

異種ネットワーク環境における適応的通信デバイス制御

元濱 努[†] 瀧本 栄二^{††} 鈴木 和久[†] 毛利 公一^{†††} 大久保 英嗣^{†††}

[†]立命館大学大学院理工学研究科

^{††}株式会社 ATR 適応コミュニケーション研究所

^{†††}立命館大学情報理工学部

近年、通信技術の発達により、複数の通信デバイスを備えた小型で高性能の携帯端末が急速に普及している。また、駅、空港、店舗などにおけるホットスポットの普及によって、ユーザが場所を問わず端末をネットワークに接続可能な環境が整いつつある。このような環境では、端末が通信相手に対して、複数の通信デバイスを介して通信が可能になると考えられる。しかし、従来の通信方式では、複数の通信デバイスを扱うように設計されておらず、端末の通信資源を有効に利用することは困難である。本論文では、この問題を解決するために、通信状態やユーザのプリファレンスに応じて通信デバイスを使い分ける手法を提案する。

An Adaptive Control of Multiple Network Devices in Heterogeneous Networks

TSUTOMU MOTOHAMA[†] EIJI TAKIMOTO^{††} KAZUHISA SUZUKI[†]

KOICHI MOURI^{†††} EIJI OKUBO^{†††}

[†]Graduate School of Science and Engineering, Ritsumeikan University

^{††}Adaptive Communications Research Laboratories, ATR

^{†††}College of Information Science and Engineering, Ritsumeikan University

Recently, mobile terminals which have multiple network devices are widely used by the advanced communication technology, and users can access the network everywhere. For example, host-spots to which users can connect by his/her own terminal are available in stations, airports, stores, and so on. In such environments, it is possible for each mobile terminal to communicate with the same destination host through multiple network devices. However, since the conventional communication methods are not designed for handling to multiple network devices, the communication resources of each terminal cannot be effectively used. In this paper, we propose a mechanism for using the network devices properly according to the state of the communication and user's preference.

1 はじめに

近年、通信技術の発展により、様々な無線規格が普及している。無線 LAN の規格である 802.11a/b/g や、近距離の通信に特化した規格である Bluetooth など様々な場面で用いられている。また、これらの通信デバイスを複数備えた携帯端末や、駅、空港、店舗などにお

るホットスポットの普及によって、ユーザが場所を問わず端末をネットワークに接続できる環境が整いつつある。通信を開始しようとする端末は、ホットスポットにあるアクセスポイントを介した通信や、端末同士が動的にネットワークを構成するアドホックネットワークを利用して通信を行うことが考えられる。このように、固定的なネットワークと流動性のあるネット

ワークが混在する nomadic な環境が考えられる。また、ユーザの持つおのの端末は、様々な無線規格の異なる通信デバイスを備えていることが想定され、異種のネットワーク環境が混在する heterogeneous な環境が考えられる。

このような nomadic かつ heterogeneous な通信環境では、端末の位置や通信状態が動的に変化するため、ユーザやアプリケーションのプリファレンスを満たす通信デバイスを適応的に選択する必要がある。しかし、このような作業をユーザの操作に頼ってしまうと非常に使いにくいものになってしまう。また、従来の通信方式では、複数の通信デバイスを扱うように設計されていない。すなわち、複数の通信デバイスが利用可能な状態である場合に、ある通信デバイスを利用している間は他の通信デバイスを利用していないことが多い。

以上の背景から、我々は、ユーザやアプリケーションのプリファレンスと通信状態に適応して通信デバイスを制御する手法を開発している。本手法では、端末が通信相手に対して、利用可能な通信デバイスが複数ある場合、通信状態やプリファレンスから作成されるポリシーにしたがって出力する通信デバイスを選択する。また、通信データの種類によって出力する通信デバイスを使い分けることにより、端末に備えられている通信資源を有効に使い、おのの通信デバイスの特性を考慮した通信を可能とする。

以下、本論文では、2章で想定環境と本研究が想定するシナリオについて述べ、3章で本機構の設計について述べる。4章で本機構の実装について述べ、5章で関連研究について述べる。6章でまとめと今後の予定について述べる。

2 シナリオ

2.1 想定環境

本研究では、異種のネットワークが混在している環境において、そのネットワーク内をユーザが自由に移動する環境を想定している(図1参照)。ユーザは、複数の通信デバイスを備えている携帯端末を利用し、通信プロトコルとしてIPを用いて通信する。

図1において、Mobile Host(以下、MH)と Correspondent Host(以下、CH)が通信を行なう場合を考える。MHとCHは、無線LANデバイスとBluetoothデバイスを備え、Bluetoothの電波範囲内において無線LANのアクセスポイントが利用可能であるとす。本

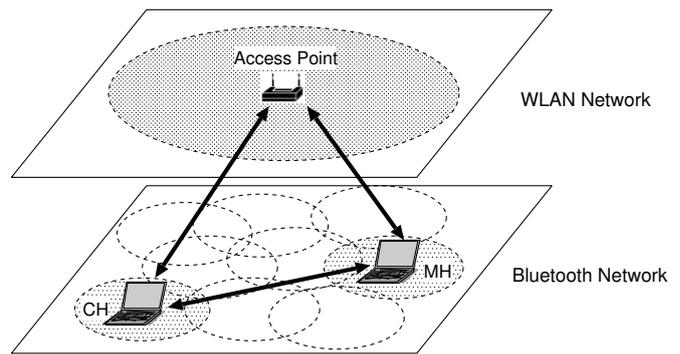


図1 想定環境

研究で提案する通信方式は、複数の通信デバイスが利用可能な場合、すなわち、複数の経路でMHとCHが通信可能な場合において、適応的に通信デバイスを制御するものである。

2.2 適応的通信デバイス制御

本研究は、前節における環境において、ユーザプリファレンスと通信状態に適応した通信デバイスを制御することを目的とする。以下に、シナリオ例を挙げる。

ユーザプリファレンス 帯域幅が狭くても電力消費量が少ない通信デバイスを使用したいというユーザプリファレンスがある場合、Bluetoothを用いて通信を行なう。

通信状態 RTT(Round Trip Time)、パケットロス率、帯域幅を考慮して、通信性能が良い通信デバイスを動的に選択して通信を行なう。

また、本研究では、通信デバイスを適応的に切り替えるだけでなく、複数の通信デバイスを協調させて通信することにより、資源を有効に利用することを考えている。協調の例として、パケットを分割し、並列に送信することにより、単位時間当たりの転送量を増加させることが考えられる。しかし、各通信デバイスのリンク状況によって、スループットの向上率が低い場合がある。また、通信状態を考慮したパケットスケジューリングや、分割されたパケットを統合する機構を実装する必要があるため、処理にかかるコストが高くなってしまふ。そこで、本研究では、アプリケーションや通信するデータの種類によって通信デバイスを使い分けることを考えている。本研究では、通信するデータを以下のように分類する。

イベントデータ 実時間制約、あるいは高速な通信が

求められるデータ (情報家電の制御コマンド, 制御の結果など).

バルクデータ 実時間制約をもたないデータ (静止画, HTML 文書など).

マルチメディアデータ 実時間制約は強く要求されるものの, その制約がソフトリアルタイムであるデータ (動画や音声などのストリームデータ).

これらのデータの種類, 通信状態, アプリケーションのプリファレンスによって, 通信デバイスを使い分ける. これにより, おのおの通信デバイスの利点を活かして適応的に通信を行うことが可能となる.

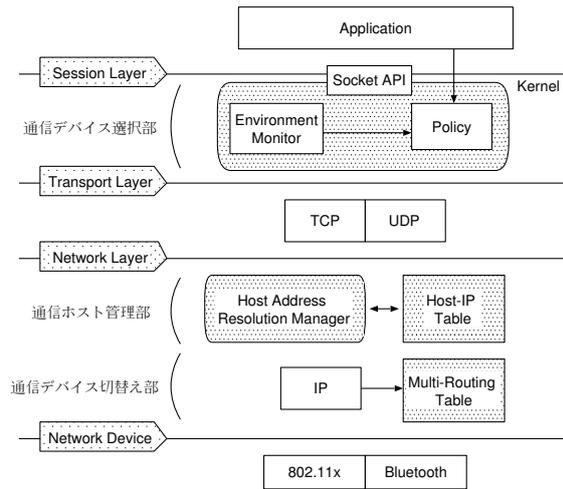


図2 適応的通信デバイス制御機構の全体構成

3 適応的通信デバイス制御機構の設計

本章では, まず提案手法を実現するために解決すべき課題について述べ, 本機構の設計方針について述べる. 次に, 本機構の全体構成と構成要素について述べる.

3.1 解決すべき課題

2章で挙げたシナリオを実現するためには, 次に挙げる課題を解決する必要がある.

通信デバイスの選択 通信相手と複数の経路で通信可能なとき, ユーザプリファレンスや通信状態に基づいて通信デバイスを選択する. このため, 送信先アドレスによって通信デバイスを決定する従来の通信機構では, 本機能の実現が困難である. したがって, 通信相手の通信デバイスとIPアドレスの構成から適応的に通信に用いる通信デバイスを選択する機能が必要となる.

ホストの同一性と接続の維持 IP アドレスは, 個々の通信デバイスに対して与えられる. しかし, 通信デバイスの切替えによってIPアドレスが変化することにより, ホストの同一性がとれないといった問題がある. また, TCP を用いて通信を行う場合, IP アドレスの変更によって接続を維持することができない. したがって, 通信相手の複数のIPアドレスが同一ホストであるということを保証する機構が必要となる.

3.2 設計方針

(1) 通信デバイスの切替えや協調動作は, アプリケーションから透過的に行う. これは, アプリケーション

に対するインタフェースは, 従来の Socket API と同様にすることにより, これらの処理を隠蔽することを意味する. したがって, TCP/IP による通信を行う既存のアプリケーションに変更を加えることなく適用可能となる.

(2) 通信を行う End-to-End 間でのみ本機構の実装を必要とする. これは, 通信を行うホスト間でのみ本機構を適用することを意味する. すなわち, 前節で述べた課題を解決するネットワーク機構を実現するために, 特別なインフラを必要としない. このため, ネットワーク全体の負荷を減らし, インフラの設置や維持などのコストを減らすことが可能となる.

3.3 全体構成

本機構の全体構成を図2に示す. 本機構は, 通信デバイス切替え部, 通信ホスト管理部, 通信デバイス選択部により構成される.

通信デバイス切替え部は, 通信デバイス選択部の Environment Monitor が監視・取得する通信状態や, ユーザプリファレンスを基に作成されたポリシーに従って通信デバイスを選択し通信する. 通信デバイスの切替えによるIPアドレスの変化は, 通信ホスト管理部によって上位レイヤに隠蔽される.

3.4 通信デバイス切替え部

通信デバイスの切替えは, ルーティングテーブルのエントリを書き換えることにより実現する. しかし, 従来のカーネルのルーティングテーブルでは, 提案手

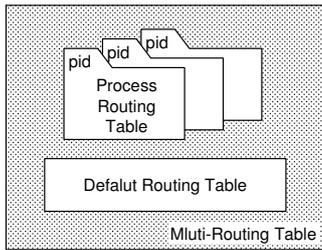


図 3 マルチルーティングテーブル

Routing Table (pid = 200)

daddr	gateway	dport	tproto	lface
10.1.1.10	*	all	tcp	wlan0
10.1.1.10	*	8002	*	wlan1

daddr : destination IP address tproto : transport protocol
 gateway : gateway IP address lface : output interface
 dport : destination port

図 4 プロセスルーティングテーブル

法である通信データの種類によって通信デバイスを使い分けるルールの記述が困難である。このため、本機構では、従来のカーネルのルーティングテーブルを拡張したマルチルーティングテーブルを用いる(図 3 参照)。

マルチルーティングテーブルは、従来のルーティングテーブルに加え、プロセスごとのプロセスルーティングテーブルを持つ。プロセスごとにルーティングテーブルを持つことによって、プロセスごとのルーティングエントリを明確に指定することができる。すなわち、1つのテーブルにすべてのエントリを記述する従来のルーティングテーブルと比較してテーブルの管理を容易にすることができる。

プロセスルーティングテーブルは、新しいポリシーとして送信先ポート番号(dport)、トランスポートプロトコル(tproto)を指定することができる(図 4 参照)。図 4 の 1 行目は、送信先 IP アドレスが 10.1.1.10 でトランスポートプロトコルが TCP の場合、通信デバイス wlan0 を用いるということを指示するポリシーである。また、2 行目は、送信先 IP アドレスが 10.1.1.10 でポート番号が 8002 の場合、通信デバイス wlan1 を用いるということを指示するポリシーである。これらのルールに当てはまらない場合は、従来のルーティングテーブルを参照する。このようなエントリを記述することにより、通信データの種類に応じて通信デバイスを切替え可能とする。

通信デバイス選択部は、プロセスがソケットを生成

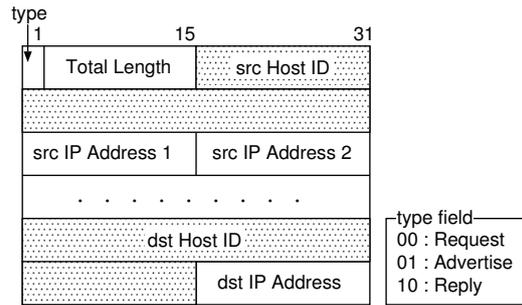


図 5 制御パケットのフォーマット

すると、プロセスルーティングテーブルを生成し、自身の持つポリシーにしたがってルーティングエントリを書き込む。ポリシーを持たないプロセスは、プロセスルーティングテーブルを生成せず、従来のルーティングテーブルを参照する。

3.5 通信ホスト管理部

通信ホスト管理部は、3.2 節で述べた課題であるホストの同一性やコネクションの維持を解決する。通信相手を持つ複数の IP アドレスを同一ホストのものであることを確認し、上位レイヤであるトランスポートプロトコルやアプリケーションに対して通信デバイスの切替えによる IP アドレスの変化を隠蔽する。本機構は、Host-IP Table と Host Address Resolution Manager (以下、HARM) により構成される。

3.5.1 Host-IP Table

Host-IP Table は、通信相手のホスト ID と IP アドレスの組を管理するテーブルである。ホスト ID は、通信デバイスの任意の MAC アドレスから決定される。本機構において、Host-IP Table の管理に用いる制御パケットの構造を図 5 に示す。制御パケットは、IP のオプション処理の際に、本機構によって処理される。また、参照されていないテーブルは、一定時間経過後に自動的に削除される。制御パケットには、Request, Advertise, Reply の 3 種類がある。

- Request パケット

通信先のホスト ID と IP アドレスを取得する際に送信するパケットである。Request パケットは、通信開始時に通信先となる端末に送信する。送信元 IP アドレスは、自身の端末が持つすべての IP アドレスを記述する。ただし、src IP Address 1 に

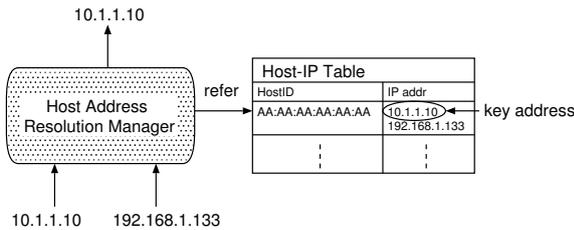


図 6 HARM 動作例

は、このパケットを送信する IP アドレスを記述する。これは、他のパケットタイプも同様である。送信先ホスト ID は、Request パケットを送信する時点では不明なため、すべて 0 に設定する。

- Advertise パケット
通信先にネットワークの移動などにより自身の IP アドレスが変更した際に通信先に送信するパケットである。パケットの内容は、Request パケットと同様である。ただし、送信先ホスト ID は、自明なため記述する。
- Reply パケット
Request パケットや Advertise パケットを送信した端末に対し、ホスト ID と自身の端末が持つすべての IP アドレスを記述し、返信するパケットである。

3.5.2 Host Address Resolution Manager

HARM は、Host-IP Table を参照し、通信相手を持つ複数の IP アドレスが同一ホストであることを確認するための機構である。HARM の動作例を図 6 に示す。

HARM は、上位レイヤのプロトコルに対して、同じ IP アドレスを持つホストからパケットが到着していると見せかける必要がある。このため、HARM において IP ヘッダの送信元アドレスを適切に書き換える必要がある。そこで、HARM は、IP からトランスポートプロトコルに渡されるすべてのパケットをフィルタリングする。フィルタリングしたパケットが、Host-IP table に記述のある IP アドレスと一致する場合、そのホスト ID が持つキーアドレスに IP ヘッダの送信元アドレスを書き換え、トランスポートプロトコルにパケットを渡す。通信相手を持つ IP アドレスのうちの 1 つにキーアドレスが指定される。キーアドレスとは、アプリケーションやトランスポートプロトコルが通信

表 1 ソケットオプション

type	説明
EVENT	RTT を重視
MLUTIMEDIA	帯域幅を重視
BULK	上記に該当しないデータ

相手を初めて認識した IP アドレスである。すなわち、通信相手に Request パケットを送った IP アドレスである。Host-IP Table に記述する IP アドレスがない場合、そのままトランスポートプロトコルにパケットを渡す。

3.6 通信デバイス選択部

通信デバイス選択部は、Environment Monitor (以下、EM) とポリシーテーブルによって構成される。EM は、ネットワークと通信デバイスの状態を観測し、以下に示す情報を取得する。

- ネットワーク
 - 帯域幅
 - RTT
 - パケットロス率
 - 経路情報
 - Host-IP table
 - ソケットの情報 (pid, port, protocol)
- 通信デバイス
 - IP アドレス
 - リンク情報
 - 無線電波強度

これらの EM の観測結果から、通信デバイスの選択のトリガとなるポリシーを作成する。通信デバイス選択部は、このポリシーにしたがってマルチルーティングテーブルのエントリを変更する。また、アプリケーションやユーザのプリファレンスを EM に通知することにより、ポリシーを設定することができる。プリファレンスは、端末の通信全体に適用されるが、Socket API である `setsockopt` 関数を利用して、個々のソケットにも同様に指定できる。指定できるソケットオプションを表 1 に示す。

4 実装

マルチルーティングテーブルを Linux カーネル 2.6.11 に実装した。従来のルーティングテーブルと同様に `ioctl` システムコールを用いて操作することが可能である。マルチルーティングテーブルへのエント

```

struct prntentry
{
    pid_t pid; /* Target Process ID */
    struct sockaddr prt_dst; /* Target address */
    struct sockaddr prt_gateway; /* Gateway addr */
    struct sockaddr prt_genmask; /* Target network mask */
    unsigned short prt_port; /* Target port */
    int protocol; /* Target Transport Protocol */
    char *prt_dev; /* Forcing the device */
};

// プログラム例
struct prntentry prt;
int skfd;

/* 構造体へ各要素の代入 */
skfd = socket(AF_INET, SOCK_DGRAM, 0);

if (/* エントリの追加 */)
    ioctl(skfd, SIOCADDPRT, &prt);
else /* エントリの削除 */
    ioctl(skfd, SIOCDELPRT, &prt);

```

図 7 プログラム例

り追加のプログラム例を図 7 に示す。マルチルーティングテーブルの内容は、Linux の proc ファイルシステムを利用し、“/proc/net/mroute”のファイルを読み込むことによって確認することができる。

HARM は、netfilter [1] を用いて実装する。netfilter とは、Linux カーネル 2.4 からカーネルに統合されたパケットフィルタリングソフトである。netfilter を用いて、IP レイヤを通過するパケットをフィルタリングすることにより実現する。

5 関連研究

Mobile IP [2] は、MH がネットワークセグメントを超えて移動した場合でも、端末のネットワーク上の識別子である IP アドレスを変更せず、同一 IP アドレスによる通信を実現する。Mobile IP で複数の通信デバイスを利用可能とするためには、外部エージェントやホームエージェントといったインフラが対応するか、他のホストによるブリッジを利用する必要がある。本手法では、特別なインフラを必要としない。しかし、本手法では、端末の移動により通信先の IP アドレスが変化し通信不可能になる場合がある。Mobile IP では、不変のアドレスであるホームアドレスを用いるので、移動後に気付けアドレスを再度登録することにより、通信を再開することが可能となる。したがって、Mobile IP は広域のネットワークに、本手法は、狭域のネットワークに適応するといった利用場面が異なることが考えられる。

Mobile SCTP (Stream Control Transmission Proto-

col) [3] は、multi-homing 可能なメッセージ指向の通信プロトコルである。サーバとクライアントとの通信路を多重化することで、いずれかの通信路が通信不能になった場合も、他の通信路を介して通信を継続することが可能である。しかし、Mobile SCTP は、TCP や UDP との互換性がない。したがって、SCTP に対応していない端末との通信を行うことができない。また、SCTP は、トランスポートレイヤのプロトコルであり、Socket API を変更する必要がある。したがって、既存のアプリケーションを使用するためには、ソースコードの修正と再コンパイルが必要となる。

6 おわりに

本論文では、異種ネットワーク環境における通信デバイスの制御手法について述べた。本手法では、ユーザやアプリケーションのプリファレンスと通信状態に応じて、適応的に通信デバイスを制御することが可能である。マルチルーティングテーブルを用いることにより、ポート番号やトランスポートプロトコルによって出力する通信デバイスを使い分けることが可能である。これにより、端末に備えられている通信資源を有効に使い、おのおの通信デバイスの特性を考慮した通信を行なうことが可能となる。

今後は、ホストの同一性を確認するために設計した HARM を他のレイヤで実現することを考えている。HARM では、TCP を用いた通信の通信デバイスの切り替えによって、コネクションを維持できない場合がある。これは、通信デバイスの切り替えに時間がかかった場合、再送タイムアウトによってコネクションが切断するためである。また、再送タイムアウトにならなくても再送タイマの待ち時間によっては、通信デバイスを切り替えてからパケットを受け取るのに時間がかかってしまう場合が考えられる。

参考文献

- [1] netfilter/iptables project, web page, <http://www.netfilter.org/>
- [2] Perkins, C.: IP mobility support, RFC 2002, Internet Request For Comments (1996).
- [3] Riegel, M and M. Tuexen,: Mobile SCTP, IETF draft-riegel-tuexen-mobile-sctp-03.txt, Aug. (2003).