

TCP のふくそう制御機構に関する研究動向

長谷川 剛[†] 村田 正幸^{††}

Research Trends on TCP Congestion Control Mechanisms

Go HASEGAWA[†] and Masayuki MURATA^{††}

あらまし Transmission Control Protocol (TCP) はインターネットにおいて最も頻繁に用いられるトランスポート層プロトコルであり、近年増加している P2P トラフィックや動画ストリーミングなどのアプリケーションのほとんどが利用している。そのため、TCP の性能、特にふくそう制御機構がネットワーク性能に与えるインパクトは極めて大きい。本論文では、TCP のふくそう制御手法における近年の研究動向をまとめる。特に、無線ネットワーク環境や高速・広帯域ネットワーク環境など、従来の TCP が想定していなかったようなネットワーク環境における性能改善手法に着目する。

キーワード Transmission Control Protocol (TCP), ふくそう制御, 無線ネットワーク, 高速・広帯域ネットワーク

1. ま え が き

Transmission Control Protocol (TCP) [1] は現在のインターネットにおけるトランスポート層プロトコルとして最も多くのネットワークアプリケーションが利用しており、TCP トラフィックは現在のインターネットトラフィックの大部分を占めている [2]。また、インターネットは様々な種類のネットワークを取り込むことによって大規模化し、指数関数的な拡大を続けている [3]。その結果、TCP が誕生した 1970 年代当初には想定することができなかったネットワーク環境が発生している。TCP において最も重要な機能はネットワークふくそうを回避・検知・解消するふくそう制御機構 [4] であり、多様化するネットワーク環境において安定的に TCP による通信を行うことができる大きな要因である。

TCP がどのようなネットワーク環境においても安定的な通信を行うことができる、というロバスト性は、逆にいうと、個々のネットワーク環境における性能の最適化の観点では劣る場合があるということ在意

味する。これは、インターネットが様々なネットワークを IP というルーティングプロトコルで接続しているという性質を鑑みると、やむを得ない性質であると考えられる。しかし、特に近年の光ファイバ技術や無線ネットワーク技術によるアクセスネットワーク環境の劇的な進歩に伴い、そのような環境における TCP の性能が着目されることが多くなり、様々な問題が指摘されつつある。例えば、ネットワークふくそうに加えてリンクエラーによりパケット廃棄が発生、及び変動する無線ネットワーク [5]~[10] や、端末の移動によりエンド間の経路が通信中に変化し、スループット低下が発生するモバイル環境 [11] などが挙げられる。これらをはじめとする様々な問題を解決するために、これまでに多くの TCP に対する改善が行われてきた (例えば [12], [13])。

また、近年のネットワークの高速化により、TCP コネクションが利用できるネットワークの帯域遅延積 (リンク帯域とエンドホスト間の伝搬遅延時間の積) が飛躍的に増大している。例えば、ラウンドトリップ時間 (RTT) が約 130 ms となる太平洋を狭んだ 2 台のエンドホスト間の最低帯域が 100 Mbit/s から 1 Gbit/s 程度である、という環境も一般に利用可能となりつつある [14]。このような高速・高遅延ネットワーク環境において、現在多くの OS の TCP 実装が基本としている TCP Reno を用いると、そのふくそう制御方式の

[†] 大阪大学サイバーメディアセンター, 豊中市
Cybermedia Center, Osaka University, 1-32 Machikaneyama-cho, Toyonaka-shi, 560-0043 Japan

^{††} 大阪大学大学院情報科学研究科, 吹田市
Graduate School of Information Science and Technology,
Osaka University, 1-5 Yamadaoka, Suita-shi, 565-0871 Japan

特徴が原因となって、リンク帯域を十分使うことができない、という問題が指摘されている。これは、TCP Reno が旧来の低速ネットワークを想定して設計されていること、またインターネットユーザがよりスループットなどの性能に敏感になっていることなどに起因していると考えられる。

本論文では、このような新たなネットワーク環境に対応するために行われてきた、TCP のふくそう制御機構に関する近年の研究動向について概説する。特に、無線ネットワーク環境、及び高速・高遅延ネットワーク環境における従来 TCP のふくそう制御機構の問題点を整理し、提案されている改善手法について述べる。

以下、2. においては TCP のふくそう制御機構のうち、特に本論文で着目するふくそうウィンドウサイズの制御手法について概説する。3. 及び 4. においては、それぞれ無線ネットワーク環境及び高速・高遅延ネットワーク環境における TCP のふくそう制御機構の問題点、及び近年提案されている様々な改善手法について述べる。最後に 5. において本論文のまとめを述べる。

2. TCP のふくそう制御方式

TCP Reno のふくそう制御方式は、スロースタートフェーズ及びふくそう回避フェーズと呼ばれる二つのフェーズから構成され、それぞれにおいてふくそうウィンドウサイズ (cwnd) の増加速度が異なる。スロースタートフェーズにおいては、一つの ACK パケットを受信するごとにふくそうウィンドウサイズを 1 パケット増加させる。一方、ふくそう回避フェーズにおいては、一つの ACK パケットを受信するごとにふくそうウィンドウサイズをその逆数分だけ増加させる。すなわち、TCP Reno のふくそうウィンドウサイズを w_{reno} とすると、その更新アルゴリズムは以下のように表すことができる。

$$w_{\text{reno}} \leftarrow \begin{cases} w_{\text{reno}} + 1 & (w_{\text{reno}} < s_{\text{reno}}) \\ w_{\text{reno}} + \frac{1}{w_{\text{reno}}} & (w_{\text{reno}} \geq s_{\text{reno}}) \end{cases} \quad (1)$$

ここで、 s_{reno} は、TCP Reno がスロースタートフェーズからふくそう回避フェーズに移行するときのしきい値 s_{sthresh} である。送信側端末において ACK パケットを受信するたびに上式が用いられることによって、 $w_{\text{reno}} < s_{\text{reno}}$ の場合には RTT ごとに cwnd が 2 倍に

なり、 $w_{\text{reno}} \geq s_{\text{reno}}$ の場合には RTT ごとに cwnd が 1 だけ増加する。

一方、パケット廃棄を検出した場合には、次式のようにふくそうウィンドウサイズを減少させる。

$$w_{\text{reno}} \leftarrow \begin{cases} w_{\text{reno}}/2 & (\text{重複 ACK}) \\ 1 & (\text{タイムアウト}) \end{cases} \quad (2)$$

すなわち、TCP Reno はパケット廃棄を検出するまでふくそうウィンドウサイズを増加させ続け、パケット廃棄をきっかけに減少させる。これは、TCP Reno がパケット廃棄の発生をネットワークふくそうの指標とみなしていることに起因する。本論文ではこのようにパケット廃棄をネットワークふくそうの指標として用いる手法を loss-based 手法と呼ぶ。

3. 無線環境における TCP ふくそう制御機構

3.1 問題点

無線ネットワーク環境における TCP の性能評価に関しては、数多くの研究がこれまでに進行されており [5], [6], [15] ~ [17], それらを通じていくつかの問題が明らかになっている。本節では、それらのうち、ふくそうに無関係なパケット廃棄、及びコネクション間の不公平性に関して説明する。

無線ネットワークは有線ネットワークに比べて通信路のビットエラー率が非常に高く、無線リンクロス、すなわち、ビットエラーが原因となるパケット廃棄が頻繁に発生する。一方、TCP はパケット廃棄を検出すると、それはネットワークふくそうの徴候であると判断し、ウィンドウサイズを減少させることによって、自身のデータ転送速度を低下させる [4]。したがって、無線ネットワーク環境における TCP データ転送において、無線リンクロスによってパケット廃棄が発生すると、不必要なウィンドウサイズの減少が発生し、スループットが低下する。

また、特に無線 LAN 環境において、TCP コネクション間のスループットに不公平が生じることが指摘されている [18]。これは、通信路が上りと下りで帯域を共有することによって、無線基地局からクライアント端末へのパケット送信がふくそうを起こし、パケット廃棄が発生することに起因している。このとき、上りと下りの TCP コネクションが混在していると、データパケットが廃棄されるコネクションと ACK パケットが廃棄されるコネクションが存在する。TCP はデー

タパケットの廃棄に対してはふくそう制御を行うが、ACK パケットの廃棄に対しては、ウィンドウサイズ分すべての ACK パケットが失われぬ限りにおいてはふくそう制御を行わない。そのため、大きな不公平が発生する。

3.2 改善手法

無線ネットワークにおける TCP 性能の改善に関する研究はこれまで多数行われており、無線基地局との連携手法 [19] ~ [25]、送信側 TCP の改変手法 [26] ~ [32]、及び MAC 層と TCP/IP 層の連携に基づくクロスレイヤ制御 [33] などが挙げられる。本章では、それらの中でも特に、無線基地局の改変を必要とするものと、送信側 TCP のみの改変を行うものに分類し、それぞれの利点と欠点を述べる。

3.2.1 無線基地局の改変を伴う手法

文献 [19], [23] ~ [25] などにおいては、有線ネットワークと無線ネットワークの境界に存在する無線基地局を改変し、性能向上が図られている。これは、有線ネットワークと無線ネットワークの特性の違いが与える影響をそれぞれ局所化することによって TCP 性能を向上させるものである。これらの手法は主に、基地局においてコネクション分割を行うもの、及びコネクション監視 (スヌーピング) を行うものに分類される。図 1 に、無線ネットワークに存在する端末が受信側 TCP となる場合の、両方式の挙動を示す。

文献 [20] などにおいて用いられているコネクション分割を行う手法においては、通常エンド端末間に 1 本設定される TCP コネクションを、無線基地局において分割する (図 1 (a))。有線ネットワーク側においては、有線側のエンド端末とのデータパケット及び ACK パケットのやり取りを無線側端末に代わって行う。こうすることによって、有線ネットワークにおけるデータ転送速度が、無線ネットワーク環境に影響を受けないようにすることができる。また、無線ネットワーク側においては、発生するパケット廃棄をすべて無線リンクロスに基づくものとみなすことができるため、無線リンクロスによるパケット廃棄に対してはウィンドウサイズを減少させない、といった制御により、スループット低下を回避することができる。また、無線ネットワークで発生したパケット廃棄に対するパケット再送を、基地局から行うことができるため、再送効率が向上する。更に、コネクション分割手法を用いることにより、分割点において TCP レベルの経路制御を行うことが可能となるため、上位層プロトコルにおける

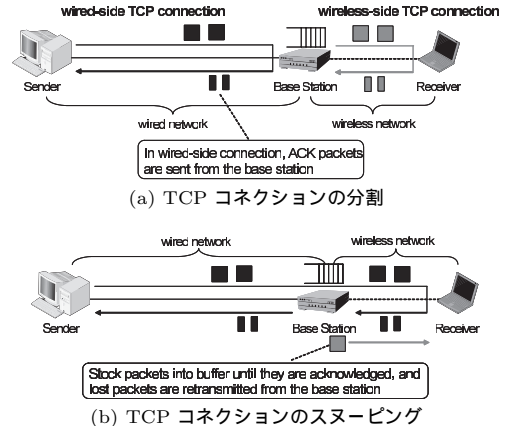


図 1 無線基地局の改変を伴う手法

Fig. 1 Wireless TCP methods with modifications to access points.

経路制御やトラフィック制御を実現するオーバレイルーチング [34] との親和性も高いと考えられる [35]。

一方、文献 [19], [21] などにおいて提案されているコネクション監視 (スヌーピング) 手法は、廃棄されたパケットの基地局からの再送を、コネクション分割を行わずに実現する手法である (図 1 (b))。具体的には、基地局に監視のためのエージェントを導入し、通過する TCP コネクションのデータパケットを、対応する ACK パケットが逆向きに通過するまでキャッシュとして保存し、ACK パケットのシーケンス番号を監視することによって、再送すべきデータパケットを決定する。また、重複 ACK パケットが通過する際には、それを適宜削除することによって、エンド端末からのパケット再送を抑制する。

無線 LAN 環境における TCP コネクション間の不公平性の改善に関しても、様々な手法が提案されている [18], [36] ~ [39]。文献 [18], [38] では、受信側 TCP から返信される ACK パケット内の広告ウィンドウサイズの書換え、あるいは送信される ACK パケット数を制御することにより、コネクション間の不公平を改善している。文献 [36] では、基地局の MAC プロトコルのパラメータを変更することによって、また [39] では、上下コネクションに対してレート制御を行うことによって、それぞれ公平性を改善している。

これらの基地局における手法は、無線ネットワーク環境において従来問題となっている、ふくそうによるパケット廃棄と無線リンクロスによるパケット廃棄の区別などの、ネットワーク環境の違いを認識する必要

がないため、TCP のふくそう制御機構にとって有利となる。しかし、基地局においてコネクション情報の管理やパケットのキャッシュ処理のために必要となるメモリや CPU 資源量の増加が問題となる。また、IPSec や VPN 技術などによりエンド端末間でパケット暗号化が行われる場合には、基地局において TCP ヘッダを参照することができないため、これらの手法を適用することができない。

またこれらの手法は、インターネットにおけるプロトコル設計の際の指針として従来考えられてきたエンドツーエンド原理 [40] に反するものであるが、近年はオーバレイネットワークやファイアウォールシステムなど、ネットワーク内においてセッションの切断・中継を前提とするシステムが普及しており、導入障壁は以前ほど高くないと考えられる。

3.2.2 送信側 TCP の改変を伴う手法

文献 [26] ~ [32] などにおいては、送信側 TCP を改造することによって、無線アクセスネットワーク環境における TCP 性能の改善を目指している。これらの手法において主眼となるのは、パケット廃棄や遅延変動などの事象が、無線ネットワーク部分で発生しているのか、有線ネットワーク部分で発生しているのかの区別を、基地局によるコネクション分割や監視を行うことなく実現することである。これにより、無線ネットワーク部分で発生する無線リンクロスが引き起こす、ふくそうウィンドウサイズの不当な減少を回避し、スループットを維持することが可能となる。

文献 [31] において提案されているレートベース手法は、送信側 TCP において ACK パケットの到着間隔から受信側端末における平均受信レートを推定し、推定値に基いてふくそう制御機構のパラメータである、スロースタートフェーズとふくそう回避フェーズの切換のためのしきい値を設定する。これにより、無線リンクロスによるパケット廃棄を検出した際にふくそうウィンドウサイズが小さくなることを防止する、あるいは素早く回復させることが可能となる。

一方、Jitter-based TCP (JTCP) [30] に代表される手法は、送信側 TCP において検出したパケット廃棄が、無線リンクロスに起因するものか、ネットワークふくそうによるものかを判別し、適切な制御を行うものである。それらの手法の多くは、有線ネットワークにおけるネットワークふくそうによってパケット廃棄が発生する直前に観測される、ラウンドトリップ時間 (Round Trip Time: RTT) の増大を利用し、パケッ

ト廃棄を検知した際の RTT 値を参照することによって、そのパケット廃棄が有線ネットワークのふくそうによって発生したのか、無線ネットワークの無線リンクロスによって発生したものを判別する。無線リンクロスによるものと判断した場合には、ウィンドウサイズを減少させないことで、スループット低下を防止する。

これらの手法は、無線基地局の改変が必要な方式に比べて、送信側 TCP の変更のみで実現することができるため、より導入が容易であると考えられる。

4. 高速・高遅延環境における TCP ふくそう制御機構

4.1 問題点

TCP Reno を高速・高遅延ネットワーク環境において用いると、大きなリンク帯域を十分使う程度のスループットを得ることができないという問題が指摘されている [41]。図 2 に、送受信端末間のラウンドトリップ時間 (RTT) が 100 ms、リンク帯域が 10 Gbit/s であるような大きな帯域のリンク上で、パケット長が 1500 Byte である 1 本の TCP Reno コネクションを用いてデータ転送を行ったときの、ふくそうウィンドウサイズの変化の様子を示す。この環境においてはリンク帯域を十分使う程度のスループットを得るためには、パケット廃棄率が 2×10^{-10} 以下である必要がある [41]。これは現在の光ファイバ技術では実現困難な性能である。また図 2 から、一度パケット廃棄が発生すると、ふくそうウィンドウサイズが回復するまでに、40000 RTT (約 4000 秒) 以上の時間を要することが分かる。これは、TCP Reno において 1 Gbit/s を超えるスループットを継続的に得ることは現実的には不可能であることを表している。この問題は、式 (1) から分かるように、ふくそうウィンドウサイズの増加の

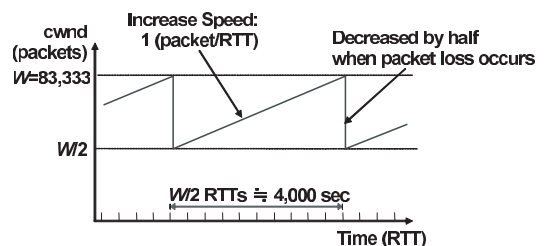


図 2 TCP Reno の高速・高遅延環境における問題点
Fig. 2 TCP Reno problems in high-speed and long-delay networks.

幅がラウンドトリップ時間 (RTT) ごとに 1 パケットと非常に小さいにもかかわらず、式 (2) に示すように、パケット廃棄を検出した際にウィンドウサイズを 1/2 以下へと大きく減少させるために、ふくそうウィンドウサイズがなかなか大きくなりにくいことに起因している。

このような高速・高遅延ネットワークにおける TCP Reno の問題点に対する数多くの改善手法が近年提案されている。本章では、それらのうち主要なものを紹介する。それらの改善手法の多くは、TCP Reno がパケット廃棄をネットワークふくそうの指標として判断しているのに対して（あるいはそれに加えて）、別の指標を導入している。本章では新たに導入している指標によって分類を行っている。また本章では、特に指定しない限りは、ふくそう回避フェーズにおけるふくそうウィンドウサイズの増減アルゴリズムについて説明する。なお、スロースタートフェーズに関しては、多くの改善手法が TCP Reno と同じアルゴリズム（式 (1) の 1 行目）を採用している。また、ふくそう回避フェーズにおいては、ふくそうウィンドウサイズがある値以下である場合には、TCP Reno と同じアルゴリズム (2.) を用いることで、低速ネットワーク環境における TCP Reno との親和性を確保している手法も存在する。

4.2 Loss-based 手法

4.2.1 HighSpeed TCP (HSTCP) [41]

HSTCP は、高速・高遅延環境向けの改善手法として比較的初期に提案された手法である。HSTCP は TCP Reno と同様、パケット廃棄のみをネットワークふくそうの指標として利用するが、現在のふくそうウィンドウサイズの大きさに合わせて、ふくそうウィンドウサイズの増加速度及びパケット廃棄検出時の減少幅を次式に従って調整する。

$$w_{\text{hstcp}} \leftarrow \begin{cases} \text{(廃棄未検出時)} \\ w_{\text{hstcp}} + \frac{a(w_{\text{hstcp}})}{w_{\text{hstcp}}} \\ \text{(廃棄検出時)} \\ (1 - b(w_{\text{hstcp}}))w_{\text{hstcp}} \end{cases}$$

$$a(w) = \frac{2w^2 \cdot b(w) \cdot p(w)}{2 - b(w)}$$

$$b(w) = \frac{\log(w) - \log(W_{\text{low}})}{\log(W_{\text{high}}) - \log(W_{\text{low}})} (b_{\text{high}} - 0.5) + 0.5$$

$$p(w) = \exp \left[\frac{\log(w) - \log(W_{\text{low}})}{\log(W_{\text{high}}) - \log(W_{\text{low}})} \cdot \{\log(P_{\text{high}}) - \log(P_{\text{low}})\} + \log(P_{\text{low}}) \right]$$

なお、TCP Reno は $a(w) = 1$ 及び $b(w) = 1/2$ に相当する。また、 P_{high} 、 P_{low} 、 W_{high} 及び W_{low} のパラメータは、ネットワークのパケット廃棄率と目標とするスループットから算出される値であり [41] においてはパケットサイズが 1500 Byte、パケット廃棄率が 10^{-7} の環境において、10 Gbit/s のスループットが達成できるようなパラメータが一例として紹介されている。この式は、HSTCP は現在のふくそうウィンドウサイズが大きいほど、その増加速度を大きくし、減少幅を小さくすることを意味している。これにより、HSTCP はネットワークの帯域遅延積の大きさに応じたウィンドウサイズの増加・減少速度を用いることができる。

しかし、HSTCP や後述する Scalable TCP などの、TCP Reno に比べて積極的にふくそうウィンドウサイズを増加させるプロトコルが、TCP Reno コネクションと共存した場合、それらのプロトコルは高いスループットを獲得する反面、共存する TCP Reno コネクションのスループットを低下させるという問題点がある。この問題に対する改善手法として我々の研究グループでは gentle HighSpeed TCP 手法を提案している [42]。

4.2.2 Scalable TCP (STCP) [43]

STCP は、TCP Reno のふくそう回避フェーズのようなふくそうウィンドウサイズを直線的に増加させるフェーズはもたず、スロースタートフェーズと同じ、ふくそうウィンドウサイズを指数的に増加させる式のみを用いる。

$$w_{\text{stcp}} \leftarrow \begin{cases} w_{\text{stcp}} + 0.01 & \text{(廃棄未検出時)} \\ w_{\text{stcp}} \cdot 0.875 & \text{(廃棄検出時)} \end{cases} \quad (3)$$

上式が ACK パケットを受信するたびに用いられると、ウィンドウサイズが現在の値から k 倍になるために必要な RTT 数が、現在のウィンドウサイズの値そのものに依存せず一定となる。これにより、リンク帯域の大きさに関係なく、一定のリンク利用率を維持することができる。また式から、TCP Reno や HSTCP が Additive Increase Multiplicative Decrease (AIMD) に従ってウィンドウサイズを増減するのに対して、Multiplicative Increase Multiplicative

Decrease (MIMD) を採用している．しかし，ボトルネックリンクを共有しているすべての送信側端末へ，ネットワークふくそうの指標が同時に伝達される環境においては，AIMD はフロー間の公平性を維持することができるが，MIMD では公平性を維持することができないことが文献 [44] において指摘されている．したがって，STCP は，RTT が同じ STCP コネクション間においても公平性を達成できないという問題をもつ．

4.3 Delay-based 手法

Loss-based 手法はネットワーク内でパケット廃棄が発生するまでふくそうウィンドウサイズの増加を停止しないため，理想的に動作した場合においても，周期的なパケット廃棄の発生を避けることができない．一方，ルータの出力リンクにおいて高負荷時にパケットが蓄積されるバッファが FIFO 規律に従っている場合，バッファがいっぱいになりパケット廃棄が発生する前に，そのリンクにおいてバッファリング遅延が増大することが期待される．そこで，各パケットの RTT を監視し，その増大をネットワークふくそうの初期段階の指標として利用する手法（本論文では delay-based 手法と呼ぶ）が提案されている．理想的に動作すると，loss-based 手法では避けることのできないパケット廃棄を完全に回避することが可能となる．

4.3.1 TCP Vegas [45]

TCP Vegas は 1995 年に発表された手法であり，高速・高遅延ネットワーク向けに提案された手法ではないが，その基本アルゴリズムはその後登場した多くの手法において利用されている．TCP Vegas においては，以下の式を用いることによって，ネットワーク内で滞留している（バッファリングされている）と考えられるパケット数を推測する．

$$Expected = cwnd/baseRTT$$

$$Actual = cwnd/RTT$$

$$Diff = (Expected - Actual) \cdot baseRTT$$

ここで， $cwnd$ はふくそうウィンドウサイズ， $baseRTT$ はこれまでに観測された最小の RTT， RTT は現在の RTT， $Diff$ はネットワーク内滞留パケット数の推測値である．TCP Vegas は $Diff$ の値に基づいて以下の式に従ってふくそうウィンドウサイズを RTT に 1 回増減させる．

$$w_{vegas} \leftarrow$$

$$\begin{cases} w_{vegas} + 1, & \text{if } Diff < \frac{\alpha}{base_rtt} \\ w_{vegas}, & \text{if } \frac{\alpha}{base_rtt} < Diff < \frac{\beta}{base_rtt} \\ w_{vegas} - 1, & \text{if } \frac{\beta}{base_rtt} < Diff \end{cases}$$

すなわち，パケットの RTT の増大に伴い $Diff$ が大きくなると，パケット廃棄が発生していなくてもふくそうウィンドウサイズを小さくする．これにより，周期的なパケット廃棄を避けることができる．

しかし，ふくそうウィンドウサイズの増加・減少速度が TCP Reno のふくそうウィンドウサイズの増加速度である 1 packet/RTT であるため，高速・高遅延ネットワーク環境においては TCP Reno と同様の問題をもつと考えられる．

4.3.2 FAST TCP [46]

FAST TCP は TCP Vegas と同様に，観測された最小の RTT である $baseRTT$ と現在の RTT である RTT を利用し，以下の式に従ってふくそうウィンドウサイズを増減させる．

$$w_{fast} \leftarrow \min \left\{ 2w_{fast}, \left((1 - \gamma)w_{fast} + \left(\frac{baseRTT}{RTT} w_{fast} + \alpha \right) \right) \right\}$$

α はパラメータであり，ネットワーク内滞留パケット数の目標値に相当する．TCP Vegas と異なり，目標となるふくそうウィンドウサイズが現在値に比べて大きい場合にはふくそうウィンドウサイズを指数的に増加させるため，ネットワークの帯域遅延積が大きい場合にも素早くネットワーク利用率を向上させることができる点が特長である．その反面，パラメータ α の適切な設定が難しいという欠点をもつ．

4.4 Hybrid 手法

TCP Vegas や FAST TCP などの delay-based 手法は，それが単独で用いられる場合にはスループット，公平性，収束速度などの面で優れていることが明らかになっている [45], [46]．しかし，TCP Reno や HSTCP のような loss-based 手法と混在した環境においては，delay-based 手法を用いるコネクションのスループットが低下するという問題が文献 [47], [48] などにおいて指摘されている．これは以下の理由による．混在環境においてネットワーク帯域が使い切れ，ルータバッファにパケットが蓄積し始めると，RTT が増加する．その際，delay-based 手法は RTT の増加に伴いふくそうウィンドウサイズを小さくするが，

loss-based 手法はパケット廃棄が発生するまでふくそうウィンドウサイズを大きくし続ける。したがって、ボトルネックリンクを両手法のコネクションが共有した場合、delay-based 手法のコネクションは、loss-based 手法のコネクションに比べてスループットが低下する。

この問題に対し、delay-based 手法に loss-based 手法を組み合わせる Hybrid 手法が提案されている [49], [50]。これらの手法は、通常の delay-based 手法と同様に、RTT が増加しておらずネットワーク帯域が使い切られていないと判断された場合にはふくそうウィンドウサイズを TCP Reno よりも大きく増加させる。その際、文献 [50] においては TCP Reno と同じ速度でふくそうウィンドウサイズを増加した場合の仮想値を管理しておく。また文献 [49] においては、ふくそうウィンドウサイズの増加幅を TCP Reno 相当の部分と、そうでない追加部分に分けて管理する。その後、RTT が増加し始めると、ふくそうウィンドウサイズの（大幅な）増加を停止する。その後、TCP Reno と同じ増加幅でふくそうウィンドウサイズを増加させ、loss-based 手法を用いる（パケット廃棄の発生までふくそうウィンドウサイズを大きくし続ける）。すなわち、ネットワークの未使用帯域がある場合には、delay-based 手法によってそれを高速に使い切るように動作し、ふくそう時には loss-based 手法で動作することによって、共存する TCP Reno との公平性を維持している。

4.5 その他の手法

その他、パケット廃棄が発生したときのふくそうウィンドウサイズを記憶し、その値を基にその後のふくそうウィンドウサイズの制御を行う BIC-TCP [51] や CUBIC-TCP [52]、また ACK パケットの到着間隔から現在のスループットを推測し、その値をふくそうウィンドウサイズの制御に用いる TCP Westwood [53] などが提案されている。また、Explicit Congestion Notification (ECN) [54] などを使って、ネットワーク側から利用可能性帯域やふくそうの有無などの情報を明示的に取得することで、エンド端末におけるふくそう制御の効率を高める検討も行われている [55] ~ [57]。

4.6 並列 TCP 手法

これまでに紹介した改善手法は TCP のふくそう制御方式そのものを変更することによって問題を解決する手法であるが、その他の手法として、複数本の TCP (Reno) コネクションを並列的に用いてデータ転送を行う並列 TCP 手法が考えられる。本手法は TCP ふ

くそう制御手法を改変する手法ではないが、高速・高遅延ネットワーク環境においてスループットを向上させる有効な手段であるため、ここで簡単に紹介する。

並列 TCP 手法は OS のカーネルの改変が必要なく、アプリケーションプログラムによって実現可能であるため、アプリケーションのデータ転送スループットを向上させる方法としては非常に有用である。例えば GridFTP [58] には並列 TCP 手法によるデータ転送方式が組み込まれている。

並列 TCP 手法によって効率的なデータ転送を行う際には、同時に利用する TCP コネクション数の適切な設定が重要である。例えば文献 [59] においては並列 TCP 手法によるデータ転送スループットを数学的解析によって導出している。また文献 [60] においては並列コネクション数を動的に調整する手法が提案されている。しかし、適切なコネクション数はリンク帯域、伝搬遅延時間、競合するコネクション数、用いる TCP のふくそう制御方式の特性など、多くのパラメータに大きく依存する [61]。また、並列 TCP コネクション数の動的な制御は、上述した TCP のふくそう制御方式の改善手法に比べてネットワーク環境の変動への追従性や端末負荷などの面で劣ると考えられる。

5. む す び

本論文では、TCP のふくそう制御機構に関する近年の研究動向を概説した。主に無線ネットワーク環境及び高速高遅延ネットワーク環境における問題点を整理し、近年提案されている様々な改善手法を紹介した。無線ネットワーク環境に関しては、今後も WiMAX や LTE など、従来とは異なる特性をもつネットワークが登場するため、これらのネットワークにおいて TCP を用いた場合の性能評価や、特性を考慮したふくそう制御機構の改善が求められると考えられる。また、高速・高遅延ネットワーク環境向けの改善手法に関しては、有効な改善手法が出揃った印象があり、今後は、どの改善手法が最も有効なのか、あるいは、条件に応じたふくそう制御機構の切替手法などに関する研究が進むものと考えられる。

文 献

- [1] J.B. Postel, "Transmission control protocol," Request for Comments 793, Sept. 1981.
- [2] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of Internet traffic in 1998-2003," Proc. Winter International Symposium on Information and Communication Technologies (WISICT

- 2004), Jan. 2004.
- [3] Hobbes' Internet timeline 10. available at <http://www.zakon.org/robert/internet/timeline/>
- [4] V. Jacobson, "Congestion avoidance and control," Proc. ACM SIGCOMM'88, pp.314–329, Aug. 1988.
- [5] F. Lefevre and G. Vivier, "Understanding TCP's behavior over wireless links," Proc. Communications and Vehicular Technology, pp.123–130, Oct. 2000.
- [6] V. Tsoussidis and I. Matta, "Open issues on TCP for mobile computing," Wireless Communications and Mobile Computing, vol.2, no.1, pp.3–20, Feb. 2002.
- [7] E.S. Chang and R. Taborek, "Recommendation of 10e-13 bit error rate for 10 gigabit ethernet," Proc. IEEE802.3 High Speed Study Group July 1999 Plenary Week Meeting, July 1999.
- [8] IEEE, "IEEE standard for local and metropolitan area networks: Overview and architecture," IEEE Std 802-2001, Dec. 2001.
- [9] C.H. Nam, S.C. Liew, and C.P. Fu, "An experimental study of ARQ protocol in 802.11b wireless LAN," Proc. Wireless Personal Multimedia Communications (WPMC 2002), Oct. 2002.
- [10] 3GPP, "Services and service capabilities," Technical Specification TS 22.105 v6.2.0 (2003-6), June 2003.
- [11] T. Goff, J. Moronski, D.S.Phatak, and V. Gupta, "FreezeTCP: A true end-to-end TCP enhancement mechanism for mobile environments," Proc. IEEE INFOCOM 2000, March 2000.
- [12] V. Jacobson and R. Braden, "TCP extensions for long-delay paths," Request for Comments 1072, Oct. 1988.
- [13] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," Request for Comments 2582, April 1999.
- [14] C. Marcondes, A. Persson, M.Y. Sanadidi, M. Gerla, H. Shimonishi, T. Hama, and T. Murase, "Inline path characteristic estimation to improve TCP performance in high bandwidth-delay networks," Proc. PFLDnet 2006, Feb. 2006.
- [15] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and solutions," IEEE Commun. Mag., vol.43, issue 3, pp.27–32, March 2005.
- [16] A. Eshete, A. Arcia, D. Ros, and Y. Jiang, "Impact of WiMAX network asymmetry on TCP," Proc. IEEE WCNC 2009, April 2009.
- [17] S. Hassayoun, P. Maille, and D. Ros, "On the impact of random losses on TCP performance in coded wireless mesh networks," Proc. IEEE INFOCOM 2010, March 2010.
- [18] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP fairness over wireless LAN," Proc. IEEE INFOCOM 2003, vol.2, pp.863–872, March 2003.
- [19] F. Sun, L.S.C. Li, and O.K. Victor, "Design of SNACK mechanism for wireless TCP with new snoop," Proc. IEEE WCNC 2004, March 2004.
- [20] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," Proc. 15th International Conference on Distributed Computing Systems, pp.136–143, May 1995.
- [21] H. Balakrishnan, S. Seshan, and R.H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," ACM/Baltzer Wireless Networks, vol.1, no.4, pp.469–481, Dec. 1995.
- [22] K. Wang and S.K. Tripathi, "Mobile-end transport protocol: an alternative to TCP/IP over wireless links," Proc. IEEE INFOCOM 1998, vol.3, pp.1046–1053, March 1998.
- [23] K. Ratnam and I. Matta, "WTCP: An efficient mechanism for improving TCP performance over wireless links," Proc. Third IEEE Symposium on Computers Communications, pp.74–78, June 1998.
- [24] H. Balakrishnan and R.H. Katz, "Explicit loss notification and wireless web performance," Proc. IEEE GLOBECOM Internet Mini-Conferenve, Nov. 1998.
- [25] K. Jin, K. Kim, and J. Lee, "SPACK: Rapid recovery of the TCP performance using SPlit-ACK in mobile communication environments," Proc. IEEE Region 10 Conference, vol.1, pp.761–764, Sept. 1999.
- [26] C. Carlo, F. Rosario, and L. Daniele, "The TCP adaptive-selection concept," IEEE Syst. J., vol.2, pp.83–89, March 2008.
- [27] L. Cui, S.J. Koh, X. Cui, and Y.J. Kim, "Adaptive increase and adaptive decrease algorithm for wireless TCP," Proc. ICNC 2007, Aug. 2007.
- [28] F. Ge and L. Tan, "A partial super fast recovery algorithm for fast TCP," Proc. AusWireless 2007, Aug. 2007.
- [29] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," IEEE J. Sel. Areas Commun., vol.22, no.4, pp.747–756, May 2004.
- [30] E.H.K. Wu and M.-Z. Chen, "JTCP: Jitter-based TCP for heterogeneous wireless networks," IEEE J. Sel. Areas Commun., vol.22, no.4, pp.757–766, May 2004.
- [31] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," Proc. ACM MOBICOM, pp.287–297, July 2001.
- [32] H. Lai, K.-C. Leung, and V.O. Li, "Enhancing wireless TCP: A serialized-timer approach," Proc. IEEE INFOCOM 2010, March 2010.
- [33] A. Shadmand and M. Shikh-Bahaei, "TCP dynamics and adaptive MAC retry-limit aware link-layer adaptation over IEEE 802.11 WLAN," Proc. CNSR 2009, May 2009.
- [34] D.G. Andersen, H. Balakrishnan, M.F. Kaashoek, and R. Morris, "Resilient overlay networks," Proc. 18th ACM Symposium on Operating Systems Principles, Oct. 2001.

- [35] I. Maki, G. Hasegawa, M. Masayuki, and T. Murase, "Performance analysis and improvement of TCP proxy mechanism in TCP overlay networks," Proc. ICC 2005, May 2005.
- [36] Y. Fukuda and Y. Oie, "Unfair and inefficient share of wireless LAN resource among uplink and downlink data traffic and its solution," IEICE Trans. Commun., vol.E88-B, no.4, pp.1577–1585, April 2005.
- [37] J. Ha and C.-H. Choi, "TCP fairness for uplink and downlink flows in WLANs," Proc. IEEE Global Telecommunications Conference 2006, pp.1–5, Nov. 2006.
- [38] F. Keceli, I. Inan, and E. Ayanoglu, "TCP ACK congestion control and filtering for fairness provision in the uplink of IEEE 802.11 infrastructure basic service set," Proc. IEEE International Conference on Communications, pp.4512–4517, June 2007.
- [39] N. Blefari-Melazzi, A. Detti, I. Habib, A. Ordine, and S. Salsano, "TCP fairness issues in IEEE 802.11 networks: Problem analysis and solutions based on rate control," IEEE Trans. Wirel. Commun., vol.6, pp.1346–1355, April 2007.
- [40] J.H. Saltzer, D.P. Reed, and D.D. Clark, "End-to-end arguments in system design," ACM Trans. Comput. Syst., vol.2, pp.277–288, Nov. 1984.
- [41] S. Floyd, "HighSpeed TCP for large congestion windows," Request for Comments 3649 (Experimental), Dec. 2003.
- [42] Z. Zhang, G. Hasegawa, and M. Murata, "Performance analysis and improvement of HighSpeed TCP with TailDrop/RED routers," IEICE Trans. Commun., vol.E88-B, no.6, pp.2495–2507, June 2005.
- [43] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," ACM SIGCOMM Computer Communication Review, vol.32, no.2, April 2003.
- [44] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," J. Computer Networks and ISDN Systems, vol.17, pp.1–14, June 1989.
- [45] L.S. Brakmo and L.L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," IEEE J. Sel. Areas Commun., vol.13, no.8, pp.1465–1480, Oct. 1995.
- [46] C. Jin, D.X. Wei, and S.H. Low, "FAST TCP: motivation, architecture, algorithms, performance," Proc. IEEE INFOCOM 2004, March 2004.
- [47] J. Mo, R.J. La, V. Anantharam, and J. Walrand, "Analysis and comparison of TCP reno and vegas," Proc. IEEE INFOCOM'99, March 1999.
- [48] G. Hasegawa, K. Kurata, and M. Murata, "Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet," Proc. IEEE ICNP 2000, Nov. 2000.
- [49] K.T.J. Song, Q. Zhang, and M. Sridharan, "Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks," Proc. PFLDnet 2006, Feb. 2006.
- [50] H. Shimonishi, T. Hama, and T. Murase, "TCP-Adaptive Reno: Improving efficiency-friendliness tradeoffs of TCP congestion control algorithm," Proc. PFLDnet 2006, Feb. 2006.
- [51] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," Proc. IEEE INFOCOM 2004, March 2004.
- [52] I. Rhee and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," Proc. PFLDnet 2005, Feb. 2005.
- [53] TCP WESTWOOD Home Page. available at <http://www.cs.ucla.edu/NRL/hpi/tcpw/>
- [54] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," RFC 3135, Sept. 2001.
- [55] A. Kuzmanovic, "The power of explicit congestion notification," Proc. IEEE SIGCOMM 2005, Aug. 2005.
- [56] R. Diana and E. Lochin, "ECN verbose mode: a statistical method for network path congestion estimation," Proc. IEEE INFOCOM 2010, March 2010.
- [57] I. Qazi, L. Andrew, and T. Znati, "Congestion control using efficient explicit feedback," Proc. IEEE INFOCOM 2009, April 2009.
- [58] W. Allcock, "GridFTP: Protocol extensions to FTP for the Grid," available at: <http://www.ggf.org/documents/GFD.20.pdf>, April 2003.
- [59] T. Hacker and B. Athey, "The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network," Proc. 16th IEEE-CS/ACM International Parallel and Distributed Processing Symposium (IPDPS), Aug. 2001.
- [60] T. Ito, H. Ohsaki, and M. Imase, "Automatic parameter configuration mechanism for data transfer protocol GridFTP," Proc. 2006 International Symposium on Applications and the Internet (SAINT 2006), Jan. 2006.
- [61] Z. Zhang, G. Hasegawa, and M. Murata, "Reasons not to parallelize TCP connections for long fat networks," Proc. SPECTS 2006, Aug. 2006.

(平成 22 年 11 月 4 日受付, 23 年 1 月 17 日再受付)



長谷川 剛 (正員)

平 7 阪大・基礎工退学．平 9 同大大学院基礎工学研究科博士前期課程了．同年同後期課程退学．同年同大経済学部助手．平 10 同大大学院経済学研究科助手．平 12 同大サイバーメディアセンター助手．平 14 同助教授．主としてトランスポート層プロトコル，オーバレイネットワーク，無線ネットワーク，ネットワーク計測などの研究に従事．情報処理学会，IEEE 各会員．博士（工学）．



村田 正幸 (正員)

昭 57 阪大・基礎工卒．昭 59 同大大学院博士前期課程了．同年 IBM 東京基礎研究所入社．昭 62 阪大大型計算機センター助手．平元同大基礎工学部助手．平 3 同講師．平 4 同助教授．平 11 同大大学院基礎工学研究科教授．平 12 同大サイバーメディアセンター教授．平 16 同大大学院情報科学研究科教授．また平 19 より（独）情報通信研究機構上席研究員（現在，上席招へい研究員）．現在，新世代ネットワーク戦略プロジェクト研究統括．ネットワークアーキテクチャなどに関する研究に従事．IEEE，ACM 各会員．平 21 より IEEE COMSOC Japan Chapter Chair．工博．