

MIBsocket: 移動型計算機環境におけるネットワークエンティティの状態変化に対応する管理機構の設計と実装

湧川 隆次[†] 植原 啓介[‡] 田村 陽介[†] 徳田 英幸[‡]

慶應義塾大学 政策・メディア研究科[†] 慶應義塾大学 環境情報学部[‡]

本研究の目的は、ネットワークエンティティの動的な状態変化に対応する管理機構の実現である。MIBsocketはメッセージ交換を用いてネットワークエンティティにアクセスする。また、移動等により起きるネットワークエンティティの動的な状態変化を検知しネットワークソフトウェアに通知する。メッセージ交換を実現するためソケットを用いて実装した。本論文では、MIBsocketの適応例としてインタフェース切替機構を実装した。MIBsocketのネットワークエンティティにアクセスするパフォーマンスを測定した結果、ソフトウェアはネットワークエンティティの状態変化に動的に対応して処理することができた。また、インタフェース切替機構はネットワークエンティティの状態変化をトリガーにして、インタフェースを動的に切替えることができた。リンクの切断による通信切断をインタフェースを切替える事により回避する事もできた。

MIBsocket: Integrated Control Mechanism for Network Entity in Mobile Network

Ryuji Wakikawa[†] Keisuke Uehara[‡] Yosuke Tamura[†] Hideyuki Tokuda[‡]

Graduate School of Media and Governance, Keio University[†]

Faculty of Environmental Information, Keio University[‡]

The purpose of this research is to enable the mobile computer to integrate the control mechanism for Network Entity. The states of Network Entity are passed between kernel and network softwares by messages. The mechanism is implemented as the MIBsocket (a socket). MIBsocket detects the Network Entity changing and informs it to network softwares. MIBsocket pass the request and reply messages of the Network Entity states to network softwares. We measured, the overhead of MIBsocket, and applied the MIBsocket to switch network interfaces. As a result, the overhead time of the MIBsocket was reasonable, and we have successfully applied the MIBsocket so that the network software acknowledges the changes, and switches to the available interface.

1 はじめに

携帯型計算機の小型化、高性能化、様々な通信デバイスの登場、そして移動を支援する研究の充実により広く使われるようになった。利用者はネットワークソフトウェアや通信デバイスを用いて、World Wide Webや電子メールなどを利用する。しかし、通信中の計算機環境の変化によりネットワークが切断され、利用者が自ら回避しなくてはならないことがある。移動型計算機環境の充実により、計算機の移動を気にすることなく常にネットワークを利用できることが望まれる。計算機自身が移動を検知し、自動的に移動先の通信環境に最適な設定を行わせるシステムが必要となる。

しかし問題は、移動型計算機環境の最大の特徴であるネットワークエンティティの状態変化が頻繁に起きることにある。ネットワークエンティティとは、ネットワークに関する機能(プロセス)をモデル化したもので計算機内に1つ以上存在する。計算機内でネットワークエンティティ同士がIPアドレスと経路情報のように相互関係を持つため、それらが連鎖的に変わることがある。計算機が移動しながらの通信を行うためには、ネットワークエンティティの変

化により起こる通信の切断や通信品質の低下(高遅延、低帯域)を回避しなくてはならない。これら動的な回避を行うためには、計算機がネットワークエンティティを統一的に管理することが重要である。現在では、Multihomed Host [1]やMANET [3]のように、ネットワークエンティティの情報取得やネットワークエンティティの状態変化の検知が可能なことを前提とした技術も多い。しかし、現在の計算機では有線ネットワークを想定したシステムであるため、ネットワークエンティティの状態変化に十分対応できていない。

本研究では、移動型計算機環境におけるネットワークエンティティの状態変化に対応する管理機構としてMIBsocketを提案する。MIBsocketは、ネットワークエンティティの状態をManagement Information Base (MIB) [6]のデータ構造に基づいて管理する。MIBに準拠することで汎用性が生まれる。

本稿では第2章で、移動型計算機における問題点について述べる。第3章で、関連研究を挙げその問題点やMIBsocketとの違いを示す。第4章で、我々の提案するMIBsocketの概要について説明し、第5章で設計と実装について詳解する。第6章では、イン

タフェース切替え機構を用いて MIBsocket の適応例を示す。第7章は、インタフェース切替え機構等を用いた MIBsocket の評価を行う。最後にまとめと今後の課題を述べる。

2 移動型計算機における問題点

計算機システムでは、カーネル及びネットワークソフトウェアが計算機内の多くの資源を共有している。例えば、IP アドレスはカーネル内で実際に保持される一方で、Mobile IP[8]、FTP、Telnet などのネットワークソフトウェアも利用している。このように、IP アドレスはカーネルとネットワークソフトウェアで共有されている。固定された計算機ではネットワークエンティティの情報は、システムの起動時に決定され不変であるため共有していても問題はない。しかし、移動型計算機ではネットワークエンティティの状態を移動先のネットワークへの適応などの理由から、DHCP[4]のようなネットワークソフトウェアやカーネルが動的に変更するため問題が生まれる。なぜなら、ネットワークエンティティに複数で変更を行うと競合しシステム内でネットワークエンティティの状態に整合性が無くなる。例えば、複数のネットワークインタフェースを持った移動型計算機が DHCP クライアントと PPP[7] を同時に起動している場合を考える。経路情報のデフォルト経路はそれぞれに共有されており、DHCP と PPP が同時に設定すると、デフォルト経路の競合が起きソフトウェア間で整合性が無くなる。設定したデフォルト経路がシステム内で有効であると DHCP は処理を進めるが、実際は PPP が後から設定したために値が変わってしまっているという事が起こりうる。

また、既存の固定された計算機では、ネットワークエンティティの状態は利用者あるいはネットワークソフトウェアからは基本的に参照することができないものが多い。これらのネットワークエンティティは、システム内で一意に不変で既知な情報でありカーネル内で保持されていることが多い。例えば、ネットワークインタフェースのリンクの状態がそれである。有線ネットワークではリンクの状態は不変な情報であるため、ネットワークソフトウェアは参照する必要が無い。しかし、移動型計算機環境では移動することによりリンクの状態が変化する。現在の計算機システムでは、ネットワークソフトウェアがネットワークエンティティを直接参照する手段が少なく、ネットワークエンティティの状態変化を動的に通知するメカニズムも限られている。ネットワークソフトウェアは移動によるネットワークエンティティの動的な変化に対応することが難しい。

3 関連研究

これらの問題を解決するため、今日までさまざまな研究が行われている。本章では、移動型計算機環境におけるネットワークエンティティ管理の研究について紹介する。

Oregon Graduate Institute で行われている Quasar プロジェクト [5] では、利用インタフェースをネット

ワーク環境に応じて切替え、インタフェースの帯域に応じたデータ量を制御する研究を行っている。ネットワークエンティティの状態変化を ICMP のエコーリクエストを用いて検知するためコストが掛かり、無線などの低帯域のリンクでは問題がある。Stanford 大学の MosquitoNet プロジェクト [2] では移動体通信環境における OS とアプリケーションの問題について研究している。特に Mobile IP を用いた移動透過性の研究を中心に行っている。これらを実現するために、アプリケーションとネットワークエンティティの間で透過的なシステムを作ることを目標としている。しかし、ネットワークエンティティの管理機構に汎用性が欠ける。CMU 大学の Odyssey プロジェクト [9] では、アプリケーションとオペレーティングシステム間で協力して動作するシステムを研究している。Odyssey は、MIBsocket とアプローチが似ているが、ソフトウェアに計算機の移動を知らせるのが目的であり、ネットワークエンティティの管理ではない。

4 MIBsocket の概要

MIBsocket の目的はネットワークエンティティの状態管理である。また、移動型計算機環境でのネットワークエンティティの状態変化にも対応しなくてはならない。

4.1 機能

MIBsocket ではネットワークエンティティの状態管理を実現するため以下の5つの機能を提供する。

- ネットワークエンティティの状態取得機能
- ネットワークエンティティの設定機能
- ネットワークエンティティの状態変化通知機能
- ネットワークエンティティへのアクセス制御機能
- 必要なネットワークエンティティの情報フィルタリング機能

4.2 特徴

ネットワークソフトウェアは、MIBsocket を用いることによりネットワークエンティティの状態を管理でき、状態変化に対応した動作が可能になる。例えば、DHCP はリンクが有効になると IP アドレスの取得要求を送信し、リンクが無効になるとリリース要求で IP アドレスを開放できる。

また、MIBsocket を用いることにより複数のネットワークソフトウェアを統合して動作させることが可能になる。例えば、インタフェース切替え機構が、DHCP のアドレス取得のイベントをトリガーにしてインタフェースの選択、切替えの動作をさせることが可能になる。MIBsocket をネットワークエンティティとネットワークソフトウェアのインタフェースとすることにより、ネットワークソフトウェアは互いにネットワークエンティティへのアクセス状況を知ることができるからである。

5 MIBsocket の設計と実装

MIBsocket は複数のネットワークソフトウェアで利用でき、ネットワークエンティティにアクセスで

きなくてはならない。以下に MIBsocket の設計と実装について説明する。

5.1 設計

MIBsocket は、ネットワークエンティティの状態を取得、設定、状態変化の通知が行える必要がある。アプローチとして Shared memory, ポーリング, メッセージ交換の 3 つの方法が考えられる。MIBsocket では、メッセージ交換を採用する。メッセージ交換は、ネットワークエンティティの設定を要求メッセージの送信で行うため到着順に処理され、競合が起きにくいなどの特徴がある。

図 1 は、MIBsocket の概念である。ネットワークソフトウェアはネットワークエンティティの操作を、MIBsocket への要求メッセージを用いて行う。MIBsocket は要求に対する回答として応答メッセージをネットワークソフトウェアに返す。もし要求に対する処理が失敗した場合はエラー番号が返る。また、ネットワークエンティティの状態変化の通知はトラップメッセージとして MIBsocket からネットワークソフトウェアに通知される。これらの処理を 1 つの MIBsocket で行うため、3 つのモード ACTIVE, PROACTIVE, REACTIVE を定義した。それぞれ設定、取得、トラップを意味する。その他にも、MIBsocket にフィルタリングを設定することにより、ネットワークソフトウェアは必要な情報のみを受け取ることができる。また許可も同様で、ネットワークソフトウェアのネットワークエンティティへのアクセス制御を行うことができる。複数の MIBsocket が同時にネットワークエンティティの設定を行った時に起こる競合を、許可の機能を用いて回避することができる。

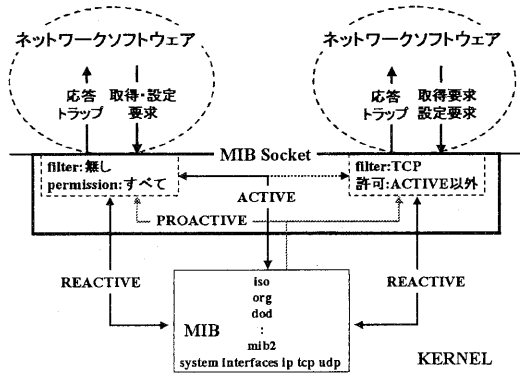


図 1: MIBsocket の全体像

また、MIBsocket で扱うネットワークエンティティの情報は、MIB のデータ構造に準拠している。ネットワークエンティティの情報はカーネル内で保持されていることが多い。例えば、*ifnet* 構造体にはインタフェースに関する情報が含まれている。MIBsocket ではこれらの情報と MIB のオブジェクト識別子とを対応づけて利用する。

5.2 実装

MIBsocket では、UNIX 上でメッセージ交換を実現するためにソケットを用いる。ソケットを用いて、メッセージの送受信を行いネットワークエンティティの管理を行う。MIBsocket は、新しく追加した PF_MIBSOCKET ドメインの Raw ソケットを用いる。

MIBsocket で扱うメッセージには、オブジェクト識別子、識別子長、モード、qflag、データのエントリがある。qflag は、メッセージが複数に跨る場合用いられる。

図 2 が MIBsocket のフローチャートである。MIBsocket 内部でモード毎に処理されているのが分かる。ネットワークソフトウェアからのメッセージは、MIBsocket が許可について調べる。ACTIVE モードのメッセージを受け取ると MIBsocket は設定のためのパラメータを調べ設定のための関数を呼ぶ。設定が成功するとトラップメッセージが送信される。REACTIVE モードのメッセージの場合、要求されたネットワークエンティティのオブジェクト識別子の実体を調べ応答する。PROACTIVE モードの場合は、カーネル内でネットワークエンティティの状態変化を検知し、トラップメッセージを作成し送信する。MIBsocket は、状態変化検知のためネットワークエンティティ毎にデバイスドライバやルーティングソケット [10] を用いる。デバイスドライバはインタフェースのリンクの状態を通信デバイスからの割り込みやリンクのポーリングによって検知する。ルーティングソケットは、IP アドレスや経路の状態変化を検知する。MIBsocket にはトラップを送信するための関数 (*mib_trap()*) が用意されているため、自由にシステム内で任意のトラップの送信が可能である。

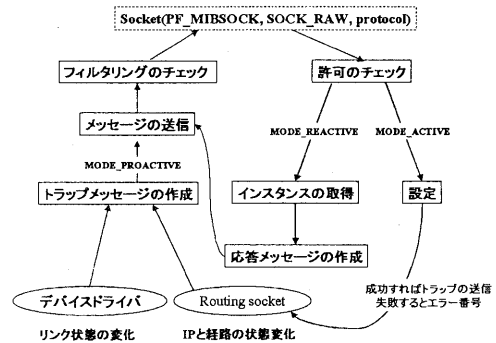


図 2: MIBsocket のフローチャート

MIBsocket のフィルタリングや許可の設定は *ioctl* を用いる。 *ioctl* によって設定された情報はソケットのコントロールブロック (*rawcb* 構造体) に保持される。 *ioctl* のコマンド *SIOCMIBFILTER* は引数のネットワークエンティティに対するメッセージをフィルタリングの設定を行う。 *ioctl* のコマンド *SIOCMIBMODE* は、引数の MIBsocket のモードに対して操作の許可を与えることができる。

6 MIBsocket の適応例

MIBsocket による管理機構の構築の例としてインタフェース切替え機構を設計、実装した。インタフェース切替え機構は、環境の変化により起こる通信切断に対応しネットワークインタフェースを切替えて通信を継続する。

6.1 インタフェース切替え機構の設計

図3がインタフェース切替えの全体像である。MIBsocket からインタフェースの状態及び状態変化の情報を取得し携帯型計算機に最適なインタフェースを、独自のポリシーにより選択する。インタフェースの切替えは MIBsocket を通じて設定、処理する。インタフェース切替え機構の IP アドレス取得方法として、DHCP を用いた動的な取得と設定ファイルからの静的な取得がある。ネットワークエンティティの状態の競合を防ぐため、DHCP の MIBsocket には設定の許可を与えないため、IP アドレスの設定は計算機に反映されない。しかし、インタフェース切替え機構が DHCP の MIBsocket への設定要求メッセージを受け取り、IP アドレスの取得を知ることができ、必要なら設定を行う。

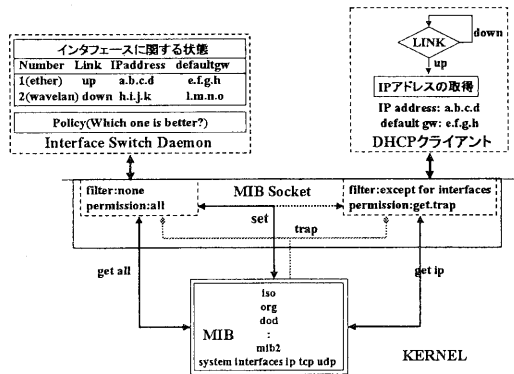


図 3: インタフェース切替え機構

6.2 インタフェース切替え機構の実装

図4にインタフェース切替え機構のフローチャートを示す。インタフェース切替え機構は主に以下の3つの機能に分けることができる

- インタフェースの情報蓄積
- インタフェースの選択
- インタフェースの設定

インタフェースに関する情報蓄積は、起動時に設定ファイルから初期化し、トラップなどの情報を元に随時更新されていく。

インタフェース選択機能は、インタフェースの情報蓄積を元に、どのインタフェースを選択するのが適切かを判断し決定する。判断基準としては、リンクの状態及び IP アドレスの有無などのインタフェースの有効性を用いる。また、インタフェースのリンクの変化、IP アドレスの有無、経路情報の変更(主にデフォルト

経路)などのネットワークエンティティの状態変化が起きることにより、再選択を行う。ネットワークエンティティの状態変化は、MIBsocket の PROACTIVE モードを用いて監視する。インタフェースの決定の後、インタフェースの設定機能により通信が行えるように設定を完了する。インタフェースの切替えは、IP アドレス及び経路の設定を更新する事により実現する。インタフェースの設定は、MIBsocket の ACTIVE モードを用いる。設定が成功すると PROACTIVE モードのメッセージが MIBsocket から送られるので、インタフェース切替え機構は設定が成功しインタフェースが切替えられたかどうかを知る事ができる。

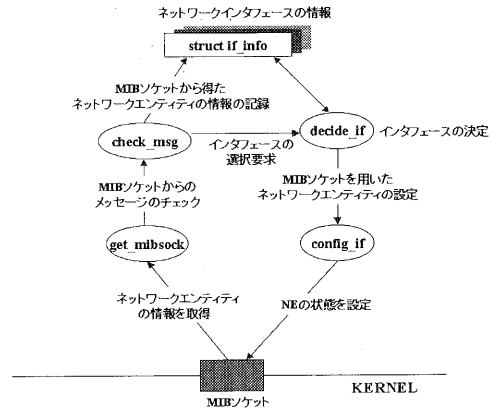


図 4: インタフェース切替え機構フローチャート

7 評価

MIBsocket の処理に関する評価と MIBsocket の適応例として実装したインタフェース切替え機構を用いて評価を行う。

7.1 MIBsocket の性能評価

測定環境は、OS が FreeBSD-2.2.7 + PAO-0913 である。CPU は pentium MMX 200MHz でメインメモリは 96MB である。

MIBsocket を用いてカーネル内の情報を取得する (REACTIVE モード) のにかかる時間を測定した。測定は、オブジェクト識別子から実体を取得する時間、応答メッセージの作成、及び MIBsocket に送信するにかかる時間である。送信にかかる時間にはフィルタリングにかかる時間も含まれる。測定では、インタフェースのインデックス (ifindex) 及び計算機内の IP アドレス (ipAdEntAddr) を取得した。表 1 が測定結果である。MIBsocket では、オブジェクト識別子からオブジェクトの実体へは線形探索で行っているため、オーバーヘッドがある。しかし、どのオブジェクトに対しても取得にかかる時間は約 100usec である。またメッセージの作成にかかる時間は 1 メッセージ毎に約 150usec である。結局、送信が完了するまでのトータルな時間は約 400usec (オブジェクトの

取得は始めだけ行われるので以後のメッセージは約300usec) がある事が分かる。このように、MIBsocketの取得機能のオーバヘッドは無視できる範囲内に処理できる。ネットワークネットワークソフトウェアはMIBsocketを介してリアルタイムなネットワークエンティティの状態を取得できる。

表 1: REACTIVE モードにおける MIBsocket のパフォーマンス

Operation	Times (μ sec)	
	ifIndex	ipAdEntAddr
オブジェクトの取得	104	109
メッセージの作成	733	296
メッセージの送信	778	346
計	1615	751

表2がMIBsocketのトラップ(PROACTIVEモード)の性能評価である。今回、ネットワークインタフェースのリンクの状態変化のトラップについて評価する。リンクの状態を検知できるイーサネットの場合、デバイスからの割り込みがかかるため約10 usecで検知できる。無線LANでは、無線の電波強度をデバイスドライバでポーリングしているため性能が落ちる。状態検知からMIBsocketにトラップが送信されネットワークソフトウェアが受け取るまでの時間は約500 usecである。これはネットワークソフトウェアが、ネットワークエンティティの状態変化にシステムが影響を受ける事なく対応できる時間である。MIBsocketのトラップ機能は、ネットワークエンティティの状態変化を検知しネットワークソフトウェアに通知することができた。また、トラップ機能の性能はデバイスドライバやルーティングソケットなどの状態変化の検知にかかる時間に依存している事が分かる。

表 2: MIBsocket のトラップ機能の性能

MIBsocket の Operation	Time (usec)
リンクの変化の検知	ドライバに依存 (sn: 9)
mbufの取得	96
メッセージの作成	137
メッセージの送信	182

7.2 インターフェース切替えを用いた評価

TCP と ICMP を用いた通信中にインタフェースを切替えて評価した。インタフェースは、リンクの状態変化を受けて動的に切替える。

7.2.1 評価環境

図5が、インタフェース切替えの測定におけるネットワークのトポロジである。AT&TのWaveLANカードとMegahertzのX-Jackカードを用いて行う。MIBsocket及びインタフェース切替えが動くMobile Hostは、OSがFreeBSD-2.2.7のラップトップである。pentium MMX 200MHzでメインメモリは96MBである。TCPを用いた場合のみ、Mobile IPを用いた。Home Agentはルータの機能もする。Home AgentはpentiumII 450MHzでメモリが128MBである。Correspondent Hostは、pentium pro 200MHzでメインメモリが192MBである。

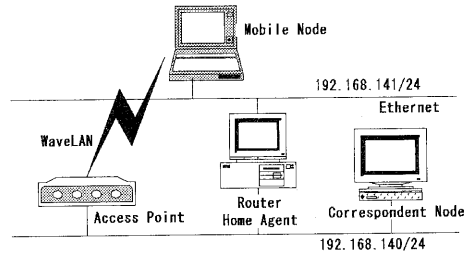


図 5: 実験のネットワークトポロジ

7.2.2 TCP を用いた評価

図6は、TCPのセッションを張りながらインタフェースを切替えた場合の packetsの到着時間とシーケンス番号である。測定は評価プログラムであるtcpを用いてデータを送り、tcpdumpでログを取った。受信側のホストでtcpdumpを行い、送信ホストでインタフェース切替えを行った。インタフェース切替えは2度起き、イーサネットから無線LANへと無線LANからイーサネットへの切替えである。本実験では、TCPのセッションが、アドレスの変更により切断されてしまうため、Mobile IPを用いた。なぜなら、TCPは通信に相手計算機のIPアドレスを情報として用いるため、インタフェースを切替えることでTCPのセッションは切断されてしまうからである。Mobile IPを用いることにより、計算機は不変であるHome addressを用いてセッションを張ることができ、切断を防ぐことができる。

TCPには輻輳制御機能を持っているため、インタフェースを切替えることにより起きるパケットの破棄を検知しスロウスタートが起動され、再送を試みる。イーサネットから無線LANへの切替えの時により多くのコストがかかっているのは、帯域の10Mbpsから2Mbpsへの急激な減少により、TCPが輻輳と判断したためである。しかし、TCPのセッションがタイムアウトする前に、リンクの状態変化を検知しインタフェースを再選択し切替えることができた。また、インタフェース切替え機構がネットワークエンティティの状態変化に動的に対応できているためTCPのセッションへの影響を与えていない。よって、MIBsocketのトラップ機能が有効である事が分かる。

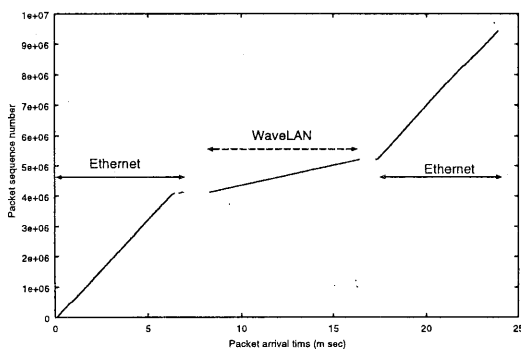


図 6: TCP を用いた性能評価

7.2.3 ICMP を用いた評価

図7は、ICMP を用いた場合のエコー要求・応答パケットの到着時間とシーケンス番号である。MIBsocket がリンクの状態を検知しインタフェースを実際に切替えた時間も示す。測定は、インタフェース切替えを行うホスト上で pentium カウンタ¹を用いて行った。ICMP を用いたのは TCP の測定での Mobile IP と TCP の処理自体によるオーバーヘッドが影響し、インタフェース切替えの正確なパフォーマンスが把握できないためである。ICMP では、プロトコル処理にかかるオーバーヘッドが少ないため採用した。

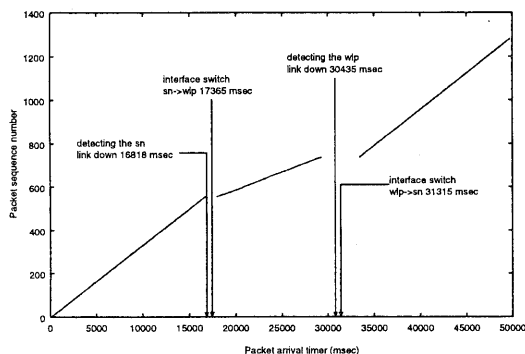


図 7: ICMP を用いた性能評価

リンクの状態変化検知から実際にインタフェースの選択及び切替え設定が完了するまでに約 1000msec かかっている。つまり、インタフェース切替えの性能は状態変化の検知にかかるコストに依存していることが分かる。イーサネットから無線 LAN への切替えのコストがその逆に比べて速いのは、イーサネットのリンクの状態変化の検知にコストがかからないからである。

¹pentium カウンタは CPU の中の内部レジスタである。1 クロック毎にカウンタが上がっていく。

8 終りに

MIBsocket を用いることにより、動的なネットワークエンティティの変化に対応する機構が実現できた。測定評価の結果、ネットワークエンティティの状態の取得あるいはトラップにかかる MIBsocket のオーバーヘッドも最小にすることができた。また、ネットワークエンティティの状態を介してそれぞれのアプリケーションが協調動作することも確認できた。インタフェース切替え機構は DHCP の IP アドレス取得をトリガーにインタフェースの選択を行えた。

今後の課題として、移動体計算機の様々な問題点に対応していくことがある。電力管理などもそれである。電力が無ければ、システムは動かないので電力の有効利用や移動体計算機環境では重要である。本システムを応用して移動体計算機環境の支援を行っていく。

謝辞

本研究において、Mobile IP を始め御指導を頂いた株式会社東芝の石山政浩氏に心から感謝します。また、本研究を進める上で様々な議論や助言を頂いた R3 プロジェクトと Rover プロジェクトの皆様へ感謝の意を表します。

参考文献

- [1] 砂原秀樹, 比良木貴志, 植原啓介, 尾家祐二: “移動体端末装置における通信インターフェースの自動選択機能の実現,” DiCoMo, 1997.
- [2] M. Baker, X. Zhao, S. Cheshire and J. Stone.: “Supporting Mobility in MosquitoNet” In *Proceedings of the 1996 USENIX Technical Conference* Jan. 1996.
- [3] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, J. Jetcheva.: “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols” In *Proceeding of MOBICOM '98* Oct. 1998.
- [4] R. Droms.: “Dynamic Host Configuration Protocol,” Request for Comments: 2131 March 1997.
- [5] J. Inouye, S. Cen, C. Pu and J. Walpole.: “System Support for Mobile Multimedia Applications,” In *Proceedings of the 7th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '97)* June 1997.
- [6] K. McCloghrie, and M. Rose, : “Management Information Base for Network Management of TCP/IP-based Internets: MIB-II,” Request for Comments: 1213 Mar 1991.
- [7] G. McGregor: “PPP Internet Protocol Control Protocol(IPCP),” Request for Comments: 1332 May 1992
- [8] C.Perkins.: “IP Mobility Support,” Request for Comments: 2002 Oct. 1996.
- [9] M. Satyanarayanan, B. Noble, P. Kumar, M. Price.: “Application-Aware Adaptation for Mobile Computing,” In *Proceedings of the 6th ACM SIGOPS European Workshop, Sep. 1994*
- [10] W. R. Stevens, W. Richard.: “TCP/IP Illustrated, Volume 2”, Addison Wesley; 10/94; ISBN: 020163354X