

## Mobile IP SHAKE での TCP 通信における HA での再送抑制方法の評価

荻野 秀岳<sup>†</sup> 石原 進<sup>††</sup>

筆者らは、Mobile IP 環境において、複数の移動端末が協調し、各端末が持つ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、利用する各端末のリンクの性質がそれぞれ異なるために、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、その解決手法として HA 不要 ACK 破棄法および RAH (RTT between AL and HA) 経路選択法を提案・評価する。HA 不要 ACK 破棄法は、Mobile IP SHAKE においてトラフィックの分配・集約を行う Home Agent (HA) が、中継するセグメントの情報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 を破棄する手法である。RAH 経路選択法は、HA がデータセグメント転送時に送信端末へ送信する擬似 ACK を用いて、移動端末が利用する各経路ごとに AL-HA 間の RTT を測定し、HA にセグメントが最も早く到着すると予想される経路を送信経路に選択する手法である。HA 不要 ACK 破棄法および RAH 経路選択法をシミュレーションにより評価した。シミュレーションの結果、両方式によってパケットの順序逆転の抑制および不要な再送の抑制が確認できた。

### Evaluation of a scheme for avoiding unnecessary retransmission of TCP on HA of Mobile IP SHAKE

HIDETAKE OGINO<sup>†</sup> and SUSUMU ISHIHARA<sup>††</sup>

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). 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. In this paper, we address a problem that frequent useless retransmission due to unnecessary duplicate TCP ACKs caused by arrival of out of order segments. We also propose two schemes to solve this problem. In this scheme, a Home Agent (HA) of mobile IP drops unnecessary duplicate ACKs by managing a sequence number of segments transmitted on each links and a mobile node selects a link that can transmit a segment earliest to the HA by using round trip time between HA and the sending node. Simulation results show that the proposed scheme decreases out of order packets and unnecessary retransmission.

#### 1. はじめに

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<sup>1)</sup>。

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、重複 ACK の発生に伴う無駄な再送が頻発する。この結果、TCP の輻輳ウィンドウが増加しにくくなり、高いスループットを得ることができない。

本稿では、Mobile IP SHAKE の Alliance から Alliance 外部への上り通信において、セグメントの順序逆転によって発生する不要な重複 ACK に伴う再送の抑制

<sup>†</sup> 静岡大学大学院工学研究科  
Graduate School of Science and Engineering, Shizuoka University  
<sup>††</sup> 静岡大学創造科学技術大学院  
Graduate School of Science and Technology, Shizuoka University

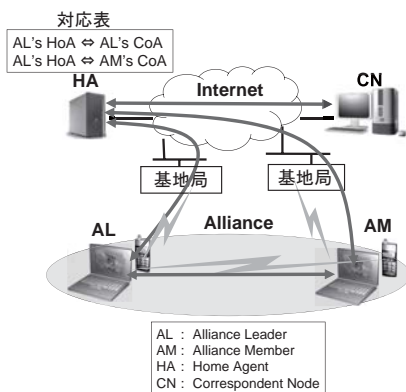


図1 Mobile IP SHAKE

方法を提案・評価する。

Mobile IP SHAKE では、Home Agent ( HA ) は複数の端末の移動情報の管理とトラフィックの分配・集約を行うため、負荷が集中しやすい。したがって、TCP での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、本来であれば端末ごとに持つ処理の重い機能の追加は望ましくない。HA に機能を追加する場合、軽量の機能の追加にとどめることが望ましい。そこで、HA への軽量の機能追加による再送抑制手法として、HA が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出し、セグメントロスでなく、セグメントの順序逆転によって発生する不要な重複 ACK を破棄する手法を提案する。以下、この手法を HA 不要 ACK 破棄法と呼ぶ。

さらに、AL と HA 間の各経路での遅延を測定することで、できるだけ AL-HA 間の複数経路でのセグメントの順序逆転が起きないようにする手法を提案する。以下、この手法を RAH ( RTT between AL and HA ) 経路選択法と呼ぶ。

複数経路を用いた TCP 通信に関しては、多くの試みがなされている。例えば、Rojviboonchai らは、各経路のボトルネック部の帯域幅を一定時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輻輳制御手法を提案している<sup>2)</sup>。また、Lee らは、利用する複数の経路の数に応じて、再送処理へ移行するまでの重複 ACK 数を決定し、Delayed ACK を用いた場合でも、再送セグメントに対してのみ即座に ACK を返信することで、早期に高速再送処理を完了させる手法を提案している<sup>3)</sup>。これらの方式は、両エンドの TCP が複数経路に対応することを必要としている。一方、本稿では Mobile IP SHAKE 環境での TCP 通信の効率化を対象とし、移動端末以外の 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。

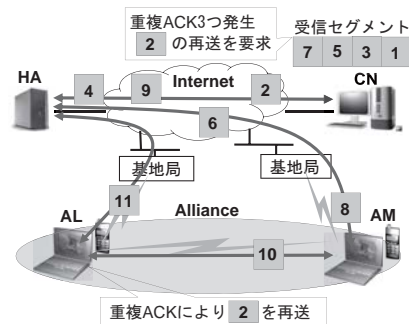


図2 Mobile IP SHAKE における TCP 通信の問題

## 2. Mobile IP SHAKE

### 2.1 Mobile IP SHAKE の概要

Mobile IP SHAKE は、Mobile IP において移動端末の移動情報を管理する Home Agent ( 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、複数の経路を同時に利用した通信を実現する。Alliance 内の移動端末が持つ長距離無線リンクを利用して実際に通信を行う移動端末である Alliance Leader ( AL ) は、AL に通信帯域を提供する他の移動端末である Alliance Member ( AM ) の Care of Address ( CoA ) を HA に通知する。HA は、AL の Home Address ( HoA ) と CoA の対応および AL の HoA と AM の CoA の対応を保持する。HA は、この対応に基づいて通信相手 ( CN: Correspondent Node ) が AL に送信するパケットを HA と AL 間および HA と AM 間を結ぶ複数の経路へ分配する。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 の外部リンクを用いた経路へパケットを分配する。

### 2.2 Mobile IP SHAKE での TCP 通信の問題

Mobile IP SHAKE で用いる複数の経路、すなわち AL および各 AM の外部リンクの通信帯域や遅延はそれぞれ異なり、さらにそれぞれ変動する。したがって、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、パケットの順序が逆転して CN に到着してしまうことが避けられない。TCP 通信では、累積確認応答を行うため、セグメントの順序が逆転して CN 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、重複 ACK を AL へ送信する。重複 ACK を連続して 3 つ受信した AL は、ロスしていないセグメントを無駄に再送し、かつ輻輳ウィンドウを縮小する。このため、スループットが低下してしまう。

図2の例では、AL の外部リンクを用いた経路の遅延に比べて、AM を経由する経路の遅延が大きいため、セグメントの順序逆転が発生する。CN は、セグメントの順序逆転が発生したことを検知できないため、重複 ACK を AL へ送信しなければならない。AL の TCP は、重複 ACK を 3 つ受信すると、実際には経路の遅延によって遅れてセグメント 2 が配送されているにもかかわらず、ロスし

ていないセグメント 2 を再送しなければならない。ロスしていないセグメントの無駄な再送が発生すると、TCP の輻輳ウィンドウが縮小されてしまう。

通常の TCP では、無駄な再送の原因である不要な重複 ACK の発生を避けることができない。ただし、CN 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<sup>4)5)</sup>、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である。しかし、Mobile IP SHAKE 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ため、CN の TCP による不要な重複 ACK の送信を抑制できない。AL においては、複数経路に対応した TCP を備えることが可能であるが、そのような TCP であっても、CN が送信する重複 ACK が不要であるかを AL 単体では判別することができない。

### 3. 不要な重複 ACK 破棄による再送の抑制

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。そこで、HA と AL において、不要な再送を抑制する機能の追加を考える。

Mobile IP SHAKE においてセグメントの順序逆転の多くは AL-HA 間のリンクで発生するため、下り通信では、HA および AL はセグメントの順序逆転を検知することができない。一方、上り通信においては、HA は CN に到着する順番とほぼ同じ順番でセグメントを受信するため、セグメントの順序逆転を検知することができる。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を管理する 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。

そこで本稿では、AL から CN への上り通信に対して、HA と AL で動作し、かつ HA 上での処理が軽量である不要な再送を抑制する手法として (i)HA 不要 ACK 破棄法、(ii)RAH 経路選択法を提案する。

#### 3.1 HA 不要 ACK 破棄法

HA 不要 ACK 破棄法では、HA は AL から CN へ中継するセグメントの TCP ヘッダ情報を用いて、セグメントの順序逆転またはロスの発生を検知し、不要な重複 ACK を破棄することで、AL での不要な再送を抑制する。HA は、重複 ACK を受信した時点で、その重複 ACK が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 は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 4 つの変数を利用する。

(1)  $M_{AL}(i)$ : 各経路  $i$  から以前に受信した最大シーケ

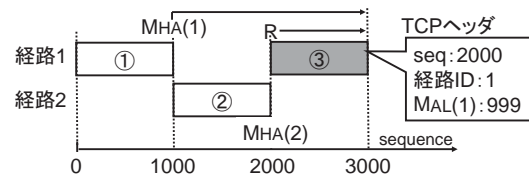


図 3 HA が正しい順序でセグメントを受信した場合の処理

ンス番号

(2)  $M_{HA}(i)$ : 経路  $i$  から受信した最大シーケンス番号

(3)  $R$ : AL から欠落なく受信したデータの最大シーケンス番号

(4)  $A$ : CN から受信した最大の ACK 番号

AL は、各経路で送信した最大シーケンス番号である  $M_{AL}(i)$  を保持し、セグメント送信時に、 $M_{AL}(i)$  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、 $M_{AL}(i)$ 、 $M_{HA}(i)$ 、 $R$  を用いてセグメントの順序逆転または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。

##### 3.1.1 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  $R$  を比較し、そのセグメントを正しい順序で受信できたか判別する。

$seq = R + 1$  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。 $M_{HA}(i)$  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 3 に沿って説明する。以下、セグメント長を 1000 とする。図 3 は、先頭シーケンス番号が 2000 の  $seg\#3$  を HA が経路 1 から受信した状況である。 $seg\#3$  を受信する前には、先頭シーケンス番号が 0 の  $seg\#1$  を経路 1 から、先頭シーケンス番号が 1000 の  $seg\#2$  を経路 2 から受信している。HA は、経路 1 から  $seg\#1$  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 から  $seg\#2$  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 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。 $R$  およびセグ

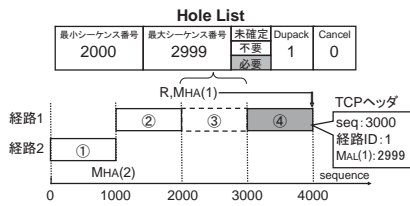


図 4 未受信のシーケンス番号空間を検知した場合 ( $seq > R + 1$ )

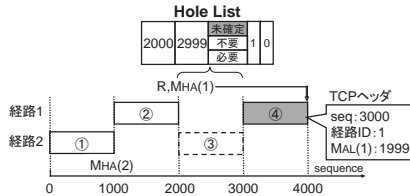


図 5  $M_{HA}(i) = M_{AL}(i)$  の場合

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する。具体例を図 4 に沿って説明する。図 4 は、 $seg\#2$  を経路 1 から受信したあとに、続けて経路 1 から  $seg\#4$  を受信した状況を示している。このとき HA は、受信した  $seg\#4$  の  $seq (= 3000)$  と  $R (= 1999)$  を比較することで、 $seq - R + 1$  であることから、シーケンス番号 2000 ~ 2999 の空間が未受信であることを検知する。HA が未受信のシーケンス番号空間を検知した場合、未受信のシーケンス番号空間を管理する *HoleList* を作成する。HA は、1 つの連続した未受信シーケンス番号空間に対して 1 つの *HoleList* 要素を作成する。*HoleList* 要素は、以下の 5 項目から構成される。

- 未受信シーケンス番号空間の最小シーケンス番号
- 未受信シーケンス番号空間の最大シーケンス番号
- ACK 受信時の処理を決定するモード (*Mode*)  
未確定、不要、必要の 3 モードがあり、未受信シーケンス番号空間がロスまたは遅延により遅れているのか判断できない状況では未確定モード、未受信シーケンス番号空間を埋めるセグメントを受信した状況では不要モード、未受信シーケンス番号空間に対するセグメントがロスした状況では必要モードとなる。
- 破棄可能であると推定される重複 ACK 数 (*Dupack*)
- 破棄された重複 ACK 数 (*Cancel*)

HA は、*HoleList* を作成すると、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、セグメントの順序逆転またはロスがどの経路で発生しているかを判別する。

(a)  $M_{HA}(i) < M_{AL}(i)$  の場合 (図 4)

経路  $i$  でセグメントロスが発生した状況である。 $seg\#4$  を受信した HA では、TCP ヘッダ内の  $M_{AL}(1) (= 2999)$  と  $M_{HA}(1) (= 1999)$  を比較し、 $M_{HA}(i) < M_{AL}(i)$  であ

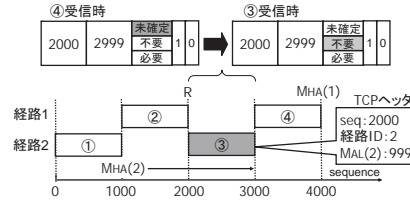


図 6 *HoleList* に対応するセグメントを受信した場合

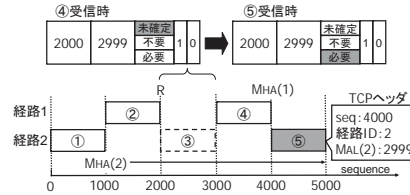


図 7 *HoleList* に対応するセグメントがロスした場合

ることから、 $seg\#3$  が経路 1 から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る。HA は  $M_{HA}(1)$  に  $seq + L - 1$  を代入し、 $seg\#4$  を CN へ転送する。

(b)  $M_{HA}(i) = M_{AL}(i)$  の場合 (図 5)

経路  $i$  以外の経路でロスが発生した、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。 $seg\#4$  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 から  $seg\#2$  が送信されたことがわかる。つまり  $seg\#3$  は、経路 1 以外の経路で送信されている。このとき、 $seg\#3$  は経路 2 でロスしているか、経路 2 の遅延により HA への到着が遅れている。 $seg\#4$  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、*HoleList* 要素の *Dupack* を 1 増加させる。そして  $M_{HA}(1)$  に  $seq + L - 1$  を代入し、 $seg\#4$  を CN へ転送する。

*Mode* の変更

HA は、*HoleList* に対応するセグメントを受信した場合、および *HoleList* に対応するセグメントがロスした場合に、*HoleList* 要素の *Mode* を変更する。

- *HoleList* に対応するセグメントを受信した場合  
*HoleList* に対応するセグメントは、すでに CN へ転送されているため、*HoleList* に対応する重複 ACK は不要な ACK と判別できる。そこで、*HoleList* 要素の *Mode* を不要モードにセットする。具体例を図 6 に示す。HA は、経路 1 から  $seg\#4$  を受信後、経路 2 から  $seg\#3$  を受信している。HA が  $seg\#3$  に対応する *HoleList* 要素の *Mode* を不要モードにすることで、重複 ACK (ACK 番号: 2000) を破棄する。

- *HoleList* に対応するセグメントがロスした場合  
*HoleList* に対応するセグメントがロスしていることが検知された場合、*HoleList* に対応する重複 ACK は、セグ

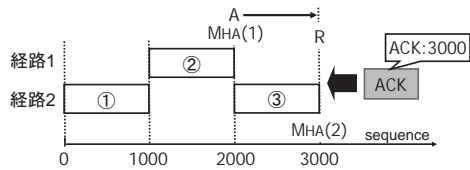


図 8  $ack > A$  の場合

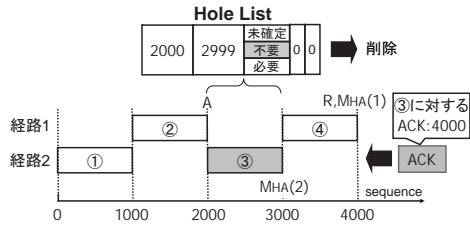


図 9 HoleList 要素を削除する場合

メントのロスを検知するために必要な重複 ACK である。HA は、HoleList 要素の Mode を必要モードにセットする。具体例を図 7 に示す。HA は、経路 1 から seg#4 を受信後、経路 2 から seg#5 を受信している。HA が seg#3 に対応する HoleList 要素の Mode を必要モードにすることで、重複 ACK (ACK 番号:2000) を破棄することなく、AL へ転送する。

### 3.1.2 ACK 受信時の処理

CN から ACK (シーケンス番号:  $ack$ ) を受信した HA は、HA が保持する受信した最大 ACK 番号である  $A$  と  $ack$  を比較することで、重複 ACK であるかを判別する。 $ack > A$  の場合

受信した ACK は重複 ACK ではないので、 $A$  に  $ack$  を代入した後、ACK を AL へ転送する。図 8 は、seg#3 受信後に ACK ( $ack:3000$ ) を受信した状況を示している。このとき、 $ack$  の値から 3000 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。そこで、 $A$  の値を 3000 まで更新し、その ACK を AL へと転送する。

HA では、 $ack > A$  である ACK を受信した際、不要になった HoleList 要素を削除する。図 9 は、経路 1 で seg#4 を受信後、経路 2 で 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。seg#3 を受信した CN は、seg#4 の最大シーケンス番号まで欠落なく受信しているので  $ack:4000$  の ACK を AL へ送信する。ACK ( $ack:4000$ ) を受信した HA は、ACK 番号が 4000 未満の ACK を今後受信することがないため、シーケンス番号が 4000 未満の HoleList 要素を削除する。

#### $ack < A$ の場合

受信した ACK は重複 ACK である。HA は、 $ack$  に対応する HoleList 要素の Mode によって ACK の処理を決定する。

#### - Mode が未確定モードの場合

対応する HoleList 要素のシーケンス番号空間のセグメントが、経路の遅延差によって遅れて配送されているの

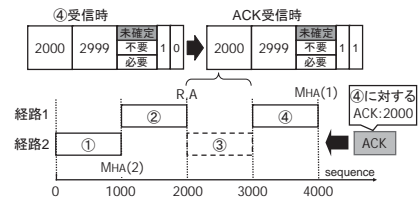


図 10 HoleList 要素の Mode が未確定モードの場合

か、セグメントのロスが発生しているのか判別できない状況である。そこで、経路の遅延差によってセグメントの到着が遅れている可能性を考え、HA は受信した ACK を破棄する。そして破棄した重複 ACK の数をカウントする変数である HoleList 要素の Cancel を 1 増加する。Cancel には、セグメントがロスしている場合には、重複 ACK を AL へ転送する必要があることを想定し、一定の閾値  $\theta_c$  を設ける。Cancel の値が閾値  $\theta_c$  を超えた場合、HA はその ACK の TCP ヘッダに再送フラグをセットし、AL へ転送する。具体例を図 10 に示す。図 10 は、経路 1 で seg#2 を受信後、経路 1 で seg#4 を受信し、seg#4 に対する ACK を受信した状況である。このとき、seg#3 は未受信であるため、HoleList 要素の Mode は未確定モードになっている。そこで ACK を破棄するとともに HoleList 要素の Cancel を 1 増加させる。

#### - Mode が不要モードの場合

対応する HoleList 要素のシーケンス番号空間のセグメントを HA がすでに受信した状況である。重複 ACK によるセグメントの再送は望ましくないため、その ACK を破棄するとともに、HoleList 要素が保持する破棄可能な重複 ACK 数 Dupack を 1 減少させる。Dupack が 0 の場合、重複 ACK を破棄することなく AL へ転送する。HA では、セグメントを受信した時点で HoleList 要素に対する ACK 数を Dupack を用いてカウントすることで、破棄する ACK 数に制限を設ける。これにより、HA-CN 間でセグメントがロスした場合であっても、必要以上に重複 ACK を破棄しないことで、AL がセグメントのロスを検知することができる。具体例を図 11 に示す。図 11 は、経路 1 で seg#2、seg#4 の順でセグメントを受信後、経路 2 で seg#3 を受信し、seg#4 に対する ACK を受信した状況である。このとき、HoleList 要素の Mode が不要モードであるため、その ACK を破棄し、Dupack の値を 1 減少させる。

#### - Mode が必要モードの場合

対応する HoleList 要素のシーケンス番号空間のセグメントは、ロスしたことが明白である。よって HA は受信した ACK を AL へと配送する。

## 3.2 RAH 経路選択法

AL における経路選択を適切に行うことによって、セグメントの順序逆転を防ぐ方法として、AL-HA 間 RTT および送信キューでの待ち時間に基づく、RAH (RTT be-

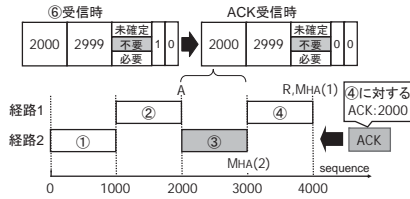


図 11 HoleList 要素の Mode が不要モードの場合

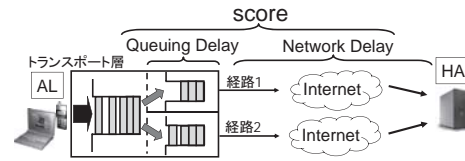


図 12 AL における RAH 経路選択法

tween AL and HA) 経路選択法を提案する。概要を図 12 に沿って説明する。HA は、AL からセグメントを受信すると、AL へ擬似 ACK を送信する。AL は、データセグメントの送信から擬似 ACK の受信までの時間によって、利用する各経路ごとに AL-HA 間 RTT を測定する。さらに、AL がセグメントを送信する時点で、送信キューにある送信待ちデータ量と、一定時間内に受信した擬似 ACK から算出されるスループットから、各経路の送信キューでの送信待ち時間を算出する。そして AL は、利用する各経路の AL-HA 間 RTT および送信キューでの送信待ち時間の合計として *SCORE* を算出する。AL は、この *SCORE* が最小の経路、つまり HA にセグメントがもっとも早く到着する経路を送信経路として選択する。これにより、AL-HA 間におけるセグメントの順序逆転の発生頻度を可能な限り減少させることができる。文献 6) では、複数経路を利用した TCP 通信において、End-to-End での RTT と送信キューでの送信待ち時間を利用した送信経路選択手法が提案されている。本提案は、文献 6) で提案された手法を、送信端末である AL とセグメント中継端末である HA 間に応用したものである。AL-HA 間 RTT の測定および送信キューでの待ち時間の算出方法を以下に詳しく説明する。

### 3.2.1 AL-HA 間の RTT 測定

AL は、セグメントを送信する際、送信する経路の ID を TCP ヘッダに付加する。セグメントを受信した HA は、AL に返信する擬似 ACK の TCP ヘッダに含まれる経路 ID フィールドおよびタイムスタンプエコーオプションフィールドに、受信したセグメントの TCP ヘッダに含まれる経路 ID およびタイムスタンプオプションをセットし、AL へ送信する。HA では、受信した擬似 ACK の経路 ID およびタイムスタンプオプションおよびタイムスタンプエコーオプションを用いることで、利用する各経路の AL-HA 間 RTT を測定することができる。また、AL-HA 間 RTT は、リンク状態によって大きく変動する可能性がある。そこで式 (1) を用いて平滑化を行う。

$$srtt(i) = \frac{7}{8}srtt(i-1) + \frac{1}{8}rtt(i) \quad (1)$$

$rtt(i)$  は、経路  $i$  で送信したセグメントに対する擬似 ACK を受信した際に測定した AL-HA 間 RTT である。 $srtt(i-1)$  は、以前に経路  $i$  で送信したセグメントに対する擬似 ACK を受信した際に算出された AL-HA 間 RTT

を用いて平滑化された RTT である。

### 3.2.2 送信キューでの送信待ち時間の測定

AL は、セグメントを送信する際、利用する各経路の送信キューに溜まっている送信待ちデータ量をチェックする。そして送信待ちデータ量と各経路ごとに測定するスループットを用いることで、各経路における送信キューでの送信待ち時間 ( $G(i)$ ) を測定する。ここで、各経路ごとに測定するスループットとは、一定時間内に受信する擬似 ACK によって確認応答されたデータ量である。各経路ごとに測定するスループットにおいても AL-HA 間 RTT 同様に式 (2) を用いて平滑化を行う。

$$G(i) = \frac{1}{2}G(i-1) + \frac{1}{2}TPUT(i) \quad (2)$$

$TPUT(i)$  は、経路  $i$  で一定時間ごとに算出されるスループットである。

### 3.2.3 各経路における HA までの到着時間の算出

AL は、AL-HA 間 RTT の測定および送信キューでの送信待ち時間の算出結果から、式 (3) を用いることで、セグメントを送信した場合の HA までの到着時間 (*SCORE*) を各経路ごとに算出する。

$$SCORE(i) = \frac{Q(i)}{G(i)} + \frac{srtt(i)}{2} \quad (3)$$

ここで  $Q(i)$  は、経路  $i$  の送信キューに溜まっているデータ量である。AL は、利用する各経路の *SCORE* を算出し、*SCORE* が最小の経路を送信経路として選択する。

## 4. シミュレーション評価

HA 不要 ACK 破棄法および RAH 経路選択法の性能評価を行うため、ns2 を用いてシミュレーションを行った。この結果に基づき、以下の 3 つの送信経路選択方法における HA 不要 ACK 破棄法の効果を比較する。

- RoundRobin 方式
- 遅延差方式 (利用する 2 経路の遅延の比に応じた比率で送信機会を提供する方式)
- RAH 経路選択方式

シミュレーションに用いたネットワークトポロジを図 13 に示す。図 13 において、AM-HA 間の遅延を 10~100ms、パケットロス率を 1~5% の範囲で変動させた。シミュレーションでは、100 秒間のファイル転送を行い、10 回の平均値を評価に用いた。なお、3.1.1 節で示した、HA での再送要請までの閾値  $\theta$  は 5 としている。

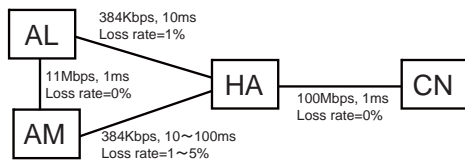


図 13 シミュレーショントポロジ

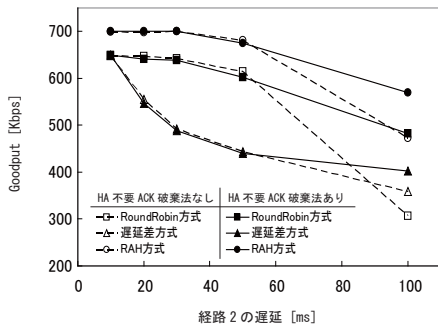


図 14 経路の遅延差とグッドプットの関係

#### 4.1 経路の遅延差の影響

AM-HA 間のパケットロス率を 1% で固定し、遅延を 10~100ms で変化させたときのシミュレーション結果を図 14 に示す。送信経路選択方法に RAH 経路選択法を用いた場合、RoundRobin 方式および遅延差方式に比べて、高いグッドプットを得ることができた。さらに、各送信経路選択方法において、HA 不要 ACK 破棄法を併用することで、各送信経路選択方法のみ用いた場合に比べて、より高いグッドプットを得ることができた。経路 2 の遅延が 50ms 以下の状況では、再送が発生するほど複数のセグメントを跨いだ順序逆転の発生頻度が低いため、HA 不要 ACK 破棄法を用いても、用いない場合とほぼ同等のグッドプットを得られた。

HA 不要 ACK 破棄法の効果を確認するため、送信経路選択方法に RoundRobin 方式を用いた場合における経路 2 の遅延が 100ms の場合の輻輳ウィンドウの推移の一例を図 15 に示す。HA 不要 ACK 破棄法を用いない場合、シミュレーション時間内において頻繁に再送が発生し、輻輳ウィンドウが 1~7 程度を推移している。一方、HA 不要 ACK 破棄法を用いた場合、HA 不要 ACK 破棄法を用いなかった場合に比べて、再送が頻発することなく、輻輳ウィンドウの縮小を抑制していることがわかる。図 15 における 0~7s の拡大図を図 16 に示す。図 16 において、HA 不要 ACK 破棄法を用いない場合、7 秒間に 13 回の再送が発生している。しかし、実際にセグメントロスが発生し、再送されるべきセグメントは 2 つである。一方、HA 不要 ACK 破棄法を用いた場合、実際にセグメントロスが発生したときのみ再送されることがわかる。

#### 4.2 パケットロス率の影響

AM-HA 間の遅延を 10ms で固定し、パケットロス率を 1~5% で変化させたときのシミュレーション結果を

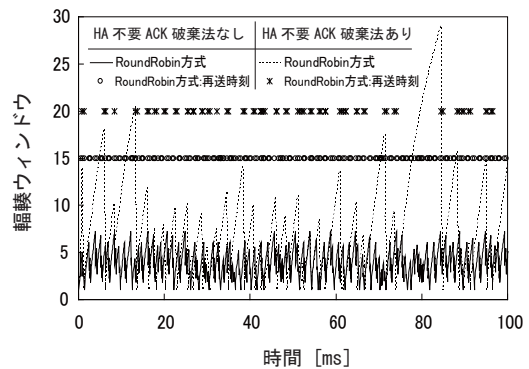


図 15 経路 2 の遅延が 100ms の場合の輻輳ウィンドウの推移

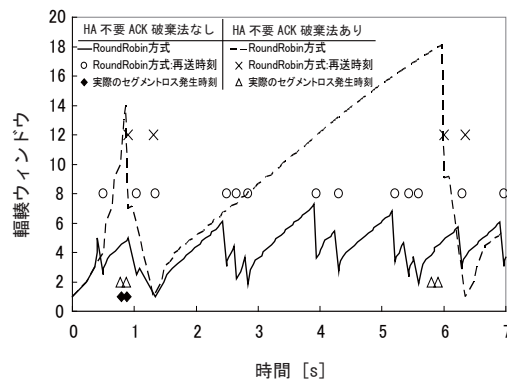


図 16 図 15 の拡大図

図 17 に示す。パケットロス率が上がるにつれて、HA 不要 ACK 破棄法を用いた場合のグッドプットが、HA 不要 ACK 破棄法を用いない場合を下回ってしまうことが確認できる。さらに、送信経路選択方法として RoundRobin 方式を用いた場合、パケットロス率が上がるにつれて、HA 不要 ACK 破棄法を用いた場合は、用いない場合に比べ、グッドプットの低下が著しい。

主な理由として、再送タイムアウトの増加が考えられる。HA 不要 ACK 破棄法では 3.1.1 で述べたように、HA が重複 ACK を受信した状況では、対応する HoleList 要素の Mode が未確定モードの場合、HoleList 要素に対応するセグメントが経路の遅延によって遅れて配送されている可能性を考え、破棄する。しかし、HoleList 要素に対応するセグメントが実際にロスした場合、HA が重複 ACK を破棄してしまった分、AL でロスを検知するまでの時間が長くなってしまふ。ロスを検知するまでの時間が長くなると、再送タイムアウトが発生してしまう。

送信経路選択方法として RoundRobin 方式を用い、HA 不要 ACK 破棄法を用いた場合と用いない場合の高速再送および再送タイムアウトの発生回数を図 18 に示す。HA 不要 ACK 破棄法を用いない場合、パケットロス率が増加すると、高速再送および再送タイムアウトの発生回数はほぼ一定の割合で増加する。一方、HA 不要 ACK 破棄法を用いる場合、パケットロス率が増加すると、高速再

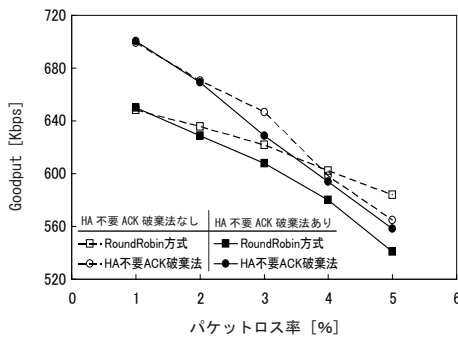


図 17 パケットロス率と Goodput の関係 (HA 不要 ACK 破棄法)

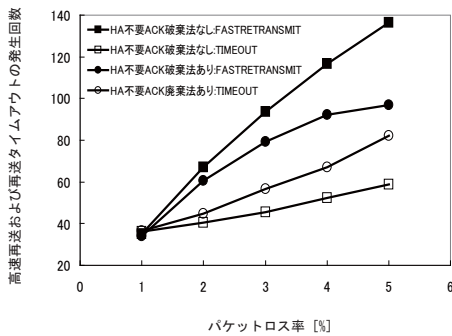


図 18 高速再送および再送タイムアウトの発生回数

送の発生回数の増加率が下がり、再送タイムアウトの発生回数の増加率が上がっていることが確認できる。TCP では、高速再送でロスが検知されると、輻輳ウィンドウの値を半分にする。一方で、再送タイムアウトでロスが検知されると輻輳ウィンドウの値が 1 まで減少してしまう。HA 不要 ACK 破棄法を用いた場合、高速再送の発生頻度を抑制することができても、再送タイムアウトの発生頻度が増加することで、高いグッドプットを得ることができない。

このような問題を解決するため、HA 不要 ACK 破棄法に改良を加える。HA において不要と判別され、破棄される重複 ACK の中で、AL で再送が発生しない重複 ACK の上限である 2 つまで、破棄せず AL へ転送する。以下、この方法を TDAR (Two Duplicate ACK Relay) 方式と呼ぶ。

AM-HA 間のパケットロス率を変化させた場合の TDAR 方式のシミュレーション結果を図 19 に示す。TDAR 方式を用いた場合、パケットロス率が上がっても、TDAR 方式を用いない場合に比べてグッドプットが下がることなく、5~10Kbps 程度のグッドプットの向上がみられる。

## 5. まとめ

本稿では、Mobile IP SHAKE における TCP 通信において、利用する各端末の外部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を示し、その

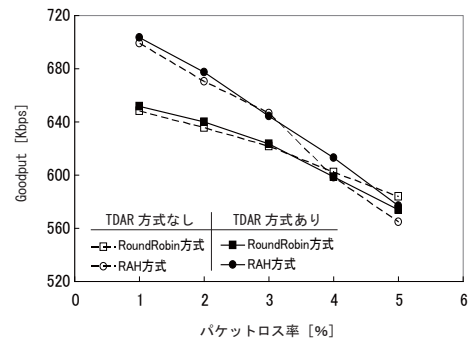


図 19 パケットロス率と Goodput の関係 (TDAR 方式)

改善策として、HA 不要 ACK 破棄法および RAH 経路選択法を提案した。シミュレーションの結果、RAH 経路選択法を用いることで、RoundRobin 方式および各経路の遅延比率に応じて分配する方式に比べ、高いグッドプットを得ることができた。さらに、各送信経路選択に HA 不要 ACK 破棄法を併用することで、不要な重複 ACK による無駄な再送が抑制され、輻輳ウィンドウが無駄に下がることなく通信を行えることを確認した。さらに、HA 不要 ACK 破棄法を用いた場合、パケットロス率が増加すると、HA 不要 ACK 破棄法を用いない場合に比べてグッドプットが著しく低下してしまう問題に対して、HA 不要 ACK 破棄法を拡張し、AL で再送が発生しない上限まで不要な重複 ACK を AL へ転送する TDAR 方式を提案した。TDAR 方式を用いた場合、パケットロス率が上がっても、TDAR 方式を用いない場合に比べて、グッドプットが低下しないことを確認した。今後の課題として、利用する各経路の帯域幅が異なる場合の評価および、HA における AL への ACK 転送経路の選択手法の検討、HA で破棄するセグメント数の上限である  $\theta_c$  の評価が挙げられる。

## 参考文献

- 1) K. Koyama, et al., "Performance evaluation of TCP on Mobile IP SHAKE," IPSJ Journal, Vol.45, No.10, pp.2270-2278 (2004).
- 2) K. Rojviboonchai, et al., "R-M/TCP:Protocol for Reliable Multi-path Transport over the Internet," In proc. of AINA '05 (2005).
- 3) Y. Lee, et al., "Improving TCP Performance in Multipath Packet Forwarding Networks," Journal of Communications and Networks, (2002).
- 4) M. Zhang, 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 (2004).
- 5) K. H. Kim, et al., "Improving TCP Performance over Wireless Networks with Collaborative Multi-homed Mobile Hosts," In proc. of MobiSys '05 (2005).
- 6) Y. Hasegawa, et al., "Improved data distribution for multipath TCP communication," In proc. of GLOBECOM '05, Vol. 1 (2005).