

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

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
〒 560-8531 大阪府豊中市待兼山町 1-3
Phone: 06-6850-6587, Fax: 06-6850-6589
i-maki@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: 06-6850-6863, Fax: 06-6850-6868
{hasegawa, murata}@cmc.osaka-u.ac.jp

§ 日本電気株式会社 システムプラットホーム研究所
〒 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、ユーザの要求するサービスは多種多様で高度になっている。この要求に応えるため、我々はアプリケーション層や IP 層で品質制御を行うのではなく、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関する研究を行っている。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) で終端することによって TCP コネクションを複数に分割し、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転送を行う。その結果、エンドホスト間のデータ転送におけるスループットの向上が期待できる。しかし、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を各中継ノードにおいて独立に動作させる場合、それらが互いに干渉し、その結果期待するほどのスループットが得られないことが明らかになった。本稿では、こ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示し、その妥当性をシミュレーションとの比較により検証する。また解析結果から、効果的な TCP コネクション分割機構の導入シナリオについて議論する。

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、 解析

Performance Analysis of TCP Proxy Mechanism in TCP Overlay Networks

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

†Graduate School of Information Science and Technology, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
Phone: +81-6-6850-6587, Fax: +81-6-6850-6589
i-maki@ist.osaka-u.ac.jp

‡Cybermedia Center, Osaka University
1-32 Machikaneyama, Toyonaka,
Osaka, 560-0043, Japan
Phone: +81-6-6850-6863, Fax: +81-6-6850-6868
{hasegawa, murata}@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. It is expected to enhance the end-to-end throughput of the TCP connection because of the reduction of the Round Trip Time and the packet dropping ratio of each split TCP connection, but performance degradation may happen due to the undesired interaction among the split TCP connections. In this paper, we introduce an analysis approach to estimate the end-to-end throughput of the TCP proxy mechanism in consideration of the performance degradation. We also discuss about the effectiveness of TCP proxy mechanism through our analysis results.

key words TCP (Transmission Control Protocol), Overlay network, TCP proxy, Analysis

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。一方、CDN (Contents Delivery Network) におけるプロキシキャッシュサーバなどのように、品質制御をアプリケーション層で行う技術も研究されている [3] が、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設定等が困難である、などの問題がある。

そこで我々は、IP 層やアプリケーション層において品質制御を行うのではなく、IP 層においては従来のルーティングなど必要最低限の機能のみを提供し、品質制御をトランスポート層において行う TCP オーバレイネットワーク [4] に関する研究を行っている。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 (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することが可能になるため、スループットの向上を期待することができる。また、TCP オーバ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可能になる。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 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。

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 [5-8]。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 [5, 6]、コネクション分割によるデータ転送効率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 [7, 8]。分割・中継処理を行う際に、既存システムの変更を最小限にとどめるためには、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。

そこで本稿では、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し、コネクション分割を行うことによりエンドホスト間のデータ転送速度が向上することを、簡単な数値例を用いて示す。しかし、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を各中継ノードにおいて独立に動作させる場合、それらが互いに干渉し、その結果期待するほどのスループットが得られない。そこで、こ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示し、その妥当性をシミュレーションとの比較により検証する。また解析結果から、効果的な TCP コネクション分割機構の導入シナリオについて議論する。

2. TCP コネクション分割機構

TCP コネクション分割機構は、エンドホスト間に設定される TCP

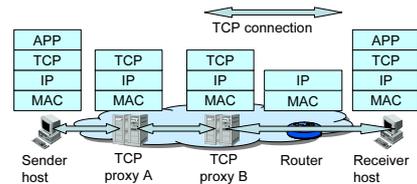


図 1 TCP オーバレイネットワーク

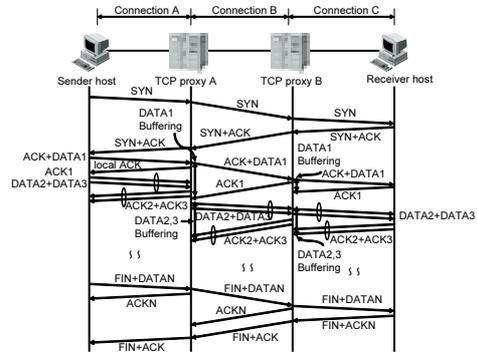


図 2 TCP コネクション分割機構

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 TCP コネクションを複数に分割し、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実現する機構であり、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。図 2 は、図 1 に示すようにエンドホスト間に設定された TCP コネクションを TCP プロキシ A, B の 2 個所で分割した場合の、パケット中継の様子を表している。ここで、送信側ホストから近い順番に中継 TCP コネクション A, B, C と呼ぶことにする。このとき、中継 TCP コネクション A を用いて送信側ホストからパケットが送信されて TCP プロキシ A が受信すると、TCP プロキシ A はそのパケットを中継 TCP コネクション B へ中継転送する。同様に、TCP プロキシ B は中継 TCP コネクション C へ受信したパケットを中継転送する。

また、TCP コネクション分割機構ではこのように受信したパケットを中継転送するだけでなく、受信側ホストにかわって疑似的な ACK パケットを返信する機能を有している。この ACK パケットを local ACK パケット [9] と呼ぶ。例えば、送信側ホストは受信側ホストからの ACK パケットを待たずに、TCP プロキシ A からの local ACK パケットにより新たなデータパケットの送信を行うことができるため、送信側ホストの持つ RTT が小さくなり、データ転送効率が向上する。同様に、中継 TCP コネクション B および C のデータ転送効率も向上するため、エンドホスト間のデータ転送効率が向上する。

さらに、各 TCP コネクション分割機構では、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 (図 2)。例えば TCP プロキシ B と受信側ホスト間でパケットが廃棄された場合には、中継 TCP コネクション C がそれを検出し、廃棄パケットを再送することができるため、送信側ホストから再送を行う現在の方式と比較してデータ転送速度が向上する。local ACK パケットを用いると、送信側ホストは受信側ホストにデータパケットが到達する前に、そのデータパケットに対応する ACK パケットを受信するため、厳密には TCP のセマンティクスに違反しているが、再送は各中継 TCP コネクションにおいて確実かつ局所的に行われるためエンドホスト間のデータ転送の信頼性は保持される。また、コネクション設定・切断時に送信される SYN・FIN パケットに対しては、local ACK パケットを用いずに中継転送されるため、コネクションの設定・切断も確実に実行される。

ここで、TCP コネクション分割機構を用いてデータ転送を行い、局所的な再送や local ACK パケットを利用することによって性能向上が期待できることを、簡単な数値例によって示す。この数値例では、エンドホスト間に設定された TCP コネクションを

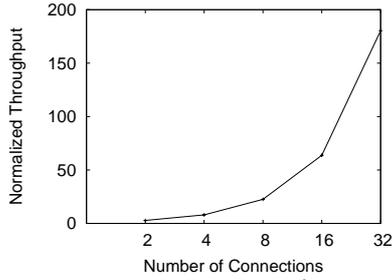


図3 コネクション分割によるスループット向上の効果

ネットワーク内で分割した際の各中継 TCP コネクションのスループットを、文献[10]で示された(1)式を用いて計算し、その中で最小値をとる中継 TCP コネクションのスループットをエンドホスト間のスループット ρ_e とする((2)式)。ここで b は delayed ACK のオプションであり、受信側ホストは b 個の packets を受け取るごとに 1 個の ACK パケットを返す。 T_o は初期タイムアウト時間である。また、 m 分割した場合の k 番目のコネクションの packet 廃棄率、ラウンドトリップ時間(RTT)を、それぞれ p_k 、 RTT_k とする。

$$\rho_k = \frac{1}{RTT_k \sqrt{\frac{2bp_k}{3}} + T_o \cdot \min\left(1, 3\sqrt{\frac{3bp_k}{8}}\right) p_k (1 + 32p_k^2)} \quad (1)$$

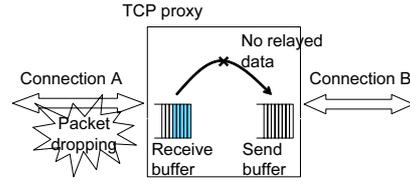
$$\equiv \text{TCP_RHO}(RTT_k, p_k, T_o, b) \quad (1)$$

$$\rho_e = \min_{1 \leq k \leq m+1} \rho_k \quad (2)$$

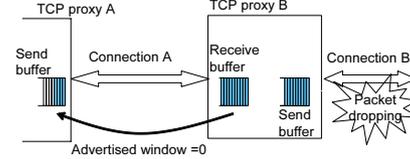
図3は32ホップの直線的なネットワークにおいて、ネットワーク内にある TCP コネクション分割機構で、エンドホスト間に設定された TCP コネクションを、各中継 TCP コネクションのホップ数が等しくなるように分割した場合のコネクション分割機構の効果を示したものである。横軸は分割後の中継 TCP コネクションの本数、縦軸は分割前のエンドホスト間のスループットに対する分割後のエンドホスト間のスループットの割合である。また、各ホップのリンクの帯域、パケット廃棄率、RTTはそれぞれ、100 [Mbps]、0.0005、0.01[s]とする。この図より、コネクション分割数を増やすことでスループットが大きく向上していることが分かる。これは、各中継 TCP コネクションの RTT だけでなく、パケット廃棄率も同時に小さくなるためである。

以上のように、エンドホスト間に設定される TCP コネクションを分割することによって、理論的にはエンドホスト間のスループットが大きく向上する。しかし、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を各中継ノードにおいて独立に動作させる場合、それらが互いに干渉し、中継処理によって期待するほどのスループットが得られない。その主な原因は次の2つである。図4(a)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、コネクション B 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して、スループットが低下する様子を示している。これは、コネクション A 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の移動が行われなくなるためである。その結果、TCP コネクション分割機構によって得られるスループットが、コネクション B のスループットよりも低くなることが考えられる。

また図4(b)のように、TCP コネクション分割機構から広告ウィンドウサイズが 0 である local ACK パケットが返されることによって、スループットが低下することが考えられる。この ACK パケットを受信したコネクション A はデータの中継転送が一時的に停止するため、その間にコネクション B が TCP プロキシ B の送受信バッファに格納されていた packets をすべて転送してしまっただけでスループットが低下することが考えられる。広告ウィンドウサイズが 0 である local ACK パケットが返されるのは、コネクション B の輻輳によるパケットロスにより、TCP プロキシ B の送信バッファが一杯になり、さらに受信バッファが一



(a) 問題点 1



(b) 問題点 2

図4 TCP コネクション分割機構の中継処理の問題点

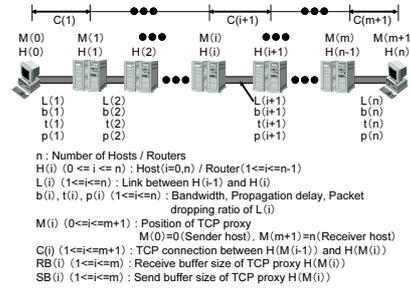


図5 解析モデル

杯になるためである。FreeBSD など多くの OS の実装では、送信側ホストに広告ウィンドウサイズが 0 である ACK パケットが返信された場合にも、受信側ホストが受信バッファに空きができるたびに ACK パケットを返信する[11]。そのためデータの中継転送が停止する時間は短く、スループット低下はほとんど発生しないと考えられる。

また、コネクション B においてタイムアウトなどによって一時的にデータ転送が停止した際に、コネクション A からの packet 転送速度が大幅に低下する問題が文献[12]において指摘されている。本稿では、図4(a)に示した問題を考慮した解析を行う。

3. スループット解析

3.1 解析モデル

解析には図5のネットワークモデルを用いる。解析は文献[10]で示されている TCP コネクションの平均スループットに関する解析結果を利用するため、この文献で挙げられている仮定に関しては、スペースの都合上割愛する。本解析では、次の2つの仮定を新たに導入する。TCP コネクション分割機構では local ACK を返信するものとし、そのタイミングはデータパケットが受信バッファに格納された時点であるとし、TCP コネクション分割機構での処理時間は1のものとする。また本解析では、エンドホスト間に設定された1本のコネクションを対象とする。

3.2 解析手法

3.2.1 概要

n ホップの送受信ホスト間に設定された TCP コネクションを m 分割する場合を考える。このとき、TCP コネクション分割機構の中継処理の問題を考慮したエンドホスト間の平均スループット ρ_e の導出は以下のように行う。まず、TCP コネクション分割機構の packet の中継処理が問題とならない、すなわち受信バッファに packet がたまず、送信バッファに常に packet があると仮定した場合の各コネクションの平均スループット $\rho_k[0]$ ($1 \leq k \leq m+1$) を計算し、これを各コネクションの packet 転送レート s_k ($1 \leq k \leq m+1$) とする。また、 $\min_{1 \leq k \leq m+1} \rho_k[0]$

を中継処理の問題を考慮しない場合のエンドホスト間の平均スループット $\rho_e[0]$ とする。次に、図 4(a) に示した問題によるスループット低下を考慮した場合のエンドホスト間の平均スループットを計算し、これを $\rho_e[h]$ とする。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 (このときの $\rho_e[h]$ を ρ_e とする)。不等式を満たさない場合は繰り返し計算を行う。ここで、 h は h 回目の計算過程を表す。

$$\left| \frac{\rho_e[h-1] - \rho_e[h]}{\rho_e[h-1]} \right| < \epsilon \quad (3)$$

3.2.2 解 析

まず、TCP コネクション分割機構の packets の中継処理が問題とならない場合の送受信ホスト間の平均スループット $\rho_k[0]$ を計算する。 m 分割したうち、 k 番目のコネクション C_k に着目すると、このコネクションのスループットが経由するリンクの帯域遅延種で制限されないと仮定すれば、(1) 式と同様にして計算でき、これを $\rho'_k[0]$ とする。このとき、 p_k および RTT_k は以下のように計算する。

$$\begin{aligned} p_k &= 1 - (1 - p(M(k-1) + 1)) \cdot (1 - p(M(k-1) + 2)) \cdot \dots \cdot (1 - p(M(k))) \\ &= 1 - \prod_{i=1}^{M(k)-M(k-1)} (1 - p(M(k-1) + i)) \\ RTT_k &= 2 \sum_{i=M(k-1)+1}^{M(k)} t(i) \end{aligned}$$

コネクション C_k のスループットが経由するリンクの帯域遅延種で制限される場合、このコネクションの平均スループット $\rho_k[0]$ は経由するリンクのうち最小の帯域となるので以下ようになる。

$$\rho_k[0] = \min_{M(k-1)+1 \leq i \leq M(k)} b(i)$$

また、コネクション C_k のスループットが TCP コネクション分割機構 $H(M(k))$ の受信バッファサイズ $RB(k)$ で制限される場合、このコネクションの平均スループット $\rho_k[0]$ は以下ようになる。

$$\rho_k[0] = \frac{RB(k)}{RTT_k}$$

したがって、 $\rho_k[0]$ および $\rho_e[0]$ は次のように計算できる。

$$\rho_k[0] = s_k = \min(\rho'_k[0], \rho''_k[0], \rho'''_k[0])$$

$$\rho_e[0] = \min_{1 \leq k \leq m+1} \rho_k[0]$$

次に、図 4(a) に示した問題によるスループット低下を考慮した場合の平均スループット $\rho_e[h]$ を計算する。このとき、コネクション C_1 から順に C_{m+1} まで計算し、 C_{m+1} 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。コネクション C_1 のスループットは、TCP コネクション分割機構 $H(M(1))$ の受信バッファの平均空き容量でスループットが制限される可能性がある。TCP コネクション分割機構 $H(M(1))$ の受信バッファへの平均到着率を平均スループット $\rho_e[h-1]$ 、送信バッファでの平均サービス率をコネクション C_2 の packet 転送レート s_2 として、送受信バッファでの振舞いを M/M/1/K でモデル化し、受信バッファの平均空き容量 $RB_r(1)$ を計算する。

$$RB_r(1) = \min \left(RB(1), K - \frac{\rho \{1 - (K+1)\rho^K + K\rho^{K+1}\}}{(1-\rho)(1-\rho^{K+1})} \right)$$

ここで、 $K = SB(1) + RB(1)$ 、 $\rho = \rho_e[h-1]/s_2$ とする。したがって、この場合のコネクション C_1 の平均スループットを次のように計算する。

$$\rho_1[h] = \min \left(\text{TCP_RHO}(RTT_1, p_1, T_o, b), \right.$$

$$\left. \min_{M(0)+1 \leq i \leq M(1)} b(i), \frac{RB_r(1)}{RTT_1} \right)$$

次に、コネクション C_k ($2 \leq k \leq m+1$) に関して、 h 回目の

計算過程に着目して説明を行う。コネクション C_k の平均スループット $\rho_k[h]$ を求めるために、まず TCP コネクション分割機構 $H(M(k-1))$ の packets の中継処理でスループットが低下することを考える。このときコネクション C_{k-1} で packet 廃棄が発生してから、次に packet 廃棄が起こるまでの時間を考える。コネクション C_{k-1} で packet 廃棄が発生したときに TCP コネクション分割機構 $H(M(k-1))$ の送信バッファに格納されている packets が j 個である確率を $p_{s(j)}$ 、このとき次にコネクション C_{k-1} で packet 廃棄が起こるまでの時間のコネクション C_k の平均スループットを $\rho_{k(j)}$ とすると、コネクション C_k の平均スループットは以下のように計算できる。

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

以降では $p_{s(j)}$ 、 $\rho_{k(j)}$ の導出方法に関して説明する。TCP コネクション分割機構 $H(M(k-1))$ の送信バッファへの平均到着率をコネクション C_{k-1} の平均スループット $\rho_{k-1}[h]$ 、送信バッファでの平均サービス率をコネクション C_k の packet 転送レート s_k として、送信バッファでの振舞いを M/M/1/K でモデル化すると、送信バッファに j 個の packets が格納されている確率は、

$$p_{s(j)} = \begin{cases} \frac{\rho'^j (1 - \rho')}{1 - \rho'^{K+1}} & \rho' \neq 1 \\ \frac{1}{K+1} & \rho' = 1 \end{cases}$$

となる。ただし、 $K = SB(k-1)$ 、 $\rho' = \rho_{k-1}[h]/s_k$ とする。次に $\rho_{k(j)}$ について考える。ここでは、コネクション C_{k-1} で packet 廃棄が発生してからその再送 packets が TCP コネクション分割機構 $H(M(k-1))$ に到着するまでの時間とそれ以降に分けて考える。前半の平均時間を T_f 、その時間内にコネクション C_k が送信した packet 数を N_f とし、後半の平均時間を T_l 、その時間内に送信した平均 packet 数を N_l とする。送信バッファが空にならない場合、つまりコネクション C_{k-1} の再送 packets が到着するまでの間に、送信バッファに格納されている packets をコネクション C_k がすべて送信しなければスループット低下は発生しないため、以下の条件を満たすような j が存在するとき、 $\rho_{k(j)} = s_k$ となる。

$$T_f \cdot s_k < j \quad (0 \leq j \leq SB(k-1))$$

また、送信バッファが空になる場合は $\rho_{k(j)}$ を以下のように計算することができる。

$$\rho_{k(j)} = \frac{N_f + N_l}{T_f + T_l}$$

以下では、これら N_f, T_f, N_l, T_l を順に求める。 N_f はコネクション C_{k-1} で packet 廃棄が発生したときに TCP コネクション分割機構 $H(M(k-1))$ の送信バッファに格納されていた packet 数である。したがって、

$$N_f = j$$

となる。fast retransmit によって廃棄された packets の再送が行われる場合は、再送 packets が到着するまでに RTT_{k-1} 時間かかるとし、タイムアウト後に再送が行われる場合はタイムアウト時間とコネクション C_{k-1} が経由するリンクの伝播遅延時間だけの時間がかかるとすると、 T_f は以下のように計算することができる。

$$T_f = \left(1 - \min \left(1, \frac{3}{w(k-1)} \right) \right) RTT_{k-1}$$

$$+ \min \left(1, \frac{3}{w(k-1)} \right) \left(T_o + \frac{1}{2} RTT_{k-1} \right)$$

ここで $w(k-1)$ はコネクション C_{k-1} で packet 廃棄が発生した時のコネクション C_{k-1} のウィンドウサイズであり、 $\min(1, 3/w(k-1))$ はコネクション C_{k-1} が packet 廃棄の際にタイムアウトになる確率で、共に文献 [10] から得られる。 N_l はコネクション C_{k-1} で packet 廃棄が発生してから次に packet 廃棄が起こるまでの時間であるから、

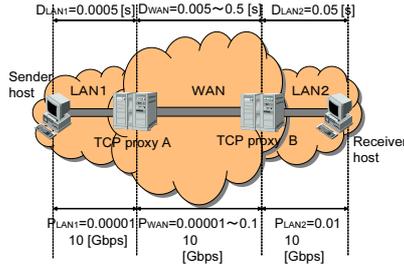


図6 シミュレーションモデル

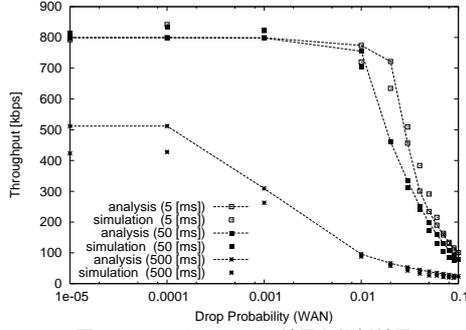


図7 シミュレーション結果と解析結果

$$N_i = \frac{1}{p_{k-1}}$$

となる。最後に、 T_i について考える。再送パケットの到着後は、 $k-1$ 番目のコネクションのスループットあるいは k 番目のコネクションのパケット転送レートの小さい方のレートでパケットを送信することが可能であると仮定すると、

$$T_i = \frac{1}{\min(\rho_{k-1}[h], s_k)}$$

となる。さらに、コネクション C_k の平均スループットは TCP コネクション分割機構 $H(M(k))$ の受信バッファサイズ、経由するリンクの帯域遅延積で制限される可能性があるため、 $K = RB(k) + SB(k)$ 、 $\rho'' = \rho_e[h-1]/s_{k+1}$ とすると、以下のように計算できる。

$$RB_r(k) = \min\left(RB(k), K - \frac{\rho'' \{1 - (K+1)\rho''^K + K\rho''^{K+1}\}}{(1-\rho)(1-\rho''^{K+1})}\right)$$

$$\rho_k[h] = \min\left(\rho_{k-1}[h], \rho_k'''[h], \min_{M(k-1)+1 \leq i \leq M(k)} b(i), \frac{RB_r(k)}{RTT_k}\right)$$

$$\rho_e[h] = \rho_{m+1}[h]$$

3.3 数値例

本節では、前節で示したスループット解析の妥当性を、シミュレーション結果と比較することによって示す。シミュレーションでは図6に示すネットワークを用い、 D_{LAN1} 、 D_{WAN} 、 D_{LAN2} を各リンクの伝播遅延時間、 P_{LAN1} 、 P_{WAN} 、 P_{LAN2} を各リンクのパケット廃棄率とする。また、解析においては(3)式の ϵ の値を 0.01 に設定した。以下の数値例では、ユーザ(受信側ホスト)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として、TCP プロキシ B-受信側ホスト間の廃棄率、伝播遅延時間が大きい場合を想定し、WAN のネットワーク状況を変化させた場合の、コネクション分割によるエンドホスト間のスループットの解析およびシミュレーション結果を示す。

図7は、横軸に WAN の廃棄率、縦軸にエンドホスト間のスループットを示したグラフであり、WAN の伝播遅延時間を 5、50、500 [ms] とした場合の結果を示している。WAN の伝播遅延時間が 5、50 [ms] の場合には、WAN の廃棄率が 0.01 より小さい領域では、分割することによって WAN の廃棄率の増加によるスループットへの悪影響を排除することができ、LAN2 の部分のコネクションのスループットが最小になるため、エンドホスト間の

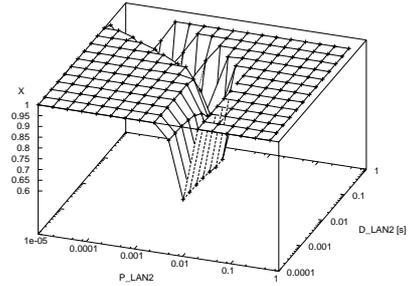


図8 中継処理によるスループット低下の割合

スループットが低下しない。これに対して WAN の廃棄率が 0.01 より大きい領域では、分割したときにスループットが最小となるのが WAN の部分になる。このとき、WAN のパケット廃棄率が大きい場合にスループットが小さくなるため、この部分を境にエンドホスト間のスループットが減少する。WAN の伝播遅延時間を 500 [ms] に設定した場合についても、WAN の廃棄率が 0.0001 を境として同様のことがいえる。

これらの図から、解析結果がシミュレーション結果に適合しない部分もあるが、ほぼ同じ傾向を示しているといえる。特に WAN の伝播遅延時間が小さいときに、解析結果はシミュレーション結果と比較して適合しない傾向にあるが、これは本稿で示した解析が原因ではなく、文献[10]で用いられている仮定に問題があるためである。文献[10]では、パケットロスの検出が重複 ACK のみによる場合には、廃棄率が小さい領域でのみ正しく解析できることが指摘されている。また、文献[10]では、スロースタートを考慮していないため、特に WAN の廃棄率が大きい領域では解析結果が不正確になる。

4. TCP コネクション分割機構の性能評価

本章では 3 章で示した解析結果を用いて、TCP コネクション分割機構の性能に関して考察を行い、本機構を用いる場合の効果を議論する。まず数値例をもとに、中継処理の問題によるスループット低下がどのような状況で発生し、その際にどの程度スループットが低下するのかを示す。また、スループット低下の簡単な回避策について説明し、その有効性を示す。さらに、TCP コネクション分割機構の配置場所について、数値例をもとに議論する。

図8は図6の TCP プロキシ B に着目して、WAN のパケット廃棄率 ($P_{WAN} = 0.01$)、伝播遅延時間 ($D_{WAN} = 0.05$ [s]) を固定して LAN2 のパケット廃棄率、伝播遅延時間を変化させた場合の、図4(a)に示した問題点が原因で劣化するスループットの割合 (X) を示したものである。この図より、TCP プロキシ B の前後のコネクションのネットワーク状況の差異が少ないとき、つまり $P_{LAN2} = 0.01$ 、 $D_{LAN2} = 0.05$ [s] 付近でスループット低下が大きくなる傾向があることが分かる。また、この例においては、理想的なスループットに対して最大で約 40% のスループット低下が発生する。ネットワーク状況の差異が少ない場合には、送信バッファにバッファリングされているパケット数が平均的に少ないため、スループット低下が大きくなる。

TCP コネクション分割機構の中継処理によるスループット低下は、2 章で示したように送信バッファが空になることが原因であるため、送信バッファを大きくすることでスループット低下が回避できると考えられる。図9は、図8においてスループット劣化がもっとも大きい状況 ($P_{WAN} = P_{LAN2} = 0.01$ 、 $D_{WAN} = D_{LAN2} = 0.05$ [s]) を想定して、TCP プロキシ B の送信バッファサイズを変化させたときの、スループット低下の割合を示したものである。スループット低下の割合は、TCP コネクション分割機構の中継処理の問題を考慮しない場合のエンドホスト間の平均スループットに対する、中継処理の問題を考慮した場合のエンドホスト間の平均スループットの割合とする。この図より、送信バッファサイズを大きくすることにより、スループット低下が抑えられることが分かる。これは送信バッファを大きくす

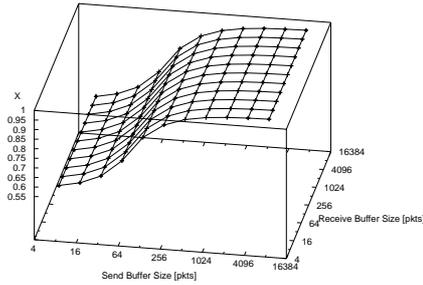


図9 TCPコネクション分割機構の送受信バッファサイズを変化させた場合のスループット低下の割合

ることによって、送信バッファが空になる確率が小さくなるためである。この例においては、スループット低下が発生しない場合にはエンドホスト間のスループットが約 800 [kbps] となる。解析ではパケットサイズを 1 [Kbytes] としており、LAN2 の RTT は 0.1 [s] であることから、送信バッファサイズとして 10 パケットあればこのスループットを達成することができる。多くの OS では送信バッファとして 64–128 [Kbytes] が割り当てられるため、この環境において 800 [kbps] のスループットを達成するには十分である。しかしながら図 9 より、TCP コネクション分割機構を用いて、十分なスループットの向上を得るためには、約 512 [Kbytes] 以上の送信バッファを割り当てる必要があることが分かる。一方、受信バッファサイズに関しては、スループット低下への影響はほとんどないことが図 9 から分かる。ただし、受信バッファサイズが小さい場合には、それ自身が TCP コネクションのスループットを低下させる原因となるため、受信バッファサイズとしてはネットワークの帯域遅延積を目安に設定することが望ましい。

最後に、本稿の解析結果を用いて、TCP コネクションを分割する適切な場所に関する検討を行う。表 1 は図 10 においてネットワーク状況が均一である場合 ($P_{LAN1} = P_{WAN1} = P_{WAN2} = P_{LAN2} = 0.00001$, $D_{LAN1} = D_{WAN1} = D_{WAN2} = D_{LAN2} = 0.005$ [s]) に、TCP プロキシ A~C のいずれか 1 箇所において、コネクション分割を 1 回行った場合のエンドホスト間の平均スループットを示している。この表から、ネットワークの状況に偏りがない場合には、RTT が等しくなるように、つまり TCP プロキシ B においてコネクション分割を行う場合に最もスループットが大きくなることが分かる。これに対して表 2 は、ネットワーク状況が不均一である場合 ($P_{LAN1} = P_{WAN1} = P_{WAN2} = 0.00001$, $P_{LAN2} = 0.01$, $D_{LAN1} = D_{WAN1} = D_{WAN2} = D_{LAN2} = 0.005$ [s]) の平均スループットを示している。表 2 より、RTT が等しくなるように TCP プロキシ B でコネクションを分割しても最適とはならないことが分かる。これはエンドホスト間のスループットが LAN2 を含む受信側ホストに近いコネクションのスループットに大きな影響を受けるため、このコネクションのパケット廃棄率、TCP プロキシでのバッファサイズやその制御方法、および RTT をより小さくするように、TCP プロキシ C でコネクション分割を行うことが最適になるためである。これらの結果から、最適な分割場所を決定するためには、各中継 TCP コネクションの RTT のみを考慮するのではなく、パケット廃棄率や本稿で指摘したスループット低下などを考慮する必要があることが明らかになった。

5. まとめと今後の課題

本稿では、ネットワーク内の TCP コネクション分割機構において、エンドホスト間に設定されたコネクションを分割することによって生じる問題を指摘し、その問題を考慮したスループット解析を行い、シミュレーション結果と比較することにより、解析の妥当性を検証した。TCP コネクション分割機構を用いた場合には、この前後のコネクションのネットワーク状況の差異が少ない場合に、中継処理の問題によるスループット低下が発生するが、送信バッファを大きくすることでこの低下を抑えることができる

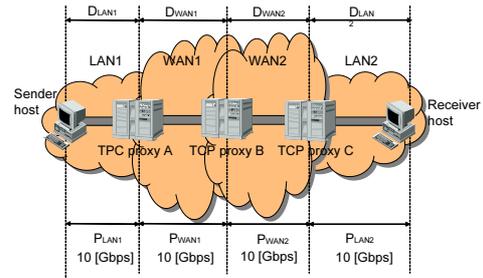


図 10 シミュレーションモデル 2

表 1 分割地点の違いによるエンドホスト間の平均スループット (ネットワークが均一な場合)

分割地点	エンドホスト間の平均スループット [Mbps]
A	59.494
B	108.53
C	59.494

表 2 分割地点の違いによるエンドホスト間の平均スループット (ネットワークが不均一な場合)

分割地点	エンドホスト間の平均スループット [Mbps]
A	1.860
B	2.299
C	3.008

ことを示した。また解析結果から、RTT のみではなくパケット廃棄や本稿で指摘したスループット低下を考慮して、最適な分割場所を決定する必要があることが分かった。今後は、TCP コネクション分割機構の処理負荷や複数の TCP コネクションの存在を考慮した性能評価を行う予定である。

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。

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] 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.
- [4] 村瀬 勉, 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.
- [5] 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.
- [6] 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.
- [7] R. Cohen and S. Ramanathan, "Using proxies to enhance TCP over hybrid fiber coaxial broadband access networks," *Computer Communications*, vol. 20, Jan. 1998.
- [8] P. Rodriguez, S. Sibal, and O. Spatscheck, "TPOT: translucent proxying of TCP," *Computer Communications*, vol. 24, no. 2, pp. 249–255, 2001.
- [9] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," *RFC 3135*, June 2001.
- [10] 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.
- [11] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Massachusetts: Addison-Wesley, 1995.
- [12] 山崎 康広, 村瀬 勉, 長谷川 剛, 村田 正幸, "TCP 中継における輻輳伝達制御方法と性能評価," 電子情報通信学会通信ソサイエティ大会報告 (IN03-136), Dec. 2003.