Vol.2010-IOT-8 No.30
2010/3/2

TCP over TCP の通信効率向上のための動作特性分析

中 河 清 博[†1]　中 村 康 弘[†1]

近年，ネットワーク通信の安全性，利便性の向上を目的として，トンネリング通信が広く利用されている．しかしながら，TCP プロトコルのペイロードに TCP パケットをカプセル化する TCP over TCP のトンネリング通信では，上下層の TCP の相互関係や干渉条件により通信効率の低下が生じる．本研究では，効率低下の原因と考えられる上下層 TCP 間の通信バッファのサイズの相関およびエラー発生時の再送要求タイミングの相関がアプリケーション層のスループットに与える影響を調べ，通信を効率的に行うための条件を求めることを目的とする．

Operating Characteristic Analysis for Efficiency Improvement in TCP over TCP Connection

Kiyohiro Nakagawa[†1] and Yasuhiro Nakamura[†1]

These days, tunneling connection is widely used in computer networks for the purpose of improving security and convenience. TCP over TCP tunneling is implemented through encapsulation of TCP packets into another TCP protocol payload. The disadvantage of the above approach is that the correlation and interference between the upper and lower TCP layers cause a slowdown on the throughput. In order to reduce the impact of the throughput lag, we study the correlation of the buffer size and the retransmission request timing between the upper and lower TCP on throughput.

†1 防衛大学校
National Defense Academy of Japan

1. はじめに

近年，ネットワーク接続機器へのリモート接続やネットワークアクセス型大容量記憶媒体といった組織内 LAN に存在する資源に，外部ネットワーク環境から接続する機会が増加している．また，広域にわたり拠点を保有する大規模組織などにおいては，拠点間のネットワークの確保などが問題となる．このようなネットワークの接続には，専用回線の設置が望ましいが，設置・維持には多大なコストがかかる．一方，近年はモバイル通信環境を含め，ネットワークインフラが充実しており，これらのサービスを利用することでコストの節減が可能になる．しかしながら，このような外部の共有ネットワークサービスの利用は，セキュリティ面などで問題もある．そこで，このような接続には，VPN（Virtual Private Network）などが用いられることが多い．VPN はトンネリング技術を利用して構築され，共有ネットワーク上に仮想的なプライベートネットワークを構築することができる．

トンネリングは，図 1 に示すとおり，トンネリング機能を実装したルータなどを用いて，特定の 2 点間にあらかじめ「トンネル」を構築しておき，上位レイヤーのプロトコルをトンネルの構築に利用しているプロトコルのペイロードとしてカプセル化し，通信を行うことにより実装される（図 2）．

トンネルの構築にセキュアなプロトコルを用いることで，本来セキュアでないアプリケーションのデータをセキュアな状態で伝送することが可能となる．

このトンネルの実装に，TCP(Transmission Control Protocol) を用いて実装する「TCP トンネル」があり，VTun[1)], SSH[2)], PacketiX VPN 2.0[3)] などのトンネリングアプリケーションで利用されている．

一方，TCP トンネルを用いた通信では，ある条件下でエンド間の TCP の性能が劣化することが指摘されている．これは，TCP はエラー検出や再送機能を持つプロトコルであり[4)]，この信頼性のある TCP プロトコルのペイロード部に，さらに TCP パケットをカプセル化することにより，信頼性を維持しようとする機能が各レイヤーで別々に行われ，結果として輻輳状態が継続することが知られている[6)7)]．

輻輳を防止するためには，エラー発生率の高いネットワーク環境下においてもスループットが低下しないようなトンネリング通信の条件について検討する必要がある．

そこで，本稿では，第 2 章で TCP over TCP の通信効率に関する関連研究を紹介する．続いて，第 3 章では通信効率に影響があると考えられる，アプリケーションが送出するデータブロックサイズと通信効率について，第 4 章では，RTO(Retransmisson Time Out) と

Vol.2010-IOT-8 No.30
2010/3/2

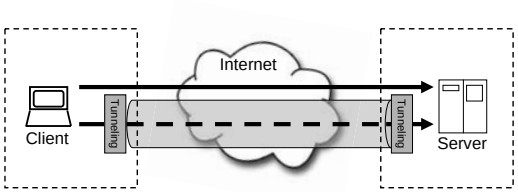


図 1 トンネリング通信

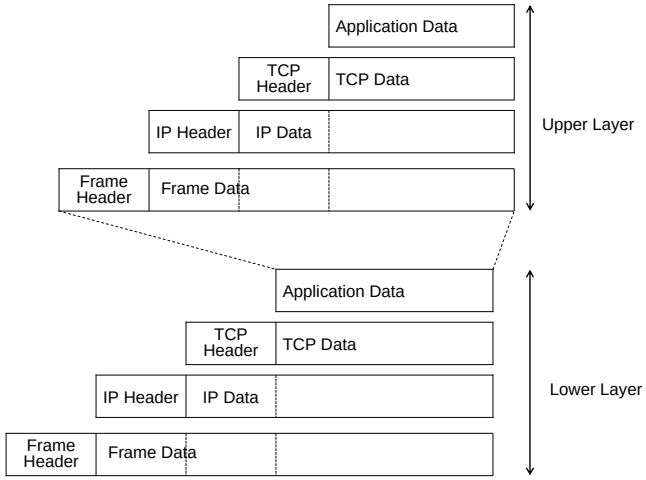


図 2 カプセル化

通信効率についてそれぞれ検証及び考察を行い，最後に第 5 章でまとめと今後の課題について述べる．

2. トンネリングによる通信効率の低下

TCP の性能評価や TCP トンネルを用いた場合のエンド間の TCP 性能に関する研究は

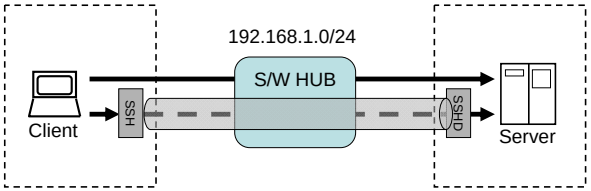


図 3 実験環境

幾つか行われている．

文献 7) では，TCP over TCP 環境において，エンド端末から TCP トンネルまでの伝搬遅延と TCP トンネルの伝搬遅延の大小に着目した評価を行っており，TCP トンネルの伝搬遅延がエンド端末から TCP トンネルまでの伝搬遅延より大きいネットワークでは，グッドプットが大幅に低下することや，トンネル TCP のソケットバッファサイズの影響などについて述べられている．しかしながら，システムパラメータや TCP パラメータについての評価は行われていない．

また，文献[8] では，暗号通信に対するトラフィックフロー分析への対策を検討し，トラフィックフロー分析に耐性のあるトンネリング手法として，パケット長と送信間隔を隠ぺいするトンネリング手法提案している．その中で，アプリケーションが送出するデータブロックのサイズがスループットに与える影響が示唆されており，特にブロックサイズが 500byte から 1100byte の間では，不規則的にスループットが変動することが示されている．しかしながら，アプリケーションが送出するブロックサイズとの関係や要因については具体的な検討がされておらず，通信高率を向上させるためには，この関係を明確にする必要がある．

3. データブロックサイズと転送効率

TCP over TCP による通信効率の低下を防ぐためには，データブロックサイズが通信効率に与える影響ついて調べる必要がある．そこで，データブロックサイズを変化させた際のスループットの変化を測定し，TCP over TCP を考慮した最適なブロックサイズについて検討し，効率的な通信パラメータを導出する．

3.1 実験環境及び手法

実ネットワーク上に，サーバ（受信用）及びクライアント（送信用）端末を設置し，TCP ソケットを経由してデータの送受信する（図 3）． TCP トンネルの実装には，SSH のポート転送機能を利用し，比較のため，SSH トンネルを経由しない通信（直接通信）についても同様の転送実験を行った．

表 1 データブロックサイズ実験におけるパラメータ

	Block Size[byte]	Step[byte]	Total Size[GB]
(1)	10 − 1500	10	2
(2)	10 − 600	10	4
(3)	1 − 25	1	2
(4)	500 − 600	1	4

直 接 通 信

(1) サーバ及びクライアントにしてそれぞれ TCP ソケットを生成し，一定サイズのデータを送受信するプログラムを実行する．

(2) サーバに待ち受けポートを確保しクライアントからの TCP 接続の待ち受けを行う．

(3) クライアントはサーバの待ち受けポートに対し直接接続し，システムコール「write()」を呼び出し，ソケットにデータの書き込みを行う．

(4) write() のでの書き込みサイズをを変化させることで，アプリケーションの送出するデータブロックサイズの変化をシミュレートする．

(5) 一定サイズのデータを送受信するのに要する時間を測定する．この時，測定する所要時間は，1 回目の書き込み直前から，最後の書き込み直後までとする．

SSH トンネリング通信

(1) 直接通信と同様 TCP ソケットを生成しデータの送受信を行う．

(2) あらかじめ，SSH のポートフォーワーディングを利用し，クライアントの特定ポートからサーバの特定ポートにフォーワーディングの設定を行う．

(3) クライアントは，転送元ポート対しデータの書き込みを行う．

実験に用いたパラメータを表 1 に示す．

3.2 実 験 結 果

表 1(1)，(3) 及び (4) の測定結果を，それぞれ図 4 から図 6 に示す．図中の横軸は書き込みサイズ，縦軸は所要時間を表す．これらの測定値から以下の様な結果が得られた．

(1) ブロックサイズが極端に小さい場合（15byte 未満）はブロックサイズが転送効率に大きく影響を与えるが，それ以上では全体に与える影響は少なくなる．

(2) ブロックサイズが 17byte から 290byte の間では，トンネルを使用した方が，それ以外では直接接続の方が効率が良い．

(3) 直接接続で転送を行った場合は，明確な 4byte 周期でに所要時間がお幅に減少したのに対し，トンネリング通信では明確な周期性はみられなかった．

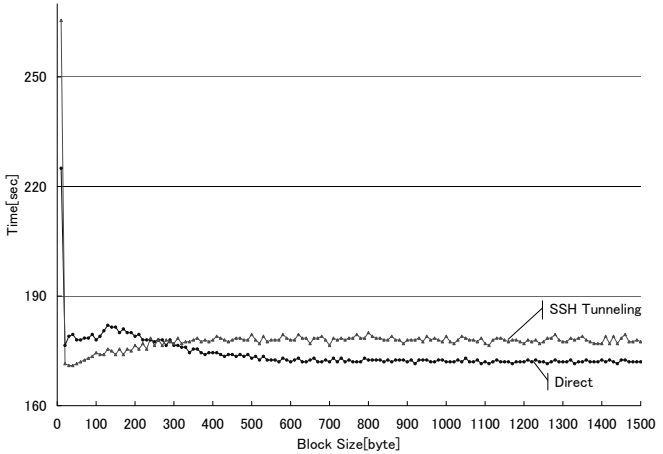


図 4 表 1(1) の測定結果

(4) ブロックサイズが小さい場合は，方式によらず転送効率が悪い．ブロックサイズ 50byte の場合は，10byte の場合に比べ，SSH トンネリング環境下では約 60%，直接接続では 75%効率が良くなる．

3.3 考　　察

直接通信の環境では，データブロックサイズが小さい場合，送出されるパケットのペイロードが小さくなり，パケット数が増加し，転送に要する時間が増える．逆に，ブロックサイズが大きくなると，送出されるパケットのペイロードがパケットサイズの上限程度まで増加し，通信効率は向上するが，効率の向上率は収束する．

また，ブロックサイズが大きい場合，スループットは 4byte ごとに周期的に変化することが確認できた．これは，メモリアライメントやアプリケーションが送出するデータブロックサイズ及びタイミングと，パケットがネットワーク上に送出されるタイミングに影響していると考えられる．

一方，SSH トンネリング環境下では，直接接続時に観測された通信効率の周期的な変化は見られなかったものの，全体的にはブロックサイズが大きいほど効率が良くなることが確認できた．

これは，SSH を用いた場合，上位 TCP のデータは，下位 TCP のペイロードとして暗号化処理されて送出されるためであると考えられる．

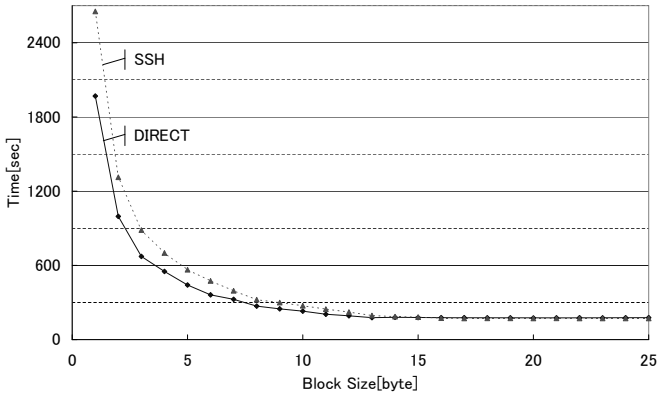


図 5 表 1(3) の測定結果

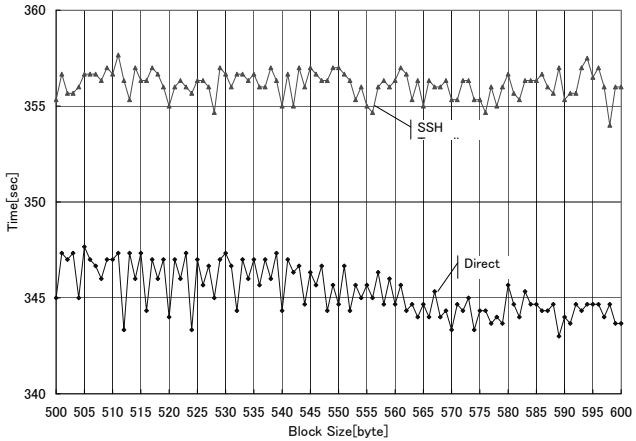


図 6 表 1(4) の測定結果

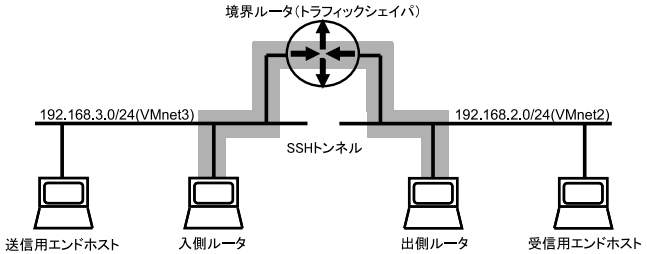


図 7 実験環境

表 2 パラメータ

エンド RTO[sec]	OS 既定値
トンネル RTO[sec]	OS 規定値，1 - 10(間隔：1，各値同一)
パケット損失率 [%]	0 - 0.1(間隔:0.01)

4. 下位 TCP の RTO と転送効率

次に，通信経路上にて一定の通信エラーが発生する状況で，上下層 TCP の RTO に差がある場合の上位 TCP のスループットを測定する．

4.1 実験環境及び手法

実験は，VMware Workstation[10] 上に構築した仮想ネットワークを用いて行った（図 7）．図 7 中のトンネル入口側ルータと出口側ルータ間に，SSH のポート転送機能を利用した TCP トンネルを構築し，送信用エンドホストから，受信用エンドホストに，データブロックサイズ 640byte で 2GB のデータを転送し，送受信完了までに要する時間を測定した．

各エンドホストの RTO 値は OS(Solaris 10) の初期設定値を使用し，下位レイヤーの RTO 値を，カーネルパラメータ（tcp_rexmit_interval_[init | min | max]）により任意の値に設定し，エラー発生率と RTO の関係がスループットに与える影響を調べた．

なお，Solaris10 の RTO は，初期値=3sec, 最小値=400msec，最大値=60sec に設定されており，算出には，Jacobson のアルゴリズム[9] が適用されている．また，トラフィックエラーのエミュレーションには，Dummynet のトラフィックシェイパを利用し，境界ルータ上でパケットロスを発生させた．

本実験で用いたパラメータを表 2 に示す．

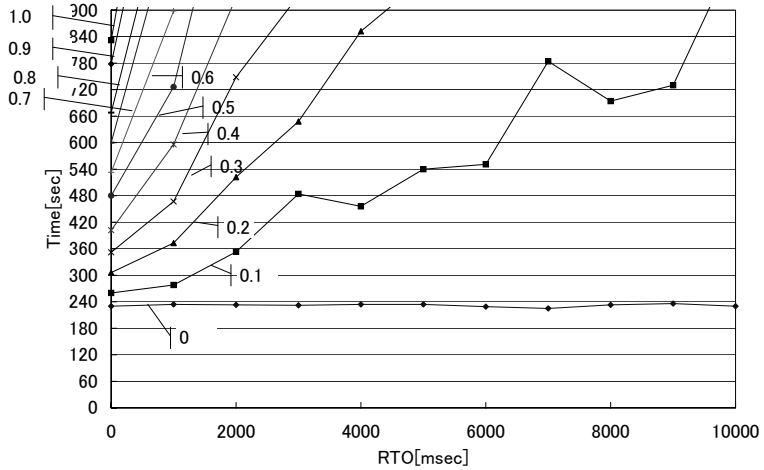


図 8 下位 TCP の RTO と所要時間

4.2 実験結果

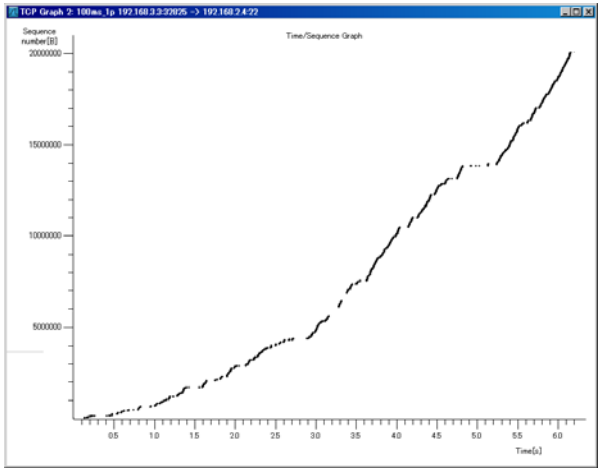
各パケット損失率における RTO と転送完了までに要する時間（上限：900sec）を図 8 に示す．横軸はトンネル TCP の RTO，縦軸は所要時間を示し，RTO=0 は OS 規定値の状態を示す．

パケット損失率が同じ時，トンネル TCP の RTO が長いほど転送に要する時間は長くなる．パケットロス率が 0.1[%] 程度のとき，転送時間を 2 倍以内にするには，下位 TCP の RTO は 5sec 以上に大きくすることはできないことがわかった．

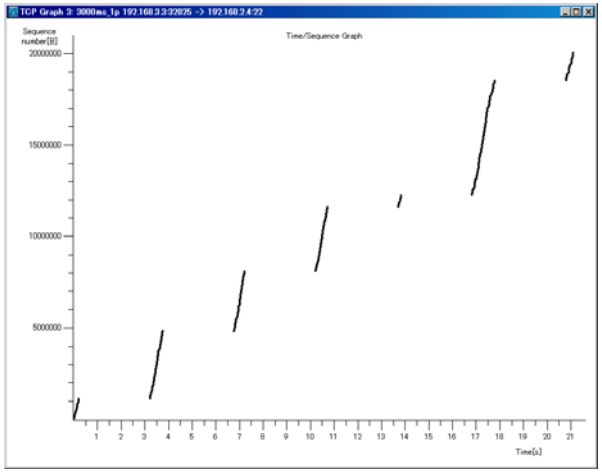
4.3 考　　察

下位 TCP の RTO が全体の通信効率に与える影響を調べた結果，下位 TCP の RTO はエンドホストの RTO より短い方が効率が良くなることが確認できた．エラー発生率 0.1%では，トンネルの TCP の RTO が 1sec の場合に比べ，10sec の場合では，約 8 倍スループットが低下することが分かった．

また，補足実験として，同一環境下で，RTO を 100msec 及び 3000msec に固定し，20MB のデータを転送した際のシーケンス番号の推移を測定した（図 9）．エラー発生のみでは，通信効率が低下するものの，継続的な輻輳状態には至らないことがわかった．



(a) RTO=100 msec



(b) RTO=3000 msec

図 9 シーケンス番号の推移

5. まとめと今後の課題

本研究では，TCP over TCP 通信における通信効率の低下を防止するため，効率低下の要因として考えられるアプリケーションが送出するデータブロックサイズと RTO に着目し，それぞれが通信効率に与える影響について具体的に検証した．

アプリケーションが送出するデータブロックサイズに関する実験では，データブロックサイズが 20byte の場合は，1byte の場合と比較し，スループットは約 93.6%向上したが，400byte と 20byte では，約 2.3%しか改善されななかった．これは，データブロックの書き込みよりも，カプセル化を 2 重に行うため，オーバーヘッドが必要となり，ネットワーク上に送出される 1 パケットあたりのアプリケーションデータ量が減少するためであると考えられる．

トンネル TCP の RTO に関する実験では，下位 TCP の RTO を上位 TCP の RTO に比べ小さく設定することで，通信エラー発生時の再送に伴う輻輳状態を軽減出来ることが確認できた．パケットロス率 0.1[%] では，OS 規定の RTO 初期値である 3[sec] の場合と比較し，1sec では，スループットは約 57%向上し，下位 TCP の RTO を短くすることが有効であると考えられる．

今後の課題として，通信効率の低下を極限できる条件について更に細かく検討する必要がある．具体的には，ペイロードの最大値に近いサイズのパケットを送出するためのアプリケーション送出データブロックサイズの検討，および，上位 TCP の RTO に対する下位 TCP の RTO の相互関係の検討を行う必要がある．RTO に関しては，特殊な設定をしない場合，上下レイヤーはそれぞれ，規定のアルゴリズムに従い，通信状況に応じた最適 RTO を算出し新たな RTO として更新する仕組みになっている．しかしながら，現状の方法では，TCP over TCP の通信環境下において，ネットワーク上でエラーが頻繁に発生した場合や遅延が大きい場合には輻輳状態が長く続く可能性が高い．従って，エラーの頻発や大きな遅延が発生しても，状況に応じ TCP パラメータを調整するアルゴリズムを考慮することが有効であると考える．

参 考 文 献

1) VTun - virtual tunnels over TCP/IP networks, http://vtun.sourseforge.net.
2) OpenSSH, http://www.openssh.org/.
3) PacketiX VPN 2.0, http://www.softether.co.jp/jp/vpn2/.
4) J.Postel：Transmission Control Protocol, RFC 793 (1981).
5) J.Postel：Computing TCP's Retransmission Timer, RFC 2988 (2000).
6) Olaf Titz：Why TCP Over TCP Is A Bad Idea, http://sites.inka.de/bigred/devel/tcp-tcp.html.
7) 本田 治, 大崎 博之, 今瀬 真, 石塚 美加, 村山 純一，TCP over TCP の性能評価－TCP トンネルがエンド-エンドのスループットおよび遅延に与える影響－, 電子情報通信学会技術研究報告, Vol.104 No.438，pp.79-84(2004).
8) 三村 守, 中村 康弘：トラフィックフロー分析に耐性があるトンネリング手法の検討, コンピュータセキュリティシンポジウム 2007 (CSS2007) 論文集, pp.325-330 (2007).
9) Van Jacobson, Michael J. Karels：Congestion Avoidance and Control(1988)
10) VMware Workstation, http://www.vmware.com/jp/products/ws/