを用いてパケット廃棄率を推定する手法を提案した。提案法では、プローブフローによる観測結果が互いに交換可能なフローをツリー状に探索することで、すべてのフローの情報を活用する。提案法の有効性を検討するために NS-3 を用いたシミュレーションを行い、パケット廃棄率の RMSE を計算した。提案法はほぼ全てのプローブフローにおいて従来法よりも高い精度でパケット廃棄率を推定し、ツリー探索により、提案法は従来法よりも RMSE を平均 57% 低減させ、また、RMSE を最大 83% 低減させることを確認した。

今後の研究では、実ネットワークにおける評価を検討している。提案法では、パケット廃棄の原因をキューイングによるキュー溢れのみとしていたが、その他の廃棄原因が推定精度にどの程度の影響を与えるか検証する必要がある。

## 謝 辞

本研究の一部は、JSPS 科研費 JP17K00008、および JP18K18035 の助成を受けたものである。

## 文 献

- [1] K. Watabe, S. Hirakawa, and K. Nakagawa. “Accurate De-

lay Measurement for Parallel Monitoring of Probe Flows”. In *Proceedings of 13th IEEE International Conference on Network and Service Management (CNSM 2017)*, pp. 1–9. IEEE, 2017.

- [2] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu. “Understanding Network Delay Changes Caused by Routing Events”. *ACM SIGMETRICS Performance Evaluation Review*, Vol. 35, No. 1, pp. 73–84, 2007.
- [3] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. “A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis”. *IEEE Transactions on Emerging Topics in Computing*, Vol. 2, No. 3, pp. 267–279, 2014.
- [4] Z. Wang and J. Crowcroft. “Bandwidth-Delay Based Routing Algorithms”. In *Proceedings of 1995 IEEE Global Telecommunications Conference (GLOBECOM 1995)*, Vol. 3, pp. 2129–2133. IEEE, 1995.
- [5] A. Hinneburg and D. Keim. “An Efficient Approach to Clustering in Large Multimedia Databases with Noise”. In *Proceedings of 4th International Conference on Knowledge Discovery and Data Mining (KDD 1998)*, Vol. 98, pp. 58–65, 1998.
- [6] G. Hamerly and C. Elkan. “Learning the k in k-means”. In *Proceedings of Advances in Neural Information Processing Systems 16 (NIPS 2003)*, pp. 281–288, 2003.
- [7] H. Li, K. Zhang, and T. Jiang. “Minimum Entropy Clustering and Applications to Gene Expression Analysis”. In *Proceedings of 2004 IEEE Computational Systems Bioinformatics Conference (CSB 2004)*, pp. 142–151. IEEE, 2004.
- [8] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena. “Network Simulations with the ns-3 Simulator”. *SIGCOMM Demonstration*, Vol. 14, No. 14, p. 527, 2008.
- [9] “Internet2 Network NOC.”. <https://globalnoc.iu.edu/i2network/>, 2019.