第13部

移動体通信環境

第 1 章　はじめに

近年の計算機技術の進歩と、ネットワークの普及により、可搬型計算機やスマートフォン、PDA などの利用が盛んになっている。一方、移動型の計算機からのインターネットの利用をみると、接続環境やネットワーク上の移動への適応など、解決すべき課題は多い。

Rover BOF は、移動計算機環境に関する様々な研究と議論を行うグループである。我々は通信プロトコル、ミドルウェア、アプリケーションなど、移動計算機用のネットワーク利用について、上位層から下位層まで広い分野に渡って研究を進めている。

本年度は、これまで Rover BOF で議論してきた研究項目のうち、2 つの項目について報告する。一つは、今後必要となるであろう IPv6 での移動計算機環境に着目し、我々が新たに提案する「縮退モデル」に基づく IPv6 用の移動体通信プロトコルについて報告する。もう一点は、これまであまり明らかにされていなかった可搬型計算機のネットワーク利用の実体についての報告である。Rover では、メンバーが使用している可搬型計算機のネットワークの利用調査をおこなってきた。本報告では、可搬型計算機のネットワーク利用に関する解析結果について述べる。

第 2 章　縮退アドレスモデルに基づく IPv6 上の移動透過プロトコル

2.1 はじめに

携帯端末の性能の向上と、移動通信機器の普及に伴い、移動先から携帯端末を利用した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われるようになってきた。また、インターネットへの接続点および接続方法のさらなる増加が見込まれるため、移動透過性を保証する通信プロトコルへの期待が高まっている。Internet Engineering Task Force (IETF) では、次世代インターネットプロトコルである IPv6[125] において、移動ノードの透過的なインターネットアクセスを提供するためのプロトコルである Mobile IPv6[126][127] の標準化を行っている。しかし、[128] で指摘したように、現在の Mobile IPv6 には普及を阻害する根本的な問題があると我々は考える。

本報告では、Mobile IPv6 が抱えている問題を持たない、新しい移動透過性保証のプロトコルを提案する。

まず 2.2 章で Mobile IPv6 プロトコルの概要を示す。2.3 章では、現在のインターネットの利用形態を踏まえて、Mobile IPv6 自身のアーキテクチャの問題点について議論する。そして、2.4 章にて我々の提案方式について述べる。提案方式は、本来ネットワーク層で必要と考えられる位置識別子とノード識別子を、IPv6 アドレス内に縮退させることにより、効率的な移動透過性保証プロトコルを提供する。最後に、2.5 章にて提案方式の考察を行う。

2.2 Mobile IPv6 のプロトコル概要

本章では、まず Mobile IPv6 が提供する移動透過性ついて述べ、続いてプロトコルの概略を示す。

2.2.1 移動透過性

Mobile IPv6 は、ネットワークの物理的な接続点が変化しても同じ IP アドレスを使用し続けることにより、IP 層による移動透過性 (mobility)[129] を提供することを目的としている。

移動透過性は、単にノードがあるネットワークから他のネットワークへと移動できる可搬性 (nomadicity)[129] と呼ばれる能力に比べ、以下の 2 点で優れている。

- 移動時のセッションの保持
  移動透過性を有するノードは、ノードの移動に関わらず既存のセッション、サービスをそのまま継続して使用することができる。可搬性のみを持つ移動ノードがネットワーク間の移動すると、IP アドレスが変化するため、セッションの継続が不可能になる。

- 移動によって変化しないノード識別子
  移動透過性を有するノードは、ネットワークの位置に依存しない不変の識別子を持つ。このため、移動ノードを IP 層で識別可能である。可搬性のみのノードがネットワーク上を移動

する場合、Internet Protocol ではノードの識別子が IP アドレスであるため、移動に従い識別子が変化し、識別はできない。

2.2.2 Mobile IPv6 の概念

現在の IP アーキテクチャでは、1 つの IP アドレスがネットワーク上の位置を表す情報と、ノードを識別する情報という 2 重の意味を持つ。ゆえに、現状の IP アーキテクチャは移動透過性を提供できない。Mobile IPv6 は、この問題を解決するために提案された。

Mobile IPv6 では、ノードは最低 2 つのアドレスを使用する。ノード固有に割当てられた Home Address (Haddr) と、現在接続されているネットワーク上で取得した Care-of-Address (CoA) である。移動ノードと通信するノードは、Haddr を用いて通信する。CoA は移動の度に変化するが、Haddr は変化しないため、移動ノードがネットワーク上の接続点を変化させても通信を継続することができる。

Haddr は、単なる IPv6 アドレスであるので、このアドレスをパケットの配送に使用できる。Mobile IPv6 はこの事実を最大限に活用する。まず、Haddr をノードの識別子ではなく、「移動ノードが本来接続しているべき位置のアドレス」と定義し、Haddr までの経路を利用する。すなわち、Haddr 宛のパケットは、移動ノードが現在接続されている位置ではなく、この Haddr が所属するネットワークへと配送される。Haddr が所属するネットワークには、Home Agent (HA) と呼ばれるルータを配置する。移動ノードは、Haddr と、現在の CoA との対応 (これを binding と呼ぶ) を定期的に HA に通知する。HA は、移動してしまったノードの Haddr 宛のパケットを代理受信し、CoA へと転送する。

また、移動ノードは、現在の通信相手に binding を直接通知してもよい。binding を持つ通信相手は、パケットの宛先は Haddr のまま、IPv6 の始点による経路制御の機能を利用して、HA を介することなくパケットを移動ノードへと最適な経路で配送できる。

2.2.3 Mobile IPv6 の動作概要

本節では Mobile IPv6 の基本的な動作を示す。詳細については [127] を参照されたい。

Mobile IP の概略を図 2.1 に示す。

- 移動ノード (MN) は 1 つの Home Address (Haddr) を持つ。これは MN が、任意のネットワークの接続点において、永続的に使用できる IPv6 アドレスである。
- Care-of-Address (CoA) は、MN が移動先ネットワークで割り当てられるアドレスである。
- Haddr が属する IP サブネット (Home Network) には Home Agent (HA) と呼ばれるルータがある。これは移動ノードから現在位置の登録を受け取って管理し、移動先のノードにパケットをカプセル化転送する役割を果たす。

MN は、ネットワーク上を移動すると、まず移動先のネットワークにおいて CoA を取得する。CoA を取得した後、HA に対して現在の CoA を通知する。

MN は通信相手 (CN) に対してパケットを送信する場合、パケットの source address には CoA を使用する。同時に、送信パケットに対して必ず Home Address Option (HAO) を付与する。Home Address Option 内には、MN の Haddr が格納されている。このパケットの受信者は、パケットの送信者を CoA として認識するのではなく、HAO 内に埋め込まれている Haddr であると認識する。HAO は、IPv6 の拡張ヘッダである Destination Option のうちの 1 つとして定義されている。

MN は、自分の現在位置を通知するために CN に対して Binding を通知してもよい。通知されたノードは、現在の Binding を一時的に記憶する Binding Cache を生成してもよい。

CN が MN にパケットを送信する場合、パケットの destination address を MN の Haddr としてパケットを送信する。もし Binding Cache を持っていれば、パケットに対して Routing Header (RH)[125] を付与し、CoA 経由で MN へと到達するようにする。RH は、パケットの送信者がパケットの経由点を明示的に指示できる IPv6 の拡張ヘッダである。

RH がパケットに付与されていない場合、パケットは Haddr の属するサブネットワークへと到達する。HA は、このパケットを代理受信し、CoA に向けて転送する。転送には IPv6 in IPv6 のカプセル化による tunneling を使用する。この場合は MN, CN, HA 間での冗長な三角経路が発生する。

一方、RH がパケットに付与されている場合、パ

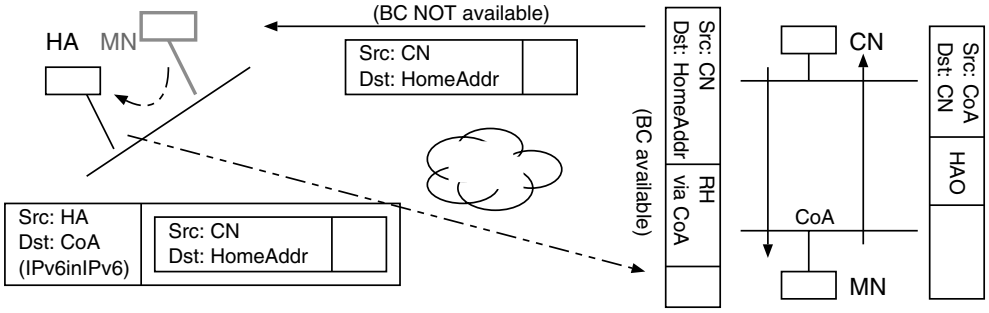

図 2.1 Mobile IPv6 の動作: MN からのパケットには必ず HAO が付与される。binding の情報 (BC) をもつ CN は、RH を使用して直接 MN にパケットを返す。そうでなければ HA を経由する。

ケットはまず CoA を持つノードへ到達する。このノードは MN であるので、MN はパケットを直接受信できる。この場合は冗長な経路は発生しない。

2.3 Mobile IPv6 アーキテクチャの問題点

Mobile IPv6 のアーキテクチャは、現在のインターネットの利用モデルと、これから期待される応用モデルを考えると、問題を多く抱えている。我々は、Mobile IPv6 のアーキテクチャ自身が持つ主な問題は以下の 4 点と考える。

- 三角経路による耐障害性の低下
- ヘッダ長増大によるオーバヘッド
- HA による耐障害性の低下
- HA 設置の困難さ

以下、これらの問題点について議論する。

2.3.1 三角経路による耐障害性の低下

通常はノード–ノード間のネットワークが正常であればその 2 つのノード間で通信が可能である。一方、HA を経由する三角経路は、移動ノードと通信相手間だけではなく、通信相手と HA 間、そして移動ノードと HA 間という 3 つのネットワーク経路が正常であることが要求される。これはネットワークの障害に対してより脆弱となることを意味している。また、End-to-End の通信に比べてネットワーク帯域をより多く消費し、また当然ながら遅延も大きくなる。

2.3.2 ヘッダ長増大によるオーバヘッド

Mobile IPv6 では、まず自分の Haddr を伝えるために IPv6 Destination Option である HAO を必ず付与しなければならない。また、移動ノードに対してのパケットにおいて、三角経路を避けるためには Routing Header を利用する。Mobile IPv6 を利用する 2 つの移動ノードが通信を行う場合を考える。この場合、パケットには HAO と RH が付与される。HAO は最低でも 20byte であり、RH は 24byte である。結果移動ノード間で行われる通信は Mobile IPv6 を使用する限り必ず 44byte のパケット長オーバヘッドが生じる。

現在 Voice over IP (VoIP) がインターネット上のアプリケーションで注目を集めているが、VoIP は小さいパケットで低遅延を要求する。また、ヘッダの圧縮等を行いできるだけオーバヘッドの削減を行っている。このようなアプリケーションにおいては、Mobile IPv6 のオーバヘッドは致命的と言える。

2.3.3 HA による耐障害性の低下

Mobile IPv6 では、HA に障害が発生した場合には、たとえば CN からの発呼が全くできなくなるなど、MN の通信に多大な影響を与える。よって、HA は MN の通信において不可欠な機能であるといえる。

このため、耐障害性を高めるために、HA を分散配置するなどしてシステムの冗長性を大きくすることが望まれる。しかし、HA の位置は Haddr に依存する。すなわち、ある Haddr のための HA は、ある唯一のサブネットワークにしか置くことはできない。HA を複数個置くことは可能であるが、HA を広域に分散配置することは不可能であり、耐障害性を高めることは難しい。

2.3.4 HA の設置の困難さ

HA の設置場所も問題となる。企業や大学など、固定的で常時接続のネットワークを持っている場合には問題はないが、ダイアルアップサービスなどの間欠接続ネットワークユーザは、どこかに HA を設置してもらう必要がある。

HA には移動ノードが生成する実トラフィックが流れ込む可能性があるため、仮に HA のサービスを行う場合、消費するネットワークのバンド幅は大きくなると予想され、これは第三者が容易に HA の代理サービスを行うことを難しくする。

また、Mobile IPv6 の Haddr が属するサブネットには必ず HA が必要となる。これは移動ノードの数に依存することなく、たとえあるサブネットに 1 台しか移動ノードが存在しなくても、HA はそのサブネットに最低でも 1 台用意しなければならない。これは設置コストを高価にする。

ゆえに HA の存在は、Mobile IPv6 を導入することに対する大きな障壁となりうる。

2.3.5 Mobile IP アーキテクチャの問題点についてのまとめ

以上の議論をまとめると、現在の Mobile IPv6 アーキテクチャが内包する問題はすべて HA の存在と、通信時に付加されるヘッダによるプロトコルオーバヘッド、そして発生する三角経路の 3 点に起源している。しかし Mobile IPv6 の根幹はこの 3 点ともいえるため、Mobile IPv6 自身がこれらの問題の解決することは非常に困難である。

よって、この問題を持たない新しい IP 層による移動透過性を提供するプロトコルが IPv6 に望まれると我々は考える。

2.4 提案方式

2.4.1 提案方式の方針

提案方式は、2.3 章で述べた Mobile IPv6 における 4 つの問題を回避する。加えて、提案方式を理解しない既存のノードとも通信可能にするべきである。これは緩かな移行を可能にし、新しいプロトコルの普及を容易にする。

よって、提案方式は以下の 5 点を満たすようにすべきと考える。

- ノード間の通信を End-to-End で終端
- 通信パケットのオーバヘッドの回避
- 通信に不可欠な機能には十分な冗長性が提供可能
- 導入の容易さ
- 既存ノードとの通信が可能

我々は、この 5 点を満たす、新しい移動透過性保証のプロトコルを提案する。

2.4.2 提案方式の概念

あるノードがネットワーク上を移動することを考える場合、ネットワーク層は次の 2 つの情報が必要であると考えられる。

- 位置指示子
  ネットワーク上のインターフェイスの位置を一意に定めるもので、経路制御に使用される。位置指示子は、ノードのネットワーク接続点に割当てられる。また、1 つのノードは複数の位置指示子をもってもよい。
- ノード識別子
  ネットワーク上のノードを一意に定めるもので、ネットワーク上の位置に依存しない。また、ノードが複数のネットワークに接続されている場合でも高々一つでよい。すなわち、ノード識別子はネットワークインターフェイスに与えられるのではなく、ノードそのものに与えられる。

本来、ネットワーク層である IP 層より上位の層は位置指示子を意識する必要はなく、ノード識別子により通信するべきである。これによってノードのネットワーク上の位置に関わらず同じ識別子を使用できるため、移動透過性保証が得られる。ノード識別子の形式は位置指示子の形式から独立にすることも可能である。

これを実現するには、ネットワーク層を位置指示子による経路制御のための層と、ノード間の通信の識別子として使用されるノード識別子の層に分割すればよい。このような手法によって移動透過性を保証するプロトコルに VIP[130] がある。

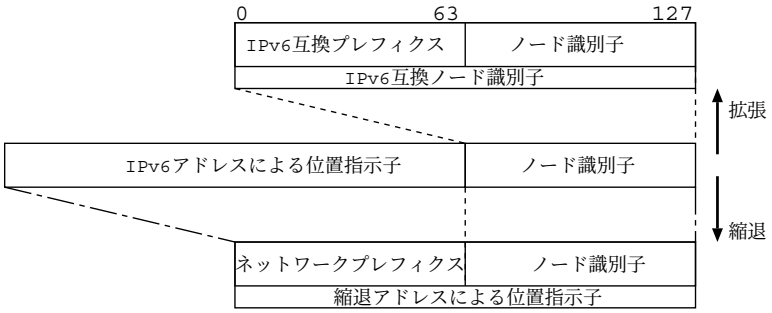

図 2.2 アドレスの縮退と拡張: 本来 128bit の位置指示子と 64bit のノード識別子である情報を、ネットワークプレフィックスとノード識別子の結合に縮退させて位置指示子とする。またノード識別子を IPv6 互換プレフィクスを使用して拡張する。どちらの結果も IPv6 のアドレス形式と互換性を持つ。

だが、ネットワーク層の単純な分割には 2 つの問題点がある。一つはヘッダ長のオーバヘッドである。位置指示子とノード識別子の 2 つの情報をパケットに与えなければならないために従来の Internet Protocol に対してプロトコルオーバヘッドの増加が避けられない。もう一つは、後方互換性が失われることである。なぜなら、この方式自体が IP アーキテクチャとは完全に異なっており、IP アーキテクチャそのものを変更する必要があるからである。

2.4.3 縮退アドレスモデル

提案方式は、前節で述べたオーバヘッドと後方互換性の問題を、IPv6 のアドレス空間の広さを利用することで解決する。

本方式では、ノード識別子として 64bit の値を使用する。また、位置指示子としては 128bit の IPv6 アドレスを使用する。このままでは、2.4.2 節でのべたように、ネットワーク層にはノード識別子と位置指示子、すなわち 192bit 分の情報が必要となる。

現在の IPv6 の Global address の割当ては、主に Aggregatable Global Unicast Address (AGUA)[131] に従っている。これは、128bit のアドレスのうち上位 64bit をネットワークプレフィクスに使用し、下位 64bit に EUI-64[132] 形式のインターフェイス識別子を埋め込む形になっている。

そこで、この 64bit のノード識別子を、AGUA のインターフェイス識別子として使用することによって、ノード識別子を位置指示子の中に縮退させる。以下、この方式の IPv6 アドレスを特に縮退アドレスと呼ぶ。縮退アドレスは、従来の AGUA 形式と互換性を持ちながら、128bit の位置指示子と 64bit のノード識別子の 2 つの意味を持っている。すなわち、パケットの受信者は、従来の IPv6 パケットに余分な情報を付加させること無く、ノード識別子と位置指示子を得ることができる。

しかし、従来の IPv6 アプリケーションすべてをこの 64bit ノード識別子を使うように変更することは、前節で述べたように互換性の問題から現実的ではない。このため、我々は IPv6 互換プレフィクスという概念を導入する。IPv6 互換プレフィクスは、あらかじめ定められた 64bit の固定値であり、本方式のノードは事前に知っているものとする。IPv6 互換プレフィクスとノード識別子を結合する。この拡張されたノード識別子は 128bit になる。以後この拡張されたノード識別子を IPv6 互換ノード識別子と呼ぶ。IPv6 互換ノード識別子は AGUA のフォーマットに準ずるものとなり、上位層から見るとこのノード識別子は従来の IPv6 アドレスとして扱うことができる。これらのアドレスの関係を図 2.2 に示す。

位置指示子である縮退アドレスは、ネットワーク内においてパケットのルーティングに使用される。一方、IPv6 互換ノード識別子はノード内でのみ意味を持つ。本方式のノードは、パケットの送受信の際、必要最小限の処理を行った後は縮退/拡張の相互変換を行う。すなわち送信時には、IPv6 互換ノード識別子から縮退アドレスへと縮退変換し、パケット内に埋め込む。受信時には、パケット内の縮退アドレスから IPv6 互換ノード識別子に拡張変換して処理を行う。

ノード内の IP 層と上位層では、IPv6 互換ノード識別子で処理され、これは意味的には完全にノード識別子で処理されることに等しい。また、ノード外

では、パケット内の縮退アドレスは正しいネットワークプレフィクスが付与された IPv6 アドレスであるので、従来の IPv6 ネットワーク内を適切に経路制御され、移動ノードへと到達する。

2.4.4 ノード識別子からの位置識別子の解決

ここで問題になるのが、あるノード識別子は現在どのような位置指示子を持っているのかという情報である。以下このノード識別子と位置指示子の関係を binding と呼ぶ。本方式では、移動ノードと通信するためには IPv6 互換ノード識別子と現在のノードの位置指示子の相互変換を行うが、この変換を行うためには binding を知る必要がある。

Mobile IPv6 や VIP に代表される従来の移動透過性保証は、ノード識別子が IP アドレスと互換であることを利用して、実際の IP パケットの宛先としてこのアドレスを埋め込み、HA 等で転送し、必要であれば binding を通知するという方式であった。しかし、このような HA の存在には 2.3 章で述べた問題がある。

本方式では、実際にネットワークを通過する IP パケットの中には IPv6 互換ノード識別子を使用せず、必ず縮退アドレスを使用する。この方式で考えなければならないのは、いかにして binding を取得するかということである。ノード識別子の空間は 64bit と広大であるので、1 つのデータベースで集中管理するのは明らかに現実的ではない。類似の情報検索として現在インターネット上で機能しているものに Domain Name System (DNS) がある。あるノード識別子から位置指示子を得るという検索は、IP アドレスに割付けられた Fully Qualified Domain Name (FQDN)[1]を知るという、DNS の機能に近い。DNS は現実的に動作しており、DNS で行っているような分散管理はうまく働くように考えられる。よって binding の取得の方法として、Dynamic DNS update[133] などを利用し、動的な binding 情報を DNS サーバに管理させる手法が考えられる。しかし、DNS は変更頻度の少ない静的なデータを扱うように設計されている。この設計ゆえに cache 等が効果的に働き、DNS が破綻することなく動作しているとも言える。一方、binding は頻繁に変更される情報である。よって DNS で直接扱うには問題がある。

我々はこの問題の解決のため、Binding Agent (BA) を導入する。Binding Agent はノードの binding を管理する。DNS には、IPv6 互換ノード識別子と BA との対応を登録する。これは、ある IPv6 アドレスがどのような FQDN 名を割当てられているかを DNS に問合わせることができるのと同じように、ある IPv6 互換ノード識別子がどの BA と関連付けされているかを DNS に問合わせることが出来ることを意味する。あるノードと BA との対応を変更することは稀であると考えられる。よってこの対応情報は静的な情報であり、DNS 上でうまく稼働することが予想される。また、ノードと BA の対応を IPv6 互換ノード識別子と BA の IP アドレスで表現することによって、既存の DNS には変更を加えること無く利用できる[2]。ノードの binding についての問合わせは、BA に対して直接行なう。すなわち、静的な情報の関係は従来の DNS のシステムを利用し、動的な情報については BA との直接の通信で解決する。

よって、あるノードの binding を得るには、まず DNS を使って IPv6 互換ノード識別子から BA のアドレスの集合を取得し、BA に対して binding を要求すればよい。得られた binding は一時的に記憶される。この記憶領域を binding table と呼ぶ。binding はライフタイムを持ち、ライフタイムの切れた binding は破棄される。

また、移動ノードは、移動したら必ず自ノードに関連づけられている BA に対して、現在の binding を登録する。

提案方式の通信機構をまとめたものを図 2.3 に示す。

2.4.5 ノードの移動

移動したノードは、自分の BA に現在の binding を再登録する。また、現在の通信相手に対して、この新しい binding 情報を通知する。現在の通信相手は、ノードが保持している他ノードの binding を調べることで得られる。binding が何らかの理由で消えている場合 (たとえば binding の保持のための記憶領域以上に通信相手が存在するなど) や、パケットロスなどで binding の通知が途中で失われた場合は、通信相手は新しい binding を得られない。これ

[1] FQDN とは、たとえば foo.bar.com といった形式で表されるノードに割当てられた名前である

[2] 但し最終の DNS サーバには BA を表すレコードを理解できるように拡張を施す必要がある

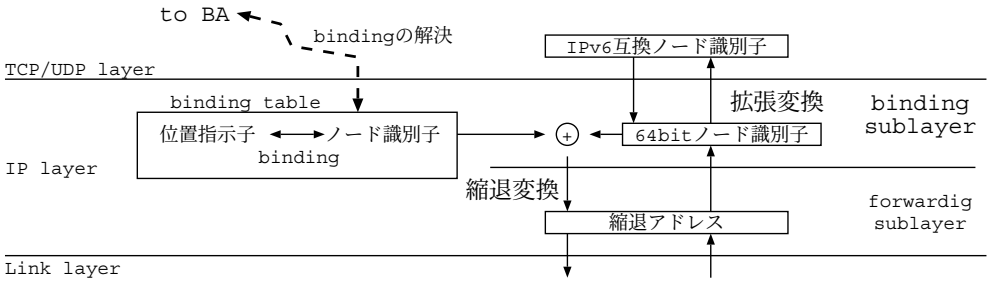

図 2.3 提案方式の通信機構: 位置指示子と IPv6 互換ノード識別子は binding を使用して相互に変換される。上位レイヤは IPv6 互換ノード識別子で処理され、経路制御は位置指示子で行われる。binding は BA に問合わせることで解決される。

に対処するために、本方式では ICMP Destination Unreachable Message[134](dst unreach) を利用する。dst unreach は、ノードへの到達性が失われた場合に、経路上のルータからパケットの送信者に対して送られる。本方式の通信者は、移動ノードへのパケットに対して dst unreach が返された場合、binding を BA に問合わせる。得られた binding が変更されていれば新しい位置指示子に対してパケットが送られる。変更がなければ、binding の内容を変更せず、dst unreach も上位層に伝えない。binding のライフタイムが切れたところでもう一度 binding を問合わせる。binding が新しくなっていれば、新しい位置指示子に対してパケットが送られる。これは、dst unreach が返ったときは移動中であり、ライフタイムが切れるまでに移動が完了したことを意味する。binding が変わらない、あるいは BA の持つ binding もライフタイムが切れて失われていた場合には、dst unreach が返ってきたことを上位層に伝える。これは対象の移動ノードが長時間オフライン状態であると考えられる。

2.4.6 従来ノードとの通信における互換性

従来ノードとの通信について考える。本方式で、従来ノードと通信をする場合には、binding を使用した IPv6 互換ノード識別子と位置指示子との変換を行わなければよい。パケットの送信の際には、上位層から与えられたアドレスが IPv6 互換ノード識別子である場合は本方式を使用し、そうでなければ従来方式を使用する。パケットの受信の場合は、パケットの送信者が本方式を使用しているかそうでないのかを判断する必要がある。しかしパケット自体は従来方式と本方式との間に差はないため、このままでは判断できない。

望ましい解は IP ヘッダの中の 1bit を取得して、本方式か従来方式かを示すというものである。だがこの方法では、従来ノードすべてがこの 1bit を処理するコードをすぐに必要とするために現実的ではない。

そこで我々は、緩やかな移行を行うために 64bit のノード識別子の形式を EUI-64 に適合させる。EUI-64 は、先頭 24bit は IEEE によって割当てられる Organizationally Unique Identifier (OUI) であり、残り 40bit は OUI の管理者が値を割付ける形式になっている。本方式では、OUI の割当てを受けて、64bit のノード識別子にこの値を使用し、この値をもって本方式を使用するかどうかを判定する。

パケットの受信者は、パケット内のアドレスが AGUA の形式である場合、インターフェイス識別子の先頭の 24bit が割当てられた OUI であれば、本方式を使用した通信であると判断する。そうでなければ、従来方式の通信であるとする。AGUA のインターフェイス識別子は EUI-64 を使用することが要求されているため、従来ノードが本方式の OUI から始まるインターフェイス識別子を使うことは考えにくい。

この方式ではノード識別子として使用できる空間は 40bit となる。これは現在の IPv4 よりも広い空間であり、この方式が普及するまでの時間であれば十分な空間であると判断する。万が一移行途中において 40bit の空間が枯渇した場合でも、新たな OUI を取得すれば良い。

ただし、提案方式では従来ノードとの通信では可搬性のみ保証され、移動透過性は保証されない。

2.4.7 提案方式の通信例

具体的な通信例をもちいて、本方式の通信手順を説明する。図 2.4 のようなネットワークを例にとる。

以下 IPv6 互換プレフィクスを O、IP パケット内の source address を src, IP パケット内の destination address を dst と表記する。MN1、MN2 は本方式を実装したノードであり、CN は従来ノードとする。また、各ネットワークで広告されるネットワークプレフィクスを G1, G2, G3 とする。MN1, MN2 が持つ 64bit ノード識別子を i1, i2 で表し、MN1, MN2, CN が持つ、AGUA のインターフェイス識別子はそれぞれ e1, e2, c とする。

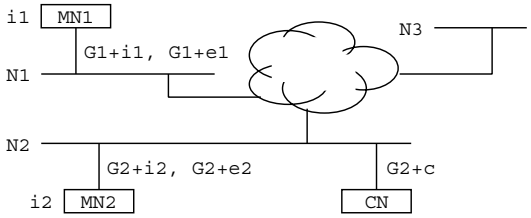

図 2.4　ネットワーク例

2.4.8 初期状態

ノード識別子 i1 を持つ移動ノード MN1 がネットワーク N1 に接続されることを考える。MN1 は最低でも 2 つの global address を得る。1 つは従来の AGUA (G1+e1 とする) であり、もう一つはルータから得たプレフィックスを上位 64bit、ノード識別子を下位 64bit とした縮退アドレス (G1+i1 とする) である。

MN1 は、まず自分の BA に現在の binding、すなわち (i1, G1+i1) を通知する。MN1 は近傍の DNS サーバに O+i1 に対応する BA のリストを要求する。これによって MN1 は自分の BA のリストを取得する。仮に BA11, BA12 とする。MN1 は得られた BA のうち、任意の BA に対して binding を登録する。BA は、あるノードに関わる BA のリストを DNS から同様に取得して、BA 間の binding 情報の一貫性を保つ。

2.4.9 本方式のノードとの通信

ノード識別子 i2 を持つ移動ノード MN2 に対する通信を考える。MN2 の BA を BA21 とし、すでに MN2 の binding 情報の登録は終了しているとする。MN2 の縮退アドレスは G2+i2 であるとする。MN1 上のアプリケーションは、MN2 に対して通信を行うときは、O+i2 を指定する。するとこれは本方式の通信であることがわかるので、パケットの出力の際の src, dst に対して縮退アドレスを使用する。MN2 の binding を持たないため、縮退変換はまだ行えない。MN1 は MN2 の BA を DNS に問合わせ、BA21 のアドレスを得る。MN1 は BA21 に対して i2 の binding を問合わせる。MN1 は得られた binding から縮退変換を行い、MN2 の縮退アドレス G2+i2 を得る。よって MN1 から MN2 へのパケットの src は G1+i1 であり、dst は G2+i2 となる。

MN2 がこのパケットを受け取ると、まず下位 64bit を検査し、OUI が本方式を使用する値であるかどうかを確認する。この場合本方式を使用する値であるので、縮退アドレスは IPv6 互換ノード識別子に拡張変換される。

よって、MN1-MN2 間の通信は、ネットワーク上を流れるパケットは G1+i1 – G2+i2 間の通信に見えるが、MN1 および MN2 の上位層は O+i1 – O+i2 の通信として認識することになる (図 2.5)。

2.4.10 ノードの移動

MN1 がネットワーク N3 へと移動する場合を考える。 MN1 はここでは G3+e1, G3+i1 のアドレスを得たとする。 MN1 は BA21 に対して新しい binding (i1, G3+i1) を通知する。自分の BA が BA21 であることはキャッシュできるので DNS に問合わせる必要はない。つぎに、MN1 は自分の持つ binding table を見る。(i2,G2+i2) が存在するので、G2+i2 に対して現在の binding を通知する。MN2 はこれを受け取ると、i1 についての binding を更新する。よって、MN1 と MN2 の通信は継続できる。

2.4.11 従来ノードとの通信

IPv6 アドレス G2+c を持つ従来ノード CN に対する通信を考える。MN1 上のアプリケーションは、CN に対して通信を行うときは、G2+c を指定する。するとこのアドレスは IPv6 互換プレフィクスを持たないことから、従来ノードとの通信であることがわかるので、binding も、縮退アドレスも使用しない。よって、ネットワーク上を流れるパケットは、G1+e1 – G2+c 間の通信となる。CN からの応答パケットは、下位 64bit である c を検査する。c の OUI は本

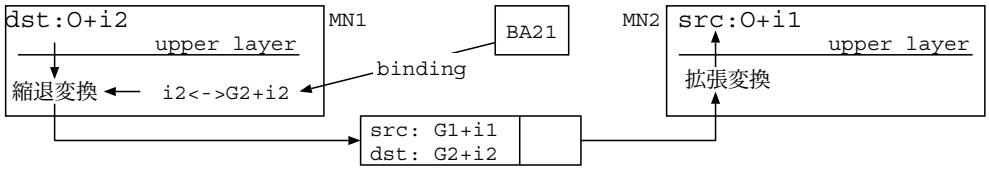

図 2.5 本方式の通信例: MN1 と MN2 の上位層は IPv6 互換識別子でおたがいを認識し、ネットワーク上を流れるパケットは縮退アドレスが使用される。

方式の OUI ではないため、binding を使用しない従来の通信であると判断される。

なお、移動ノードと、BA および DNS サーバ間の通信には、この従来方式が使用される。

2.5 提案方式についての考察

本章では、2.4.1 節で述べたプロトコルの設計方針をふまえて、Mobile IPv6 との比較を行いながら提案方式の考察を行う。

2.5.1 移動ノードとの End-to-End 通信

提案方式は、各ノードがノード識別子と縮退アドレスの変換を各個に行うため、通信は必ず End-to-End で終端する。一方、Mobile IPv6 では、通信相手の binding cache が存在しない場合および、はじめて移動ノードに対してパケットを送信する場合には、パケットは HA を経由するので三角経路が発生する可能性がある。

2.5.2 通信に不可欠な機能の冗長性

提案方式では、binding の解決のために BA が必要であり、BA が通信に不可欠な存在となっていることは否定できない。一方、Mobile IPv6 では、HA が通信に不可欠な存在となっている。しかし、提案方式における BA は、HA のようにあるアドレスに束縛された位置に配置する必要はない。あるノードのための BA はネットワーク上の任意の位置に配置することが可能となる。これは、BA による方式が HA を用いる方式にくらべてより高い耐障害性を持つことを意味する。

2.5.3 導入の容易さ

提案方式では BA を導入する必要がある。BA の台数や配置場所は、BA が管理するノードの数と、ノードの重要度に応じて、管理者が自由に設定できる。HA のように、移動ノードの Haddr が所属するサブネットごとに HA を置く必要はなく、設置コストは非常に低いといえる。また、移動ノードは自ノードが登録するべき BA を知っている必要はなく、DNS に問合わせれば良い。これは各ノードの管理コストを引き下げるという利点を持つ。

2.5.4 通信パケットにおけるオーバヘッドの回避

提案した縮退アドレスアーキテクチャは、一つの IPv6 アドレスに位置指示子とノード識別子の情報が含まれている。このため、位置指示子とノード識別子の分割というプロトコルデザインをもちながら、ノード間の通信に使われるパケットは従来の IPv6 ヘッダのままでよい。すなわち、通信パケットのオーバヘッドは存在しない。一方、Mobile IPv6 では、通信パケットに Home Adress Option と Routing Header を付加する必要があり、大きなオーバヘッドを必要とする。

2.5.5 既存ノードとの通信

提案方式は既存ノードとの通信が可能であるが、既存ノードとの通信では移動透過性はなく、可搬性のみの保証となる。一方、Mobile IPv6 は既存ノードとの通信でも移動透過性が保証される。IPv6 のスタックを搭載した OS はすでに販売も行われており、普及段階にはいりつつあると考えられる。しかし、現在使用されている IPv6 のプロトコルスタックを見ると、Mobile IPv6 が要求する Home Address Option の処理が実装されているものはほとんどない。Mobile IPv6 では、HAO を処理できないノードとの通信は困難なため、Mobile IPv6 をこれから普及させるにはこれは大きな障害になるのではないかと危惧される。提案方式にはこのような問題はないため、緩やかな移行が可能である。ゆえに、後方互換性という点においても本方式は Mobile IPv6 に対して優位であると言える。

2.6 おわりに

本報告では、現在提唱されている Mobile IPv6 の問題点を指摘し、これらの問題を持たない新たな移動透過性プロトコルの提案を行った。

本方式では縮退アドレスモデルを提案し、このモデルが移動透過性を保証しながらも、End-to-End で通信が完結することを示した。同時に提案方式は後方互換性を持つため、従来ノードとは可搬性での通信が可能である。よって提案方式は、緩かな移行が可能である。

現在我々は NetBSD 上で本方式の実装を行っている。今後は実装を用いた本方式の性能評価と運用実験を行い、提案方式が現実のインターネット上で動作することを証明していきたい。

第 3 章　移動型計算機のトラフィック解析

3.1 はじめに

近年、可搬型計算機は急速に普及し、これを利用してインターネットに接続する場面も順調に増加している。一般に可搬型計算機は一人のユーザによって利用され、さまざまな場所でネットワークに接続される。この際、モデム、イーサネット、無線 LAN など様々なネットワーク接続メディアが利用される。この様な特殊な利用環境のため、固定型計算機と比較した場合、可搬型計算機のネットワーク利用パターン、つまりネットワークトラフィックは特徴的であると言われている。しかし、実際には可搬型計算機のネットワークトラフィックを統括的に解析した例は無い。

本研究では、この様な現状を踏まえ、可搬型計算機のネットワークトラフィックを統括的に解析し、その特徴を明らかにする。数名の被験者の協力を得ることにより、実際に利用されている可搬型計算機のネットワークトラフィックを長期に渡り収集し、収集されたデータを後処理によって解析する。

本研究の成果は、可搬型計算機で使われるアプリケーションの設計や、各種ネットワークエンティティのパラメータ設定に有効な情報を提供する。可搬型計算機はその性質上、余分な資源を消費することが許されない場合が多い。PDA 等の小型計算機やネットワーク家電等の組み込み型計算機では、大きなメモリ領域を消費するような内部表をネットワークエンティティが持つことは許されない。また、ユーザがネットワークを利用する時の特徴を知ることによって、アプリケーション間の連係を図ることが可能となり、より良い利用環境をユーザに提供することが可能となる。

本報告は以下の様な構成から成る。始めに、3 章でこれまでのネットワークトラフィック解析について調査し、3.2.3 章において可搬型計算機のトラフィック解析を行う為の方針と手法を固める。3.3.2 章では収集したトラフィックデータをもとに実際に多くの角度から解析を行なう。その後、3.4.6 章において可搬型計算機のネットワークトラフィックの特徴について考察し、3.4.6 章で本報告を締めくくる。

3.2 ネットワークトラフィック解析の現状

これまでにも様々なネットワークトラフィック解析の試みが行なわれてきた。これらの解析は、トラフィックのモデリングを目的としたもの、管理目的のもの、統計目的のものなどその目的も様々である。本章では、これらのネットワークトラフィック解析の現状について述べる。

3.2.1 モデリング目的の解析

ネットワークトラフィックのモデリングを目的としたトラフィック解析には、パケットの遅延に着目したものが多い。パケットの遅延をモデリングすることによりアプリケーションの挙動を明らかにすることを目的としている。これまでの研究では、エンドエンドでの遅延をポアソン分布、自己相関関数などで表現する試みが行われてきた [135][136][137]。また、これらのモデリングには相転換が存在するといった報告もある [138]。

これらの研究では、実際のトラフィックをプローブパケットを使って測定し、得られた遅延から解析を行っている。つまり、細かなデータを予め能動的に収集しておき、後程解析する手法をとっている。また、アプリケーションの性質を記述する場合には、バルク転送とインタラクティブ通信の様な極端なものを対象としている場合が多い。

3.2.2 管理、統計目的の解析

ネットワークの管理やネットワークを正常に保つ為の統計データ収集を目的としたネットワークトラフィック解析も継続して行われている。これらの解析は明確な目的をもって行われていることが多く、収集しているデータも多種に渡っている。

データ収集のツールとしては、以前から Information Science Institute (ISI) により配布されていた NNstat[139] が利用されてきた。NNstat は設定により、観測点を通るパケットの送受信者毎、ポート毎等の統計情報を得ることができる。NNstat を利用して得られた統計情報はリンク上を通るパケットの性質 (通信先分布、利用されているアプリケーションの分布) を表す指標として利用される。

NNstat より、より細かな情報が欲しい場合は、`tcpdump`[140] を利用した全パケットキャプチャによる解析が行われる。予めキャプチャしておいた情報をもとに、欲しい情報を取り出す。多くの場合、欲しい情報毎にその都度適当な処理が施されることとなり、一般的な手法は存在しない。この手法は、収集目的が曖昧な時に有効であるが、記憶容量を大量に消費する欠点がある。

日常の大まかトラフィックを知る為には、近年、MRTG[141] [142] が多く利用される。MRTG は基本的には Simple Network Management Protocol (SNMP) を用いてルータ等から情報を収集し、定期的にグラフを作成するツールである。設定により、多数の観測点の情報を設定したスケールでグラフにできる為、トラフィック異常の発見が可能であり、ネットワーク管理用のツールとして有効である。

このような解析手法はインターネットの多様化とともに一般的なものを用意することが難しくなっている。現状では、多くの場合、管理者が状況に応じて様々なツールを組み合わせることにより管理を行っている。

3.2.3 現状の解析に関する考察

3.2.1、3.2.2 節で述べたネットワークトラフィック解析の現状をみると、これまでの解析はインターネットの挙動を解析することが主な目的であったことが分かる。インターネットの挙動を知ることにより、ネットワークの異常検知をおこなったり、インターネットで利用するのにより適したトランスポート層プロトコルの開発等を行なっている。

これまでのネットワーク解析は明確な仮定と意味をもってインターネットの性質を調べるものがほとんどである。一方で、あるホストに着目したネットワークトラフィックの挙動が調べられることは殆ど無かった。

本研究では、可搬型計算機上のアプリケーションやプロトコル作成、または、パラメータ調整を行うにあたり、おおまかな可搬型計算機の挙動を知ることを目的としている。この様な目的の下では、予め目的のホストのトラフィック情報を収集しておき、後処理による解析を行うことが適当である。

3.3 解析方針と手法

本研究の目的は、現在明らかになっていない可搬型計算機のネットワークトラフィックの特徴を明らかにすることである。前章で述べたように、本研究の目的では様々な視点で解析を行う必要があるため、予め細かなトラフィック情報を収集・保存しておき、後に解析をおこなう手法が適当である。

3.3.1 トラフィック収集

本研究における解析では、ネットワークトラフィック情報を収集・蓄積しておき、後にそれを解析する手法をとる。ここで、まず、問題となるのが収集の手法である。今回は、ネットワークトラフィックの情報収集の手法として、Lawrence Berkeley Laboratory (LBL) の `tcpdump` コマンドを利用することにした。

`tcpdump` は Berkeley Packet Filter (BPF)[143] 等のインターフェイスを介して、あるネットワークインターフェイスを通ったパケットを取得するためのツールである。`tcpdump` ではフィルタを設定することによって、ある条件に合致するパケットをタイムスタンプとともに取得することができる。また、取得にあたってはパケットの先頭からのバイト数を指定して、パケット全体ではなく、一部を取得することも可能である。

今回の解析では、対象が可搬型計算機である。そこで、PC-Card のネットワークデバイスを介してネットワークを利用する場合は、PC-Card が挿入された時、ダイアルアップ型のネットワークデバイスを使用した場合は、Point to Point Protocol (PPP) のリンクが確立した時に、自動的に自分が送受信したトラフィック情報の収集を開始するような仕組み

を用意して実際のデータ収集を行なった。収集した情報は、一時、可搬型計算機ローカルのディスクに蓄積され、適当なタイミングでユーザによって、より大きなディスクを持つ固定型計算機に移すようにした。

最終的には、収集された情報から、TCP/UDP/IP 等のヘッダ情報のみを抜きだして保存してある。ヘッダの抜きだしには、WIDE Project による tcpdpriv [144] を利用した。

今回の研究では、日常的に BSD 系の UNIX をノート計算機で利用している研究者 5 人の協力を仰いで行なった。収集した期間は 1999 年約 1 年間であるが今回の解析では全てのデータが揃っている 6 月 6 日から 6 月 19 日の 2 週間分のみを対象とした。被験者は自動的に tcpdump を起動するプログラムを組み込んだノート計算機をほぼ毎日利用していた。

3.3.2 トラフィック解析

トラフィックの解析は、大きく 3 種類に分類することができる。一つはインターネットへの接続毎の解析、二つ目はパケット毎の解析、最後の一つはコネクション毎の解析である。

今回のネットワークトラフィック収集ではノート型計算機がインターネットに接続される度に tcpdump コマンドを起動してパケットをキャプチャし、ファイルを作成している。このため、インターネットへの接続毎の履歴を得ることができる。インターネットへの接続毎の解析はこのことを利用して行なった。

パケット毎の解析は、tcpdump コマンドと独自のツールを用いて解析を行なった。tcpdump によって目的のパケットを取り出し、独自ツールでパケットのバイト数の計数などをして処理をした。

収集した情報では UDP のトランザクションを得ることは難しい為、コネクション舞いの解析は tcptrace[145] を利用して行なった。tcptrace による解析をもとに、送受信者アドレス、ポート番号、転送バイト数、転送パケット数を測定し、出力したものを目的毎に解析した。

3.4 可搬型計算機のトラフィック解析

本章では前章で述べた手法を使って、実際のトラフィックを解析する。解析は、ユーザの挙動についての解析、パケット単位での解析、コネクション単位での解析および固定ホストとの比較について行なう。

3.4.1 被験者

始めに各被験者のノート型計算機利用の特徴についてまとめておく。

被験者 1

基本的に Ethernet、無線 LAN、PHS を利用する。無線 LAN はオフィスで使用し、Mail、Web、ssh 等のアプリケーションを利用する。Ethernet は自宅のネットワークで利用し、自宅はほとんどがテレホーダイ時の ISDN によるダイアルアップである。PHS は外出時に Mail を読むのに利用する。

被験者 2

基本的に Ethernet、無線 LAN、モデムを利用する。Ethernet はオフィスで使用し、Mail、Web、ssh、NFS などのアプリケーションを利用する。無線 LAN は会議室など別室に移動した時に、主に Mail、Web に利用する。モデムは自宅からの利用で、Mail、Web、ssh が主な利用アプリケーションである。

被験者 3

基本的に Ethernet、モデムを利用する。Ethernet はオフィスで使用し、ssh、ウインドウの表示などほとんどの作業をここでおこなう。モデムは 33.6kbps で自宅からの接続で、ssh による Mail、Web などに利用する。

被験者 4

オフィス、自宅ともに Ethernet を利用する。自宅は ISDN ダイアルアップにより接続されている。オフィス、自宅ともに Mail 以外の用途にはほとんど利用しない。

被験者 5

基本的に Ethernet、無線 LAN、モデムを利用する。Ethernet はオフィス及び外出先のネットワークに接続して Mail、Web、ファイルのバックアップなどに利用している。無線 LAN も Ethernet とほぼ同等に利用しているが、移動する機会が多い時にとくに利用する。モデムは自宅からのダイアルアップ接続で、テレホーダイ時の利用以外では、ほぼ、Mail のみ利用している。

表 3.1 移動特性

	移動回数	接続時間				
		合計 (秒)	平均 (秒)	最大 (秒)	最小 (秒)	標準偏差 (秒)
被験者 1	85.0	853000	10000	56100	8.78	14900
被験者 2	58.0	440000	7580	41500	10.5	10300
被験者 3	37.0	252000	6820	37800	122	10300
被験者 4	48.0	246000	5130	32900	48.3	8340
被験者 5	38.0	231000	6080	65000	11.5	12400
全体	266	2020000	7600	65000	8.78	12041
平均	53.2	404000	7590			

表 3.2 利用プロトコル分布 (パケット数)

	TCP	UDP	ICMP	その他
被験者 1	79.1	17.2	3.7	0.0
被験者 2	42.7	53.9	0.1	3.3
被験者 3	41.4	58.5	0.1	0.0
被験者 4	71.3	24.1	4.6	0.0
被験者 5	77.3	22.0	0.7	0.0

(%)

表 3.3 利用プロトコル分布 (バイト数)

	TCP	UDP	ICMP	その他
被験者 1	93.0	6.2	0.8	0.0
被験者 2	33.7	63.1	0.0	3.2
被験者 3	11.1	88.9	0.0	0.0
被験者 4	76.9	21.2	1.9	0.0
被験者 5	89.4	10.5	0.2	0.0

(%)

表 3.4 利用ポート分布 (パケット数)

	SMTP	POP	HTTP
被験者 1	1.2	31.0	14.9
被験者 2	1.0	3.7	1.1
被験者 3	0.2	0.0	0.2
被験者 4	1.6	8.2	15.8
被験者 5	0.1	12.8	51.5

(%)

3.4.2 移動

今回行なったトラフィック収集方法では、インターネットに接続されるたび (移動するたび) に新たなファイルが作成される。このことを利用して 2 週間の移動回数と接続時間を計数した。結果は表 3.1 の通りである。2 週間の移動回数の平均は 53.2 回となった。1 日平均 3.80 回程度移動していることになる。これは、自宅→オフィス→自宅の様に落ち着いた場所ではネットワークに接続するという挙動を示している。

表 3.5 利用ポート分布 (バイト数)

	SMTP	POP	HTTP
被験者 1	0.3	6.0	10.7
被験者 2	0.7	3.2	1.3
被験者 3	0.2	0.0	0.4
被験者 4	2.3	13.4	31.4
被験者 5	0.0	10.7	43.5

(%)

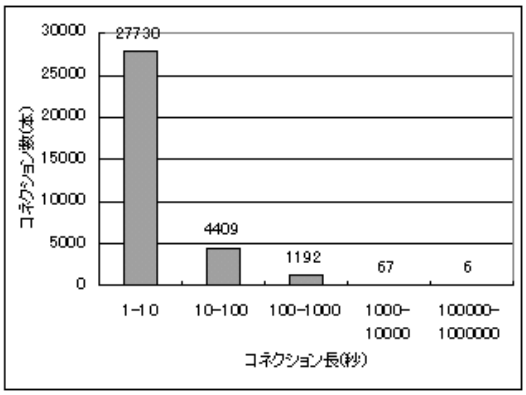

図 3.1 コネクションの接続時間分布

表 3.8 最大同時接続コネクション数

	本数 (本)
被験者 1	14
被験者 2	9
被験者 3	14
被験者 4	12
被験者 5	13

3.4.3 プロトコル分布

被験者が利用しているプロトコルの調査をおこなった。パケット数およびバイト数別の利用プロトコル分布を表 3.2、3.3 に示す。

ここで、被験者 2 の UDP はほとんどが NFS であったのに対し、被験者 3 の UDP トラフィックはほとんどが SMB のブロードキャストパケットを受

表 3.6　コネクションの接続時間

	本数 (本)	合計 (秒)	平均 (秒)	最大 (秒)	最小 (秒)	標準偏差 (秒)
被験者 1	15700	201000	12.8	3870	0.100	70.0
被験者 2	11900	342000	28.1	29900	0.100	506
被験者 3	297	58500	197	3920	0.103	498
被験者 4	1690	59500	35.2	5320	0.100	219
被験者 5	3840	123000	31.9	18400	0.107	334
全体	33400	776000	23.2	29900	0.100	333

表 3.7　コネクション毎の転送バイト数

	本数 (本)	合計 (byte)	平均 (byte)	最大 (byte)	最小 (byte)	標準偏差 (byte)
被験者 1	15700	265000000	16900	34300000	2.00	468000
被験者 2	11900	204000000	17100	27200000	1.00	388000
被験者 3	297	14800000	49800	3500000	28.0	259000
被験者 4	1690	35900000	21300	3220000	6.00	127000
被験者 5	3840	66400000	17300	6330000	18.0	144000
全体	33400	586000000	17500	34300000	1.00	400000

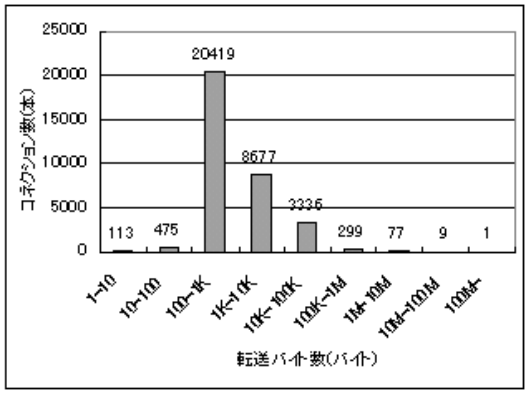

図 3.2　コネクション毎の転送バイト数分布

表 3.9　コネクション当たりの転送バイト数

	SMTP	POP	HTTP
被験者 1	2310	687	5500
被験者 2	6550	12900	5700
被験者 3	2470	-	11400
被験者 4	11100	32700	9680
被験者 5	2200	26500	8800

(バイト)

け取ったものであった。被験者 3 の場合通信量が少ない為、このようなトラフィックが目立ったものとして現れた。

また、TCP トラフィックのうち、よく可搬型計算機で利用されると言われている Mail、Web のトラフィック比率を調査した。調査結果を表 3.4 および表 3.5 に示す。ただし、SMTP はポート 25 番を、POP はポート 110 番を HTTP はポート 80 番を使うと仮定している。

3.4.4 TCP コネクションの解析

TCP にしぼって可搬型計算機のコネクションの解析を行なった。

まず、コネクションの接続時間についての解析を行なった。この際、データを全く送ることができなかったコネクションは予め除去した。結果を表 3.6 に示す。TCP コネクションの接続時間は被験者によって個人差が大きいことが分かる。特に、被験者 3 は他の計算機にリモートログインをおこなうことが多く、他の被験者より接続時間がかなり長くなっている。

次に、転送データのバイト数で同様の解析をおこなった。バイト数は送受信の合計である。解析結果を表 3.7 に示す。

全てのホストによるコネクション毎の接続時間および転送バイト数の分布を図 3.1、3.2 に示す。

次に、可搬型計算機で同時に張られるコネクション数を計測した。同時に張られるコネクション数の最大数を知ることにより、内部のメモリ領域をどれくらい確保しておけば良いかを知ることができる。各被験者における最大同時接続コネクション数を表 3.8 に示す。

また、各ポート別の 1 コネクション毎の転送バイト数の平均を計数した。結果を表 3.9 に示す。ここで、被験者 3 の POP のデータが無いのは POP を

表 3.10 コネクションの接続時間

	本数 (本)	合計 (秒)	平均 (秒)	最大 (秒)	最小 (秒)	標準偏差 (秒)
ラップトップ	3840	123000	31.9	18400	0.107	334
デスクトップ	1690	60200	35.6	19600	0.100	522

表 3.11 コネクション毎の転送バイト数

	本数 (本)	合計 (byte)	平均 (byte)	最大 (byte)	最小 (byte)	標準偏差 (byte)
ラップトップ	3840	66400000	17300	6330000	18.0	144000
デスクトップ	1690	264000000	146000	156000000	3.00	3830000

利用しなかったことを示している。

3.4.5 固定ホストとの比較

可搬型計算機と固定型計算機の比較をおこなうために、被験者 5 (オフィスでは Mail 以外は主にデスクトップ型計算機を利用) のオフィスのデスクトップ型計算機にラップトップ計算機と同様の仕掛けをおこない、トラフィックを収集した。この時のコネクションの接続時間およびコネクション毎の転送バイト数をそれぞれ表 3.10、3.11 に示す。また、同時に張られたセッション数はラップトップが 13 本だったのに対して、デスクトップでは 10 本であった。

3.4.6 個人のトラフィックパターン

収集したネットワークトラフィックをもとに、5 分毎のデータ送受信バイト数を計上し、送受信レートのグラフを作成した。グラフを図 3.3、3.4、3.5、3.6、3.7 に示す。

これらのグラフをみると、それぞれのユーザは 2 週間を通じて、同じようなトラフィックを生成していることが分かる。

3.5 考察

本研究により、可搬型計算機のネットワークトラフィックを解析することにより、ある程度ユーザの挙動を知ることができることが分かった。表 3.1 のように可搬型計算機がインターネットに接続された時間を知ることによってユーザがどのようにインターネットを利用しているのかがわかる。また、更にポート毎に細かく見ていくことによって、どのようなアプリケーションをどのようなタイミングで利用しているかを知ることも可能となる。今回は `tcpdump` を利用してネットワークトラフィックを蓄積したが、新たな蓄積/解析用のソフトウェアを構築する際もこのことを考慮することにより、ユーザの挙動をも推し量ることができるソフトウェアを開発することが可能である。

また、表 3.2、3.3、3.4、3.5 より、可搬型計算機のネットワークトラフィックは個人差が大きいことが分かる。ユーザはあるアプリケーションを設定すると、通常、そのアプリケーションを使い続ける。また、ある作業をするための環境をひとつの計算機上に構築すると、その作業は主に設定した計算機で行なうようになる。そのため、同じプロトコルであっても利用しているアプリケーションや計算機環境によってネットワークトラフィックに個人差が生まれることになる。逆に、ある人のネットワークトラフィックパターンは一度決まってしまえば、そのパターンに従うと言うこともできる。このことは、表 3.6、3.7 でも同様の結果となっている。

可搬型計算機を利用する場合、同時に張られるコネクション数は表 3.8 にあるようにあまりユーザに依存しないことが分かった。これは、一つの計算機を利用しているユーザが一人であるため、あまり多くのアプリケーションを同時に実行していることが少ない為である。同時に張られるコネクション数はアプリケーションより、その計算機を同時に利用しているユーザ数に左右されると考えられる。

3.6 結論と今後の課題

予想された結果ではあったが、今回の解析では被験者数が少なかった為、個人差が結果に大きく反映されることとなった。しかし、逆にこのことによって、可搬型計算機では個人の設定がそのトラフィッ

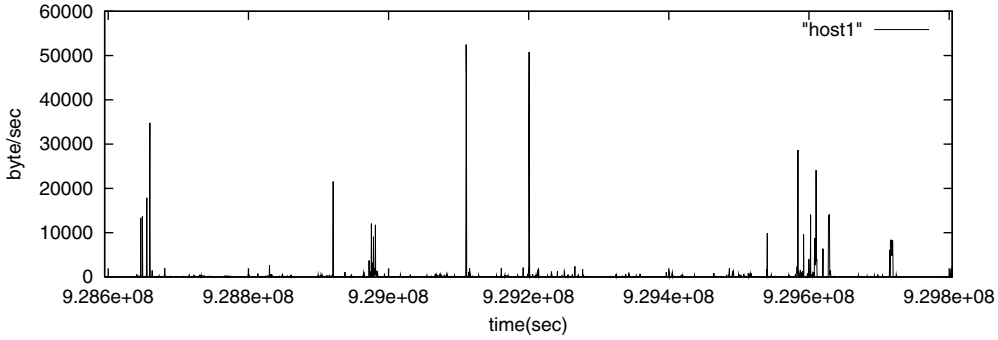

図 3.3　被験者 1 のトラフィック

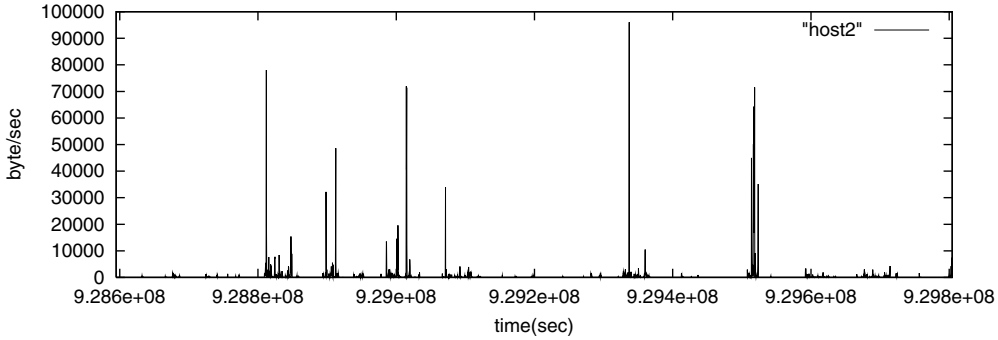

図 3.4　被験者 2 のトラフィック

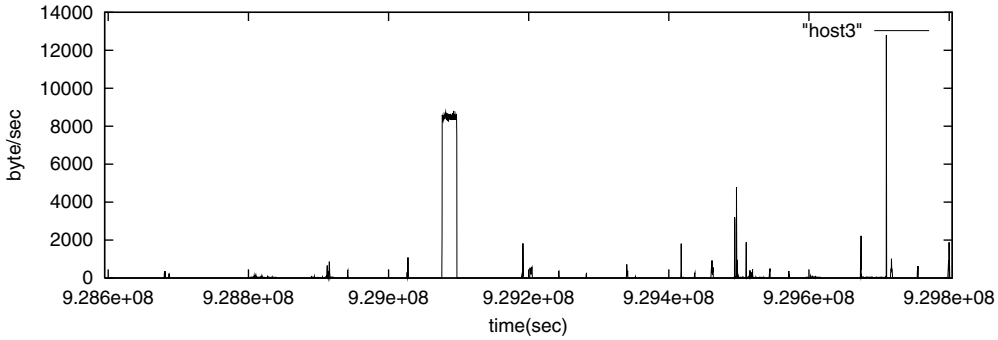

図 3.5　被験者 3 のトラフィック

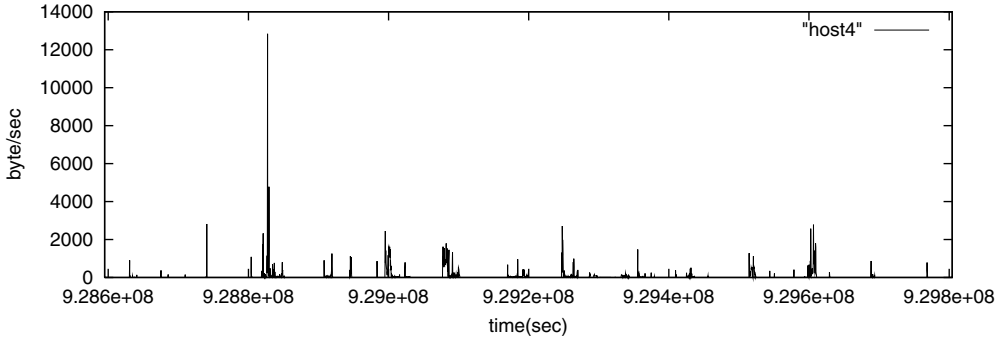

図 3.6　被験者 4 のトラフィック

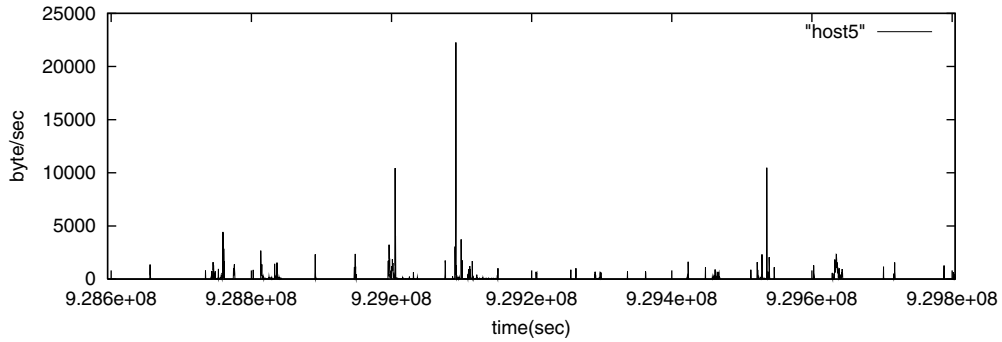

図 3.7 被験者 5 のトラフィック

クを大きく左右することが明らかになった。特に自動的に通信をはじめるようなアプリケーションは、ユーザが起動するアプリケーションに比べて頻繁に通信を行ない、可搬型計算機のトラフィックパターンを大きく変化させる傾向にある。

以上のことより、利用するアプリケーションとその使われ方を限定すれば、トラフィックパターンは大方予想が可能であり、PDA や組み込み系の計算機などではそれに合ったパラメータ設定が可能であることが分かった。

今回の解析では、おおまかな特徴を明らかにするため、ある程度手間の多いトラフィック収集と解析手法をとった。今後はこの解析を活かしたツールの作成が必要となる。また、今回はネットワークトラフィック収集ツールの制限によりトラフィックパターン収集 OS を BSD 系の UNIX に限定したが今後は Windows などの他の OS でのトラフィック収集解析ツールの開発も必要である。この 2 点を解決することにより、より多くの被験者を獲得することが可能となり、更に一般的な可搬型計算機のトラフィック解析を行なうことが可能となる。

第 4 章　おわりに

Rover BOF は本年度、月に 2 回の割合でミーティングを行い、移動計算機環境に関する議論を活発におこなってきた。今回は、数ある研究アイテムの中から縮退アドレスモデルに基づいた IPv6 における新しい移動透過製保証プロトコルの研究と移動計算機のトラフィック解析について報告をおこなった。

今後もこのアクティビティを保ちつつ移動計算機環境に関する研究を続けていく予定である。なお、Rover BOF のより詳細な活動については、`http://www.sfc.wide.ad.jp/rover/`を参照されたい。