

# TCP オーバレイネットワークの性能解析および評価

牧 一之進† 長谷川 剛‡ 村田 正幸† 村瀬 勉§

† 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  
Phone: 06-6879-4542, Fax: 06-6879-4544  
{i-maki, murata}@ist.osaka-u.ac.jp

‡ 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  
Phone: 06-6850-6864, Fax: 06-6850-6868  
hasegawa@cmc.osaka-u.ac.jp

§ 日本電気株式会社  
システムプラットフォーム研究所  
〒 216-8555 神奈川県川崎市宮前区宮崎 4 丁目 1-1  
Phone: 044-856-2123, Fax: 044-856-2230  
murase@ccm.cl.nec.co.jp

あらまし 近年の多種多様なユーザの要求品質を満たすため、我々はトランスポート層において品質制御を行う TCP オーバレイネットワークに着目している。TCP オーバレイネットワークにおいては、通常エンドホスト間に設定される TCP コネクションをネットワーク内のノード (TCP プロキシ) で終端し、分割されたコネクションごとにパケットを中継しながら転送を行う。過去の研究において、生存時間の長いコネクションに着目した場合、エンドホスト間に設定されるコネクションを分割すると、分割後のコネクション間で干渉が発生するため、期待するほどのスループット向上が見られないことを明らかにした。しかしながら、分割しない場合と比較して得られる平均スループットは大きく向上することが分かった。本稿では、多くの Web コネクションのように転送するドキュメントサイズが小さい TCP コネクションの転送遅延時間に着目した、TCP コネクション分割機構の性能評価を行う。転送データサイズが小さい TCP コネクションを分割する場合、TCP プロキシにおける処理遅延やバッファリング遅延により、分割を行わない通常のコネクションよりも転送遅延時間が大きくなる可能性がある。また本稿では、転送遅延時間の増加を防ぐための方針について議論する。

和文キーワード TCP (Transmission Control Protocol)、 オーバレイネットワーク、 コネクション分割機構、 性能評価

## Performance Analysis and Evaluation of TCP Overlay Networks

Ichinoshin Maki† Go Hasegawa‡ Masayuki Murata† Tutomu Murase§

† Graduate School of Information Science and Technology, Osaka University  
1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan  
Phone: +81-6-6879-4542, Fax: +81-6-6879-4544  
{i-maki, murata}@ist.osaka-u.ac.jp

‡ Cybermedia Center, Osaka University  
1-32 Machikaneyama, Toyonaka,  
Osaka, 560-0043, Japan  
Phone: +81-6-6850-6864, Fax: +81-6-6850-6868  
hasegawa@cmc.osaka-u.ac.jp

§ System Platforms Research Laboratories,  
NEC Corporation  
1-1 Miyazaki 4-Chome, Miyamae-ku, Kawasaki,  
Kanagawa, 216-8555, Japan  
Phone: +81-44-856-2123, Fax: +81-44-856-2230  
murase@ccm.cl.nec.co.jp

**Abstract** TCP overlay network, which controls data transmission quality at the transport layer, is paid much attention as the users' demands for diversified services in the Internet becomes large. In the TCP overlay network, a TCP proxy is a fundamental mechanism which splits a TCP connection between sender and receiver hosts into multiple TCP connections at some nodes in the network and relays data packets from the sender host to the receiver host via the split TCP connections. In our previous work, we find that the end-to-end throughput of the TCP connection can be considerably improved by the TCP proxy mechanism. In this paper, we investigate the performance of the TCP proxy mechanism when it handles Web traffic and discuss the strategy for preventing the increase of document transfer delay due to the processing delay of a TCP proxy node.

**key words** TCP (Transmission Control Protocol), Overlay network, TCP proxy, Performance evaluation

## 1 はじめに

ADSL や FTTH といった広帯域アクセス網技術の進展により、近年ますますインターネットが発展し、ユーザ数の爆発的な増加に伴い、要求されるサービスが多様化している。それらの中には、エンドホスト間のスループットなどに関して高いネットワーク品質を要求するサービスもあるが、現在のインターネットはベストエフォート型であり、ユーザの要求品質を満たすことはできない。

この問題を解決し、IP 層において品質制御を行う技術として IntServ [1] や DiffServ [2] などが存在する。例えば DiffServ では、サービスの種類によってルータにおけるパケット処理の優先順位を決定することによって、各フローの通信品質の差別化を行うことを目的としている。しかしながら、IntServ や DiffServ を実現するためには、フローが通過するすべてのルータに品質制御機能が実装されている必要があり、ネットワーク規模に対するスケーラビリティ、導入コストなどの面から実現は困難であると考えられる。

また、IP 層より下位あるいは上位層において品質制御を行う技術も存在し、MPLS (Multi-Protocol Label Switching) [3] や GMPLS (Generalized MPLS) [4] はアンダーレイネットワーク技術の代表例である。例えば MPLS では、あるコネクションに属するパケットはルータごとに経路決定が行われるのではなく、エッジルータですべての経路決定が行われる。これらの技術は、データ転送が単一の ISP (Internet Service Provider) 内部で完結する場合には十分機能するが、複数の ISP を経由する場合には bandwidth broker [5] などの新たな制御機構が必要である。したがって、IP 層で実現する品質制御技術と同様にスケーラビリティなどに問題がある。

CDNs (Contents Delivery Networks) [6] におけるプロキシキャッシュサーバや P2P (Peer to Peer) ネットワークにおけるメディアストリーミング技術などはオーバーレイネットワーク技術の代表例である。オーバーレイネットワークにおいては、送信ホストから送信されたパケットはそのネットワークに属している他のホスト/ノードを経由して受信ホストに転送される。その際、パケットが転送される送受信ホスト間の経路は、オーバーレイネットワーク上の論理的な仮想パスから構成されている。そのため、オーバーレイネットワークが複数の ISP 上に構築された場合にも、既存の IP ネットワークを変更することなく、様々なサービスを提供することが可能である。オーバーレイネットワークにおいては、モニタリングやシグナリングによって下位の IP ネットワークから情報を取得することによって通信品質の制御を行う。例えば [7, 8, 9] に示されるようなアプリケーションオーバーレイネットワークにおいては、オーバーレイノード間のラウンドトリップ時間 (RTT) やホップカウントを用いることによって、オーバーレイネットワークの構築や送受信ノード間の適切な経路選択を行っている。また、RON (Resilient Overlay Network) [10] や FBR (Feedback Based Routing) [11] はルーティングオーバーレイネットワークの例である。これらは、オーバーレイノード間の経路の品質を計測によって獲得し、その情報をもとにオーバーレイノード間の経路選択を行っている。

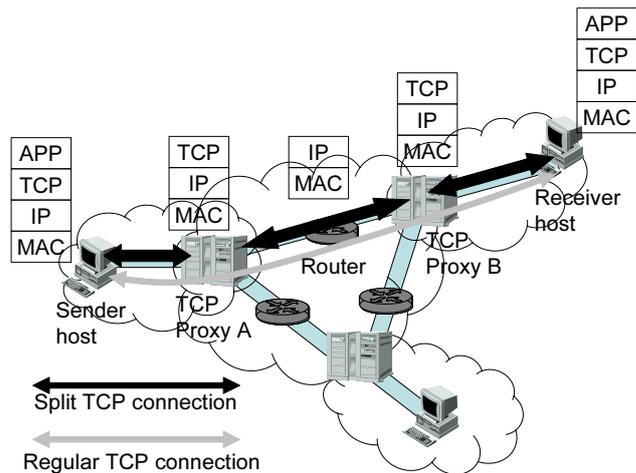


図 1: TCP オーレイネットワーク

しかしながらこれらのオーバーレイ技術は、ネットワーク品質の計測やオーバーレイノード間で計測結果を交換するための付加トラフィックやシグナリングメッセージが必要である。また、アプリケーション層において品質制御を行う場合、各アプリケーションに特化した複雑な制御を必要とする、所望の性能を得るためのパラメータセッティング等が困難である、などの問題がある。

本稿では、トランスポート層においてオーバーレイネットワークを構築する TCP オーレイネットワーク [12] に着目する。TCP オーレイネットワークにおいては、IP 層においては従来のルーティングなど最低限の機能のみを提供し、トランスポート層において品質制御を行う。そのため、IntServ や DiffServ のようにネットワーク層に新たな品質制御機能を必要としない。TCP オーレイネットワークにおいては、通常エンドホスト間に設定される TCP コネクションをネットワーク内のノード (TCP プロキシ) で終端し、分割されたコネクションごとにパケットを中継しながら転送を行う (図 1)。これにより、TCP コネクションのフィードバックループを小さくできるため、スループットの向上が期待できる。また、TCP オーレイネットワークを構築することによって、ネットワーク環境の違いを吸収・隠蔽することが可能になるため、要求されるサービス品質に応じた制御を行うことが容易になる。例えば、送受信ホスト間に無線ネットワークが含まれる場合、一般的にはそのネットワークの高いパケット廃棄率と大きな伝播遅延時間によって TCP コネクションのスループットは大幅に低下する。しかし、無線ネットワーク部分でデータ転送が独立するように、その前後でコネクション分割を行うことにより、性能劣化を最小限に抑えることが可能である。

また、アプリケーションは下位の TCP オーレイネットワークを利用することによる恩恵を享受することができる。まず第一に、TCP プロキシ間に設定された TCP コネクションから得られる情報を用いるため、IP ネットワークの品質の計測や経路選択といった付加的な機能を必要としない。第二に、TCP 層から得られる情報は IP 層から得られる情報よりもアプリケーションにとって適切で

ある。例えば、トランスポート層の情報として得られる TCP スループットは、RON や FBR では情報が不正確であったり、情報そのものが得られない。

さらに、TCP オーバレイネットワークは高い拡張性を持つ。TCP プロキシの導入に際しては、IntServ や DiffServ のようにすべてのルータに付加的な制御機能を必要としない。そのため、ネットワーク内に単一の TCP プロキシしか配置されていない場合にも、様々なサービスが提供可能である。ネットワーク内の TCP プロキシの数が増加すれば、それにともない得られる性能はさらに大きくなる。したがって、複数の ISP を経由するデータ転送においても、TCP プロキシを段階的に導入することによって徐々にデータ転送効率が向上する。さらに、TCP プロキシがユーザの設定する TCP コネクションを自動的に分割するため、エンドホストのプロトコルスタックの変更を必要としない。

TCP コネクションを分割し、各コネクションに応じた処理を行うことでデータ転送速度の向上を目指す研究はこれまでも行われている [13, 14, 15, 16]。しかしながら、それらの研究では衛星回線、無線回線など特定のネットワークを対象とするものや [13, 14]、コネクション分割によるデータ転送効率が向上することの利点のみが示されており、分割・中継処理において新たに発生する問題が考慮されていない [15, 16]。分割・中継処理を行う際には、既存システムの変更を最小限にとどめるために、TCP の輻輳制御アルゴリズムが各中継 TCP コネクションで独立に動作するため、それらが互いに干渉し、期待するほどのスループットが得られないという問題が生じることが考えられる。文献 [17] において我々は、そのような中継処理の問題を考慮したエンドホスト間のスループット解析を行い、その結果、TCP コネクション分割機構を用いた場合には、TCP プロキシの前後の分割 TCP コネクションが通過するネットワーク状況の差異が少ないときに、大きなスループット低下が発生することが分かった。また、TCP プロキシの送信バッファを大きくすることでその低下を抑えることが可能であることを明らかにした。

本稿では、多くの Web コネクションのように転送するドキュメントサイズが小さい TCP コネクションの転送遅延時間に着目した、TCP コネクション分割機構の評価を行う。転送データサイズが小さな TCP コネクションを分割する場合、TCP プロキシにおける処理遅延やバッファリング遅延により、分割を行わない通常のコネクションよりも転送遅延時間が大きくなることが考えられる。そのため、転送遅延時間の増大を防ぐための手法を提案し、その有効性を示す。

以下、2章では TCP オーバレイネットワークにおいて必要不可欠である TCP コネクション分割機構について説明する。3章では生存時間の長い TCP コネクションに着目した平均スループット解析の概要を示す。また、転送するドキュメントサイズが小さい TCP コネクションに着目した転送遅延時間の評価を行う。最後に 4章でまとめと今後の課題を述べる。

## 2 TCP コネクション分割機構

TCP コネクション分割機構は TCP オーバレイネットワークを構築するための基本的な技術要素である。通常、エンドホスト間に一本設定される TCP コネクションを、ネットワーク内のノード (TCP プロキシ) で終端することによって複数コネクションに分割し、分割されたコネクションごとにパケットを中継しながらデータ転送を行うことを可能にする。また、TCP コネクション分割機構は受信したパケットを中継転送するだけでなく、受信側ホストに代わって疑似的な ACK パケットを返信する機能を有している。これにより、送信側ホストは受信側ホストからの ACK パケットを待つことなく新たなデータパケットの送信が可能となるため、送信側 TCP が持つ見かけの RTT が小さくなり、データ転送効率が向上する。さらに、コネクション分割をしている TCP プロキシは、データパケットに対する ACK パケットを受信するまでそのデータパケットをバッファリングするため、例えば TCP プロキシと受信側ホスト間でパケットが廃棄された場合に、TCP プロキシから廃棄パケットを再送がすることが可能である。

## 3 性能評価

### 3.1 平均スループット解析

#### 3.1.1 概要

コネクション間の干渉によるスループット低下は、TCP コネクション分割機構の送信バッファが空になることで、受信側コネクションに転送すべきデータがなくなり、データの中継転送が一時的に停止することが大きな原因である。これは、送信側コネクションの輻輳によるパケットロスが原因で生じ、TCP コネクション分割機構の受信バッファから送信バッファへのデータ移動が行われなくなるためである。この問題を考慮した場合の平均スループットは、送信ホストに設定されたコネクションから順に受信ホストに設定されたコネクションまで計算し、受信ホストに設定されたコネクションのスループットをある計算過程での送受信ホスト間の平均スループット  $\rho_e[h]$  とする。 $h$  は  $h$  回目の計算過程を表す。このとき計算したスループットが以下の不等式を満たす場合に計算を終了する。

$$\left| \frac{\rho_e[h-1] - \rho_e[h]}{\rho_e[h-1]} \right| < \epsilon$$

エンドホスト間に設定されるコネクションを分割した場合の、分割後のあるコネクションに着目する。このコネクションのスループットは、A) 送信側よりのコネクションのスループットで抑えられる B) 送信側よりのコネクションの輻輳により、さらにスループットが低下する可能性がある C) TCP コネクション分割機構の受信バッファの空き容量で抑えられる D) 着目するコネクションの経由するネットワーク状況に左右する、の 4 つの場合が考えられる。B では、送信側コネクションでパケット廃棄が発生してから次にパケット廃棄が発生するまでの時間を考える。送信側コネクションでパケット廃棄が発生したときに TCP コネクション分割機構の送信バッファに格納されているパケットが  $j$  個である確率を  $p_{s(j)}$ 、このと

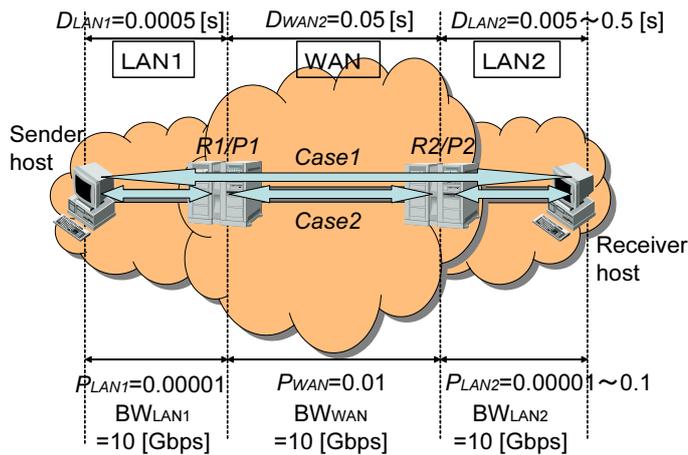


図 2: 解析モデル

き次に送信側接続でパケット廃棄が発生するまでの時間の着目する接続の平均スループットを  $\rho_{k(j)}$  とすると、着目する接続の平均スループットは以下のように計算できる。

$$\rho_k [h] = \sum_{i=0}^{SB(k-1)} P_{s(i)} \cdot \rho_{k(i)}$$

$SB(k-1)$  は TCP 接続分割機構の送信バッファの容量である。解析の詳細に関しては文献 [17] を参照されたい。

### 3.1.2 解析

TCP 接続分割機構を用いることによりエンドホスト間のスループットが向上することを、解析による数値例を用いて示す。解析モデルは図 2 に示す通りであり、 $D_{LAN1}$ 、 $D_{WAN}$ 、 $D_{LAN2}$  を各リンクの伝播遅延時間 [ms]、 $P_{LAN1}$ 、 $P_{WAN}$ 、 $P_{LAN2}$  を各リンクのパケット廃棄率、 $BW_{LAN1}$ 、 $BW_{WAN}$ 、 $BW_{LAN2}$  を各リンクの帯域 [Mbps] とする。このとき、ルータ  $R1$  および  $R2$  を経由してエンドホスト間に一本接続が設定される場合 (Case1) とエンドホスト間に一本設定される接続を TCP プロキシ  $P1$  および  $P2$  の 2 箇所で接続分割する場合を考える。このとき、文献 [17] で指摘した中継処理によるスループット低下を考慮するものとする (Case2)。送信すべきデータは無限にあると仮定する。

図 3 は  $P_{LAN2}$  および  $D_{LAN2}$  を変化させた場合の、TCP 接続分割機構によるスループット増加割合、すなわち、Case1 における平均スループットに対する Case2 における平均スループットの割合を示している。Case1 における平均スループットは文献 [18]、Case2 は文献 [17] による解析結果を利用している。図より、Case2 における平均スループットは Case1 よりも常に大きく、最大で 3 倍以上になっていることが分かる。これは、分割後の接続間の干渉によるスループット低下を考慮した場合にも、TCP 接続分割機構が有効であることを示している。

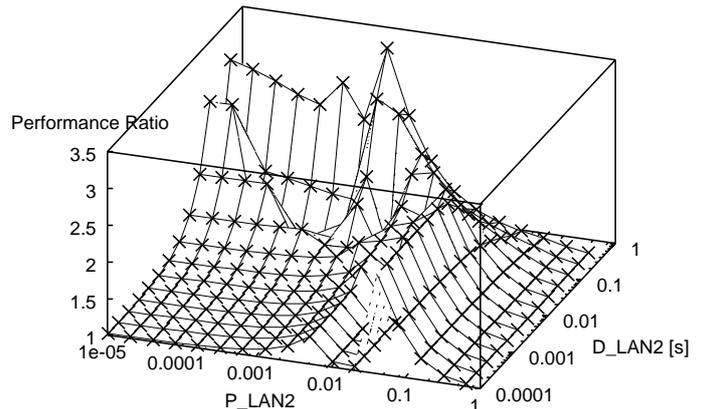


図 3: TCP 接続分割機構によるスループット向上の割合

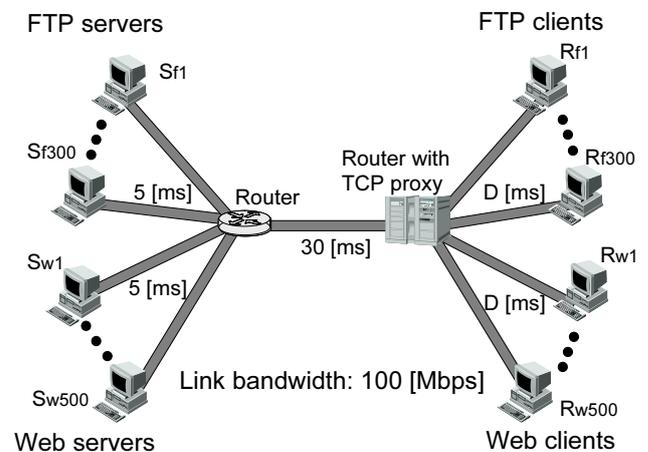


図 4: シミュレーションモデル

### 3.2 転送遅延時間評価

本節では、転送遅延時間に関して NS シミュレータ [19] を用いて評価を行う。本稿では、SYN を送信し始めてから要求したドキュメントをすべて受信するまでの時間を転送遅延時間として定義する。図 4 のシミュレーションモデルを用い、 $S_{f1} \sim S_{f300}$  は FTP サーバ、 $R_{f1} \sim R_{f300}$  は FTP クライアント、 $S_{w1} \sim S_{w500}$  は Web サーバ、 $R_{w1} \sim R_{w500}$  は Web クライアントであるとする。 $D$  は受信側ホストのネットワークの伝播遅延時間 [ms] とする。FTP 接続はすべて接続分割され、シミュレーション開始から 1 秒後までに SYN パケットを送信し、これらの接続が十分安定した状態で Web 接続が SYN パケットを送信し始めるものとする。Web サーバから転送するドキュメントサイズ分布およびドキュメントの転送間隔は文献 [20] において得られている結果を利用する。また、接続分割を行う場合、接続設定時に TCP プロキシにおいて行う処理を考慮し、SYN および SYN/ACK パケットの処理時間を、通常のデータパケットの 10 倍とする。

TCP プロキシを通過する接続をすべて分割すると、TCP プロキシの負荷が増大し、バッファリング遅延がボトルネックになる可能性がある。そのため、TCP

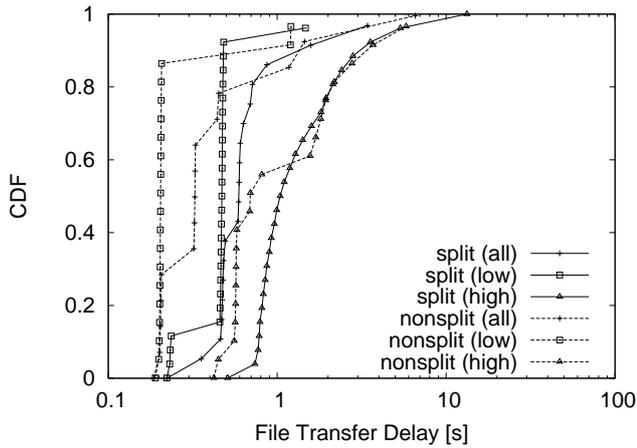


図 5: 方針 A における転送遅延時間分布

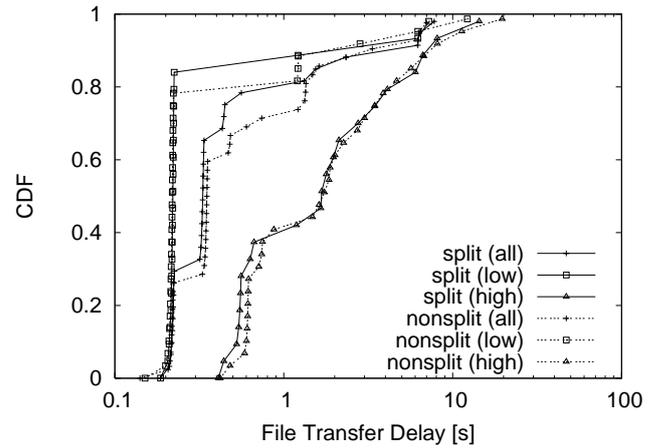


図 7: 方針 C における転送遅延時間分布

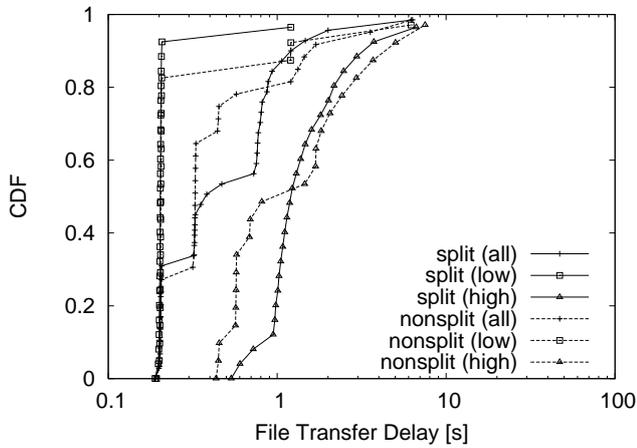


図 6: 方針 B における転送遅延時間分布

プロキシを通過する接続のうち一部の接続のみを分割することが考えられる。以下では、一般的に想定される 3 種類の分割方針 A) 一部のユーザの接続のみ分割を行う、B) ドキュメントサイズの大きい接続のみ分割を行う、C) TCP プロキシの負荷が低い場合にのみ分割を行う、場合の転送遅延時間を評価する。方針 A では、300 ユーザの接続を TCP プロキシで分割する。方針 B では、ドキュメントサイズの小さい接続の転送時間が大きくなるように、ドキュメントサイズが 2000 バイト以上である接続のみ分割する。また方針 C では、TCP プロキシの負荷が 90% 未満の場合に接続を分割する。TCP プロキシの負荷はプロキシノードの CPU 利用率と考えることができるが、本稿では CPU 負荷を 1 秒間に TCP プロキシに到着するパケットの数と考える。時刻  $T$  [s] における TCP プロキシの負荷  $L(T)$  は以下のように計算する。

$$L(T) = \text{SampleLoad}(T) \quad (T = 1)$$

$$L(T) = \frac{7}{8}L(T-1) + \frac{1}{8}\text{SampleLoad}(T) \quad (T = 2, 3, \dots)$$

ここで、 $\text{SampleLoad}(T)$  は時刻  $T-1 \sim T$  の 1 秒間に TCP プロキシに到着したパケット数とする。ある接続の SYN パケットが到着したとき、TCP プロキシで

の 1 パケットあたりの処理時間が  $t$  [s] であるとする、上式で与えられる  $L(T)$  が

$$L(T) < \frac{1}{t} \cdot \frac{90}{100}$$

を満たすとき、TCP プロキシの負荷が低いと判断してその TCP コネクションを分割する。SYN パケットを受信した場合は、データパケットに対して 10 倍の処理時間を必要とすると仮定したため、 $\text{SampleLoad}(T)$  の値を 10 増加させる。

図 5~7 はそれぞれ、 $D = 5$  のときの方針 A、B および C を用いた場合の転送遅延時間に関する累積度数分布を示している。 $\text{split}(\text{low})$  は分割された Web コネクションのうちドキュメントサイズ分布の下位 10%、 $\text{split}(\text{high})$  は上位 10%、 $\text{split}(\text{all})$  は全体の転送遅延時間に対する累積度数分布を示している。同様に、 $\text{nonsplit}$  は同じシミュレーションにおいて分割されなかった Web コネクションの転送遅延時間を示している。図 5、6 においては、分割された Web コネクションの転送遅延時間が分割されなかった Web コネクションよりも大きくなっていることが分かる。これは、TCP プロキシの負荷が大きく、TCP プロキシで大きな遅延が発生しているためである。これに対して図 7 においては、分割されたコネクションの転送遅延時間が分割されなかったコネクションよりも小さくなること分かる。これは、TCP プロキシの CPU 負荷が 100% を超えないように制御し、TCP プロキシでバッファリング遅延が発生しないためである。

図 8 は  $D = 30$  とし、受信側のネットワークにパケット廃棄率として明示的に 0.01 を与えたときの転送遅延時間分布を示している。これは、ユーザが衛星あるいは無線ネットワークを経由してインターネットアクセスしているような状況を想定している。図 8 より、図 7 と比較して分割されたコネクションと分割されなかったコネクションの転送遅延時間の差が大きくなっていることが分かる。これは、分割後のコネクションの RTT やパケット廃棄率の差が小さくなるためであり、平均スループット評価においても同様の傾向が見られている [17]。

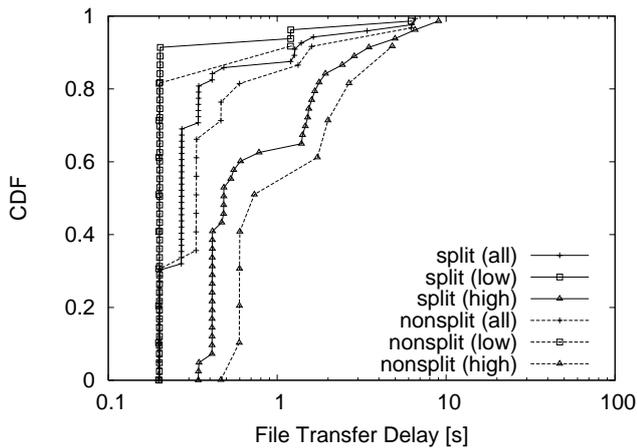


図 8: 受信側のネットワークの packets 廃棄率・RTT が大きい場合のドキュメント転送時間分布

#### 4 まとめと今後の課題

本稿では、TCP オーバレイネットワークにおける基本的な技術である TCP コネクション分割機構の性能評価を行った。TCP コネクション分割機構を用いることにより、エンドホスト間にコネクション設定する場合と比較して平均スループットが向上することを示した。また、TCP プロキシのオーバーロードが性能に大きな影響を与えることを明らかにし、それを防止するための有効手法を提案した。今後は、分散配置された TCP プロキシの効率的な利用手法の検討を行う予定である。

#### 謝辞

本研究の一部は、総務省における研究プロジェクトである戦略的情報通信研究開発推進制度委託研究「ユビキタスインターネットにおける高位レイヤスイッチの研究開発」によっている。ここに記して謝意を表す。

#### 参考文献

- [1] J. Wroclawski, “The use of RSVP with IETF integrated services,” *RFC 2210*, Sept. 1997.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated service,” *RFC 2475*, Dec. 1998.
- [3] E. C. Rosen, A. Viswanathan, and R. Callon, “Multi-protocol label switching architecture,” *RFC 3448*, Jan. 2001.
- [4] E. Mannie, “Generalized Multi-Protocol Label Switching architecture,” *IETF Internet Draft draft-ietf-ccamp-gmpls-architecture-07.txt*, May 2003.
- [5] R. Bless and K. Wehrle, “IP multicast in differentiated services networks,” *IETF Internet Draft draft-ietf-diffserv-multicast-00.txt*, Nov. 2000.
- [6] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, “Removal policies in network caches for World-Wide Web documents,” in *Proceedings of ACM SIGCOMM’96*, 1996.

- [7] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing,” tech. rep., Technical Report UCB/CSD-01-1141 UC Berkeley, Apr. 2001.
- [8] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” *Lecture Notes in Computer Science*, vol. 2218, pp. 329–350, 2001.
- [9] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, “Topologically-aware overlay construction and server selection,” in *Proceedings of IEEE INFOCOM’02*, June 2002.
- [10] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, “Resilient overlay networks,” in *Proceedings of ACM SOSP’01*, Oct. 2001.
- [11] D. Zhu, M. Gritter, and D. R. Cheriton, “Feedback based routing,” *ACM SIGCOMM Computer Communication Review*, vol. 33, pp. 71–76, Jan. 2003.
- [12] 村瀬 勉, 下西 英之, 長谷川 洋平, “TCP オーバレイネットワークの提案,” 電子情報通信学会 通信ソサイエティ大会報告 (B-7-49), Sept. 2002.
- [13] A. Barkre and B. Badrinath, “I-TCP: Indirect TCP for mobile hosts,” in *Proceedings of 15th International Conference on Distributed Computing Systems*, pp. 136–143, June 1995.
- [14] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Transactions on Networking*, vol. 5, pp. 756–769, Aug. 1997.
- [15] R. Cohen and S. Ramanathan, “Using proxies to enhance TCP performance over hybrid fiber coaxial broadband access networks,” *Computer Communications*, vol. 20, Jan. 1998.
- [16] P. Rodriguez, S. Sibal, and O. Spatscheck, “TPOT: translucent proxying of TCP,” *Computer Communications*, vol. 24, no. 2, pp. 249–255, 2001.
- [17] 牧 一之進, 長谷川 剛, 村田 正幸, 村瀬 勉, “TCP オーバレイネットワークにおける TCP コネクション分割機構の性能解析,” 電子情報通信学会 技術研究報告 (IN03-198), Feb. 2004.
- [18] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation,” in *Proceedings of ACM SIGCOMM’98*, pp. 303–314, Sept. 1998.
- [19] “UCB/LBNL/VINT network simulator - ns (version 2).” available at <http://www.mash.cs.berkeley.edu/ns/>.
- [20] P. Barford and M. Crovella, “Generating representative Web workloads for network and server performance evaluation,” in *Proceedings of ACM SIGMETRICS International Conference on Measurement and Model of Computer Systems*, pp. 151–160, July 1998.