

衛星インターネット環境に適した TCP-STAR の実装及び評価

小畑 博靖^{†a)} 平 和弘^{††} 石田 賢治[†]

Implementation and Evaluation of TCP-STAR: TCP Congestion Control Method for Satellite Internet

Hiroyasu OBATA^{†a)}, Kazuhiro TAIRA^{††}, and Kenji ISHIDA[†]

あらまし 衛星回線を用いた衛星インターネットサービスは、近年各地で多発する大規模災害に強く、安定した高速通信サービスを提供できるため注目されている。しかし、広帯域・高遅延といった特性をもつ衛星回線において、従来の TCP (例えば NewReno) を用いると十分なスループットが得られないことが知られている。このような衛星インターネットに適した TCP ふくそう制御方式として TCP-STAR が提案されている。TCP-STAR では衛星インターネット上において、従来 TCP と比べて高いスループットを得られることがシミュレーションにより確認されている。しかし、実ネットワークではシミュレーションのような理想的な環境で通信を行えることは少なく、実装による性能評価が不可欠である。そこで本論文では、まず実ネットワークにおける TCP-STAR の評価を行うため、NetBSD1.6.2-Release のカーネル内に TCP-STAR を実装した。次に、ハードウェアネットワークエミュレータを用いて衛星回線を考慮したテストベッドを構築し、このテストベッド上で既存 TCP と TCP-STAR とをスループットに関して比較評価した。その結果、実装した TCP-STAR は一つの接続のみ存在し最大性能が得られる環境において、従来の TCP (TCP-NewReno)、無線ネットワークを考慮した TCP (TCP-J)、及び有線の Long Fat Network (LFN) を考慮した TCP (HighSpeed TCP, Scalable TCP) と比べて高いスループットが得られ、衛星回線において TCP-STAR の機能が効果的に動作することを確認した。また、TCP-STAR のみ存在する環境において、TCP-STAR は公平性が高いことも確認した。

キーワード 衛星インターネット, TCP-STAR の実装, ふくそう制御

1. ま え が き

無線アクセス技術の進歩に伴い、無線 LAN ネットワークや衛星回線を用いた衛星ネットワーク等の様々な無線通信サービスに注目が集まっている。特に、衛星回線を用いた衛星インターネットサービスは、近年多発する大規模災害に強く安定した高速通信サービスを提供できるため注目されている。また、衛星回線は有線ネットワークと同様に広帯域化が進められている。例えば、広帯域衛星通信システムとして、現在研究開発が行われている WINDS (Wideband InterNetworking

engineering test and Demonstration Satellite) システム [1] がある。WINDS システムでは、上り・下り回線ともに衛星回線を用い、最大 1.2 Gbit/s の超高速通信が可能となる予定である。更に、航空機や船舶等、ネットワークの施設が不可能または困難な場所におけるインターネットサービスの需要も増加している。このような場所においてサービスを提供するために、Connexion by Boeing システム [2] 等の衛星インターネットが利用される。

衛星インターネットでは、WWW (World Wide Web) や FTP (File Transfer Protocol) 等の代表的なアプリケーションがよく利用される。これらのアプリケーションの多くが TCP (Transmission Control Protocol) を利用する。しかし、従来の TCP (例えば、NewReno) は、もともと有線ネットワーク向けに開発されており、無線ネットワークでの利用を考慮していないため、衛星回線の性質により大きくスループットが低下する [3], [4]。このような問題を解決するために、

[†] 広島市立大学情報科学部情報工学科, 広島市
Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University, 3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima-shi, 731-3194 Japan

^{††} 広島市立大学院情報科学研究科情報工学専攻, 広島市
Graduate School of Information Sciences, Hiroshima City University, 3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima-shi, 731-3194 Japan

a) E-mail: obata@ce.hiroshima-cu.ac.jp

衛星インターネットに適したふくそう制御方式として TCP-STAR が提案されている [5] ~ [7] . TCP-STAR は衛星インターネットにおいて, 従来の TCP (TCP-NewReno, TCP-Westwood [8], TCP-J [9]) と比べて高いスループットが得られることが ns2 [10] を用いたシミュレーションにより確認されている . しかし, 実ネットワークではシミュレーションのような理想的な環境で通信を行えることは少なく, 実装による性能評価は不可欠 [11] である .

そこで本論文では, まず実ネットワークにおける TCP-STAR の評価を行うため, NetBSD1.6.2-Release のカーネル内に TCP-STAR を実装した [12] . 次に, ハードウェアネットワークエミュレータ [13] を用いて衛星回線を考慮したテストベッドを構築し, このテストベッド上において既存 TCP と TCP-STAR とをスループットに関して比較評価した . その結果, 実装した TCP-STAR は一つの接続のみ存在し最大性能が得られる環境において, 従来の TCP (TCP-NewReno), 無線ネットワークを考慮した TCP (TCP-J), 及び有線の Long Fat Network (LFN) を考慮した TCP (HighSpeed TCP [14], Scalable TCP [15]) と比べて高いスループットが得られ, 衛星回線において TCP-STAR の機能は効果的に動作することを確認した . また, TCP-STAR のみ存在する環境において, TCP-STAR は公平性が高いことも確認した .

以下, 2. では TCP-STAR の概要を, 3. では TCP-STAR の実装について説明する . 4. では, 実装した TCP-STAR の評価を行う . 最後にまとめを 5. で述べる .

2. TCP-STAR の概要

TCP-STAR のふくそう制御方式について説明する . TCP-STAR は, (I) LWC (Lift Window Control), (II) CWS (Congestion Window Setting), (III) AEN (Acknowledgement Error Notification), の三つのふくそう制御機構をもつ [5] ~ [7] . LWC, CWS では帯域見積り機構として ABE (Available Bandwidth Estimation) [9] を用いる . また, TCP-STAR は, 送信側端末の変更のみ必要とする End-to-End 型のふくそう制御方式となっている .

(I) LWC (Lift Window Contorol)

LWC では, TCP-NewReno のふくそうウィンドウサイズ $cwnd_{reno}$ と, 測定帯域 BW から得た測定ウィンドウサイズ $cwnd_{abe}$ の和を用いて図 1 のようにふ

```

if (  $cwnd_{reno} < ssthresh$  )
   $cwnd_{reno} = cwnd_{reno} + 1$ 
   $cwnd = cwnd_{reno} + cwnd_{abe}$ 
else if (  $cwnd_{reno} \geq ssthresh$  )
   $cwnd_{reno} = cwnd_{reno} + \frac{1}{cwnd_{reno}}$ 
   $cwnd = cwnd_{reno} + cwnd_{abe}$ 

```

図 1 LWC (Lift Window Contorol) の手続き
Fig. 1 Procedure of LWC (Lift Window Contorol).

```

if (  $cwnd < ssthresh$  )
   $ssthresh = cwnd_{abe}$ 
   $cwnd \Rightarrow$  keep recent value
else if (  $cwnd \geq ssthresh$  )
   $ssthresh = cwnd_{abe}$ 
   $cwnd = ssthresh$ 

```

図 2 CWS (Congestion Window Setting) の手続き
Fig. 2 Procedure of CWS (Congestion Window Setting).

くそうウィンドウサイズ $cwnd$ を決定する . 図 1 において $ssthresh$ はスロースタートしきい値を示す . これにより, 衛星インターネットのように RTT の大きなネットワークにおいて迅速なデータ転送レートの向上が可能となる . また測定ウィンドウサイズ $cwnd_{abe}$ は, 測定帯域 BW を用いて式 (1) で得られる .

$$cwnd_{abe} = \frac{BW \times RTT_{min}}{MSS} \quad (1)$$

ただし, RTT_{min} は最小ラウンドトリップ時間, MSS は最大セグメントサイズを表す .

また, 帯域は ACK の受信間隔と ACK が通知する受信確認されたデータ量を用いて導出する .

(II) CWS (Congestion Window Setting)

CWS では, データセグメントロス発生後, 測定帯域 BW に基づき, 図 2 のようにスロースタートしきい値 $ssthresh$ とふくそうウィンドウサイズ $cwnd$ を設定する . これにより, ビット誤りによるデータセグメントロス時の転送レート低下を防ぐ .

(III) AEN (Acknowledgement Error Notification)

衛星回線のように RTT が大きいネットワークでは, ACK セグメントのロスや遅延によって再送タイムアウトが発生しやすい . その結果, 誤再送及び不当な転送レート抑制が起こり, スループットが大きく低下する [16] ~ [18] . AEN はこのような ACK セグメントのロス/遅延によるスループットの低下を防止する .

図 3 に AEN のフローチャートを示す . まず, 再送

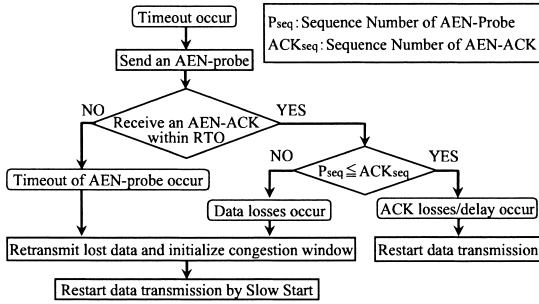


図 3 AEN のフローチャート
Fig. 3 Flow chart of AEN.

タイムアウトが発生すると、送信済みの最新シーケンス番号をもつデータセグメント（以下、AEN-probe と呼ぶ）の再送を行う。次に、再送セグメントに対する ACK セグメント（以下、AEN-ACK と呼ぶ）を用いてデータセグメントのロス、ACK セグメントのロス/遅延の有無の確認を行う。データセグメントのロスと ACK セグメントのロス/遅延の判定は、AEN-probe のシーケンス番号 (P_{seq}) と AEN-ACK のシーケンス番号 (ACK_{seq}) を用いて行う。タイムアウトの原因が ACK セグメントのロス/遅延であれば ($P_{seq} \leq ACK_{seq}$)、タイムアウト前のウィンドウサイズをそのまま用いて新しいデータの送信を再開する。データのロスが原因であれば ($P_{seq} > ACK_{seq}$)、損失したデータセグメントを再送し、スロースタートによりデータ送信を再開する。また、AEN-probe がタイムアウトを起こした場合、データのロスが発生した場合と同様の処理を行う。

3. TCP-STAR の実装

本研究では、TCP-STAR を NetBSD1.6.2-Release のカーネル内に実装した。まず、TCP-STAR は TCP-J の帯域測定方式 ABE を利用しているため、TCP-J の実装 [11] を利用して NetBSD に ABE を追加した。次に、追加した ABE で得られる帯域を用いる LWC と CWS を実装した。更に、図 3 に基づいて AEN の実装を行った。

3.1 TCP-STAR の改良

TCP-STAR の実装を行い通信テストを行った結果、不具合を発見したため、TCP-STAR に以下の変更を加えた。

TCP-STAR では、TCP-J の ABE 機構を利用して、式 (2) のように帯域を見積もっている。ただし、 BW_{sample} は測定帯域、 $ack_interval$ は ACK の受信

間隔、 d は受信した ACK に対応するデータ量をそれぞれ示す。

$$BW_{sample} = \frac{d}{ack_interval} \quad (2)$$

また、 t_k を k 番目の ACK の受信時間とすると、 $ack_interval = t_k - t_{k-1}$ となる。通信開始後、最初の ACK 受信時に行う帯域計算では、前回受け取った ACK がなく $ack_interval$ の計算ができない。そのため、従来の ABE 機構の実装 [11] では、 $ack_interval$ の初期値として 1 マイクロ秒が設定されている。しかし、 $ack_interval$ の値が小さすぎるために、今回の実験環境の実帯域 (10 Mbit/s) を超える非常に大きな値を見積もっていることが分かった。

その結果、TCP-STAR では、帯域値を用いてウィンドウサイズを増加させるため、データ送信開始後すぐに実帯域よりも大きな測定帯域を利用して必要以上にウィンドウサイズを増加させていた。そこで、データ送信開始後、必要以上にウィンドウサイズが大きくなるように、実験に基づき $ack_interval$ の初期値をもとの値よりも比較的大きな値 (1 ミリ秒) に変更した。

3.2 カーネル内設定の変更

TCP-STAR は広帯域、高遅延な衛星インターネットにおいて、高いスループットを実現するために多量のデータを連続して送出する。そのため、カーネル内のネットワークパラメータが広帯域・高遅延なネットワーク用に設定されていないオリジナルの NetBSD に TCP-STAR の機能を加えただけでは TCP-STAR の性能を十分に発揮できないことが分かった。そこで、TCP-STAR を実装した際、TCP-STAR の性能を十分に発揮させるために、TCP-STAR の仕様 [5] に加えて、NetBSD のカーネル内のネットワークパラメータを次のように変更した。

(a) インタフェースレイヤの出力キューサイズ (ifq_len) [19] をデフォルトの 50 パケットから、帯域幅遅延積よりも十分大きな値 (8192 パケット) に変更。

(b) 受信ウィンドウサイズの値をウィンドウスケールオプション [20] により拡張するため、以下のカーネル設定変数

- `kern.sbmax` ... 1 コネクションに割り当てるソケットバッファサイズの最大値
- `net.inet.tcp.recvspace` ... 受信バッファサイズ
- `net.inet.tcp.sendspace` ... 送信バッファサイズ
- `kern.mbuf.nmbclusters` ... TCP コネクション

が送受信データを格納するバッファ (mbuf) の最大数を帯域幅遅延積を考慮した値に変更した。

4. TCP-STAR の実験的評価

本章では、TCP-STAR の実装を用いて性能評価を行った実験結果を示す。まず、実験環境について説明する。次に、TCP-STAR のスループット性能と公平性に関する評価を行う。

4.1 ネットワークパラメータ

図 4 に、実験に用いたテストベッドを示す。本研究では衛星回線の環境を実現するために、ハードウェアネットワークエミュレータ [13] を利用した。送受信ホストの仕様を表 1 に示す。受信側ホストの OS は NetBSD1.6.2-Release とした。また、データセグメントサイズは 1448 Byte, ACK セグメントサイズは 52 Byte とした。表 2 にネットワークエミュレータに設定した衛星回線の特徴づけるネットワークパラメータ [4], [12] を示す。また、送受信ホストの最大送信・受信ウィンドウサイズは想定する衛星回線に適した帯域幅遅延積の値 (625 kByte) に設定した。TCP のトラヒックは Netperf [21] を用いて発生させた。

4.2 単一コネクション時のスループット性能の評価

実装した TCP-STAR と従来 TCP を用いてスループット性能の比較を行った。比較対象として、従来 TCP の TCP-NewReno, 無線ネットワーク用 TCP である TCP-J^(注1), 及び衛星回線と同様に広帯域かつ高遅延な有線ネットワーク (Long Fat Network) を

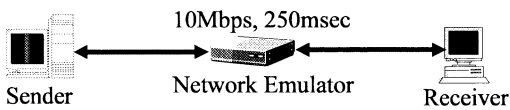


図 4 実験ネットワーク 1 (テストベッド)
Fig. 4 Experimental network 1 (testbed).

表 1 送受信ホストの仕様
Table 1 Machine specification.

	CPU	Memory
送信側	Pen4 3.8 GHz	4 GByte
受信側	Pen4 1.5 GHz	512 MByte

表 2 実験パラメータ
Table 2 Experimental parameters.

帯域	10 Mbit/s
伝搬遅延時間	250 ms
Bit Error Rate	0, 10^{-8} , 10^{-7} , 10^{-6} , 10^{-5}

対象とした TCP (HighSpeed TCP, Scalable TCP) を用いた^(注2)。比較対象はすべて TCP-STAR と同様に End-to-End 型のふくそう制御方式である。また、本論文において、TCP-STAR 及びすべての比較対象の TCP は、SACK とタイムスタンプオプションを利用している。

本節における評価では、最大性能を評価するため、ネットワークに存在するコネクション数は 1 とした。また、各 TCP が動作する OS においても 3.2 で述べたようにネットワークパラメータを変更して最大性能が出るように設定した。実験では、実装した TCP-STAR のふくそう制御機構の動作確認と性能評価を行うため、データセグメントのロスが発生する場合、ACK セグメントのロスが発生する場合、に分けて評価した。

4.2.1 データセグメントロスが発生する場合

図 5 は、ビット誤り率 (BER) を 0 (error free) から 10^{-5} の間で変化させたときの TCP-STAR (STAR), TCP-J (J), HighSpeed TCP (HSTCP), Scalable TCP (Scalable), TCP-NewReno (NewReno) のスループットをそれぞれ示す。ここで、衛星回線における典型的な BER の値は、 10^{-8} から 10^{-6} の間の値となることが知られている [4], [7]。ただし、台風や梅雨末期の豪雨の場合には、BER は 10^{-5} に近い値となる可能性がある。また、実験結果は、5 回実行した平均値を示す。図 5 より、TCP-STAR は常に従来 TCP より高いスループットが得られていることが分かる。例えば図 5 の BER が 10^{-8} のとき、TCP-STAR は TCP-NewReno と比べて約 152%, HighSpeed TCP と比べて約 95.9%, Scalable TCP と比べて約 41.6%, TCP-J と比べて約 13.5% スループットが向上している。これらの性能の向上は、LWC を用いて送信開始時にふくそうウィンドウサイズを急速に増加していること、CWS を用いてデータセグメントロス発生後に

(注1): TCP-J ではシミュレーションによる評価 [9] 及び実装を用いた実験の評価 [11] により、伝搬遅延時間の小さい地上系の無線ネットワークで高いスループット性能が得られることが示されている。本研究で利用した TCP-J の実装 [11] においては、ソースコードを解析した結果 [9] と比較してスロースタートしきい値の最大値と再送タイムアウト後のふくそうウィンドウサイズの設定方法に変更が加えられていることが分かった。本研究では [11] の設定を衛星環境で利用できるように変更した。具体的には (a) スロースタートしきい値の最大値を衛星回線の帯域幅遅延積まで拡張 (b) 図 3 で示した TCP-STAR のデータロスによる再送タイムアウト時の動作と同様に、再送タイムアウト後のふくそうウィンドウサイズを初期値に設定、を行った。

(注2): TCP-NewReno, TCP-J, 及び TCP-STAR は NetBSD 上に実装したものを利用し、HighSpeed TCP と Scalable TCP については Linux 上に実装されているものを利用した。

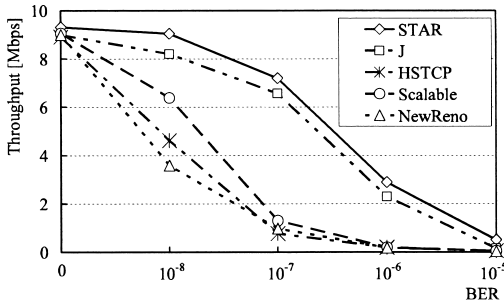


図5 データセグメントロスが発生したときのスループット
Fig. 5 Throughput in case of data losses.

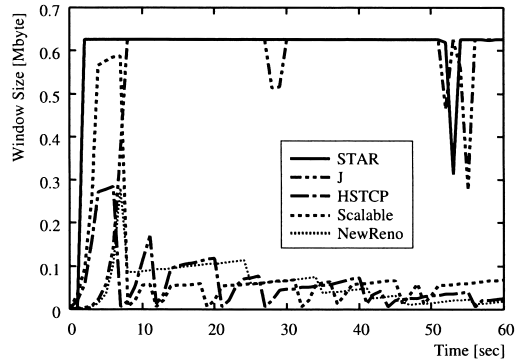


図6 ふくそうウィンドウサイズの様子 (BER = 10⁻⁷)
Fig. 6 Congestion window size (BER = 10⁻⁷).

不要なふくそうウィンドウサイズの低下を防いでいること、により得られている。図6は図5において、BER = 10⁻⁷のときのふくそうウィンドウサイズの変化の様子を示す。図6より、TCP-STARは従来TCPと比べて、常にふくそうウィンドウサイズを高く維持できていることが分かる。その結果、従来TCPよりもTCP-STARのスループットが向上している。また、TCP-STARのふくそうウィンドウサイズを見ると、ほぼ実験ネットワークの帯域幅遅延積に相当する値(625 kByte)に収束していることが分かる。

図5より、BER = 0のスループットを見ると、TCP-STAR以外のTCPはほぼ一致しているが、TCP-STARだけが他のTCPよりもスループットが高い。この差は以下の理由による。図7は、図5におけるBER = 0のときのふくそうウィンドウサイズの様子を示す。図7より、TCP-STARの場合は、送信開始時(0~2sの間)においてLWCを用いてふくそうウィンドウサイズを急激に増加していることが分かる。一方で、TCP-STAR以外のTCPではスロースタートによりふくそうウィンドウサイズをゆっくりと増加している(0~9sの間)。したがって、送信開始時におけるふくそうウィンドウサイズの差がBER = 0のときのスループットの差となっている。

以上より、データセグメントロスが発生するとき、LWC機構とCWS機構が有効に動作し、TCP-STARは従来TCPより高いスループットが得られることを確認した。

4.2.2 ACKセグメントロスが発生する場合

次に、AENの効果を確認するために、ACKセグメントのロスが連続して発生した場合のスループットを評価する。図8は、データ送信開始後45sでACKセグメントが約1秒間バーストロスしたときの、TCP-

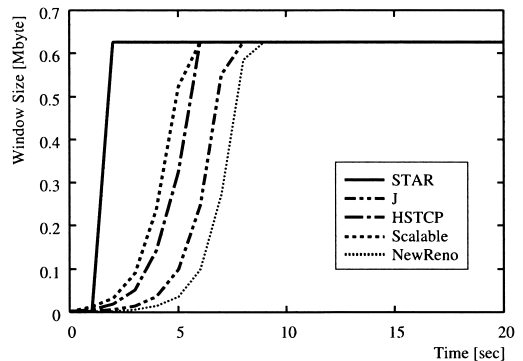


図7 ふくそうウィンドウサイズの様子 (BER = 0)
Fig. 7 Congestion window size (BER = 0).

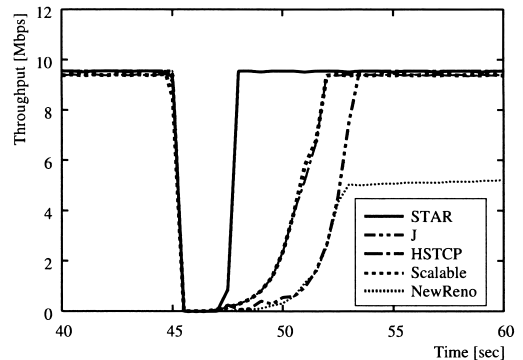


図8 ACKロスが発生した場合のスループット
Fig. 8 Throughput in case of ACK losses.

STAR, TCP-J, HighSpeed TCP, Scalable TCP, 及び TCP-NewReno のスループットをそれぞれ示す。したがって、この実験では、ACKロスによる再送タイムアウトが発生する。図8より、TCP-J, HighSpeed

TCP, Scalable TCP, 及び TCP-NewReno は ACK ロスによる再送タイムアウトを検知することができないため, ACK セグメントのバーストロス発生後にスロースタートによりデータの送信を再開している (47~53s). その結果, スループットの回復に時間がかかっている. 一方で, TCP-STAR は AEN を用いて ACK ロスを検知することができているため, ACK バーストロス発生後すぐにスループットが回復している (47~48s) ことが分かる.

以上より, TCP-STAR の AEN 機構は, 再送タイムアウトの原因を検知することができ, ACK ロスによるスループットの低下を防止できることを確認した.

また, 4.2.1 及び 4.2.2 の結果より, 衛星ネットワーク上において, TCP-STAR は無線ネットワーク向けに提案されている TCP-J, 及び Long Fat Network 向けに提案されている HighSpeed TCP, Scalable TCP よりも高いスループットが得られることを確認した. これは, まず, TCP-J は一般的な無線ネットワークを対象としているのに対して, TCP-STAR は無線ネットワークの一つである衛星ネットワークに着目し, 衛星ネットワークに特化したふくそう制御を行っているためである. また, HighSpeed TCP, Scalable TCP は広帯域かつ高遅延な有線ネットワークを対象としているが, 無線ネットワークにおいて発生するビット誤りによるパケットロスを考慮していないため, TCP-STAR よりも性能が低くなっている.

4.3 複数コネクションが存在するときの評価

本節では, 複数コネクションが存在する場合の実装した TCP-STAR のスループット性能と公平性に関して評価を行う. 図 9 に評価を行うために構築した実験ネットワークを示す. 図 9 の実験ネットワークでは, 送信端末 (Sender 1, Sender 2) と衛星ルータ (Router) 間には有線ネットワークを想定した. 衛星ルータと受信端末 (Receiver 1, Receiver 2) 間は衛星回線を想定しネットワークエミュレータにより表 2 に示した遅延及び BER を設定した. 本実験ネットワークでは, 衛星ルータにおいてふくそうによるパケットロスを発生させるために, ルータのキューサイズを変化させた. 具体的に, 衛星ルータのキューサイズは, パケットロスが頻発する重いふくそう状態 (15 kByte) とパケットロスがほとんど起きない軽いふくそう状態 (625 kByte) となるように設定した^(注3). また, 図 9 の実験ネットワークでは, Sender 1 は Receiver 1 と通信を行い (Flow1), Sender 2 は Receiver 2 と通信

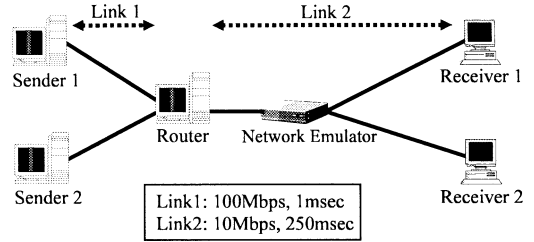


図 9 実験ネットワーク 2 (テストベッド)
Fig. 9 Experimental network 2 (testbed).

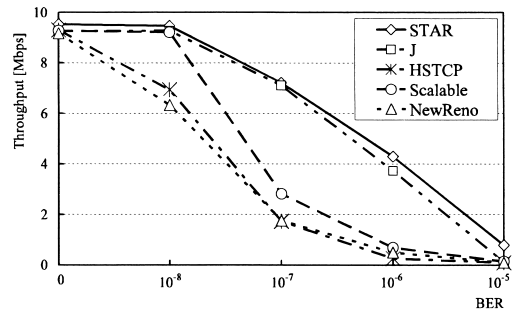


図 10 ふくそうとビット誤りによるパケットロスが発生するときの合計スループット (ルータキューサイズ: 625 kByte)

Fig. 10 Total throughput when packet losses by congestion and bit error occur, where router queue size is 625 kByte.

を行う (Flow2). 実験結果は, 5 回実行した平均値を示す.

4.3.1 複数コネクション時におけるスループット性能の評価

複数コネクションが存在し, ふくそうによるパケットロスとビット誤りによるパケットロスが混在する場合のスループット性能を評価する. 実験では, Sender 1 と Sender 2 はともに同じ TCP のバージョンを利用した. 図 10 は, ルータキューサイズが 625 kByte のとき, 軽いふくそう状態における 2 フロー分の合計スループットを示す. 同様に図 11 は, ルータキューサイズが 15 kByte のとき, 重いふくそう状態における結果を示す.

(注3): 本研究では, 利用可能な設備の都合上, 多数の送受信端末を用いてふくそうを発生させる方法はとらず, 衛星ルータのキューサイズ設定によりふくそうを発生させた. 1 台の PC に複数の TCP コネクションを実行することでふくそうを発生させる方法も考えられるが, 送信端末の通信インタフェースにおける処理がボトルネックとなる可能性がある. そのため, 本研究では 1 台の PC に一つの TCP コネクションを実行し, 衛星ルータのキューサイズ設定によりふくそうを発生させる方法を採用した.

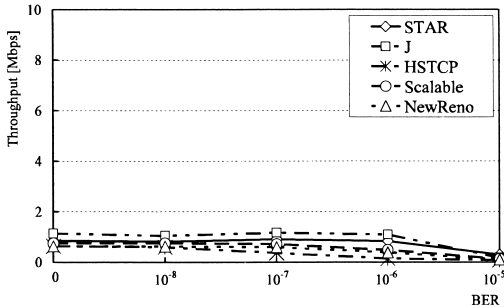


図 11 ふくそうとビット誤りによるパケットロスが発生するときの合計スループット (ルータキューサイズ: 15 kByte)

Fig. 11 Total throughput when packet losses by congestion and bit error occur, where router queue size is 15 kByte.

まず、図 10 より、軽いふくそう状態では、TCP-STAR は他の TCP と比べて最も高い合計スループットが得られ、4.2 の結果と同様に TCP-STAR の各機能が効果的に動作していることが分かる。次に、図 11 より、重いふくそう状態において、TCP-STAR は High-Speed TCP, Scalable TCP, 及び TCP-NewReno よりも高い合計スループットが得られていることが分かる。しかし、 $BER=10^{-5}$ のときを除いて、TCP-STAR は TCP-J よりも合計スループットが若干低い値となっている。これは、重いふくそう状態であるために、データセグメントロスによるタイムアウトが多発し、TCP-STAR の AEN 機能による処理に時間がかかっていることが原因である。2. の (III) で示したように、TCP-STAR はタイムアウトが発生すると AEN-probe を送信し、このプローブパケットに対する確認応答を受信してタイムアウトの原因を判定する。タイムアウト原因の判定後、タイムアウトの原因がデータセグメントロスであれば、ロスセグメントの再送を行う。したがって、データセグメントロスによるタイムアウトが発生した場合、タイムアウト原因の調査に必要な時間が加わるために、タイムアウトが発生してからデータの再送が完了するまでに要する時間は AEN を使わない場合と比べて 2 倍となる。その結果、データセグメントロスによるタイムアウトが非常に多く発生すると、ロスセグメントの再送に時間がかかり AEN 機能がない場合と比べてスループットが低下する。

4.3.2 公平性に関する基礎的考察

実装した TCP-STAR の公平性^(注4)について基礎的な評価を行う。公平性評価における比較対象の改良

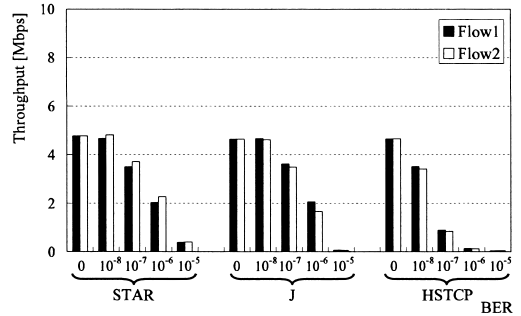


図 12 同種間の公平性の評価 (ルータキューサイズ: 625 kByte)

Fig. 12 Fairness evaluation in homogeneous scenario when router queue size is 625 kByte.

TCP として、TCP-J と HighSpeed TCP^(注5)を利用した。

(a) 同種の TCP のみ存在する場合

はじめに、同種の TCP のみ存在する場合について評価する。

まず、衛星ルータにおいてほとんどふくそうによるパケットロスが発生しない場合の結果を示す。図 12 は、ルータキューサイズを 625 kByte とし、Sender 1 と Sender 2 がともに TCP-STAR, TCP-J (J), High-Speed TCP (HSTCP) のときのそれぞれの端末のスループットを示す。図 12 より、混在する二つの TCP-STAR のフローは、TCP-J 及び HighSpeed TCP と同様に、ほぼ同等のスループットが得られていることが確認できる。ここで、図 13, 図 14 は、TCP-STAR の場合の、 $BER=0$ のときと $BER=10^{-8}$ のとき^(注6)のそれぞれの送信端末のふくそうウィンドウの様子を示す。

図 13 の $BER=0$ の場合、ふくそうによるパケットロスが発生しふくそうウィンドウが多少変動しているが、混在する二つのフローの値はほぼ一致しており、非常に安定していることが分かる。一方、図 14 の $BER=10^{-8}$ の場合、ビット誤りによるパケットロスのためにふくそうウィンドウが大きく変化しているが、ほぼ同様な変動をしていることが確認できる。その結果、図 12 に示したように、どの BER においてもス

(注4): 本研究における公平性の定義は、「すべてのコネクションが等しく利用可能な帯域を分け合っている状態」とした。

(注5): 4.3.1 のスループット評価で比較していた Scalable TCP については、HighSpeed TCP と同様な設計思想で、実験した結果がほぼ同じ値になったため、本文中の記述を省略した。

(注6): $BER=10^{-8}$ は、晴天時における値に相当する [4]。

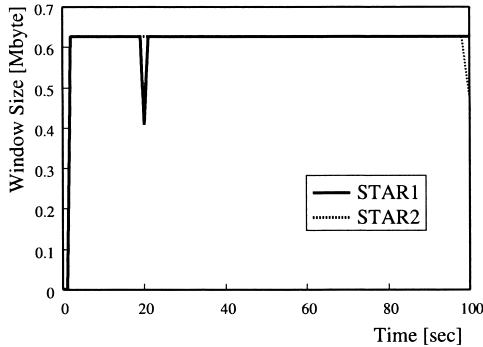


図 13 TCP-STAR のふくそうウィンドウの変化 (BER=0, ルータキューサイズ: 625 kByte)
 Fig. 13 Congestion window size of TCP-STAR (BER=0, router queue size is 625 kByte).

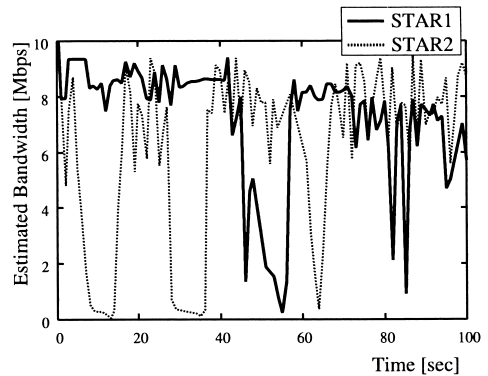


図 15 ABE による見積り帯域の変化 (BER=10⁻⁸, ルータキューサイズ: 625 kByte)
 Fig. 15 Estimated Bandwidth by ABE (BER=10⁻⁸, router queue size is 625 kByte).

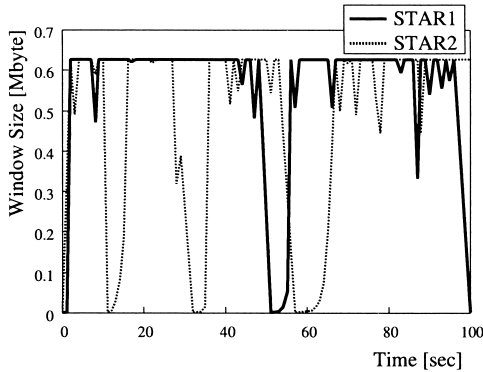


図 14 TCP-STAR のふくそうウィンドウの変化 (BER=10⁻⁸, ルータキューサイズ: 625 kByte)
 Fig. 14 Congestion window size of TCP-STAR (BER=10⁻⁸, router queue size is 625 kByte).

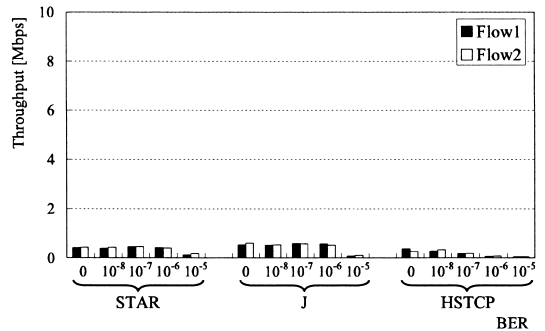


図 16 同種間の公平性の評価 (ルータキューサイズ: 15 kByte)
 Fig. 16 Fairness evaluation in homogeneous scenario when router queue size is 15 kByte.

ルーブットがほぼ等しくなっていることが分かった。

ここで、ABE 機構の本来の趣旨から考えると、コネクション数に反比例して測定帯域値が減少し、ふくそうウィンドウサイズもこれに応じた値に収束すべきと思われる。しかし、コネクション数が 2 である図 14 を見ると、ふくそうウィンドウサイズは帯域幅遅延積 (625 kByte) の半分程度に収束していないことが分かる。これは、次の二つの原因により生じている。まず一つ目の原因は、見積られた利用可能帯域がボトルネックリンクの帯域 (10 Mbit/s) の半分になっていないためである。図 15 は、図 14 における ABE 機構によって得られたそれぞれのフローの見積り帯域の変化を示す。図 15 より、見積り帯域はボトルネックリンクの帯域の半分 (5 Mbit/s) ではなく、平均して約 8 Mbit/s で推移している。この見積り帯域の値は、

図 15 より、二つのフローの値が交互に増減していることで得られている^(注7)。次に二つ目の原因は、2. の (I) で示したように、TCP-STAR が TCP-NewReno のふくそうウィンドウサイズに見積り帯域から計算したウィンドウサイズを追加するためである。したがって、コネクション数が 2 になったとしても、ふくそうウィンドウサイズは単純に帯域幅遅延積の半分程度に収束せず TCP-NewReno のふくそうウィンドウサイズ分だけ大きな値で変動することになる。以上より、TCP-STAR のふくそうウィンドウサイズは帯域幅遅延積 (625 kByte) の半分程度に収束していない。

次に、衛星ルータにおいてふくそうによるパケットロスが頻発する場合の結果を示す。図 16 は、図 12 と

(注7): TCP-STAR と同じ ABE 機構をもつ TCP-J による見積り帯域の変化を観測した結果、TCP-STAR と同様な結果となることを確認した。

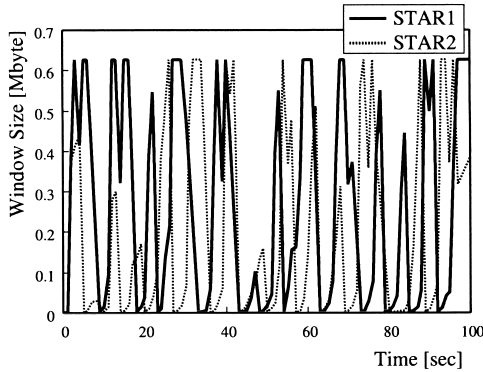


図 17 TCP-STAR のふくそうウィンドウの変化 (BER=0, ルータキューサイズ: 15 kByte)
Fig. 17 Congestion window size of TCP-STAR (BER=0, router queue size is 15 kByte).

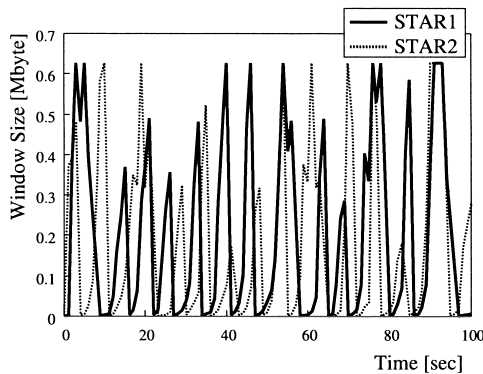


図 18 TCP-STAR のふくそうウィンドウの変化 (BER=10⁻⁸, ルータキューサイズ: 15 kByte)
Fig. 18 Congestion window size of TCP-STAR (BER=10⁻⁸, router queue size is 15 kByte).

同様にルータキューサイズが 15 kByte のときのスループットを示す。図 16 より、ふくそうによるパケットロスが発生しているため、図 12 のときよりも TCP-STAR のスループットは低下しているが、TCP-J 及び HighSpeed TCP と同様に、混在するフローはほぼ同等のスループットが得られている。ここで、図 16 における、BER=0 のときと BER=10⁻⁸ のときのそれぞれの送信端末のふくそうウィンドウの様子を図 17 及び図 18 に示す。図 17、図 18 より、ふくそうによるパケットロスが頻発し見積り帯域が大きく変動しているために、ふくそうウィンドウの変動も大きくなっていることが分かる。しかし、混在するフローのそれぞれのふくそうウィンドウの平均値がほぼ同等の大きさとなっている(例えば、図 17 の BER=0 の場合、STAR1 は 266 kByte、STAR2 は 210 kByte となって

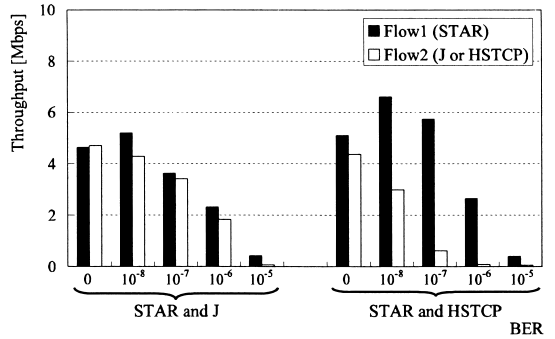


図 19 異種間の公平性の評価 (ルータキューサイズ: 625 kByte)
Fig. 19 Fairness evaluation in heterogeneous scenario when router queue size is 625 kByte.

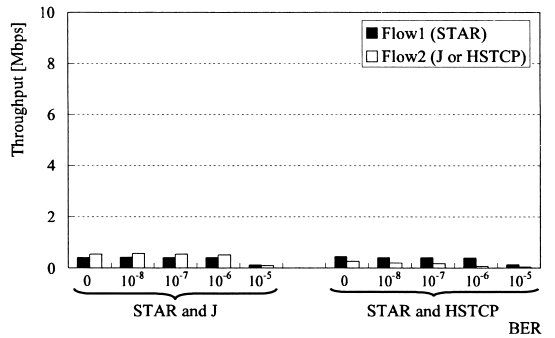


図 20 異種間の公平性の評価 (ルータキューサイズ: 15 kByte)
Fig. 20 Fairness evaluation in heterogeneous scenario when router queue size is 15 kByte.

いる。)ため、重いふくそう状態であっても公平なスループットが得られていることが分かった。

以上より、TCP-STAR は、衛星ルータにおけるふくそう及び典型的な衛星回線に対応するビット誤りによるパケットロスが発生した場合、TCP-J 及び HighSpeed TCP と同様に、同種の TCP のみ存在する環境において、高い公平性が得られることが分かった。

(b) 異種の TCP が混在する場合

次に、TCP-STAR と従来の改良 TCP (TCP-J、HighSpeed TCP) が混在する場合の公平性について評価する。図 19 は、ルータキューサイズが 625 kByte のとき、Sender 1 が TCP-STAR、Sender 2 が TCP-J (J) または HighSpeed TCP (HSTCP) の場合のそれぞれの端末のスループットを示す。同様に図 20 は、ルータキューサイズが 15 kByte のときの結果を示す。図 19 より、軽いふくそう状態であるとき、TCP-STAR は TCP-J とほぼ同等のスループットとなって

いる。しかし、HighSpeed TCP と混在した場合は、大きく差が生じている。これは、ふくそうによるパケットロスが少なくパケットロスがほとんどビット誤りにより発生していることが原因である。つまり、HighSpeed TCP はビット誤りによるパケットロスに対応できていないため、TCP-STAR のスループットが高くなっている。一方、図 20 より、重いふくそう状態の場合、TCP-J が混在したときは、TCP-STAR と TCP-J の値はほぼ同程度となっている。TCP-J の方が TCP-STAR よりも若干スループットが高いが、これは 4.3.1 で示したように AEN 機能により TCP-STAR の方が TCP-J よりもデータの再送に時間がかかっているためである。また、HighSpeed TCP が混在した場合は、軽いふくそう状態のときと同様に、TCP-STAR と HighSpeed TCP 間に大きな差が生じることが分かった。

最後に、図 12、図 19 を見ると、TCP-STAR と混在したときの TCP-J 及び HighSpeed TCP のスループットは、それぞれ TCP-J のみが存在するとき、及び HighSpeed TCP のみが存在するときとほぼ同じ値になっている。したがって、TCP-STAR は混在する他の方式のスループットを圧迫していないことが分かる。

4.4 実装による実験結果とシミュレーションによる実験結果について

まず、実装した TCP-STAR により得られた結果とシミュレーションで得られた値 [5] ~ [7] について検討する。シミュレーションでは、衛星回線の BER が 10^{-8} のとき、TCP-J と比べて約 13.3% スループットが向上していた。一方、実装した TCP-STAR による実験結果を見ると、BER が 10^{-8} のとき、約 13.5% のスループット向上が認められ、シミュレーション結果とほぼ同程度の向上率であることが分かった。また、他の BER の場合においても、実装による実験結果とシミュレーションによる結果はほぼ同等の向上率となっていることを確認した。以上より、実装した TCP-STAR は実環境において、シミュレーションとほぼ同程度のスループット性能を示すことができたと考えられる。

次に、本章の評価項目は、文献 [5] ~ [7] におけるシミュレーション評価で行われた実験と同じであるが、以下の点を新たに明らかにした。

まず、スループット評価において、有線の Long Fat Network を対象とした HighSpeed TCP と Scalable TCP との比較評価を行った結果、HighSpeed TCP と

Scalable TCP よりも衛星ネットワーク上において高いスループット性能が得られることが分かった。これは、TCP-STAR が地上系の Long Fat Network にはないビット誤りによるパケットロスを考慮しているためである。

また、衛星ルータにおけるふくそうと衛星回線におけるビット誤りを考慮して公平性を評価した。その結果、TCP-STAR のみ存在する環境において、ふくそうによるパケットロスとビット誤りによるパケットロスが発生していても TCP-STAR は公平性が得られていることが分かった。ここで、TCP-STAR は他のフローに遠慮してふくそうウィンドウサイズの増加率を抑えることをしないため、パケットロス発生後それぞれのフローで LWC により短時間でふくそうウィンドウサイズが大きく増加する (図 13, 図 14, 図 17, 図 18 参照)。その結果、ふくそうやビット誤りによるパケットロスが発生しても、それぞれのフローのふくそうウィンドウサイズが同じように増加するため、平均ウィンドウサイズが近い値となり、同等のスループットが得られていると考えられる。

5. む す び

本論文では、NetBSD1.6.2-Release のカーネル内に TCP-STAR の実装を行った。次に、ハードウェアネットワークエミュレータを用いたテストベッド上で TCP-STAR の実験的評価を行った。まず、一つのコネクションのみ存在し最大性能が得られる環境において評価した結果、従来 TCP (TCP-NewReno, TCP-J, HighSpeed TCP, Scalable TCP) と比べて、実装した TCP-STAR は高いスループットが得られ、TCP-STAR の各機能は効果的に動作することが分かった。また、複数コネクションが存在し、ふくそうとビット誤りによるパケットロスが混在する環境で評価を行った。その結果、軽いふくそう状態では、TCP-STAR は最も高いスループットが得られるが、重いふくそう状態では、TCP-STAR の AEN 機能のために TCP-J よりも多少低くなることが分かった。また、TCP-STAR の公平性について基礎的な評価を行った。まず TCP-STAR のみ存在する環境において、従来 TCP と同様に公平性が高いことを確認した。次に、異なる TCP が存在する環境で評価を行った。その結果、TCP-J と混在した場合はほぼ同等のスループットが得られたが、ビット誤りによるパケットロスを考慮していない HighSpeed TCP と混在した場合はスループットに差

が生じることが分かった。

今後の課題として、TCP-STAR のより詳細な公平性の評価がある。また、100 Mbit/s を超える広帯域環境における実験、実際に高価な衛星回線を利用した実験の評価も今後の課題である。

謝辞 本研究の一部は日本学術振興会科学研究費補助金基盤研究 C (課題番号 16560345)、及び広島市立大学特定研究費 (番号 6112) により行われた。

文 献

- [1] Japan Aerospace Exploration Agency, "WINDS (Wideband InterNetworking engineering test and Demonstration Satellite)," <http://i-space.jaxa.jp/satellite/WINDS.html>, 2006.
- [2] The Boeing Company, "Connexion by boeing," <http://www.connexionbyboeing.com/>, 2006.
- [3] C. Partridge and T. Shepard, "TCP performance over satellite links," *IEEE Netw.*, vol.11, no.5, pp.44-49, 1997.
- [4] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP over satellite channels using standard mechanisms," RFC 2488, 1999.
- [5] 竹内 達, 小畑博靖, 石田賢治, "衛星インターネットに適した TCP の輻輳制御方式," *信学技報*, IN2004-143, 2004.
- [6] H. Obata, S. Takeuchi, and K. Ishida, "A new TCP congestion control method considering adaptability over satellite Internet," *Proc. 4th International Workshop on Assurance in Distributed Systems and Networks (ADSN2005)*, pp.75-81, 2005.
- [7] H. Obata, K. Ishida, S. Takeuchi, and S. Hanasaki, "TCP-STAR: TCP congestion control method for satellite Internet," *IEICE Trans. Commun.*, vol.E89-B, no.6, pp.1766-1773, June 2006.
- [8] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless link," *Proc. International Conference on Mobile Computing and Networking (Mobicom 2001)*, pp.287-297, 2001.
- [9] 佐藤徳彦, 國司光宣, 寺岡文男, "TCP-J: 無線ネットワークに適したトランスポートプロトコル," *情処学論*, vol.43, no.12, pp.3848-3858, 2002.
- [10] UCB Multicast Network Research Group, "Network simulator ns(ver.2)," <http://www.isi.edu/nsnam/ns/>, 2006.
- [11] 斉藤俊介, 寺岡文男, "無線ネットワーク環境に適したトランスポートプロトコル TCP-J の実装, 解析及び評価," *信学論 (D-I)*, vol.J87-D-I, no.5, pp.508-515, May 2004.
- [12] 小畑博靖, 平 和弘, 石田賢治, "衛星インターネットに適したトランスポートプロトコル TCP-STAR の実装および評価," *信学技報*, IN2005-170, 2006.
- [13] 富士通九州ネットワークテクノロジーズ, "IP ネットワーク・エミュレータ NXS7000F," <http://jp.fujitsu.com/group/qnet/services/nxs/7000/>, 2006.
- [14] S. Floyd, "HighSpeed TCP for large congestion windows," RFC 3649, 2003.
- [15] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol.33, no.2, pp.83-91, 2003.
- [16] 北辻佳憲, 窪田 歩, 山崎克之, 小西和憲, 江崎 浩, "広帯域・遅延環境における TCP 伝送の性能と評価," *信学技報*, IN2002-198, 2003.
- [17] 長谷川剛, 村田正幸, 宮原秀夫, "誤再送を考慮した TCP のふくそう制御方式/エラー回復方式の改善," *信学論 (B)*, vol.J82-B, no.11, pp.2074-2084, Nov. 1999.
- [18] M. Allman, J. Griner, and A. Richard, "TCP behavior in networks with dynamic propagation delay," *Proc. IEEE Global Communication Conference (GlobeCom2000)*, 2, pp.1103-1108, 2000.
- [19] G.R. Wright and W.R. Stevens, *TCP/IP illustrated, vol.2: The implementation*, Addison-Wesley, 1995.
- [20] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," RFC 1323, 1992.
- [21] Netperf, <http://www.netperf.org/netperf/NetperfPage.html>, 2006.

(平成 18 年 10 月 10 日受付, 12 月 28 日再受付)



小畑 博靖 (正員)

IEEE, 情報処理学会各会員。



平 和弘

IEEE, 情報処理学会各会員。



石田 賢治 (正員)

IEEE, ACM, 情報処理学会各会員。