

## モバイル IP ハンドオフにおける TCP 通信の高速化手法に関する検討

海老原 成<sup>†</sup>    加藤 聡彦<sup>†</sup>    伊藤 秀一<sup>†</sup>

近年、携帯端末を用いたモバイルコンピューティングが急速に普及している。このような環境では、パケットロスの増加が原因で、TCP 通信において不必要な輻輳制御が多発し、スループットが低下する場合がある。この課題に対して広く研究が行われているが、その多くは伝送誤りに対応するものであり、また端末の移動に伴うハンドオフを対象とした研究例もいくつかはあるが、Mobile IP プロトコルとの協調を考慮していない、端末からのデータ送信についての検討していないなどの問題点が存在する。そこで筆者らは、モバイルインターネットの基本プロトコルである Mobile IP の移動管理手順と関連付け、ネットワーク移動時のハンドオフにおいて発生する連続的なパケットロスに対して、TCP のスループットを低下させない方式を、移動端末がデータを受信している場合と送信している場合の双方向について検討している。本稿ではその詳細について報告する。

### High Speed TCP Communication Procedure for Mobile IP Handoff

Joe Ebihara<sup>†</sup>    Toshihiko Kato<sup>†</sup>    Shuich Itoh<sup>†</sup>

Recently, the mobile communication environment is widely spread. In such an environment, it is pointed that the packet losses caused by transmission errors in wireless links and hand off during terminal moving degrade the TCP communication throughput. There are a lot of researches for this problem, but most of them focus on the packet losses caused by transmission errors, and those for hand off do not take account of the coordination with Mobile IP procedures. In this paper, we propose a high speed TCP communication procedure dealing with Mobile IP hand off related packet losses. We proposes two procedures, one of which is dealing with the data transfer from fixed hosts to mobile terminals, and the other is dealing with that from mobile terminals to fixed hosts. The former extends the M-TCP procedures to the Mobile IP environment. The latter introduces the procedures in which the ineffective congestion control will be cancelled after the hand off procedure. This paper describes the detail of our procedures.

#### 1. はじめに

近年、携帯端末を用いたモバイルコンピューティングが急速に普及している。このような環境では、無線通信によるビット誤りの増加や、端末の移動に伴うハンドオフに起因する瞬断により、パケットロスが頻繁に発生する場合が

ある。パケットロスが多い環境で TCP 通信を行った場合、不必要な輻輳制御が多発し、スループットが低下するという問題点が指摘され、この問題を解決するために多くの研究が行われている[1]。しかし、これらの研究の多くは、ランダムまたはバーストの伝送誤りが生じた場合の対応に主眼が置かれ、ハンドオフに伴う瞬断に対応しているものは少ない。ハンドオフ対応の方式の代表例としては、M-TCP[2]が挙

---

<sup>†</sup> 電気通信大学 大学院 情報システム学研究科  
<sup>†</sup> University of Electro-Communications

げられるが、この方式は、固定ホストから移動端末へのデータ転送しか対象としていない、移動管理は下位のレイヤが行い TCP 高速化の仕組みとは独立であると仮定しているなどの課題がある。

モバイルインターネットの基本プロトコルである Mobile IP[3]では、移動端末 (MN: Mobile Node)が移動した場合は、移動先ネットワークの FA (Foreign Agent)が MN を収容し、MN の HA (Home Agent)と協調して連続的な通信を行おうとする。このため、このような環境での TCP 高速化を実現する場合は、FA や HA で使用されている Mobile IP の手順と関連付けた方式を用いる必要がある。また、端末の移動に伴うハンドオフによる瞬断に対して、移動端末側からデータ送信を対象とした TCP 高速化の研究はほとんどない。

そこで筆者らは、ネットワーク移動時の Mobile IP ハンドオフ手順において発生する瞬断によるパケットロスに対して、TCP のスループットを低下させない方式を、移動端末がデータを受信している場合と送信している場合の双方について検討している[4]。本稿ではその詳細について述べる。以下本論文は以下のように構成される。2章で本方式の基本アプローチを示し、3章と4章で、それぞれ、移動端末のデータ受信方向とデータ送信方向に対する TCP 高速化の詳細手順を示す。

## 2. アプローチ

筆者らは、TCP 高速化を実現するために以下のようなアプローチを用いた。

(1) Mobile IP では FA から定期的な送信される Agent Advertisement により、MN がネットワークの移動を検知する。さらに HA の登録が終了しないとパケットの送受信が正しく行われなない。このため、実際に移動した時点と、移動後新たな FA との間でパケットの送受信を開始する時点とにずれが生じ、その間に通信相手 (CN: Correspondent Node)に送信または CN が MN に送信したパケットは紛失する。本稿ではこのパケットロスにより起動される輻輳制御を原因とするスループット低下を防ぐことを目的とする。

(2) MN、FA、HA という Mobile IP をサポートするノードに対しては変更を行うが、一般のノードである CN の変更は行わないこととする。また、MN がデータを受信する方向と、データを送信する方向について、基本的には独立な方式を採用する。

(3) データ受信の方向については、Mobile IP のハンドオフ中に CN が送信したパケットが紛失し輻輳制御が起動されるのを防ぐ必要がある。このためにはハンドオフ中に CN のパケット送出を停止させ、ハンドオフが終了した後に再開させる必要がある。このために本稿では M-TCP を基本とする。以下に、M-TCP の概要を示す。

- M-TCP では、通常、基地局が MN からの ACK を送信元を送る際に、確認応答シーケンス番号を 1 バイト分だけ少なくして転送する。MN に転送したデータに対して応答がないと、基地局は MN が移動したと推定し、保留しておいた 1 バイトの ACK を、ウィンドウサイズを 0 として送信元を送る。これにより送信元はウィンドウが閉じた状態となり、再送タイムアウトを起こさない。
- CN は MN からウィンドウサイズ 0 を広告された後、パーシストタイマを用いて定期的にウィンドウが開くのを検査する 1 バイトのデータを MN に送る。ハンドオフが完了すると MN はそれに対してウィンドウ更新の ACK を送信する。
- データ通信が正常に終了すると、基地局は保留していた 1 バイト分の ACK を、CN がタイムアウトを起こす前に返す。

上記の M-TCP に Mobile IP 手順を追加し、ハンドオフ後の速やかなデータ転送の実現を目指すこととする。

(4) データ送信の方向については、有効な方式は提案されていない。M-TCP に類似した方式を用いるとしても、MN が移動を検出した後にウィンドウを閉じるなどの対応ができるのみで、ハンドオフ中にタイムアウトが発生した場合は、輻輳制御が起動されてしまう。このため、この方向については、ハンドオフ中に起動された不必要な輻輳制御を、ハンドオフ終了後取り

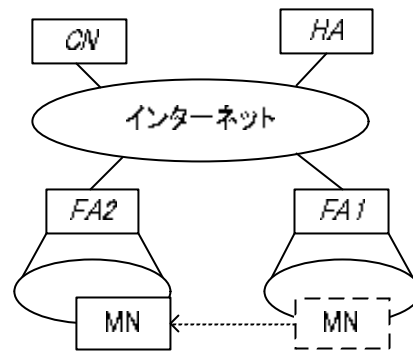
消すというアプローチを用いる。

### 3. データ受信方向の高速化手順

#### 3.1 設計方針

M-TCP を Mobile IP に対応させるために、以下のような設計方針を立てた。

- (1) M-TCP における 1 バイト ACK の保留機能や、移動検出時のウィンドウ閉鎖機能などは、FA に実現させる。
- (2) MN が移動した場合は、新たな FA が、これまでの旧 FA における処理状況 (1 バイト ACK の保留を行ったかどうか、そのシーケンス番号は何かなど) を知る必要がある。このため、MN の移動後に HA が旧 FA から必要な情報を検索し、新たな FA に通知するという方法を用いる。この処理は MN が FA/HA と Registration Request および Registration Reply を交換する手順に連携させることとする。
- (3) FA が MN からの ACK を一定時間受信せず MN が移動したと判断し、CN にウィンドウを閉じる ACK を送信した後に、MN からのパケットを受信した場合は、CN に対してウィンドウを開く ACK を送信する。



- (4) M-TCP ではハンドオフ後のデータ転送再開にパーストタイマによるウィンドウ検査を用いている。しかし持続タイマの値は数秒程度のオーダーであるため、ハンドオフの直後にデータ転送が再開できないという問題がある。そこで、Mobile IP のハンドオフの手順が終了後、HA がウィンドウを開かせる ACK を CN に送信する手順を追加する。

#### 3.2 シーケンス例

次に具体的なシーケンス例を用いて、提案手法の概要を示す。図 1 のようにホームエージェ

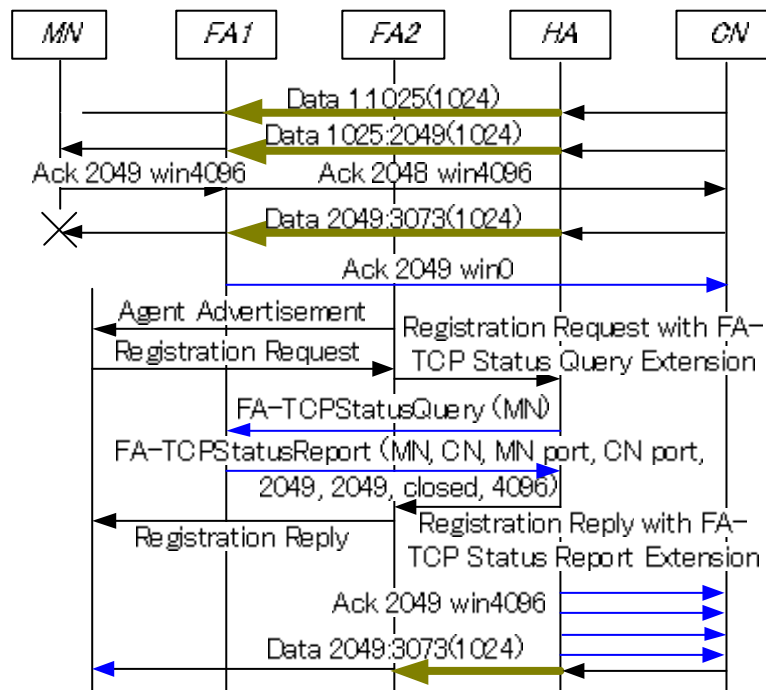


図 2 MN のデータ受信方向の手順

ント *HA* に属する移動端末 *MN* が、*CN* からのデータを受信中に *FA1* から *FA2* のネットワークに移動した場合を想定する。その手順例を図 2 に示す。

最初に *MN* は *CN* から 1024 バイトのデータを 2 つ受信している。図の「1:1025(1024)」はシーケンス番号 1 の 1024 バイトのデータであることを示す[5]。また *HA* から *FA1* までの太い矢印は、カプセル化して転送されていることを示す。これに対し *MN* はシーケンス番号 2048 までのデータの受信と、ウィンドウサイズ 4096 を示す ACK を返す。*FA1* はこれを取り込むと確認応答シーケンス番号を 1 だけ減らして 2047 までのデータを受信したとして *CN* に転送する。

次に *MN* が移動し、その後にシーケンス番号 2049、長さ 1024 バイトのデータを受信すると、*FA1* はその応答が返らないことで、*MN* が移動したと推定する。すると、2048 までの受信を示す ACK (Ack 2049) をウィンドウ 0 で送信し、*CN* をパーシストモードにする。

一方 *MN* は、*FA2* からの Agent Advertisement により移動を検知し *HA* に登録を行う。このとき、*FA2* は新たに移動してきた *MN* に対する TCP の状態の問い合わせを要求する Extension (FA-TCP Status Query Extension) を Registration Request に追加して *HA* に転送する。

*HA* はこれを受信して *MN* が移動前に制御を属していた *FA1* に対して、問い合わせ要求 (FA-TCPStatusQuery メッセージ) を送出する。*FA1* はこれに対し、対応する移動端末の TCP 通信の状態を FA-TCPStatusReport メッセージにより通知する。TCP 通信の状態は以下のものを含む。

- 通信相手の IP アドレス
- 移動端末と通信相手のポート番号
- 移動端末から受信した ACK 番号(2049)
- 通信相手に通知した ACK 番号(2049)
- ウィンドウを閉じたかどうかのフラグ
- 移動端末から受信したウィンドウサイズ (4096)

*HA* はこれらの情報を Registration Reply メッセージの FA-Status Report Extension により *FA2* に通知する。

さらに *HA* は、*MN* が正常にハンドオフを終了したことを知り、閉じていた TCP のウィンドウを開き、紛失したパケットの再送を要求するために、送信元アドレスを *MN* とした、ACK 番号 2049、ウィンドウサイズ 4096 の ACK パケットを 4 つ *CN* あてに送信する。これによりハンドオフ終了直後に、紛失したパケットの再送に引き続いて、次のデータ転送を再開させることができる。

### 3.3 詳細手順の検討

次に本方式を実現するに当たり、課題となる項目に対する詳細手順の検討結果を示す。

#### 3.3.1 複数の FA 間の同期について

M-TCP では移動端末の移動管理は M-TCP の下位で行われ、移動端末からの ACK パケットは必ず、Supervisor Host と呼ばれる特定の M-TCP ノードを経由することが想定されている。しかし本方式では、*MN* の移動後は新たな *FA* が M-TCP の処理を行うため、ハンドオフに当たって、移動前の *FA* と移動後の *FA* が整合性の取れない TCP 手順を実行する可能性がある。例えば、移動後の *FA* 経由で *MN* がウィンドウを開く ACK を送出したにもかかわらず、その後で移動前の *FA* がウィンドウを閉じる ACK を送出する状況や、移動前の *FA* が保留した 1 バイト ACK を送出したにもかかわらず、移動後の *FA* で 1 バイトを保留した ACK パケットを送出する状況などが考えられる。

本方式ではこれらの *FA* 間の同期ずれの問題に対して以下のように対応する。

- *FA* は *HA* から FA-TCPStatusQuery メッセージを受信し、要求された情報を FA-TCPStatusReport メッセージで返答した後は、対応する *MN* に対して M-TCP の処理を行わない。さらに対応する *MN* の情報を Visitor List から消去する。
- FA-TCPStatusQuery メッセージを受信した時点で、指定された *MN* の情報が Visitor List に存在しない場合が考えられる。これは問い合わせの前に Visitor List の生存時間が終了した場合などに発生する。その場合、*FA* は情報が存在しないとして、FA-TCPStatusReport メッセージを返答す

る。新しい FA は対応する MN の情報は設けずに処理を行うが、その場合は MN からの最初の ACK に対しては 1 バイト分を保留した ACK を送信しない。

### 3.3.2 CN に対する ACK の送信について

M-TCP においては、CN からの最後にデータが送出されている場合、すなわち CN から次のデータが送信されない場合は、CN に対して保留した 1 バイト ACK を送信する必要がある。これは、M-TCP ノードにおいて、CN との間の RTT を推測し、CN においてタイムアウトが発生する前に 1 バイト ACK を送信することにより行われる。

前述のように本方式では、MN の所属する FA が M-TCP ノードの働きをするため、ハンドオフ時に CN においてタイムアウトが発生しないようにする必要がある。このため本方式では、FA が FA-TCPStatusQuery メッセージを受信した時点で、CN からのデータのすべてに対して、MN が ACK パケットを送出しており、さらに CN に対しては 1 バイトを保留した ACK しか送信していないような TCP コネクションに対して、保留した 1 バイトを応答する ACK を送信することとする。なおこの ACK のウィンドウサイズは M-TCP と同様に 0 とはせず、MN から通知されたウィンドウサイズの値を設定するものとする。

### 3.3.3 FA と HA の管理用データ構造について

FA は通常の Mobile IP で用いられる Visitor List に追加して、MN が確立している TCP コネクションごとに次のような情報を有する。

- CN の IP アドレス
- MN と CN のポート番号
- MN に送信したデータの最大シーケンス番号
- CN から受信した最大の ACK 番号
- MN に送信した最大の ACK 番号
- MN から通知されたウィンドウサイズ
- CN に対してウィンドウを閉じたかどうかのフラグ
- MN-FA 間の RTT の推定値

- MN へのタイムアウト時間
- FA-CN 間の RTT の推定値
- CN へのタイムアウト時間

一方、HA は MN の TCP について永続的に情報を管理することはしない。HA は、MN から FA 経由で Registration Request を受信すると、その MN をこれまで管理していた旧 FA に対して FA-TCPStatusQuery メッセージを送出し、FA-TCPStatusReport メッセージを待ち、その情報を新たな FA に通知する。またウィンドウが閉じられている場合は、通信中の CN に対してウィンドウ更新と重複 ACK を送信する。

## 4. データ送信方向の高速化手順

### 4.1 設計方針

Mobile IP では、新たな FA からの Agent Advertisement により移動を検出し、Registration Request と Registration Reply を交換した後に、MN のルーティングテーブルを変更する。従って、Registration Reply を受信するまでは、移動前の FA をデフォルトルータとして、IP パケットを送出し続ける。このため、実際の移動後からハンドオフ処理終了までは、送信したパケットは紛失することになる。すなわち、Mobile IP の手順だけでは、移動前のネットワークを離れたことなどをすばやく検出することは困難である。このため、ネットワーク層の機能のみでは、ハンドオフ中に送信データに対するタイムアウトが発生することは避けられないと思われる。そこで、MN からのデータ送信方向での TCP 高速化に対して、以下のような設計方針を立てた。

- (1) ハンドオフ中のタイムアウトが原因で生じた輻輳制御は、不必要なものであるため、これを、ハンドオフ終了後に取り消すというアプローチを基本とする。
- (2) 不必要な輻輳制御を取り消すために、MN は TCP の再送タイマを起動するたびに、その直後の輻輳制御パラメータを記録する。そして、Registration Reply の受信によりハンドオフの完了を確認すると、ハンドオフ直前に送信したと考えられるデータ送信直後の状態に戻し、さらにハンドオフ直前に送信したデータがタイムアウトが生じていれば、タイムアウト再送を

起動する。

(3) MN が送信したデータがハンドオフ中に無駄に転送されたかどうかを判断するために、FA に対して、管理している MN ごとに、送信されたデータのシーケンス番号とデータ長を記録させる。MN は新たな FA の下に移動すると、移動前の FA に対して自分が送信したデータの記録を問い合わせ、その情報と自分が記録した輻輳制御パラメータの値とをつき合わせ、取り消すべきデータ転送の動作(戻るべきデータ転送)を決定する。

(4) FA が保持している情報を MN に伝えるためには、MN の移動後に HA が旧 FA から必要な情報を検索し、新たな FA を通じ、MN に通知するという方法を用いる。この処理は MN が FA/HA と Registration Request および Registration Reply を交換する手順に連携させることとする。

## 4.2 シーケンス例

具体的なシーケンス例を用いて提案手順の概要を示す。想定する状況は 3.2 と同様に、移動端末 MN が CN にデータを送信しながら、FAI から FA2 のネットワークに移動した場合である。手順例を図 3 に示す。

MN は 1024 バイトのデータを 3 つ(図中、  
、  
 )送信した後移動する。この内 は に  
より確認応答され、 は輻輳により紛失する。  
移動の時点で FAI はシーケンス番号 1025 から  
1024 バイト、2049 から 1024 バイトを 1 回ずつ  
受信したことを記録する。

移動後 と でタイムアウトが発生したデータが送信されるが、これは移動前の FA である FAI をネクストホップとして送られるため紛失する。MN は、4.1 で示したように、  
と の直後の輻輳パラメータの値および の  
データ送信がタイムアウトしたことを記録している。

MN は FA2 からの Agent Advertisement に対応して送られる Registration Request により、HA に対して移動前の FA が保持している TCP の状況の照会を要求する MN-Data Status Query Extension を送出する。この情報には通信中の CN の IP アドレス、MN と CN のポート番号、

タイムアウトを起こした TCP セグメントのシーケンス番号 (1025)が設定される。

この情報は FA2 を経由して HA まで転送される。HA はこれを受信すると、MN が移動前に属していた FAI に対して、問い合わせ要求 (MN-TCPStatusQuery メッセージ)を送出する。このメッセージは MN-TCP Status Query Extension と同様の値を含んでいる。FAI はこのメッセージを受信すると、移動端末 MN に対する記録を検索し、シーケンス番号 1025 を含む TCP セグメントを 1 回受信していることおよびその端末からの転送されたセグメントの最大シーケンス番号+1 の値が 3073 であるという情報を得る。この情報を MN-TCPStatusReport メッセージにより HA に通知する。HA はこれらの情報を Registration Reply メッセージの MN-TCP Status Report Extension により FA2 を通じ、MN に通知する。

その情報を受け取ると、MN は、タイムアウトを発生したシーケンス番号 1025 を含むセグメントを FAI が 1 回受信していることおよび最大 3072 のバイトまで FAI が受信していることを知る。そこで MN は以下のように自身の状態を設定しなおす。

- まず、自身の輻輳制御のパラメータ(cwnd / ssthresh)を、1025 のセグメントを送信した直後の値に設定する。
- 次に送信制御のパラメータ (snd\_max / snd\_nxt)の値を 3073 に設定する。
- さらに、MN は最初に送信した 1025 のセグメントに対してタイムアウトを生じているため、1025 のセグメントに対してタイムアウト再送を行い( )、輻輳制御を 1 回起動する。

以上のようにして、ハンドオフに起因する輻輳制御を取り消し、データ転送を再開させることができる。

## 4.3 詳細手順の検討

### 4.3.1 制御用データ構造

MN は本方式を実装するために、再送タイマをセット(スタート)するごとに、以下の情報を記録する。

- 再送タイマに対応するシーケンス番号

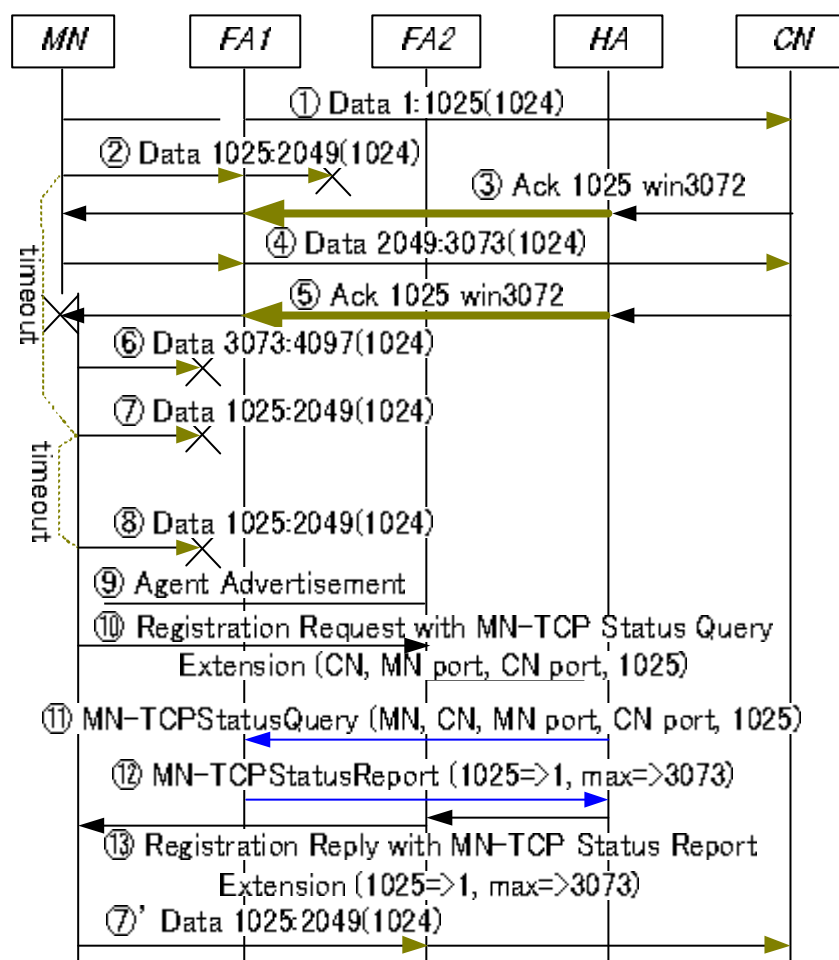


図3 MNのデータ受信方向の手順

- 送出されるセグメントのデータ長
- cwnd
- ssthresh

再送タイマをセットするタイミングは、新たにデータを送信する場合、タイムアウト後にデータを再送する場合、一部の受信応答を行うACKを受信した場合である。上記のシーケンス番号とは、前者2つの場合は、送出するデータのシーケンス番号となり、また最後の場合は、ACKされていない最も古いデータのシーケンス番号となる。また、この情報は対応するシーケンス番号の受信応答を受信した時点で解放する。TCPでは、各時点では高々1つの再送タイマが動作しているため、ここで記録する情報も高々1つのセグメントに対してのみとなる。

一方FAは、管理するMNのビジタリストの中に、以下の情報を保持する。

- TCP通信をするCNのIPアドレス
- MNとCNのポート番号
- MNから受信されたデータセグメントのシーケンス番号とデータ長およびそのセグメントに対してCNからACKを受信したかどうかのフラグ

同一のシーケンス番号を含むデータセグメントが、MNから複数送信された場合は、個別に記録することとする。これはTCPによる再パケット化に対応するためである。また、この情報はCNからACKを受信した後、一定時間たった後に解放される。ACKを受信したかどうかのフラグはこのために用いられる。

#### 4.3.2 処理アルゴリズム

4.2節で示したように、MNは新たなFAの下に移動すると、HAへのRegistration Requestを

送信する時点で、確立している TCP コネクションのすべてに対して、コネクションの識別情報と、再送タイマを起動しているシーケンス番号 (ACK を受信していない最も古いシーケンス番号に相当する) を通知する。なお、送信したセグメントがすべて受信応答されている場合はこのシーケンス番号の通知は行わない。

次に、HA から MN-TCPStatusQuery メッセージを受信した FA は、対応する MN のビジタリスト・エントリから、指定されたシーケンス番号を含むセグメントを受信した回数、その MN から受信したデータの最大のシーケンス番号を検索し、HA に MN-TCPStatusReport メッセージを返答する。FA はその後、MN に対するビジタリスト・エントリを解放する。

MN は Registration Reply の MN-TCP Status Report Extension の情報を用いて以下の処理を行う。

- MN の送信制御用パラメータ(snd\_max と snd\_nxt)を通知された最大シーケンス番号 +1 の値に設定する。
- 再送タイマを起動したシーケンス番号を通知した場合は、受信回数に応じて、そのセグメントを最後に送出した直後の cwnd と ssthresh の値を用いて、これらのパラメータを設定しなおす。ここで、受信回数が 0 の場合は、そのセグメントを送出された直後の値を用いて設定しなおす。セグメントの送信においては cwnd と ssthresh の値が変更されないため、この処理で十分であると考えられる。
- 対応するセグメントのタイムアウトが生じている場合は、再送を行い、続いてタイムアウト後の輻輳制御を行う。タイムアウトが生じていない場合は、再送タイマをセットして、現在の送信パラメータの値からデータ転送を再開する。

## 5. おわりに

本稿では、モバイル IP のハンドオフにおけるパケットロスによる不必要な輻輳制御に起因する TCP のスループット低下を回避する方法について述べた。MN がデータを受信する方向については、代表的な既存方式である

M-TCP をモバイル IP に適応する方法を提案した。一方、MN がデータを送信する方向については、MN において輻輳制御のパラメータを記録し、ハンドオフ終了後に移動直前の状態に戻す方法を提案した。さらにこれらの 2 つの方式について、Mobile IP の登録手順の中で、MN/新旧の FA/HA が協調して必要な情報を交換するように設計を行った。今後、本方式の評価を進める予定である。

## 参考文献

- [1]: N. Vaidya, "TCP for Wireless and Mobile Hosts," Tutorial at MobiCom'99, available at <http://www.crhc.uiuc.edu/~nhv/presentations.html>.
- [2]: K. Brown, et al., "M-TCP: TCP for Mobile Cellular Network," ACM Computer Communications Review, Vol.27, No.5, 1997.
- [3]: C. Perkins, Ed., "IP Mobility Support for IPv4," RFC3220, Jan. 2002.
- [4]: 海老原他, "モバイル IP におけるハンドオフ時の TCP 通信高速化に関する検討," 情報第 65 回全大, 2H-5, Mar. 2003.
- [5]: W. R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," Addison-Wesley, 1994.