

移動透過通信支援機能のユーザランドにおける実装

浮田悠太[†] 林直樹[‡] 大石恭弘[‡] 前田香織[‡] 近堂徹[§] 相原玲二[§]

[†]広島市立大学情報科学部 [‡]広島市立大学大学院情報科学研究科

[§]広島大学情報メディア教育研究センター

概要：MIPv6 (Mobile IPv6) など異種ネットワークを渡り歩いても通信途絶のない移動透過通信を実現するアーキテクチャでは従来端末のプロトコルスタックにモビリティ機能を搭載することで移動透過通信を支援している。著者らが開発しているアーキテクチャ MAT (Mobility support Architecture and Technologies) も同様である。こうした端末のプロトコルスタックの改変は移動透過通信の普及の障害になっている。本稿では、この問題を解決するため、主要な機能をユーザランドにおいて実装した移動透过通信 MATについて述べる。

Userland Implementation of an IP Mobility Support Function

Yuta Ukida[†] Naoki Hayashi[‡] Yasuhiro Ohishi[†] Kaori Maeda[‡]
Tohru Kondo[§] Reiji Aibara[§]

[†]Faculty of Information Sciences, Hiroshima City University

[‡]Graduate School of Information Sciences, Hiroshima City University

[§]Information Media Center, Hiroshima University

1. はじめに

近年の移動端末は、Wi-Fi, LTE などといった複数のネットワークを状況に合わせて切り替えて利用しているものが一般的である。その中で、異なるネットワーク間を通信が途切れることなくシームレスにハンドオーバし、IP 通信が継続できる移動透過通信の必要性が高まっている。これを解決するため、IP 層における移動透过通信機能 (IP モビリティ) が提案されている。例えば、MIPv6 (Mobile IPv6) [1] や著者らの開発している MAT (Mobility support Architecture and Technologies) [2] がある。これらのアーキテクチャを用いて端末に移動透过通信機能をもたせるためにはプロトコルスタックに IP モビリティを搭載する必要があり端末をカーネルレベルで変更しなければならない。これは利用の難易度が高いという問題がある。そこで端末への導入を容易にするため MAT をユーザランドで実装する。また他の OS 等への移植性についても検討する。

ユーザランドでの実装によって、カーネルでの処理に比べて処理時間が大きくなる可能性もあるため、処理性能に関する評価方法も検討する。ユーザランドで移動透过性を実現する技術として、仮想 IP アドレスとトンネリング技術を用いて移動透过性を実現するための NTMobile[3] が提案されており、Android 端末にも移植されている[4]。NTMobile の性能評価については、ハンドオーバ時に発生する通信途絶時間やパケットロスに基づいている。ユーザランドで実装した MAT についてもこれらを考慮して評価し、比較していく。

2. MAT によるアドレス変換

MAT はノードがネットワークを越えて移動しても同一アドレスを使用して通信を開始できるインターネット上のアーキテクチャである。MAT では端末を識別するアドレスであるホームアドレス (HoA) と、端末の位置を識別するアドレスであるモバイルアドレス (MoA) を相互に変換することによりモビリティを確保する。このマッピングのため、移動端末 (Mobile Node : MN) 及びその通信相手端末 (Correspondent Node : CN) は IP Address Mapping Table (IMT) を持つ。MN と CN は、HoA と MoA のマッピング情報の管理や配信を行う IP Address Mapping Server (IMS) から MN のマッピング情報を呼び出し、自身の持つ IMT の情報を更新する。MN のマッピング情報はマッピングの更新を IMS に送信することで反映される。この更新のタイミングは、MN が異なるネットワークへ移動したときである。IMS は MN の MoA を更新するためだけに存在し、MN と CN 間の全ての通信は IMS を介さないという点がホームエージェントと大きく異なる。

HoA と MoA の変換は、図 1 で示すようにネットワーク層で IP 処理の前後に行われる。デバイスから入ってきたパケットは IMT を参照することで、送信元 IP アドレスを MoA から HoA に変換して上位層に渡す。逆に上位層から渡された HoA は IMT を参照し、宛先を MoA に変換して下位層に渡す。トランスポート層ではコネクション生成時に送出インターフェース選択を行なうが、

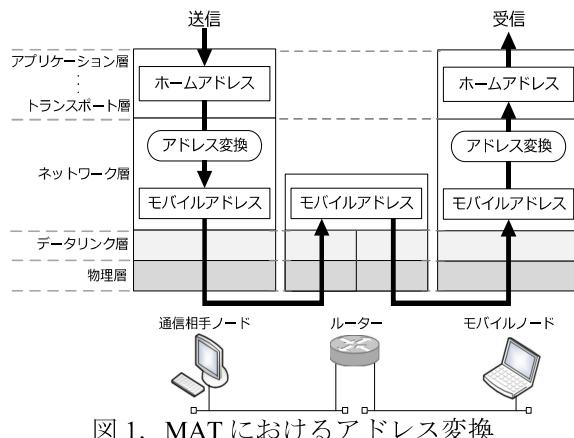


図1. MATにおけるアドレス変換

トランスポート層に渡される宛先 IP アドレスは HoA なので、HoA に基づくインターフェース選択が行われる。そこで MAT ではアドレス変換後にインターフェースの再選択を行う。以上のアドレス変換により、MAT では最短経路による通信を可能にしている。しかし、カーネルでのアドレス変換の実装のため、利用の難易度が高くなってしまっている。

3. ユーザランドでの実装方針

基本的な通信の手続きの流れは従来の MAT と同じだが、MATにおいてカーネル内で行われていた処理をユーザランドで実装する。想定するパケット受信時および送信時の処理の流れを図2に示す。MATでは、ネットワーク層で HoA と MoA の変換を行い、移動透過性を実現している。このアドレス変換の処理およびそれに伴うトランスポート層ヘッダのチェックサム計算やインターフェースの再選択をアプリケーション層で実装する。

ユーザランドでこの機能を実現するために、TUNデバイスを用いる[5]。TUNデバイスはネットワーク層をエミュレートできる仮想ネットワークデバイスであり、これにより IP パケットを操作することができる。TUNデバイスで提供される仮想インターフェースを通ったパケットをユーザプログラムで扱うことができる。このとき、MoA宛てに届いたパケットについては TUN の仮想インターフェースを通るようにルーティングを追加する。また、TUN の仮想インターフェースを通ったパケットに含まれるアドレスを HoA と MoA で相互に変換する。この変換の後に、作成したパケットを TUN の仮想インターフェースに対して送信する。ここで、TUN の仮想インターフェースに自身の HoA を付与しておくことで、受信パケットの場合には変換後の HoA 宛てのパケットを自分宛てのパケットだと認識でき、そのまま上位層に向かって処理される。

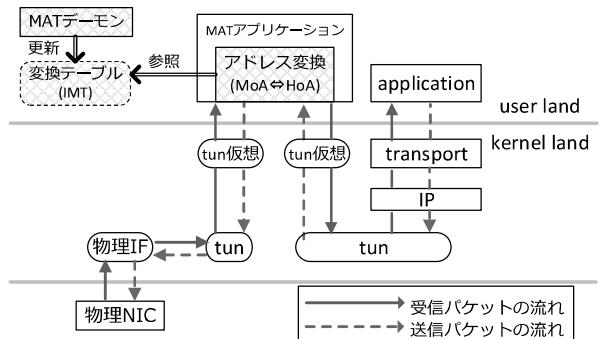


図2. パケットの流れ

TUNデバイスはLinux系OSやWindowsを含む様々なOSに実装されているので、使用するOSに依存することなく、幅広い端末で利用することが可能となる。

従来のMATではIMSからマッピング情報を取得するため、ユーザランドで動作するアプリケーションデーモン(MATデーモン)を利用していた。そのため、IMTの管理、およびIMSとの通信には、MATデーモンの仕様に変更を加えたものを利用する。すなわち、従来のMATデーモンはカーネルとのやりとりを前提としていたが、これを実装するMATアプリケーションとのやりとりに対応させるように変更する。

4. おわりに

現在、MATユーザランドでの実装方針に沿って、実装を進めている。TUNデバイスを用い、ユーザプログラムで処理を行うためオーバヘッドが生じる可能性がある。このオーバヘッドをなるべく減らし、アプリケーションで許容できる範囲内に留めることが重要である。そのため、ユーザプログラムでの処理を最適化する必要がある。動作検証を行った後、既存のMATや関連研究との通信性能の比較をし、評価を行う。

謝辞

本研究にあたって、実装について意見を頂いたMATプロジェクトメンバーの皆様に感謝します。なお本研究の一部は日本学術振興会科学研究費助成金24300027, 24500083の支援を受けて実施しています。

参考文献

- [1] D.B.Johnson, C.E. Perkins, and J. Arkko, "Mobility Support in IPv6," Internet Engineering Task Force (IETF), RFC3775, June 2004.
- [2] 相原玲二, 藤田貴大, 前田香織, 野村嘉洋, "アドレス変換方式による移動透過インターネットアーキテクチャ," 信学技法, Vol.43.12, pp.3889-3897, Dec.2002.
- [3] 内藤克浩, 西尾拓也, 水谷智大, 鈴木秀和, 渡邊晃, 森香津夫, 小林英雄, "NTMobileにおける移動透過性の実現と実装," DCOMO2011論文集, Vol.2011, No.1, pp.1349-1359, 2011.
- [4] 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊晃, "NTMobileのAndroid端末への実装と評価," モバイルコンピューティングとユビキタス通信研究報告, vol.2012-MBL-62, No.19, pp.1-8, 2012.
- [5] "Universal TUN/TAP device driver," The Linux kernel Archives, <https://www.kernel.org/doc/Documentation/networking/tun.txt>, cited 2014-09-02.