

# Mobile IP SHAKE における TCP の輻輳ウィンドウ縮小回避方法

荻野 秀岳<sup>†</sup> 石原 進<sup>††</sup>

<sup>†</sup> 静岡大学大学院工学研究科 〒432-8561 浜松市中区城北 3-5-1  
<sup>††</sup> 静岡大学創造科学技術大学院 〒432-8561 浜松市中区城北 3-5-1  
E-mail: †{ogino,ishihara}@ishilab.net

あらまし 筆者らは、端末の移動透過性を保障する Mobile IP 環境において、複数の移動端末が短距離高速リンクでローカルネットワークを構築し、ローカルネットワーク内の各端末の長距離低速リンクを同時に利用することで高速かつ信頼性のある通信を実現する Mobile IP SHAKE (SHAring multipath procedure for a cluster network Environment) を提案している。本稿では、Mobile IP SHAKE における TCP 通信において、利用する各端末の長距離低速リンクの性質がそれぞれ異なることで、セグメントの順序が逆転して受信端末に到着し、不要な重複 ACK の発生に伴う無駄な再送が頻繁に発生するという問題を挙げ、その対策を検討する。Mobile IP SHAKE において端末の移動情報を管理する Home Agent (HA) では、各経路ごとに中継するセグメントのシーケンス番号を管理することでセグメントの順序逆転を検知し、受信端末から送信される不要な重複 ACK を破棄する。送信端末では、HA がデータセグメント転送時に送信端末へと送信する擬似 ACK から HA 間 RTT を測定し、送信バッファでの待ち時間と HA 間 RTT から HA にセグメントが最も早く到着すると予想される経路の選択を行う。

キーワード Mobile IP, モバイルネットワーク, 複数経路, TCP

## A scheme for avoiding frequent reduncion of TCP congestion window on Mobile IP SHAKE

Hidetake OGINO<sup>†</sup> and Susumu ISHIHARA<sup>††</sup>

<sup>†</sup> Graduate school of Engineering, Shizuoka University  
Johoku 3-5-1, Naka-ku, Hamamatsu-shi, Shizuoka, 432-8561 Japan  
<sup>††</sup> Graduate school of Science and Technology, Shizuoka University  
E-mail: †{ogino,ishihara}@ishilab.net

**Abstract** We have developed a system that increases the speed and reliability of communication of mobile computers, called Mobile IP SHAKE (SHAring multipath procedure for a cluster network Environment). It lets mobile node construct a temporary network with neighboring mobile nodes through local high-speed links in order to communicate with a host outside of the local network. In Mobile IP SHAKE, a mobile node uses both its own link and the links of other hosts in the local network simultaneously and disperses the traffic to multiple links between the local network and the external host. Because of the different delay of links between the external nodes and the nodes in the local network, arrival of the packets at the receiver tends to be out of order. This paper addresses a problem that flequent useless retransmission due to unnecessary duplicate TCP ACKs caused by arrival of out of order segments. We propose a scheme to solve this problem. In this scheme, a Home Agent(HA) of mobile IP deletes unnecessary duplicate ACKs by managing a sequemce number of segments transmitted on each links and a mobile node selects a link that can transmit a segment earliest to HA by using round trip time between HA and the sending node.

**Key words** Mobile IP, mobile network, multipath, TCP

## 1. はじめに

近年、携帯通信端末の普及やインターネット接続環境の多様化に伴い、いつでもどこでもインターネットへの接続が可能となりつつある。また、複数のネットワークインタフェースを搭載した携帯通信端末が広く普及しており、ユーザはそれらのネットワークインタフェースをインターネット接続環境に応じて使い分けることができる。しかし、現在の無線通信では、無線 LAN などの短距離無線通信であれば比較的高速な通信が可能であるが、比較的低速な広域無線サービスを利用せざるを得ない状況が存在する。さらに無線通信では、障害物などによって通信が遮断され、接続の安定性に乏しいといった問題もある。

そこで筆者らは、無線通信環境において高速かつ接続の安定した通信を実現する手法として通信回線共有方式 (SHAKE: SHARING multipath procedure for a cluster network Environment) を提案している。SHAKE では、複数のインタフェースを搭載した移動端末が近接する移動端末と短距離無線リンクを用いて一時的にローカルネットワーク (Alliance) を構築する。Alliance 内の各移動端末の長距離低速リンクを同時に利用し、各リンクにトラフィックを分散させ、ネットワーク資源を有効に活用することで高速かつ接続の安定した通信を実現する。SHAKE の実現方法として、端末の移動透過性を保障する Mobile IP を SHAKE に応用することで、移動透過性を保障しながら SHAKE を用いた通信を実現する Mobile IP SHAKE が提案・実装されている [1]。

Mobile IP SHAKE 環境での TCP 通信では、Alliance 内の移動端末が持つ性質が必ずしも同じでない長距離無線リンクを同時に利用する可能性がある。そのため各経路の状態によってセグメントの順序が逆転して受信端末に到着してしまい、不要な重複 ACK の発生に伴う無駄な再送が頻発することで TCP の輻輳ウィンドウを増加させることが困難であるため、高いスループットを実現することができない。これは上りと下りの両方向の通信における問題である。

本稿では、Mobile IP SHAKE の Alliance から Alliance 外部への上り通信において、パケットの到着順序逆転に伴う TCP の輻輳ウィンドウの縮小を回避する方法を検討する。Mobile IP SHAKE でトラフィックの分配に使用する Home Agent (HA) では、複数の端末の位置情報の管理とトラフィックの分配に加えて複数の TCP 通信の負荷が集中してしまうために、転送するセグメントのバッファリングや欠落セグメントの再送処理といった処理の重い機能の追加は望ましくなく、軽量な機能の追加にとどめることが望ましい。そこで HA は、各経路ごとに転送するセグメントを監視することでセグメントの順序逆転を検知し、パケットロスでなくセグメントの順序逆転によって発生する不要な重複 ACK を破棄する。送信端末では、HA がセグメント転送時に送信端末へと送信する擬似 ACK を用いて HA と送信端末間の RTT を測定し、セグメントが最も早く HA へ到着する送信経路の選択を行う。

複数経路を用いた TCP 通信に関する研究が広く行われている。文献 [5] では、各経路のボトルネック部の帯域幅を一定時

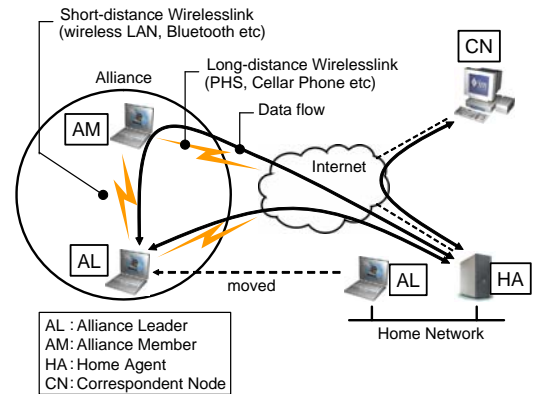


図 1 Mobile IP SHAKE

間内の確認応答から算出し、各経路の輻輳ウィンドウがボトルネック部の帯域幅より大きくなった時点で輻輳回避段階へと移行することで輻輳制御を行っている。文献 [6] では、一定時間内の確認応答から各経路の帯域幅と RTT を算出し、各経路ごとに適したスロースタート初期値を設定することで輻輳制御を行っている。これらの方法では、両エンドの TCP が輻輳経路に対応している必要がある。一方、本稿は Mobile IP SHAKE 環境での TCP 通信の効率化を対象とし、両エンドの TCP が複数経路を用いた通信に対応した拡張がされていることを前提としない。

## 2. Mobile IP SHAKE

### 2.1 Mobile IP SHAKE の概要

Mobile IP SHAKE (図 1) は、Mobile IP において端末の移動情報を管理する HA および移動端末の IP 層に、トラフィックの分配機構を追加することで実現される。HA は、Alliance 内の端末が持つ外部リンクを利用して実際に通信を行う Alliance Leader (AL) のホームアドレス (HoA) および AL の気付アドレス (CoA) の対応を保持し、HoA 宛のパケットを CoA へと転送することで AL が別のネットワークに移動しても継続して通信を行うことができる。Mobile IP SHAKE では、AL に通信帯域を提供する Alliance 内の端末である Alliance Member (AM) の CoA あるいは HoA を AL が HA に通知し、HA が AL の HoA および CoA と AM の CoA の対応を保持する。この対応に基づいて HA は、通信相手 (CN: Correspondent Node) が AL に送信するパケットを HA と AL および AM 間を結ぶ複数の経路へと分配することで、端末の移動透過性を保障しながら複数経路を同時に利用した通信を行うことができる。また、AL が CN へパケットを送信する場合、AL は自身の外部リンクおよび AM を介した経路へとパケットを送信する。

### 2.2 Mobile IP SHAKE における TCP 通信の問題

Mobile IP SHAKE では、複数の通信経路を同時に利用することで高速かつ接続が安定した通信を実現する。しかし、AL および AM の外部リンクの通信帯域や遅延がそれぞれ異なり、変動するためにパケットの送信経路の選択に適切な手法を用いたとしても、順序が逆転して受信相手に到着してしまうことが

避けられない。

TCP 通信では、セグメントの順序が逆転して受信端末に到着してしまうと、先に送信されたセグメントが送信経路の遅延により遅れて転送されている状況があるにもかかわらず不要な重複 ACK が発生する。これにより、欠落していないセグメントを再送しなければならない、再送とともに TCP の輻輳ウィンドウが縮小されてしまう。受信端末の TCP が複数経路を利用した通信に対応したものであれば、ACK の返信経路を 1 本に限定することで使用する各経路の片側 RTT を測定し、送信経路の選択を行ったり [2] [3]、Delayed ACK の使用によってこのような問題を未然に防止することがある程度可能だが、Mobile IP SHAKE 環境では、AL の通信相手として通常の TCP を備えた任意の端末を想定する。したがって、通信相手の TCP が不要な重複 ACK を送信することを抑制できない。さらに、複数経路に対応した TCP を備えた AL においても、通信相手が送信する重複 ACK が不要な重複 ACK であるかどうかを判別することができない。また、送信端末と受信端末の双方が複数経路に対応した TCP を備えていたとしても、常に性質が変動する無線通信環境では完全にセグメントの順序逆転を防止することができない。

### 3. 輻輳ウィンドウ縮小回避方法

#### 3.1 前提条件

Mobile IP SHAKE は、HA と AL の IP 層に特殊な機能を追加することで実現される。HA と AL において、輻輳ウィンドウ縮小回避の機能の追加を考えた場合、既に IP 層に特殊な機能を追加している AL では、TCP 層への機能の追加は比較的容易である。一方、複数の移動端末の移動情報の管理を行う HA では、複数の TCP 通信の負荷が集中してしまうために、TCP 層への機能の追加は軽量である必要がある。以上の制限を前提に、本稿では AL から CN への上り通信時の HA と AL で動作する輻輳ウィンドウ縮小回避方法を提案する。

HA は、CN へ中継するセグメントの受信経路とシーケンス番号からセグメントの順序逆転またはパケットロスの発生を検知し、セグメントの順序逆転によって発生した重複 ACK を破棄することで無駄な再送を抑制し、輻輳ウィンドウの縮小を回避する。また、HA は AL からセグメントを受信した時点でそのセグメントを受信した経路を用いて擬似 ACK を AL へと送信する。AL は、HA が送信する擬似 ACK から AL-HA 間の RTT を各経路ごとに測定し、各経路の送信バッファでの待ち時間と AL-HA 間 RTT から AL がセグメントを送信してから HA が受信するまでの時間を各経路ごとに算出し、その時間が最も短い経路を送信経路とする。

以下、本稿は Mobile IP SHAKE 環境において複数経路を利用した場合のセグメントの順序逆転の対応策を検討する。通常 HA-CN 間の単一経路においてもセグメントの順序逆転やパケットロスが発生するが、本稿での議論では HA-CN 間ではセグメントの順序逆転およびパケットロスが発生しないことを前提とする。

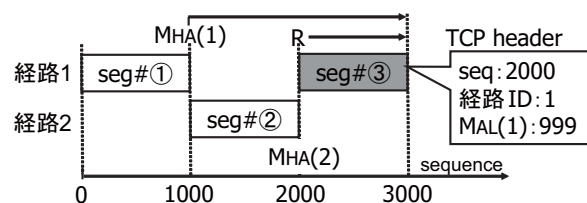


図 2 HA が正しい順序でセグメントを受信した場合の処理

#### 3.2 HA における不要な重複 ACK の破棄

HA での輻輳ウィンドウ縮小回避方法として、セグメントの順序逆転によって発生する不要な重複 ACK を破棄する手法を提案する。HA は、AL から CN へと中継するセグメントの受信経路とシーケンス番号を用いてセグメントの順序逆転またはパケットロスの発生を検知する。HA は、重複 ACK を受信した時点で、その重複 ACK がパケットロスによるものでなく、セグメントの順序逆転によって発生したと判別するとその重複 ACK が不要であるとみなして破棄する。HA は、複数経路を用いた通信におけるセグメントの順序逆転で発生する不要な重複 ACK を検知するため、以下の 6 つの変数を利用する。

- (1)  $M_{AL}(i)$ : 各経路  $i$  で以前に送信された最大シーケンス番号
- (2)  $M_{HA}(i)$ : 経路  $i$  で送信されたデータ内で欠落なく受信した最大シーケンス番号
- (3)  $R$ : AL からセグメントの欠落なく受信したデータの最大シーケンス番号
- (4)  $d$ : 破棄可能な重複 ACK の推定個数
- (5)  $e$ : 破棄した重複 ACK の個数
- (6)  $A$ : CN から受信した最大の ACK 番号

AL は、セグメント送信時にその利用する各経路ごとに管理する変数  $M_{AL}(i)$  と経路の識別子を送信セグメントの TCP ヘッダに付加する。HA は、 $M_{AL}(i)$ 、 $M_{HA}(i)$ 、 $R$ 、 $A$  を用いてセグメントの順序逆転またはパケットロスがどの経路で発生しているかを検知する。以下に、HA でのセグメントと ACK 受信時の処理の詳細を示す。

##### 3.2.1 データセグメント受信時の処理

AL から経路  $i$  を介してデータセグメント (シーケンス番号:  $seq$ , セグメント長:  $L$ ) を受信した HA は、 $seq$  と  $R$  を比較することでそのデータセグメントが正しい順序で受信できたかを判別する。

- $seq = R + 1$  の場合

HA は、正しい順序でデータセグメントを受信しているので、 $M_{HA}(i)$  および  $R$  に  $seq + L - 1$  を代入し、そのセグメントを CN へと転送する。

このときの HA の処理を図 2 に沿って説明する。図 2 は、シーケンス番号が 2000~2999 である  $seg\#3$  を HA が経路 1 から受信した状況を示す。 $seg\#3$  を受信する前には、シーケンス番号が 0~999 である  $seg\#2$  を経路 1 で、シーケンス番号が 1000~1999 である  $seg\#2$  を経路 2 で受信している。HA は、経路 1 で  $seg\#1$  を受信した時点で  $M_{HA}(1)$  と  $R$  (ともに初期値 0) を  $seg\#1$  の最大シーケンス番号である 999 まで更新し、経路 2 で

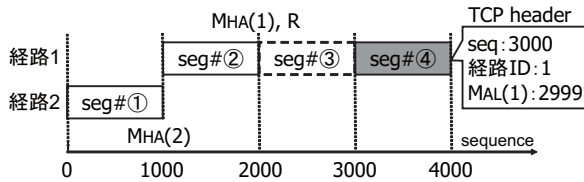


図3 パケットロスが発生した場合の HA の処理

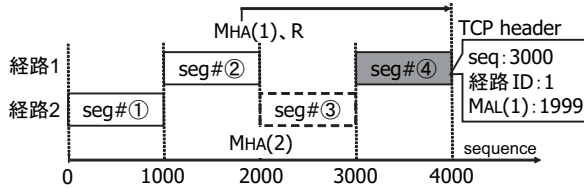


図4 セグメントの順序逆転が発生した場合の HA の処理

seg#2を受信した時点で  $M_{HA}(2)$  と  $R$  を seg#2 の最大シーケンス番号である 1999 まで更新している。そして seg#3 を受信した時点では、seg#3 の最大シーケンス番号である 2999 まで AL から欠落なくデータを受信しているため、 $M_{HA}(1)$  および  $R$  を 2999 まで更新する。

- $seq < R + 1$  の場合

HA が受信したセグメントは、再送セグメントである。HA が CN へと転送したセグメントが HA-CN 間の経路でパケットロスした場合には、 $R + 1$  より小さなシーケンス番号のセグメントが HA に到着する。この場合、HA は受信した再送セグメントを CN へと転送するが  $R$  およびセグメントを受信した経路の  $M_{HA}$  は更新しない。

- $seq > R + 1$  の場合

HA は、セグメントの欠落もしくは順序逆転が発生していることを検知し、受信したセグメントの TCP ヘッダに含まれる  $M_{AL}(i)$  と  $M_{HA}(i)$  を比較することで、セグメントの欠落もしくは順序逆転がどの経路で発生しているかを判別する。 $M_{HA}(i)$

$M_{AL}(i)$  の場合を図 3、 $M_{HA}(i) = M_{AL}(i)$  の場合を図 4 に沿って説明する。

- (a)  $M_{HA}(i) < M_{AL}(i)$  の場合

経路  $i$  でパケットロスが発生した状況である。シーケンス番号が 3000 ~ 3999 である seg#4 を受信した HA では、TCP ヘッダ内の  $M_{AL}(1)$  を参照することで、2999 までのシーケンス番号のセグメントである seg#3 が経路 1 で以前に送信されたことがわかる。同一経路上で seg#3 より先に seg#4 を受信しているため、seg#3 は経路 1 でロスしたことが明確となり、HA は  $M_{HA}(i)$  と  $R$  を更新しないでそのセグメントを CN へと転送する。

- (b)  $M_{HA}(i) = M_{AL}(i)$  の場合

経路  $i$  以外の経路でパケットロスした、もしくは遅延によりセグメントが遅れている状況である。シーケンス番号が 3000 ~ 3999 である seg#4 を受信した HA では、TCP ヘッダ内の  $M_{AL}(1)$  を参照することで、seg#4 を受信する以前に経路 1 ではシーケンス番号が 1999 である seg#2 が送信されたことがわかる。つまりシーケンス番号が 2000 ~ 2999 である seg#3 は、経路 1

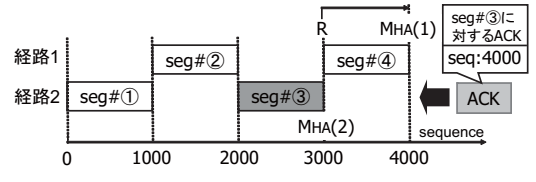


図5 HA での ACK 受信時の処理

以外の経路で送信されている。このとき、seg#3 は経路 2 でパケットロスしているか経路 2 の遅延により HA への到着が遅れている。そこで seg#3 が経路 2 の遅延によって HA への到着が遅れている可能性があることから seg#4 に対する ACK は、ACK 受信時の seg#3 の受信状況によっては、破棄することが可能なので破棄可能な重複 ACK の数として保持する変数 ( $d$ ) を 1 増加させる。そして  $M_{AL}(1)$  に  $seq + L - 1$  を代入し、seg#4 を CN へと転送する。

### 3.2.2 ACK 受信時の処理

CN から ACK (シーケンス番号:  $ACKseq$ ) を受信した HA は、各経路の  $M_{HA}$  の最大値と  $ACKseq$  を比較することで、その ACK が重複 ACK であるかを判別する。

- $ACKseq > A$  の場合

受信した ACK は重複 ACK ではないので、 $A$  に  $ACKseq$  を代入した後、ACK を AL へと転送する。ACK の転送経路は、HA が AL からのセグメント受信時に更新する AL-HA 間の片道 RTT (後述) が最小の経路を用いる。図 5 は、seg#4 受信後に seg#3 を受信し、seg#3 に対する ACK を受信した状況を示している。このとき ACK のシーケンス番号から 3999 までのデータセグメントが CN に到着していることがわかるので  $R$  の値を 3999 まで更新し、その ACK を AL へと転送する。

- $ACKseq < A$  の場合

受信した ACK は重複 ACK である。この場合、 $d$  と  $R$  の値に応じて以下の処理を行う。

- (i)  $d = 0$  の場合

HA において破棄可能な重複 ACK の発生が検知されていないので、その ACK は必要な重複 ACK であると判断し、通常通りその ACK を AL へと転送する。

- (ii)  $d > 0$  かつ  $ACKseq < R$  の場合

HA は、受信した ACK のシーケンス番号に対応するデータセグメントをすでに CN へと転送している。よってこの ACK は破棄可能な重複 ACK であると判別でき、 $d$  の値を 1 つ減少させ、その ACK を破棄する。

図 6 は、HA が seg#4 に対する ACK を受信した時点ですでに seg#3 を CN へと転送している状況を示している。HA は、seg#2 に続いて経路 1 で受信した seg#4 を CN へと転送し、CN から seg#4 に対する ACK ( $ACKseq: 2000$ ) を受信する。このとき HA はすでにこの ACK のシーケンス番号に対するデータセグメントである seg#3 を CN へと転送している。よって HA は、シーケンス番号が 3999 までのデータセグメントを欠落なく CN へと中継しているため、受信した ACK を無駄な再送を引き起こす重複 ACK であると判別し、破棄する。

図 7 は、図 6 の状況が発生した場合のシーケンス図の一例を

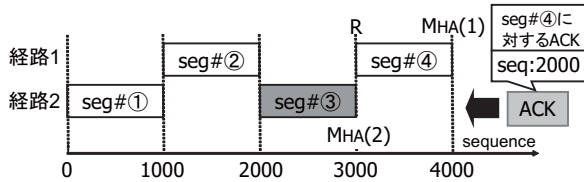


図 6 重複 ACK 受信時に対応するデータセグメントが転送済みの場合

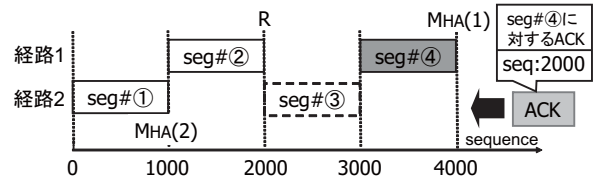


図 8 重複 ACK 受信時に対応するデータセグメントが未転送の場合

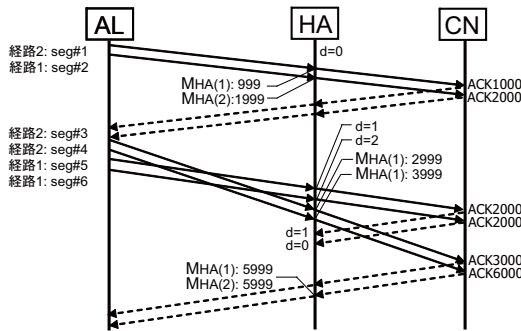


図 7 重複 ACK 受信時に対応するデータセグメントが転送済みの場合のシーケンス図

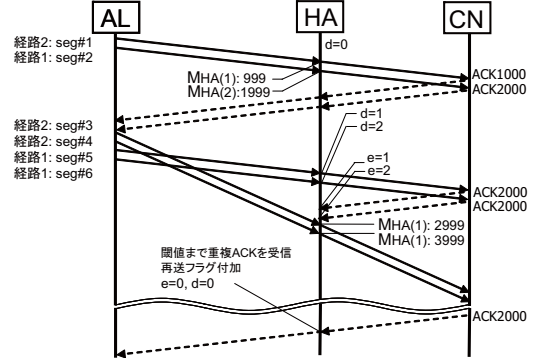


図 9 重複 ACK 受信時に対応するデータセグメントが未転送の場合のシーケンス図

示している。HA は、経路 1 から seg#5 と seg#6 を受信した時点で、経路 2 から先に送信されている seg#3 と seg#4 を転送していない。このとき HA は、seg#5 と seg#6 の TCP ヘッダに含まれる  $M_{AL}(1)$  から seg#3 と seg#4 が経路 1 の遅延によって遅れている可能性があることを検知し、 $d$  の値を増加させる。CN が seg#5 と seg#6 を受信することで発生する重複 ACK を HA が受信した時点で、seg#3 と seg#4 をすでに CN へと転送しているのでそれらの重複 ACK を破棄するとともに  $d$  の値を減少させる。こうすることで、seg#3 と seg#4 が経路の遅延によって遅れた場合であっても HA が不要な重複 ACK を破棄することによって無駄な再送ならびに TCP の輻輳ウィンドウの縮小を回避することができる。

(iii)  $d > 0$  かつ  $ACK_{seq} = R$  の場合

HA は、受信した ACK のシーケンス番号に対応するデータセグメントをまだ CN へと転送していない。このとき HA は、未転送のセグメントがパケットロスの発生または遅延のどちらが原因で HA に到着していないのか判断することができない。そこで、遅延によるセグメントの順序逆転に対応するため受信した重複 ACK を破棄するとともにこの処理で破棄した重複 ACK の数 ( $e$ ) を 1 つ増加する。閾値以上の個数の重複 ACK を受信した時点で、その重複 ACK を AL へと転送し、 $d$  および  $e$  を 0 に初期化する。このとき、AL へ転送する重複 ACK には、AL で直ちに再送処理へと移行することを指示する「再送フラグ」を付加する。この再送フラグが付加された重複 ACK を受信した AL は、その重複 ACK で確認応答されているデータセグメント部を直ちに再送する。破棄した重複 ACK の数 ( $e$ ) の閾値は、通常 TCP において再送処理が行われる 3 より大きく、タイムアウトによる再送が発生しない値である必要がある。

図 8 は、seg#4 に対応する ACK を受信した時点で、HA がまだ seg#3 を CN へと転送していない状況を示している。HA は、seg#2 に続いて経路 1 で受信した seg#4 を CN へと転送

し、CN から seg#4 に対する ACK (ACKseq:2000) を受信する。このとき HA はまだ seg#3 を CN へと転送していない。HA では seg#3 の未転送がパケットロスの発生または遅延のどちらが原因で HA に到着していないのか判断できない。そこで、遅延によるセグメントの順序逆転である可能性もあるとみなして、ACK を破棄するとともに  $e$  の値を 1 つ増加させる。

図 9 は、図 8 の状況が発生した場合のシーケンス図の一例を示している。HA は、経路 1 から seg#5 と seg#6 を受信した時点で、seg#3 と seg#4 が経路 1 の遅延によって遅れている可能性があることを検知し、 $d$  の値を増加させる。HA が seg#5 と seg#6 に対する重複 ACK を受信した時点で、seg#3 と seg#4 はまだ CN へと転送されていない。このとき、受信した重複 ACK を破棄するとともに破棄した重複 ACK の数だけ  $e$  を増加させる。その後、HA が seg#3 と seg#4 を遅れて受信した場合は  $e$  と  $d$  の値を 0 に初期化し、seg#3 と seg#4 に対する ACK を AL へと転送する。seg#3 と seg#4 を受信しないまま  $e$  の値が閾値に達した場合、最後に受信した重複 ACK に再送フラグを付加して AL へと転送する。このとき  $e$  と  $d$  を 0 に初期化する。

### 3.3 AL-HA 間の RTT 測定に基づく送信経路選択

Mobile IP SHAKE 環境におけるパケットの順序逆転は AL-HA 間の各経路における通信帯域や遅延の違いによって発生する。そこで AL は、利用する各経路の AL-HA 間の RTT と各経路の送信バッファでの待ち時間から HA に最も早くセグメントが到着すると予想される経路を選出し、セグメントの送信経路に設定する(図 10)。AL は、利用する各経路ごとに AL-HA 間 RTT の値を保持する。AL-HA 間の RTT の測定には、HA がデータセグメント受信時に直ちに AL へと送信する擬似 ACK を利用し、AL は擬似 ACK を受信するごとにその擬似 ACK に対応するセグメントが送信された経路の AL-HA 間 RTT を更新する。AL がセグメントを送信する際には、その時点でセ

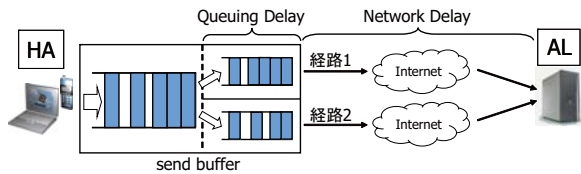


図 10 AL-HA 間 RTT を用いた送信経路選択

グメントが最も早く HA へ到着すると推測される経路を用いてセグメントを送信する。文献 [4] では、送信端末が一定時間内に確認応答されたデータセグメント量から各経路の帯域幅の予測値と送信バッファ内の送信待ちデータセグメントの量を用いることで送信バッファ内での待ち時間を算出する。また、各経路ごとに受信端末間の RTT を測定し、送信バッファでの待ち時間と RTT から送信したデータセグメントが受信端末に到着する時間を算出する。送信端末がセグメントを送信する際には、受信端末に到着する時間が最も短い経路を使用する。この方法を AL-HA 間で用いることで AL-HA 間の各経路においてセグメントの順序逆転を抑制する。

#### 4. 考 察

- HA-CN 間のセグメントの順序逆転とパケットロス

本稿で提案した手法では、HA-CN 間のセグメントおよび ACK の順序逆転またはパケットロスが発生しないことを前提とした。HA-CN 間でセグメントの順序逆転またはパケットロスが発生した場合、CN が送信する ACK のシーケンス番号 ( $ACKseq$ ) は、HA が保持する  $A$  より小さな値である。よって HA では、ACK 受信時の処理として、3.2.2 節における  $ACKseq < A$  の場合の処理を行う。このとき各経路の遅延差によってセグメントの順序逆転が発生している場合と HA-CN 間でセグメントの順序逆転またはパケットロスの発生を区別することができない。つまり AL-HA 間でセグメントの順序逆転が発生した状況において、HA-CN 間でセグメントの順序逆転またはパケットロスが発生すると、再送されなければいけないセグメントに対する重複 ACK が HA で破棄されてしまう。そこで、HA において重複 ACK の破棄する場合に、連続して破棄する数に閾値を設ける、もしくは不要と判別された重複 ACK をすべてでなく、一定の割合に応じた数だけ破棄するといった拡張が必要であると考えられる。

- 複数のセグメントの順序逆転およびパケットロス

本稿の第 3 章では、セグメントの順序逆転やパケットロスが単体で発生した場合について説明した。しかし、AL がセグメントの送信に利用する経路の 1 つで複数のパケットロスが発生したり、各経路の遅延差の増大によって複数のセグメントの順序逆転が発生することが考えられる。

そのような状況の一例として図 11 に AL-HA 間で複数のパケットロスもしくはセグメントの順序逆転が発生した状況を示す。HA は、経路 2 の遅延もしくはパケットロスによって  $seg\#3$  と  $seg\#5$  を受信する前に経路 1 で  $seg\#6$  を受信している。経路 2 の遅延によって遅れて  $seg\#3$  を受信した、場合もしくは経路 1 で送信されている  $seg\#4$  および  $seg\#6$  に対する ACK

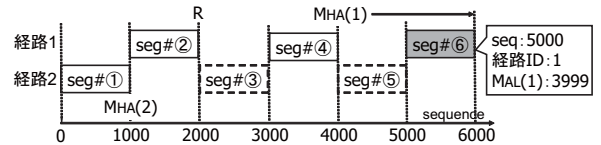


図 11 複数のセグメントの順序逆転またはパケットロスの発生

の AL での受信に伴って再送された  $seg\#3$  を受信した HA は、 $seg\#3$  に対する ACK ( $ACKseq = 4000$ ) により  $R$  を 4000 まで更新し、次に  $seg\#5$  の転送状況について  $seg\#3$  と同様の処理を行う。また、HA は  $seg\#3$  を受信する前に  $seg\#5$  を受信することが考えられる。この場合、 $seg\#3$  は経路 2 でパケットロスしているので、HA は CN から受信する重複 ACK を破棄することなく AL へと転送する。

実環境での Mobile IP SHAKE 環境における TCP 通信では、HA-CN 間のセグメントの順序逆転およびパケットロスや AL-HA 間での複数のセグメントの順序逆転およびパケットロスが複合的かつ頻繁に発生することが予想される。今後、シミュレーションも交え、詳細な動作の検証が必要である。

#### 5. ま と め

本稿では、Mobile IP SHAKE における TCP 通信において、利用する各端末の長距離低速リンクの性質がそれぞれ異なることで、セグメントの順序が逆転して受信端末に到着し、不要な重複 ACK の発生に伴う無駄な再送や輻輳ウィンドウの縮小が頻発するという問題に対して、その改善策を検討した。提案手法では、HA が各経路ごとに中継するセグメントのシーケンス番号を管理することでセグメントの順序逆転を検知し、受信端末から送信される不要な重複 ACK を破棄する。送信端末では、HA がデータセグメント転送時に送信端末へと送信する疑似 ACK を用いて HA 間 RTT を測定し、送信バッファでの待ち時間と HA 間 RTT から HA にセグメントが最も早く到着すると予想される経路の選択を行う。今後、シミュレーションによる提案手法の性能評価とともに、ECN や ELFN をの併用等による改善策の検討を行う予定である。

#### 文 献

- [1] K. Koyama, Y. Ito, S. Ishihara, et al., "Performance evaluation of TCP on Mobile IP SHAKE," IPSJ Journal, Vol.45, No.10, pp.2270-2278 (2004.10).
- [2] M. Zhang, J. Lai, et al., "A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths" In proc. of 2004 USENIX Annual Technical Conference
- [3] K. Kim and K.G. Shin., "Improving TCP Performance over Wireless Networks with Collaborative Multi-homed Mobile Hosts," MobiSys '05 pp107-120, (2005)
- [4] Y. Hasegawa, I. Yamaguchi, et al., "Improved data distribution for multipath TCP communication," GLOBECOM '05, Volume 1, (2005)
- [5] K. Rojviboonchai, T. Osuga, H. Aida., "R-M/TCP: Protocol for Reliable Multi-path Transport over the Internet," AINA '05, pp.801-806 vol.1.
- [6] S. Mascolo, C. Casetti, M. Gerla, al., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," Mobicom '01 pp287-297.