

PathRakeTCP: 多数 TCP コネクションを用いた ファイアウォール対応の可用帯域推定手法

伊藤 尚幸[†] 大芝 崇[‡] 里田 浩三[‡] 村瀬 勉[†]

[†]名古屋大学 [‡]NEC セキュアシステムプラットフォーム研究所

あらまし 本稿では、通信経路上に UDP 通信を拒絶するファイアウォールがあっても可用帯域の推定を可能とする方式 PathRakeTCP を提案する。PathRakeTCP は、複数の推定パケットを送信することで可用帯域を推定する、いわゆるパケットトレイン手法の 1 つである。ファイアウォールは TCP 通信を通過させることが多く、PathRakeTCP は、TCP パケットでパケットトレインを構成するため、ファイアウォールを通過可能である。従来の多くのパケットトレイン手法は UDP パケットを用いており、各推定パケットの送信タイミングを正確に制御できる。一方、TCP パケットのパケットトレインを単純に送信すると、各推定パケットの送信タイミングが TCP の輻輳制御機構により乱される、という従来手法にはなかった課題が生じる。そこで PathRakeTCP では、推定パケットごとに TCP コネクションを設け、各 TCP コネクションで 1 パケットだけ送信することで、上記課題を解決する。PathRakeTCP が各 TCP パケットの送信タイミングを正確に制御できることを実証するために、実機で評価実験を行い、PathRakeTCP の推定誤差が、UDP パケットのパケットトレイン手法と同程度であることを示す。

キーワード 可用帯域, パケットトレイン, TCP, ファイアウォール

PathRakeTCP: Available Bandwidth Estimation Using Many TCP Connections for Passing through Firewalls

Naoyuki ITO[†], Takashi OSHIBA[‡], Kozo SATODA[‡], and Tutomu MURASE[†]

[†] Nagoya University [‡] Secure System Platform Research Laboratories, NEC Corporation

Abstract We propose PathRakeTCP, can estimate the available bandwidth even if there is a firewall rejects UDP on the path. firewalls often allow TCP to pass through them, and PathRakeTCP can pass through firewalls because the packet train is constructed with TCP packets. Many conventional packet train methods use UDP, allowing accurate control over the transmission timing of each estimation packet. On the other hand, if a packet train of TCP packets is simply sent, the transmission timing of each estimation packet will be disrupted by TCP's congestion control, which is a problem not found in conventional methods. To demonstrate that PathRakeTCP can accurately control the timing of TCP packet transmission, we evaluated PathRakeTCP on an actual device and found that it is capable of accurately controlling the timing of TCP packet transmission. Experiments will be conducted to show that the estimation error of PathRakeTCP is comparable to that of an UDP packet train method.

Keywords Available Bandwidth, Packet Train, TCP, Firewall

1. 背景

大学や企業のネットワーク管理者には、エンド端末間の経路上の空き帯域である可用帯域[1]の、長期間にわたる定期的な把握などにより、利用者に快適なネットワーク環境を提供することが求められる。可用帯域の把握には、複数の推定パケットを送信する、いわゆるパケットトレイン手法が有用である。この手法は、高精度・短時間・低負荷に可用帯域を推定できる。パケットトレイン手法の 1 つとして、我々が過去に開発した PathQuick3 [2]がある。PathQuick3 を含む従来手法の多くは UDP パケットでパケットトレインを構成している。しかし、UDP 通信は、ファイアウォールで拒絶されることが多い[3][4]。そのため、大学や企業のネットワークでは、ファイアウォールに阻まれて可用帯

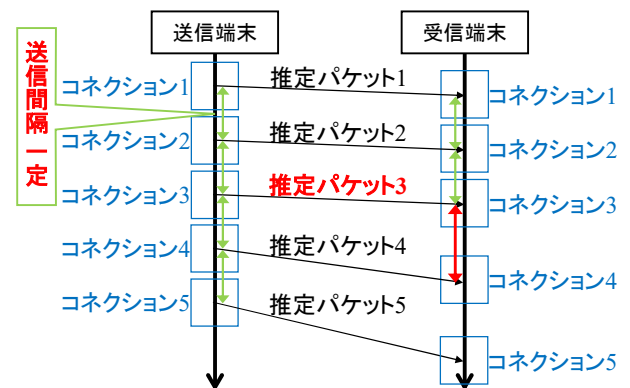


図 1. PathRakeTCP の概要. 推定パケットごとに TCP コネクションを設け、各 TCP コネクションで 1 パケットだけ送信することで、各推定パケットの送信タイミングを正確に制御

域を推定できない事態が多発してしまう。

そこで、本研究では、通信経路上に UDP 通信を拒絶するファイアウォールがあっても可用帯域の推定を可能とするパケットトレイン手法 PathRakeTCP を提案する。ファイアウォールは一部の TCP 通信を通過させることが一般的であり、PathRakeTCP は、通過が許可されるポート番号の TCP パケットでパケットトレインを構成するため、ファイアウォールを通過可能である。

以降、2 章では、従来手法について述べ、3 章では、従来手法の課題について述べる。4 章では、提案手法の仕組みを、PathQuick3 と比較しながら説明する。5 章では、実機実験により提案手法の有効性を評価し、6 章でまとめを述べる。

2. 従来のパケットトレイン手法

2.1. 従来手法における可用帯域推定の原理

PathQuick3, pathChirp [5], Pathload [6]などの従来のパケットトレイン手法では、UDP パケットで構成された 3 個以上の推定パケット列（パケットトレイン）を送信し、次に述べる原理により可用帯域を推定する。

もし、パケットトレインの送信端末での送信レートが可用帯域未満である場合、通信経路上のボトルネック箇所のルータやスイッチにおいて推定パケットに対するキューイング遅延は発生しない。そのため、パケットトレインの受信端末での推定パケットの受信間隔は送信端末での送信間隔と等しくなる。

一方、送信レートが可用帯域を超過した場合、ボトルネック箇所のルータやスイッチにおいて、推定パケットに対するキューイング遅延が発生する。その結果、受信端末での推定パケットの受信間隔は送信端末での送信間隔よりも広がり始める。受信端末では、受信間隔が広がり始める推定パケットを発見し、その直前の推定パケットの送信レートを、可用帯域の推定値とする。

2.2. PathQuick3 における可用帯域推定の仕組み

PathQuick3 は、(1)単一のパケットトレイン内で推定パケットごとの送信レートを線形に増加させ、(2)キューイング遅延が発生し始める推定パケットの送信レートを発見し、その直前の推定パケットの送信レートを可用帯域の推定値とする。上記(1)(2)の仕組みを、下記に示す。

(1) 図 2 に示すように、送信間隔を一定に保ち、パケットサイズを線形に増加させることで、推定パケットごとの送信レートを線形に増加させる。

(2) 図 3 では、推定パケット 4 で初めて推定パケットごとの送信レートが可用帯域を超えるものとする。この場合、推定パケット 3 まではキューイング遅延が発生しないため、受信間隔は送信間隔と等しくなる。

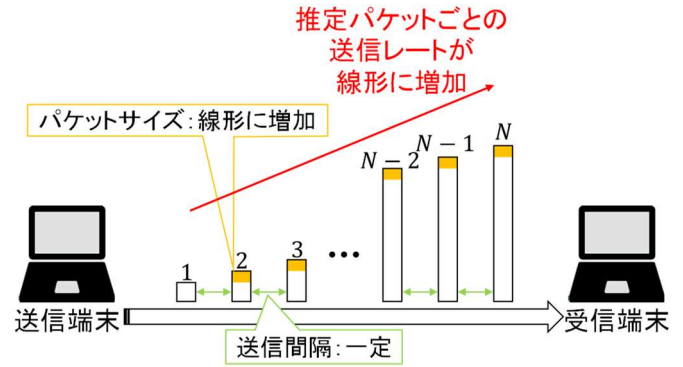


図 2. PathQuick3 における送信レートの線形な増加

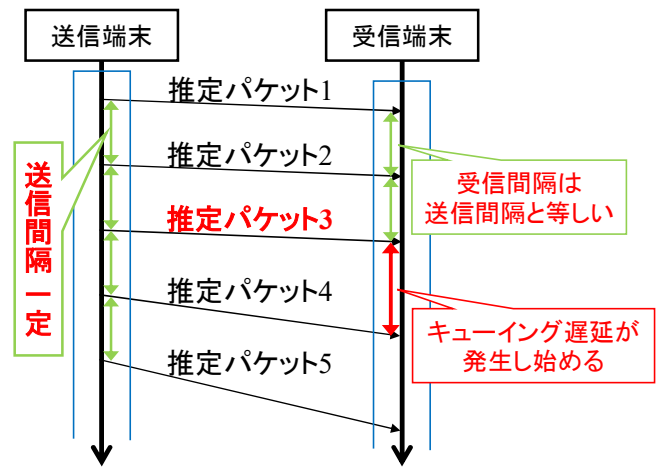


図 3. 送信レートの増加によるキューイング遅延の発生

一方、推定パケット 4 の送信レートは可用帯域を超えているため、通信経路上のボトルネック箇所のルータやスイッチにおいてキューイング遅延が発生する。その結果、受信間隔が送信間隔よりも広がる。受信端末において、受信間隔が広がり始めた推定パケットが推定パケット 4 であるので、その直前の推定パケット 3 の送信レートを可用帯域の推定値とする。

なお、PathQuick3 では、パケット送信間隔、先頭パケットのサイズ、パケットサイズの増分、推定パケット数の変更により、推定可能な帯域の上限、推定可能な帯域の下限、推定の分解能を調整できる。上記推定上限、推定下限、分解能は文献[2]の推定パケットごとの送信レートの計算式である式(3)から導出できる。

3. 従来手法の課題

3.1. UDP パケットを用いたパケットトレイン手法

従来のパケットトレイン手法には、ファイアウォールを通過できない課題がある。1 章で述べた通り、従来の UDP パケットを用いたパケットトレイン手法は、

図 4 のように推定パケットがファイアウォールに拒絶され、可用帯域推定を推定できない。

3.2. TCP パケットを用いたパケットトレイン手法

UDP 通信で提案された手法を TCP 通信にそのまま流用することはできない。その理由は、TCP パケットのパケットトレインを単純に送信すると、各推定パケットの送信タイミングが TCP の輻輳制御機構により乱される、という従来手法にはなかった課題が生じるからである。TCP の通信開始時の輻輳ウィンドウサイズは 2, 4, 10 のいずれかであり [7], パケットトレイン手法では数十個以上の推定パケットを送信することが一般的なので、推定パケット数がこの値を超えることになる。図 5 に示すように、推定パケット数が輻輳ウィンドウサイズを超えると、TCP の応答(ACK)が返ってくるまで、次の推定パケットを送信できないため、送信タイミングが TCP の輻輳制御機構により乱され、可用帯域の推定精度が低下してしまう。

3.3. TCP 通信を用いた可用帯域推定手法

TCP 通信を用いた可用帯域推定手法として、ImTCP [8]や文献[9][10][11]などがある。しかし、ImTCP は、利用するマシンの OS の TCP スタックを改造する必要がある。そのため、大学や企業のネットワーク管理者は手軽には利用できない。

なお、文献[9][10][11]は、パケットトレイン手法のように可用帯域の把握のためにネットワーク管理者がアクティブに推定パケットを送信する手法ではなく、一般の利用者のパケットをキャプチャして分析するパッシブな計測手法である。

4. ファイアウォール対応の可用帯域推定方式

PathRakeTCP の提案

本稿では、従来手法の課題である、(1)ファイアウォールの通過、(2)送信タイミングの正確な制御、(3)手軽な利用、の 3 つをすべて解決する可用帯域推定手法 PathRakeTCP¹を提案する。(1)については、TCP 通信を利用することでファイアウォールを通過できる。ネットワーク管理者は、自分たちが管理しているネットワーク内のファイアウォールの設定内容にアクセスできる。そのため、ネットワーク管理者はファイアウォールを通過可能な TCP のポート番号を知ることができる。PathRakeTCP では、上記ポート番号を使うことで、ファイアウォールを通過して、可用帯域を推定できる。(2)については、次の通りである。各 TCP コネクションが送信するパケットは 1 個ずつである。通信開始時の輻輳ウィンドウサイズは、2, 4, 10 のいずれかなので、

¹ PathRakeTCP の Rake は「熊手」を意味し、多数の TCP コネクションから推定パケットを櫛状に送信する様子が、熊手の歯の形状に似ていることに由来する。

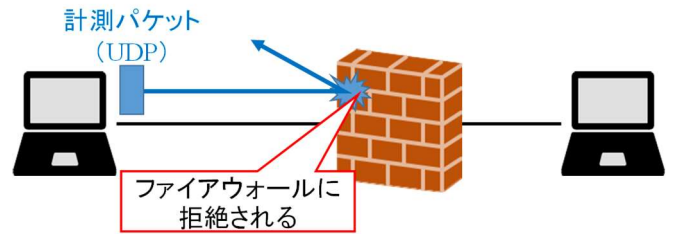


図 4. UDP 通信がファイアウォールに拒絶されて、推定できない

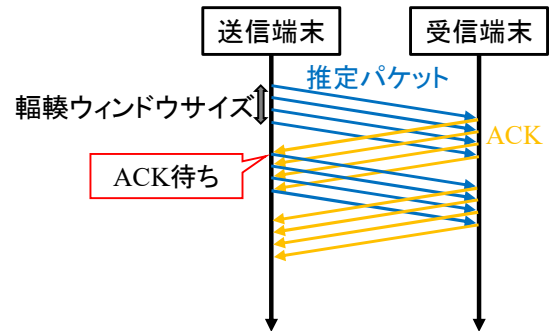


図 5. 各推定パケットの送信タイミングが TCP の輻輳制御機構により乱される

推定パケット数が輻輳ウィンドウサイズを超えることはなく、輻輳制御機構の影響を受けない。そのため、各推定パケットの送信タイミングを正確に制御できる。(3)については、OS の TCP スタックを改造する必要がなく、ユーザランドで実装可能であるため、手軽に利用できる。

4.1. 多数 TCP コネクションを用いた可用帯域推定手法

PathRakeTCP では、従来の UDP 通信を用いた手法と同等の正確さで各推定パケットの送信タイミングを制御できる。このことを、PathRakeTCP と PathQuick3 の比較として、図 1 と図 3 を用いて説明する。PathRakeTCP では、図 1 に示すように、TCP の輻輳制御機構の影響を受けずに、各推定パケットを一定の送信間隔で送ることができる。これは 4 章冒頭(2)で説明したように、パケット毎に TCP コネクションを設けているため、推定パケット数が輻輳ウィンドウサイズを超えないからである。そのため、図 1 と図 3 を見比べると、まったく同じように推定パケットの送信を制御できていることがわかる。これは、TCP 通信でも UDP 通信と同様に送信タイミングを正確に制御可能ということである。

また、この手法は、推定パケットの送信間隔を調整

することで、理論上は、任意の範囲の可用帯域を推定できる。

5. 実機を用いた推定誤差の比較評価実験

PathRakeTCP と PathQuick3 は、利用するトランスポート・プロトコルが異なるものの、パケットトレインの構造は同じである。したがって、PathRakeTCP が各推定パケットの送信タイミングを、PathQuick3 と同等の正確さで制御できれば、両者の推定誤差は同程度になることが期待される(厳密には、TCP パケットと UDP パケットではヘッダサイズが異なるが、この違いが推定誤差に与える影響は無視できるほど小さい)。そこで、実機を用いて両者の推定誤差の比較評価実験を行う。

5.1. 比較評価実験の条件

図 6 に比較評価実験の機材と接続構成を示す。この実験では、他のユーザが通信をしている最中に、パケットトレイン送信用 PC からパケットトレイン受信用 PC へ PathRakeTCP あるいは PathQuick3 のパケットトレインを送信して可用帯域を推定する状況を想定している。他のユーザの通信は、背景トラヒック送信用 PC から背景トラヒック受信用 PC へ iperf を用いて送信する UDP の背景トラヒックにより模擬する。背景トラヒックの通信速度を c Mbps とし、 c を 0 Mbps から 90 Mbps まで 10 Mbps 刻みで変化させながら、パケットトレインを送信して推定値を記録する。可用帯域の真値は、式(1)で表される。

真値 = ボトルネック箇所の物理帯域 - 背景トラヒック (1)

また、推定誤差は式(2)で表され、その平均値である式(3)の Mean Absolute Error (MAE) により評価を行う(式(3)の n は総推定回数である)。

$$\text{推定誤差} = |\text{推定値} - \text{真値}| \quad (2)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n (\text{推定誤差})_i \quad (3)$$

ここで、ボトルネック箇所であるハブ 1 の物理帯域は 100 Mbps である。この実験でボトルネック箇所の物理帯域を 100 Mbps とする理由は、ネットワークアプリケーションの中でも特に広い帯域幅が要求される動画配信サービスやビデオ通話サービス(表 1 に代表例を示す)が快適に利用できる物理帯域の環境で、可用帯域の推定誤差を確認したいからである。

また、推定に使用したパケット送信間隔、先頭パケットのサイズ、パケットサイズの増分、推定パケット数はそれぞれ、0.1 ms, 32 Bytes, 12 Bytes, 120 個であ

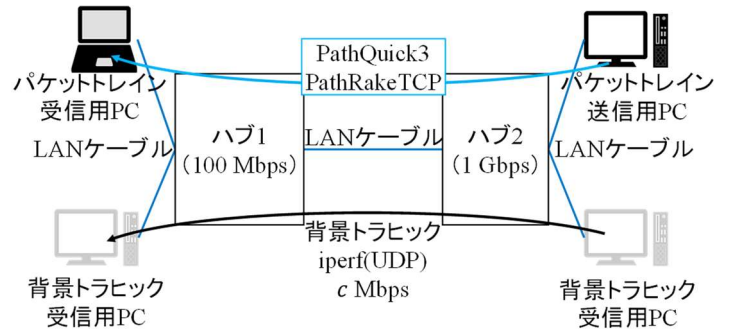


図 6. 比較評価実験で用いた機材と接続構成

表 1. 各アプリケーションで要求される通信速度

アプリケーション名	要求通信速度 (最大値)
YouTube [12]	20 Mbps
Netflix [13]	15 Mbps
Zoom [14]	4 Mbps
Microsoft Teams [15]	4 Mbps

表 2. パケットトレイン送受信用 PC のスペック

	送信用 PC	受信用 PC
CPU	AMD Ryzen 5 PRO 4650G with Radeon Graphics 3.70Ghz	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
メモリ	32 GBytes	16 GBytes
OS	Windows 10 Home	Windows 10 Home

り、これらの値から算出される推定下限および推定上限は、それぞれ 2.6 Mbps, 116.0 Mbps である(算出方法は文献[2]を参照)。これら値を採用する理由は、今回想定される可用帯域の範囲を推定可能範囲に含み、かつ、ネットワーク管理者が可用帯域を把握する用途において十分な分解能であると考えられる 1 Mbps 程度の分解能を持つようにしたいからである。

なお、PathRakeTCP では、パケットトレインの送信の前に推定パケット数と同じ本数(今回は 120 本)の TCP コネクションを設ける必要がある。この処理は、今回パケットトレイン送受信用に利用した PC(表 2 にスペックの詳細を示す)では瞬時に完了するため、すぐにパケットトレインを送信できる。

5.2. 推定誤差の実験結果

図 7, 図 8 はそれぞれ、PathRakeTCP と PathQuick3 の推定結果である。図の各点は、背景トラヒックを変化させた際の推定値の中央値(各背景トラヒック c について 5 回推定を行った際の中央値)である。推定値が真値に完全に一致する場合には、斜め 45° の直線上に点がプロットされる。この斜線と点との距離(図 7 の双方向矢印の長さ)が短いほど推定誤差が小さいこ

とを意味する。図 7, 図 8 から, 両者の推定誤差は同程度であることが読み取れる。

両者の推定誤差を定量的に比較する。全 10 パターンの背景トラフィックの通信速度 c について 5 回ずつ, すなわち計 50 回(式(3)において $n = 50$) の推定を行う。PathRakeTCP の MAE は, 15.6 Mbps (ボトルネック箇所の物理帯域に対する比率は $15.6/100 = 15.6\%$) であり, PathQuick3 の MAE は, 18.2 Mbps (ボトルネック箇所の物理帯域に対する比率は $18.2/100 = 18.2\%$) であり, 両者の MAE は同程度である。このことから, PathRakeTCP は, 各 TCP パケットの送信タイミングを, TCP の輻輳制御機構により乱されることなく, 正確に制御できていることが確認できる。

6. まとめ

本稿では, 通信経路上に UDP 通信を拒絶するファイアウォールがあっても可用帯域の推定を可能とする可用帯域推定手法として, PathRakeTCP を提案した。PathRakeTCP では, パケットトレイン中の推定パケットごとに TCP コネクションを設け, 各 TCP コネクションで 1 パケットだけ送信する。PathRakeTCP は 3 つの特長を持つ。(1)ファイアウォールは TCP 通信を通過させることが多く, PathRakeTCP は, TCP パケットでパケットトレインを構成するため, ファイアウォールを通過可能である。(2)推定パケットごとに TCP コネクションを使用するため, 各推定パケットの送信タイミングが輻輳制御機構によって乱されることがなく, 送信タイミングの正確な制御が可能である。(3) OS の TCP スタックを改造する必要がなく, ユーザランドで実装可能であるため, 手軽に利用できる。

PathRakeTCP が各 TCP パケットの送信タイミングを正確に制御できることを実証するために, 実機による推定誤差の評価を行う。評価の結果, PathRakeTCP の推定誤差は, 我々が過去に開発した UDP パケットのパケットトレイン手法 PathQuick3 と同程度であることから, PathRakeTCP が各 TCP パケットの送信タイミングを正確に制御できていることが確認できる。

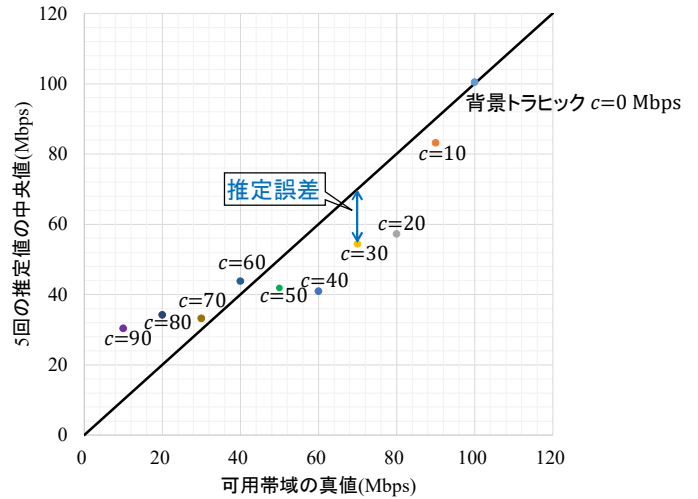


図 7. PathRakeTCP の推定結果

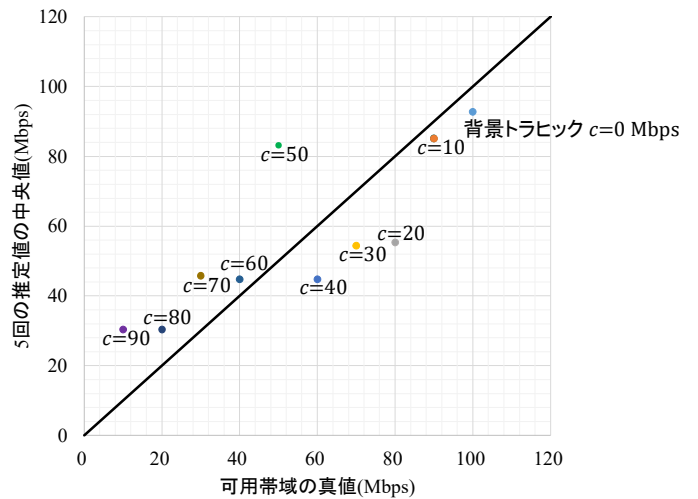


図 8. PathQuick3 の推定結果

文 献

- [1] 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.
- [2] T. Oshiba, K. Nogami, K. Nihei, and K. Satoda, "Robust available bandwidth estimation against dynamic behavior of packet scheduler in operational LTE networks," *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp.1276–1283, 2016.
- [3] T. Reddy, P. Patil, D. Wing, and B. V. Steeg, "WebRTC udp firewall traversal," *IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI)*, 2015.
- [4] H. Y. Yang, K. H. Lee, and S. J. Ko, "Communication quality of voice over TCP used for firewall traversal," *2008 IEEE International Conference on Multimedia and Expo (ICME)*, pp.29–32, 2008.
- [5] V. J. Ribeiro, R. H. Riedi, R.G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient available bandwidth estimation for network paths," *Passive and Active Measurement (PAM) workshop*, 2003.

- [6] M. Jain and C. Dovrolis, “End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput,” *ACM SIGCOMM*, pp.295–308, 2002.
- [7] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, “Increasing TCP's initial window,” IETF RFC 6928, 2013.
- [8] C. L. T. Man, G. Hasegawa, and M. Murata, “ImTCP: TCP with an inline measurement mechanism for available bandwidth,” *Computer Communications (COMCOM)*, pp. 1614–1626, 2006.
- [9] F. Ciaccia, I. Romero, O. A. Abella, D. Montero, R. S. Gracià, and M. Nemirovsky, “SABES: Statistical available bandwidth estimation from passive TCP measurements,” *2020 IFIP Networking Conference*, pp.743–748, 2020.
- [10] S. K. Khangura and M. Fidler, “Available bandwidth estimation from passive TCP measurements using the probe gap model,” *2017 IFIP Networking Conference and Workshops*, pp.1–9, 2017.
- [11] M. Zangrilli and B. B. Lowekamp, “Applying principles of active available bandwidth algorithms to passive TCP traces,” *International Workshop on Passive and Active Network Measurement (PAM)*, pp.333–336, 2005.
- [12] Google, “システム要件,” <https://support.google.com/youtube/answer/78358?hl=ja>, ref. April. 15, 2022.
- [13] Netflix, “推奨されるインターネット接続速度,” <https://help.netflix.com/ja/node/306>, ref. April. 15, 2022.
- [14] Zoom, “Zoom system requirements: Windows, macOS, Linux,” <https://support.zoom.us/hc/en-us/articles/201362023-Zoom-system-requirements-Windows-macOS-Linux>, ref. April. 15, 2022.
- [15] Microsoft, “Microsoft Teams 用に組織のネットワークを準備する,” <https://docs.microsoft.com/ja-jp/microsoftteams/prepare-network>, ref. April. 15, 2022.