

移動透過保証プロトコル LIN6 の概要とソフト公開

國司 光宣
慶應義塾大学 理工学部

寺岡 文男
ソニーコンピュータサイエンス研究所

1 LIN6 の概要

モバイルコンピューティングにおいては移動透過な通信機能が必要となる。通信機能における移動透過性とは以下の2つの性質を有することである。1つはノードの位置に依存せずノードとの通信を開始できることであり、もう1つはノードが移動してもノードとの間に確立したコネクションが維持されることである。

LIN6 は IPv6 上で移動透過な通信をネットワーク層で実現するためのプロトコルである。既存のネットワーク層プロトコルでは IP アドレスはノードの位置を表すだけでなく、ノードの識別子としても利用される。移動によりアドレスが変化するとノードの位置情報だけでなくノードの識別子まで変化してしまう。したがって、コネクションを維持できない。そこで LIN6 は IPv6 のアドレス構造を利用し、位置指示子と位置に依存しないノード識別子とに分離して考える。

LIN6 は従来の IPv6 アドレスの他に、LIN6 アドレスと呼ばれる 128bit のネットワークアドレスを導入する。LIN6 アドレスの下位 64bit は、IPv6 におけるインタフェース識別子の代わりにグローバルユニークなノード識別子を埋め込む。このノード識別子は位置に依存せず、かつ各ノードに固有な値である。上位 64bit は IPv6 と同様にネットワークプレフィクスであり、位置に依存する。一方、ネットワーク層より上位層では下位 64bit のノード識別子を上位層での識別子として利用することにより移動不変な識別子となる。しかしながら、上位層での識別子を 64bit にすることは既存の IPv6 との互換性の問題から現実的ではない。そこで LIN6 では、LIN6 プレフィクスと呼ばれる 64bit の固定値を導入し、この固定値を上位 64bit に挿入することにより 128bit の上位層での識別子とする。この上位層の識別子は LIN6 汎用識別子と呼ばれ、前述の通り移動に不変な識別子となる。アプリケーションはこの LIN6 汎用識別子を利用して通信することで移動透過性が保証される。

実際にパケットを通信相手に配送するためには、通信相手のネットワーク上の現在の位置を知る必要がある。LIN6 では、ノード識別子と現在の位置指示子との対応づけ (*Mapping*) を管理するために *Mapping Agent* (MA) を導入する。LIN6 を使用するノード (ここでは MN とする) は、ネットワーク上を移動するたびに自分の MA に対して Mapping を通知する。ある LIN6 汎用識別子を持つノード (ここでは CN とする) と通信する場合、MN は CN のノード識別子を鍵として MA に問い合わせ、現在の CN の位置指示子を取得する。

2 LIN6 の実装

LIN6 の実装は図1で示すように MN のユーザ空間で動作する Mapping Resolver と Motion Detector、ユーザ空間とカーネル空間との通信をするための LIN6 socket、カーネル空間内の Mapping のキャッシュである Mapping Table と LIN6 の通信のための関数群から構成される。また、Mapping Agent はユーザ空間の独立プログラムとして実装した。

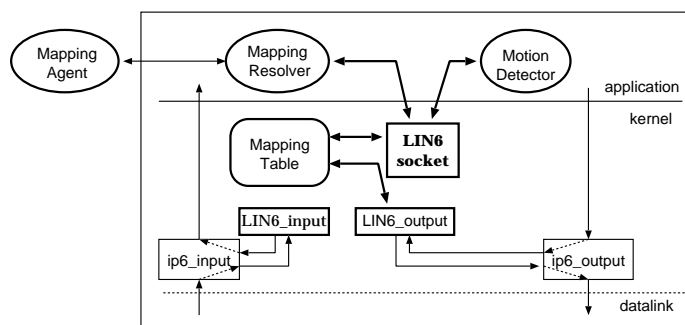


図 1: LIN6 の実装モデル図

Mapping Resolver は Mobile Node 上で Mapping Agent との仲介をする daemon である。Mapping Resolver は Mapping Table の更新を LIN6 socket を介して行なう。Motion Detector は移動を検出するための daemon であり、LIN6 socket を介して Mapping Resolver に移動を通知する。Mapping Resolver は新たなネットワークプレフィクスを取得した際、それを Mapping Agent へと登録する。

カーネル内の処理として、パケット受信の際は LIN6 アドレスから LIN6 汎用識別子への変換を行なう。パケット送信の際には、通信相手の現在のネットワークプレフィクスを Mapping Table の検索により LIN6 汎用識別子から LIN6 アドレスへの変換を行なう。Mapping Table にネットワークプレフィクスの情報が無い場合には Mapping Resolver に Mapping 情報の取得を依頼する。

3 ソフトウェアの公開

現在、実装したソフトウェアを

<http://www.csl.sony.co.jp/tera-project/>で公開中である。パスワード設定がなされているので、実験に協力して頂ける方は tera-project@csl.sony.co.jp までご連絡下さい。

表 1: 現在サポートされている OS list
FreeBSD 4.1 NetBSD 1.5_ALPHA2
NetBSD 1.4.2 BSDI 4.1