

5K-05 移動透過性とマルチホーミングを実現する ネットワークアーキテクチャLIN6のためのAPI設計*

松本存史[†] 藤川賢治[‡] 岡部寿男[†] 太田昌孝[‡] 寺岡文男[‡] 國司光宣[‡] 石山政浩[‡]
 京都大学[§] 東京工業大学 慶応義塾大学 東芝研究開発センター

1 はじめに

携帯端末の性能の向上と移動体通信機器並びにインターネットの普及により、出先や移動しながらのインターネットへのアクセス、いわゆるモバイルコンピューティングが活発に行なわれるようになってきた。移動中にも通信が途切れないシームレスなネットワークアクセス(移動透過性)を提供する様々なネットワークプロトコルが、IETF[1]を始めいくつかの団体から提案されているが、これらは三角ルーティングやパケットヘッダー長増大によるオーバーヘッドなどの重大な問題を抱えている。一方、LIN6プロトコル[2]はアドレスを位置情報部分とノード識別情報部分に分離して捉え、DNSとLocationAgent(後述)を利用することでそれらの多くの問題を解決するものである。本研究ではこのLIN6プロトコルのマルチホーミング対応のための拡張を行ない、それを用いてより頑健な通信を実現するAPIの設計を行った。

本稿でのマルチホーミングとはEnd-to-Endマルチホーミングと呼ぶもので、中継ルーターに依らず、複数の上流回線とアドレスを持つエンドノードと接続相手ノードの二者によって実現されるものである。

以下に拡張したLIN6プロトコルの概要と設計したAPIについて述べる。

2 LIN6プロトコルとマルチホーミングのための拡張

2.1 用語定義

Locator, node-id: LIN6ではIPアドレスを上位64bit(Locator)と下位64bit(node-id)に分けて考え、Locatorは位置情報を表し、node-idはノードを識別するために用いる。

LIN6 アドレス: Locator + node-id のこと。

MN(MobileNode): LIN6に対応し移動することを前提とする端末。

LA(LocationAgent): MNの現在の位置情報を持つMNと通信したいノードはMNが位置情報を登録しているLAに対してクエリーを行なう。

CN(CorrespondNode): MNに対する通信相手。

2.2 プロトコル概要

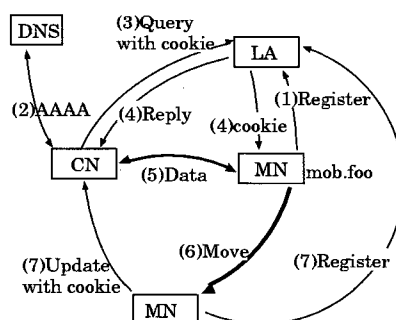


図 1: LIN6 プロトコルの通信概要

- 1 MNはLAに対して位置情報を登録する
- 2 CNはmob.fooというMNと通信する時にDNSにAAAAを問い合わせ、その結果MNのLocationAgent(LA)のアドレスを得る
- 3 LAに現在のMNの位置を尋ねる この時ランダムな文字列(cookie)を一緒に渡す
- 4 LAはMNにcookieを渡し、CNにMNの位置情報を返す
- 5 MNは同様の手段でCNのアドレスを取得し、CNとMNの間で直接通信が行なわれる
- 6 MNが移動し、アドレスが付け替わる
- 7 MNは[4]で受け取ったcookieを用いてCNに対して位置情報の更新を行う

2.3 マルチホーミングへの対応

マルチホーミング対応への拡張として位置情報(Locator)をやり取りする箇所において、複数のLocator

* API design for network architecture LIN6 that supports mobility and multi-homing

[†] Arifumi Matsumoto, Kenji Fujikawa, Yasuo Okabe

[‡] Masataka Ohta, Fumio Teraoka, Mitsunobu Kunishi, Masahiro Ishiyama

[§] Graduate school of Informatics, Kyoto Univ.

を扱えるようにパケットフォーマットを変更した。DNS から LA の Locator が複数個得られた場合、CN は応答が得られるまでそれらの Locator を順次試していく。また CN から LA への Query には CN の Locator を複数記述することができ、LA は Ack が返りパケットの受信が確認できるまでその Locator を順に試していく。MN から LA、MN から CN への Registration や Update パケットも同様である。

以上のような拡張を行なうことで、CN、MN、LA 全てのマルチホーミングに対応し、また Query などのデータ通信の前処理においても複数 Locator を利用することで、より信頼性の高い通信が可能になる。

3 API の設計と利用

ホスト名から DNS への問い合わせを通してアドレスを複数取得する API である *getaddrinfo()* の LIN6 への対応への変更を加えた。さらに新たにホスト名から DNS 及び LA への問い合わせを行ない、現在の Locator を全て取得する *getaddrinfo2()* という API を設計した。引数は *getaddrinfo()* とほぼ同じであるが、取得できる構造体に修正を加え、複数の Locator を扱うようにした。

次に下図のようなマルチホーミングしているネットワークへ MN が移動し、CN が MN と通信を開始する場合の様子を示す。

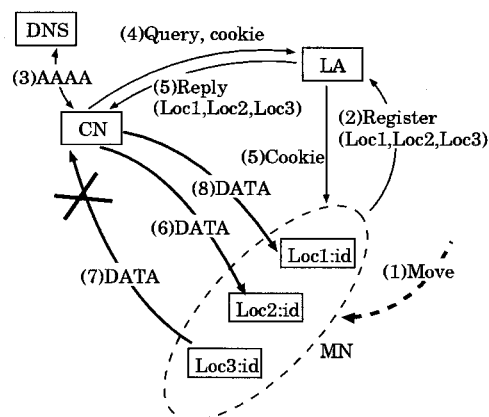


図 2: マルチホーミングしているノードとの通信

1,2 MN がマルチホームしている位置に移動し複数の Locator (Loc1, Loc2, Loc3) を取得し、これらを LA に登録する。

3-5 CN 上のアプリケーションで *getaddrinfo2()* が実行され、ホスト名からアドレスを取得する。

このとき内部的には DNS で LA のアドレスを取得し、その後 LA に対して MN の Locator を取得する。

6 アプリケーションはそのうちのどれかを選択しコネクションを開始する。

7-8 MN が CN に対して送信したパケットがなんらかの原因により CN に届かなかった場合、CN は *setsockopt()* という API を用いて接続先の Locator を変えて再度パケットを送信する。以下に簡易コードでそのプログラム例を示す。

```

struct addrinfo2 hints, *res, *res2;
/* ホスト名からアドレス取得 */
getaddrinfo2("mob.foo", "http", &hints, &res);
do { /* 接続が成功するまで Locator を試す */
    sock = socket(res->ai_family,
                  res->ai_socktype, res->ai_protocol);
    if ( sock < 0 ) continue;
    [res->ai_loc[i] と res->ai_id から addr 生成]
    if (connect(sock, addr, res->ai_addrlen)==0)
        continue;
    break;
} while ((res = res->ai_next) != NULL);
sig_handler(int sig) { /* シグナル処理 */
    [エラーシグナルを捕捉した場合]
    if ( res->ai_tloc[++] != NULL )
        /* 接続先変更 */
        setsockopt(sock, SO_DST, res->ai_tloc[i]);
}

```

4 おわりに

本研究ではマルチホーミングに対応するための LIN6 プロトコルへの拡張とそれを利用するための API 設計を行なった。今後は NetBSD への実装を行ない、評価を行なう予定である。

参考文献

- [1] IETF <http://www.ietf.org>
MobileIP: IP Mobility Support (RFC2002)
MobileIPv6: Mobility Support in IPv6 (Internet Draft)
- [2] "LINA: A New Approach to Mobility Support in Wide Area Networks", IEICE Trans. Communications, 2001