

# Paced TCP がルータのバッファサイズ設定に与える影響

多田 健太郎<sup>†</sup> 長谷川 剛<sup>††</sup> 村田 正幸<sup>†††</sup>

<sup>†</sup> 大阪大学 基礎工学部情報科学科 〒 560-8531 大阪府豊中市待兼山町 1-3

<sup>††</sup> 大阪大学 サイバーメディアセンター 〒 560-0043 大阪府豊中市待兼山町 1-32

<sup>†††</sup> 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

E-mail: <sup>†</sup>k-tada@ics.es.osaka-u.ac.jp, <sup>††</sup>hasegawa@cmc.osaka-u.ac.jp, <sup>†††</sup>murata@ist.osaka-u.ac.jp

あらまし 現在、インターネットルータのバッファサイズの決定には帯域遅延積を指標とする方法が広く利用されている。これに対し、TCP を用いた通信が多数存在するという条件の下であれば、ネットワークリンクの利用率を維持するためには帯域遅延積をフロー数の平方根で除算しただけのサイズで十分であるという方法が提唱されている。また、TCP ペーシングを用いることによってデータを送信する際のバースト性が軽減される場合、さらに小さい数十パケットのバッファサイズで十分であるという主張も提起されている。しかし、これら主張はボトルネックリンクの利用率以外の視点からの十分な評価が行われていない。そこで本稿では、ns-2 を用いたシミュレーションにより、TCP ペーシングがルータのバッファサイズの設定に与える影響を、さまざまな視点から考察する。その結果、一部の限定的な状況を除いて、バッファサイズを縮小して TCP ペーシングを用いた場合、バッファサイズを帯域遅延積分確保して TCP Reno を用いた場合よりも性能が劣化することが明らかとなった。

キーワード ルータ、バッファサイズ、Transmission Control Protocol (TCP)、TCP ペーシング

## Effect of Paced TCP on Router Buffer Sizing

Kentaro TADA<sup>†</sup>, Go HASEGAWA<sup>††</sup>, and Masayuki MURATA<sup>†††</sup>

<sup>†</sup> School of Engineering Science, Osaka University 1-3, Machikaneyama, Toyonaka, Osaka, 560-8531, Japan

<sup>††</sup> Cybermedia Center, Osaka University 1-32, Machikaneyama, Toyonaka, Osaka, 560-0043, Japan

<sup>†††</sup> Graduate School of Information Science and Technology, Osaka University 1-5, Yamadaoka, Suita, Osaka, 565-0871, Japan

E-mail: <sup>†</sup>k-tada@ics.es.osaka-u.ac.jp, <sup>††</sup>hasegawa@cmc.osaka-u.ac.jp, <sup>†††</sup>murata@ist.osaka-u.ac.jp

**Abstract** Buffer sizes for today's Internet routers is configured based on a rule-of-thumb called the *bandwidth-delay product rule*, which means a product of the link bandwidth and the average round trip time (RTT) of flows passing through the router. On the other hand, the recent research has proposed that when the number of flows is sufficiently large, the buffer size can be decreased to the bandwidth-delay product divided by the square-root of the number of flows (*sqrt-N rule*), without introducing the under-utilization of the network link. In addition, the another research has proposed that when smoothing the behavior of TCP traffic by using TCP Pacing, the buffer size can be more decreased to dozens of packets. However, these proposal has been evaluated only from the aspects of utilization of the bottleneck link. In this paper, we consider that effect of TCP Pacing on router buffer sizing, focusing on the performance of TCP connections traversing the router. Through extensive simulation experiments, we show that when the buffer size is decreased by using TCP Pacing, it would degrade the TCP performance when compared with the bandwidth-delay product rule with using TCP Reno.

**Key words** router, buffer size, TCP, TCP Pacing

### 1. はじめに

現在、インターネットルータのバッファサイズの決定には帯域遅延積を指標とする方法（以下 normal 指標と称する）が広く利用されている。これは、リンク帯域  $C$  とそのルータを通過する TCP フローの平均ラウンドトリップ時間（以下  $RTT$  と

表記する）から算出される  $B_n = C \times \overline{RTT}$  をバッファサイズとするもので、[1] によると、これによってリンク利用率を高い水準で維持しつつキューイング遅延を最小にすることができる。また [2] にも、ルータは一般に帯域遅延積で決定されるサイズのバッファを持つ、と記されている。

しかし [3] によると、normal 指標のバッファサイズを持つ

ルータを構築することは困難になっているという。現在のバックボーンネットワークのリンクの帯域幅は 2.5 Gbps や 10 Gbps であり、そこに 10000 以上のフローが流れている [4]。そのため、10 Gbps の帯域を持つリンクに流れる TCP フローの平均ラウンドトリップ時間が 250 ms であれば、normal 指標では必要とされるバッファサイズが  $10 \text{ Gbps} \times 250 \text{ ms} = 2.5 \text{ Gbits}$  となる。ルータのバッファに利用されるメモリデバイスは SRAM と DRAM の 2 種類がある。現在商用に用いられている SRAM は最大のサイズが 36 Mbits とそれほど大きくないため、必要なバッファサイズを確保するためには多数の SRAM を並列的に用いて構築しなければならない。たとえば 2.5 Gbits のバッファサイズを持つルータを構築するには、約 70 個の SRAM が必要となる。しかし、SRAM は DRAM に比べて非常に高価であるため、多数の SRAM を用いてバッファを構築することは経済的に困難である。また他にも、必要とされる設置面積が増大すること、発熱量が大きくなることなど物理的にも多くの問題がある。一方 DRAM は SRAM よりも安価であり、最大 1 Gbits のサイズを持つが、アクセス時間が 50 ns と大きく、使用するにつれてアクセス時間が 7%/年の割合で低下することなどから、やはり大容量のバッファサイズのルータを構成することは難しい。以上のことから、10 Gbps を超えるリンクにおいて、normal 指標のバッファサイズを持つルータを構築することは難しい、と述べられている [3]。

さらに [3] では、リンクを通過する TCP フローが多数存在する場合（主張では 500 程度以上）には、帯域遅延積に対してリンクを通過する TCP フロー数  $N$  の平方根で除算した値をバッファサイズとする方法（以下 sqrtN 指標と称する）、すなわち  $B_s = \frac{C \times RTT}{\sqrt{N}}$  で十分であるということを示している。この方法では流れているフロー数が 10000 であれば、バッファサイズを帯域遅延積の 100 分の 1 にすることができ、[3] で示されている結果によるとリンク利用率はほとんど低下しない。これに対して我々の研究グループでは [5] において、sqrtN 指標に基づくルータのバッファサイズ設定が、ネットワークや TCP 性能に与える影響に関する詳細な評価を行った。その結果、sqrtN 指標を用いた場合、TCP フローが多数存在するときはバッファサイズを小さくしても高いリンク利用率を得られる半面、normal 指標を用いる方法に比べてパケット廃棄率が高くなるため、ボトルネックリンクの伝播遅延時間が小さい場合や TCP コネクションで転送するデータサイズが小さい場合を除いて、TCP の性能が劣化することが明らかになった。また、ネットワーク負荷が低い場合においては、リンク帯域が大きい場合においてもパケット廃棄率がある値よりも低下しないため、データ転送遅延時間などに悪影響を及ぼす、ということを示した。

一方、[6] では、通常バースト的に送出される TCP のデータパケットを、間隔を空けて送出することによりバースト性を軽減する、TCP ペーシングを用いることで、リンク利用率は若干低下するものの、バッファサイズをさらに小さい数十パケットにすることが可能であることが示されている。これは、近年リンク帯域が飛躍的に向上しているため、リンク利用率の若干の低下によるデメリットよりも、バッファサイズを大幅に縮小することが出来ることのメリットのほうが大きいという主張に基づくものである。しかし [6] では、[3] と同様に性能指標としてボトルネックリンクの利用率のみに着目しているため、サイズの小さいデータの転送遅延時間などの、ユーザにとって重要な性能指標に関する評価は行われていない。また、パケット廃棄率については極めて基礎的な考察が行われているのみであり、シミュレーションなどによる検証は行われていない。

そこで本稿では、TCP Reno あるいは TCP ペーシングが利用される環境の下で、ルータのバッファサイズがネットワークのリンク利用率やパケット廃棄率、および各転送データサイズにおけるパケット廃棄率やデータ転送時間に与える影響を、計算機上のシミュレーションによって評価する。評価に際してはボトルネックリンクの帯域幅、データ転送を行うホスト数やバッファサイズなどをパラメータとしてシミュレーションを行い、バッファサイズ設定と TCP のさまざまな組み合わせにおける性能評価を行うことで、特に TCP のバースト性がルータのバッファサイズの決定に与える影響に着目した考察を行う。

以下、2 章では本稿において着目する TCP ペーシングについて説明する。3 章においてはシミュレーション環境として、ネットワーク構成、各種パラメータ、および評価方法について説明する。4 章ではシミュレーション結果を示し、性能評価を行う。最後に、5 章で本稿のまとめと今後の課題を示す。

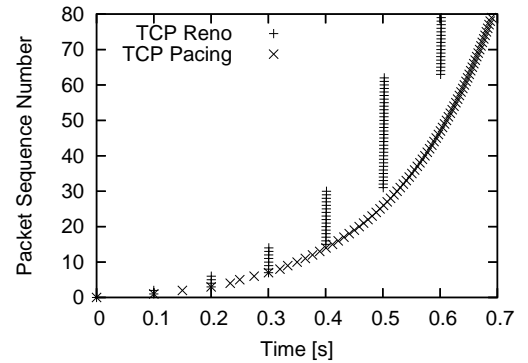


図 1 各 TCP のパケットの送信間隔

要素	生成関数	パラメータ
サイズ - Body	$p(x) = \frac{e^{-(\ln x - \mu)^2 / 2\sigma^2}}{x\mu\sqrt{2\pi}}$	$\mu = 9.357; \sigma = 1.318$
サイズ - Tail	$p(x) = \alpha k^\alpha x^{-\alpha+1}$	$k = 133K; \alpha = 1.1$
生成間隔	$p(x) = \alpha k^\alpha x^{-\alpha+1}$	$k = 1; \alpha = 1.5$

表 1 Web アクセスモデル (SURGE モデル)

## 2. TCP ペーシング

TCP ペーシングとは、送信側 TCP がデータパケットを送信する際に、ウィンドウサイズ分のパケットをバースト的に送信するのではなく、一定の間隔を持たせることで TCP トラヒックを平滑化するものである。[7] では、均一な間隔を用いてパケットを送信する TCP ペーシングを用いることにより、データパケットをバースト的に送信する通常の TCP Reno に比べて、パケット廃棄率が低下し、キューイング遅延時間が小さくなる、とされている。一方、送信パケットを分散することでキューイング遅延時間が小さくなる、ネットワーク輻輳検知が遅れる。そのため、ネットワークが輻輳状態になったときにボトルネックリンクを通過するフロー全てのパケット廃棄が同時に起こりやすくなる、とも述べられている [7]。

本稿では、[6,7] において用いられている一定の間隔でデータパケットを送信する TCP ペーシング（以下では TCP Pacing と称する）を用いる。なお本稿においては、TCP コネクションの輻輳ウィンドウサイズを  $cwnd$  とすると、TCP Pacing はデータパケット送信してから  $RTT/cwnd$  秒後に次のデータパケットを送信するものとする。図 1 に、通常の TCP Reno および TCP Pacing におけるデータパケットの送信の典型的な様子を示す。図から、TCP Reno がウィンドウサイズ分のパケットをバースト的に送出しているのに対し、TCP Pacing においては 1 ラウンドトリップ時間内に分散して転送が行われていることがわかる。

## 3. シミュレーション環境

### 3.1 ネットワークとトラヒックモデル

図 2 に示すシングルボトルネックのネットワークモデルを仮定し、送信ホスト数と受信ホスト数である  $n$ 、ボトルネックリンクの帯域  $C$  および伝播遅延時間  $D$ 、ボトルネックリンクの出力バッファサイズを変化させながらシミュレーションを行う。なお、アクセスリンク帯域はボトルネックリンク帯域と等しいものとし、アクセスリンクの伝播遅延時間は 5 ms とする。各ホストから送出されるトラヒックには、各ホストが時間オーダでは無限大の転送データサイズを持つとした P2P トラヒック、および [8] で述べられている表 1 の SURGE モデルに従って生成された Web トラヒックを想定する。シミュレーションには ns-2 [9] を用いるものとし、TCP ペーシングに関しては [10] で公開されているシミュレーションコードを利用する。

### 3.2 シミュレーション結果の評価方法

ボトルネックルータのリンクにおいて、通過するパケットの挙動を観測し、単位時間当たり通過したパケットの数とリンク帯域からリンクの利用率を求める。また、パケット廃棄率はルータに到着したパケット数とバッファあふれによって廃棄されたパケット数から導出する。Web トラヒックを用いたシミュレーションの場合には、各転送データサイズにおけるデータ転送遅延時間とパケット廃棄率をあわせて性能指標として用い

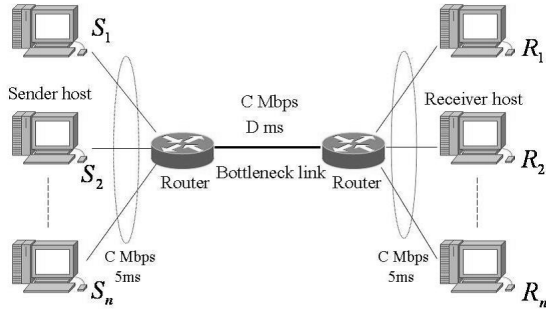
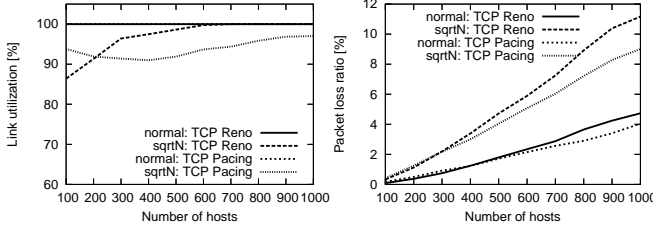


図2 ネットワークモデル



(a) リンク利用率

(b) パケット廃棄率

図3 P2P トラフィックの場合のリンク利用率とパケット廃棄率

る。データ転送遅延時間は、各フローが転送するデータの最初のパケットの送信が開始されてから、最後のパケットに対するACKを受信するまでの時間とし、各ホストのパケット送受信ログより求める。また、転送したデータサイズを記録し、送信ホストのデータの転送が終了するまでに送信したパケット数と各転送において廃棄されたパケット数から各転送データサイズにおけるパケット廃棄率を導出する。以上のリンク利用率、パケット廃棄率、各転送データサイズにおけるデータ転送遅延時間、およびパケット廃棄率を性能評価指標として用いる。

## 4. シミュレーション結果と考察

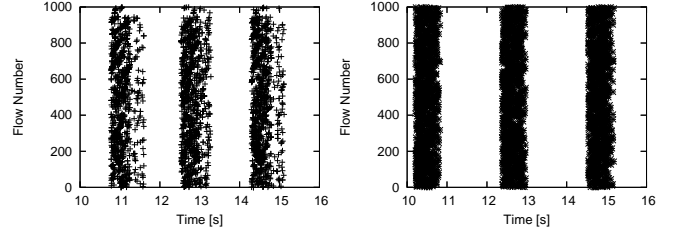
### 4.1 P2P トラフィック

#### 4.1.1 リンク利用率とパケット廃棄率

図3は、 $C = 100$  Mbps、 $D = 90$  ms のネットワークに P2P トラフィックを与えた環境において、ホスト数  $n$  を変化した場合のリンク利用率およびパケット廃棄率の変化を示したものである。また、図4は、 $n = 1000$  としたときに、 $x$  軸を時間、 $y$  軸をフロー番号とし、 $y$  番目のフローのパケットが時刻  $x$  にルータバッファで廃棄されたときに  $(x, y)$  に点を打ったグラフであり、各フローのパケット廃棄の発生が同期的に発生しているか否かを調べたものである。図3(a)から、sqrtN 指標の TCP Reno はホスト数が増加するとほぼ 100% のリンク利用率を示しているが、sqrtN 指標の TCP Pacing はホスト数が増加してもリンク利用率が上昇しない。これは、図4(c)より、sqrtN 指標の TCP Reno は各フローのパケット廃棄の発生が同期していないため、文献[3]における議論が前提としている「TCP コネクションが同期しない」という仮定が成立し、ホスト数が多い場合はリンク利用率を高く保つことができるためである。しかし、図4(d)より、TCP Pacing ではバッファサイズに sqrtN 指標を用いた場合でも各フローのパケット廃棄の発生が同期しているため、ホスト数が多い場合でもリンク利用率を高く保つことができない。また、図4(a)と図4(b)、図4(c)と図4(d)をそれぞれ比較すると、TCP Pacing を用いることでフロー間の同期がより強くなることもわかる。パケット廃棄率においては、図3(b)より、バッファサイズに関係なく、TCP ペーシングの本来の目的である、バースト性の緩和によるパケット廃棄率の低下の効果が出ていることが確認でき、ホスト数が増加するにつれその効果は大きくなることわかる。

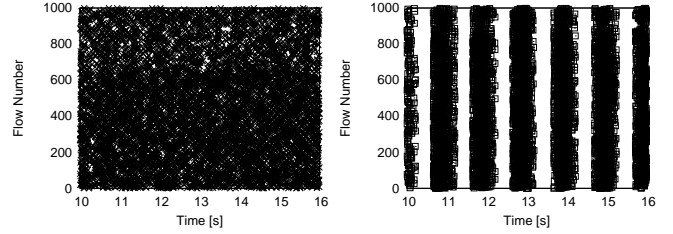
#### 4.1.2 バッファサイズ設定による影響

図5は、 $C = 100$  Mbps、 $D = 90$  ms のネットワークに P2P トラフィックを与えた環境において、それぞれ  $n = 500$  および



(a) normal 指標の TCP Reno

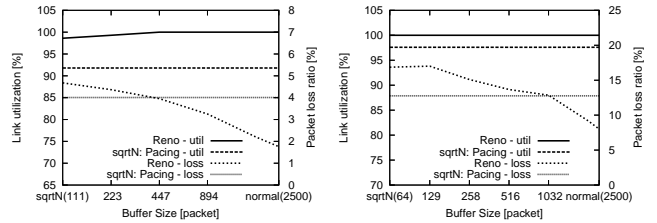
(b) normal 指標の TCP Pacing



(c) sqrtN 指標の TCP Reno

(d) sqrtN 指標の TCP Pacing

図4 各フローのパケット廃棄発生時刻



(a)  $n = 500$  の場合

(b)  $n = 1500$  の場合

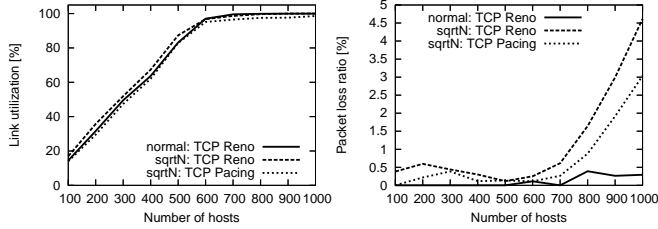
図5 バッファサイズ設定によるリンク利用率とパケット廃棄率の影響

$n = 1500$  として、バッファサイズを変化させた場合の TCP Reno、および sqrtN 指標の TCP Pacing の、リンク利用率とパケット廃棄率の変化を示したものである。図5(a)においては、バッファサイズを normal 指標の約 5 分の 1 にして TCP Reno を用いると、リンク利用率はほぼ 100% となり、パケット廃棄率は sqrtN 指標の TCP Pacing とほぼ同等である。しかし、図5(b)では、TCP Reno のパケット廃棄率が sqrtN 指標の TCP Pacing と同等になるのはバッファサイズが normal 指標の約 5 分の 2 のしたときである。これは、P2P トラフィックを与えた環境においては、TCP ペーシングによるパケット廃棄率の低下の効果が大きくなるため、TCP Reno のパケット廃棄率が sqrtN 指標の TCP Pacing のパケット廃棄率と同等になるために必要なバッファサイズが増大することによる。

### 4.2 Web トラフィック

#### 4.2.1 リンク利用率とパケット廃棄率

図6は、 $C = 100$  Mbps、 $D = 90$  ms のネットワークに Web トラフィックを与えた環境において、ホスト数  $n$  を変化した場合のリンク利用率およびパケット廃棄率の変化を示したものである。図6(a)より、ホスト数に関係なく、normal 指標の TCP Reno、sqrtN 指標の TCP Reno および sqrtN 指標の TCP Pacing のリンク利用率にほとんど差はない。これは、Web トラフィックは表1の SURGE モデルに従って生成されることにより、TCP フローが少ない場合でも各フローは同期して動作しないため、sqrtN 指標を用いた場合でも normal 指標と同等のリンク利用率を保つことができると考えられる。一方、パケット廃棄率は、P2P トラフィックの場合と同様に TCP Pacing を用いることで若干減少している。しかし、normal 指標の TCP Reno ではホスト数が 700 程度より小さい場合はパ



(a) リンク利用率 (b) パケット廃棄率

図 6 Web トラフィックの場合のリンク利用率とパケット廃棄率

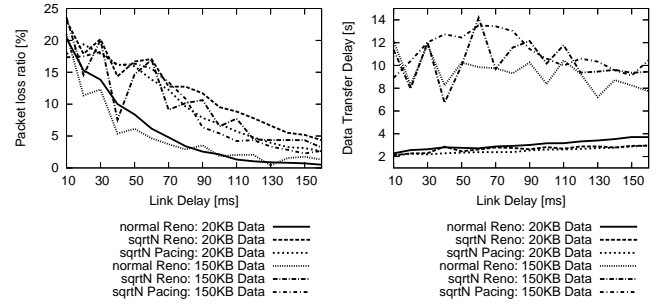
ケット廃棄率がほぼ 0 % であるのに対して、sqrtN 指標では TCP の種類に関係なく、ホスト数が少ない場合でもパケット廃棄が発生している。そのため、sqrtN 指標を用いた場合、ボトルネックリンクの利用率は normal 指標と同程度にすることができても、パケット廃棄が発生するため、個々のフローに着目すると性能が劣化しているといえる。

#### 4.2.2 ボトルネックリンクの伝播遅延時間の影響

図 7 は、 $C = 100$  Mbps のネットワークに Web トラフィックを与えた環境において、 $n = 1500$  としてボトルネックリンクの伝播遅延時間  $D$  を変化させた場合の、20KByte および 150KByte のデータ転送におけるパケット廃棄率およびデータ転送遅延時間の変化を示したものである。図 7(a) より、ボトルネックリンクの伝播遅延時間が小さい場合は sqrtN 指標の TCP Reno のパケット廃棄率にばらつきがあるものの、ボトルネックリンクの伝播遅延時間がある程度大きくなると、パケット廃棄率は sqrtN 指標の TCP Reno が最も高く、normal 指標の TCP Reno が最も低い。しかし、ボトルネックリンクの伝播遅延時間が 10 ms の場合は sqrtN 指標の TCP Pacing のパケット廃棄率が最も低くなっている。これは、RTT が小さくなるとパケットを送信してから ACK が返ってくる時間が短くなるため、輻輳ウィンドウサイズの増加速度が上がり、TCP Reno はそのバースト性によりパケット廃棄率が增大しているものと思われる。また、図 7(b) より、データ転送遅延時間においてはボトルネックリンクの伝播遅延時間が小さい場合、もしくは転送データサイズが小さい場合は normal 指標の TCP Reno が最も大きい。これは sqrtN 指標を用いた場合、キューイング遅延時間が減少するためである。図 7(a) より、ボトルネックリンクの伝播遅延時間が小さい場合はパケット廃棄率の差が小さくなるため、パケットの再送にかかる時間やパケット廃棄による輻輳ウィンドウサイズの減少によるデータ転送遅延時間の差が小さくなるのがわかる。また、転送データサイズが小さい場合は、そのデータを送信している間に輻輳ウィンドウサイズがあまり大きくならないため、キューイング遅延時間がデータ転送時間に与える影響が大きくなる。そのため、ボトルネックリンクの伝播遅延時間が非常に小さい場合や、転送データサイズが小さい場合は、sqrtN 指標で TCP Pacing を用いた場合が最もデータ転送遅延時間が小さくなる。しかし、それ以外の場合は normal 指標の TCP Reno のデータ転送遅延時間が最も小さく、TCP Pacing を用いても、sqrtN 指標の TCP Reno とほぼ同じである。

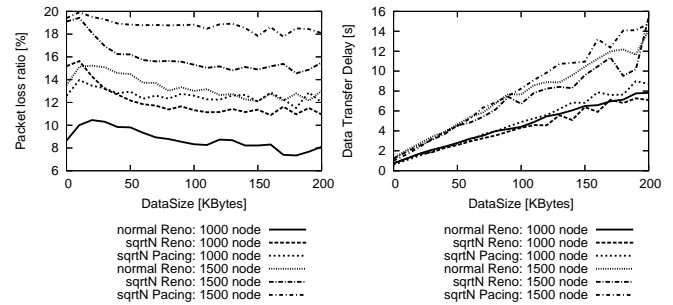
#### 4.2.3 ネットワークトラフィック量の影響

図 8 および図 9 は、それぞれ  $C = 100$  Mbps、 $D = 20$  ms および  $C = 100$  Mbps、 $D = 90$  ms のネットワークに Web トラフィックを与えた環境において、 $n = 1000$  および  $n = 1500$  とした場合の各転送データサイズにおけるパケット廃棄率およびデータ転送遅延時間の変化を示したものである。図 8(a) および図 9(a) より、転送データサイズやホスト数に関係なく、パケット廃棄率は normal 指標の TCP Reno が最も低いことがわかる。図 8(a) では、ホスト数が増加すると、normal 指標の TCP Reno と sqrtN 指標の TCP Reno のパケット廃棄率の差は小さくなっているのに対して、sqrtN 指標の TCP Pacing と normal 指標の TCP Reno のパケット廃棄率の差は拡大している。図 9(a) においても、ホスト数が増加すると、sqrtN 指標の TCP Pacing と normal 指標の TCP Reno のパケット廃棄率の差は拡大しているため、sqrtN 指標の TCP Pacing を normal 指標の TCP Reno と比較すると、ホストが多数存在する場合に、より性能が劣化するといえる。また、図 8(a) においては、ホ



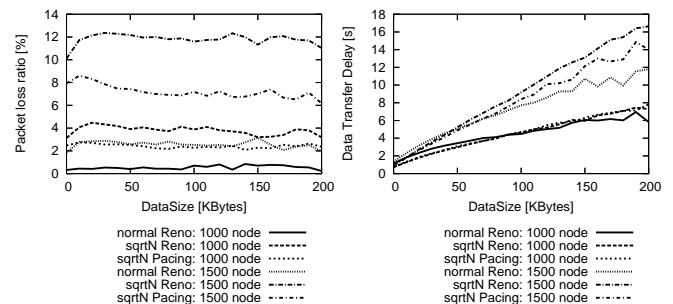
(a) パケット廃棄率 (b) データ転送遅延時間

図 7 ボトルネックリンクの伝播遅延時間の影響



(a) パケット廃棄率 (b) データ転送遅延時間

図 8  $D = 20$  ms の場合の Web トラフィックの性能



(a) パケット廃棄率 (b) データ転送遅延時間

図 9  $D = 90$  ms の場合の Web トラフィックの性能

スト数にかかわらず、転送データサイズが小さい場合、sqrtN 指標の TCP Reno のパケット廃棄率が高い。これは、sqrtN 指標ではバッファサイズが小さく、TCP Reno のバースト性により転送データのほとんどのパケットが廃棄されることがあるためである。一方、パケットを分散して送信する TCP Pacing ではバッファサイズ設定に sqrtN 指標を用いても、転送データサイズが小さい場合にパケット廃棄率が高くない。また、図 8(b)、9(b) より、ホスト数が増加すると、データ転送遅延時間も増加していることがわかる。これはパケット廃棄率が増加するためであり、図 9(b) において、ホスト数が増加するとデータ転送遅延時間の差が拡大することも、パケット廃棄率の差が拡大することに起因すると考えられる。

#### 4.2.4 ボトルネックリンクの帯域の影響

図 10 および図 11 は、それぞれ  $D = 20$  ms および  $D = 90$  ms のネットワークに Web トラフィックを与えた環境において、 $n = 1500$  としてボトルネックリンクの帯域  $C$  を変化させた場合のリンク利用率およびパケット廃棄率の変化を示している。

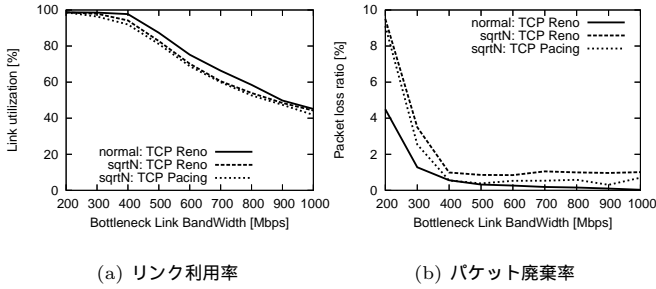


図 10  $D = 20$  ms の場合の Web トラヒックの性能

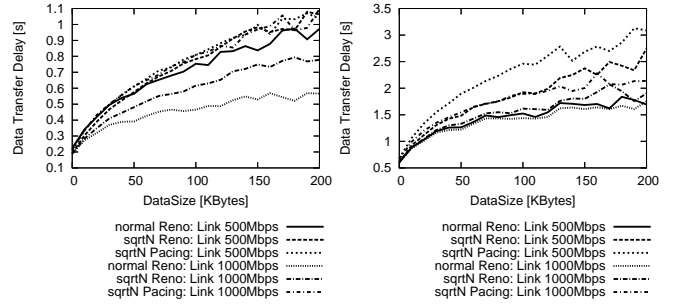


図 12 データ転送遅延時間

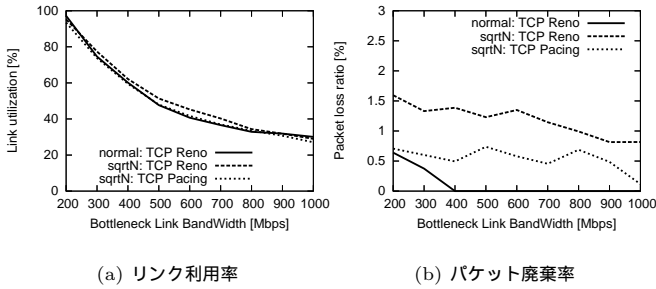
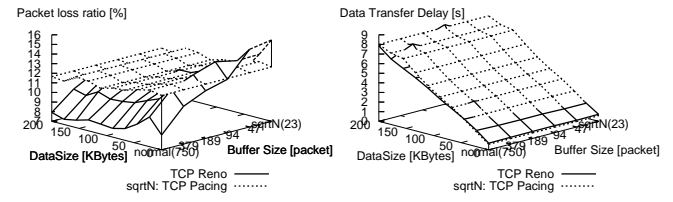


図 11  $D = 90$  ms の場合の Web トラヒックの性能



(a)  $n = 1000$  の場合のデータ転送遅延時間 (b)  $n = 1000$  の場合のデータ転送遅延時間

(c)  $n = 1500$  の場合のデータ転送遅延時間 (d)  $n = 1500$  の場合のデータ転送遅延時間

図 13  $C = 100$  Mbps,  $D = 20$  ms の場合のバッファサイズ設定による影響

また、図 12 は、 $C = 500$  Mbps および  $C = 1000$  Mbps の場合の各転送データサイズにおけるデータ転送遅延時間の変化を示したものである。図 10(a)、11(a) より、リンク利用率は 4.2.1 節のホスト数を変化させた場合と同様にほとんど差はない。しかし、図 10(b)、11(b) より、normal 指標の TCP Reno はボトルネックリンクの帯域を大きくすることでパケット廃棄率が 0% になるのに対し、sqrtN 指標の TCP Reno ではある値より小さくならないのがわかる。文献 [5] では、sqrtN 指標を用いた場合、パケットがバースト的に到着した場合にそれらのパケットをバッファで吸収することができないため、ボトルネックリンクの帯域が大きい場合においてもパケット廃棄が発生する、と述べられている。このことは TCP Pacing を用いた場合も同様である。すなわち、TCP Pacing を用いることによって、バースト性が軽減されるため、sqrtN 指標の TCP Reno に比べてパケット廃棄率が低下しているものの、ボトルネックリンクの帯域が大きい場合においてもパケット廃棄率は 0% にならない。さらに、図 12 より、ボトルネックリンクの帯域が大きい場合、sqrtN 指標の TCP Pacing のデータ転送遅延時間が最も大きいことがわかる。これは、TCP Pacing はパケットの送信間隔を空けることでバースト性を軽減しているが、ボトルネックリンクの帯域が大きく、パケット廃棄がほとんど発生しない環境においては、パケットの送信間隔を空けることによって TCP Reno に比べてパケットの送信が遅れ、データ転送遅延時間に悪影響を及ぼすと考えられる。

#### 4.2.5 バッファサイズ設定による影響

図 13 および図 14 は、それぞれ  $C = 100$  Mbps,  $D = 20$  ms、および  $C = 100$  Mbps,  $D = 90$  ms のネットワークに Web トラヒックを与えた環境において、 $n = 1000$  および  $n = 1500$  として、バッファサイズを変化させた場合の TCP Reno、および sqrtN 指標の TCP Pacing の、各転送データサイズにおけるパケット廃棄率およびデータ転送遅延時間の変化を示したものである。また、図 15 および図 16 は、それぞれ  $C = 1000$  Mbps,  $D = 20$  ms、および  $C = 1000$  Mbps,  $D = 90$  ms のネットワークに Web トラヒックを与えた環境において、 $n = 1500$  として、バッファサイズを変化させた場合の TCP Reno、および sqrtN 指標の TCP Pacing の、各転送データサイズにおけるリンク利用率とパケット廃棄率およびデータ転送遅延時間の変化を示している。バッファサイズを変化させて TCP Reno を用いた場合、図 13 においては、バッファサイズを少し大きくすると、転送データサイズが小さい場合にパケット廃棄率が高くなるという現象が見られなくなることがわかる。また、

図 14(a)、14(b) より、バッファサイズを normal 指標の約 4 分の 1 としたとき、TCP Reno のパケット廃棄率が sqrtN 指標の TCP Pacing を下回るが、バッファサイズを大きくすることで、転送データサイズが 50-100KByte 程度のデータ転送遅延時間が sqrtN 指標の TCP Pacing より大きくなっていくことがわかる。図 14(c)、14(d) では、パケット廃棄率の差が大きいため、バッファサイズが小さい場合、TCP Reno のほうがデータ転送遅延時間が大きくなるが、バッファサイズを normal 指標の約 5 分の 1 にすると、パケット廃棄率は sqrtN 指標の TCP Pacing を下回り、転送データサイズが大きい場合のデータ転送遅延時間も TCP Reno のほうが小さくなる。ボトルネックリンクの帯域を大きくした場合は、図 15(a)、16(a) より、sqrtN 指標の TCP Reno と TCP Pacing のパケット廃棄率にほとんど差がないため、TCP Reno のパケット廃棄率が sqrtN 指標の Pacing と同程度となるために必要なバッファサイズはそれほど大きくない。しかし、normal 指標のようにパケット廃棄率が 0% となるには、かなり大きなバッファサイズが必要であることがわかる。また、図 15(b)、16(b) より、データ転送遅延時間においては、バッファサイズが増加するにつれ、TCP Reno のデータ転送遅延時間は小さくなるため、sqrtN 指標の TCP Pacing との差は拡大する。

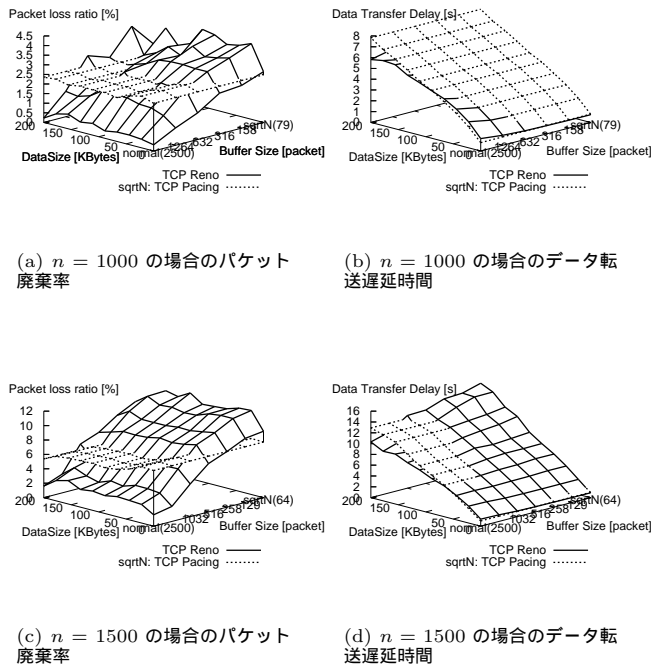


図 14  $C = 100$  Mbps,  $D = 90$  ms の場合のバッファサイズ設定による影響

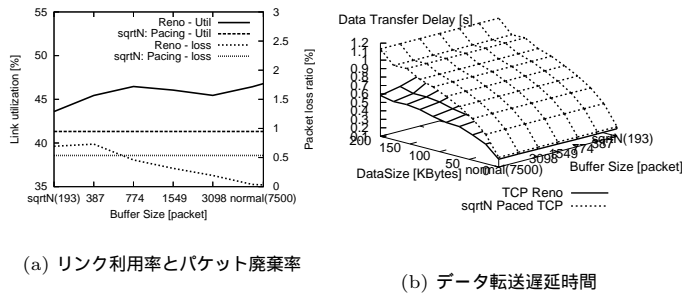


図 15  $C = 1000$  Mbps,  $D = 20$  ms,  $n = 1500$  の場合のバッファサイズ設定による影響

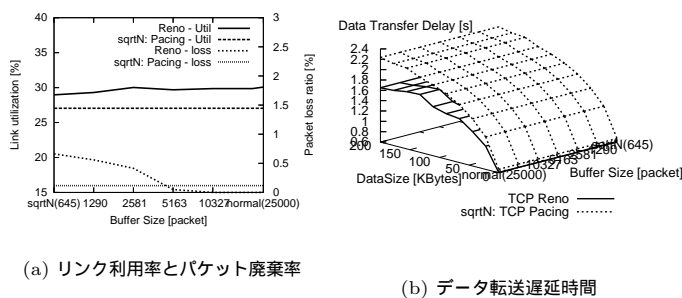


図 16  $C = 1000$  Mbps,  $D = 90$  ms,  $n = 1500$  の場合のバッファサイズ設定による影響

## 5. おわりに

本稿では ns-2 によるシミュレーションを用いて、TCP のパースト性がルータのバッファサイズの決定に与える影響の評価を

行った。その結果、バッファサイズ設定に sqrtN 指標を用いた上で TCP Reno と TCP Pacing を比較すると、ほとんどの場合において、TCP Pacing の導入によりパケット廃棄率が小さくなることが明らかとなった。しかし、P2P トラフィックを用いた場合には、TCP フローが多数存在する場合でも同期現象が原因となりリンク利用率が上昇せず、ボトルネックリンクの伝播遅延時間が非常に小さい場合や、転送データサイズが小さい場合を除いて、normal 指標の TCP Reno に比べて性能が劣ることがわかった。特に、ボトルネックリンクの帯域が大きいような状況で TCP Pacing を用いると、パケット廃棄率の低下の効果が小さく、TCP Pacing はパケットの送信間隔を空けることによって TCP Reno に比べてパケットの送信が遅れ、データ転送遅延時間に悪影響を及ぼす。

また、sqrtN 指標から normal 指標までバッファサイズを変化させて TCP Reno を用いた場合と sqrtN 指標の TCP Pacing の性能を比べた結果、バッファサイズを normal 指標の 4~5 分の 1 程度にしたときの TCP Reno と sqrtN 指標の TCP Pacing のパケット廃棄率がほぼ同じであった。一方、バッファサイズが増加するとキューイング遅延時間が増大するため、転送データサイズが小さい場合のデータ転送遅延時間は sqrtN 指標の TCP Pacing のほうが小さい。しかし、sqrtN 指標の TCP Reno と sqrtN 指標の TCP Pacing のパケット廃棄率に差が出にくい環境でデータサイズの大きいパケットを転送する場合は、バッファサイズが増加するにつれ、TCP Reno のデータ転送遅延時間は小さくなるため、sqrtN 指標の TCP Pacing との差は大きくなる。そのため、転送データサイズが小さい、など、一部の限定的な状況では sqrtN 指標で TCP Reno を用いるよりも sqrtN 指標で TCP Pacing を用いたほうが良いが、それ以外では sqrtN 指標の TCP Reno と比べてもほとんど性能は改善せず、バッファサイズを sqrtN 指標よりも大きくして TCP Reno を用いたほうが良い場合が多いことが明らかになった。

今後は、TCP Reno と TCP Pacing が混在する環境が各 TCP の性能に及ぼす影響を考察したい。また、送信間隔にランダム性を導入した TCP ペーシングを用いた場合や、RTT の異なるコネクションが混在した場合の評価を行う予定である。

## 文 献

- [1] C. Villamizar and C. Song, "High performance TCP in ANSNET," *SIGCOMM Computer Communications Review*, vol. 24, pp. 45–60, Oct. 1994.
- [2] R. Bush and D. Meyer, "Some Internet architectural guidelines and philosophy," *RFC 3439*, Dec. 2003.
- [3] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols For Computer Communications*, Sept. 2004.
- [4] C. J. Fraleigh, *Provisioning Internet backbone Networks to support latency sensitive applications*. PhD thesis, Stanford University, Department of Electrical Engineering, June 2002. Adviser-Fouad A. Tobagi.
- [5] 富岡 健史, 長谷川 剛, 村田 正幸, "ルータのバッファサイズが TCP に与える影響の一考察," 電子情報通信学会技術研究報告, vol. 105, pp. 431–436, Mar 2006. IN2005-229.
- [6] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: Routers with very small buffers," *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 83–90, July 2005.
- [7] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the performance of TCP pacing," in *Proceedings of IEEE INFOCOM 2000*, pp. 1157–1165, Mar. 2000.
- [8] P. Barford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation," in *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp. 151–160, July 1998.
- [9] T. V. Project, "UCB/LBNL/VINT network simulator - ns (version 2)." <http://www.isi.edu/nsnam/ns/>.
- [10] D. X. Wei, "A TCP pacing implementation for NS2." <http://www.cs.caltech.edu/~weix1/technical/ns2pacing/index.html>.