社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Webプロキシサーバにおける TCP 受信バッファの動的割り当て方式

岡本　卓也† 　長谷川　剛†† 　村田　正幸††

† 大阪大学 大学院基礎工学研究科　〒 560-8531 大阪府豊中市待兼山町 1-3
†† 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
E-mail: †tak-okmt@ics.es.osaka-u.ac.jp, ††{hasegawa,murata}@cmc.osaka-u.ac.jp

あらまし　現在のインターネットにおいては、Web プロキシサーバを経由した Web サーバへのアクセスが数多く存在する。Web プロキシサーバはその性質上、Web クライアントホストからの TCP コネクションおよび Web サーバへの TCP コネクションを収容しなければならないため、Web プロキシサーバにおいて効率的な資源管理が必要になる。そこで本稿では、Web プロキシサーバの高速・高機能化を実現するための資源管理方式を提案する。提案方式は、Web プロキシサーバの特性を考慮し、Web プロキシサーバが Web サーバからドキュメントをダウンロードする際に、各 TCP コネクションのネットワーク状況に応じた大きさの TCP 受信バッファを動的に割り当てる。したがって、ネットワーク帯域が大きい TCP コネクションにはより多くの受信ソケットバッファを割り当て、またネットワーク帯域が小さい TCP コネクションにはより少ない受信ソケットバッファを割り当てることが可能となるため、Web プロキシサーバの資源を効率的に利用することができる。本稿では、提案方式の有効性をシミュレーションを用いて検証した。その結果提案方式を用いることによって、Web プロキシサーバの特性を考慮した効率的な資源管理を行うことができ、資源管理をしない従来方式と比較して最大 2 倍のスループットが得られ、応答時間も約 1/4 に短縮できることを示す。
キーワード　Web プロキシサーバ、TCP、受信ソケットバッファ

Dynamic Receive Socket Buffer Allocation at Web Proxy Servers

Takukya OKAMOTO†, Go HASEGAWA††, and Masayuki MURATA††

† Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
†† Cybermedia Center, Osaka University
1-32 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
E-mail: †tak-okmt@ics.es.osaka-u.ac.jp, ††{hasegawa,murata}@cmc.osaka-u.ac.jp

Abstract　In the current Internet, many HTTP sessions are established between Web clients and servers via Web proxy servers. Since the Web proxy server accommodates lots of TCP connections from Web client hosts, and those to Web servers, it cannot show sufficient performance if the resources of the proxy server should be utilized effectively. In this paper, we propose a new resource management scheme for Web proxy servers to improve their performance and to reduce Web document transfer time via it. Our proposed scheme assigns the proper size of receive socket buffer to each TCP connection, which downloads the original Web document from the Web server via the Web proxy server. That is, a larger size of the receive socket buffer is assigned for a TCP connection with larger link bandwidth, and vice versa. We validate an effectiveness of our proposed scheme through simulation experiments. The simulation results show that our proposed scheme can manage resouces at the Web proxy server and an about twice throughput of the Web Proxy server and up to one-quater response time of document transfer is obtained our proposed scheme as compared with the conventional method.
Key words　Web proxy server, TCP, receive socket buffer

1. はじめに

インターネットの急速な発展により、WWW (World Wide Web) を利用して非常に多くの情報を取得することが可能になっている。WWW の普及によりインターネットの利用者は爆発的に増加し、それに伴なうネットワークトラヒックの増大のためにネットワークにおける輻輳が引き起こされている。この問題に対してこれまで行なわれてきた研究の多くは、ネットワーク帯域の増強や、輻輳の抑制に関するものであった。これは、これまでネットワークの処理速度と比較してエンドホストにおけるデータ転送処理速度は高速であり、エンドホストがボトルネックとなるような状況は想定されていなかったためである。しかし、これまでの研究成果や技術革新により、ネットワーク帯域が飛躍的に増加したため、Web サーバのようなエンドホストがデータ転送のボトルネックとなる状況が生まれつつある。

この問題に対し我々の研究グループでは、エンドホストにおけるデータ転送処理の高速・高機能化のための手法として SSBT (Scalable Socket Buffer Tuning) 方式を提案している [1]。SSBT 方式は、TCP によるデータ転送効率と複数コネクション間の公平性の向上を目的とした、動的な送信ソケットバッファ割り当て手法である E-ATBT (Equation-based Automatic TCP Buffer Tuning) 方式、および高速なデータ転送時の通信処理軽減手法である SMR (Simple Memory-copy Reduction) 方式からなる。[1] では、シミュレーションおよび実験ネットワー

クにおいて、実ネットワーク上での Web アクセス分布を考慮したワークロードを用いたデータ転送実験を行ない、SSBT 方式の有効性を確かめた。しかし現在のインターネットでは、Web プロキシサーバを経由した HTTP アクセスも多い [2]。Web プロキシサーバは、ISP (Internet Service Provider) 等が顧客へのサービスために提供している場合が多いため、多くのユーザからのドキュメント転送要求を同時に受け付け処理する必要がある。そのため、ネットワークに輻輳が発生しておらず、また Web サーバの稼働能力が十分であるにもかかわらず、Web プロキシサーバの処理能力が十分でないために、Web プロキシサーバがドキュメント転送のボトルネックとなることが考えられる。すなわち、Web プロキシサーバにおいても、高速・高機能化のための手法を考慮する必要がある。

Web サーバと同様に SSBT 方式を適用することによって、Web プロキシサーバの高速・高機能化を実現できると考えられる。しかし上述したように、Web プロキシサーバは多くの TCP コネクションを同時に受け付けなければならず、また Web サーバと異なり、ドキュメントをダウンロードする際には受信ホストとしてふるまう。そのため、SSBT 方式を単純に適用するだけでは不十分であり、Web プロキシサーバの特性を考慮した資源管理方式を適用する必要がある。そこで我々は [3] において、Web プロキシサーバの特性を考慮した資源管理手法として、送受信ソケットバッファの動的割り当て手法、およびドキュメント転送終了後の persistent な TCP コネクションを管理するコネクション管理手法を提案した。送受信ソケットバッファの動的割り当て手法は、[1] で提案した E-ATBT 方式を Web プロキシサーバへ適用したものであり、各 TCP コネクションのネットワーク状況に応じて受信ソケットバッファを与える手法である。さらに Web プロキシサーバの特徴である、クライアントホストへの TCP コネクションと Web サーバへの TCP コネクションの依存関係、すなわち両コネクション間のスループットの違いを考慮したソケットバッファ管理手法をあわせて提案した。しかし受信ソケットバッファの割り当て手法に関しては、その必要性および設計指針を述べるにとどまっており、詳細なアルゴリズムを示していない。そこで本稿では、受信ソケットバッファの動的な割り当て手法について詳細なアルゴリズムを示す。提案アルゴリズムは、受信ホストにおいて受信ソケットバッファの利用率を監視し、受信ソケットバッファが不足すればより多くのバッファを割り当て、逆に受信ソケットバッファを余剰に割り当てている場合には、バッファの割り当てサイズを減少させる手法である。また本稿では、提案アルゴリズムの有効性をシミュレーションを用いた性能評価結果を通じて確認する。

以下、2 章では Web プロキシサーバの概略と TCP の送受信ソケットバッファについて述べる。3 章では TCP 受信バッファの動的割り当て手法を説明し、詳細なアルゴリズムを示す。4 章では提案方式の有効性をシミュレーションを用いて検証する。最後に、5 章で結論と今後の課題について述べる。

2. 背　　景

2.1 Web プロキシサーバ

Web プロキシサーバとは、Web クライアントホストからオリジナル Web サーバへのドキュメント転送要求を代理で行なうサーバである (図 1)。Web プロキシサーバはクライアントホストからドキュメント転送要求を受信すると、要求されたドキュメントをクライアントに代わってオリジナル Web サーバからダウンロードする。その際、要求されたドキュメントをキャッシュに保存しておき、別のクライアントホストから同じドキュメントの転送要求が発生した際には、キャッシュしたドキュメントを直接クライアントホストに転送する。これにより、クライアントの感じるドキュメント転送時間を短かくすることが可能となる。[4] では、Web プロキシサーバを利用することによって、ドキュメント転送時間を約 30%削減することが可能となると報告されている。また、キャッシュしたドキュメントをクライアントホストへ直接転送することにより、オリジナル Web サーバへの TCP コネクションの確立を行なう必要がないため、ネットワーク内のトラヒック量を抑え、さらにオリジナル Web サーバの負荷を軽減することができる。

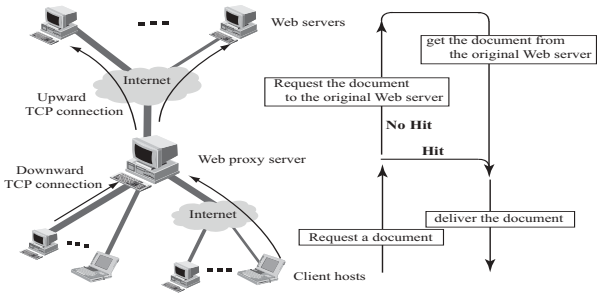


図 1　Web プロキシサーバの概略

Web プロキシサーバは、多数の TCP コネクションを同時に処理しなければならず、またその性質上送信ホストにも受信ホストにもなる。そのため、Web プロキシサーバにおいて十分な資源管理がなされていなければ、Web プロキシサーバの資源を浪費してしまうため、ネットワーク帯域およびオリジナル Web サーバの処理能力に余裕があるにもかかわらず、Web プロキシサーバがドキュメント転送のボトルネックとなり、満足するサービスを提供できなくなる。したがって、Web プロキシサーバの性能を向上させるためには、Web プロキシサーバにおいて効率的な資源管理が必要である。

Web プロキシサーバが持つ資源には、mbuf やファイルディスクリプタ、TCP コネクション情報を格納するメモリ領域、ソケットバッファ等が挙げられる [3]。本稿ではこれらの資源のうち、TCP によるデータ送受信の際に各 TCP コネクションに割り当てられる資源であるソケットバッファに着目する。

2.2 ソケットバッファ

TCP を用いたデータ転送では、各 TCP コネクション毎に転送データを格納するための領域である、ソケットバッファが送受信ホストにおいて割り当てられる。ユーザアプリケーションが TCP を用いてデータ転送を行なう時には、送信データは一旦送信ソケットバッファにコピーされ、続いてカーネルのメモリ空間にある mbuf にコピーされネットワークに送出される。受信ホストにおいては、受信したデータパケットは受信ソケットバッファに蓄積された後、ユーザアプリケーションに渡される。TCP によるデータ転送においては、各 TCP コネクション毎に割り当てられるソケットバッファサイズがその性能に大きな影響を与える。例えば、64 [Kbps] のリンクで接続しているクライアント A と、100 [Mbps] のリンクで接続しているクライアント B が存在する場合に、OS が同じ大きさの送信ソケットバッファを両コネクションに割り当てると、クライアント A にとってはバッファサイズが大きい、あるいはクライアント 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。このように、エンドホスト資源の一つであるソケットバッファを有効に利用するためには、それぞれの TCP コネクションの持つリンク容量 (帯域遅延積) に応じたソケットバッファの割り当てを行なう必要がある。

我々の研究グループでは、Web サーバにおいて RTT (Round Trip Time)、RTO (Retransmission Timeout)、パケットロス率などから各 TCP コネクションのスループットを解析的に推測し、そのスループットに応じて動的に送信ソケットバッファを割り当てる、E-ATBT 方式を提案している [1]。E-ATBT 方式では、送信ホストが上述のパラメータを利用し、[5] に示されている解析手法により平均ウィンドウサイズを導出する。その

ウィンドウサイズから、さらに TCP の再送タイムアウトを考慮した上で、各 TCP コネクションの平均スループットを導出する。このようにして求めた平均スループットを基に、各 TCP コネクションが必要としている送信ソケットバッファの大きさを導出する。我々はこの E-ATBT 方式を Web サーバへ適用し、その有効性を確かめた。

Web プロキシサーバは、Web サーバと同様に様々な環境から非常に多くの TCP コネクションを収容している。そのため、E-ATBT 方式を適用することによって Web プロキシサーバの高速化を実現することが可能であると考えられる。しかし、Web プロキシサーバは Web サーバと異なり、ドキュメントダウンロード時のオリジナル Web サーバへの TCP コネクションにおいては受信ホストとしてふるまうため、E-ATBT 方式で考慮していなかった受信ソケットバッファの制御が必要となる。そこで次章において、TCP 受信バッファの動的割り当てアルゴリズムを提案する。

3. TCP 受信バッファの動的割り当て

これまで述べたように、Web プロキシサーバの高速・高機能化のためには、Web プロキシサーバのふるまいを考慮した動的なソケットバッファの割り当てを行なう必要がある。一方、TCP の輻輳制御方式、およびスループットに関するこれまでの研究のほとんどにおいては、エンドホストではなくネットワークにボトルネックがあり、TCP コネクションの受信ソケットバッファの大きさは十分に大きいと仮定されてきた。そのため、従来の OS によって割り当てられる受信ソケットバッファの大きさは、例えば FreeBSD では 16KB と、TCP コネクションの状態に関係なく固定的に割り当てられる [6]。しかし、近年のインターネットの発展によってリンク帯域が飛躍的に増加している状況においては、従来の割り当て方法による受信ソケットバッファの大きさは十分ではなく、データ転送のボトルネックになることがある [7]。この問題を避けるためには、受信ホストにある受信ソケットバッファの大きさを、送信ホストの輻輳ウィンドウサイズ以上にする必要がある。したがって、受信ホストにおいて送信ホストの輻輳ウィンドウサイズを知る必要がある。

受信ホストにおいて送信ホストの輻輳ウィンドウサイズを知る方法として、我々の研究グループでは [3] において、受信ソケットバッファの利用率を監視する方法、および送受信するパケットのヘッダにコネクション情報を埋めこむ方法の 2 種類の指針を示した。後者の方法は、前者の方法に比べてより正確に送信ホストの輻輳ウィンドウサイズを知ることが可能であるが、送受信ホストの TCP を改造する必要があるという大きな問題点がある。そこで本章では、前者の指針に基づいて、受信ソケットバッファの利用率の変化の特性を利用した受信ソケットバッファの割り当てアルゴリズムを示す。

3.1 割り当てアルゴリズム

TCP を用いたデータ転送においては、受信ホストが受信したデータパケットは、ビットエラーのチェック等を行なった後、受信ソケットバッファに格納される。パケットが廃棄されることなく順調に受信されている場合は、受信ソケットバッファに格納されたパケットは通常直ちにユーザアプリケーションに渡される。そのため、受信ソケットバッファの利用率 (パケットが格納されている割合) は低い。しかし、ネットワーク内でパケットが廃棄されると、それ以降に受信されたパケットは、その廃棄されたパケットが再送によって受信ホストに届くまで受信ソケットバッファに蓄積される。したがって、受信ソケットバッファの利用率は高くなる。本提案方式は、この受信ソケットバッファの利用率の変化の特性を利用し、受信ソケットバッファの割り当てを行う。以降、パケット廃棄が発生する場合としない場合について、提案アルゴリズムに関する説明を行なう。

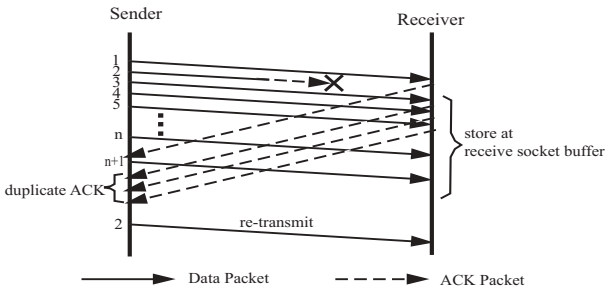


図 2　受信ソケットバッファの動作

3.1.1 パケット廃棄がある場合

ネットワーク内でパケット廃棄等が発生すると、それ以降に受信したパケットは、廃棄されたパケットが再送され受信ホストに到着するまで受信ソケットバッファに蓄積される。そのため、受信ソケットバッファの利用率は一時的に高くなる。ネットワーク内で廃棄されたパケットが再送され受信ホストに到着すると、それまで蓄積されていたパケットはまとめてユーザアプリケーションに渡され、受信ソケットバッファの利用率は低くなる。このとき、廃棄されたパケットが Fast Retransmit アルゴリズム [8] によって再送されるとすると、廃棄されたパケットの再送は約 1 RTT (Round Trip Time) 後に行なわれる。そのため、受信ソケットバッファに蓄積されるパケット数は送信ホストが 1 RTT に送信することができるパケット数、すなわち送信ホストの輻輳ウィンドウサイズにほぼ等しくなる (図 2)。そこで、受信ホストで観測して得られた受信ソケットバッファの利用率を用いて、受信ソケットバッファの割り当てを以下のように行なう。なお、受信ホストにおけるネットワーク内でのパケット廃棄の検出に関しては、受信したパケットの TCP ヘッダに記されているシーケンス番号の検査を行ない、シーケンス番号が連続していなければパケットが廃棄されたと判断する。

- パケット廃棄があり、受信ソケットバッファの利用率が高い場合
  送信ホストから 1 RTT で送出されたパケットによって受信ソケットバッファがほとんど使い切られてしまっているため、受信ソケットバッファの大きさがデータ転送のボトルネックとなっている可能性がある。そこで、受信ソケットバッファを、観測された送信ホストのウィンドウサイズを考慮した大きさに変更する。
- パケット廃棄はあるが、受信ソケットバッファの利用率が低い場合
  現在のネットワーク状況では、受信ソケットバッファが必要以上に多く割り当てられていると考えられるので、割り当てサイズを小さくする。

3.1.2 パケット廃棄がない場合

ネットワークにおいてパケット廃棄が発生せず、ユーザアプリケーションの処理速度 (受信ソケットバッファにあるデータパケットを処理する速度) が十分に速いときは、受信ソケットバッファの利用率は低い。この場合、以下の 2 つの状況が考えられる。

(a) 受信ソケットバッファは十分に大きく、送信側のウィンドウサイズがデータ転送のボトルネックになっている
(b) 受信ソケットバッファが小さすぎて、データ転送のボトルネックとなっている

これらの違いは、受信ソケットバッファの割り当てサイズが TCP コネクションの利用可能な帯域遅延積より大きいかどうかによって決まる。しかし受信ホストではデータパケットを送信しないため、ネットワーク状況を知ることができな。そこで本方式では、受信ソケットバッファの大きさを変化させ、スループット (受信ホストにおけるデータパケット受信レート) の変化

を観測することによって以下のように判断する。

- 受信ソケットバッファを増加させた時に、スループットが向上しない場合
  既に受信ソケットバッファが十分に割り当たっていると考えられるため、(a) の場合に相当する。したがって、受信ソケットバッファを減少させる。
- 受信ソケットバッファを増加させた時に、スループットが向上する場合
  受信ソケットバッファの増加によりスループットが向上したと考えられるため、(b) の場合に相当する。したがって、さらに受信ソケットバッファを増加させる。

また、パケット廃棄が発生しない場合においても、ユーザアプリケーションの処理速度が低いために、受信ソケットバッファの利用率が高くなることがある。これは、ユーザアプリケーションの処理速度が遅いために、受信ソケットバッファに蓄積されたパケットがなかなか処理されないためである。この場合は、ユーザアプリケーションの処理速度がデータ転送のボトルネックであるため、受信ソケットバッファを増加させてもスループットは向上しない。したがって、受信ソケットバッファを増減させず、そのままの大きさを維持する。

3.1.3 アルゴリズム

前節までの考察に基づいた、具体的な受信ソケットバッファの動的割り当てアルゴリズムを以下に示す。ある TCP コネクションの i 番目の観測期間における受信ソケットバッファの最大利用率を U_i、観測期間中に割り当てられている受信ソケットバッファサイズを B_i、観測期間中のスループット (データパケットの受信速度) を ρ_i とすると、新たな割り当てサイズ B_{i+1} は以下のようになる。

- 観測期間 i にパケット廃棄が発生した場合

$$\begin{cases} B_{i+1} = \alpha \times U_i \times B_i & \text{when} \quad U_i < T_u \\ B_{i+1} = \beta \times B_i & \text{when} \quad U_i < T_l \\ B_{i+1} = B_i & \text{otherwise} \end{cases}$$

- 観測期間 i においてパケット廃棄が発生しなかった場合

$$\begin{cases} B_{i+1} = \alpha \times B_i & \text{when} \quad U_i < T_l \ \text{ and } \ \rho_i \geq \rho_{i-1} \\ B_{i+1} = \beta \times B_i & \text{when} \quad U_i < T_l \ \text{ and } \ \rho_i < \rho_{i-1} \\ B_{i+1} = B_i & \text{otherwise} \end{cases}$$

ここで、α、β、は受信ソケットバッファの割り当てサイズの増減割合を決定するパラメータであり、T_u、T_l は受信ソケットバッファの利用率が高いか低いかを判断するための閾値である。

4. シミュレーション

本章では、前章において提案した受信ソケットバッファの動的割り当て方式について、ns-2 [9] を用いたシミュレーションによってその有効性を確かめる。なお以下のシミュレーションでは、3.1.3 節で示した変数 α、β、T_u、T_l は、それぞれ、2、0.1、0.9、0.4 としている。

4.1 ファイル転送実験による性能評価

まず、提案した TCP 受信バッファ割り当てアルゴリズムの基本的性質を確認するために、無限長ファイルの FTP 転送による性能評価を行なう。シミュレーションを行なったネットワークトポロジを図 3 に示す。図 3 において、受信ホストとルータの間のリンク帯域を 100 Mbps、伝播遅延時間を 0.001 sec とする。また、送信ホスト数を 45 とし、帯域と伝搬遅延時間が (100 Mbps, 1 sec)、(100 Mbps, 0.1 sec)、(100 Mbps, 0.01 sec)、(1.5 Mbps, 1 sec)、(1.5 Mbps, 0.1 sec)、(1.5 Mbps, 0.01 sec)、(128 Kbps, 1 sec)、(128 Kbps, 0.1 sec)、(128 Kbps, 0.01 sec) であるような 9 種類のリンクをそれぞれ 5 本ずつ用意し、45 台の送信ホストとルータの間のリンクとする。ボトルネックルータのバッファサイズは 50 とする。このネットワークにおいて、各送信ホストは受信ホストに向かって無限長のファイル転送を FTP によって行なう。すわなち、受信ホストは 45 本の TCP コネクションをシミュレーション中保持し、データを受信する。シミュレーション時間は 1000 秒とした。

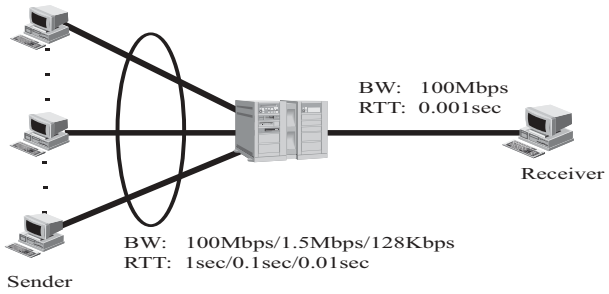


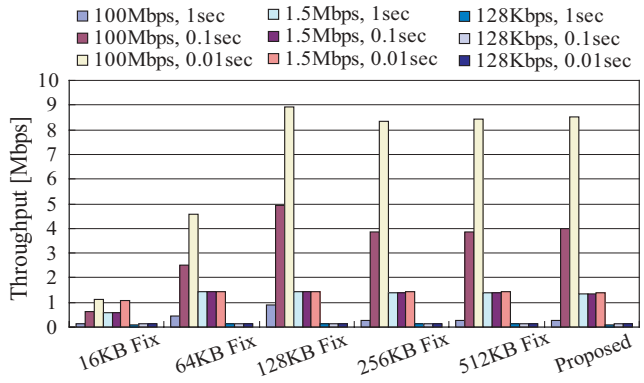
図 3 シミュレーショントポロジ (1)

図 4 に、受信ホストにおける TCP コネクション 1 本あたりの受信ソケットバッファの大きさとして、16 KB、64 KB、128 KB、256 KB、512 KB の固定値、および提案方式を用いた場合のシミュレーション結果を示す。図 4(a) は、リンク帯域と伝播遅延時間がそれぞれ異なる 9 種類のコネクションの平均スループットを示し、図 4(b) は、受信ホストが 9 種類の TCP コネクションに割り当てている受信ソケットバッファの平均サイズを示している。また、各グラフの上部には各条件下においてすべての TCP コネクションを収容するために必要であった受信ソケットバッファの総量を示している。

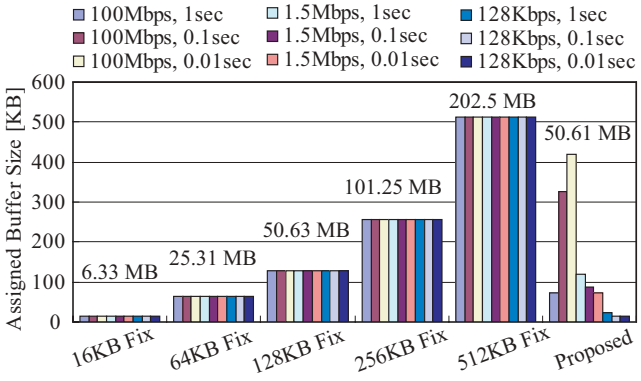
この結果から、固定的に 16 KB や 64 KB の受信ソケットバッファが割り当てられる場合においては、リンク帯域が 100 Mbps や 1.5 Mbps のリンク上の TCP コネクションに対して受信ソケットバッファが小さすぎるため、スループットが低下していることが分かる。また逆に、各 TCP コネクションに 512 KB の受信ソケットバッファを割り当てた場合は、十分なスループットが得られているが、その値は 128 KB や 256 KB の場合とほとんど変わらず、過剰な受信ソケットバッファが割り当てられている。

一方、提案方式を用いた場合には、各コネクションは十分大きなソケットバッファを割り当てた場合 (512 KB) とほぼ同一のスループットを獲得しながら、使用している受信ソケットバッファの総量を小さく抑えることができている。また、図 4(b) より、リンク帯域が 128 Kbps のリンク上の TCP コネクションには、その帯域での十分な大きさである 15〜20 KB の受信ソケットバッファを割り当て、リンク帯域が大きい 100 Mbps のコネクションにはより大きな 300〜400 KB の受信ソケットバッファを割り当てていることが分かる。すなわち提案方式は、各 TCP コネクションが必要とするバッファサイズに応じた受信ソケットバッファの割り当てを行ない、少ない受信ソケットバッファの割り当てサイズで多くの TCP コネクションを収容し、その結果高いスループットを得ることができていることを示している。このことは、受信ソケットバッファの総量を固定した次節での Web プロキシサーバにおける性能評価で明らかにする。

また、このシミュレーション結果における提案方式の性能および受信ソケットバッファの割り当て総量は、128 KB の固定サイズを割り当てた場合とほぼ同じ結果になっている。しかし、実際の Web プロキシサーバの性能は収容するコネクション数や受信ソケットバッファの総量などに依存しており、最適な受信ソケットバッファの割り当てサイズは変化する。提案方式では、この変化に対応し動的な割り当てが可能であるため、どのような環境においても高いスループットを獲得することができる。このことも、次節であわせて明らかにする。



(a) スループット



(b) 受信ソケットバッファサイズの使用量

図 4　シミュレーション結果 (FTP 転送: スループットと受信ソケットバッファの割り当てサイズ)

4.2　Web プロキシサーバの性能評価

次に、提案方式を Web プロキシサーバに適用した際の性能評価を行なう。シミュレーションに用いたネットワークトポロジを図 5 に示す。図 5 において、Web プロキシサーバと Web サーバ間のリンク帯域を 100 Mbps/10 Mbps/1.5 Mbps、伝播遅延時間を 1 sec/0.1 sec/0.01 sec とするような Web サーバ数を 99 (各 11 台× 9 種類)、また Web プロキシサーバと Web クライアントホスト間のリンク帯域を 10 Mbps/1.5 Mbps/128 Kbps、伝播遅延時間を 1 sec/0.1 sec/0.01 sec とするような Web クライアントホスト数を 450 (各 50 台× 9 種類) とする。各クライアントホストは、99 台の Web サーバからランダムに 1 台選択し、ドキュメント転送要求を Web プロキシサーバに送信する。Web プロキシサーバはキャッシュヒット率 (このシミュレーションでは 0.5 に固定) に基づいて、キャッシュヒットかキャッシュミスかを判断し、キャッシュヒットであればそのままクライアントホストにドキュメントを転送し、キャッシュミスであれば Web サーバからダウンロードした後、クライアントホストへドキュメント転送を行なう。各クライアントホストは、[10] に示されている、実際の Web クライアントの動作解析によって得られたドキュメント転送要求間隔分布、要求ドキュメントサイズ分布にしたがってドキュメント転送要求を発生するものとする。シミュレーション時間は 1000 秒とする。

図 6 は、Web プロキシサーバの受信ソケットバッファの総量を 3.2 MB/6.4 MB/12.8 MB と変化させ、Web プロキシサーバの受信ソケットバッファの大きさを 16 KB、64 KB、128 KB、256 KB の固定値、および提案方式を用いたときの、Web プロキシサーバがクライアントホストに向けて転送したドキュメントの総バイト数 (図 6(a))、およびクライアントホストがドキュメント転送要求を送信してからドキュメントのダウンロードが完了するまでにかかった応答時間を示している (図 6(b))。図 6(a) から、Web プロキシサーバの受信ソケットバッファの総量が十分にある (12.8 MB) 場合、4.1 節でのシミュレーション結果と同様に、受信ソケットバッファの割り当てサイズが小さい場合には TCP コネクションのスループットが低く、Web プロキシサーバの性能が低下していることがわかる。一方、受信ソケットバッファの割り当てサイズが大きい場合はより高いスループットが得らている。また、図 6(b) から、受信ソケットバッファの割り当てサイズが小さい場合、応答時間が長いことが分かる。これは、TCP コネクションのリンク帯域と比較して受信ソケットバッファの割り当てサイズが小さすぎるため、TCP コネクションのスループットが低く、ドキュメント転送時間が大きくなっているためである。

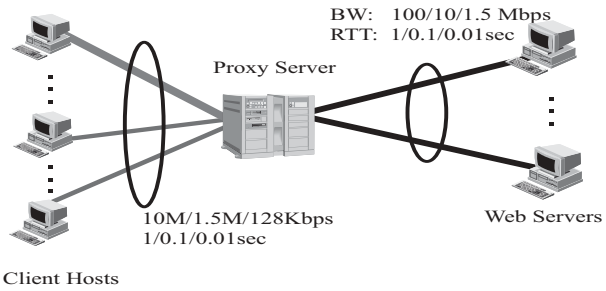
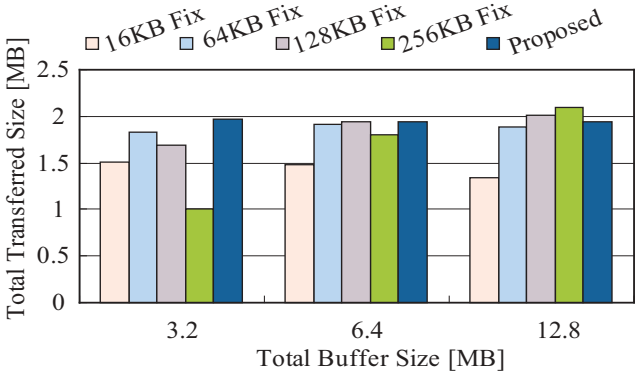
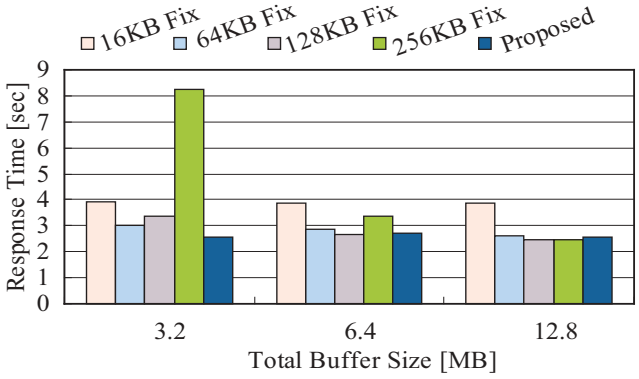


図 5　シミュレーショントポロジ (2)



(a) 転送バイト数



(b) 応答時間

図 6　シミュレーション結果

逆に、Web プロキシサーバの受信ソケットバッファの総量が少ない (3.2 MB) 場合は、受信ソケットバッファの割り当てサイズが大きい時には、応答時間が非常に長く (図 6(b))、Web プロキシサーバの転送量も少なくなっていることが分かる (図 6(a))。これは 256 KB と大きなサイズの受信ソケットバッファを各 TCP コネクションに割り当てると、受信ソケットバッファが不足し、Web プロキシサーバに到着するすべての TCP コネクションを収容することができなくなるためである。その結果、収容されない TCP コネクションは他の TCP コネクションが転送を終了し、受信ソケットバッファを開放するまで待たされるため、ドキュメント転送に必要な時間が増大する。また、256 KB の受信ソケットバッファサイズはほとんどの TCP コネクションとっては大きすぎるため、受信ソケットバッファの使用効率が低下し、Web プロキシサーバの転送量が低下している。

一方、提案方式を適用した場合、受信ソケットバッファの総量が十分ある時は、従来方式において十分大きな受信ソケットバッファを割り当てた場合とほぼ同等のスループットが得られ、また応答時間も短いことが分かる。さらに、受信ソケットバッファの総量が少ない場合においても、高いスループットを維持し、かつ応答時間も短いことが分かる。これは、リンク帯域の小さい TCP コネクション (今の場合、128 Kbps) の使用していない受信ソケットバッファを、リンク帯域の大きい TCP コネクション (10 Mbps) に再割り当てを行なうことによって、リンク帯域の大きい TCP コネクションの受信ソケットバッファの不足を解消することができるためである。またその結果、各 TCP コネクションに割り当てられた受信ソケットバッファの使用効率が向上するため、Web プロキシサーバの転送量が大きくなり、応答時間を短くすることができる。図 7 は、ドキュメント転送のために TCP コネクションを張ろうとした時に、受信ソケットバッファの不足のためにコネクション確立要求が拒否された確率 (ブロッキング率) を示したグラフである。この図から、受信ソケットバッファを固定的に割り当てる従来方式では、特に受信ソケットバッファの総量が小さい時に、TCP コネクション 1 本あたりの割り当てサイズが大きくなると、ブロッキング率が非常に高くなっていることがわかる。これは、Web プロキシサーバにおいて受信ソケットバッファが不足するため、新規の TCP コネクションを確立できないためである。さらに、確立されている TCP コネクションに割り当てられている受信ソケットバッファの実際の使用効率は低い。このため、図 6 に示したように、Web プロキシサーバの性能が劣化し、ドキュメント転送にかかる時間が増大している。一方、提案方式では、コネクション確立要求が拒否されることはなく、新規の TCP コネクションが到着すると、ネットワーク状況に応じて必要とする受信ソケットバッファを割り当てるため、受信ソケットバッファの使用効率が向上する。これにより、受信ソケットバッファの総量に依存せず、高い Web プロキシサーバの性能および短い応答時間を実現している。

また、図 6 より、受信ソケットバッファの総量が 3.2 MB の時は 64 KB、6.4 MB の時は 128 KB、および 12.8 MB のときは 256 KB の固定サイズを割り当てると、Web プロキシサーバのスループットが一番高く、かつ応答時間が短い。このように、従来の固定的な割り当て方式においては、受信ソケットバッファの総量などによって最適な割り当てサイズが異なる。一方、提案方式を用いることで、どのような状況下においても Web プロキシサーバのスループットはほぼ最大の値を得られ、また応答時間も非常に短い。すなわち提案方式はネットワーク状況に応じた効率的な受信ソケットバッファの割り当てが可能であるといえる。

5. まとめと今後の課題

本稿では、Web プロキシサーバの特性を考慮した、TCP 受信ホストにおける受信ソケットバッファの動的管理割り当て方式を提案した。提案方式では、Web プロキシサーバの特性である TCP 受信ホストとしてのふるまいを考慮し、受信ソケットバッファの利用率に応じてその割り当てサイズを変更する、動的な受信ソケットバッファの割り当てを行なう。提案方式の有効性はシミュレーションによって評価した。その結果、提案方式を用いることによって、効率的な資源割り当てが可能となり、受信ソケットバッファの総量が小さい場合においても、高い性能および短い転送時間を達成することができることが明らかとなった。今後の課題としては、本方式を実プロキシサーバへ実装し、実ネットワーク上での性能評価を行うことが挙げられる。

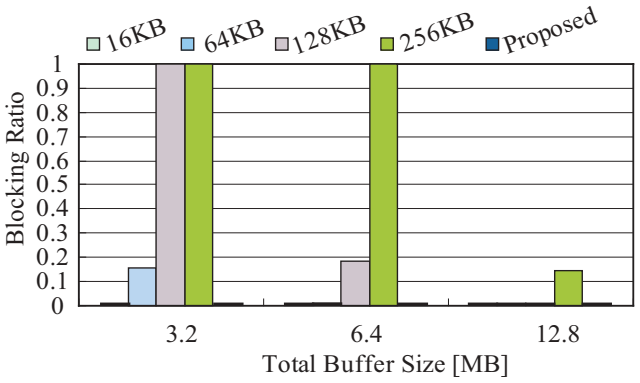


図 7　ブロッキング率

文　献

[1] G. Hasegawa, T. Terai, T. Okamoto, and M. Murata, "Scalable socket buffer tuning for high-performance Web servers," in *Proceedings of IEEE ICNP 2001*, Nov. 2001.

[2] Proxy Survey, available at *http://www.delegate.org/survey/proxy.cgi*.

[3] T. Okamoto, T. Terai, G. Hasegawa, and M. Murata, "A resouorce/connection management scheme for HTTP proxy servers," in *Proceedings of Second International IFIP-TC6 Networking Conference*, pp. 252–263, May 2002.

[4] A. Feldmann, R. Caceres, F. Douglis, G. Glass, and M. Rabinovich, "Performance of Web proxy caching in heterogeneous bandwidth environments," in *Proceedings of IEEE INFOCOM '99*, pp. 107–116, 1999.

[5] G. Hasegawa, T. Matsuo, M. Murata, and H. Miyahara, "Comparisons of packet scheduling algorithms for fair service among connections on the internet," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000.

[6] M. K. McKusick, K. Bostic, M. J. Karels, and J. S. Quarterman, *The Design and Implementation of the 4.4 BSD Operating System*. Reading, Massachusetts: Addison-Wesley, 1999.

[7] M. Allman, "A Web server's view of the transport layer," *ACM Computer Communication Review*, vol. 30, pp. 10–20, Oct. 2000.

[8] W. Stevens, "TCP slow start, congestion avoidance, fastretransmit, and fast recovery algorithms," *Request for Comments (RFC) 2001*, Jan. 1997.

[9] The VINT Project, "UCB/LBNL/VINT network simulator - ns (version 2)." available at `http://www.isi.edu/nsnam/ns/`.

[10] P. Barford and M. Crovella, "Generating representative Web workloads for network and server performance evaluation," in *Proceedings of the 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 151–160, July 1998.