

スケーラビリティおよび公平性を考慮したTCPバッファ制御

松尾 孝広† 長谷川 剛‡ 村田 正幸† 宮原 秀夫†

† 大阪大学基礎工学研究科情報数理系専攻

〒 560-8531 大阪府豊中市待兼山 1-3

Phone: 06-6850-6588, Fax: 06-6850-6589

E-mail: {t-matuo,murata,miyahara}@ics.es.osaka-u.ac.jp

‡ 大阪大学経済学部

〒 560-0043 大阪府豊中市待兼山町 1-7

Phone: 06-6850-5233

E-mail: hasegawa@econ.osaka-u.ac.jp

あらまし ネットワークの輻輳問題に対しては、これまでも様々な解決策が提案・検討されているが、エンドホストに関してはこれまであまり検討がなされていない。しかし、インターネットユーザの増加に伴うネットワーク回線の急激な増強を考慮すると、エンドホストの高速化・高機能化は重要である。そこで本稿では、TCP によるデータ転送効率とコネクション間における公平性の向上を目的とした、サーバホストにおける新たなバッファ制御方式である Scalable Automatic Buffer Tuning (SABT) 方式を提案する。SABT 方式においては、送信側端末がまず接続されている各 TCP コネクションの平均スループットを解析的に推測し、その値に基づいて各コネクションに割り当てる送信ソケットバッファサイズを決定する。バッファが不足する場合には Max-Min Fairness を考慮しつつバッファ割り当てを行う。また、各コネクションのバッファサイズ及びウィンドウサイズの初期値は WWW トラフィックの特性を考慮して決定する。本稿ではこれらの提案方式の有効性を、シミュレーション及び実コンピュータ上に実装しての稼働実験によって検証し、従来の方式に比べスループット、公平性の両面で優れていることを明らかにしている。

キーワード スケーラビリティ、公平性、TCP (Transmission Control Protocol)、TCP バッファ割り当て、World Wide Web

Scalable Automatic Buffer Tuning to Provide High Performance and Fair Service for TCP Connections

Takahiro Matsuo† Go Hasegawa‡ Masayuki Murata† Hideo Miyahara†

†Department of Infomatics and Mathematical Science

Graduate School of Engineering Science, Osaka University

1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan

Phone: +81-6-6850-6588, Fax: +81-6-6850-6589

E-mail: {t-matuo,murata,miyahara}@ics.es.osaka-u.ac.jp

‡Faculty of Economics, Osaka University

1-7, Machikaneyama, Toyonaka, Osaka 560, Japan

Phone: +81-6-6850-5233

E-mail: hasegawa@econ.osaka-u.ac.jp

Abstract Although many research efforts have been devoted to the network congestion against an increase of network traffic in the Internet, only a few discussions on the performance improvement of the endhosts are recently made. In this paper, we propose a new architecture, which is called Scalable Automatic Buffer Tuning (SABT), to provide high performance and fair service for many TCP connections at the Internet endhosts. In SABT, the sender host estimates 'expected' throughput of the TCP connections through simple mathematical equations, and assign the buffer to them according to the estimated throughput. If the socket buffer is short, the max-min fairness policy is utilized to allocate the buffer. Since SABT uses the measurement-based approach to estimate the throughput, an initial values of the buffer size cannot be determined. Thus, an initial buffer size of the connection is determined by taking account of the characteristics of WWW traffic. We confirm an effectiveness of our proposed algorithm through both of a simulation technique and an actual experimental system where we implement our algorithm.

Keyword scalability, fairness, TCP (Transmission Control Protocol), buffer assignment, World Wide Web

1 はじめに

ネットワークトラフィックの増加に伴うネットワークの輻輳に対して、これまでも様々な解決案が提案・検討されている。しかし、例えばアクセス頻度の高い WWW (World Wide Web) サーバではピーク時には毎秒数百のドキュメントに対するリクエストを処理しなければならなくなっており、ネットワークではなくエンドホスト自身がデータ転送全体のボトルネックになる状況は既に発生している。従って、エンドホストの高速化、高機能化も併せて重要である。

エンドホストの高速化のためのプロトコル処理の軽減方法としては、例えば [1] で提案されている 'fbuf' (fast buffer) 方式が挙げられる。この方式は、エンドホストにおいてカーネル領域とユーザ領域のメモリを共有することによって、エンドシステムにおける TCP データ転送の際のボトルネックの一つであるメモリコピーを省略し、エンドホストの高速化を実現している。しかし、[1] を始めとする様々な提案方式の多くは、エンドホストに接続している各コネクションへの有効な割り当てバッファ量の制御、及びコネクション間の公平性は考慮していない。例えば、送信側ホストに 64Kbps のリンクで接続しているユーザ A と、100Mbps のリンクで接続しているユーザ B が存在する時に、これら 2 つのコネクションに送信バッファを等分して割り当てると、各コネクションに必要な送信バッファサイズが異なるため、ユーザ A にとってはバッファが大きい、あるいはユーザ B にとってバッファサイズが小さいという状況が発生し得る。このような場合ではユーザ A に割り当てられたバッファの内、使用されていないバッファをユーザ B に割り当てることによって双方のコネクションが高スループットを得られる。特に、アクセス頻度の高い WWW サーバにおいては、同時に処理する TCP コネクション数も多く、それらの TCP コネクションはラウンドトリップ時間、帯域等がそれぞれ異なるため、それらを考慮した公平な送信バッファの割り当てが必要になる。[2] においては、各 TCP コネクションのウィンドウサイズに基づいて要求バッファサイズを決定し、動的にバッファ割り当て量を変化させる Automatic Buffer Tuning (ABT) 方式を提案している。しかし、TCP のウィンドウサイズは変動が大きく、また必ずしも TCP のウィンドウサイズがスループットに比例するとは限らないため、ABT 方式では公平かつ効率的なバッファ割り当てが行えない。

そこで本稿では、送信側端末において公平かつ効率的なバッファ割り当てを行うアルゴリズムである Scalable Automatic Buffer Tuning (SABT) 方式を提案する。本方式では、一定時間毎に各 TCP コネクションのスループットを p (パケットロス率)、RTT (ラウンドトリップ時間)、RTO (再送タイムアウト時間) の値から数学的解析手法により推測し、その値に基づき各コネクションに送信バッファを割り当てる。また、バッファが不足する場合にはコネクション間の Max-Min Fairness を考慮して割り当てバッファサイズを決定する。なお、転送ファイルサイズの小さなコネクションについては、パケットロス率、RTT、RTO の正確な測定が困難であるため、スループットの推測が不安定になる。そこで、SABT 方式では、このようなコネクションをバッファ割り当てアルゴリズムの対象外とし、適切な初期バッファサイズ及びウィンドウサイズを設定する。ただし、紙面の制限上、その効果については省略する。

以下、2 章では ABT 方式と提案方式である SABT 方式

の説明を行う。3 章でコンピュータ上でのシミュレーションによる提案方式の性能評価を行い、4 章において提案方式を実コンピュータ上に実装し、稼働実験による性能評価を行う。最後に 5 章で本稿の結論を述べる。

2 送信バッファサイズの制御方法

2.1 関連研究: Automatic Buffer Tuning [2]

文献 [2] では、送信側端末のバッファ制御方式の一つとして Automatic Buffer Tuning (ABT 方式) と呼ばれる方式が提案されている。これは、各 TCP コネクションの現在のウィンドウサイズの変動に応じて割り当てバッファサイズを変化させる方式である。もし、すべての TCP コネクションに必要なバッファサイズの合計値が送信側端末の持つ送信バッファサイズよりも大きい場合には、Max-Min Fairness に基づいた公平なバッファ割り当てを行う。すなわち、送信側端末はまずバッファを各コネクションに対して均等に割り当てようと試みた後、割り当てられたバッファが要求量よりも大きいコネクションが存在する場合には、その余剰バッファを他のより大きなバッファを要求するコネクションに均等に再割り当てする。ABT 方式においては以上の処理によりコネクションごとのバッファの要求量の違いを考慮した公平なバッファ割り当ての実現を試みている。

2.2 提案方式: Scalable Automatic Buffer Tuning

2.2.1 安定かつ公平なバッファ割り当て

ABT 方式は各コネクションの TCP のウィンドウサイズに基づいてバッファサイズを決定するため、次章で示す様にバッファ割り当ての非安定性が問題になる。そこで SABT 方式では、ウィンドウサイズ以外の 3 つのパラメータから各コネクションのスループットを推測し、その値に基づいて要求バッファサイズを決定する。

TCP コネクションのスループットの推測には [3] における解析結果を利用する。[3] では、複数のコネクションが異なるリンク帯域でボトルネックのルータに接続されており、そのルータが RED アルゴリズム [4] を実装している場合を想定し、各コネクションの平均スループットを解析的に導出している。解析では以下の 3 つのパラメータを用いている。

- p : RED アルゴリズムにおけるパケットロス率
- rtt : TCP コネクションの RTT (Round Trip Time) の平均値
- rto : TCP コネクションの RTO (Retransmission Time Out) の平均値

この解析結果を、RED アルゴリズムのパケットロス率 p を送信側端末で観測するパケットロス率とみなすことにより、SABT 方式のスループット推測に適用する。また、3 つのパラメータは以下の様にして送信側端末で観測することができる。 rtt と rto に関しては送信側端末の TCP が保持している変数を直接利用する。また p については、転送を完了したパケット数と、ACK パケットを基に推測するパケットロス数から算出する。その際、文献 [3] における解析は、RED ルータによってパケットロスがランダムに発生することを仮定しているため、Drop-tail ルータによってパースト的にパケットロスが発生する場合には推測値に大きな誤差が生じる可能性がある [4]。この誤差に関する評価は 3 章において行う。

各コネクション i の平均スループット $\bar{\rho}_i$ が求まると、各コネクションの要求バッファサイズ B_i を、

$$B_i = \bar{\rho}_i \times rtt_i \quad (1)$$

と決定する。ここで rtt_i はコネクション i の RTT である。

また、SABT 方式では ABT 方式と同様にバッファ割り当ての際に Max-Min Fairness を考慮するが、ABT 方式では余剰バッファを各コネクションに均等に再割り当てしているのに対し、SABT 方式では解析的手法によって導出した各コネクションのスループットに比例して再割り当てを行う。こうすることによって余剰バッファの効率的な再割り当てが期待できる。

2.2.2 サイズの小さいファイルの転送時間の改善

SABT 方式は、送信側端末において観測したパラメータを用いて各 TCP コネクションのスループットを推測する。従って、転送するデータサイズが小さい TCP コネクションに対してはパラメータの観測が十分に行えず、正確にスループットが推測できないため、バッファ割り当てが不正確になることが考えられる。特に WWW サーバを考慮すると、[5] にある様に WWW サーバの平均ドキュメントサイズは 10 [KBytes] 未満と非常に小さいため、スループット推測の不正確性の影響が大きくなる。

そこで SABT 方式では、転送ファイルサイズが小さいコネクションに関しては、提案方式によるバッファ割り当てアルゴリズムの対象から外し、バッファサイズ及びウィンドウサイズの初期値を、転送ファイルサイズに等しい値に設定する。その結果、サイズの小さいファイルはコネクション設定後 1 RTT で転送することができるため、小さい WWW ドキュメントの転送時間を短縮できると共に、大きなファイルを転送するコネクションに対しては安定かつ公平なバッファ割り当てが実現できると考えられる。

3 シミュレーションによる評価

本章ではネットワークシミュレーションツール ns [6] を用いたシミュレーション結果を示し、バッファ割り当てアルゴリズムの性能評価を行う。評価にあたっては、以下の 3 つのバッファ割り当てアルゴリズムを対象とする。

- EQ 方式: 各コネクションに対して等分でバッファサイズの割り当てを行う
- ABT 方式: ABT 方式 (2.1 節) によるバッファ割り当てを行う
- SABT 方式: SABT 方式 (2.2 節) によるバッファ割り当てを行う

評価に際しては、ルータのパケット処理方式として RED 方式 [4] 及び Drop-tail 方式を対象とした。SABT 方式が利用している文献 [3] に示された解析は、RED 方式によるランダムなパケットロスをもとに仮定しているため、RED 方式を用いた場合は各コネクションのスループットを正確に推測できると考えられる。一方 Drop-tail 方式を用いた場合は、ルータのバッファ溢れによってパケットロスが発生するため、パケットロスがバースト的に発生し、SABT 方式のスループット推測が不正確になる。そこで本章では、Drop-tail 方式のルータを用いた場合の結果を示す。なお、RED ルータの場合は、他の方式に比べて SABT 方式が有効であることが確認されている。

本章で用いるネットワークモデルを図 1 に示す。モデルは 1 つの送信側端末、7 つの受信側端末及び 6 つのルータが

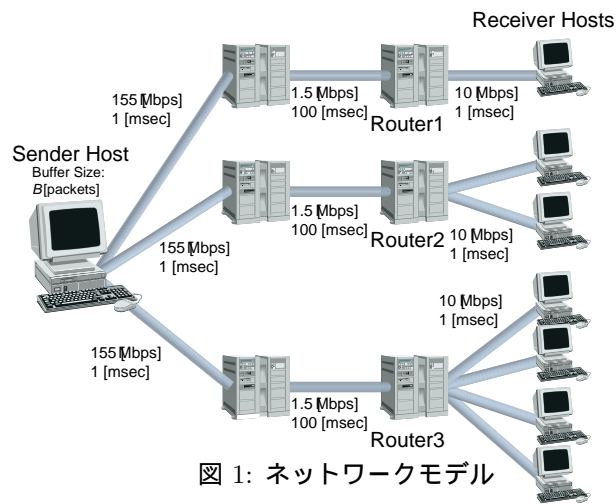


図 1: ネットワークモデル

ら構成され、7 本の TCP コネクションが送信側端末と各受信側端末の間で転送を行うものとする。送信側端末とルータ間のリンク、ルータ間のリンク、及びルータと受信側端末間のリンクの帯域はそれぞれ 155 [Mbps]、1.5 [Mbps]、及び 10 [Mbps] とし、伝播遅延時間はそれぞれ 1 [msec]、100 [msec]、1 [msec] とする。シミュレーションにおいては、7 本のコネクション i ($1 \leq i \leq 7$) が $t = (i-1) \times 500$ [sec] にデータ転送を開始し、 $t = 5000$ [sec] で転送を終了するものとする。従って、7 本のコネクションすべてがデータ転送を開始した時に、コネクション 1 のスループットが 1.5 [Mbps]、コネクション 2 及び 3 のスループットが 0.75 [Mbps]、コネクション 4~7 のスループットが 0.375 [Mbps] になると理想的な状態であるといえる。

図 2 は送信側端末のバッファサイズを $B = 350$ [packets] とした場合の、3 方式におけるバッファ割り当てとスループットの時間的変化を示したものである。図 2(a) と図 2(d) から、EQ 方式においてはルータ 3 を通るコネクション 4~7 は全く同じ条件であるにもかかわらず不公平なスループットになっていることがわかる。これは、文献 [7] で報告されている、TCP の輻輳制御アルゴリズムによる不公平性が原因である。また、図 2(b) と図 2(e) から、ABT 方式におけるバッファ割り当てが不安定で、コネクション 1 のスループットがしばしば低下していることがわかる。これは、ABT 方式が急激に変動する TCP のウィンドウサイズに基づいてバッファ割り当てを行っているために、割り当てバッファサイズの変動が激しくなり、それほど大きなバッファを必要としないコネクション 2~7 に必要以上のバッファサイズが割り当てられ、コネクション 1 に十分なバッファが割り当てられないためである。一方、SABT 方式では、図 2(c) と図 2(f) に示すように、安定したバッファ割り当てが行われ、高い公平性が維持されている。

次に、送信側端末のバッファサイズを 200 [KBytes] に減少させた場合のシミュレーション結果を図 3 に示す。図 3(d) と図 3(e) から、EQ 方式及び ABT 方式ではコネクション間の公平性が維持できないことがわかる。特にコネクション 1 のスループットが大きく減少し、またルータ 3 に接続されているコネクション 4~7 のスループットが大きく異なっている。これは、EQ 方式及び ABT 方式の各コネクションへのバッファの割り当てが原因である。EQ 方式は、図 3(a) に示すように各コネクションごとに必要なバッファサイズの違いを全く考慮せず、コネクション間で等分しているために不公平が生じている。また ABT 方式のバッ

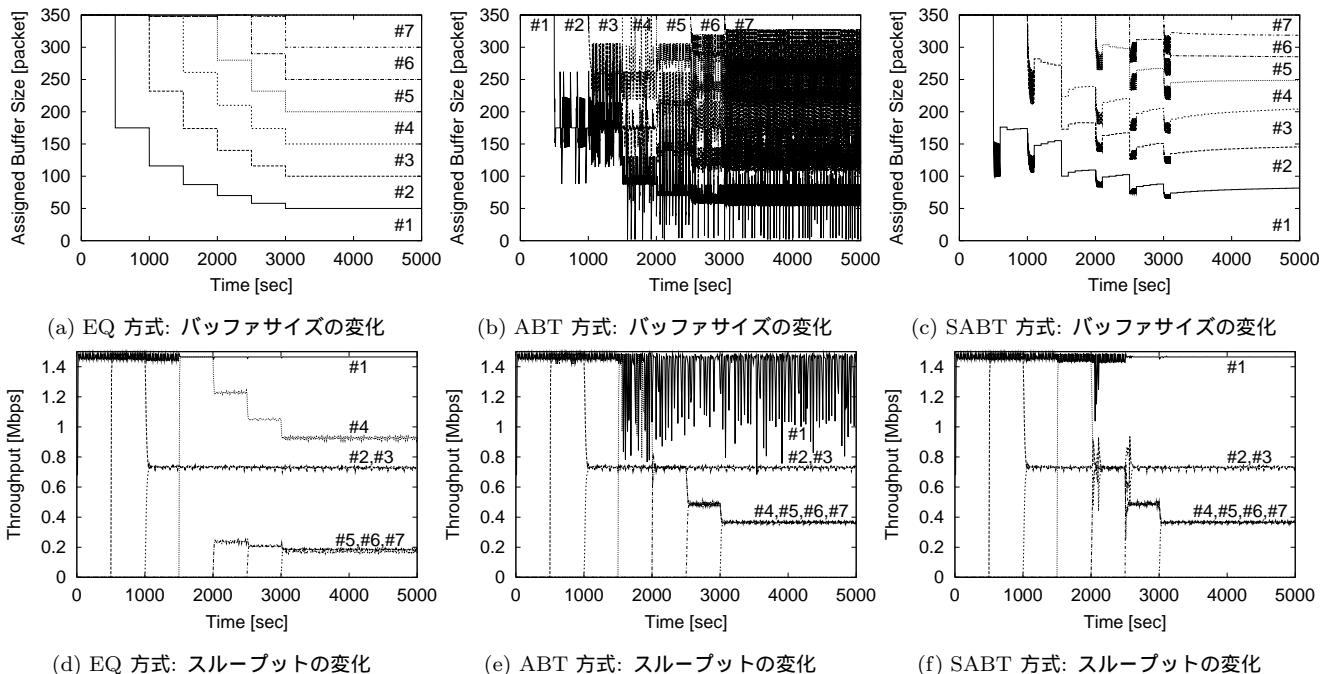


図 2: 3 方式のバッファ割り当ての比較: $B = 350$ [packets] の場合

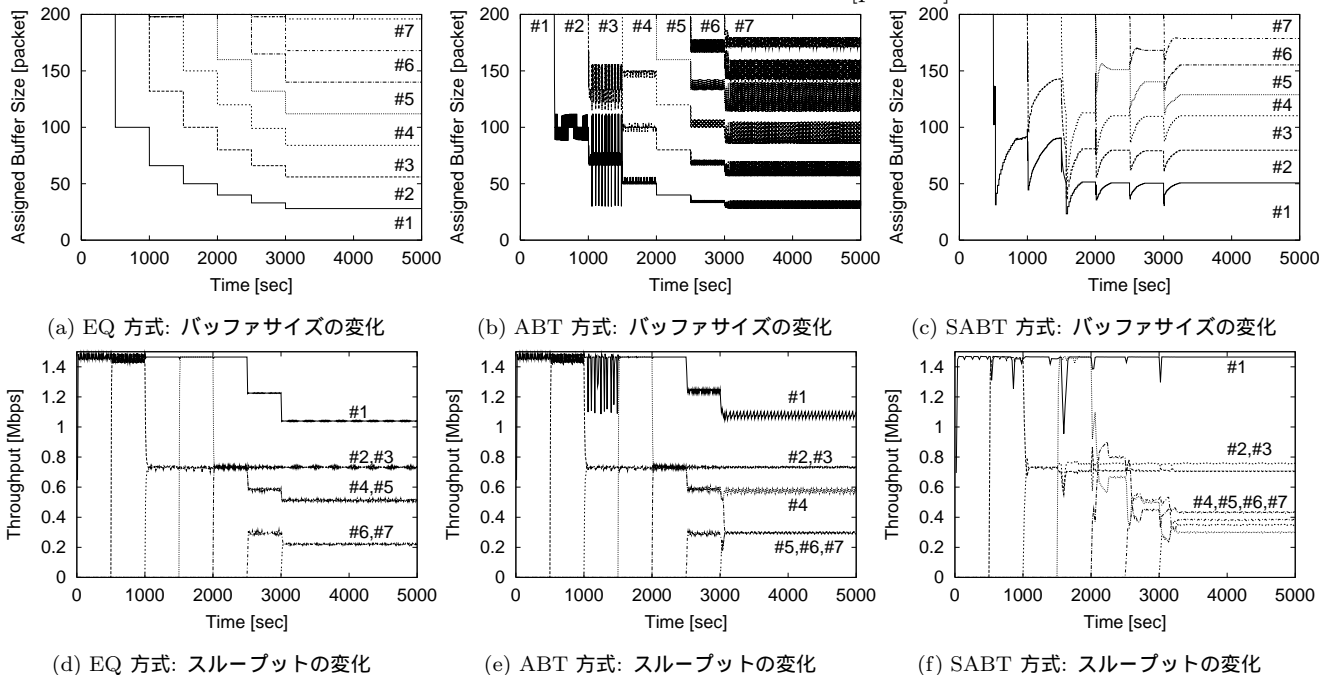


図 3: 3 方式のバッファ割り当ての比較: $B = 200$ [packets] の場合

ファ割り当て (図 3(b)) では図 2(b) の場合と同様に割り当てバッファサイズの変動が非常に大きいため、正しくバッファサイズが割り当てられない場合がある。また、TCP のウィンドウサイズは必ずしもスループットに比例しないため、ウィンドウサイズの変化のみを觀察してバッファサイズを決定する ABT 方式では公平なバッファ割り当てが実現できない。

一方、SABT 方式では、図 3(c) 及び図 3(f) に示すように、接続 1 は高いスループットを達成しており、またその他の接続間の公平性も高い。このことから、SABT 方式が各接続のスループットを適切に推測し、それに基づいて公平にバッファ割り当てを行っていることがわかる。なお、接続 2 と 3 のスループット、及び接続 4~7 のスループットが若干異

なっているのは、各接続の必要バッファサイズの推測に誤差が生じているためであると考えられる。これは Drop-tail ルータにおいては、パケットロスがバースト的に発生するため、SABT 方式のスループット推測に必要なパラメータ (p , r_{tt} , r_{to}) が変動し、SABT 方式のバッファ割り当てに影響しているためである。

4 実装実験による各方式の評価

本章では各バッファ割り当て方式を実コンピュータへ実装し、稼働実験によって各方式の性能評価を行う。実装を行ったマシンの CPU は Pentium-III 700 [MHz]、メモリは 512 [MByte]、また OS は FreeBSD 2.2.8 [8] である。

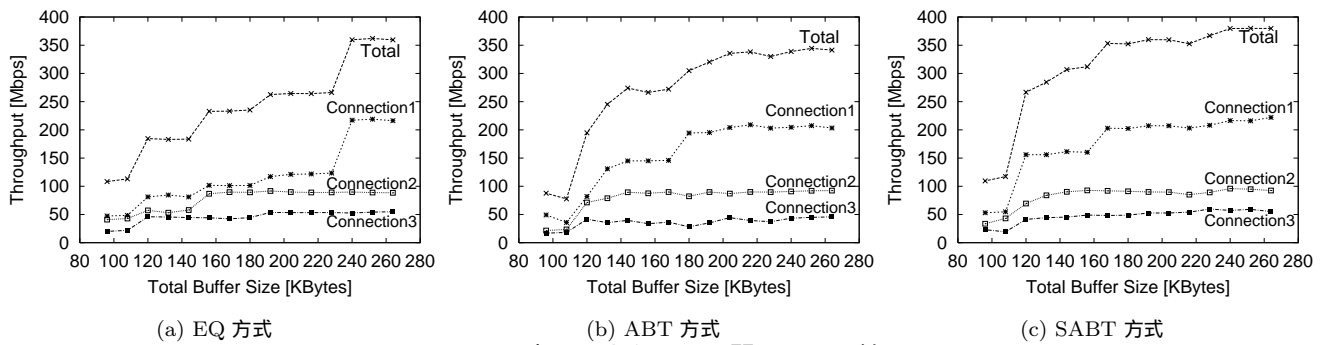


図 5: 3 本のコネクション間での公平性

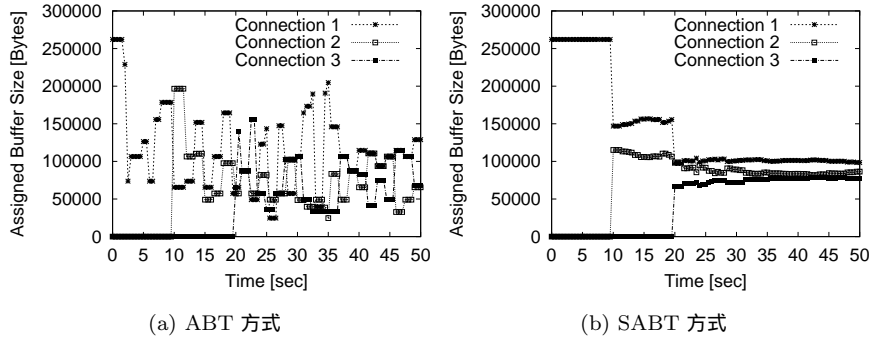


図 6: 割り当てバッファサイズの時間的变化

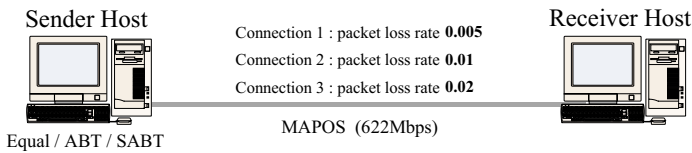


図 4: 4.1 節におけるネットワーク環境

4.1 バッファ割り当ての公平性

まず、送信側端末に複数の異なるコネクションが存在する場合におけるバッファ割り当ての公平性について評価を行う。本実験では図 4 に示す通り、622 [Mbps] の帯域を持つ MAPOS (Multiple Access Protocol Over SONET/SDH) [9] リンクを用いて 2 台のマシンを接続し、送受信端末間に 3 本の TCP コネクションを張ってデータ転送を行う。ここで、3 本の TCP コネクションでは、パケットロスを受信側端末で意図的に発生させている。コネクション 1~3 のパケットロス率はそれぞれ 0.005、0.01、0.02 である。各コネクションが達成できる最大スループットはそれぞれ約 55、90、220 [Mbps] となり、必要となるバッファサイズも異なる。

図 5 は各方式 (EQ、ABT、SABT 方式) における送信側端末のソケットバッファサイズに対する 3 本のコネクションのスループットと合計のスループットの関係を表している。EQ 方式 (図 5(a)) では送信側端末のソケットバッファサイズが 240 [KBytes] になると 3 本の TCP コネクションが最大スループットに達することがわかる。一方、ABT 方式 (図 5(b)) では 200 [KBytes]、SABT 方式 (図 5(c)) では 170 [KBytes] で最大スループットに達する。これは、EQ 方式においてはコネクション 2 と 3 では大きなバッファが必要でないにもかかわらず、3 本のコネクションに等しくバッファを割り当てているためである。

次に、送信側端末のバッファサイズが 120 [KBytes] の場合に注目すると、SABT 方式におけるコネクション 1 のス

ループット及びトータルのスループットが、ABT 方式に比べて高くなっていることがわかる。これは ABT 方式と SABT 方式における余剰バッファの再割り当て方式の違いによるものである。この場合、コネクション 3 はそれほどバッファを必要としないため余剰バッファがコネクション 1 と 2 に再割り当てされる。ABT 方式はコネクション 1 と 2 に等分に余剰バッファを割り当てるのに対し、SABT 方式では各コネクションの要求バッファサイズに比例して割り当てる。このため SABT 方式においてはコネクション 2 に再割り当てされるバッファが ABT 方式よりも小さいため、スループットは ABT 方式と比較して若干低い、より多くのバッファを必要とするコネクション 1 により大きなバッファを割り当てるため、コネクション 1 は大きなスループットを得ることができる。従って 3 本のコネクションの合計のスループットは SABT 方式がもっとも高い。

次に、ABT 方式と SABT 方式における各コネクションへのバッファ割り当ての時間的变化を図 6 に示す。ここでは 3 本の TCP コネクションはそれぞれ 0 [sec]、10 [sec]、20 [sec] でデータ転送を開始している。また、送信側端末のソケットバッファサイズは 256 [Kbyte] である。図 6(a) から、ABT 方式では各コネクションのバッファサイズが大きく振動していることがわかる。これは、ABT 方式は各 TCP コネクションのウィンドウサイズに基づいてバッファサイズの割り当てを行っているためである。一方、図 6(b) に示すように SABT 方式は安定したバッファ割り当てが行われている。この安定したバッファ割り当てによって SABT 方式は図 5(c) に示すような高くかつ公平なスループットを達成している。

4.2 コネクション数に対するスケラビリティ

次に、送信側端末が同時に処理するコネクション数に対するスケラビリティに着目した各方式の評価を行う。本節における実験では、図 7 に示すようなネットワーク環境で実験を用い、1 本の TCP コネクションが MAPOS リン

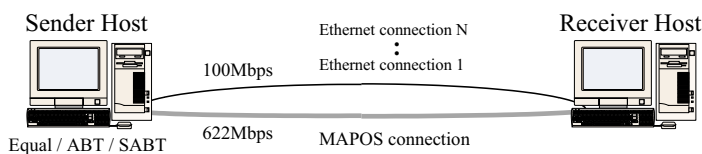


図 7: 4.2 節でのネットワーク環境

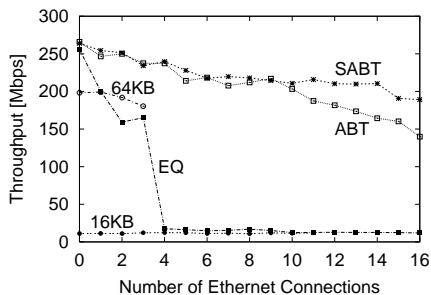


図 8: Ethernet コネクションの本数と MAPOS コネクションのスループットの関係

ク上で転送を行い (以下、MAPOS コネクションと呼ぶ)、同時に Ethernet リンク上で複数の TCP コネクション (以下、Ethernet コネクションと呼ぶ) が転送を行う。

図 8 は Ethernet コネクションの本数を変化させた場合の MAPOS コネクションのスループットの変化を示している。ここでは、EQ 方式、ABT 方式、SABT 方式に加え、各コネクションに割り当てるバッファサイズを 16 [KBytes] または 64 [KBytes] で固定した場合 (ここでは「従来方式」と呼ぶ) の評価結果を併せて示している。この固定方式は現在の主要な OS の TCP/IP の実装において一般的に用いられている方式である。また、送信側端末のバッファサイズを 256 [KBytes] と設定しているため、従来方式で各コネクションに割り当てるバッファサイズを 64 [KBytes] で固定した場合には同時に転送を行うことのできるコネクション数は 4 に制限される。

図 8 に示すように、従来方式で各コネクションのバッファサイズを 16 [KBytes] に固定した場合、MAPOS コネクションにとっては 16 [KBytes] のバッファサイズは不十分であるため、MAPOS コネクションのスループットは非常に低くなる。各コネクションのバッファサイズを 64 [KBytes] に固定すると、MAPOS コネクションのスループットは高くなるが、コネクション数が 4 本以内に制限されてしまう。このように、従来方式では適切なバッファ割り当てを行うことができない。

一方、EQ 方式ではコネクション数に対する制限はないが、Ethernet コネクションが必要とするバッファサイズは MAPOS コネクションに比較して小さいにもかかわらず、すべてのコネクションに等しくバッファを割り当てるため、Ethernet コネクションの本数が増加するにつれて MAPOS コネクションに割り当てられるバッファサイズが減少し、それに伴い MAPOS コネクションのスループットも大きく低下する。

それに対して ABT 方式及び SABT 方式では、Ethernet コネクションの本数が増加しても、MAPOS コネクションのスループットは大きく低下しない。ただし、Ethernet コネクションの本数が増加するにつれて若干スループットが減少しているのは、コネクション数が増加するにつれて送信側端末のプロトコル処理の負荷が大きくなるためである。

しかし、コネクション数が 11 以上になると ABT 方式でのスループットは SABT 方式に比較して大きく低下している。これは余剰バッファの再割り当て方式の違いから生じている。SABT 方式では余剰バッファを各コネクションの要求バッファサイズに比例して公平に再割り当てを行うため、コネクション間で等分して再割り当てを行う ABT 方式に比較して MAPOS コネクションに割り当てることのできるバッファサイズが大きくなり、MAPOS コネクションのスループット劣化を防いでいる。

5 結論

本稿では、同時に処理する TCP コネクション数の多い繁忙インターネットサーバを対象とし、各コネクションに対して効果的なバッファ割り当てを行う Scalable Automatic Buffer Tuning (SABT) 方式の提案を行った。本方式においては、まず送信側端末で容易に観測することのできる 3 つのパラメータを用いて各 TCP コネクションのスループットを推測し、その値と各コネクション間の Max-Min Fairness を考慮してバッファサイズの割り当てを行う。更に、Web ドキュメントサイズの分布を考慮し、サイズの小さいファイルを転送する TCP コネクションのバッファサイズ及びウィンドウサイズの初期値に関する提案を行った。また、提案した SABT 方式の有効性を、シミュレーション及び実コンピュータ上に実装しての稼働実験によって検証した。その結果、Automatic Buffer Tuning (ABT) 方式やその他の方式と比較して、SABT 方式は各 TCP コネクションに対して公平かつ効率的なバッファ割り当てが実現できることを明らかにした。

参考文献

- [1] P. Druschel and L. L. Peterson, "Fbufs: a high-bandwidth cross-domain transfer facility," in *Proceedings of the Fourteenth ACM symposium on Operating Systems Principles*, pp. 189–202, December 1993.
- [2] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP buffer tuning," in *Proceedings of ACM SIGCOMM'98*, pp. 315–323, August 1998.
- [3] T. Matsuo, G. Hasegawa, M. Murata, and H. Miyahara, "Comparisons of packet scheduling algorithms for fair service among connections," in *Proceedings of Internet Workshop '99*, pp. 193–200, February 1999.
- [4] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.
- [5] M. Nabe, M. Murata, and H. Miyahara, "Analysis and modeling of world wide web traffic for capacity dimensioning of internet access lines," *Performance Evaluation*, vol. 34, pp. 249–271, December 1999.
- [6] Network Simulator - ns (version 2), available from <http://www-mash.cs.berkeley.edu/ns/>.
- [7] S. Floyd and V. Jacobson, "On traffic phase effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, pp. 397–413, August 1992.
- [8] FreeBSD Home Page, available from <http://www.freebsd.org/>.
- [9] K. Murakami and M. Maruyama, "MAPOS - multiple access protocol over SONET/SDH, version 1," *Request for Comments 2171*, June 1997.