

リンク層情報を利用した高速ハンドオーバ支援機構の設計と実装

神谷弘樹^{†1} 國司光宣^{†2} 寺岡文男^{†1}

^{†1} 慶應義塾大学理工学部情報工学科 ^{†2} 慶應義塾大学大学院理工学研究科

1 はじめに

近年、モバイルコンピューティングが一般的になり、IETF では Mobile IP の標準化が進んでいる。ノードが通信しながら移動する場合、Internet 上のある LAN から別の LAN へ接続を切り替えるハンドオーバ処理を高速に行う必要がある。しかし、ネットワーク層のハンドオーバ (L3 ハンドオーバ) 処理を高速化しただけでは最適なハンドオーバを実現できない。なぜなら Internet Protocol は機能ごとに独立して開発できるように階層構造をなしており、ネットワーク層ではデータリンク層でのハンドオーバ (L2 ハンドオーバ) に関する情報を得ることができないためである。

この問題を解決するために、リンク層情報を利用した高速ハンドオーバプロトコル Low Latency Handoffs in Mobile IPv4[1]、Fast Handovers for Mobile IPv6[2] が提案されている。これらのプロトコルでは L3 で扱えるように抽象化したリンク層情報 (L2 Trigger) を動的に伝達する方式を用いてハンドオーバを高速化している。しかし、実際にデータリンク層情報の抽象化や、動的に情報を伝達する手法については未定義である。

本研究では高速ハンドオーバの実現に必要な不可欠なリンク層情報の扱い方を定義するために、レイヤ間情報交換方式の設計と実装、リンク層情報の抽象化を行う。そしてレイヤ間情報交換方式の応用例として、リンク層情報を利用したハンドオーバ検知機構と無線基地局探知機構を実装し、情報伝達に要する時間などの基本性能評価を行い、将来のノード主導高速ハンドオーバの実現に貢献することを目的とする。

2 設計

2.1 レイヤ間情報交換アーキテクチャ

高速ハンドオーバを実現するためには少なくとも L2 から L3 への情報伝達が必要である。しかし本研究ではレイヤ間協調の幅広い必要性を考慮し、隣接しないレイヤ間でも情報を交換できる図 1 のようなアーキテクチャを提案する。

本方式で情報交換を行う際には、レイヤ構造の破壊を防ぐために各レイヤが持つ情報は抽象化されている必要がある。したがって各レイヤに Abstracted Entity (AE) を設ける。情報交換方式 Inter-layer System (ILS) は各レイヤとの間に Service Access Point (SAP) を持つような構造となっており、非隣接レイヤ間の抽象化情報の交換をサポートする。

各 AE と ILS の間の SAP は、AE がサービスプロ

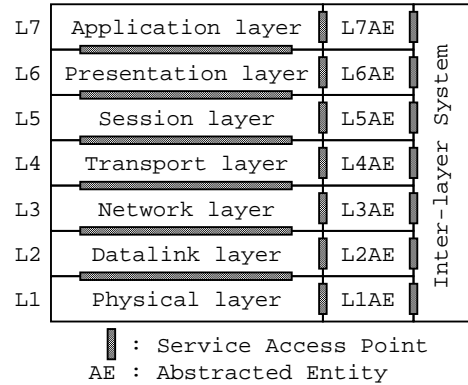


図 1: Abstracted Entity and Inter-layer System

バイダとなりユーザである ILS に対してサービスを提供するというモデルである。さらに、ILS は AE 間の仲介役となりサービスを中継する。また、サービスの定義は情報を抽象化したプリミティブという形で表すことができる。プリミティブは図 2 に示すとおり、プロバイダに対する要求 (request) とユーザに対する確認 (confirm)、ユーザに対する知らせ (indication) とプロバイダに対する返事 (response) の 4 種類のタイプに分けられる。AE 間でのコミュニケーションはすべてこの 4 種類のメッセージと付属パラメータを交換することで可能となる。

RX-Primitive.Primitive_Type(parameters)

RX: service provider entity(e.g.,MAC)
Primitive: The name of primitive(e.g.,Data)
Primitive_Type: Request, Indication, Response, Confirm
parameters: A list of parameters for the primitive

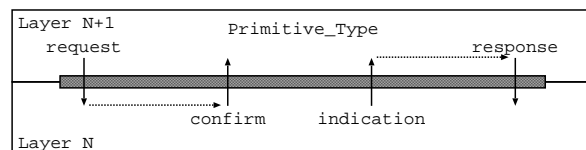


図 2: Primitive

2.2 リンク層情報の抽象化

本研究では、モビリティサポートプロトコルの位置する L3 の AE へ高速ハンドオーバを支援する情報を提供するために、リンク層情報を分類、抽象化し、L2AE が提供するサービスを設計する。

高速ハンドオーバ支援という観点から主に無線インタフェースに焦点を当て、IEEE802.3 (Ethernet)、

IEEE802.11 (無線 LAN)、W-CDMA (3GPP)、cdma2000 (3GPP2)、PIAFS という 5 種類のリンク層情報を分類した。ハンドオーバーのきっかけのような極めて動的な情報が必要なことから、リンク層情報の分類はネットワーク管理用データベースである MIB に加えて隣接レイヤ間のインタフェースについても行った。隣接レイヤ間のインタフェースは図 2 に示すプリミティブという形で各ネットワークインタフェース毎に標準化されている。高速ハンドオーバーに必要な情報をプリミティブという形で抽象化するための、L2AE が提供するサービスの大きな分類を表 1 に示す。

表 1: Services of L2 Abstracted Entity

L2AE Service	L2AE Primitive Type
情報の取得	request, confirm
情報の設定	request, confirm
通知すべきイベントの登録	request, confirm
イベント発生時の通知	indication, response

設計した L2AE は大きく分けて 8 種類ある。ハンドオーバーで次に接続する候補となる接続先情報を提供する L2AE-CAP、接続先の発見通知を行う L2AE-CAPFIND、接続先の喪失通知を行う L2AE-CAPLOST、ハンドオーバー完了を通知する L2AE-HOAFTEER、ハンドオーバー開始の兆候を伝達 L2AE-HOBEFORE、ハンドオーバー開始の要求を受け付ける L2AE-HOSTART、リンク状況を動的に伝達する L2AE-LINKEV、リンク状況を要求に応じて伝達する L2AE-LINKSTAT である。付属パラメータや分類した 5 種類のリンク層情報との対応は紙面の都合上割愛する。

3 応用例

3.1 Horizontal Handover の検出

Horizontal Handover とは単一のネットワークインタフェースの接続先を切り替えるハンドオーバーのことである。従来のモビリティサポートプロトコルはリンク層情報を利用できないため、図 3 に示すように L2 ハンドオーバーが完了した後、L3 ハンドオーバー処理開始までの遅延が生じていた。このようなハンドオーバー処理は、レイヤ間情報交換方式と抽象化されたリンク層情報を応用して最適化することができる。

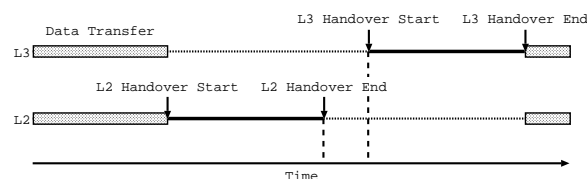


図 3: L2 and L3 Handover

解決方法は、まず最初に L3AE が L2AE-

HOAFTEER.request を利用して L2 ハンドオーバー通知要求を行う。次に L2AE は通知要求の確認を L2AE-HOAFTEER.confirm によって L3AE に伝える。これらの準備を終えた後、L3AE は L2AE からの L2AE-HOAFTEER.indication を待つ。ケーブルの差し替えなどの L2 ハンドオーバーが発生し、ケーブルの抜き差しが完了した時点で L3AE は L2AE-HOAFTEER.indication を受信し瞬時に L3 ハンドオーバー処理を開始する。こうすることで図 4 に示すように遅延がほとんど生じず処理を継続することができる。L2 ハンドオーバーの完了通知を利用する本解決手法はハンドオーバー処理の最適化だけでなく TCP の輻輳制御を解決するなどのさらなる応用が期待される。

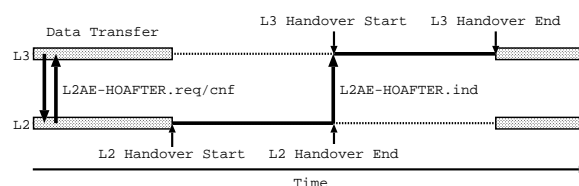


図 4: Optimized Handover

3.2 L3 主導 Horizontal Handover

本節では、通信中に前もって L3 ハンドオーバーの準備処理をしてからモバイルノードの L3 主導で L2 ハンドオーバーを行う方法をレイヤ間情報交換方式を応用して実現する。この方法は無線インタフェースでは通信途絶時間を短縮できる有効な方法であり、Low Latency Handoffs in Mobile IPv4 や Fast Handovers for Mobile IPv6 でも提案されている。

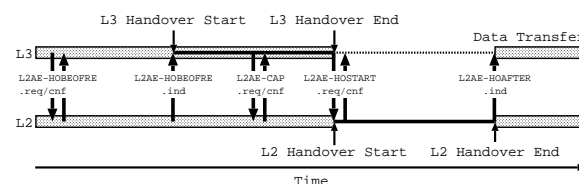


図 5: Anticipated Horizontal Handover

ノード主導 Horizontal Handover の手順を図 5 に示す。この方式では L2 ハンドオーバーが必要であるということを示す抽象化したリンク層情報、L2AE-HOBEFORE.indication を L3 ハンドオーバーのきっかけとして利用する。あらかじめ L2AE-HOBEFORE.request/confirm で通知機能を有効にしておき、L2AE-HOBEFORE.indication をきっかけに L3 ハンドオーバー処理を行う。L3 ハンドオーバー処理に次の接続先の情報が必要となる場合、処理中に L2AE-CAP.request/confirm で情報を得ることができるほか、L2AE-CAPFIND/CAPLOST などを利用して継続した情報収集も可能である。そして、L3

ハンドオーバー処理の最後の手順が L2 へのハンドオーバーの開始要求 L2AE-HOSTART.request/confirm である。この後しばらく通信が途絶し、L2 ハンドオーバー完了とともに通信が可能になる。

3.3 Vertical Handover

モバイルノードが複数のネットワークインタフェースを持ち、それらを状況に応じて切替えることを Vertical Handover という。このハンドオーバー形態では L2 が複数存在し、L3 の判断で L2 を選択する機能が必要である。

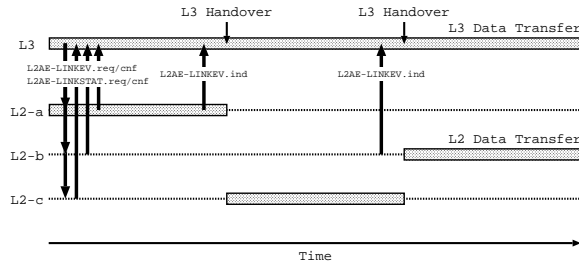


図 6: Vertical Handover

Vertical Handover の手順を図 6 に示した。L3 はあらかじめ切り替えのポリシーを持っていることにする。ハンドオーバーの前には判断材料となる各 L2 のリンク状況を知っておく必要があるため、L2AE-LINKSTAT.request/confirm を利用して状態を確認する。また、その後の状態の変化をイベントとして受け取るために L2AE-LINKEV.request/confirm で条件指定とともに通知要求を行う。

図 6 に示した例では、最初に L3 は L2-a が最適と判断した状態から始まる。その後、L2-a から L2AE-LINKEV.indication によって L2-a は好ましくないリンク状況になったことが L3 に通知され、L3 は以前得ていた情報から L2-c が現時点で最良の選択であると判断する。またしばらく時間が経過した後、今度は L2-b のリンク状況が改善したという通知 L2AE-LINKEV.indication が発生し、L3 は L2-b を利用すると決定する。このように、L3 は複数の L2 を L3 ハンドオーバーによって切り替えることで単一のネットワークインタフェースに縛られることなく通信を継続できる。また、Vertical Handover と Horizontal Handover が同時並行的に発生する場合や複数のネットワークインタフェースを同時に使用する場合には今後の課題とする。

4 実装

4.1 実装環境

OS には 2003 年 1 月現在の NetBSD1.6L-current の上にモビリティサポートプロトコル LING が実装されている kernel を用い、レイヤ間情報交換方式を実装した。

4.2 レイヤ間情報交換方式

OS では厳密なレイヤの境界はないが kernel と userland は厳密に区別されており、L3 の機能は部分的に userland の daemon として実装されていることがあるため、レイヤ間情報交換方式を kernel-userland 間の動的な情報交換方式として実装した。実装は kernel と userland 間の動的な情報交換を socket を介して行う MSOCK と、kqueue を介して行う MITAC という 2 種類の方法で行った。MSOCK と MITAC ではイベント通知の扱い方が異なるが、双方とも動的な kernel 状態の変化を迅速に userland に伝達する機能を持ち、情報の抽象化機能は kernel 内の各モジュール単位で情報を加工する実装である。

4.2.1 MSOCK

図 7 が MSOCK の構造である。kernel 内の各モジュールはイベントが発生すると mssock_trap() 関数を呼び、MSOCK にイベント発生のお知らせと付随情報の伝達を行う。そして MSOCK は socket を介して kernel で発生したイベントを userland へ伝達する。MSOCK のメッセージヘッダはイベントの発生場所、イベントの種類、付随情報の大きさを含んでおり、可変長の付随情報がメッセージ本体という扱いになっている。

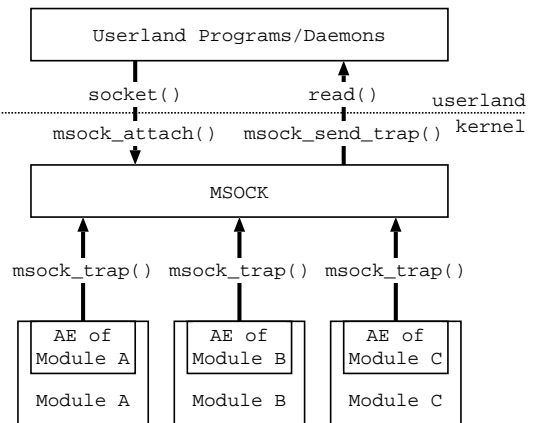


図 7: MSOCK

4.2.2 MITAC

MITAC は既存の kqueue をネットワークインタフェースに応用した実装である。kqueue とは kernel 内のイベントを userland に効率良く通知する機構であり、イベント登録方式、通知機構の効率性、拡張性などの特徴を備えている。

図 8 が MITAC の構造である。あらかじめ userland のソフトウェアは kevent() でイベント通知の種類を登録し、kernel 内には kqfilter() でイベント通知の登録が行われる。kernel 内の各モジュールは KNOTE() でイベント通知を kernel event queue (kqueue) に入れ、ソフトウェアは通常の kqueue の使用方法でイベント通知を kqueue から取り出すこと

ができる。MITAC はイベントの付属情報を伝達する方法を持たず、必要に応じて別途 `ioctl()` などで行う必要がある。

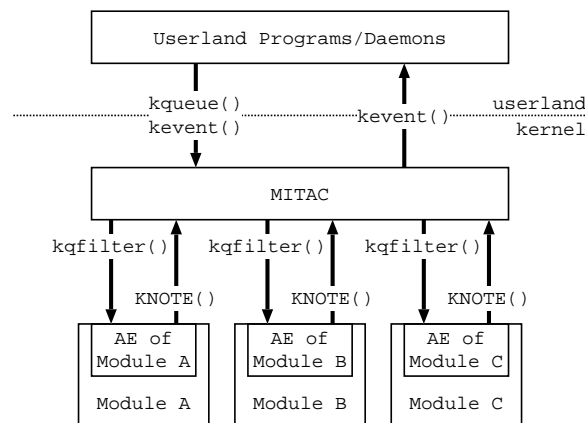


図 8: MITAC

4.3 応用例

4.3.1 Horizontal Handover の検出

MSOCK、MITAC で可能な L2 ハンドオーバー完了通知 (L2AE-HOAFTEER.indication) を応用し、`rt-sold`、`lin6resolvd` という既存のソフトウェアを改良し、高速ハンドオーバーに有用な機構を実装した。L2AE-HOAFTEER.indication は、有線ではケーブルが差し込まれた時、無線 LAN では接続先の Access Point (AP) が切り替わった時に発生するように kernel の各デバイスドライバ内で抽象化して実装した。改良は `rt-sold` では IPv6 通信の開始準備、`lin6resolvd` ではモビリティサポートプロトコル LIN6 の L3 ハンドオーバー処理が開始するタイミングを L2 ハンドオーバー直後になるように行った。

4.3.2 L3 主導 Horizontal Handover

MSOCK、MITAC で可能な、MN がハンドオーバーする必要があるという通知 (L2AE-HOBEFORE.indication) を応用し、基地局探知機構を実装した。この機構は L3 ハンドオーバー処理を終わってから L2 ハンドオーバーを行う方式に有用である。L2AE-HOBEFORE.indication は、有線では不可能なため無線 LAN での電波強度がある閾値を下回った時に発生するようにデバイスドライバ内で抽象化した。さらに kernel の無線 LAN デバイスドライバを定期的に基地局スキャンを行うように改良し、userland へ毎回のスキャン結果と基地局発見通知、基地局喪失通知を伝達できるようにした。これらの実装は L2AE-CAP.request/confirm、L2AE-CAPFIND.indication、L2AE-CAPLOST.indication に相当する。

userland のソフトウェアではこれらのリンク層情報を受信し、任意のポリシーを持つことでハンドオーバーの開始時期と次の接続先の決定を行うことができる仕様となっている。基地局を MN 主導で切替

える機能 (L2AE-HOSTART.request) も実装する予定であったが、無線 LAN ネットワークインタフェースのハードウェア仕様の問題で実装することができなかった。

5 評価

5.1 レイヤ間情報交換方式

本研究ではレイヤ間情報交換方式 MSOCK と MITAC を実装することにより、kernel と userland 間の動的な情報交換を実現した。情報交換に socket を用いた MSOCK の特徴は、任意の付属情報をイベント通知と同時に伝達できる点である。kqueue を用いた MITAC の特徴は既存の kqueue にイベント通知機能があるため実装が容易な点である。モバイル Pentium III-M 800MHz、メインメモリ 640MB のノート PC で測定したところ、どちらの方式も情報交換には約 2msec を要した。また、リンク層情報の抽象化を具体的に定義したことにより、インターフェース間の差異を意識することなくイベントの通知を行うことが可能となった。

5.2 応用例

MITAC と MSOCK の応用例として、Horizontal Handover に関する 2 例を設計、実装し、Vertical Handover については設計のみを行った。実装例では L3 主導でのハンドオーバー開始や、L2 ハンドオーバー完了後の処理の最適化など、高速ハンドオーバーに不可欠な L2 と L3 の協調を実現した。また、レイヤ間の協調を実現することにより Vertical Handover に必要な情報を提供できるということが判明した。

6 まとめ

本研究ではノード主導の高速ハンドオーバーを実現するために必要なリンク層情報の扱い方を議論した。MSOCK と MITAC を設計、実装してレイヤ間の動的な情報交換を実現したことに加え、各ネットワークインタフェースのリンク層情報の分類と抽象化を行い L3 への動的な情報提供を実現した。さらに、これらを実装することでハンドオーバー処理を最適化することができた。

今後の課題は、MSOCK と MITAC を応用し、MN による L3 主導高速ハンドオーバープロトコルを設計し実装することである。また、レイヤ間情報交換のモバイルコンピューティングへの応用としてはリンク層情報による TCP の最適化、Vertical Handover の実現などが挙げられる。

参考文献

- [1] MIPv4 Handoffs Design Team, "Low Latency Handoffs in Mobile IPv4", Internet Draft, IETF, Jun. 2002. work in progress.
- [2] R. Koodli et al., "Fast Handovers for Mobile IPv6", Internet Draft, IETF, Sep. 2002. work in progress.