

マルチメディアコミュニケーションの QoS を 確保するための実時間可用帯域推定方式

大芝 崇[†] 中島 一彰[†]

本稿では、エンド端末間の可用帯域を実時間で推定する方式 PathQuick を提案する。ビデオ会議やリアルタイム写真共有などのマルチメディアコミュニケーションにおいて、映像や写真などのメディアを送信する直前に、PathQuick を用いて最新の可用帯域を短時間に推定することで、メディアの送信時の消費帯域を可用帯域以下に抑える制御が可能になる。その結果、ルータのキューイング処理による遅延やパケットロスを防ぎ、リアルタイム性が確保できるため、マルチメディアコミュニケーションの QoS が確保できる。評価実験により、PathQuick の推定時間が従来比 1/4 以下の 176 ミリ秒であることを示す。また、リアルタイム性を確保するために推定時間に上限を設けた条件下において、PathQuick における探索可能な帯域の範囲が従来比 125.6 倍の広さを持つことを示す。さらに、PathQuick は従来比 4.3 倍細かい計測の分解能を持ち、計測にかかるネットワーク負荷を従来方式と同等に設定した場合でも、従来方式よりも推定精度が高いことを示す。

Real-Time End-to-End Available Bandwidth Estimation for QoS of Multimedia Communication

Takashi Oshiba[†] and Kazuaki Nakajima[†]

We propose PathQuick, a real-time end-to-end available bandwidth estimation method. When PathQuick is used in multimedia communication, such as video conferencing and real-time photo sharing, just before the transmission of media data such as video and photo, it can quickly complete the estimation of the latest available bandwidth. Consequently, the bandwidth consumption of media data can be controlled to within the available bandwidth estimated at the beginning of media data transmission, and this will prevent delay and packet loss. Thus, QoS at the beginning of real-time photo sharing and video conferencing can be ensured. Our evaluation of PathQuick has shown that its estimation duration is several hundred milliseconds, which is more than four times as fast as a conventional method. We also found that its probable range is more than a hundred times as wide as that of the conventional method when a limited waiting time is set to avoid degrading real-time responsiveness.

1. はじめに

近年、ビデオ会議[1]やリアルタイム写真共有[2]、E ラーニング[3]などの IP ネットワーク上のマルチメディアコミュニケーションが普及しつつある。映像や写真などのメディアの消費帯域が、エンド端末間の経路上の空き帯域である可用帯域[4]を超過した場合、メディアの送信時に遅延やパケットロスが発生してしまう。そのため、メディアの送信開始時の消費帯域を可用帯域以下に抑える制御を行うために、メディアを送信する直前に、最新の可用帯域を短時間に推定することが重要である。これにより、マルチメディアコミュニケーションの QoS が確保できる。

例えば、ビデオ会議において、映像の送信端末が映像の送信を開始する直前に可用帯域を推定し、推定結果に基づいて映像の初期ビットレートを動的に決定[5]できる。また、リアルタイム写真共有において、写真の送信端末が写真を送信する際に写真データを次のように動的に再圧縮することで、写真送信処理のリアルタイム性を確保できる。写真送信の直前に推定した可用帯域と、あらかじめユーザが定義した写真送信時間の許容値に基づいて写真の圧縮率を動的に決定[6]することで、許容時間以内に写真送信処理を完了できる。

しかし、従来の可用帯域推定方式[4]には、2 章で詳述するように、可用帯域の推定に数秒から数十秒の時間を要するという課題がある。したがって、メディアの送信直前に従来方式を用いると、メディアの送信開始までに長い待ち時間が発生してしまうため、従来方式はマルチメディアコミュニケーションでの利用には適さない。

本稿では、エンド端末間の可用帯域を実時間で推定する方式 PathQuick [7]を提案する。PathQuick では、各計測パケットを等間隔に配置し、パケット番号が進むに連れてパケットサイズが線形に増加する計測パケット列を送信することで、実時間での可用帯域推定を実現する。文献[7]ではシミュレーションによる評価結果を報告した。本稿では、実機による評価実験を行ったので、その評価結果も合わせて報告する。

2. 関連研究

計測パケットを送信することでエンド端末間の可用帯域を推定する、アクティブ計測と呼ばれる方式が数多く提案されている。アクティブ計測は、2 個の計測パケットをペアにして送信するパケットペア方式と 3 個以上の計測パケットで構成される計測パケット列を送信するパケットトレイン方式に大別される[4]。前者の代表例として Abing [8]と Spruce [9]があり、後者の代表例として pathChirp [10]と Pathload [11]がある。ここで、pathChirp は推定時間の短さと推定精度の高さのバランスが取れており、PathQuick との関連性も高いため、5 章で pathChirp と PathQuick の比較評価を行う。

[†] NEC サービスプラットフォーム研究所
Service Platforms Research Laboratories, NEC Corporation

従来のパケットペア方式やパケットトレイン方式は、オーバレイネットワークでの最適経路選択[12]や CDN でのサーバ選択[13], SLA 検証[14]などの、主に非リアルタイム系のアプリケーション向けに設計されている[9][11]。従来方式は秒オーダーの推定時間を要し、Abing, Spruce, pathChirp, Pathload はそれぞれ 1.3 秒, 11.0 秒, 5.5 秒, 7.0~22.0 秒の推定時間がかかることが知られている[15]。秒オーダーの推定時間は非リアルタイム系のアプリケーションでは特に問題にはならないが、リアルタイム性が重要なマルチメディアコミュニケーションでは大きな問題となる。

2.1 パケットペア方式

Abing の推定時間は 1.3 秒と比較的短いですが、パケットペア方式は一般にパケットトレイン方式に比べて推定精度が低いことが指摘[16]されており、追試実験により実証[15]されている。したがって、Abing や Spruce などのパケットペア方式は、マルチメディアコミュニケーションでの利用には適さない。

2.2 パケットトレイン方式

pathChirp と Pathload は共に probe rate model (PRM) [17]と呼ばれる、次に述べる原理により可用帯域を推定する。(a)送信端末におけるパケットトレインの送信レートが可用帯域未満である場合には、ルータにおいて計測パケットに対するキューイング遅延が発生しないため、受信端末で観測される計測パケットの受信間隔は送信端末における送信間隔と等しくなる。一方で、(b)送信レートが可用帯域を超過する場合には、ルータにおいて計測パケットがキューイングされるため、受信間隔が送信間隔より広がる。PRM では、(a)と(b)の変化点となる送信レートを検出し、可用帯域の推定値とする。

2.2.1 pathChirp の課題

pathChirp の 5.5 秒という長い推定時間の原因は、パケットトレインの構造にある。pathChirp では、パケット番号が進むに連れて各計測パケットの送信間隔が指数関数的に減少し、全パケットサイズが等しいパケットトレインを送信する。そのため、単一のパケットトレイン内で計測パケット毎の送信レートが指数関数的に増加する。広範囲の帯域を探索するためには、探索可能な帯域の下限値を低くするために 1 番目と 2 番目の計測パケットの送信間隔を長くする必要がある。しかし、この最初の長い送信間隔によりパケットトレイン全体が長くなるため、推定時間が長くなってしまふ。

2.2.2 Pathload の課題

Pathload では、pathChirp とは異なり固定ビットレート (CBR) のパケットトレインを送信する。単一のパケットトレインの送信では必ずしも PRM の変化点を検出できるとは限らないため、二分探索アルゴリズムを用いて送信レートを変えながら繰り返しパケットトレインを送信する。その結果、推定時間が 7.0~22.0 秒と長くなってしまふ。Pathload は送信間隔が等間隔である点で PathQuick と類似であるが、Pathload ではパケットサイズが全計測パケットで同一であるのに対し、PathQuick ではパケット

トレイン内でパケットサイズが変化するという点で異なる。

2.2.3 他のパケットトレイン方式

近年、pathChirp や Pathload 以外にもパケットトレイン方式が提案されている[18][19][20]。しかし、これらの方式の推定時間は 5.6 秒[18], 10.0 秒[19], 20.0 秒[20]であると報告されている。したがって、これらの方式は、マルチメディアコミュニケーションでの利用には適さない。

2.3 パケットサイズ変動方式

pathchar [21], clink [22]は計測パケットのパケットサイズを変化させるという点で PathQuick と類似であるが、これらの方式は物理帯域を推定する方式であり、物理帯域と PathQuick が対象とする可用帯域とは基本的に異なる概念である[4]。また、pathchar と clink は計測パケットを繰り返し送信する必要があり、それぞれ 21 秒と 300 秒の推定時間がかかると報告されている[23]。文献[24]は pathchar に基づくパケットサイズ変動方式であるが、225 秒もの推定時間がかかる[24]。

3. 実時間可用帯域推定方式 PathQuick の提案

本稿では、エンド端末間の可用帯域を実時間で推定するパケットトレイン方式 PathQuick を提案する。

3.1 実時間可用帯域推定の要件

これまでに述べた従来方式の課題、特に 2.2.1 小節と 2.2.2 小節の課題を考慮すると、実時間での可用帯域推定の実現には、次の 2 つの要件を満たす必要がある。

- (1) **短いパケットトレイン長**：パケットトレインの全体長を、例えば数百ミリ秒に収まるように短くする必要がある。
- (2) **単一のパケットトレインによる広範囲の帯域の探索**：単一のパケットトレインの送信により広範囲の帯域を探索することで、複数のパケットトレインの繰り返し送信を不要にする必要がある。

3.2 PathQuick におけるパケットトレインの設計

我々が設計した PathQuick のパケットトレインは、次の 2 つの特長を持つ。

- (1) **計測パケットを等間隔に配置**：パケットトレイン内の各計測パケットの間隔を短くすることで、要件(1)を満たせる。そのために、各計測パケットの送信間隔が等間隔となるようにパケットトレインを設計した (図 1-(1)を参照)。
- (2) **パケットサイズを線形に増加**：単一のパケットトレイン内で計測パケット毎の送信レートを変化させることで、要件(2)を満たせる。そのために、パケット番号が進むに連れて各計測パケットのパケットサイズが線形に増加するようにパケットトレインを設計した (図 1-(2)を参照)。なお、Pathload と pathChirp は、パケットサイズが共に固定サイズである点で PathQuick と異なる。

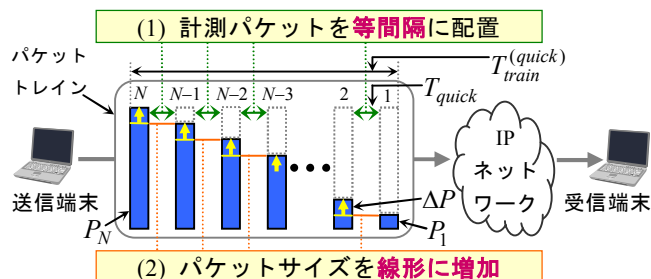


図 1 PathQuick におけるパケットトレインの設計

3.3 実時間推定の仕組み

PathQuick では、送信端末が UDP パケットで構成されるパケットトレインを受信端末に送信する。各計測パケットは送信端末における送信時刻を含み、受信端末における受信時刻と共に可用帯域の推定のために利用される。受信端末は可用帯域を推定後、UDP パケットを用いて結果を送信端末に返信することで、1 回の推定処理が完了する。

3.3.1 計測パケットの等間隔な配置

パケットトレイン内の計測パケット数を N 個とする。送信端末は各計測パケットを等間隔 T_{quick} で配置する (図 1-(1))。パケットトレイン長 $T_{train}^{(quick)}$ は数式(1)で表される。

$$T_{train}^{(quick)} = T_{quick} \cdot (N - 1) = T_{quick} \cdot N - T_{quick} \quad (1)$$

単純化のため、パケットトレインの送信時間には各計測パケットの伝播遅延 (シリアライゼーション遅延とも呼ばれる) を含めていない。 T_{quick} はミリ秒オーダーであることを仮定しており、伝播遅延は通常サブミリ秒オーダーと十分に短いため、伝播遅延を無視することの影響は十分小さい。

このように、 $T_{train}^{(quick)}$ はパケット数 N に対する線形関数となる。4.2.3 小節で後述するように、PathQuick におけるこの $O(N)$ の性質により、PathQuick ではパケットトレイン長を pathChirp よりも短くできる。したがって、PathQuick は要件(1)を満たす。

3.3.2 パケットサイズの線形な増加

各計測パケットのパケットサイズ P_i は数式(2)で表される。

$$P_i = P_1 + (i - 1) \cdot \Delta P = \Delta P \cdot i + (P_1 - \Delta P) \quad (2)$$

ここで、 $i = 1, 2, \dots, N$ であり、定数 ΔP はパケットサイズの増分である (図 1-(2))。 P_1 は定数であるため、パケットサイズ P_i は i に対する線形関数となる。

i 番目の計測パケットの送信レート R_i はパケットトレイン内での瞬時的な送信レートであり、数式(3)で表される。

$$R_i = \frac{P_i}{T_{quick}} = \frac{\Delta P}{T_{quick}} i + \frac{P_1 - \Delta P}{T_{quick}} \quad (3)$$

このように、送信レート R_i は i に対する線形関数となる。そのため、PathQuick では単一のパケットトレイン内で計測パケット毎に送信レートを増加することができるため、広範囲の帯域を探索できる。したがって、PathQuick は要件(2)を満たす。

3.4 PRM に基づく可用帯域推定

受信端末における $(i - 1)$ 番目と i 番目の計測パケットの受信間隔を T_i^{rcv} 、送信端末における i 番目の計測パケットの送信時刻を t_i と定義する。

CBR のクロストラヒックを仮定し、受信端末は 2.2 節で説明した PRM の原理に基づいて数式(4)に示す(a)と(b)の場合分けの判定を行う。

$$\begin{aligned} \text{(a)} \quad T_i^{rcv} &= T_{quick}, & \text{if } R_i \leq B[t_1, t_N] \\ \text{(b)} \quad T_{i-1}^{rcv} &< T_i^{rcv}, & \text{otherwise,} \end{aligned} \quad (4)$$

ここで、 $B[t_1, t_N]$ は時刻 $t_1 \sim t_N$ の区間における実際の可用帯域である。受信間隔が広がり始める計測パケットが k 番目であるとする。PathQuick では、 k 番目の計測パケットの送信レートである $R_k = P_k / T_{quick}$ を、可用帯域の推定値とする。図 2 は数式(4)の意味を図示したものである。図 2 右半分の $T_2^{rcv} = T_3^{rcv} = \dots = T_{k-2}^{rcv} = T_{k-1}^{rcv} = T_{quick}$ の部分が数式(4)の(a)に、図 2 左半分の $T_k^{rcv} < T_{k+1}^{rcv} < \dots < T_N^{rcv}$ の部分が数式(4)の(b)に相当する。つまり、 k 番目の計測パケットが PRM の変化点であり、 k 番目の計測パケットの送信レートが PathQuick における可用帯域の推定値となる。

ここで、クロストラヒックとして CBR を仮定すると上述したが、実ネットワークで

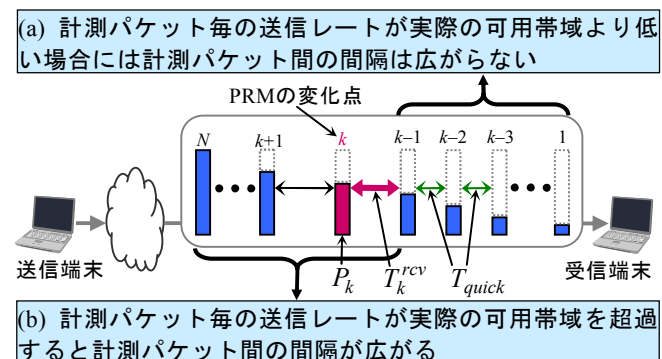


図 2 PathQuick における PRM に基づく可用帯域の推定

は必ずしもこの仮定が常に正しいとは限らない。実ネットワークでは間欠的でバースト性のあるクロストラヒックの発生によりルータでのキューイング遅延が時間的に変動する。そのため、受信端末において観測される計測パケット毎の受信間隔は単調増加になるとは限らない。したがって、単純に数式(4)を用いると推定誤差の発生を招いてしまう。PathQuick ではこの問題の対策のために、pathChirp における excursion segmentation と呼ばれる手法[10]を用いることで推定誤差の発生を防止している。

4. 評価実験

pathChirp は推定時間の短さと推定精度の高さのバランスが取れており、PathQuick との関連性も高いため、両方式について比較評価を行った。

4.1 定量評価のためのパラメータ選択

評価の詳細な説明に入る前に、定量評価で用いたパラメータについて説明する。

4.1.1 探索可能な帯域の上限

動画や Web コンテンツの CDN 業者である Akamai は文献[25]の図 17 において、Akamai の CDN で観測したスループットの実測値を公開している。北米、欧州、アジアにおけるブロードバンドの代表国である米国、スウェーデン、日本においてスループットが 10 Mbps 未満である割合はそれぞれ 95.4%、87.6%、81.4%となっている。この結果から、インターネットを対象としたエンド端末間の可用帯域推定では、探索可能な帯域の上限が 10 Mbps 以上あれば十分であると考えられる。そのため、本稿の評価では探索可能な帯域の上限を、10 Mbps 以上となるように設定した。

4.1.2 RTT

国内主要都市を対象に片方向遅延(OWD)を実測評価した文献[26]の表 1 において、国内インターネットの OWD は平均 26.21 ミリ秒であることが報告されている。そのため、本稿の評価ではエンド端末間の往復遅延(RTT)として 52 ミリ秒を設定した。

4.2 推定時間

両方式の推定時間を、同じ探索可能範囲と RTT の条件下で比較した。推定時間は、パケットトレイン長と、ルータにおけるキューイング遅延、RTT の合計である。3.3.1 小節における各計測パケットの伝播遅延と同様に、4.4.1 小節でのシミュレーションにおいて、ルータにおけるキューイング遅延は無視できるほど十分小さいことを確認した。また、RTT は同一の値を用いる。そのため、本節では PathQuick のパケットトレイン長 $T_{train}^{(quick)}$ と pathChirp のパケットトレイン長 $T_{train}^{(chirp)}$ の違いに着目する。

4.2.1 PathQuick のパケットトレイン長

探索可能な帯域の下限値と上限値をそれぞれ B_{min} と B_{max} と定義する。まず、 N を P_1 , P_N , B_{min} , B_{max} を引数とする関数として定式化し、次に、 $T_{train}^{(quick)}$ を P_1 , P_N , B_{min} , B_{max} を引数とする関数として定式化する。明らかに

$$T_{quick} = \frac{P_N}{B_{max}} \quad (5)$$

が成り立つ。2 番目の計測パケットのパケットサイズは

$$P_2 = P_1 + \Delta P = P_1 + \frac{P_N - P_1}{N-1} = \frac{(N-2)P_1 + P_N}{n-1} \quad (6)$$

である。数式(5)と(6)により、

$$B_{min} = \frac{P_2}{T_{quick}} = \frac{B_{max}}{P_N} \cdot \frac{(N-2)P_1 + P_N}{n-1} \quad (7)$$

が成り立つ。数式(7)を N について解くことで、

$$N = \frac{P_N(B_{min} + B_{max}) - 2P_1B_{max}}{P_NB_{min} - P_1B_{max}} \quad (8)$$

と定式化できる。数式(1), (5), (8)により、 $T_{train}^{(quick)}$ は数式(9)のように定式化できる。

$$T_{train}^{(quick)} = T_{quick} \cdot (N-1) = \frac{P_N(P_N - P_1)}{P_NB_{min} - P_1B_{max}} \quad (9)$$

4.2.2 pathChirp のパケットトレイン長

前述の通り、pathChirp ではパケット番号が進むに連れて各計測パケットの送信間隔が指数関数的に減少し、全パケットサイズが等しいパケットトレインが送信される。図 3 に PathQuick と pathChirp のパケットトレインの構造の違いを示す。計測パケットの送信間隔の初期値を T_{chirp} とする。拡散率と呼ばれる γ は計測パケットの指数関数的な送信間隔を制御する。パケットサイズを P_{chirp} とする。前小節と同様に、パケットトレイン内の計測パケット数 M を γ , B_{min} , B_{max} を引数とする関数として定式化し、 $T_{train}^{(chirp)}$ を P_{chirp} , γ , B_{min} , B_{max} を引数とする関数として定式化する。明らかに

$$T_{chirp} = \frac{P_{chirp}}{B_{min}} \quad (10)$$

が成り立つ。数式(10)により、

$$B_{max} = \frac{P_{chirp}\gamma^{M-2}}{T_{chirp}} = \frac{P_{chirp}\gamma^{M-2}}{P_{chirp}/B_{min}} = \gamma^{M-2}B_{min} \quad (11)$$

となる。数式(11)により $\gamma^{M-2} = B_{max}/B_{min}$ となり、 $(M-2)\log\gamma = \log(B_{max}/B_{min}) = \log B_{max} - \log B_{min}$ となるため、

$$M = \frac{\log B_{\max} - \log B_{\min}}{\log \gamma} + 2 \quad (12)$$

と定式化できる. 数式(10)と(12)により, $T_{train}^{(chirp)}$ は数式(13)のように定式化できる.

$$\begin{aligned} T_{train}^{(chirp)} &= T_{chirp} \sum_{i=2}^M \frac{1}{\gamma^{i-2}} = T_{chirp} \frac{1 - (1/\gamma)^{M-1}}{1 - 1/\gamma} \\ &= \frac{P_{chirp}}{B_{\min}} \cdot \frac{1 - (1/\gamma)^{\frac{\log B_{\max} - \log B_{\min} + 1}{\log \gamma}}}{1 - 1/\gamma}. \end{aligned} \quad (13)$$

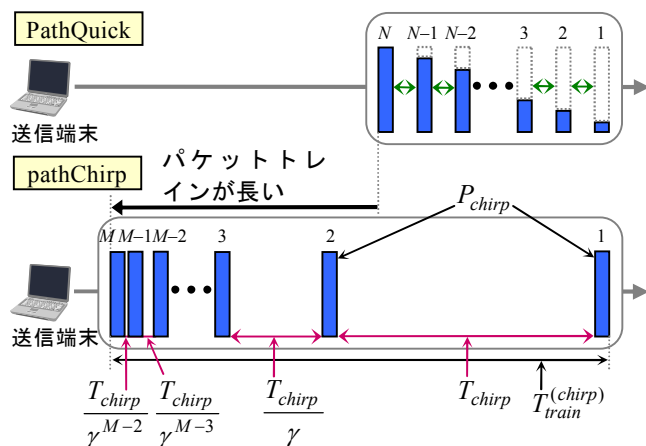


図 3 PathQuick と pathChirp におけるパケットトレインの構造

4.2.3 パケットトレイン長の比較

PathQuick と pathChirp のパケットトレイン長を比較した. $T_{train}^{(quick)}$ と $T_{train}^{(chirp)}$ を共に B_{\max} の関数として比較するために, T_{quick} , ΔP , P_1 , B_{\min} , P_N , γ , P_{chirp} のパラメータの値を固定した. つまり探索可能範囲が同一の条件下でパケットトレイン長を比較した.

まず, 各パラメータに値を設定する. PathQuick における探索可能な帯域の上限を 10 Mbps 以上にするために, $T_{quick} = 1$ ミリ秒, $P_1 = 1$ byte, $\Delta P = 12$ bytes と設定した. この時, 数式(7)により, $B_{\min} = 8 \times (1+12) / 0.001 = 104$ kbps となる. 数式(5)と $T_{quick} = 1$ ミリ秒により, B_{\max} は P_N に依存する. B_{\max} を最大化するために, 1,500 bytes の maximum transmission unit (MTU) 以下である $P_N = 1,489$ bytes を設定した. その結果, PathQuick のパケットトレインは $N = 1 + (1,489 - 1) / 12 = 125$ 個のパケットにより構成され, 数式(5)により $B_{\max} = 8 \times 1,489 / 0.001 = 11,912$ kbps となる. これにより, 10 Mbps 以上の探索可能

な帯域の上限を達成している.

pathChirp では高精度な推定のためにデフォルト値として $\gamma = 1.2$ が推奨されている [10] ため, デフォルト値を利用した. また, $P_{chirp} \geq 1,000$ bytes であることも推奨されている [10] ため, $P_{chirp} = P_N = 1,489$ bytes と設定した.

図 4 に, 各パラメータを上述した通り $T_{train}^{(quick)}(P_1, P_N, B_{\min}, B_{\max}) = T_{train}^{(quick)}(1, 1489, 104, B_{\max})$, $T_{train}^{(chirp)}(\lambda, P_{chirp}, B_{\min}, B_{\max}) = T_{train}^{(chirp)}(1.2, 1489, 104, B_{\max})$ と設定し, B_{\max} を $104 \leq B_{\max} \leq 11,912$ の範囲で変動させた場合のパケットトレイン長の比較結果を示す. $T_{train}^{(quick)}$ の $O(N)$ の性質により, PathQuick のパケットトレイン長が pathChirp よりも常に短くなっている. 例えば $B_{\max} = 11,912$ kbps の場合には $T_{train}^{(quick)} = 124$ ミリ秒, $T_{train}^{(chirp)} = 682$ ミリ秒である. したがって, pathChirp のパケットトレイン長は PathQuick よりも $682 / 124 = 5.5$ 倍長い. 52 ミリ秒の RTT を考慮すると, PathQuick の推定時間は $124 + 52 = 176$ ミリ秒で, pathChirp の推定時間は $682 + 52 = 734$ ミリ秒である. したがって, PathQuick は pathChirp に比べて $734 / 176 = 4.2$ 倍高速に推定を完了することができる.

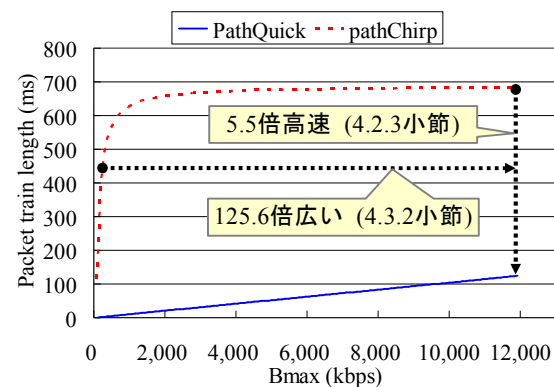


図 4 パケットトレイン長の比較

4.3 単一のパケットトレインによる探索可能な帯域の範囲

4.3.1 推定時間の上限

両方式に対して推定時間に同一の上限を設けた条件下において, 単一のパケットトレインによる探索可能な帯域の範囲を比較した.

リアルタイム性が求められるマルチメディアコミュニケーションでは, 推定時間に上限を設けることが重要である. 例として, voice-over-IP (VoIP) とリアルタイム写真共有を併用したマルチメディアコミュニケーションにおいて, 1 章で説明した可用帯域の推定値に基づいて写真を再圧縮して送信するケースを想定する. VoIP における

OWD は、エンド端末間の同期誤差に相当し、許容範囲が 0~400 ミリ秒であることが知られている[27]. また、音声と静止画に関するメディア間同期誤差の最大許容値は 500 ミリ秒であることが知られている[28]. この端末間同期誤差とメディア間同期誤差を組み合わせた、対向の端末との間のメディア間同期誤差の許容範囲は 500~900 ミリ秒となる. これは、可用帯域の推定を開始してから、推定結果に基づいて写真を再圧縮し、その写真の送信を完了するまでの一連の処理にかけても良い時間が、VoIP のOWD の値に応じて 500~900 ミリ秒の範囲で変動することを意味する. ここで、可用帯域の推定開始から写真送信の完了までの一連の処理にかけても良い時間について最も厳しい条件を想定し、VoIP のOWD が 0 ミリ秒であると仮定すると、前述の一連の処理の時間は 500 ミリ秒以下に収めなければならない. 以上を考慮し、推定時間の上限値として 500 ミリ秒を設定した.

4.3.2 探索可能な帯域の範囲の比較

図 4 において、パケットトレインの送信開始後、推定時間の上限値である 500 ミリ秒から RTT である 52 ミリ秒を差し引いた 448 ミリ秒が経過した時点において、PathQuick は既に推定処理を完了している. したがって、推定時間に上述の上限を設けた条件下における PathQuick の探索可能な帯域の範囲は、104~11,912 kbps である.

一方で、pathChirp では、448 ミリ秒の時点ではまだ計測パケットを送信している最中である. pathChirp では 3 番目の計測パケットが 389 ミリ秒、4 番目の計測パケットが 486 ミリ秒の時点で送信されるため、上述の条件下では 3 個の計測パケットしか送信できず、pathChirp の探索可能な帯域の範囲は 104~200 kbps に過ぎない. したがって、上述の条件下における PathQuick の探索可能な帯域の範囲は pathChirp に比べて $(11,912-104)/(200-104)=125.6$ 倍広いことが分かる.

また、pathChirp では、可用帯域の推定時間だけで 734 ミリ秒を要するため、可用帯域の推定開始から写真送信の完了までにかかる時間は、許容値である 500 ミリ秒を大幅に超過してしまう.

4.4 推定精度

4.4.1 シミュレーションによる評価実験

両方式の推定精度を、ネットワークシミュレータ ns-2 を用いて比較評価した. 図 5 にシミュレーション評価で用いた 12 Mbps の単一ボトルネックリンクを持つネットワークポロジを示す. 各リンクの物理帯域と OWD は図 5 の通りであり、RTT は 52 ミリ秒とした. パケットサイズが 1,000 bytes であるクロストラヒックをポアソン生起で発生させ、クロストラヒックの負荷を 0~12 Mbps の範囲で変動させた. 各パラメータは 4.2.3 小節で述べた値を利用した. また、pathChirp の busy period threshold と decrease factor と呼ばれるパラメータは文献[10]に記載のデフォルト値を利用した.

図 6 に両方式の可用帯域の推定結果を示す. 誤差が 0 となる位置に斜め線を引いている. 図 6 により、PathQuick の推定精度は pathChirp よりも優れていることが分かる.

9 Mbps 以下の範囲では、PathQuick の推定誤差は概ね ± 2 Mbps となっている. 一方

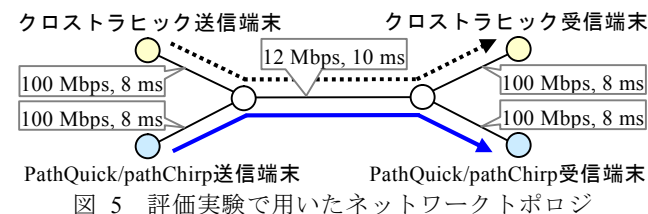


図 5 評価実験で用いたネットワークポロジ

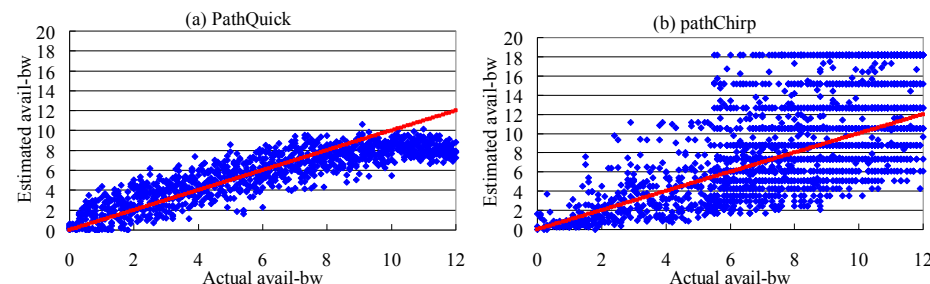


図 6 シミュレーション評価での(a)PathQuick と(b)pathChirp の推定結果 (単位: Mbps)

で pathChirp では、特に 5 Mbps 以上の範囲で推定誤差が著しく大きくなっている.

4.4.2 計測の分解能

両方式の推定精度が異なる原因を分析した結果、計測の分解能の違いに原因があることが分かった.

計測パケット毎の計測の分解能を、計測パケットの送信レートと隣の計測パケットの送信レートとの差分と定義する. ここで、計測パケット毎の計測の分解能は推定精度に直接影響を与えるものであるが、推定精度そのものではないことに注意する. 数式(3)により、PathQuick の計測パケット毎の計測の分解能は数式(14)で表される.

$$R_i - R_{i-1} = \frac{\Delta P}{T_{quick}} i + \frac{P_1 - \Delta P}{T_{quick}} - \frac{\Delta P}{T_{quick}} (i-1) - \frac{P_1 - \Delta P}{T_{quick}} = \frac{\Delta P}{T_{quick}} \quad (14)$$

このように、PathQuick の計測パケット毎の計測の分解能は定数である. 一方で、図 3 により、pathChirp の計測パケット毎の計測の分解能は数式(15)で表される.

$$\frac{P_{chirp}\gamma^{i-2}}{T_{chirp}} - \frac{P_{chirp}\gamma^{i-3}}{T_{chirp}} = \frac{P_{chirp}(\gamma-1)}{T_{chirp}}\gamma^{i-3} = \frac{P_{chirp}(\gamma-1)}{T_{chirp}\gamma^3}\gamma^i. \quad (15)$$

このように、pathChirp の計測パケット毎の計測の分解能は、パケットトレイン内で指数関数的に変化する。エンド端末間の経路が空いている場合、つまり実際の可用帯域の値が高い場合には、PRM の変化点はパケットトレインの後ろの部分で検出される。pathChirp の計測パケット毎の計測の分解能は、パケット番号が進むに連れて指数関数的に粗くなってしまいう性質がある。この性質により図 6 (b)において大きな推定誤差が観測されたと考えられる。

両方式について、単一のパケットトレインにおける平均分解能を比較した。PathQuick の平均分解能は $\Delta P/T_{quick} = 8 \times 12 / 0.001 = 96.0$ kbps となる。数式(15)により、pathChirp の平均分解能は数式(16)で表される。

$$\frac{P_{chirp}(\gamma-1)}{T_{chirp}\gamma^3} \cdot \frac{1}{M} \sum_{i=1}^M \gamma^i. \quad (16)$$

数式(12)により、 $M = 28$ 個となる。そのため、数式(16)により、pathChirp の平均分解能は 416.7 kbps となる。したがって、PathQuick の平均分解能は pathChirp に比べて $416.7 / 96.0 = 4.3$ 倍細かいということが分かる。

4.4.3 実機を用いた評価実験

PathQuick を Windows 上に実装し、Linux 上で動作する pathChirp と比較評価した。図 7 に PathQuick における推定結果の表示画面を示す。ネットワークポロジは図 5 と同じであるが、全リンクの物理帯域を 100 Mbps とした。ポアソン生起のクロストラヒックは 0~100 Mbps の範囲で 5 Mbps 刻みで変動させた。エンド端末間の RTT は 1 ミリ秒未満であった。PathQuick において $T_{quick} = 0.1$ ミリ秒、 $N = 120$ 個、 $P_1 = 32$ bytes、 $\Delta P = 12$ bytes、 $P_N = 1,460$ bytes と設定し $B_{max} = 116.8$ Mbps を実現した。pathChirp では $P_{chirp} = P_N = 1,460$ とした以外はデフォルト値を利用した。

図 8 に両方式の可用帯域の推定結果を示す。図 6 と同様に、実機評価においても PathQuick は pathChirp よりも推定精度が高いことが分かる。PathQuick の推定誤差は概ね ± 20 Mbps となっている。一方で pathChirp では、特に 50 Mbps 以上の範囲で推定誤差が著しく大きくなっている。この推定精度の差異は、前小節で説明した両方式における計測の分解能の違いに起因すると考えられる。

4.5 計測によるネットワーク負荷

両方式における単一のパケットトレインのデータ量を比較することで、計測により発生するネットワーク負荷を比較した。4.4.1 小節における PathQuick のパケットトレインのデータ量は $1+13+\dots,+1,489 = 125 \times (1+1,489) / 2 = 93.1$ KB である。pathChirp のパケットトレインのデータ量は $28 \times 1,489 = 41.7$ KB である。このように、4.4.1 小節



図 7 PathQuick の実行画面

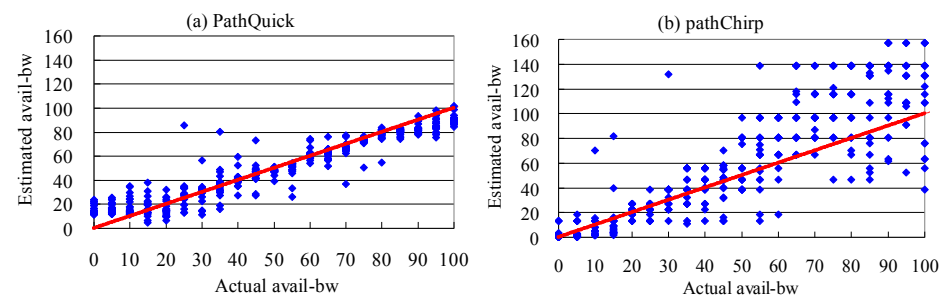


図 8 実機評価での(a)PathQuick と(b)pathChirp の推定結果 (単位: Mbps)

の条件では PathQuick のネットワーク負荷は pathChirp に比べて $93.1 / 41.7 = 2.2$ 倍高い。

しかしながら、PathQuick においてパケットサイズの増分を 2 倍の $\Delta P = 24$ bytes に変更してパケットトレイン内のパケット数を半分の $N = 63$ 個に変更し、 $P_1 = 1$ byte、 $P_N = 1,489$ bytes、 $T_{quick} = 1$ ミリ秒、 $B_{max} = 11,912$ kbps は変化させずに利用した場合には、PathQuick のパケットトレインのデータ量は $63 \times (1+1,489) / 2 = 46.9$ KB となる。このわずかにネットワーク負荷が pathChirp を上回るケースにおいても、PathQuick の平均分解能は $\Delta P/T_{quick} = 8 \times 24 / 0.001 = 192.0$ kbps となり、pathChirp に比べて 2 倍以上細かい。また、このケースにおいても PathQuick の推定精度が pathChirp を上回ることを ns-2 のシミュレーションにて確認している。

5. 考察

可用帯域推定における推定時間と推定精度の間にはトレードオフの関係があることが指摘されている[16]. 具体的には、少ないパケットトレイン数や短いパケットトレイン長により、推定時間を短縮することができるものの、推定精度が悪化することを指す。我々も評価実験を通してこのトレードオフがあることを確認した。pathChirpについて文献[15]では、5.5秒もの推定時間を要するものの推定精度は比較的高いことが報告されているが、本稿では5章で示した通り、推定時間は734ミリ秒であるものの推定精度は比較的低い。この推定精度の差異は主にパケットトレイン数の違いから生じている。文献[15]では、文献[10]に記載されている通り複数のパケットトレインを送信し、パケットトレイン毎の推定値を平均することで最終的な推定値を算出している。一方で本稿では、pathChirpの推定時間を短縮するために、単一のパケットトレインによる推定値を最終的な推定値としている。

予備実験において、PathQuickにおいても同様のトレードオフの関係があることを確認している。ただし、4.2.3小節で示した通りPathQuickの推定時間はpathChirpに比べて1/4以下と短時間であると同時に、図6、図8に示した通りより高い推定精度を実現している。したがって、PathQuickはトレードオフによる制約をpathChirpに比べて緩和できていると言える。

6. まとめと今後の課題

本稿では、エンド端末間の可用帯域を実時間で推定する方式PathQuickを提案した。評価実験の結果、PathQuickの推定時間がpathChirpに比べて1/4以下の176ミリ秒であることを示した。また、リアルタイム性の確保のために推定時間に上限を設けた条件下において、PathQuickはpathChirpに比べて125.6倍の広さの帯域の範囲を探索可能であることを示した。さらに、PathQuickは計測の平均分解能がpathChirpに比べて4.3倍細かく、計測にかかるネットワーク負荷をpathChirpと同等に設定した場合でも、pathChirpに比べて推定精度が高いことを確認した。

今後の課題としては、インターネットや携帯電話網などを対象にした評価実験の実施が挙げられる。

参考文献

[1] L. D. Cicco et al., "Skype video responsiveness to bandwidth variations," *ACM NOSSDAV*, pp. 81–86, 2008.
[2] Y. Sumi, J. Ito and T. Nishida, "Photochat: communication support system based on sharing photos and notes," *ACM CHI extended abstracts*, pp. 3237–3242, 2008.
[3] V. Roesler et al., "A new multimedia synchronous distance learning system: the IVA study case," *ACM SAC*, pp. 1765–1770, 2009.

[4] R. Prasad, C. Dovrolis, M. Murray and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, Vol. 17, Issue 6, pp. 27–35, 2003.
[5] T. Stockhammer, H. Jenkac and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, Vol. 6, Issue 2, pp. 268–277, 2004.
[6] S. Coulombe and S. Pigeon, "Quality-aware selection of quality factor and scaling parameters in JPEG image transcoding," *IEEE CIMSVP*, pp. 68–74, 2009.
[7] T. Oshiba and K. Nakajima, "Quick end-to-end available bandwidth estimation for QoS of real-time multimedia communication," *IEEE ISCC*, pp. 162–167, 2010.
[8] J. Navratil and L. Cottrell, "ABwE: a practical approach to available bandwidth estimation," *PAM Workshop*, 2003.
[9] J. Strauss, D. Katabi and F. Kaashoek, "A measurement study of available bandwidth estimation tools," *ACM IMC*, pp. 39–44, 2003.
[10] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil and L. Cottrell, "pathChirp: efficient available bandwidth estimation for network paths," *PAM Workshop*, 2003.
[11] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *ACM SIGCOMM*, pp. 295–308, 2002.
[12] D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient overlay networks," *ACM SOSP*, pp. 131–145, 2001.
[13] K. M. Hanna, N. Natarajan and B. N. Levine, "Evaluation of a novel two-step server selection metric," *IEEE ICNP*, pp. 290–300, 2001.
[14] Y. Cheng and W. Zhuang, "Dynamic inter-SLA resource sharing in path-oriented differentiated services networks," *IEEE/ACM Transactions on Networking*, Vol. 14, Issue 3, pp. 657–670, 2006.
[15] A. Shriram et al., "Comparison of public end-to-end bandwidth estimation tools on high-speed links," *PAM Workshop*, 2005.
[16] M. Jain and C. Dovrolis, "Ten fallacies and pitfalls on end-to-end available bandwidth estimation," *ACM IMC*, pp. 272–277, 2004.
[17] L. Lao, C. Dovrolis and M. Y. Sanadidi, "The probe gap model can underestimate the available bandwidth of multihop paths," *ACM SIGCOMM CCR*, Vol. 36, Issue 5, pp. 29–34, 2006.
[18] D. Croce et al., "Fast available bandwidth sampling for ADSL links: rethinking the estimation for larger-scale measurements," *PAM Conference*, pp. 67–76, 2009.
[19] Q. Liu and J. Hwang, "End-to-end available bandwidth estimation and time measurement adjustment for multimedia QoS," *IEEE ICME*, Vol. 3, pp. 373–376, 2003.
[20] S. S. Wang et al., "Fast end-to-end available bandwidth estimation for real-time multimedia networking," *IEEE MMSP*, pp. 415–418, 2006.
[21] V. Jacobson, "Pathchar—a tool to infer characteristics of Internet paths," April 1997, [online] <ftp://ftp.ee.lbl.gov/pathchar/msri-talk.pdf>
[22] A. B. Downey, "Using pathchar to estimate Internet link characteristics," *ACM SIGCOMM*, pp. 241–250, 1999.
[23] Y. Labit, P. Owezarski and N. Larrieu, "Evaluation of active measurement tools for bandwidth estimation in real environment," *IEEE/IFIP E2EMON*, pp. 71–85, 2005.
[24] T. Goto, A. Tagami, T. Hasegawa and S. Ano, "TCP throughput estimation by lightweight variable packet size probing in CDMA2000 1x EV-DO network," *IEEE/IPSJ SAINT*, pp. 1–8, 2009.
[25] Akamai Technologies, Inc., "The state of the Internet, 2nd quarter, 2009," Vol. 2, No. 2, October 2009, [online] http://www.akamai.com/html/about/press/releases/2009/press_100109.html
[26] K. Yoshida et al., "Inferring POP-level ISP topology through end-to-end delay measurement," *PAM Conference*, pp. 35–44, 2009.
[27] ITU-T recommendation G.1010, "End-user multimedia QoS categories," November 2001.
[28] R. Steinmetz, "Human perception of jitter and media synchronization," *IEEE JSAC*, Vol. 14, Issue 1, pp. 61–72, 1996.