安定したスループットの確保を目指すTCPふくそう制御方式

赤瀬謙太郎[†a)]　小畑　博靖[†b)]　石田　賢治[†c)]

A Gentle TCP Congestion Control Method for Obtaining Stable Throughput

Kentaro AKASE[†a)], Hiroyasu OBATA[†b)], and Kenji ISHIDA[†c)]

あらまし　ベストエフォートネットワークであるIPネットワークにおいて，高い通信品質を必要とする重要通信やリアルタイムアプリケーションに通常のTCPを用いると，必要な帯域が確保できないという問題が生じる．このようなアプリケーションに安定した通信品質を提供するためには，帯域の確保が有効である．現在，送信端末のTCPふくそう制御のみを用いて帯域確保を実現する技術が提案されている．しかしながら，従来方式では帯域確保のために積極的なウィンドウ制御を行うため，通信回線が重度のふくそうに至った場合，逆に帯域確保が困難になるという問題がある．そこで本論文では，背景フロー数が多い状況下や帯域確保TCPフローが複数本存在するといったネットワークのふくそう度が高い状況において，目標帯域を一時的に下げることでふくそうを回避し，その後ふくそうの度合が緩和された際に目標帯域の確保を目指すTCPふくそう制御方式を提案する．シミュレーション評価により，提案方式は従来方式に比べ，ネットワークのふくそう度が高い場合でもボトルネックリンクの帯域利用率を低下させることなく一定の帯域を確保できることが分かった．

キーワード　TCPふくそう制御方式，帯域確保，通信品質

1. まえがき

災害時におけるネットワークへの要求条件の一つに，重要通信の確保がある[1]．重要通信の一例として，災害現場から被害状況を撮影した動画像や映像を，災害対策を行う機関へ転送する場合[2]がある．このような通信では，映像を確実に送り届ける必要があるため，トランスポートプロトコルとしてTCP（Transmission Control Protocol）が利用されることがある．大きな災害時には，地上通信路を利用できない可能性が高く，代替通信路として衛星回線[2]が利用される場合がある．しかし，衛星回線やそれに接続される地上回線の一部に被災地域からのトラヒックが集中することになるため，ふくそうが発生すると予想される．更に，TCPは単一のボトルネックリンクにおいて複数のフローが競合した場合，ふくそうにより多くのパケットが廃棄され，リンク利用率が低下する問題がある[3],[4]．そのため，このような重要通信に通常のTCPを用いると，重要通信に必要な帯域が確保できず通信品質の劣化につながる可能性がある．

また，インターネットの発展により，リアルタイムアプリケーションや動画像ストリーミングといったアプリケーションが普及している．このようなアプリケーションは，パケットの伝送遅延や遅延揺らぎなどの影響を受けやすいため，アプリケーションの品質を保つためにはある一定の安定した通信品質が必要となる．

このような高い通信品質を必要とする重要通信やアプリケーションに，安定した通信品質を提供するためには帯域の確保が有効である．帯域確保を実現する方法としてIntserv[5]やDiffserv[6]といったIP層において品質制御を行う技術が挙げられる．しかし，これらの技術は経路上のすべてのルータに品質制御機構を実装する必要があり，スケーラビリティや導入コストなどの面から実現が困難であるといえる．また，リアルタイム型のアプリケーションにはトランスポートプロトコルとしてUDP（User Datagram Protocol）を用いて通信品質を確保する方法がある．しかし，実際のネットワークには外部からの攻撃を防ぐため，ファイアウォールや家庭のブロードバンドルータなどにUDPによる通信を遮断する機能がついている．その

† 広島市立大学大学院情報科学研究科，広島市
Graduate School of Information Sciences, Hiroshima City University, Hiroshima-shi, 731–3194 Japan
a) E-mail: k-akase@nets.ce.hiroshima-cu.ac.jp
b) E-mail: obata@ce.hiroshima-cu.ac.jp
c) E-mail: ishida@ce.hiroshima-cu.ac.jp

ため，UDP で通信を行うとサービスを提供することができないといった問題が生じる可能性があり，動画像ストリーミングにおいて UDP ではなく TCP が利用されることが多くなってきている [7], [8].

そこで現在，送信側ホストの TCP ふくそう制御を用いてエンドツーエンドで帯域確保を行う技術 [9] が提案されている．この方式は，送信側ホストに修正を加えるだけで実現可能という特徴がある．しかし，この従来方式では帯域確保のために積極的なウィンドウ制御を行うため，通信回線が重度のふくそうに至った場合，帯域確保が困難になるという問題がある．

そこで本論文では，まず背景フロー数や帯域確保 TCP フローが複数本存在するといったネットワークのふくそう度が高い状況において目標帯域を一時的に下げることでふくそうを回避し，その後ふくそうの度合が緩和された際に目標帯域の確保を目指す TCP ふくそう制御方式 [10], [11] を提案する．ここで，目標帯域を一時的に下げることが許されるアプリケーションの一例として，プレイアウト制御を用いたストリーミング [12] がある．プレイアウト制御とは，サーバから送信されたパケットをクライアント側でいったんバッファに蓄積し，再生を遅らせることによってネットワークにおける遅延変動を吸収する制御である．一時的にスループットが低下したとしても，その後回復することができれば再生には問題ない場合が存在すると考えられる．次に，TCP ふくそう制御のみを用いて帯域確保を行う従来方式と提案方式とをネットワークシミュレータ ns-2 [13] により比較評価した．その結果，提案方式は従来方式に比べ，ネットワークのふくそう度が高い場合でもボトルネックリンクの帯域利用率を低下させることなく一定の帯域を確保できることが分かった．

以下，**2.** では従来の TCP ふくそう制御を用いた帯域保証技術について述べ，**3.** では提案方式について述べる．**4.** ではシミュレーションにより提案方式を評価する．最後に，**5.** でまとめと今後の課題について述べる．

2. 関 連 研 究

まず本論文では，アプリケーションから要求された帯域を要求帯域（B_d）と呼び，B_d を確保するために用いるパラメータを目標帯域（B_t）と呼ぶこととする．

TCP ふくそう制御を用いた帯域保証技術として，TCP-MB [9] が提案されている．TCP-MB は送信端末のみを修正するだけで，その他の端末を修正する必要はない．TCP-MB は文献 [14] で提案されている高速 TCP 方式を拡張したものであり，以下のようにして送信帯域 B_s を要求帯域 B_d 周辺で安定させている．

- 目標帯域 B_t の設定

TCP-MB では柔軟なふくそうウィンドウの制御を行うため，B_t を動的に変化させる．B_d と B_s の差分の累積 L を保持し，式 (1) のように $B_t(t)$ を設定する．

$$L(t) = L(t') + (B_d - B_s)(t - t')$$
$$B_t(t) = B_d(t) + \frac{L(t)}{\tau} \quad (1)$$

式 (1) において，t，t'，$B_t(t)$，$L(t)$ は計算を行った時刻，一つ前に計算を行ったときの時刻，時刻 t における目標帯域 B_t，時刻 t における差分の累積 L をそれぞれ示す．また，τ は帯域差分の累積を解消するための時間を表し，アプリケーションのスループットの変動に対する許容度によって決められる．

- 要求帯域 B_d への誘導

式 (1) で求めた $B_t(t)$ を用いて，式 (2) のようにスロースタートしきい値（$ssthresh$）を設定することで，B_d を確保するようにふくそうウィンドウ（$cwnd$）を誘導する．

$$ssthresh = B_t(t) \times \text{最小 } RTT \quad (2)$$

- セグメント喪失を検出後の動作

セグメント喪失を検出後，$cwnd$ を式 (3) のように設定する．

$$cwnd \leftarrow min(ssthresh, cwnd * 0.9) \quad (3)$$

TCP-MB はこのように，ふくそうウィンドウを常に要求帯域の周辺で安定させておき，セグメント喪失が起こってもウィンドウの下げ幅を小さくし，他フローのふくそう制御に期待することで帯域の確保を実現している．

しかし，TCP-MB は背景フローが多い場合や帯域確保 TCP フローが複数本存在するといった，ネットワークのふくそう度が高い状況において大量のパケットを送出してしまい，重度のふくそうを引き起こす．その結果，目標帯域の確保が困難となる．

また，計測した利用可能帯域に基づいてふくそう制御を行い帯域確保を行う技術 [7] や，TCP ふくそう制御を用いずに FEC 技術 [15] を用いて帯域確保を実現する方式 TCP-AFEC [8] が提案されている．しか

し, [7] の方式は帯域確保 TCP フローが複数本存在する場合に中継端末における制御が必要である．また，TCP-AFEC は送信側のホストに加えて受信側のホストにも修正を加える必要がある．そのため，本論文では送信側ホストのみを修正する方式 [9] を比較対象とし, [7], [8] の方式は比較対象としない．

3. 提案方式

提案方式では **2.** で述べた従来方式の問題点を解決し，重度のふくそうを回避するため指定したネットワークのふくそう度に応じて目標帯域 B_t を一時的に低下させる制御を行う．ここでは，ネットワークのふくそう度を推定するための指標として変数 *count* を用いる．提案方式は TCP-MB を次のように拡張したものである．

3.1 現在の送信帯域（B_s）の測定によるふくそう度の推定

ある一定時間間隔ごとの現在の送信帯域（B_s）を測定し，B_t が確保できているか否かを測定する．動画像ストリーミングなどのアプリケーションでは，できるだけリアルタイムに現在の状況を知る必要があると考えられる．そこで本論文では，ACK を受信するごとに送信帯域を測定する．式 (4) のように，B_t が確保できていれば *count* という変数に 1 を加算し，できていなければ 1 を減算する．ここで，*count* の値はほぼ一定間隔で計算されるため *count* の増減量は同じ値とした．

$$\begin{aligned}&\text{if } ((B_t - B_s) \geq 0)\\ &\quad count = count - 1\\ &\text{else}\\ &\quad count = count + 1 \end{aligned} \tag{4}$$

count の値を利用することでネットワークが重いふくそう状態に向かいつつあるか，それともふくそう気味ではあるがネットワークに若干の余裕があるかどうかを判断する．具体的には，*count* の値により以下の（a）〜（c）のようにネットワークの状態を判断する．

（a） 連続して目標帯域が確保できている状況（$count > 0$ で値が十分大きい）では，目標帯域を増やせる余裕がネットワークに十分あると判断．

（b） 連続して目標帯域を確保できない場合が発生するものの，目標帯域が確保できる場合が比較的多く発生する状況（$count > 0$ かつ 0 近辺で変動）では，ふくそう度は上昇傾向にあるが，まだネットワークに多少余裕があると判断．

（c） 目標帯域が連続して確保できず，目標帯域が確保できる場合がほとんどない状況（$count \leq 0$）では，重度のふくそう状態に向かう傾向にあると判断．

3.2 重複 ACK によるセグメント喪失検出後の動作

重複 ACK によるセグメント喪失を検出した場合は，再送タイムアウトによるセグメント喪失の検出に比べ，ネットワークのふくそう度が低いと考えられる．そこで，重複 ACK によるセグメント喪失を検出後，*count* をもとに B_t を式 (5) のように設定する．ここで式 (5) において，α は B_t を上げる割合を示し，$\alpha > 1$ である．

$$\begin{aligned}&\text{if (count} > 0)\\ &\quad B_t = (\alpha \times B_t)(\leq B_d)\\ &\text{else}\\ &\quad B_t \Rightarrow \text{keep a recent value} \end{aligned} \tag{5}$$

count が 0 より大きい場合（（a），（b）の状態），ネットワークに余裕があると判断し B_t を上げる．このとき，B_t は B_d を超えないようにする．つまり，$(\alpha \times B_t) > B_d$ のときは，B_t を B_d の値とする．*count* が 0 以下（（c）の状態）で，重複 ACK によるセグメント喪失を検出している場合，まだ重度のふくそうではなくふくそう状態が軽くなる可能性があると判断し，様子を見るために B_t を変更しない．B_t が変更された場合はスロースタートしきい値を式 (2) を用いて設定し直し，式 (1) により算出される B_t と B_s の累積差分である L 及び *count* を 0 に初期化する．

3.3 再送タイムアウトによるセグメント喪失検出後の動作

再送タイムアウトが発生する場合，ネットワークのふくそう度が高いと考えられる．そこで，再送タイムアウトによりセグメントの喪失を検出した場合，*count* をもとに B_t を式 (6) のように設定する．ここで式 (6) において，β は B_t を下げる割合を示し，$0 < \beta < 1$ である．

$$\begin{aligned}&\text{if (count} > 0)\\ &\quad B_t = (\alpha \times B_t)(\leq B_d)\\ &\text{else}\\ &\quad B_t = \beta \times B_t \end{aligned} \tag{6}$$

式 (6) では，*count* が 0 より大きい場合（(a), (b) の状態），再送タイムアウトが発生したものの帯域に余裕がある可能性があるため，目標帯域を増加させ要求帯域の確保を目指す．$count \leq 0$ の場合（(c) の状態），ネットワークが非常に混雑しふくそう状態の回復が見込めないと判断し，従来方式のように B_t を上げるのではなく，逆に B_t を下げることで重度のふくそうを回避する．その後，スロースタートしきい値を式 (2) を用いて設定し直し，*count* 及び L を 0 に初期化する．

以上のように提案方式では，ネットワークのふくそう度が高く要求帯域の確保が難しい場合には目標帯域をいったん下げ，その後ふくそう度が低くなった場合には目標帯域を要求帯域まで上げるという制御を行う．図 1 に目標帯域変更のフローチャートを示す．この制御により，ネットワークのふくそう度が高い場合においても，重度のふくそうを回避しボトルネックリンクの帯域利用率を向上できることが期待される．しかし，以上の説明に基づく方式では，目標帯域を一時的に下げることで重度のふくそうを回避し安定した通信を行う一方で，獲得できるスループットが低下し要求帯域を確保できない場合がある．また，重度のふくそうを回避したとしても，パケットロス数が多いと再送制御が追いつかなくなり，再送タイムアウトが発生しやすくなる．その結果，再送制御のために要求された帯域の獲得が困難になる場合がある．そこで，提案方式では再送性能の向上が可能な sack オプション [16] を用いることで再送タイムアウト発生を抑制し，性能を向上させている．

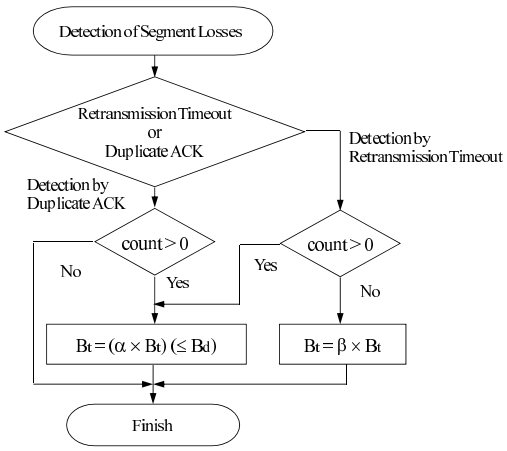

図 1　目標帯域変更のフローチャート
Fig. 1　Flow chart of changing a target bandwidth.

4. シミュレーション評価

4.1 シミュレーション環境

ネットワークシミュレータ ns-2 を用いて，以下の四つの方式を比較評価する．

- 提案方式：Proposal
- sack オプションを利用した従来方式：TCP-MB with sack
- sack オプションを利用しない提案方式:Proposal without sack
- 従来方式：TCP-MB

公平な評価を行うため，sack オプションを利用した従来方式（TCP-MB with sack）及び sack オプションを利用しない提案方式（Proposal without sack）も比較対象とした．

単一のボトルネックリンクを多数のユーザで共有して利用するようなネットワークを想定し，シミュレーションを行う．図 2 にシミュレーションに使用したネットワークモデルを示す．また，表 1 にネットワークパラメータを示す．α 及び β の値は，図 2 の環境で予備実験を行い，提案方式のスループットが最も高くなる値を求め設定した．また，従来方式及び提案方式のパラメータである τ の値は 4 であり，文献 [9] をもとに設定した．

シミュレーション評価では，背景フローが既に存在している状況で，シミュレーションを開始してから帯域確保 TCP（stable TCP と呼ぶ）フローを 60 秒か

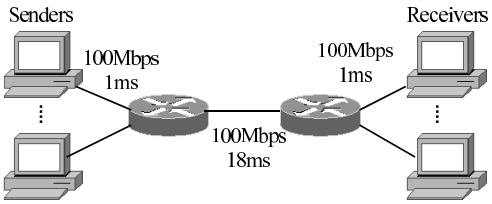

図 2　ネットワークモデル
Fig. 2　Network model.

表 1　シミュレーションパラメータ
Table 1　Simulation parameters.

パケットサイズ [Byte]	1000
ルータのバッファサイズ [packet]	500
α	3
β	0.98
背景トラヒックの TCP バージョン	TCP-Reno
シミュレーション時間 [s]	180

ら 120 秒の間に流し帯域確保を行う．また，提案方式における送信帯域の測定は ACK を受信するごとに行われ，シミュレーション実験において ACK の受信間隔は RTT と同等の値である．結果はシミュレーションを 10 回行った結果の平均値である．また，本論文では帯域確保 TCP フロー間の公平性の評価指標として式 (7) で定義される Fairness index [17] を用いる．

$$F = \frac{(\sum_{i=1}^{n} \bar{x_i})^2}{n \sum_{i=1}^{n} \bar{x_i}^2} \tag{7}$$

式 (7) において，n はフロー数，$\bar{x_i}$ はフロー i の平均スループット，F は n 本のフロー間の Fairness index を示す．Fairness index は 0〜1 までの値をとり，1 に近いほどフロー間の公平性が高いことを示す．

4. 2 long-lived 背景フローが存在する場合の評価

まず，FTP 等を想定した生存時間の長い背景フロー（long-lived フローと呼ぶ）が存在する場合について評価する．

帯域確保 TCP フローが 1 本の場合の背景フローの平均スループットを図 3 に，帯域確保 TCP フローの平均スループットを図 4 に，ボトルネックリンクの帯域利用率を図 5(注1)に示す．この評価において，帯域確保 TCP フローの要求帯域は 20 Mbit/s である．要求帯域は文献 [7] の値をもとに設定した．図 3 の値は各背景フローの平均値である．また，図 4 中の太い実線は要求帯域（B_d）を表す．

図 3 より，sack オプションの有無にかかわらず従来方式と提案方式とで背景フローのスループットに大きな差が見られないことが分かる．しかし，図 4 より，sack オプションなしの場合，背景フロー数が 90 本以上となると従来方式よりも提案方式による帯域確保 TCP フローのスループットが向上している．また図 5 より，従来方式は背景フロー数が増加するとボトルネックリンクの帯域利用率が減少しているが，提案方式はほとんど低下していない．従来方式は要求帯域を確保するためにネットワークのふくそう度が高い場合においてもふくそうウィンドウを高く保ち続け大量のパケットを送出する．その結果，バースト的なパケット廃棄が発生しスループットが低下するとともにボトルネックリンクの帯域利用率も低下したと考えられる．これに対し，提案方式ではネットワークのふくそう度が高く要求帯域の確保が難しいと判断した場合には式 (6) のように目標帯域を一時的に下げ，累積されている差分 L を 0 に初期化する．これにより，スロースタートしきい値が従来方式よりも低く設定されるため，パケットの送出量を抑え重度のふくそうを回

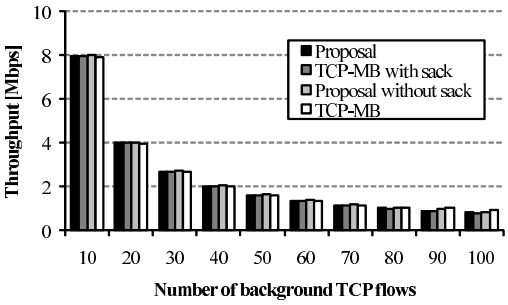

図 3 背景フロー数に対する背景フローの平均スループット（帯域確保 TCP が 1 本の場合）

Fig. 3 Average throughput of background TCP flow for the number of background TCP flows (Number of stable TCP flow is one).

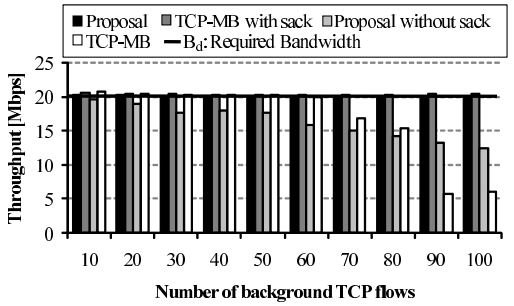

図 4 背景フロー数に対する帯域確保 TCP フローの平均スループット（帯域確保 TCP が 1 本の場合）

Fig. 4 Average throughput of stable TCP for the number of background TCP flows (Number of stable TCP flow is one).

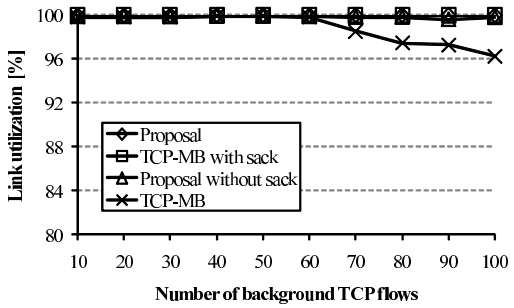

図 5 背景フロー数に対するボトルネックリンクの帯域利用率（帯域確保 TCP が 1 本の場合）

Fig. 5 Bottleneck link utilization for the number of background TCP flows (Number of stable TCP flow is one).

(注1)：図 5 において，Proposal，TCP-MB with sack，Proposal without sack のグラフはほとんど重なっている．

避することが可能となる．その結果，背景フローに対する影響が抑えられ，従来方式に比べてスループット及びボトルネックリンクの帯域利用率が向上したと考えられる．更に，図 4 及び図 5 より，sack オプションありの場合，提案方式，sack オプションを利用した従来方式ともに性能が向上しており，帯域確保 TCP フローが 1 本の場合には，提案方式と sack オプションを利用した従来方式はほぼ同等の性能であることが分かる．

次に，帯域確保 TCP フローが 4 本の場合の背景フローの平均スループットを図 6 に，帯域確保 TCP フローの平均スループットを図 7 に，帯域確保 TCP フロー間の Fairness index を図 8 に，ボトルネックリンクの帯域利用率を図 9 に示す．この評価において各帯域確保 TCP フローの要求帯域は 20 Mbit/s である．また，図 6 及び図 7 の値は各フローの平均をとったものであり，図 7 中の太い実線は要求帯域（B_d）を表す．

図 6 より，帯域確保 TCP フローが複数本で，sack オプションなしの場合，提案方式では背景フローの平均スループットが従来方式と比較して向上している．しかし，sack オプションありで背景フロー数が 30 本以上の場合，提案方式の方が sack オプションを利用した従来方式よりも背景フローのスループットが低下している．また，図 7 及び図 9 より，sack オプションなしの場合，背景フロー数が 60 本程度までであれば提案方式は従来方式よりも同程度か高いスループットを確保でき，また帯域利用率は提案方式の方が常に高い．更に，sack オプションありの場合，背景フロー数が 40 本程度までであれば提案方式と sack オプションを利用した従来方式はほぼ同等の性能であり，背景フロー数が 50 本以上の場合，提案方式のスループットの方が高く，また帯域利用率は常に提案方式の方が高いことが分かる．従来方式はふくそうウィンドウを高く保

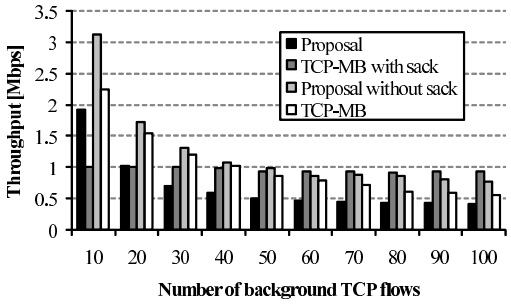

図 6　背景フロー数に対する背景フローの平均スループット（帯域確保 TCP が 4 本の場合）
Fig. 6　Average throughput of background TCP flows for the number of background TCP flows (Number of stable TCP flows is four).

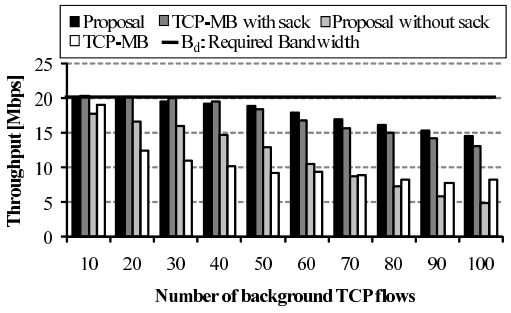

図 7　背景フロー数に対する帯域確保 TCP フローの平均スループット（帯域確保 TCP が 4 本の場合）
Fig. 7　Average throughput of stable TCPs for the number of background TCP flows (Number of stable TCP flows is four).

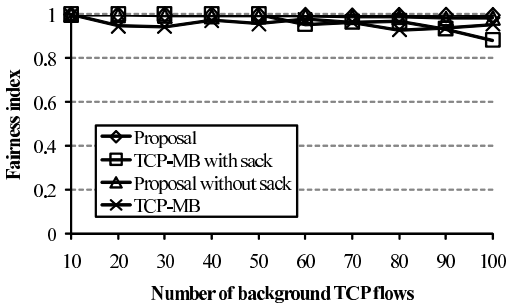

図 8　背景フロー数に対する帯域確保 TCP フロー間の Fairness index（帯域確保 TCP が 4 本の場合）
Fig. 8　Fairness index of stable TCPs for the number of background TCP flows (Number of stable TCP flows is four).

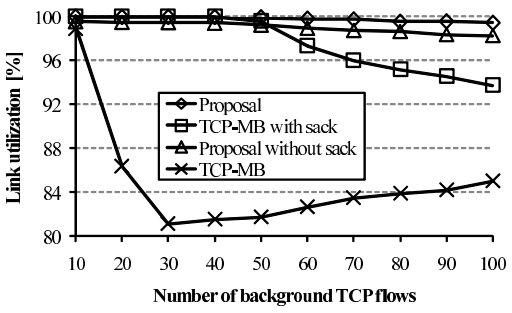

図 9　背景フロー数に対するボトルネックリンクの帯域利用率（帯域確保 TCP が 4 本の場合）
Fig. 9　Bottleneck link utilization for the number of background TCP flows (Number of stable TCP flows is four).

ち続け，他フローのふくそう制御に期待することで帯域確保を実現している．一方，提案方式では，ネットワークのふくそう度が高いと判断した場合に目標帯域を一時的に下げる制御が有効に働き，帯域確保 TCP フローが複数本存在する場合においてもふくそうを回避し，安定した通信を行える．ここで，図 9 において TCP-MB の帯域利用率は，いったん急激に下がってから背景フロー数の増加に伴って少しずつ増加している．TCP-MB は背景フロー数が多くなると，帯域を確保しようとして多量のパケットを送出し背景フローのデータ送信を阻害するため，帯域利用率が低下する．しかし，背景フロー数が多くなるとふくそうが酷くなり TCP-MB の再送タイムアウト回数も増加する．TCP-MB はタイムアウトを起こしている間，再送制御によりデータ送信が制限されるため，背景フローがスループットを獲得できる可能性が逆に出てくる．そのため，背景フローが増加するに従って TCP-MB のタイムアウト回数が増加し，背景フローの利用できる帯域が増えるため帯域利用率も向上したと考えられる．また，図 8 より提案方式は従来方式と同等の公平性を保てていることが分かる．以上より，帯域確保 TCP フロー数が増えた場合にも，提案方式のネットワークのふくそう度が高い場合には目標帯域を下げ重度のふくそうを回避するという制御が有効であるということがいえる．しかし，図 6 より提案方式と sack オプションなしの提案方式を比較すると，提案方式は sack オプションを利用することで帯域を確保できた分，背景トラヒックの平均スループットが低下している．

ここで，表 2 に帯域確保 TCP フローが 4 本の場合における，提案方式の目標帯域の減少回数を示す．表 2 中の値は，各帯域確保 TCP フローのシミュレーション実験 10 回の平均値である(注2)．表 2 より，sack オプションなしの場合，背景フロー数が増加しふくそう度が高くなるにつれ，目標帯域が減少する回数が多くなっていることが確認できる．また，sack オプションを利用することで再送タイムアウトが抑制され目標帯域の減少回数も抑えられることが分かる．

表 2　目標帯域の減少回数（単位：回）
（帯域確保 TCP が 4 本の場合）
Table 2　Number of target bandwidth decrease (Number of stable TCP flows is four).

	背景フロー数（本）									
	10	20	30	40	50	60	70	80	90	100
sack あり	1	1	2	3	3	3	4	4	6	5
sack なし	5	5	8	8	13	19	28	32	44	47

4.3 short-lived 背景フローが存在する場合の評価

最後に Web トラヒック等を想定した生存時間の短い背景フロー（short-lived フローと呼ぶ）が存在する場合について評価する．

帯域確保 TCP フローを常に 5 本流しておき，シミュレーションを開始してから 30 秒から 90 秒の間に，2 秒間の short-lived フローをランダムに流す．この環境で得られた short-lived フローの平均スループットを図 10 に，帯域確保 TCP フローの平均スループットを図 11 に，ボトルネックリンクの帯域利用率を図 12 に示す．各帯域確保 TCP フローの要求帯域は 20 Mbit/s

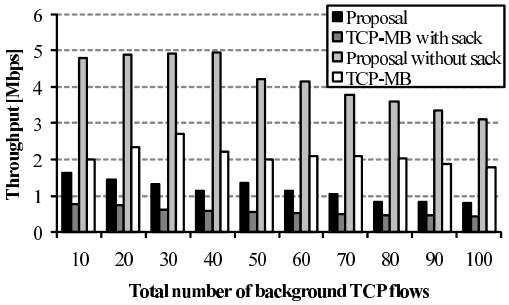

図 10　short-lived フロー数に対する short-lived フローの平均スループット（帯域確保 TCP が 5 本の場合）
Fig. 10　Average throughput of short-lived flows for the number of short-lived flows (Number of stable TCP flows is five).

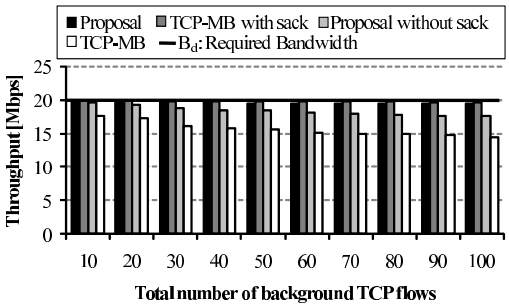

図 11　short-lived フロー数に対する帯域確保 TCP フローの平均スループット（帯域確保 TCP が 5 本の場合）
Fig. 11　Average throughput of stable TCPs for the number of short-lived flows (Number of stable TCP flows is five).

(注2)：表 2 中の値は，小数点以下を四捨五入したため整数値となっている．

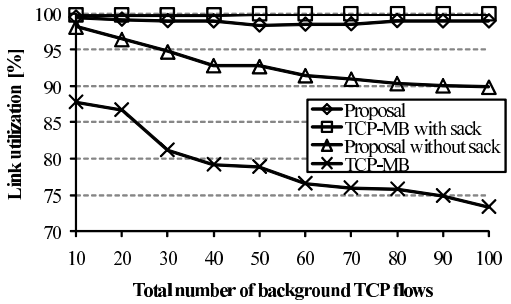

図 12　short-lived フロー数に対するボトルネックリンクの帯域利用率（帯域確保 TCP が 5 本の場合）
Fig. 12　Bottleneck link utilization for the number of short-lived flows (Number of stable TCP flows is five).

である．また，図 10 及び図 11 の値は各フローの平均をとったものであり，図 11 中の太い実線は要求帯域（B_d）を表す．

図 10 より，sack オプションがある場合ない場合のそれぞれにおいて，提案方式では short-lived フローの平均スループットが従来方式の場合と比較して約 2 倍となっている．更に，図 11 及び図 12 より，sack オプションなしの場合，提案方式は従来方式よりも高いスループットを確保できており，ボトルネックリンクの帯域利用率も向上している．また，図 11 及び図 12 より，sack オプションありの場合，提案方式，sack オプションを利用した従来方式ともに要求帯域を確保でき，ボトルネックリンクの帯域利用率も高いことが分かる．したがって，提案方式は sack オプションを利用することで short-lived フローが存在したとしても要求帯域を確保することが可能であり，従来方式及び sack オプションを利用した従来方式よりも背景の short-lived フローに与える影響が小さいことが分かる．

5. む す び

本論文では，ネットワークのふくそう度が高い場合でも重度のふくそうを回避し一定の帯域を確保する TCP ふくそう制御方式を提案した．シミュレーション実験による評価を行った結果，従来方式に比べ背景フロー数が多い場合や帯域確保 TCP フローが複数本存在するといったネットワークのふくそう度が高い状況においても，ボトルネックリンクの帯域利用率を低下させることなく一定の帯域を確保できることを確認した．

今後の課題として，目標帯域設定のためのパラメータ α 及び β をネットワーク環境に応じて適切に決定する方法の検討及び実際的なネットワークを想定した場合の評価が挙げられる．

謝辞　本研究の一部は，総務省戦略的情報通信研究開発推進制度（SCOPE-C 052308002），日本学術振興会科学研究費補助金若手研究 (B)（課題番号 19760260），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] 船越祐介，岡田忠信，“次世代情報通信ネットワークの高信頼化に向けた技術動向，” 信学誌，vol.89, no.9, pp.787–791, Sept. 2006.
[2] JSAT 株式会社，“衛星通信と重要通信，” 総務省重要通信の高度化の在り方に関する研究会（第 5 回），資料 5-1, http://www.soumu.go.jp/joho_tsusin/policyreports/chousa/jyuyou-t/，2008.
[3] C.L.T. Man, G. Hasegawa, and M. Murata, “A simultaneous inline measurement mechanism for capacity and available bandwidth of end-to-end network path,” IEICE Trans. Commun., vol.E89-B, no.9, pp.2469–2479, Sept. 2006.
[4] 西山大樹，タレブ　タリク，橋本和夫，根元義章，加藤寧，“優先度を考慮したルータ主導型 TCP 輻輳制御方式の提案，” 信学技報，NS2006-98, 2006.
[5] S. Braden, D. Clark, and S. Shenker, “Integrated services in the Internet architecture: An overview,” RFC 1633, 1994.
[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated services,” RFC 2475, 1998.
[7] 山根木果奈，長谷川剛，村田正幸，“インラインネットワーク計測に基づく TCP スループットの保証手法，” 信学技報，IN2006-50, 2006.
[8] 津川知朗，藤田範人，浜　崇之，下西英之，村瀬　勉，“TCP-AFEC: TCP によるエンドホスト間の帯域確保のための FEC 冗長度動的決定手法，” 信学技報，IN2006-227, 2007.
[9] 下西英之，村瀬　勉，“帯域確保を実現する TCP 輻輳制御方式の提案，” 2005 信学ソ大（通信），B-7-36, 2005.
[10] 小畑博靖，赤瀬謙太郎，石田賢治，“安定したスループットの確保を目指した TCP 輻輳制御方式，” 信学技報，IN2007-41, 2007.
[11] H. Obata, K. Akase, and K. Ishida, “A TCP congestion control method for securing stable throughput,” Proc. 7th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT2008), pp.220–225, April 2008.
[12] K. Fujimoto, S. Ata, and M. Murata, “Statistical analysis of packet delays in the Internet and its application to playout control for streaming applications,” IEICE Trans. Commun., vol.E84-B, no.6, pp.1504–1512, June 2001.
[13] Network Simulator - ns (version 2), available from

http://www.isi.edu/nsnam/ns/, 2007.
[14] 下西英之，浜　崇之，村瀬　勉，“高速性と公平性を両立した TCP 輻輳制御方式の提案,” 信学技報，IN2004-266, 2005.
[15] R.E. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley, Reading, Massachusetts, 1983.
[16] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP selective acknowledgment options,” RFC 2018, 1996.
[17] D. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” Computer Networks and ISDN Systems, vol.17, pp.1–14, 1989.

（平成 20 年 7 月 22 日受付，11 月 7 日再受付）

赤瀬謙太郎　（学生員）

平 19 広島市大・情報科学・情報工学卒．現在，同大大学院情報科学研究科博士前期課程在学中．TCP ふくそう制御を用いた帯域保証に関する研究に興味をもつ．

小畑　博靖　（正員）

平 12 広島市大・情報科学・情報工学卒．平 14 同大大学院情報科学研究科博士前期課程了．同年 KDDI（株）入社．平 15 広島市大情報科学部助手，平 19 より広島市大情報科学研究科助教，現在に至る．博士（情報工学）．衛星コンピュータネットワーク，及び高速トランスポートプロトコルに関する研究に従事．IEEE，情報処理学会各会員．

石田　賢治　（正員）

平元広島大学大学院工学研究科博士課程後期了．同年広島県立大学講師．同大助教授を経て，平 9 から広島市立大学情報科学部助教授．平 15 より同大情報科学部教授．工博．ネットワーク制御アルゴリズム，アシュアランスシステムに関する研究に従事．IEEE，ACM，情報処理学会各会員．