TCPの輻輳制御パラメータの設定による リンク特性の変化に対する通信性能改善の可能性

石橋 賢一† 森島 直人‡ 砂原 秀樹†

奈良先端科学技術大学院大学 情報科学研究科† 奈良先端科学技術大学院大学 附属図書館研究開発室‡

概要: MIPv6やNEMOを利用することで，移動体通信における移動透過性を確保できる．しかし，トランスポート層以上ではリンク特性の急激な変化にうまく追従できないため，アプリケーションからみたサービス品質という観点からは，移動透過性の確保だけでは不十分である．特に，TCPはインターネットにおける利用範囲が大きいため，リンク特性の変化に対するTCPの通信品質劣化を改善する意義は大きい．本稿では，リンク特性の変化に対するTCPの通信品質劣化が，リンク特性と輻輳制御パラメータの不整合に起因することを示した．さらに，移動体環境におけるTCPの通信品質劣化の改善に向け，輻輳制御パラメータを強制的に変更する手法に着目し，その可能性を実験を通じて明らかにした．
キーワード: ネットワークモビリティ，TCP，輻輳制御

Towards improving TCP performance in a mobile environment: Prospects of the approach of adjusting TCP congestion control parameter

Kenichi Ishibashi† Naoto Morishima‡ Hideki Sunahara†

Graduate School of Information Science, Nara Institute of Science and Technology†
Research Division Digital Library, Nara Institute of Science and Technology‡

Abstract: Although MIPv6 and NEMO provide mobility transparency, they are not enough for an application to retain its quality of service in a mobile environment because of possible performance degradation of communication. TCP might also degrade in such a circumstance since its congestion control is sensitive to link characteristics. This transport protocol is used so extensively that it is quite important to avoid this degradation. In this paper, we show parameters for TCP congestion control become inappropriate with a change of uplink characteristics, which causes performance degradation. Furthermore, we conduct some experiments to examine that adjustment of congestion parameters from outside of TCP stack is a prospective approach to suppress performance degradation.
Keywords: Network Mobility, TCP, Congestion Control

1 はじめに

移動体通信をインターネット層から支援するため，Mobile IPv6 (MIPv6) [1] やNetwork Mobility (NEMO) [2] が標準化されている．その主な目的は，移動体に移動透過性と呼ばれる性質を付加することである．移動透過性は，端末の移動に伴うIPアドレスの変化を隠蔽するもので，不変の論理的なIPアドレスを導入することによって実現している．この性質を利用すれば，異なるデータリンク層を横断したセッションの維持や，他の端末から常に同一のIPアドレスで移動体にアクセスすることが可能になる．しかしながら，移動体通信における問題は，移動透過性を確保するだけでは解消しない．

特に，複雑な輻輳制御を行うTCPは固定端末を想定して設計されているため，データリンク層の変化に伴うリンク特性の急激な変化に追従できない．インターネットで交換されるトラフィックの多くがTCPを利用していることから，移動体通信を実用的なものにするためには，移動透過性の確保だけでなく，リンク特性が変動する環境下におけるTCPの通信性能の改善が急務である．

そこで我々は，TCPの輻輳制御に関するパラメータを，データリンクの変化に応じて追従させることで，移動体通信環境におけるTCP通信性能の改善を目指す研究に取り組んでいる．本稿では，まず，リンク特性の変動に伴う輻輳制御パラメータの挙動を調査するために行った予備実験について報告する．予備実験の結果を考察し，すべての輻輳制御パラメータにおいて，リンク特性の変化に追従できないケースの存

在を示す．次に，リンク特性の変化に合わせて輻輳制御パラメータを設定する手法の可能性を検証するために，輻輳制御パラメータを設定する機構を実装し，その機構を用いた簡単な実験を行う．実験を通じて，輻輳制御パラメータを設定する手法の可能性，および問題点を明らかにする．

2 関連研究

従来のTCPはセッションの継続中にリンク特性が変化することを考慮していないため，移動に伴うリンク特性の変化に追従できない．このため，輻輳の発生やスループットの低下など，通信品質の低下が生じる場合がある．この問題を解決する手法として，ウィンドウ告知を用いる手法，輻輳制御に関連するパラメータを変更する手法，および，複数インタフェイスを利用する手法が提案されている．

ウィンドウ告知を利用する手法であるFreeze-TCP[3]は，受信ノードが移動に伴う通信の断絶を予測すると，受信ウィンドウサイズをゼロとして送信ノードへ告知する．送信ノードは非ゼロのウィンドウ告知を受け取るまで，ウィンドウ検査セグメントのみを送出し，データセグメントの送出を停止する．また，再送タイムアウトによる輻輳制御を行わない．接続性の回復を検知した受信ノードは，非ゼロのウィンドウ告知を送信ノードへ送信し，セッションを継続する．この機構により，移動に伴うTCPの通信品質の低下を防ぐ．しかし，この手法は通信の断絶に対応するものであり，リンク特性の変化には対応していない．移動前のリンク特性に最適化された輻輳制御パラメータは，移動後のリンク特性に適さない場合があり，通信の断絶に対応するだけでは不十分である．

輻輳制御に関連するパラメータを変更する手法には，スロースタート閾値(ssthresh)をHoeのアルゴリズムを用いて変化させる手法[4]や，輻輳ウィンドウサイズ(cwnd)をリンクの帯域に応じて変化させる手法[5]などがある．これらの手法では，輻輳制御に関連するパラメータを適切に設定することで，輻輳の発生やスループットの低下を防ぐ．しかし，リンク特性の変化がTCPの輻輳制御に与える影響は多岐にわたり，単一のパラメータを設定するだけではリンク特性の変化には追従できない．リンク特性の変化が輻輳制御に与える影響を詳細に調査した上で，各パラメータを適切に設定するアルゴリズムの検討が必要である．

また，複数のインタフェイスを利用する手法[6]も提案されている．この手法では，切り替え時に利用可能な複数のインタフェイスを用いることで，TCPセッションのシームレスな移動を実現する．この手法は移動前後に複数のインタフェイスの可用性を前提としているため，現実的には適用できる環境が限定的である．また，単一ノードの移動透過性を前提としているため，NEMO環境への応用は困難である．

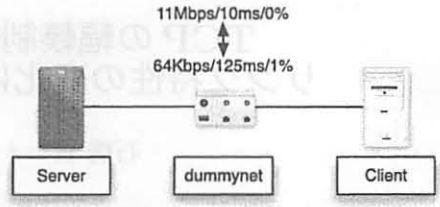

図1: 実験環境

3 予備実験

TCPでは，輻輳ウィンドウサイズ，スロースタート閾値，および，再送タイムアウト値が輻輳制御に関わる重要なパラメータである．リンク特性の変化がこれらのパラメータに与える影響を明らかにするため，擬似的にリンク特性が変化する環境で予備実験を行った．

3.1 実験環境

実験環境を図1に示す．サーバとクライアントを設置し，さらにその間にdummynet[7]を利用できるブリッジを設置している．dummynetは帯域や遅延を制御できるため，dummynetの部分は論理的にある特性(帯域，遅延，パケット廃棄率)を持ったネットワークと等価とみなすことができる．例えば，IEEE802.11b相当の特性(帯域11Mbps，片道遅延10ms)やPHS相当の特性(帯域64Kbps，片道遅延125ms)を持つリンクを擬似的に作成できる．また，実験はサーバやクライアント，dummynetを協調させて行う必要があるため，実験用のネットワークとは別にジョブ管理・メッセージ通知用のネットワークも構築している．

この環境において，dummynetでリンク特性を変化させながら，サーバとクライアントの間でHTTPを用いたファイル転送を行った．この際，輻輳ウィンドウサイズをtcpdumpを用いて計測した．以下に示す全てのグラフでは，横軸が送信したTCPセグメント数，縦軸が輻輳ウィンドウサイズ，垂直の破線がリンク特性が変化したタイミングを表す．

なお，サーバおよびクライアントにはNetBSD 3.0を利用した．TCPのバージョンはNewRenoである．dummnynetにはFreeBSD 6.1を利用した．

3.2 輻輳ウィンドウサイズ

輻輳ウィンドウサイズ(cwnd)は，TCPセグメントの転送速度を調整するための極めて重要なパラメー

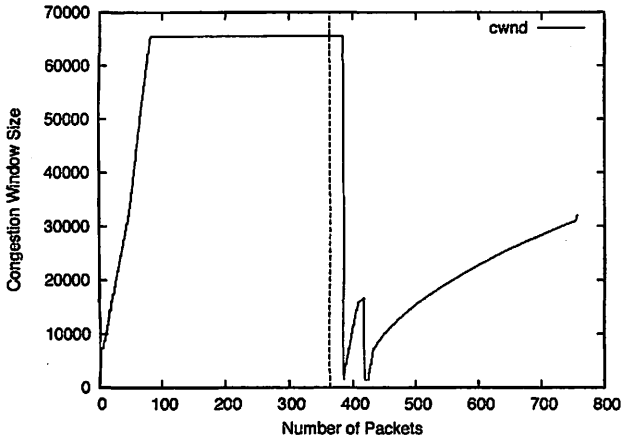

図 2: 帯域の変化と輻輳ウィンドウサイズ

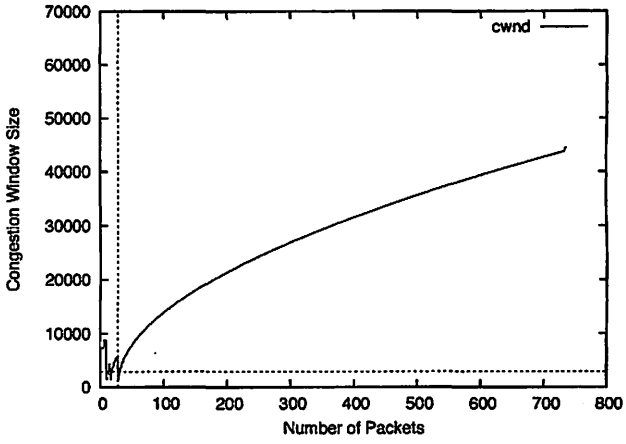

図 3: 帯域変化によるスロースタート閾値の不整合

タである．TCP は cwnd の値に基づいてセグメントを送出する．値が大きいほど同時に多くのセグメントを送出できる．

一般的に，広帯域では狭帯域と比較して同時に送出できるデータ量が大きいため，cwnd が大きくなる傾向にある．したがって，リンク特性が広帯域から狭帯域に変化した場合，広帯域に適応した cwnd の値が変化後のリンクに対して大きすぎることが多い．このような状態でセッションが継続すると，許容量以上のセグメントをネットワークに送出することになる．その結果，輻輳が発生し，パケット喪失やタイムアウトによって輻輳制御が働き，最終的には TCP の通信性能を著しく損なう原因となる．

図 2 は，リンクの帯域が急激に減少した場合の cwnd の遷移を表している．帯域は破線の前後で 11Mbps から 64Kbps に変化している．帯域が大きく変化した後，cwnd が最小値である 1MSS (Maximum Segment Size) まで減少していることがわかる．これは，パケットの喪失によって輻輳制御が働いたことを示している．このことから，帯域の変化に対する cwnd の不整合に起因した輻輳制御により，TCP の性能が損なわれることが分かる．

3.3 スロースタート閾値

TCP は通信性能をリンクに最適化するため，データの交換と同時に cwnd を上昇させていく．性能を急速にあげるためには cwnd の上昇幅を大きくとればよい．しかし，むやみな上昇は通信路上の輻輳を引き起こす原因となる．そこで，TCP では cwnd の上昇幅を二種類の段階に分けて決定している．性能向上のために上昇幅を大きくとる段階をスロースタート段階と呼ぶ．一方，輻輳を回避するために上昇幅を小さくする段階を輻輳回避段階と呼ぶ．この二つの段階は，スロースタート閾値 (ssthresh) と呼ばれる値によって決定される．すなわち，cwnd の値が ssthresh よりも小さければスロースタート段階になり，それを超えると輻輳回避段階へと移行する．

ssthresh は，最後に輻輳制御が働いた時点の $cwnd/2$ に設定される．したがって，その値は通信の経過とともにリンク特性に適応し，狭帯域リンクであれば ssthresh が小さく，広帯域リンクであれば ssthresh が大きくなる傾向にある．しかし，リンク特性が急激に変化した場合，その後の通信性能に悪影響を与えることがある．

まず，リンクが狭帯域から広帯域へ変化した場合を考えよう．狭帯域に適応した ssthresh は小さく抑えられており，広帯域に変化してもこの値は変化しない．このため，cwnd の上昇に十分な余地があるにも拘わらず，すぐに輻輳回避段階へと移行する．これによって cwnd の上昇が緩やかになり，結果としてリンクの性能を十分に活かすことができない．

図 3 は，リンク帯域が急激に増加した場合の cwnd の遷移を表している．帯域は破線の前後で 64kbps から 11Mbps に変化している．また，水平の破線は帯域が増加した瞬間の ssthresh を表している．この図からも分かるように，いったん ssthresh が小さくなると，cwnd を十分に大きくするためにはかなりの時間が必要である．そのため，帯域が増加したにも拘わらず，小さな ssthresh によって輻輳回避段階へと移行し，結果として十分に帯域を活かせていない．

一方，広帯域から狭帯域へ変化した場合を考えよう．図 2 で，帯域の変化後に cwnd が 1MSS になっていることはすでに述べた．このとき，ssthresh はタイムアウトする直前の cwnd の半分になっている．この値は変化後のリンクの適正値よりもかなり大きい．そのため，輻輳回避段階に移行するべき状況でもスロースタート段階を継続し，結果として輻輳が発生，輻輳制御によって再び cwnd が 1MSS に戻されている．

以上のことから，リンク特性が変化した後の帯域とssthreshに不整合が発生すると，リンクの性能を十分に活かすことができなかったり，不必要な輻輳制御が働いたりすることがわかる．これらの現象は，いずれの場合でもTCPの性能を低下させることにつながっている．

3.4 再送タイムアウト値

TCPは，送信したセグメントに対するACKを一定時間受信しないと，パケット喪失が発生したと判断してセグメントの再送を試みる．ACKを待つ時間のことを再送タイムアウト値(xmit_timer)と呼び，計測したRTT (Round Trip Time)から算出している．リンクのレイテンシが変化すると通信の当事者間のRTTも変化する．そのため，リンクのレイテンシが急激に変化すると，xmit_timerの算出に利用するRTTが正しいものではなくなり，xmit_timerが適正値から乖離して不具合を生じることがある．

リンクの遅延が大きくなる方向へ変化する場合を考えよう．変化前の遅延の小さなリンクで計測したRTTからxmit_timerを導出すると，RTTが大きくなった変化後のリンクに対しては短すぎる可能性が高い．このため，セグメントが正しく到着しているにも拘わらずタイムアウトが発生し，セグメントが再送されると同時に不必要な輻輳制御よってcwndが減少する．

変化前と変化後の遅延特性の組み合わせを変えながら実験したところ，片道遅延の差が100倍程度ある場合にこの問題が発生することが分かった．図4はタイムアウトが発生したときのcwndの遷移を示している．また，帯域を一定に保持したまま遅延を10msから2000msに変化させている．この図では，リンク特性の変化とほぼ同時にcwndの減少がみられる．これは，喪失していないセグメントに対するタイムアウトが発生したため，不必要な輻輳制御が働いたことを示している．

このことから，リンク特性の変化によるxmit_timerとRTTの整合性から不必要な輻輳制御が発生し，TCPの性能低下につながるケースがあることが分かる．

3.5 予備実験で得られた知見

リンク特性が急激に変化した場合，cwnd，ssthresh，および，xmit_timerの値が適正値から乖離することがある．このような値を基にcnwdの調整や輻輳回避の制御が行われると，性能があがらなかったり，不要な輻輳制御によるTCPの性能低下を招くことが明らかになった．現実には，これらのパラメータの不整合が同時に発生し，複合的要因によって性能低下が発生していると考えられる．

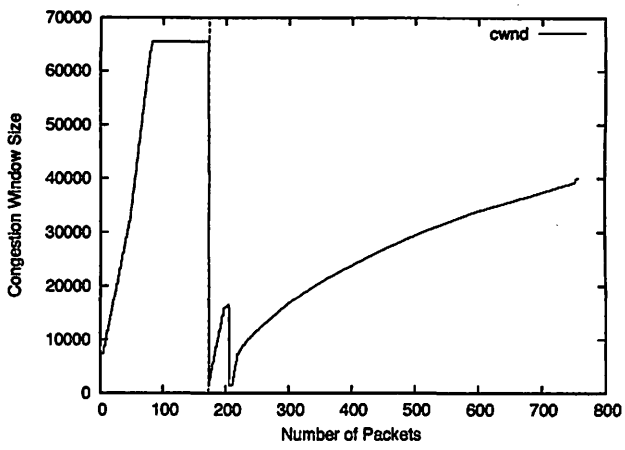

図 4: 遅延の変化による不必要な輻輳制御

4 パラメータ修正手法の検証

予備実験の結果から，リンク特性が変化する環境においては，適正値から乖離した輻輳制御パラメータの影響により，TCPの通信性能が劣化することが明らかになった．この問題は，乖離した輻輳制御パラメータを適切な値に修正する手法によって解決できると考えられる．そこで，この手法の可能性を検証するため，輻輳制御パラメータの値を設定する機構を構築し，ssthreshの値を例として実験を行った．

4.1 TCBの値を設定する機構

TCPコネクションを確立しているエンドホストは，個々のTCPコネクション毎にTCB制御情報を保持しており，輻輳制御に用いるパラメータもこのTCBに保持されている．

TCBはTCPコネクションが確立されるとともにOS(Operating System)のカーネル内部に動的に生成され，コネクションの切断とともに破棄される．本研究では，ユーザランドからTCBの値を自由に設定するために，TCB値設定機構を実装した．この機構はカーネル内に疑似デバイスとして実装しており，ユーザランドからTCBの一覧を取得したり，特定のTCBを指定して輻輳制御パラメータを変更したりすることができる．なお，OSにはNetBSD 3.0を利用した．

4.2 実験

4.1節で述べたTCB値設定機構を用いて，狭帯域のリンクから広帯域のリンクへ変化したときにssthreshの値を設定する実験を行った．

実験方法は以下の通りである．まず，3.1節で述べた実験環境において，広帯域のリンク(帯域11Mbps，片道遅延10ms，パケットロス率0%)と狭帯域のリンク(帯域64Kbps，片道遅延125ms，パケットロス率

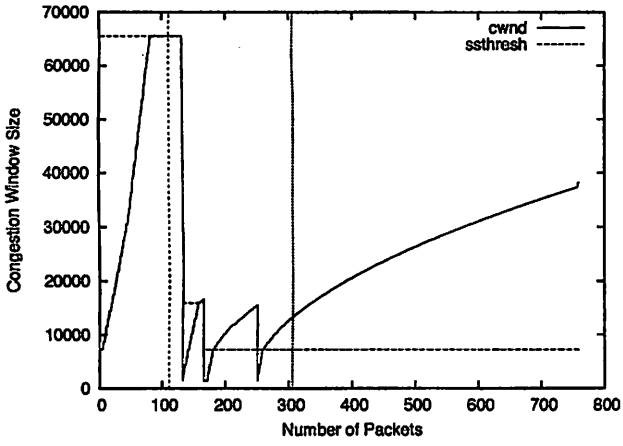

図 5: ssthresh を設定しなかった場合の cwnd の遷移

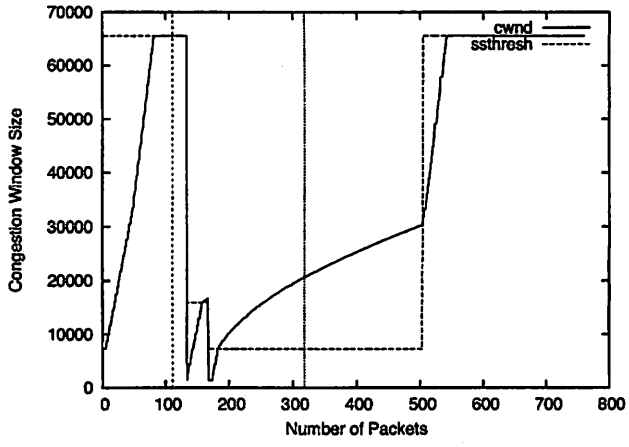

図 6: ssthresh を設定した場合の cwnd の遷移

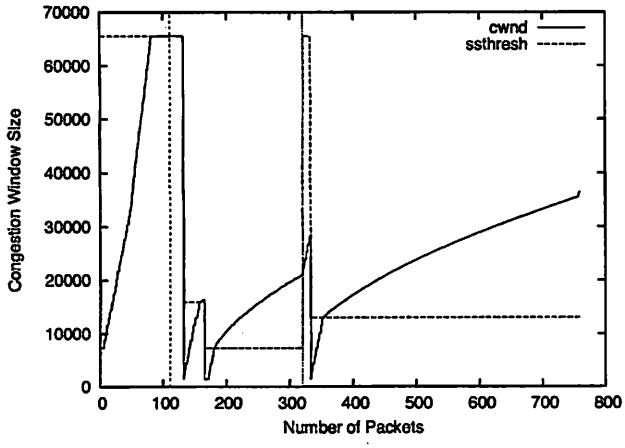

図 7: ssthresh を設定しても変化に追従できない事例

1%) を用意する．次に，リンク特性を広帯域のリンク→狭帯域のリンク→広帯域のリンクと変化させながら，1Mbytes のファイルを転送する．2 回のリンク特性の変化をそれぞれ，状態遷移 A，および，状態遷移 B とする．状態遷移 A は転送開始後 2 秒後，状態遷移 B は状態遷移 A の 40 秒後である．また，状態遷移 A の直前の ssthresh の値を $ssthresh_p$ とする．このとき，状態遷移 B の後に，ssthresh の値をユーザランドから変更しない場合，および，ユーザランドから強制的に $ssthresh_p$ に設定する場合の cwnd の変化を比較する．なお，ssthresh の値を設定するのは状態遷移 B の 3 秒後とした．

図 5，図 6，および，図 7 は転送されたパケット数に対する cwnd の変化を表している．リンク特性の変化が生じたタイミングは垂直の破線で表しており，左から 1 番目の破線が状態遷移 A，2 番目の破線が状態遷移 B である．すべての場合において，状態遷移 A 直後の急激なリンク特性の劣化によって輻輳が発生し，輻輳制御が行われている．これに伴い，cwnd の値，および ssthresh の値が減少している．

図 5 は，状態遷移 B 後に ssthresh を設定しなかった場合を表している．狭帯域リンクにおいて発生した輻輳によって減少した ssthresh の値の影響により，状態遷移 B で帯域が増大したにも拘わらず，輻輳回避段階のままとなっている．これは，ssthreash が輻輳制御によってのみ変更されるためである．この図では状態遷移 B 後に輻輳が発生していないため，ssthreash の値は変更されない．その結果，リンク A の帯域を有効に利用できるようになるまでに時間を要することがわかる．

一方，図 6 は，状態遷移 B の 3 秒後に ssthresh を $ssthresh_p$ へと強制的に変更した場合の，cwnd の変化を表している．ssthresh の値を変更するまでは，図 5 と同様に輻輳回避段階で推移するが，ssthresh を変更するとスロースタート段階へと移行していることがわかる．これによって cwnd の上昇幅が大きくなり，短時間で適切な cwnd へと到達する．このグラフは，リンクの帯域幅が急激に大きくなる際には，輻輳回避段階からスロースタート段階へと移行する必要があることを示唆している．このような輻輳制御状態の移行は通常の TCP では不可能である．したがって，輻輳制御パラメータの強制的な変更が，リンク特性の変化に追従するために有効である可能性があるといえる．

しかし，輻輳制御パラメータの変更によって，リンク特性の変化に常に追従できるとは限らない．図 7 は図 6 と同様に，状態遷移 B の後に ssthresh の値を $ssthresh_p$ に設定している．ただし，設定するタイミングは状態遷移の直後である．この場合，ssthresh の値を設定した直後に輻輳制御が行われ，cwnd と ssthresh が減少している．これは，状態遷移 B の直前に発生したセグメントの喪失が，ssthresh の変更後に検出されたことによるものであると考えられる．このことから，リンク特性が変化した直後に輻輳制御パラメータ

を設定するだけでは，TCP がその変化にうまく追従できない可能性があることを示している．

5 今後の方針

4.2 節の実験結果より，適正値から乖離した輻輳制御パラメータを修正する手法により，TCP の輻輳制御がリンク特性の急激な変化に追従できることがわかった．しかし，パラメータを設定するタイミングによっては，その効果を十分に発揮できない場合があることも明らかとなった．今後はこの結果を踏まえ，NEMO 環境における，リンク特性の急激な変化に対し，TCP の輻輳制御が追従できる手法の実現を目指す．実現には，適切な輻輳制御パラメータを決定するアルゴリズム，および，リンク特性の変化を通知する機構が必要となる．

5.1 TCB 値設定アルゴリズム

リンク特性の変化に適応して，TCB 値を設定するアルゴリズムを TCB 値設定アルゴリズムと呼ぶ．このアルゴリズムは，以下の二点が重要な要素となる．

第一に，適切な TCB の値の決定する手法を確立しなければならない．現段階においては，元のパラメータ値に設定する手法の検証しか行っていない．しかし，元のパラメータ値を設定する手法のほかにも，それぞれの特性の変化幅に応じて各パラメータの増減量を設定する手法や，リンク特性を推定してパラメータを決定する手法なども検討する必要がある．

第二に，決定した TCB の値を設定するタイミングの決定が重要な要素となる．4.2 節の実験では，単純に ssthresh の値を設定するだけではリンク特性の変化に追従できない事例を示した．送信したセグメントのシーケンス番号と受信した ACK の状況などを考慮して，各パラメータを設定するタイミングを決定する必要がある．

5.2 リンク特性の変化を通知する機構

NEMO 環境において，TCP コネクションを確立するなど，通信の主体となるのは MNN である．一方，リンク特性が変化したことを検知するのは MR(Mobile Router) である．従って，MNN がリンク特性の変化に追従するためには，MR が検知したリンク特性の変化を，MNN に通知する仕組みが必要となる．複数の MNN に効率的に通知するために，この機構にはマルチキャストの利用を予定している．

6 まとめ

本稿ではリンク特性が変化する環境を想定し，そこで生じる TCP の通信品質の低下に着目した．問題点を整理するために予備実験を行い，現状の TCP ではリンク特性の変化によって輻輳制御パラメータが適正値から乖離してしまい，通信性能の劣化を引き起こすことを示した．また，適正値から乖離した輻輳制御パラメータを修正する手法の可能性を確かめるために，TCB 値設定機構を実装し，その機構を用いた簡単な実験を行った．実験の結果，輻輳制御パラメータを設定する手法はリンク特性の変化に素早く追従できる一方で，パラメータを設定するタイミングによっては効果的に機能しないことが明らかとなった．

今後は，TCB 値設定機構を用いた実験を継続する．さらに，実験の結果から，適切な設定値を導出するアルゴリズムや，パラメータを設定するタイミングを決定するアルゴリズムを検討する．また，リンク特性変化通知機能に関しても実装を進め，NEMO 環境において TCP の通信品質を改善する枠組みの確立を目指す．

参考文献

[1] D. Johnson, C. Perkins, and J. Arkko. *Mobility Support in IPv6*, June 2004. RFC 3775.

[2] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. *Network Mobility (NEMO) Basic Support Protocol*, January 2005. RFC 3963.

[3] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta. Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In *INFOCOM (3)*, pp. 1537–1545, 2000.

[4] K. Tsukamoto, Y. Fukuda, Y. Hori, and Y. Oie. New TCP congestion control schemes for multimodal mobile hosts. *IEICE Trans. on Communications*, Vol. E89-B, No. 6, pp. 1825–1836, June 2006.

[5] 韓閏燮, 寺岡文男. ハンドオーバを考慮した無線 TCP の効率的手法の設計と実装. マルチメディア, 分散, 強調とモバイル (DICOMO2006), pp. 377–380, July 2006.

[6] K. Tsukamoto, Y. Hori, and Y. Oie. Mobility management of transport protocol supporting multiple connections. In *Proc. of Second International Workshop on Mobility Management and Wireless Access Protocols (ACM MobiWac 2004)*, pp. 83–87, October 2004.

[7] Luigi Rizzo. dummynet. `http://info.iet.unipi.it/~luigi/ip_dummynet/`.