

LIN6におけるパケット転送を利用した高速ハンドオーバープロトコル

桐山 沢子[†] 内藤 巖之[†] 藤巻 聡美[†] 後郷 和孝[†] 寺岡 文男[†]

[†] 慶應義塾大学 〒 223-8522 神奈川県横浜市港北区日吉 3-14-1
E-mail: †{kiri,matthew,satie,gogo,tera}@tera.ics.keio.ac.jp

あらまし 現在、需要が高まっているモバイルコンピューティングに必要な不可欠なリアルタイム通信の実現には、ハンドオーバー時の通信品質の低下が大きな問題となっている。この問題の解決方式として高速ハンドオーバープロトコルが提案されているが、既存の高速ハンドオーバープロトコルは Mobile IPv6 に基づくため冗長通信経路・ヘッダオーバーヘッド・一点障害性の問題が生じる。そこで本研究では LIN6 に基づく高速ハンドオーバープロトコル EFLIN6 を提案する。EFLIN6 は常に最適経路・ヘッダオーバーヘッドなしで通信可能であり、耐故障性にも優れている。また、ハンドオーバー時にはパケットを bicast することでパケットロスを低減し、通信品質の低下を防ぐ。評価の結果、EFLIN6 では通信ノード間の RTT の影響を受けず、常にパケットロスを低減しスループットを向上させることに成功した。キーワード 高速ハンドオーバー、シームレスモビリティ、パケット転送、モバイルコンピューティング、無線 LAN

Fast-handover protocol using packet forwarding in LIN6

Sawako KIRIYAMA[†], Yoshiyuki NAITO[†], Satomi FUJIMAKI[†], Kazutaka GOGO[†],
and Fumio TERAOKA[†]

[†] Keio University 3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan
E-mail: †{kiri,matthew,satie,gogo,tera}@tera.ics.keio.ac.jp

Abstract To realize real-time communication, improvement of the performance degradation at handovers is one of the most important challenges. FMIPv6 has been proposed as a fast handover protocol for Mobile IPv6. But it has several problems such as pinball routing, large header overhead due to tunneling, and a single point of failure because it is based on Mobile IPv6. In this paper, we design and implement a new fast handover protocol called EFLIN6 which is based on a node mobility protocol called LIN6. EFLIN6 is more tolerant of network failure and provides optimal routing without header overhead. In EFLIN6, the number of packet losses is decreased by employing packet bicast during a handover and the performance degradation is mitigated. In addition, EFLIN6 improves the TCP throughput.

Key words Fast Handover, Seamless Mobility, Packet Forwarding, Mobile Computing, Wireless LAN

1. はじめに

近年、インターネットの発展と無線デバイスの普及によりあらゆる場所でインターネットに接続できる環境が整ってきている。よって、いつでも、どこでも、移動しながらインターネット上のサービスの利用を可能にするモバイルコンピューティング環境を実現するためには、移動中でも通信が途切れることのない移動透過な通信が必要となる。この移動透過性を保証するプロトコルのことをモビリティプロトコルと呼び、IETF (Internet Engineering Task Force) では Mobile IPv6 (MIPv6) [1] が標準化されている。しかし、MIPv6 には冗長通信経路・トンネリングによるヘッダオーバーヘッド・一点障害性

といった問題がある。よって、我々はノードの接続位置を表す情報である位置指示子とノード自体を表す情報であるノード識別子を概念的に分離することで移動透過性を保証するモビリティプロトコル LIN6 [2] を提案しており、LIN6 では常に最適経路での通信、ヘッダオーバーヘッドなし、耐故障性を実現している。しかし、これらのモビリティプロトコルは移動透過性は保証するが接続するサブネットを切り替えるハンドオーバーの際にはパケットロスや遅延が生じてしまうため、通信品質の低下が生じ音声や動画といったリアルタイム通信においては大きな問題となる。したがって、通信品質を低下させずに移動透過な通信を実現するにはハンドオーバーの高速化とハンドオーバー時のパケットロスの低減が必要不可欠である。

現在、MIPv6 に基づく高速ハンドオーバープロトコルとして Fast Handovers for Mobile IPv6 (FMIPv6) [3] が提案されている。これはリンク層情報をネットワーク層で利用し、L3 ハンドオーバー処理の一部を事前に済ませることでハンドオーバーの高速化を可能にしている。しかし、FMIPv6 ではリンク層情報の利用は前提とされており、どのようにリンク層情報を生成するかなどは未定義である。そこで我々はリンク層情報を抽象化 [4] し、レイヤ間情報交換機構 (LIES) [5] を利用してリンク層情報をネットワーク層に伝達することにより、ネットワーク層が主導にして高速にハンドオーバーを行う方式である L3-driven fast handover [6] を提案している。しかし、L3-driven fast handover では移動端末 Mobile Node (MN) と通信相手端末 Correspondent Node (CN) との RTT (Round Trip Time) が大きくなるにつれてパケットロスも増加するという欠点がある。

そこで、本論文では LING を拡張し、ハンドオーバー時には Access Router (AR) 間でパケットを転送することによりパケットロスを低減する高速ハンドオーバープロトコル EFLING を提案する。

2. L3-driven fast handover

L3-driven fast handover では、リンク切断やリンク接続といったリンク層情報をリンク層以外のレイヤでも使用できる形に抽象化し、LIES を用いてリンク層情報をネットワーク層へ伝達することによりネットワーク層がいつどの AR にハンドオーバーするのかを判断する。

図 1 に L3-driven fast handover におけるハンドオーバー時の流れを示す。

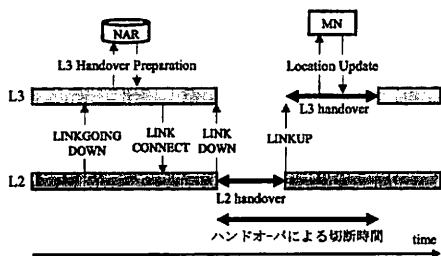


図 1 L3-driven fast handover におけるハンドオーバーの流れ

MN のリンク層は Access Point (AP) との通信品質が低下しリンク層の切断が予測されると、LINKGOINGDOWN をネットワーク層へ通知する。ネットワーク層はこれを受信すると、L3 Handover Preparation メッセージを使用し次に接続するサブネットで使用アドレスの Duplicate Address Detection (DAD) [7] の実行を要求する。そして、DAD が完了するとネットワーク層はリンク層へ LINKCONNECT を送信し L2 ハンドオーバーの開始を指示する。こうして L2 ハンドオーバーが開始され、新たな AP への接続が完了するとリンク層は LINKUP を通知する。ネットワーク層はこれを受信し L2 ハンドオーバーの完了を検知するとすぐに L3 ハンドオーバーを行う。

L3-driven fast handover では、L3 ハンドオーバー処理の一部である DAD を L2 ハンドオーバーの開始前に行っている。また従来のハンドオーバー方式では、ネットワーク層は AR が定期的に送信する Router Advertisement (RA) [8] を受信することでしか L2 ハンドオーバーの完了を検知できなかったが、LIES を用いることでリンク層が通知する LINKUP により L2 ハンドオーバーの完了を即座に検知できるため、L3 ハンドオーバー開始までに無駄な待ち時間が発生しない。これらによって、L3-driven fast handover ではハンドオーバーに要する時間を大幅に短縮している。

しかし、L3-driven fast handover ではハンドオーバー時のパケットロスに関しては考慮されていないため、L2 ハンドオーバーの開始から L3 ハンドオーバーが完了するまでの間に MN へ送信されたパケットはすべて損失してしまう。L3 ハンドオーバーではハンドオーバー後に MN が CN とメッセージを交換し位置情報を更新するため、L3 ハンドオーバーに要する時間は MN と CN 間の RTT に依存する。したがって、L3-driven fast handover では MN と CN 間の RTT の増加に伴い、パケットロスも増加するという問題点がある。

3. Fast Handovers for Mobile IPv6

MIPv6 を拡張した高速ハンドオーバープロトコルとして Fast Handovers for Mobile IPv6 (FMIPv6) が提案されている。FMIPv6 は、通常は MIPv6 と同様に動作しハンドオーバーが発生する前後のみ FMIPv6 特有の制御メッセージの交換や転送処理、バッファリングが行われる。

3.1 FMIPv6 の動作

図 2 に FMIPv6 の predictive mode でのハンドオーバー時のメッセージフローを示す。図中の黒い矢印は制御メッセージの送受信を表し、白抜き矢印は MN 宛のパケットの配送を表す。

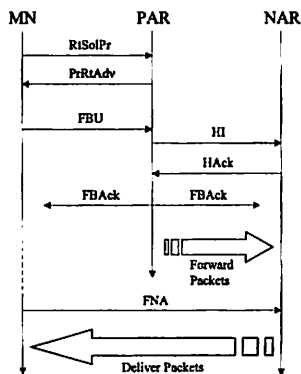


図 2 FMIPv6 のメッセージフロー

まず、MN はリンク層情報を用いて L2 ハンドオーバーの発生を事前に予測し Router Solicitation for Proxy (RtSolPr) , Proxy Router Advertisement (PrRtAdv) メッセージを使用してハンドオーバー後に接続する New Access Router (NAR) の情報を取得する。この情報を用いて、MN はハンドオーバー後のサ

ブネットで使用される New Care-of Address (NCoA) を生成し Fast Binding Update (FBU), Fast Binding Acknowledgement (FBAck) メッセージを使用して NAR における NCoA の DAD の実行や登録の準備を行う。さらに現在接続している Previous Access Router (PAR) と NAR 間では, Handover Initiate (HI), Handover Acknowledge (HACK) メッセージを使用して, MN 宛のパケットを PAR から NAR へ転送するためのトンネルを確立する。また, NAR はキューを作成し PAR からトンネリングを使用して転送されるパケットをバッファリングする。そして, MN は移動して L2 ハンドオーバーが完了すると Fast Neighbor Advertisement (FNA) メッセージを送信し, NAR でバッファリングしていたパケットを受信してハンドオーバーを完了させる。

FMIPv6 では, NCoA の生成や DAD などの L3 ハンドオーバー処理の一部を L2 ハンドオーバーの開始前に行うことでハンドオーバーに要する時間を短縮している。また, ハンドオーバー時には AR においてパケット転送とバッファリングを行うことでパケットロスを防いでいる。

3.2 FMIPv6 の問題点

FMIPv6 では, MIPv6 に基づくため冗長通信経路・ヘッダオーバーヘッド・一点障害性といった問題が伴う。

CN から MN へ送信されるパケットは図 3 に示すように, MN の位置情報を管理する Home Agent (HA) を経由し, HA と MN 間はトンネリングによって配送されるためトンネリングにより最低 40 bytes のヘッダオーバーヘッドが生じる。よって, これらのヘッダオーバーヘッドによって最大セグメントサイズを超えてしまう場合には, パケットが分割されるフラグメンテーションが発生し TCP スループットの低下につながる可能性がある。一方, LING6 では図 4 に示すように, 位置情報を管理する Mapping Agent (MA) を経由することなく MN と CN 間で直接通信できるためヘッダオーバーヘッドは発生しない。

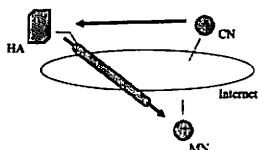


図 3 Mobile IPv6 での通信

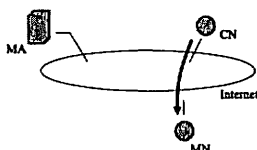


図 4 LING6 での通信

そこで, MIPv6 におけるヘッダオーバーヘッドの影響を評価するために, 図 3, 4 に示した通常通信時における MIPv6 と LING6 の TCP スループットを測定する予備実験を行った。この実験では, 第 6 章に示すマシンを用い 802.11b での無線通信を想定し帯域幅を 11 Mbps とし, MN と AR 間の無線通信におけるエラー率を変化させて測定を行った。測定結果を図 5 に示す。

図 5 が示すように, 常に LING6 の方がスループットは高く, 最大で 23 % 高くなっている。よって, MIPv6 ではヘッダオーバーヘッドによりフラグメンテーションが発生し, TCP スループットに大きな影響を与えていることが分かった。その上, FMIPv6

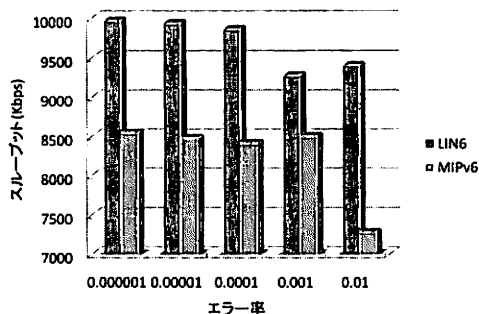


図 5 TCP スループットに与えるフラグメンテーションの影響

ではハンドオーバー時に PAR と MN 間を転送されるパケットはトンネリングされるため, 通常時の MIPv6 通信で発生するヘッダオーバーヘッドに加え, さらに 40 bytes のヘッダオーバーヘッドが生じる。以上の結果より, FMIPv6 ではヘッダオーバーヘッドによりさらに TCP スループットが悪化すると考えられる。また, バッファリングによる遅延の増加によって VoIP (Voice over Internet Protocol) のようなストリーミング通信にも悪影響を与えていると思われる。

4. 提案方式

本章では, 冗長通信経路・ヘッダオーバーヘッド・一点障害性の問題を解決したモビリティプロトコル LING6 を拡張した高速ハンドオーバープロトコルである EFLING6 (Edge Forwarding LING6) を提案する。

4.1 EFLING6 の概要

EFLING6 では, 以下に示す 3 つの主な機能を用いて高速ハンドオーバーを可能にするとともに, ハンドオーバーに伴うパケットロスの低減を実現する。

PAR による bicast

EFLING6 ではハンドオーバーに伴うパケットロスを低減させるために, ハンドオーバー時に MN に送信されたパケットを PAR がハンドオーバー前の MN のアドレスとハンドオーバー後の MN のアドレスの両方へ送信する bicast を用いる。EFLING6 における bicast 時の終点 IP アドレスを図 6 に示す。LING6 においてパケットの配送に用いられるアドレスを LING6 アドレスと呼び, LING6 アドレスは図 6 に示すように上位 64 bits のノードの接続位置を示すネットワークプレフィックスと下位 64 bits のインターネット上で一意となるノード識別子の LING ID から成る。よって EFLING6 では, PAR は終点 IP アドレスがハンドオーバー前の LING6 アドレスであるパケットを受信すると, そのパケットを複製し終点 IP アドレスの上位 64 bits をハンドオーバー後のネットワークプレフィックスに書き換えることでパケット転送を行う。LING6 では受信ノードはネットワーク層からトランスポート層にパケットを渡す前に LING6 アドレスの上位 64 bits を LING6 generalized ID と呼ばれる固定値に書き換えるため, パケット転送中の IP アドレス上位 64 bits の書き換えによる TCP チェックサムへの影響はない。よって EFLING6

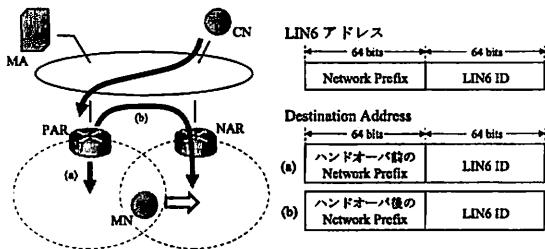


図 6 EFLIN6 における bicast 時終点 IP アドレス

ではヘッダオーバーヘッドなくパケットを転送することができる。また、FMIPv6 ではパケットロス低減のために NAR におけるバッファリングを行っているが、バッファリングされたパケットはハンドオーバー完了後に MN へ一斉に送信されるため遅延が生じるという問題点がある。よって、EFLIN6 では bicast を用いることで遅延の増加を防止している。

リンク層情報を用いた L3 主導の高速ハンドオーバー

EFLIN6 では、LIES を用いてネットワーク層とリンク層間での情報交換を行うことで高速ハンドオーバーを実現する。また高速ハンドオーバーの手法として、第 2 章で紹介した L3 主導の高速ハンドオーバーを用いることで、ネットワーク層がリンク層へ L2 ハンドオーバーの開始を指示するためハンドオーバー時の bicast を最適なタイミングで開始することができる。

Router Advertisement の拡張

高速ハンドオーバーを可能にするためには、次にどの AR へ移動するのかを予測するために近隣の AR や AP に関する情報を事前に取得しておく必要がある。そこで EFLIN6 では AR が定期的送信する RA を拡張し、近隣 AR や AP の情報を格納する Neighbor AR オプションを新たに定義することで、MN が近隣 AR や AP の情報を定期的に取得可能にする。Neighbor AR オプションのフォーマットを図 7 に示す。このオプションには AR のリンク層アドレスとグローバルアドレス、AP のチャンネルやリンクタイプ、リンク層アドレスといった情報が含まれる。また、Neighbor AR オプションは拡張 RA に未対応のノードでは不明な RA オプションと認識され、破棄されるため、従来の方式との共存が可能である。

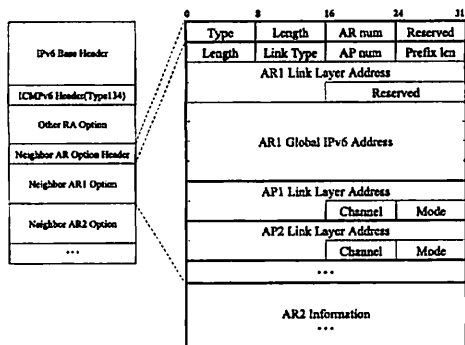


図 7 拡張 RA の Neighbor AR オプション

4.2 EFLIN6 のハンドオーバー手順

図 8 に EFLIN6 のハンドオーバー時のメッセージフローを示す。

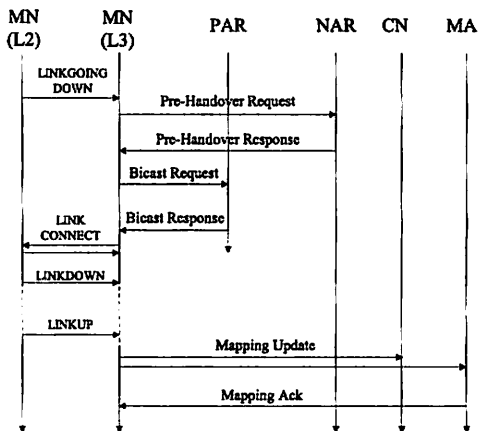


図 8 EFLIN6 のメッセージフロー

まず、MN のネットワーク層は LIES を介して LINKGOING-DOWN を受信し、L2 ハンドオーバーの発生を予測すると、NAR へ Pre-Handover Request メッセージを送信し MN がハンドオーバー後に NAR 下で使用するリンクローカルアドレスと LIN6 アドレスの DAD 処理を要請する。NAR はこれを受信すると DAD 処理を行い、アドレス重複がなければ MN が NAR 下で使用する LIN6 アドレスを Neighbor Cache へ登録する。そして、DAD 処理の結果を Pre-Handover Response メッセージに格納して MN へ返す。続いて、MN は Bicast Request メッセージを使用して PAR での bicast を要請する。Bicast は Bicast Request メッセージに格納されている Lifetime が切れるまで続けられる。MN は Bicast Response を受信し L3 ハンドオーバーの事前処理が完了すると、ネットワーク層からリンク層へ LINKCONNECT を送信し指定した AP への L2 ハンドオーバー開始を指示する。そしてネットワーク層は LINKUP を受信し、L2 ハンドオーバーの完了を検出すると L3 ハンドオーバーを実行し、MN は MA と CN へ Mapping Update メッセージを送信しハンドオーバー後の位置情報の更新を行う。

5. 実装

本方式を AR は FreeBSD 5.3-Release, MN は FreeBSD 6.1-Release に実装した。EFLIN6 の実装モジュールを図 9 に示す。EFLIN6 はもともとの LING の実装に加え、主に rtadvd デーモンと prehod デーモンの 2 つのデーモンプログラムから構成される。rtadvd デーモンは PAR と NAR で動作し、拡張 RA の送信、DAD 処理、Neighbor Cache への登録処理、bicast 処理を行う。prehod デーモンは MN で動作し、リンク層情報の利用、近隣 AR と AP 情報の取得・管理、ハンドオーバーの制御を行う。

PAR では、カーネルに bicast を行う対象のアドレスと bicast によるパケットの転送先アドレスの対応関係を保持したリストである Bicast List Table を実装した。つまり、MN のハンド

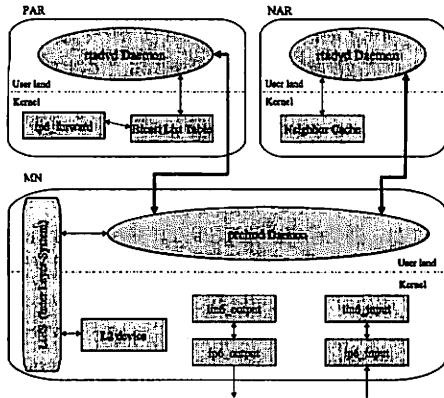


図9 EFLIN6の実装モジュール

オーバ前とハンドオーバ後の LIN6 アドレスの対応関係が保持されている。rtadvd デーモンは Bicast Request を受信するとその中に格納されている情報から MN のハンドオーバ前とハンドオーバ後の LIN6 アドレスを生成し、生成したアドレスを ioctl() システムコールを利用してユーザランドからカーネル内の Bicast List Table へ登録する。そして、パケットを転送する際に呼ばれる ip6_forward() 関数内でパケットの宛先アドレスと Bicast List Table に登録されている bicast 対象アドレスを照合することで bicast 対象のパケットを捕捉する。

NAR では、rtadvd デーモンは Pre-Handover Request を受信するとその中に格納されている情報から MN のリンクローカルアドレスと LIN6 アドレスを作成し Neighbor Cache のリストのアドレスと照合することで DAD 処理を行う。その結果、同じものがリスト内に存在すれば重複検知、存在しなければ重複なしと判断し、重複なしの場合はカーネル内の Neighbor Cache へと登録を行う。Neighbor Cache のリストの取得は sysctl() システムコールを使用して実装した。

MN に実装した EFLIN6 カーネルは LIN6 と LIES を組み合わせたものである。MN では、LIES によりカーネルとユーザランドに実装した prehod デーモンとの間でリンク層情報を交換し、prehod デーモンはこのリンク層情報を用いて Pre-Handover Request や Bicast Request といった制御メッセージを AR へ送信する。

6. 評価

本研究で提案・実装した EFLIN6 の定量的な性能評価を行った。本評価を行うにあたり構築した実験ネットワークを図 10 に示す。評価に用いたマシンのスペックは、AR は CPU VIA C3 Nehemiah+RNG+ACE, メモリ 512 MB, MN は CPU Intel®Pentium®M processor 1.60GHz, メモリ 1024 MB である。また、無線 LAN 環境は雑音や電波の干渉などの様々な要因でネットワークの状態が変化し評価を行うのが困難なため、本評価ではリアルタイムにハンドオーバ実験を行うことのできる無線環境エミュレータを使用した。無線環境エミュレータは PC 上で動作するソフトウェアで、電波強度、リンク切断時間、

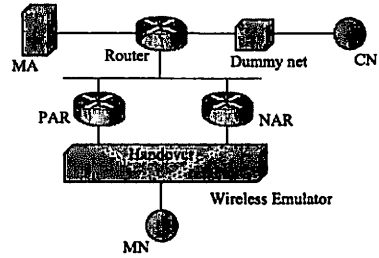


図10 実験ネットワーク

接続先などの様々なパラメータを自由に設定できる。今回の実験では、802.11b での無線通信を想定し帯域幅を 11 Mbps, ハンドオーバ時のリンク切断時間は文献 [9] での実測値に基づき 5 msec に設定した。

6.1 パケットロス

L3-driven fast handover, EFLIN6 におけるハンドオーバ時のパケットロスについて評価を行った。G.711 [10] 方式を用いた VoIP での通信を想定し、パケットサイズ 200 bytes のパケットを 20 msec 間隔で CN から MN へ送信し、遅延時間を制御することのできるツールである dummynet を使用して MN と CN 間の RTT を変化させてハンドオーバ時に発生したパケットロス数を測定した。測定結果を図 11 に示す。

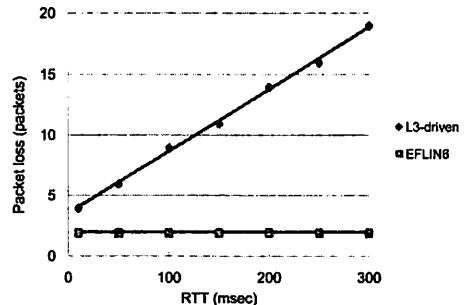


図11 パケットロス

図 11 が示すように、L3-driven fast handover は RTT の増加に伴いパケットロス数も単調増加している。これは、L3 ハンドオーバ処理の 1 つであるハンドオーバ後の位置情報を更新するために MN と CN 間で制御メッセージを交換するのに要する時間が MN と CN 間の RTT に依存するので、RTT が大きくなるとパケットロスも増加するためである。しかし、EFLIN6 では AR での bicast によりハンドオーバ時のパケットロスを防いでいるため、パケットロスが発生するのは L2 ハンドオーバ時のみとなりパケットロス数は MN と CN 間の RTT に関係なく一定となっている。よって、EFLIN6 は MN と CN 間の RTT の影響を受けることなく常にパケットロスを低減できることが分かった。

6.2 TCP スループット

LIN6, L3-driven fast handover, EFLIN6 における TCP スループットについて評価を行った。MN が車で移動し続けなが

ら通信を行う場合を想定し、移動の速さを 60 km/h, AP の電波到達範囲を半径 100 m とし、6 秒に 1 回ハンドオーバを発生させ、通信時間は 40 sec とする。

まず、MN と AR 間の無線通信におけるエラー率を変化させた場合の測定結果を図 12 に示す。このときの MN と CN 間の RTT は 100 msec とした。

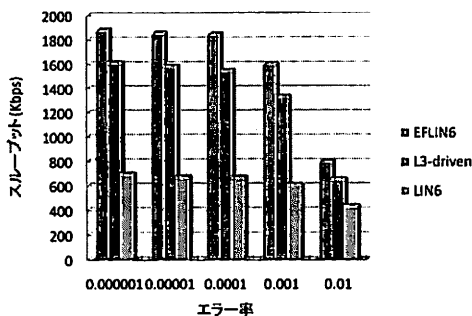


図 12 エラー率におけるスループットの比較

これらを比較すると、常に EFLIN6 のスループットが一番高くなっていることが分かる。また、EFLIN6 と L3-driven fast handover はネットワーク層主導の高速ハンドオーバ方式を用いているため、L3 ハンドオーバ時間が短縮され、LIN6 に比べ大幅にスループットが向上していることが分かる。さらに、EFLIN6 と L3-driven fast handover はエラー率が小さくなるほどスループットは上がっているが、LIN6 ではエラー率が 10^{-3} 以下ではほぼ一定となっている。つまり LIN6 ではハンドオーバ時のパケットロス数が多いため、エラー率に係わらずこれ以上のスループットを出せないことが分かる。

次に、MN と CN 間の RTT を変化させた場合の測定結果を図 13 に示す。このときの MN と AR 間の無線通信におけるエラー率は 10^{-6} とした。

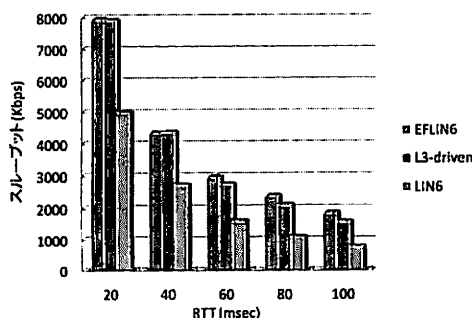


図 13 RTT におけるスループットの比較

これらを比較すると、図 12 と同様に常に EFLIN6 のスループットが一番高くなっていることが分かる。また EFLIN6 と L3-driven fast handover を比較すると、RTT が増加するにつれて EFLIN6 と L3-driven fast handover のスループットの差が大きくなっているのが分かる。これは第 6.1 節で述べたよ

うに、L3-driven fast handover は RTT の増加に伴いパケットロス数も増加するのに対し、EFLIN6 は AR での bicast によりパケットロス数は常に一定のためである。よってスループットの観点からも、EFLIN6 は RTT の影響を受けないことが分かった。

7. 結 論

本研究では、通信品質を低下させない移動透過な通信の実現手法として高速ハンドオーバプロトコル EFLIN6 を提案した。

EFLIN6 では、LIN6 を用いることで既存の高速ハンドオーバプロトコルの問題点である冗長通信経路・ヘッダオーバーヘッド・一点障害性の問題を解決し、ヘッダオーバーヘッドが発生しないことで MIPv6 と比較して TCP スループットを最大 23 % 向上できることが示された。また、ハンドオーバ時には AR で bicast を行うことで MN と CN 間の RTT の影響を受けることなく常にパケットロスの低減を実現し、ハンドオーバ時の TCP スループットの向上を実現した。よって、本方式はモバイルコンピューティング環境においてリアルタイム通信を実現するのに十分な品質が得られる方式であると結論付けることができる。

今後の課題としては MN と AR 間のセキュリティ問題の解決がある。EFLIN6 ではハンドオーバ時に MN と AR 間で制御メッセージを交換するが、その際に悪意のある第三者が MN に成りすまして Bicast Request を送信し AR にパケットを bicast させ、自身は bicast 先のアドレスを用いてパケットを受信すればパケットの盗聴が可能となる危険性がある。したがって悪意のある第三者による成りすましの被害を防ぐために、MN と AR 間の制御メッセージ送信の際に認証を行うよう、認証機能の追加が必要と考えられる。

文 献

- [1] D. Johnson, C. Perkins and J. Arkko: "Mobility Support in IPv6", RFC 3775, IETF (2004).
- [2] M. Ishiyama, M. Kunishiand, K. Uehara, H. Esaki and F. Teraoka: "LINA: A New Approach to Mobility Support in Wide Area Networks", IEICE Transactions on Communication, E84-B, 8, pp. 2076-2086 (2001).
- [3] R. Koodli: "Fast Handovers for Mobile IPv6", RFC 4068, IETF (2005).
- [4] K. Mitani, R. Shibui, K. Mitsuya, K. Gogo and F. Teraoka: "Unified L2 Abstractions for L3-Driven Fast Handover", Internet Draft, IETF (2006). draft-koki-mobopts-12-abstracts-05.txt.
- [5] 渋谷, 神谷, 寺岡: "レイヤ間情報伝達機構 LIES", 日本ソフトウェア科学会 インターネットテクノロジー研究会第 6 回インターネットテクノロジーワークショップ (WIT2004) 論文集 (2004).
- [6] K. Gogo, R. Shibui and F. Teraoka: "An L3-Driven Fast Handover Mechanism in IPv6 Mobility", Proceedings of SAINT 2006 IPv6 Workshop (2006).
- [7] S. Thomson and T. Narten: "IPv6 Stateless Address Auto-configuration", RFC 2462, IETF (1998).
- [8] T. Narten, E. Nordmark and W. Simpson: "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, IETF (1998).
- [9] 内藤, 後郷, 藤巻, 寺岡: "リンク層情報と近隣情報を利用した IPv6 高速ハンドオーバ手法の実環境における評価", 第 10 回プログラミングおよび応用のシステムに関するワークショップ (SPA X) (2006).
- [10] "Pulse code modulation (pcm) of voice frequencies", ITU-T Recommendation, ITU-T (1998).