

## 複数プローブフローの並列測定による高精度遅延推定

平川 稔太郎<sup>†</sup> 渡部 康平<sup>††</sup> 中川 健治<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: †shirakawa@kashiwa.nagaokaut.ac.jp, ††{k\_watabe,nakagawa}@nagaokaut.ac.jp

あらまし 本稿では、並列プローブフローを用いたアクティブ測定による正確な遅延推定手法を提案する。多くの測定アプリケーションでは、複数のプローブフローにより複数のパスの遅延を並列に測定する。しかし、1つのパス上の end-to-end の遅延を測定する場合、従来の方法では1つのパスの遅延推定にはそれに沿った1つのフローの情報しか利用されていない。一般にアクティブ測定で並列に計測を行う場合、プローブフローには部分的に共通なパスが存在する。提案法では、共通なパスにおいてフローの測定結果を部分的に他のフローの測定結果に変換することで、フローの遅延を正確に推定する。提案法により観測結果が互いに適切に変換されていることを確認するため、72 並列のアクティブ測定フローのシミュレーションを行った。各フローの end-to-end の遅延の 99 パーセント分位点を推定すると、従来法に比べて約 2 倍の推定精度が得られた。

キーワード アクティブ計測, QoS, 並列測定, 遅延推定, クラスタリング

## Accurate Delay Estimation in Parallel Measurement of Multiple Probe Flows

Shintaro HIRAKAWA<sup>†</sup>, Kohei WATABE<sup>††</sup>, and Kenji NAKAGAWA<sup>††</sup>

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

E-mail: †shirakawa@kashiwa.nagaokaut.ac.jp, ††{k\_watabe,nakagawa}@nagaokaut.ac.jp

**Abstract** In this paper, we propose an accurate delay estimation method by active measurement using parallel probe flows. In many measurement applications, a plurality of probe flows measure the delay of a plurality of paths in parallel. However, when measuring the end-to-end delay on one path, in the conventional method, only the information of one flow along that path is used for delay estimation of one path. Generally, when performing measurement in parallel by active measurement, there is a partial path for the probe flows. In the proposed method, flow delays are accurately estimated by partially converting the flow measurement results into measurement results of other flows in a common path. In order to confirm that the observation results are appropriately converted by the proposed method, simulation of 72 parallel active measurement flow was performed. By estimating the 99 percentile of the end-to-end delay of each flow, proposed method estimates about twice the estimation accuracy compared with the conventional method.

**Key words** Active Measurement, QoS, Parallel Measurement, Delay Estimation, Clustering

### 1. はじめに

ネットワークの性能評価では、end-to-end の品質を正確に評価することが重要である。テレビ会議や IP 電話に代表されるリアルタイムアプリケーションは、end-to-end のパケット損失または遅延に敏感であることが知られており、ITU-T 勧告 G.114 [1] では、Voice over IP (VoIP) アプリケーションでは 150 [ms] 以上の end-to-end パケット遅延が通信品質に悪影響を与えるとしている。したがって、end-to-end の品質に対する正確な測定手法が必要である。

end-to-end 品質測定の一般的な方法として、測定のためにプローブパケットをネットワークへ送信するアクティブ測定がある。アクティブ測定を利用して、遅延 [2], [3], [4], パケット損失 [5], [6], 利用可能な帯域幅 [7], [8] を測定する様々なツールが提案されている。さらに、実際のネットワークにおける大規模測定は、ネットワークの様々な特性の理解を可能とした [9], [10]。end-to-end のアクティブ遅延測定において、多数のプローブパケットを送ることは通信オーバーヘッドや intrusiveness の問題につながるため [11] [12], プローブパケットの数を増やさず正確に測定する手法が試みられている [13], [14], [15]。しかし、現

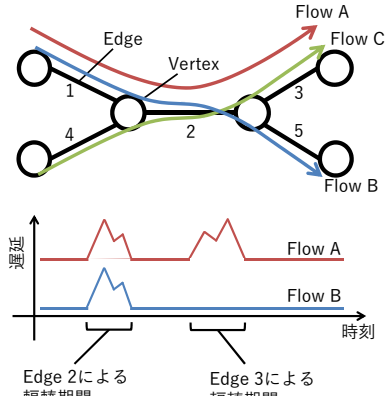


図1 複数フローの並列モニタリング。フロー A と B のパスは、Edge 1 と Edge 2 を共有する。Edge 2 によって遅延が生じた場合、同様の遅延がフロー A と B で観測される。

在のインターネットでの大きな遅延 (ITU-T 勧告 G.114 [1] で述べられているような 150 [ms] を超える遅延) は希少なイベントであるため、限られた数のプローブパケットを使用して大きな遅延を捕捉することは困難である。このため、遅延分布の高分位点の高精度測定は、未だ技術課題である。

多くの測定アプリケーションでは、複数のプローブフローで複数のパスの遅延を並列に測定してモニタリングを行うが、1つのパス上の end-to-end の遅延の測定には 1つのプローブフローだけが利用される。ネットワーク全体の特性を明確にするために、ネットワーク上の複数のパスの end-to-end の遅延を測定することは一般的であり、多くのアプリケーションでは複数のフローを並列に測定している。並列測定において、フローのパス同士は共通部分を有するため、あるフローの測定結果は別のパス上の遅延の情報を含む。図1では、フロー A と B の両方のパスに Edge 2 が含まれているため、Edge 2 が原因でフロー A に遅延が発生したとき、同様にフロー B のパケットにも遅延が発生する。したがって、フロー B に関する情報は、フロー A の遅延の推定精度を向上させるために補助的に利用できる。

本稿では、フローの測定結果に対して他のフローの測定結果を部分的に変換することにより、複数フロー並列測定による高精度遅延推定手法を提案する。提案手法では、まず各フローの測定結果から輻輳期間を取得する。次に、クラスタリング手法を利用して、遅延を引き起こす共通 edge 毎に測定結果をクラスタに分割し、クラスタ内のフローの測定結果を相互に変換する。変換では、トポロジーなど、ネットワークの内部情報を使用せず、各フローの遅延のみを使用する。提案法をシミュレーションにより評価し、提案法がプローブフローの並列測定により遅延の正確な推定を実現することを確認する。

この論文は次のように構成される。2章では、提案された測定におけるネットワークモデルといくつかの仮定について説明する。3章は、複数フロー並列測定の変換と提案法の概要をまとめる。4章では、提案法の結果を用いた遅延の推定値を説明する。5章では、シミュレーションを用いて提案法を評価する。最後に、6章で結論を述べ、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

本研究では、有線パケットネットワークにおける end-to-end 遅延の測定に着目する。ネットワークは有向グラフで表現できるとし、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インタフェースを表す。なお、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、そのインタフェース以外のネットワークデバイスの一部を表す。パスは、vertex と edge の列として定義され、パケットは、送信点から受信点までパスに沿って配送される。パスは頻繁に変更されず、パスは測定期間 (通常数分以内) で安定していると仮定する。

パケットの遅延は、パス上の vertex または edge で発生する。パケットが経験する end-to-end の遅延は、伝搬遅延、キューイング遅延、転送遅延、処理遅延の4つで構成される。処理遅延はパケットが vertex 上にあるときに発生する。他の遅延は edge で発生する。現在のインターネットでは、伝搬遅延とキューイング遅延が支配的であり、転送遅延と処理遅延は無視できる程度である [10]。本稿では、end-to-end の遅延を次のように仮定する。遅延は伝搬遅延とキューイング遅延からなり、キューイング遅延は動的に変化するが、伝搬遅延はパスに対して一定に維持される。単一 edge 上のパケットによって経験される遅延は、パケットが通過する経路に対して独立している。大規模なキューイング遅延はネットワーク内のすべての edge 間で疎であり、edge における大きなキューイング遅延を有する期間の比率は、他の期間と比較して小さい。現在のインターネットの平均リンク利用率は低く維持されており、この仮定は妥当である [16]。本稿では、輻輳 edge が高々一つであることを仮定していないことに注意する。

次に各パスの end-to-end の遅延の測定を考える。パスの遅延を測定するために、プローブパケットはネットワーク上のパスすべてまたは一部に対して定期的に送信される。送信点と受信点でのタイムスタンプの値からプローブパケットが受けた遅延が得られる。片道遅延を測定する場合、送信点デバイスと受信点デバイスの時間同期が必要である。提案法では、ネットワークトポロジーの情報を必要としない。本稿では、応用範囲の広さからトポロジー情報を前提としない手法に取り組むが、提案法はトポロジーを利用するように拡張することが可能である。トポロジー情報を用いた測定方法の開発は、今後の課題とする。

## 3. 複数フロー並列測定における変換技術

### 3.1 キューイング遅延過程の一致条件

現在のインターネットのアクティブ遅延測定では、輻輳期間の比率が他の期間より非常に小さいため、輻輳期間に関する情報を効果的にサンプリングすることが重要である。 $\chi_A(t)$  を、フロー A のパスに時刻  $t$  に送信された仮想パケットが経験する仮想遅延過程と定義する。フロー A のプローブパケットが経験する遅延は、 $\chi_A(t)$  のサンプルとみなすことができる。フロー A による  $m$  個のサンプルを、 $X_A = \{(t_A^i, x_A^i); i = 1, \dots, m\}$  で表すことにする。ここで、 $t_A^i, x_A^i$  は、フロー A の  $i$  番目のプロー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。プローブパケットは一定の間隔で送信されるため、輻輳期間内に送出されるプローブパケットの数は少ない。遅延の高分位点は、VoIPのような遅延に敏感なアプリケーションにとって重要な品質指標であるが、輻輳期間内のプローブパケットの数が少ないほど、end-to-end の遅延の高分位点の測定精度は低下してしまう。

共通の edge を持つ複数のパス上の輻輳期間内のキューイング遅延過程は、一致する可能性がある。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。輻輳期間は、大きな遅延が期間中に含まれ、キューイング遅延がゼロでない期間と定義する。以下の3つの条件が満たされるとき、キューイング遅延過程  $\hat{\chi}_A(t)$  と  $\hat{\chi}_B(t)$  は一致しているとする。

- (1) フロー A と B の2つのパスの送信点は同じである。
- (2) 図1のフロー A と B のように、パス上の送信点から最後の輻輳した edge までの経路は共通である。
- (3) 最後の輻輳した edge より後では、edge でパケットが経験するキューイング遅延は無視できる。

上記の条件 (1)–(3) が成り立つとき、 $\hat{\chi}_A(t) = \hat{\chi}_B(t)$  であるため、輻輳期間中の  $\chi_A(t) - \chi_B(t)$  は  $\bar{\chi}_A(t) - \bar{\chi}_B(t)$  に等しい。大きいキューイング遅延が発生する edge が疎であるとき、キューイング遅延過程の一致が発生する。

### 3.2 遅延過程のサンプルの変換

キューイング遅延過程  $\hat{\chi}_A(t)$  と  $\hat{\chi}_B(t)$  が一致するとき、つまり、上記3つの条件が成り立つとき、図2のようにこれらの遅延過程のサンプルを相互に変換することができる。しかし、条件 (2),(3) の成否を判断するためには、トポロジとキューの情報が必要である。我々は、代わりにプローブパケットの情報のみを使用するよう、提案法を設計する。まず、プローブパケットによって得られたサンプルから輻輳期間を検出する方法を示す。遅延が非負であり、伝播遅延が一定であることから、フロー A のパス上の伝搬遅延  $\bar{\chi}_A(t)$  を、サンプルされた遅延の最小値  $\bar{x}_A \equiv \min_{1 \leq i \leq m} x_A^i$  で推定することができる。フロー 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  $j$  番目のプローブパケット列として測定される。ここで、しきい値  $x_{th}$  は提案法の制御パラメータである。 $j$  番目の輻輳期間の開始時刻は、プローブパケット列の最初の送信時刻  $\{t_A^i; x_A^{i-1} < \bar{x}_A + x_{th} \text{ かつ } \bar{x}_A + x_{th} \leq x_A^i\}$  となる。また、同様に終了時刻は、最後の送信時刻  $\{t_A^i; \bar{x}_A + x_{th} \leq x_A^i \text{ かつ } x_A^{i+1} < \bar{x}_A + x_{th}\}$  である。

提案法では、開始時刻と終了時刻がそれぞれ等しい輻輳期間を持つフローのサンプルが相互に変換される。輻輳がネットワーク全体で最大1つである、すなわち edge の輻輳が極めて疎であると仮定するとき (この仮定は後に緩和する)、輻輳期間が同時に開始・終了するパスは、3.1 に示す条件 (2),(3) を満たす。そこで、2つのフロー A,B が以下の条件を満たす場合、A の  $j$  番目の輻輳期間内のサンプル  $X_{A,j}$  と B の  $k$  番目の輻輳期間内のサンプル  $X_{B,k}$  は相互に変換可能という。

- (i) 2つのフローの送信点は同じである。
- (ii)  $X_{A,j}$  と  $X_{B,k}$  の最初のサンプルのパケット送信時刻の差は  $\delta$  より小さい。

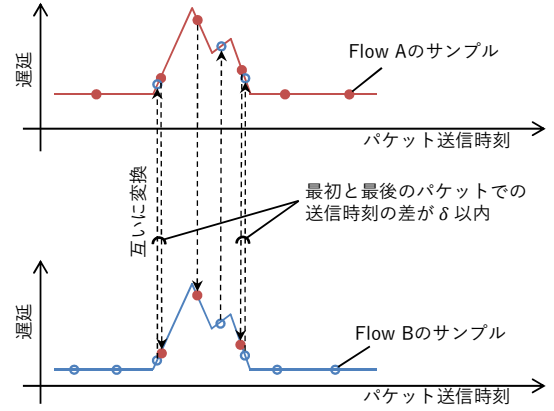


図2 2つのフローで同時に開始・終了した輻輳期間の変換

- (iii)  $X_{A,j}$  と  $X_{B,k}$  の最後のサンプルのパケット送信時刻の差は  $\delta$  より小さい。

$\delta$  はプローブパケットの送信間隔を示し、輻輳期間が開始・終了する時間を区別するために使用している。キューイング遅延が一致していても、フロー A と B の伝搬遅延が異なるため、 $X_{B,k}$  の各サンプル  $(t_B^i, x_B^i)$  は、 $(t_B^i, x_B^i - \bar{x}_B^i + \bar{x}_A^i)$  としてフロー A のサンプルに変換される。

送信点と同じフローについて議論してきたが、 $\psi_A(t)$  を、フロー A のパスに時刻  $t$  に受信された仮想パケットが経験する仮想遅延過程と定義することで、受信点と同じフローについても同様の議論が成り立つ。受信点でのサンプル  $\psi_A(t)$  は、式  $\chi_A(t) = \psi_A(t + \chi_A(t))$  によって送信時刻を用いたサンプル  $\chi_A(t)$  に変換される。

### 3.3 クラスタリングによる不適切なサンプルの除外

複数の edge が同時に輻輳する場合、3.2 で示した変換では不適切なサンプルが変換される可能性がある。開始時刻と終了時刻が同じであると予想される輻輳期間内のサンプルは3.2 に示す方法で変換される。しかし、2つのフローの輻輳期間が同時に開始/終了するかどうかを判別する3.2の条件は、キューイング遅延過程が一致する3.1の条件とは異なる。したがって、3.2 に示す条件に基づいてサンプルを変換したとしても、キューイング遅延過程は、一致しているとは限らない。

例えば、図1において、Edge 2 によって引き起こされた輻輳期間中に、Edge 3 によって開始された輻輳が発生した場合、フロー B のサンプルはフロー A のサンプルに変換してはならない (図3 参照)。この場合、フロー A と C の仮想キューイング遅延過程は一致しているため、フロー C のサンプルはフロー A に変換するべきである。一方、フロー B は Edge 3 に起因する遅延を経験していないため、フロー A と B の仮想キューイング遅延過程は一致しない。しかし、3.2 の変換では開始・終了時刻が一致すると判定して、フロー B のサンプルをフロー A のサンプルに変換してしまう。したがって、フロー B の不適切なサンプルをフロー A のサンプルから除外する必要がある。

不適切なサンプルを除外するために、機械学習のクラスタリング技術を利用する [17]。変換されたサンプルに基づいて、クラスタリング手法を使用してフローのクラスタを構築する。図3の例では、フロー A および C は同一クラスタ内にあり、フ

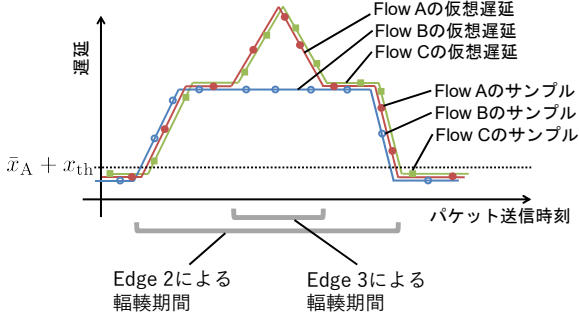


図 3 輻射した edge が複数の場合. 図 1 に示すネットワーク上で, Edge 2 によって引き起こされる輻射期間内に, Edge 3 によって引き起こされる輻射期間が開始・終了する. フロー A とフロー B の仮想遅延過程は重複しないが, フロー A とフロー C の仮想遅延過程は厳密に重複している.

ロー B は別のクラスタ内にある必要がある.

一般的なクラスタリング手法を使用するために, サンプル数とその間隔はフロー毎に異なるため, 各フローのサンプルを  $n$  次元ベクトルを構成する. フロー A の  $j$  番目の輻射期間に対してフロー B の  $k$  番目の輻射期間のサンプルを変換したときのサンプルの集合を  $X_{A,j}^{B,k}$  とする.  $F_{A,j}$  をフロー A の  $j$  番目の輻射期間に加算されるサンプル集合の集合, すなわち  $F_{A,j} = \{X_{A,j}, X_{A,j}^{B,k}, X_{A,j}^{C,l}, \dots\}$  とする.  $\mathcal{F}_{A,j}$  は,  $F_{A,j}$  におけるすべてのサンプルの集合,  $\mathcal{F}_{A,j} = \bigcup_{f \in F_{A,j}} f$  である. まず, 輻射期間毎に vertex 集合  $\{(t_f - \delta, \bar{x}_A), (t_l + \delta, \bar{x}_A)\} \cup \mathcal{F}_{A,j}$  を持つ有向グラフを構築する. ここで,  $t_f$  と  $t_l$  は  $X_{A,j}$  における最初と最後の送信時刻を表す. グラフでは, vertex  $(t_A^i, x_A^i)$  からの edge は,  $t_A^i$  より後の送信時刻を持つすべての vertex に接続されている.  $(t_A^i, x_A^i)$  から  $(t_B^j, x_B^j)$  までの edge のコストは,

$$\frac{1}{\sqrt{(\frac{\beta^2}{\delta^2}(t_{i,A} - t_{j,B})^2 + (x_{i,A} - x_{j,B})^2)}}$$

で与える. ここで,  $\beta$  は提案法のパラメータであり, 図 4 の水平軸と垂直軸の目盛の比率を調整する. フロー毎に, 最初の送信時刻を持つ vertex から, 自らのフローのすべての vertex を通る最後の送信時刻の vertex までの経路を探索する (図 4 参照). フローの vertex 間のパスは, widest path problem [18] の解である. 次に, 各フローについて vertex を等間隔に置くことで, 経路を  $n$  次元ベクトルに変換する. ベクトルの  $j$  番目の要素は, 送信時刻が経路上の  $((j-1)(t_l - t_f))/(n-1)$  である場合のキューイング遅延であり, ここで,  $n$  は元のサンプルの数  $|X_{A,j}|$  とフローの多重度  $|F_{A,j}|$  との積を表す. 以上の処理によって, 各フローのサンプルを  $n$  次元のベクトルとして表現することができる.

提案法は, 各フローのサンプルを表現する  $n$  次元ベクトルのクラスタを構築し, 異なるクラスタのフローから変換されたサンプルを除外する. クラスタリング技術に関する先行研究の文献は豊富にあるが [18], 提案法では, これらの手法のうち, 高次元ベクトルを扱うことができ, 入力パラメータとして予め定められた数のクラスタを持たない手法が利用可能である (例えば, DENCLUE [19], G-Means [20], Minimum Entropy Clustering (MEC) [21] など). 各フローのサンプル数が極端に

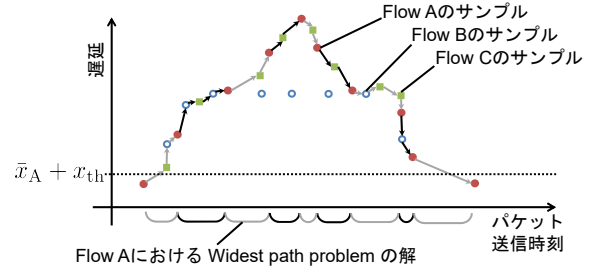


図 4 フロー A の WidestPathProblem の解. WidestPathProblem を 9 回解くことによって, 最初の点はフロー A のすべての vertex を介して最後の点までの遅延を得る.

少ない場合は, フローのクラスターを適切に分割するのは困難であるため, 輻射時のサンプル数が 1 以下であれば, 提案法で変換されたサンプルは除外される.

#### 4. end-to-end の推定量

提案法は, 仮想遅延過程のサンプル数を増加させ, これらのサンプルを end-to-end の遅延に関する様々な推定量に利用することができる. 提案法はアクティブ測定 of サンプルを単純に追加するだけなので, アクティブ測定に基づく測定技術の大部分は提案法と併用することができる. ただ, 提案法によるサンプルは, 輻射期間のみ追加されるため, 時間に対して均一に分布していないため, 多重度に応じてサンプルを重み付けする必要がある. ここでは, 平均遅延と  $q$  分位点測定の例を示す.

サンプルの重みは, サンプルを含む期間の多重度によって決定される. サンプル  $s$  の重みは, 以下のように与えられる.

$$w_s = \begin{cases} \frac{|\mathcal{F}_{A,j}|}{|F_{A,j}|} & s \in \mathcal{F}_{A,j} \quad (j = 1, 2, \dots), \\ 1 & \text{otherwise.} \end{cases}$$

3.3 で定義したように,  $F_{A,j}$  はフロー A の  $j$  番目の輻射期間に追加されたサンプル集合の集合であり,  $\mathcal{F}_{A,j} = \bigcup_{f \in F_{A,j}} f$  である.

フロー A のパス上の平均遅延の推定値は,

$$\frac{1}{|X_A|} \sum_{s \in X_A \cup \mathcal{F}_{A,*}} w_s d_s,$$

ここで,  $d_s$  と  $F_{A,*}$  は, サンプル  $s$  から得られる遅延とすべての輻射期間中のすべてのサンプル  $\bigcup_j \mathcal{F}_{A,j}$  をそれぞれ表す.  $q$  分位点測定のためには, まず以下を計算する.

$$k = \arg \max_j \left\{ \sum_{i=1}^j w_{s_i} \leq q |X_A| \right\},$$

ここで,  $s_i$  は遅延が  $i$  番目に小さいサンプルであるサンプルを表す.  $d_{s_k}$  が, end-to-end 遅延の  $q$  分位点の推定量となる.

#### 5. シミュレーションデータによる実験

NS-3 [22] シミュレーションを実施し, アクティブ測定による並列フローのサンプルが, 提案法により適切に変換されることを確認する. 図 5 に示すシミュレーションのトポロジは, Internet2 [23] のトポロジを参考としている. 図 5 のリンクの



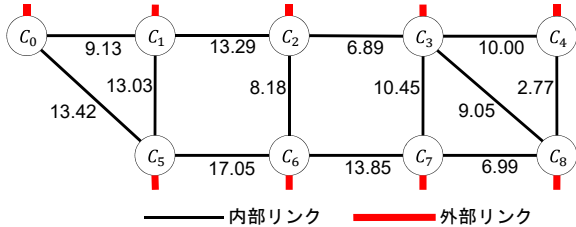


図 5 シミュレーショントポロジ. リンクの数値は伝搬遅延 ([ms]).

表 1 トラフィックの内容

定常トラフィック	パケットサイズ	600 [Byte]
	トラフィックの種類	ポアソン到着
	トラフィック量	388.8 [Kbps] (リンク帯域の 4%)
バーストラフィック	パケットサイズ	500 [Byte]
	トラフィックの種類	定期的なバーストの ON/OFF 過程
	トラフィック量	8,000 [Kbps] (バースト) 0 [bps] (アイドル)
	バースト	平均 0.1[s] の指数分布
	アイドル	平均 0.4[s] の指数分布
プローブ	パケットサイズ	74 [Byte]
	トラフィックの種類	Periodic arrivals
	パケット間隔 $\delta$	200 [ms]

横にある数値は伝搬遅延を示し, Internet2 のノード間の距離に比例するように設定した.

シミュレーションにおけるトラフィックは, 表 1 に挙げた 3 つのタイプに分類される. これらは, 9 つのノードのすべての対の間を流れる (すなわち, ネットワーク全体で 72 本のパス). プローブパケットは周期的に送信されるが, 位相はランダムとする. キュー長は十分に大きく設定されているためバッファ溢れは発生しない. シミュレーション時間は 42 [s] で, 20 [s] から 42 [s] までのデータを使用している.

提案法のパラメータは以下のように設定する. しきい値  $x_{th}$  と  $\beta$  を 0.01 [s] と 0.01 に設定する. クラスタリングには MEC を用い, radius パラメータ  $r$  とクラスタの期待数  $e$  は, それぞれ 0.001 および 10 に設定する.

図 6 に, フロー ID 5-3 について得られたサンプルを示す. 図では, 大きな遅延が観測される期間 (37.0 [s]~41.0 [s]) のみを表示している. 提案法のサンプル数が従来法より多く, サンプルは仮想遅延と一致していることが確認できる. クラスタリングにより削除されたサンプルは, 緑色の十字で示され, そのほとんどは仮想遅延と一致していない. 結果として, クラスタリングにより不適切なサンプルが除外されたことが確認できる.

次に, end-to-end 遅延の 99 パーセント分位点を測定したときの提案法の推定精度を評価する. シミュレーションは, プローブパケット送信時刻の位相を変えて, 10 回実施する. 99 パーセント分位点の真値と変換されたサンプルの数を, それぞれ図 7(上) と (中) に示す. ここでは遅延の 99 パーセント分位点が 100 [ms] を超えるフローのみを表示している. フロー ID  $s-d$  は, フローの送信元  $s$ , および宛先  $d$  から構成される. プローブパケットから得られる元のサンプル数は 110 サンプルであるが, 図 7(中) には含まれていない. 同様に, クラスタリング

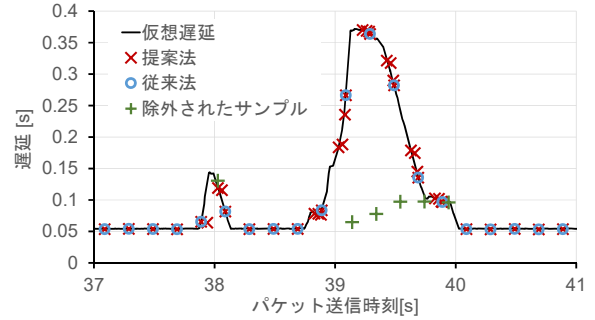


図 6 フロー ID 5-3 の提案法と従来法のサンプル. 仮想遅延にないサンプルは, クラスタリングによって適切に除外される.

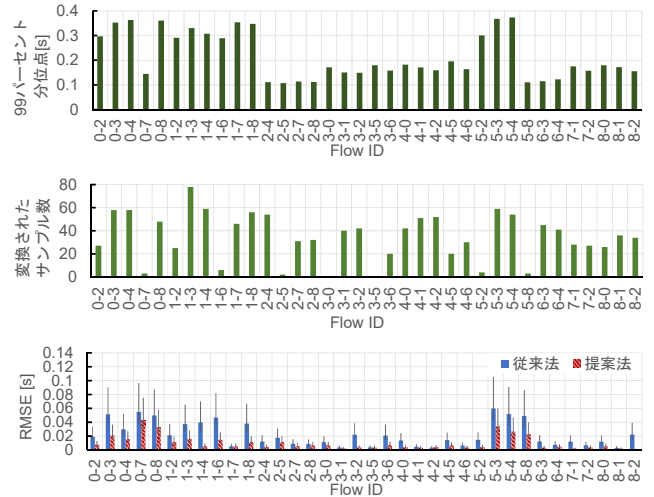


図 7 (上) 遅延の 99 パーセント分位点の真値. 遅延の 99 パーセント分位点が 100 [ms] を超えるフローのみが示されている. (中) 変換されたサンプル数. 変換されたサンプル数には, フローの元のサンプルは含まれていない. (下) 各フローの 99 パーセント分位点測定値の RMSE. エラーバーは 95 %信頼区間を表す.

処理で削除されたサンプルも含まれていない. 最大 78 サンプルがフローのサンプルに変換されており, フローの遅延が大きいほど変換サンプル数が増える傾向にある. end-to-end 遅延の 99 パーセント分位点測定値の RMSE (Root Mean Squared Errors) を計算した結果を図 7(下) に示す. エラーバーは, 95 %の信頼区間を表す. 提案法は, 従来法と比較して, 殆どのフローに対して, 同等, またはそれ以上の精度が達成されている. 提案法は, RMSE を最大 86 %減少し (フロー ID 1-4), また, RMSE の平均減少率を比較すると, 提案法がプローブパケットの数を増加させずに, 約 2 倍の精度を達成している. さらに, 全フロー中の最低 RMSE の値を比較すると, 従来法のフロー ID 5-3 に対して提案法のフロー ID 0-7 での RMSE の比率が 25 %減少している.

最後に, 提案法のパラメータ  $\beta$  に対する依存性を検証する.  $\beta$  を 0.005 から 2.56 に変更して end-to-end 遅延の RMSE を計算する. 他のパラメータは, 前述の条件から変更していない. 実験結果を図 8 に示す. 最大 RMSE 減少率は, パラメータ  $\beta$  が増加するにつれて減少率が増加し,  $\beta = 0.16$  から安定する. このグラフでは,  $\beta$  が大きい方が減少率が良いが, 0.64~2.56の間ではサンプルが不適切に変換されていた. そのため, 最低 RMSE の減少率と RMSE の中央値の減少率は, 不適切な変換

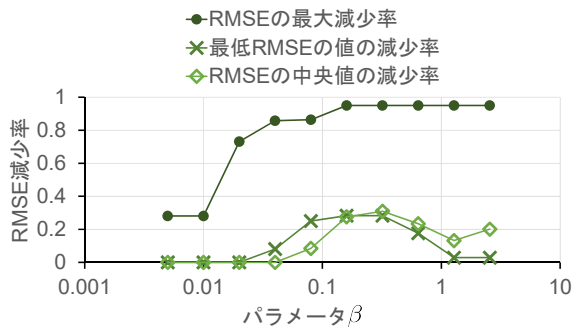


図8 パラメータ  $\beta$  に対する RMSE の依存性. 提案法の性能は, 様々な  $\beta$  で従来法と同等以上であるが, 小さな  $\beta$  では十分な推定精度の向上が得られない.

によって悪化している.

## 6. おわりに

本稿では, 複数フローの並列測定において, 測定結果を部分的に変換して正確な推定を実現する遅延推定手法を提案した. 提案法では, 他のフローで測定されたサンプルを変換することで, サンプルを追加し, クラスタリング手法を用いて不適切なサンプルを除外する. シミュレーションにより, 提案法がサンプルを適切に追加・除外できることを示した. 提案法は RMSE の平均において約 2 倍の推定精度を達成するとともに, 推定遅延の RMSE を最大 95 % まで減少し, 全フロー中の最低 RMSE 値は 28 % 低減できた.

今後の研究では, 実際のネットワークトラフィックを用いて評価する予定である. さらに, 変換のためにネットワークポロジの情報を利用する手法も考えられる. また, この手法はパケット損失測定にも拡張することができる.

## 謝 辞

本研究の一部は, 電気通信普及財団による助成を受けたものである.

## 文 献

- [1] "One-way Transmission Time," ITU-T Recommendation G.114, May 2003.
- [2] J.-C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," *Journal of High Speed Networks*, vol. 2, no. 3, pp. 289-298, 1993.
- [3] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating Latency between Arbitrary Internet End Hosts," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement (IMW 2002)*, Marseille, France, Nov. 2002, pp. 5-18.
- [4] L. De Vito, S. Rapuano, and L. Tomaciello, "One-way Delay Measurement: State of the Art," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 12, pp. 2742-2750, Dec. 2008.
- [5] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Improving Accuracy in End-to-End Packet Loss Measurement," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 157-168, Oct. 2005.
- [6] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, "Probing for Loss: the Case against Probe Trains," *IEEE Communications Letters*, vol. 15, no. 5, pp. 590-592, Mar. 2011.
- [7] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement (IMC 2003)*, Miami, FL, USA, Oct. 2003, pp. 39-44.
- [8] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proceedings of the 4th Passive and Active Measurement Conference (PAM 2003) Workshop*, San Diego, CA, USA, Apr. 2003.
- [9] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and Modelling of the Temporal Dependence in Packet Loss," in *Proceedings of the 18th IEEE International Conference on Computer Communication (INFOCOM 1999)*, New York, NY, USA, Mar. 1999, pp. 345-352.
- [10] 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, p. 73, Jun. 2007.
- [11] M. Roughan, "Fundamental Bounds on the Accuracy of Network Performance Measurements," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 253-264, Jun. 2005.
- [12] K. Watabe and K. Nakagawa, "Packet Delay Estimation that Transcends a Fundamental Accuracy Bound due to Bias in Active Measurements," to appear in *IEICE Transactions on Communications*, vol. E100-B, no. 8, Aug. 2017.
- [13] B.-Y. Choi, S. Moon, R. Cruz, Z.-L. Zhang, and C. Diot, "Quantile Sampling for Practical Delay Monitoring in Internet Backbone Networks," *Computer Networks*, vol. 51, no. 10, pp. 2701-2716, Jul. 2007.
- [14] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Accurate and Efficient SLA Compliance Monitoring," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 109-120, Oct. 2007.
- [15] F. Baccelli, S. Machiraju, D. Veitch, and J. Bolot, "On Optimal Probing for Delay and Loss Measurement," in *Proceedings of the 7th ACM Conference on Internet Measurement (IMC 2007)*, San Diego, CA, USA, Oct. 2007, pp. 291-302.
- [16] "CAIDA: The Cooperative Association for Internet Data Analysis." [Online]. Available: <http://www.caida.org/>
- [17] 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, Sep. 2014.
- [18] Z. Wang and J. Crowcroft, "Bandwidth-delay Based Routing Algorithms," in *Proceedings of 1995 IEEE Global Telecommunications Conference (GLOBECOM 1995)*, Singapore, Nov. 1995, pp. 2129-2133.
- [19] A. Hinneburg and D. Keim, "DENCLUE: 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)*, New York, NY, USA, Sep. 1998, pp. 58-65.
- [20] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proceedings of Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 2003.
- [21] 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)*, Stanford, CA, USA, Aug. 2004, pp. 142-151.
- [22] T. R. Henderson, M. Lacage, G. F. Riley, G. Dowell, and J. B. Kopena, "Network Simulations with the ns-3 Simulator," in *Proceedings of ACM SIGCOMM 2008*, Seattle, WA, USA, Aug. 2008, p. 527.
- [23] "Internet2 Network NOC." [Online]. Available: <https://globalnoc.iu.edu/i2network/>