

広帯域・高遅延ネットワークにおける TCP スループットの評価

篠崎誠太郎¹ 古屋保² 下園幸一² 升屋正人²

概要: TCP では受信者からの確認応答を待つて次のパケットを送信することでデータの到達性を保証している。このため往復遅延時間が大きくなると、帯域幅の大小に関わらず単位時間あたりに伝送できるデータの量が少なくなり帯域幅を使い切れない。これまで、高遅延環境におけるスループットの評価は広く行われてきているが、帯域幅を変えて測定した例はなく、必要以上の帯域幅があれば十分なのか、また、帯域幅が不足している場合にスループットがどうなるのかについては詳しく知られていない。そこで本研究では、さまざまな帯域幅の環境下で、往復遅延時間や輻輳制御アルゴリズムを変えてスループットを測定し、帯域幅との関係性を評価した。

キーワード: TCP, 帯域幅遅延積, 輻輳制御

Evaluation of TCP throughput in long fat networks

SEITARO SHINOZAKI¹ TAMOTSU FURUYA²
KOICHI SHIMOZONO² MASATO MASUYA²

Abstract: TCP guarantees data reachability by waiting for an acknowledgement from the receiver to send the next packet. For this reason, as the round-trip delay increases, the amount of data that can be transmitted per unit of time decreases whether the bandwidth is large or small, and the bandwidth is not fully used. So far, throughput has been widely evaluated in high latency environments, but there are no examples of measurements with various bandwidths, and it is not known in detail whether more bandwidth than necessary is enough or what happens to throughput when bandwidth is not enough. In this research, throughput was measured under various bandwidths with different round-trip delay times and congestion control algorithms, and the relationship with bandwidth was evaluated.

Keywords: TCP, bandwidth-delay product, congestion control algorithm

1. はじめに

単位時間あたりのデータ伝送量の最大値が大きいことを広帯域といい、送信者がデータを送信してから受信者が受信するまでの時間が大きいことを高遅延という。ネットワーク機器の性能向上と遠隔地接続の機会の増加により、広帯域・高遅延ネットワークの利用機会が広がってきた。これに伴い、広帯域・高遅延ネットワークでは帯域幅を使い切れない、という問題が表面化しつつある。

確認応答が帰ってくるまでの時間である往復遅延時間と帯域幅の積を帯域遅延積といい、一度に送信できるデータ量の上限である輻輳ウィンドウサイズがその 2 倍以上の大きさであれば帯域幅を使い切ることができる。ところが、輻

輳制御アルゴリズムによって効率的なデータ伝送を実現するため、輻輳ウィンドウサイズは最大値に固定されず変動している。このため、実際の単位時間あたりのデータ伝送量（スループット）は帯域幅よりも必ず小さくなる。しかし、遅延及び帯域幅とスループットの関係については十分に調べられていない。

伊藤らによる、TCP 上でファイル転送する場合において、最大ウィンドウサイズとネットワークの通信速度・遅延時間によるスループットへの影響を調査した研究[1]、北辻らによる、実ネットワークの遅延環境を用い、それぞれの区間において TCP によるデータ伝送を観測して性能を示した研究[2]、長谷川と村田による、TCP Reno の高速・高遅延ネットワーク環境での改善方法の提案の研究[3]、Sato らによ

¹ 鹿児島大学大学院理工学研究科
Graduate School of Science and Engineering, Kagoshima University
² 鹿児島大学情報基盤統括センター

Center for Management of Information Technologies, Kagoshima University

る、高遅延環境で TCP 全体に FEC を用いてパケットロスを防ぎスループットを高くする研究[4]など、広帯域・高遅延のネットワークにおけるスループットの実測や輻輳制御方法の改善の提案などに関する研究の例はある。しかし、広帯域・高遅延ネットワークにおいて、帯域幅とスループットの関係について調べた例はない。

そこで本研究では様々な輻輳制御アルゴリズムや帯域幅、遅延環境下でスループットを測定し、これらの関係を調べることにした。さらに、測定されたスループットから帯域幅をどれだけ利用できているか、すなわち、帯域利用効率を求める。この帯域利用効率から、遅延の大きさと使い切ることができる帯域幅を明らかにすることにした。なお、本研究では、高遅延を 50 ミリ秒以上の往復遅延時間、広帯域を 1 Gbps 以上の帯域幅と想定した。

2. 用いた輻輳制御アルゴリズム

輻輳制御アルゴリズムには、パケットロスを検知して輻輳ウィンドウサイズを調整する Loss-based 手法、遅延の大きさに応じて輻輳ウィンドウサイズを調整する Delay-based 手法、パケットロスと遅延の両方を基準に輻輳ウィンドウサイズを調整する Hybrid 手法がある。本研究では、Loss-based 手法である Reno と CUBIC、Delay-based 手法である BBR を評価の対象とした。Reno は古くから利用されている標準的な輻輳制御アルゴリズム、CUBIC は Linux や Windows で標準となっている代表的な Loss-based 手法のアルゴリズム、BBR は Google が開発した代表的な Delay-based 手法のアルゴリズムである。

3. 測定環境と方法

3.1 測定環境

本研究では仮想環境でスループットの測定を行った。PC にインストールした Oracle VirtualBox の中に仮想マシンを 2 台作成し、OS として Oracle Linux 8.6 を最小インストールして用いた。プロセッサ数はいずれも 1 とし、受信側の仮想マシンにはメモリを 2 GB、送信側の仮想マシンにはメモリを 1 GB 割り当てている。仮想マシンには HTTP サーバソフトウェアとして nginx を、ネットワークパフォーマンス測定ツールとして iperf3 を追加でインストールした。また、仮想ネットワークインターフェースカードとして Intel PRO/1000MT デスクトップを設定した。

3.2 ダウンロード時間の測定

スループットの測定は、サイズが 10 MB と 100 MB のファイルを、nginx を動作させた送信側の仮想マシン②に置き受信側の仮想マシン①から curl コマンドでダウンロードすることで行う (図 1)。ダウンロードしたサイズと時間からスループットを求めた。ダウンロードは 10 回実行して中央値を求めたが、帯域幅を設定していない場合のみ、外れ値が多く見られたため測定を 100 回行って中央値を求めた。

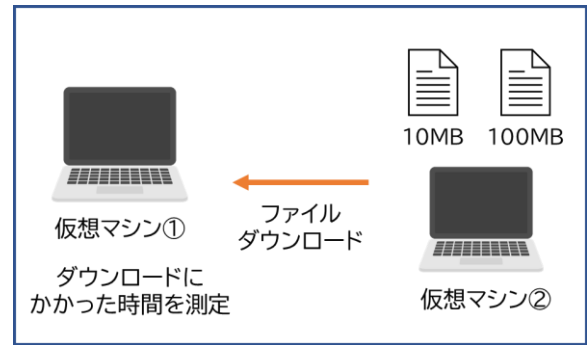


図 1 スループットを求めるためのダウンロード時間の測定方法

3.3 遅延と帯域幅及び輻輳制御アルゴリズムの設定

遅延の設定は、Oracle Linux 8.6 に標準で搭載されている tc コマンドで行った。Tc コマンドによる設定は送出パケットにのみ働くため、2つの仮想マシンそれぞれに同じ値の遅延を設定した。設定値の 2 倍の値が往復遅延時間となる。往復遅延時間は 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 250, 300, 400, 500 ミリ秒に設定して測定を行った。帯域幅の設定も同様に、tc コマンドを 2つの仮想マシンで用いることを行った。設定した帯域幅は 100 Mbps, 50 Mbps, 10 Mbps の 3 通りである。往復遅延時間とは異なり設定値が帯域幅となる。例えば、帯域を 10 Mbps に設定するときは、それぞれの仮想マシンで

```
sudo tc qdisc add dev enp0s3 root netem rate 10mbit
sudo tc qdisc del dev enp0s3 root
```

として設定を解除した後に改めて設定した。

輻輳制御アルゴリズムは例えば Reno への変更の場合、

```
sudo sysctl -w net.ipv4.tcp_congestion_control=reno
```

として変更した。CUBIC、BBR への変更も同様の方法で行い、輻輳制御アルゴリズムを変えて測定を行った。

4. スループットと帯域利用効率

4.1 輻輳制御アルゴリズムとして CUBIC を用いた場合

輻輳制御アルゴリズムとして CUBIC を使い、設定帯域と往復遅延時間を変えて測定したスループットの、往復遅延時間設定が 0, 10, 50, 100 ミリ秒のときの値を表 1 に示す。

表 1 より、往復遅延時間が小さいときは帯域を設定しない方がスループットは高くなるが、50 ミリ秒や 100 ミリ秒など往復遅延時間が大きいときは帯域を設定した方が、スループットが高くなるのがわかる。CUBIC はパケットロスが検知されるまで輻輳ウィンドウサイズを増加させ続ける Loss-based 手法の輻輳制御アルゴリズムである。帯域を設定しない場合には大きく変動する輻輳ウィンドウのサイズが、帯域を設定することで上限付近でのみ変動するようになりスループットが高くなったと考えられる。

表 1 輻輳制御アルゴリズムとして CUBIC を用いた場合のスループットの中央値 (Mbps)

		往復遅延時間(ms)			
		0	10	50	100
設定帯域 (Mbps)	設定無し	1218.6	179.1	58.7	39.4
	100	92.2	93.4	91.4	86.6
	50	47.7	46.5	46.1	44.4
	10	9.6	9.6	9.5	9.5

また、設定帯域ごとの往復遅延時間と帯域利用効率を図2に示す。帯域を設定しない場合は、往復遅延時間の増加に伴って帯域利用効率は急激に減少するが、帯域を設定した場合は利用効率の減少は緩やかである。また、設定帯域が小さいほど低下の度合いが小さいことがわかった。

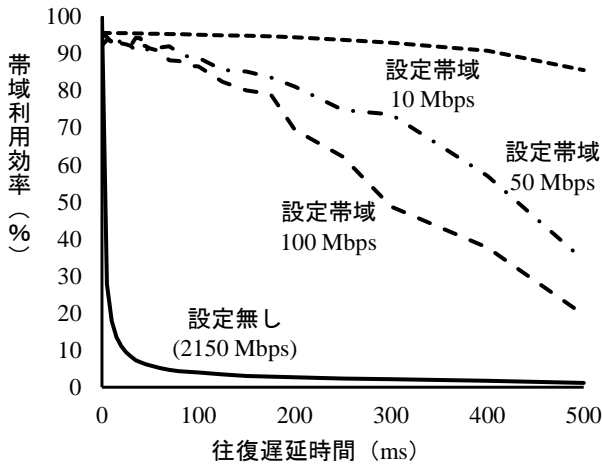


図2 輻輳制御アルゴリズムとしてCUBICを用いた場合の往復遅延時間と帯域利用効率

4.2 輻輳制御アルゴリズムとしてRenoを用いた場合

輻輳制御アルゴリズムとしてRenoを用い、設定帯域と往復遅延時間を変えて測定したスループットの、往復遅延時間設定が0、10、50、100ミリ秒のときの値を表2に示す。CUBICのときよりも往復遅延時間が大きい場合にスループットが低くなった。帯域を設定するとCUBICの場合と同様に、往復遅延時間が大きいときに帯域を設定しないときよりもスループットが高くなることがわかったが、CUBICよりもその割合は小さい。これはパケットロスを検知した際に、輻輳ウィンドウサイズをCUBICより大きく低下させるRenoの輻輳制御アルゴリズムに起因しているものと考えられる。

表2 輻輳制御アルゴリズムとしてRenoを用いた場合のスループットの中央値 (Mbps)

		往復遅延時間(ms)			
		0	10	50	100
設定帯域 (Mbps)	設定無し	1220.1	162.7	49.1	27.5
	100	94.2	93.6	68.4	33.5
	50	47.8	47.7	44.0	29.1
	10	9.6	9.6	9.5	9.5

また、設定帯域ごとの往復遅延時間と帯域利用効率を図3に示す。CUBICの場合と同傾向は同じであるが、設定帯域が100 Mbpsと50 Mbpsのときの帯域利用効率の低下の度合いがCUBICの場合に比べて大きく、帯域を設定しない場合と同様に往復遅延時間が大きくなると急激に効率が低下することがわかった。

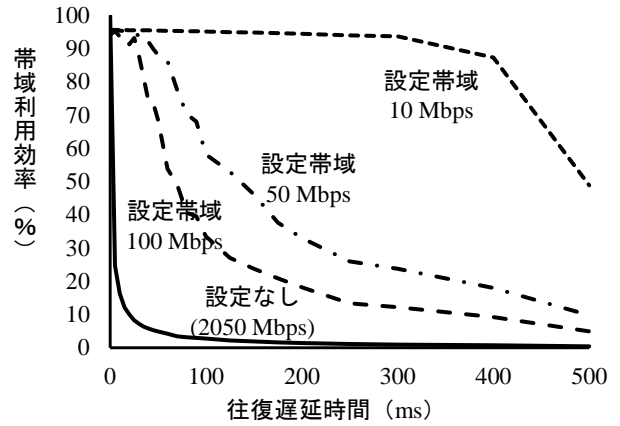


図3 輻輳制御アルゴリズムとしてRenoを用いた場合の往復遅延時間と帯域利用効率

4.3 輻輳制御アルゴリズムとしてBBRを用いた場合

輻輳制御アルゴリズムとしてBBRを用い、設定帯域と往復遅延時間を変えて測定したスループットの、往復遅延時間設定が0、10、50、100ミリ秒のときの値を表3に示す。スループットはCUBICやRenoの場合よりも高い。また、帯域を設定しても帯域を設定しない場合よりスループットが高くなることはなかった。BBRは遅延の大きさに応じて輻輳ウィンドウサイズを調整するDelay-based手法の輻輳制御アルゴリズムであり、往復遅延時間を常にモニタリングしながらデータ伝送量を調整している。遅延の大きさに対し、ネットワークが処理できる範囲内でデータ伝送量を増加させているため、帯域を設定してもBBRにおけるウィンドウサイズの変動に影響がなかったと考えられる。

表3 輻輳制御アルゴリズムとしてBBRを用いた場合のスループットの中央値 (Mbps)

		往復遅延時間(ms)			
		0	10	50	100
設定帯域 (Mbps)	設定無し	1341.0	673.5	266.7	150.8
	100	91.1	89.1	84.4	84.0
	50	45.3	45.3	44.3	42.8
	10	9.4	9.3	9.3	9.1

また、設定帯域ごとの往復遅延時間と帯域利用効率を図4に示す。CUBICの場合と同傾向は同じであるが、設定帯域が50 Mbpsのときの帯域利用効率の低下の度合いがCUBICの場合に比べて小さい。

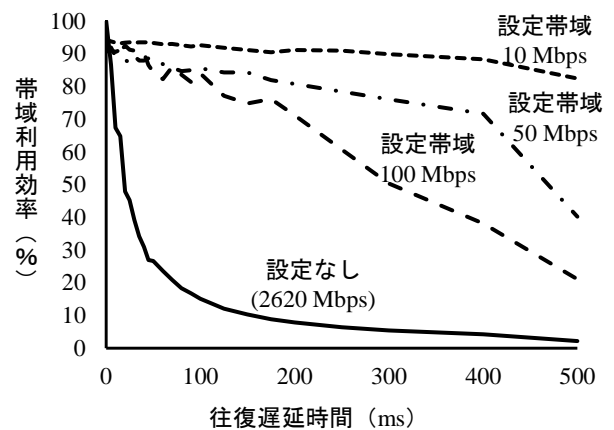


図4 輻輳制御アルゴリズムとしてBBRを用いた場合の往復遅延時間と帯域利用効率

5. まとめ

輻輳制御アルゴリズムとして CUBIC, Reno, BBR を用い、往復遅延時間と設定帯域を変えてスループット求めた。CUBIC と Reno では、往復遅延時間が大きい場合、帯域を設定しない場合に比べて帯域を設定した方が、スループットが高くなることがわかった。一方 BBR では、帯域を設定してもスループットは高くならなかった。また、帯域を設定すると設定しないときよりも帯域利用効率が高くなり、往復遅延時間が大きい方がより帯域利用効率が大きくなるが、その度合は輻輳制御アルゴリズムにより異なることがわかった。

参考文献

- [1] 伊藤 嘉浩, 石倉 雅巳, 飯作 俊一, 浅見 徹, "広帯域・高遅延ネットワーク上での TCP スループット特性の測定, "電子情報通信学会技術研究報告. IN, 情報ネットワーク, vol.95, no.28, pp.55-61, May.1995.
- [2] 北辻 佳憲, 窪田 歩, 山崎 克之, "広帯域・遅延環境における TCP 伝送の性能と評価, "電子情報通信学会技術研究報告. IA, インターネットアーキテクチャ, vol.102, no.636, pp.41- 46, Jan.2003.
- [3] 長谷川 剛, 村田 正幸, "高速・高遅延ネットワークのためのトランスポート層プロトコル, "電子情報通信学会技術研究報告. IN, 情報ネットワーク, vol.106, no.524, pp.41-46, Jan. 2007.
- [4] Sato, Y., Koga, H. and Ikenaga, T., "TCP Using Adaptive FEC to Improve Throughput Performance in High-Latency Environments", IEICE Transactions on Communications, Vol.E102 B, No.3, pp.537-544, 2019.