

ネットワーク遅延の変化を制御する シームレスハンドオーバー手法の提案と評価

上浦 大智[†] 藤田 貴大^{††} 前田 香織[‡] 相原 玲二^{‡‡}
梶原 大輔^{††} 西村 浩二^{‡‡} 岸場 清悟^{‡‡} 河野 英太郎[‡] 石田 賢治[†]

[†] 広島市立大学大学院情報科学研究科 〒731-3194 広島市安佐南区大塚東 3-4-1
^{††} 広島大学大学院工学研究科 〒739-8527 広島県東広島市鏡山 1-4-1
[‡] 広島市立大学情報処理センター 〒731-3194 広島市安佐南区大塚東 3-4-1
^{‡‡} 広島大学情報メディア教育研究センター 〒739-8511 広島県東広島市鏡山 1-4-2
E-mail: [†] daichi@v6.ipc.hiroshima-cu.ac.jp, ishida@ce.hiroshima-cu.ac.jp
^{††} {takfjt,kjwr}@hiroshima-u.ac.jp, [‡] {kaori,kouno}@ipc.hiroshima-cu.ac.jp
^{‡‡} {ray,kouji,kishiba}@hiroshima-u.ac.jp

あらまし 我々はネットワークモビリティを実現するアーキテクチャとして MAT-MONET を提案してきた。MAT-MONET ではシームレスハンドオーバーのために複数のインタフェースを同時に用いる。しかし、ハンドオーバー時の問題点として、通信の継続性を維持するだけでなく、ハンドオーバー前後でネットワーク遅延が変化することも考える必要がある。そこで、本稿では、ハンドオーバー時の遅延の変化に焦点をあてる。MAT-MONET の複数インタフェースを制御する機構を用いて、各インタフェースでハンドオーバー前後のネットワーク遅延を測定し、適切なパケットのバッファリングを行うことでネットワーク遅延の差を吸収するシームレスハンドオーバーを行う仕組みを提案し、その評価を行う。
キーワード 移動透過通信, モバイルネットワーク, ストリーム伝送, ハンドオーバー, 伝送遅延

A Proposal and Evaluation of a Seamless Handover Controlling Variation of Network Delay

Daichi UEURA[†] Takahiro FUJITA^{††} Kaori MAEDA[‡] Reiji AIBARA^{‡‡}
Daisuke KAJIWARA^{††} Kouji NISHIMURA^{‡‡} Seigo KISHIBA^{‡‡}
Eitaro KOHNO[†], and Kenji ISHIDA[†]

[†] Graduate School of Information Science, Hiroshima City University
^{††} Graduate School of Engineering, Hiroshima University
[‡] Information Processing Center, Hiroshima City University
^{‡‡} Information Media Center, Hiroshima University
E-mail: [†] daichi@v6.ipc.hiroshima-cu.ac.jp, ishida@ce.hiroshima-cu.ac.jp
^{††} {takfjt,kjwr}@hiroshima-u.ac.jp, [‡] {kaori,kouno}@ipc.hiroshima-cu.ac.jp
^{‡‡} {ray,kouji,kishiba}@hiroshima-u.ac.jp

Abstract As an architecture for network mobility, we proposed MAT-MONET. Multiple interfaces are simultaneously used for seamless handover in MAT-MONET. As a problem of the handover, not only interruption of communication but also the influence caused by the difference of transmission delay of each network should be considered. In this paper, we focus on the difference of network delay in the handover. Our approach is to smooth the difference by measuring network delays of each interface in MAT-MONET and buffering packets depending on the measurement. We describe the detail of the smoothing method and its evaluation.

Keyword Mobility support, Mobile Network, Stream Transmission, Handover, Transmission delay

1. はじめに

無線 LAN の普及によりラップトップ PC や PDA などの端末を用いて、いつでもどこでも通信が可能になりつつある。移動しながらレイヤ 3(ネットワーク層)以上の通信を継続するモビリティ(移動透過性)を実現する技術としては、Mobile IPv6[1]、LIN6[2]などが提案されている。さらに、これらホストモビリティを拡張しネットワークに適したネットワークモビリティを実現する NEMO[3]などの開発も進んでいる。我々も、アドレス変換に基づく移動透過通信アーキテクチャとして、MAT(Mobile IP with Address Translation)[4]、そして MAT をネットワークモビリティへ拡張した MAT-MONET(MAT-Mobile Network)[5][6]を提案している。これらのモビリティ技術は、それぞれシームレスなハンドオーバを実現するために様々な工夫がなされている[7][8]。

これらの工夫は、通信の継続性を確保することに焦点を当てているものが多い。しかし、移動しながらの通信は、ネットワークが変化するたびに通信相手との伝送遅延の変化が急激に起こる。このようなネットワーク遅延の変化は、大きなジッタやパケットの順序入れ替えを引き起こし、特に DV などのテレビ会議システムや VoIP などの映像・音声ストリームを伝送するアプリケーションの品質に大きな影響を与える。今後、無線インフラの広帯域化を想定すると、ストリームアプリケーションの利用機会は増えてゆくと予想され、品質劣化を回避するための対策が必要となる。

そこで本稿では、MAT(MAT-MONET)の複数インタフェース対応を利用することで、移動前に移動先ネットワークの通信相手とのネットワーク遅延を把握し、その情報を利用してハンドオーバ時に発生する問題を緩和するシームレスハンドオーバメカニズムを提案する。以下、2章ではベースとなる MAT-MONET アーキテクチャについて述べる。次に、3章ではシームレスハンドオーバを行うための提案機構を述べる。4章で提案機構の実装について述べ、5章で評価と考察を行う。最後に、6章でまとめと今後の課題を述べる。

2. MAT と MAT-MONET

Mobile IPv6 と NEMO の関係のように、MAT[4]がホストの移動透過性を実現しているのに対し、MAT-MONET[5][6]はネットワークの移動透過性を実現する技術である。基本的なアーキテクチャは MAT を継承している。MAT は IPv6 の機能を拡張し、トランスポート層以上の移動透過通信を実現している。図 1 は MAT-MONET の全体像を示す。モバイルルータ(MR: Mobile Router)内には移動透過通信機能を持たない従来ノード(LFN: Local Fixed Node)や MAT を備えたモバイルノード(MN: Mobile Node)が存在する。MR の

マッピングを管理する IMS(IP Address Mapping Server)、IMS のアドレスを管理する DNS があり、さらに、モバイルネットワーク内のノードと通信する通信相手ノード(CN: Correspondent Node)がいる。CN は MAT 機能を持つことを前提としている。

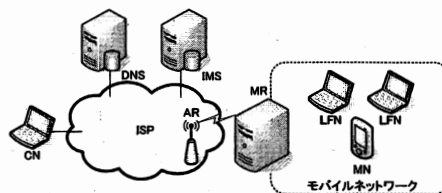


図 1. MAT-MONET 全体像

MAT では、TCP コネクションやアプリケーションプログラムの通信に用いる IP アドレスをノード識別子として、ホームアドレスと呼ぶ。ホームアドレスは MN の識別子で、MN の位置に関係なく固定である。

一方、MN には、接続しているリンクからルータ広告などの手段によって IP アドレスが割り当てられる。このアドレスは MN の位置を示し、これをモバイルアドレスと呼ぶ。MAT は、ネットワーク層においてホームアドレスとモバイルアドレスを変換することにより移動透過通信を実現する。

MAT-MONET を実装した MR は、モバイルネットワーク内にある LFN に対して、移動透過性を与える。MAT では、MN は 128bit のモバイルアドレスとホームアドレスの変換をネットワーク層で行っていたが、MAT-MONET では、MR が MAT のホームアドレスに相当するホームプレフィックスと MAT のモバイルアドレスに相当するモバイルプレフィックスをネットワーク層で変換する。モバイルプレフィックスはアクセスルータ(AR: Access Router)から割り当てられ、64bit のプレフィックス長で IMS で管理される。

MR が新たにネットワークに接続すると、接続したネットワークの AR に対して自身のホームプレフィックスの対となるモバイルプレフィックスを 2 つ要求する。1 つは移動透過性を持たない LFN に、他方は MAT を実装した MN にそれぞれ用いる。IMS に対してこれらのプレフィックスをマッピング情報として登録する。

CN が MN や LFN へパケットを送る場合の処理について述べる。MAT と同様に IMS サーバのアドレスを取得すると、CN がモバイルネットワーク内に所属しているかどうかに関わらず、IMS へホームアドレスをクエリとしてモバイルアドレスの解決を試みる。IMS はホームアドレスの上位 64bit をモバイルプレフィックスに書き換え CN へ返答する。この処理により、相手が MAT ノードか MR 配下のノードかを意識することなく通信可能となる。

以降で複数インタフェースの処理など MAT と MAT-MONET で共通して使われる基本的なアーキテクチャ部分を MAT と呼ぶことにする。

3. シームレスハンドオーバ

3.1. 複数インタフェース

MAT ではシームレスなハンドオーバを実現するために、標準で MN が複数のインタフェースを所持することを前提としている。そこで、MN のマッピング情報には 1 つのホームアドレスに複数のモバイルアドレスを対応づけることができるようにしている。複数のモバイルアドレスが存在する場合は、各モバイルアドレスに優先度を付与して使用優先順位を決定する。これにより、CN が変換に利用するモバイルアドレスを、MN 主導で制御できる。これは、CN からのパケットの受信インタフェースを MN で制御できる、と言いかえてよい。複数インタフェースを用い、一方で通信を行い、他方でハンドオーバに必要な処理を行うことにより、MAT はデータが途切れることのないハンドオーバを実現している。MAT と MAT-MONET における複数インタフェースを用いたハンドオーバ処理の詳細は [6] に委ねる。

3.2. ネットワーク遅延の変化

MAT では、3.1 のように複数インタフェースにより、ハンドオーバにかかる処理の影響をうけることなくシームレスにハンドオーバすることができる。しかし、ハンドオーバ時には移動前後の MN(MR)-CN 間のネットワーク遅延が大きく異なる場合がある。

ネットワーク遅延変化の影響を具体的に調べるため、遅延発生ノードに Lenet[9] を用い、Iperf により測定を行った。Iperf のパラメータは VoIP でよく用いられる G.711 u-law 8kHz Mono を想定し、ペイロード長 172byte、帯域 64kbps の UDP ストリームを流し片側リンクに 500ms の遅延を発生させハンドオーバを行った。測定結果は図 2 と図 3 である。図 2 は高遅延から低遅延方向(500ms→0ms)、図 3 は低遅延から高遅延方向(0ms→500ms)へハンドオーバした際のパケットの流れである。なお、横軸の 0sec がオーバーラップ開始ポイントである。

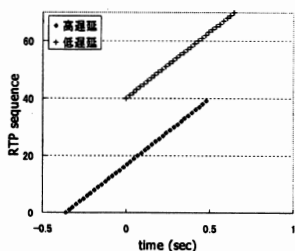


図2.高遅延から低遅延への移動

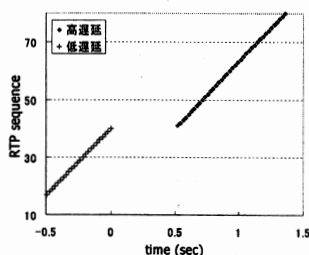


図3.低遅延から高遅延への移動

図 2、図 3 よりネットワーク遅延の変化によって、パケットの順序入れ替えと大きなジッタが生じていることが分かる。これは、ストリームアプリケーションに音飛びや映像の乱れといった品質の劣化につながり、複数インタフェースを用いたハンドオーバだけではこの問題は解決できない。

4. 遅延の差を考慮したハンドオーバ機構

4.1. MAT を用いた遅延情報の事前取得

ハンドオーバによる CN とのネットワーク遅延の変化が引き起こす問題に対処するため、ハンドオーバ後に使用するネットワークの遅延を事前に取得し、遅延の差を吸収するアプリケーションに通知する機構を提案する。

MAT の複数インタフェースを用いてハンドオーバを行うプロセスでは、モバイルアドレスの取得から始まり、IMS へマッピング情報登録、CN へマッピング変化の通知を行うことで新しいモバイルアドレスが使用可能となる。その後、電波強度などから算出されたインタフェースの優先度に基づいて通信に用いるモバイルアドレスが決定されハンドオーバを行う。新たなインタフェースの IPv6 アドレスが使用可能になった直後からハンドオーバを行うまでの間に、MR と CN 間のネットワーク遅延を測定しておく。図 4 を例にとり、情報取得の手順を示す。ここで、図中の HADDR は MR のホームアドレス、MADDR1,2 は MR のモバイルアドレスを意味する。

- 1) MR は新たに発見した AP2 から AP2 側ネットワークで利用する MADDR2 を取得する(図 4-(1)).
- 2) ここで通常の MAT 処理では IMS へマッピング情報の更新をするが、MADDR2 の取得と同時に、MADDR1 を用いて AP1 経由で CN へ(図 4-(1a)), MADDR2 を用いて AP2 経由で CN へ(図 4-(1b))遅延計測用のパケットを送信し、AP1 経由と AP2 経由でのネットワーク遅延(RTT)を計測しておく。
- 3) MR は、MADDR2 を含めたマッピング情報を

IMS へ更新する。この後の処理は MAT や MAT-MONET と同じである。

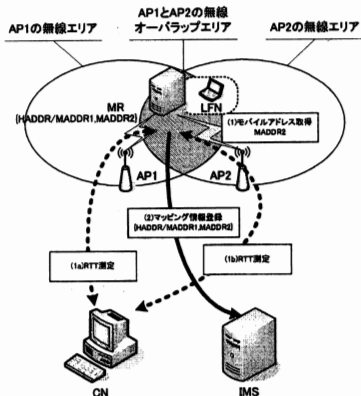


図4. ネットワーク遅延の測定

アプリケーションは取得した遅延情報を用いて必要に応じてハンドオーバーで発生するネットワーク遅延の変化に備える処理を行う。

4.2. 遅延の差の吸収

ネットワーク遅延の事前取得機構を用いて、ストリーム品質の劣化を防ぐ方法について述べる。

一般のストリームアプリケーションは再生バッファを持ち、ネットワークで発生するジッタを吸収している。しかし、テレビ会議や音声通話などのリアルタイム性を要求されるアプリケーションの場合、再生遅延を小さくするために再生バッファのサイズを小さくしようとする。再生バッファサイズは定期的にネットワークの情報に応じて再計算される。しかし、ハンドオーバーによる CN 間の経路の変化に伴う遅延の変動はアプリケーションだけでは予測することはできない。遅延の変化量が大きければ前述した大きなジッタや順序入れ替えが発生し、アプリケーションの再生バッファでは許容できずに品質の劣化を招くこととなる。

そこで、ネットワーク遅延の先読みを行い、ハンドオーバー前に移動先ネットワークの遅延情報を取得し、現在の遅延とハンドオーバー後の遅延の変化量から再生バッファと再生タイミングを調整する。つまり、アプリケーション単体では得られない情報を MAT 側が提供するアプローチとなる。

測定された遅延情報は遅延の差を吸収するために、MAT へ遅延情報の要求を出したアプリケーションゲートウェイ（APPGW）に渡され、吸収のためのバッファリング等の処理が行われる。APPGW については次章で説明する。

MAT から遅延情報を取得した APPGW の対応例を図 5、図 6 に示す。図中の矢印はパケットの流れを、破線

は APPGW によってバッファリングされ送信レートが変更されたパケットを意味している。また、オーバーラップ区間は移動前後のネットワークがオーバーラップしていなくてはならないタイミングを示している。

図 5 は CN からのストリームを受信している MR が高遅延から低遅延ネットワークへの移動を行う際に、APPGW が遅延の差を吸収する様子を表している。MAT 側からハンドオーバー通知を受け取ると、低遅延ネットワークから送られてくるパケットを最初は遅延変動時間だけバッファリングし徐々に実際の送信レートに近づけていく。エンドアプリケーションは再生タイミングを早める。この動作によりパケットの入れ替わりによる影響を防いでいる。

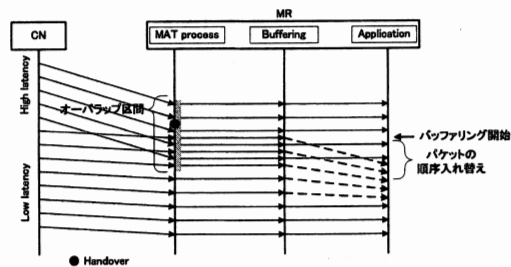


図5. 高遅延から低遅延ネットワークへの移動

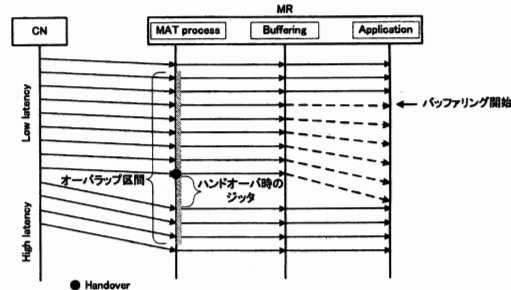


図6. 低遅延から高遅延ネットワークへの移動

逆に図 6 は MR が低遅延から高遅延ネットワークへの移動する際の吸収動作を表している。低遅延ネットワークから送られてくるパケットをハンドオーバー前に APPGW がバッファリングし遅延変動時間を埋めるように送信レートを遅らせる。エンドアプリケーションは再生タイミングを遅らせ、ハンドオーバー時に発生する大きなジッタによる影響を防いでいる。

低遅延から高遅延ネットワークへ移動する状況では、ハンドオーバーを行う前から移動後のネットワークにあわせたバッファリングを行う必要がある。したがって、2 つのネットワークがオーバーラップする時間は高遅延から低遅延ネットワークへの移動と比べて長いことが求められる。

5. 提案機構の実装

前述の提案機構を MAT-MONET 上に実装したシステムのシステムブロックを図 7 に示す。

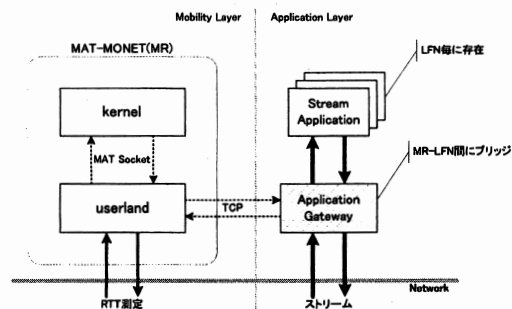


図7.システムブロック

遅延の測定は MAT-MONET のハンドオーバー等を制御する userland 側デーモンが行い、この情報を必要とする APPGW に提供する。

パケットのバッファリングは MR に搭載することもできるが、本稿では図 8 に示すシステム構成の APPGW において、Linux Kernel 2.6.14 の IPv6 スタックを基本に改良し実装したものを示す。

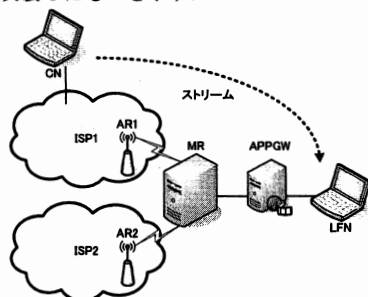


図8.システム構成

MR は、ISP1 から ISP2 へ移動するものとする。CN から送信されるストリームは MR 配下の LFN へ送られる。MR が AR1 を通して通信している際に、AR2 へハンドオーバーするときの、提案機構の動作を以下に示す。

- 1) MR は AR1 経由の CN までの往復遅延時間(RTT) と AR2 経由の RTT を測定する
- 2) RTT の測定終了後、MR は APPGW へ各 RTT 値を通知する
- 3) RTT の情報を元に、APPGW は CN から送信されてくるストリームのパケットがハンドオーバーによるネットワーク遅延の変化の影響を判断し、それに応じたバッファリング処理に入る
- 4) バッファリング準備が完了すると、APPGW は MR へハンドオーバー許可を出す。

6. 評価と考察

6.1. 動作確認

図 9、図 10 は図 2、図 3 と同条件のもとで、提案機構を適用した場合のパケットフローである。図 9 より、ハンドオーバー直後に APPGW で低遅延側経路からのパケットをバッファリングし、徐々に LFN に渡すことで、図 2 のようなパケットの順序入れ替えは解消されている。また、図 10 では図 3 で発生していた大きなジッタは、ハンドオーバー前の低遅延の経路から来るパケットの到着間隔を APPGW でバッファリングし LFN への到着を遅くすることで解消している。

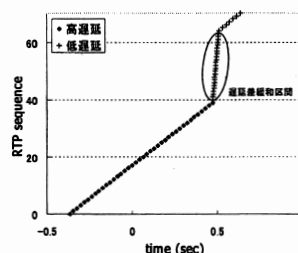


図9.高遅延から低遅延への移動

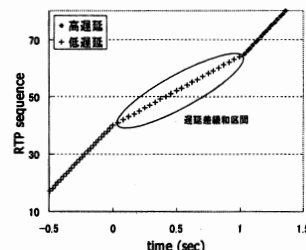


図10.低遅延から高遅延への移動

6.2. 遅延差緩和と直線の傾きに関する考察

図 9 および 図 10 における、遅延差緩和と区間の直線の傾きについて考察する。

図 8 を使って説明する。まず、高遅延(ISP1)から低遅延(ISP2)方向への移動、ハンドオーバー後に ISP2 から届くパケットをバッファリングし、一定時間経過後に LFN へ渡すという処理をしている。ISP2 からくる 1 つめのパケットのバッファリング(遅延)時間 d_0 は以下で表される。

$$d_0 = \frac{|RTT_2 - RTT_1|}{2} \quad (RTT_1, RTT_2: \text{ISP1, ISP2 経由の RTT})$$

n 番目のパケットのバッファリング時間 d_n は、

$$d_n = d_{n-1} - \Delta t \quad (\Delta t: \text{バッファリング時間の変化量})$$

となり、 $d_n = 0$ でバッファリング終了となる。ここで Δt のとりうる値は、

$$0 < \Delta t \leq \frac{1}{pps} \quad (\text{pps:1 秒間のパケット送信数})$$

である。pps は、緩和の対象となるストリームにより異なる値となる。

実験で用いた Iperf のパラメータの場合、pps は約 50 だったので、 Δt は、

$$0 < \Delta t \leq 0.02s (= 20,000\mu s)$$

の値をとる。 Δt が大きな値になるほど、低遅延方向への移動の傾きが急になり収束時間が短くなる。 LFN のストリームを受信しているアプリケーションの再生バッファがあふれない範囲で、大きな値をとる方が収束がはやくなる。 6.1 の実験では Δt に $20,000\mu s$ を設定している。

次に、逆方向である低遅延側 (ISP1) から高遅延側 (ISP2) 方向の場合、ISP1 からくる 1 つ目のパケットのバッファリング時間 d_0 は、 $d_0 = 0$ となる。 n 番目のパケットのバッファリング時間 d_n は以下である。

$$d_n = d_{n-1} + \Delta t$$

そして、

$$d_n = \frac{|RTT_2 - RTT_1|}{2}$$

でバッファリング終了となる。 Δt のとりうる値は前述と同様である。 この場合も収束時間を早めるために大きな値をとる方が好ましい。 6.1 の実験では Δt に $20,000\mu s$ を設定した。

6.3. 提案機構の処理時間

提案機構を稼働させた際にかかるハンドオーバー時間の増加を測定したものを表 1 に示す。 また、ハンドオーバー時間の内訳を表 2 に示す。 これらは、提案機構を稼働させ、片側リンクに 200ms、500ms の遅延を発生させた場合のハンドオーバーにかかる時間を測定した。 表 2 において、MAT はハンドオーバー時の MAT の処理時間、RTT は RTT 測定に必要な時間を示す。 測定結果は 10 回の平均値である。

表1.提案機構のハンドオーバー時間(ms)

遅延差	200ms	500ms
高→低	446.5	1042.4
低→高	665.2	1582.0

表2.ハンドオーバー時間の内訳(ms)

遅延差	200ms		500ms	
	MAT	RTT	提案機構	提案機構
高→低	30.6	400	15.9	11.8
低→高	30.6		234.6	551.4

表 2 より遅延時間を増加させると、高遅延から低遅延方向の場合は RTT による差はない。 一方、低遅延から高遅延方向の場合は MR がハンドオーバーを行う前に RTT に合わせたバッファリング処理が必要なため、RTT に比例して処理時間が大きくなっている。

遅延差 200ms のリンクにおいて低遅延から高遅延方向へハンドオーバーした場合のハンドオーバー時間は表 1

より 665.2ms なので、例えば時速 80km の移動の場合、14.8m の無線オーバーラップ区間があればよい。 アクセスポイントの配置にもよるが、無線 LAN のカバーするエリアを考慮すると実用的な値と考えられる。

7. まとめ

本稿では、MAT-MONET アーキテクチャと、ノードが複数インタフェースを所持することを利用し、ネットワーク遅延を考慮したシームレスハンドオーバー機構の設計と実装について述べた。 今後は、ネットワーク遅延以外のネットワーク特性の変化も考慮するとともに、本提案機構を Multiple CoA[10] の拡張を施した Mobile IPv6 や NEMO などにも適用していきたい。

謝辞

本研究に関して御助言を頂きました、ネットワンシステムズ株式会社 杉本康則氏、赤座正樹氏に深く感謝致します。 本研究の一部は広島市立大学平成 17 年度特定研究費 (5115)、日本学術振興会科学研究費補助金 (17300019, 17500037) の支援を受けて実施している。 ここに記して感謝の意を示す。

参考文献

- [1] Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, RFC3775 (Proposed Standard) (2004).
- [2] Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H. and Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks, IEICE Transaction on Communication, Vol.E84-B, No.8, pp.2076-2086 (2001).
- [3] Devarapalli, V., Wakikawa, R., Petrescu, A. and Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, RFC3963 (Proposed Standard) (2005).
- [4] 相原玲二, 藤田貴大, 前田香織, 野村嘉洋: アドレッシング変換方式による移動透過性インターネットアーキテクチャ, 情報処理学会論文誌, Vol.43, No.12, pp.3889-3897 (2002).
- [5] 藤田貴大, 野村嘉洋, 西村浩二, 前田香織, 相原玲二: MAT によるモバイルネットワークの実現, DICO2003, pp.105-108 (2003).
- [6] 上浦大智, 藤田貴大, 前田香織, 相原玲二, 梶原大輔, 西村浩二, 岸場清悟: ネットワーク特性の変化を考慮したシームレスハンドオーバーの実現, インターネットコンファレンス 2005, pp.53-61 (2005).
- [7] Koodli, R.: Fast Handovers for Mobile IPv6, RFC4068 (Experimental) (2005).
- [8] Soliman, H., Castelluccia, C., Malki, K. E. and Bellier, L.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6), RFC 4140 (Experimental) (2005).
- [9] 石野正英, 前田香織, 河野英太郎, 岸田崇志: パケットロスパターン再現可能なネットワークエミュレータの開発, 信学技報, Vol.105, No.219, pp.13-18 (2005).
- [10] Wakikawa, R., Uehara, K., Ernst, T. and Nagami, K.: Multiple Care-of Addresses Registration, Internet-draft, draft-wakikawa-mobileip-multiplecoa-04.txt (2005).