

第 13 部

移動体通信環境

第 1 章

はじめに

携帯可能な計算機の性能の向上と、移動通信機器の普及に伴い、移動計算機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。WIDE Rover BOF は、このようなネットワークを利用した移動計算機環境について研究するグループである。我々が研究の対象とする移動計算機は主にノート型 PC 等の十分な処理能力を持つ計算機であるが、PDA 等の比較的能力の低い計算機も視野に入れた議論を行なっている。

研究のアイテムは基本的にグループのメンバが個々に持っているものであり、その対象は多岐にわたる。現在グループ内で主に議論されているアイテムには例えば以下のようなものが挙げられる。

- 移動計算機に適したファイルシステム
- 移動計算機のネットワーク環境の変化に対する適応方式
- 移動計算機が生成するのトラフィックの特性についての調査
- 新しい移動透過性保証のプロトコル
- 移動計算機を支援するためのミドルウェア/アプリケーション
- 地理的位置情報システム
- ad-hoc なネットワーク上での情報資源共有のためのプロトコル

本稿では、我々の最近の研究トピックから特に次の 2 点について報告を行なう。

- 移動計算機用ファイルシステム
移動計算機では、ネットワークの切断やネットワーク間の移動などの事象が発生するため、現在広く利用されている NFS 等のネットワークファイルシステムの使用は困難である。これを解決すべく、我々は移動計算機が直面する環境に適したファイルシステムについての研究と実装を行なっている。本稿では、我々のグループで議論されている移動計算機用のファイルシステムの一方式である、Personal File System(PFS)の概要と現在の研究課題について報告する。

- 移動計算機のトラフィック解析

現在のインターネットにおけるトラフィック解析は、通常バックボーンを対象とした非常にマクロなものである。しかし、移動計算機は特有のトラフィックパターンを持っていると考えられる。現在我々は移動計算機のネットワークトラフィックを収集し、そのパターンの特徴や解析方法についての議論を行なっている。今回は、現在までに収集された限定的なデータを用いての簡単な解析例を示し、これからの解析方法についての方針について報告する。

PFS は日常の使用に耐える実装が存在しており、メンバーが実際に使用しての実践的な研究が行われていることに加えて、移動計算機のネットワーク環境の変化に対する適応方式についての議論の中においても、アプリケーション例として考えられており、我々のグループにおいて現在中心となっている研究テーマの一つである。

移動計算機のトラフィック解析は、純粋に研究的な観点だけではなく、実際に移動計算機からネットワークを煩雑に使用しているユーザの視点からも見ることによって、今までは定量的に明らかにされなかった部分を明らかにしていき、今後の研究に役立てるという意味で重要であると我々は考えている。

第 2 章

Personal File System

2.1 背景

携帯型計算機を使う上で問題になる事の一つに、従来の固定型計算機環境とのファイル共有を挙げることができる。

携帯型計算機では通信路の状況が動的に変化するため、安定した通信路を期待している従来のファイルシステムでは、極端な応答遅延が生じたり、ファイルアクセスが止まってしまうなど、そのまま使うには問題が多い。

そこで、我々のグループでは、携帯型計算機に特化した共有ファイルシステムである PFS (Personal File System) [154, 147] の評価とその改良を行なっている。

2.2 PFS の概要

PFS は、固定されたファイルサーバ上のファイルを、携帯型計算機から共有する共有ファイルシステムである。

PFS ではクライアント側のローカルディスク上にキャッシュを持ち、アクセス頻度の高いファイル及びユーザが指定したファイルをここに読み込んでおき、ファイル操作はこのキャッシュを通して行なうことにより、サーバとの通信が断たれた状況下でもファイル操作を継続することができる。

PFS はファイルサーバ上で動作する Master Server (MS) と、クライアント上で動作する Cache Server (CS) とで構成される。(図 2.1)

CS はユーザレベルで動作する擬似 NFS サーバとしてクライアントホスト上で動作し、NFS 要求に対してローカルディスク上にあるキャッシュの内容を用いて応答する。そして、必要に応じて MS と通信を行ないキャッシュの内容を更新する。クライアントでは CS をローカルホスト上で動作する通常の NFS サーバとしてマウントすることにより、PFS を利用することができる。

MS はファイルサーバ上で動作し、CS からの要求に応じてファイルサーバ上のファイル操作を行なう。

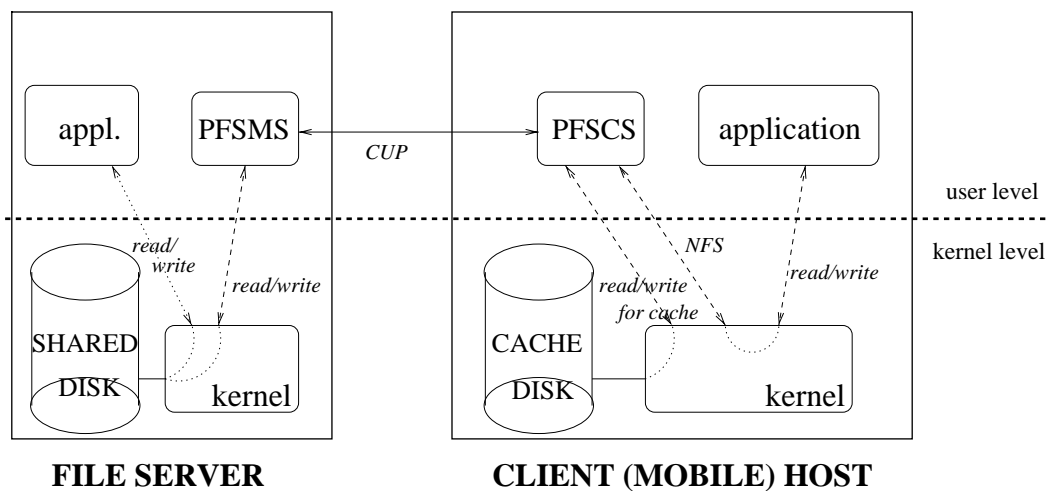


図 2.1: PFS の構成

CS と MS との間の通信プロトコルは Cache Update Protocol (CUP) と呼ばれる独自のものを利用している。CUP は TCP/IP の仮想回線上で動作し、キャッシュ更新時に更新前後のファイルのバージョンをサーバ側で確認する機能を持つほか、ファイル内容の転送と並行して他のファイル操作を許すなど、低速な通信路を使う際のファイルの一貫性保持を容易にしている。

キャッシュの更新に利用可能な通信路は携帯型計算機の移動によって変化する。したがって、一種類のキャッシュ更新アルゴリズムだけで処理するよりも、利用可能な通信路の状況に応じてアルゴリズムを変化させた方が、より効果的なキャッシュ更新を行なうことができる。そこで PFS では、複数のキャッシュ更新アルゴリズムを用意し、それを環境やユーザのポリシーに応じて動的に切り替える事により、様々な異なる環境に動的に適応する。

通信路が利用できない間は、キャッシュに読み込まれているファイルのみで動作しなければならない。したがって、作業に必要なファイルは必ずキャッシュに読み込まれる機構が必要となる。そこで PFS ではキャッシュすべきファイルのパス名のリストを内容とするファイルを指定することにより、そのファイルで指定されたファイルは定期的にキャッシュに読み込む機構を持つ。また、通常ファイル操作によって参照されたファイルもキャッシュに読み込まれるため、あらかじめ必要であると明示的に指定されたファイル群の他に、通信路が切断された時点で利用されていたファイル群もキャッシュに読み込まれた状態となる。

携帯型計算機やファイルサーバで用いられるオペレーティングシステムは様々であり、特に高度な安定性を要求されるファイルサーバのオペレーティングシステムの改変は望ましくない。そこで、既に述べたように PFS では全ての要素をユーザレベルプログラムで構成し、ファイルサーバ上でのファイル操作には標準入出力ライブラリを、クライアントでのファイルシステムへのマウントには業界標準の NFS プロトコルを採用している。これに

より、PFS はサーバ・クライアントの両端共でオペレーティングシステムの改変を必要とせず、高い移植性を持つ。

2.3 他のファイルシステムとの比較

NFS NFS [90] は Sun Microsystems Inc. が、異なる計算機間でのファイル共有を主な目的として、設計した共有ファイルシステムである。NFS はプロトコルが単純でかつ仕様が公開されていることもあり、非常に広く普及している。

NFS では原則としてサーバは状態を持たず、一つの要求に対して完全に処理が終了した後に応答を返す。この為、耐故障性に優れている。

近年普及を始めた NFS Version 3 プロトコル [28] では、処理効率を上げるためのプロトコルの改良が行なわれており、低速な回線での応答性の悪化を抑えるようなクライアントの実装が可能となっている。

しかしながら、通信路が切断された状態では全く利用できなくなることが、通信路が利用できない状態でも運用される携帯型計算機で利用する上の問題となる。

Coda Coda [92] は携帯型計算機での利用を考慮して設計された共有ファイルシステムであり、クライアントのローカルディスク上に比較的大容量のキャッシュを持ち、かつファイルサーバ側でファイルの変更があった場合に、それをキャッシュしているクライアントに通知するコールバックの機能を持つため、低速な通信路を用いても高い信頼性と応答性を確保している。

また、通信路が切断されると切断状態に遷移し、サーバとの通信を行わずにキャッシュのみでファイル操作を継続することができる。通信路が再度利用可能になると、切断状態の間のファイル操作の結果をサーバに反映させた後に、サーバとの通信を行なう通常の接続状態に遷移する。

この機能により、Coda では通信路の状態に関らずにファイル操作を継続することができる。

しかし、Coda を利用するにはオペレーティングシステムのカーネルに変更が必要であり、特にサーバ側において導入の際の問題になる。

rsync rsync[123] はファイルシステムではなく、ネットワークを通じたファイルの複製を行なうコマンドであるが、オプションを指定することによって、再帰的にファイルの更新時間を比較して新しいファイルのみを複製する事ができる。ファイルを共有すべき計算機間で相互に rsync による複製をおこなう事により、特定ディレクトリ以下のファイルの同期を行なうことが可能となる。

rsync は SSH (Secure SHell) を用いた認証と通信の暗号化を容易に行なうことができるうえ、一部分が更新されたファイルの転送を行なう際には、ファイルの差分情報を生成して転送することにより、転送時のトラフィックを削減することができる。

しかし、rsync は単純にファイルの時間を比較して古いファイルを新しいファイルで上書きしてしまうため、ファイルを共有している両端でファイルを変更した場合には片方での変更が消えてしまううえ、計算機間で時刻情報がずれていると予期せぬ結果が生じる原因となる。また、ファイルの名前の変更や消去は反映できず、両端で明示的に消去しない限りは、複製によって再度作成されてしまうという問題がある。

2.4 現状

PFS は現在標準的な C 言語と、rpcgen コマンドでコンパイルできる RPC 言語でプロトタイプが実装されている。全てをユーザレベルプログラムとして実装し、かつ標準的な NFS プロトコルを利用しているために、移植性は非常に高く、これまでに FreeBSD 2.2.8-RELEASE、BSD/OS 3.1, RedHat Linux 5.2, Solaris7 で動作することを確認している。

また、次世代 IP である IP version 6 にも対応済みである。

2.5 現在の問題点

現在の PFS では、以下の要因によってキャッシュ更新モードの遷移を行なっている (図 2.2):

- TCP 接続の切断
- TCP 接続の成功
- NFS リクエストに対して一定時間内に応答できない
- ファイル転送のスループットが一定速度以上
- サーバへのリクエストに対する応答時間が一定以上
- サーバへのリクエストに対する応答時間が一定以下
- ユーザによる明示的な指示

上記のように、基本的に PFS では自らの通信性能を測定してモード遷移を行なっている。しかしながら、実際の携帯型計算機における運用においては、通信路の負荷状況によって測定結果が大きく変動し、複数のモードの間を交互に遷移する現象が観測された。

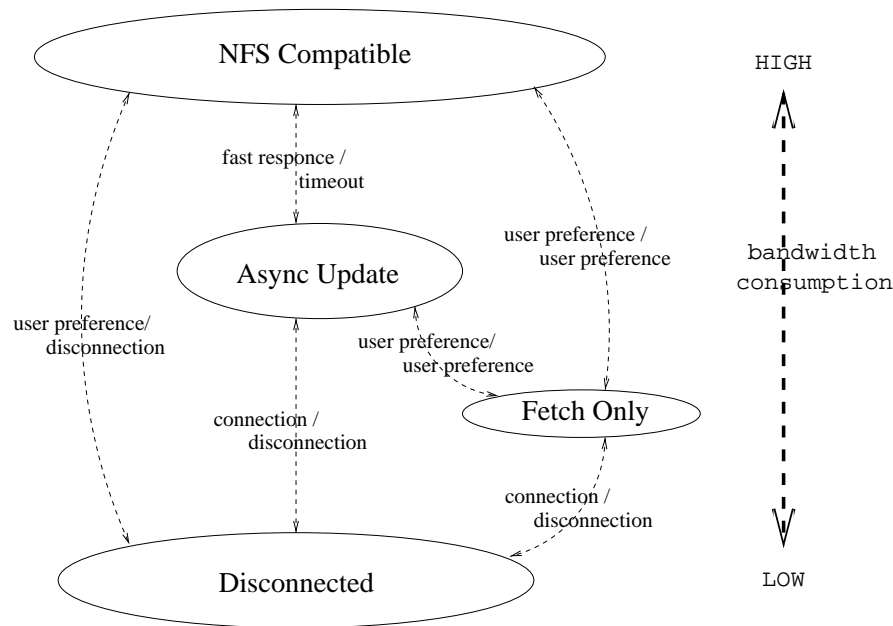


図 2.2: モード間の遷移

この原因の一つとして、オペレーティングシステムに対する PFS からの送信要求が、ネットワークの各階層でバッファリングされるため、実際の通信路の遅延よりも大きな遅延が検出されてしまうという問題がある。

また、PC カードの活線挿抜などによる予期しないネットワーク切断を、通信が一定時間途絶したことによってしか検知できないため、PFS が回線の切断を検知するまでの間は、ユーザの操作がタイムアウトに必要な時間だけブロックされてしまう。さらに移動により別ネットワークに接続した場合でも、移動前のネットワークにおける通信が切断されない限りは新規の通信を開始できず、やはりユーザ操作に対するブロックが発生する。

これらの問題は、通信路の状況をオペレーティングシステムが隠蔽してしまい、アプリケーションからは正しく取得できないために生じている。

2.6 通信情報取得システム

前節で述べたように、現在はオペレーティングシステムが通信路の状況を隠蔽してしまい、アプリケーションが状況に動的に適應する障害となっている。

そこで、より正確な通信路の状況をオペレーティングシステム内部から取得し、それを抽象化してアプリケーションに提供する機構を検討し、設計している。この機構を利用することにより、アプリケーションが通信環境に動的に適應することが容易となる。

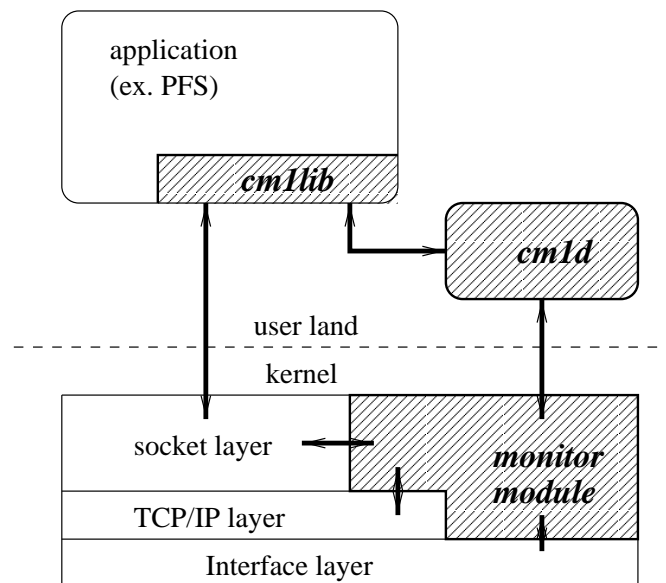


図 2.3: CM1 の構成

現在設計中の通信情報取得システム Communication Monitor 1 (CM1) は、図 2.3 に示すように、kernel module, cmld, cmllib の 3 つの要素から構成される。

kernel module はオペレーティングシステムのカーネルから、通信路の状態に関する情報を収集する部分である。情報はネットワークインターフェイス、TCP/IP モジュール、ソケットインターフェイスモジュールなどから収集され、新規に設計されたソケットによりユーザレベルの cmld に通知される。

cmld はユーザレベルで動作するデーモンプログラムであり、kernel module から収集した情報を各アプリケーションが必要とする形に一般化し、アプリケーションにリンクされた cmllib に通知する。

cmllib はアプリケーションにリンクするライブラリであり、アプリケーションから cm1 を使うための手続きが含まれる。アプリケーションは cmllib を通じて cm1 システムに自らの適応に必要な情報を要求し、その応答を環境への適応に利用する。

cmld 及び cmllib ではアプリケーションに対して、オペレーティングシステムから得られた生の情報だけでなく、通信デバイスの情報を抽象化し、一般化した情報と、さらにユーザやアプリケーションのポリシーを適用した結果としての情報も与える。(図 2.4)

このようにすることにより、アプリケーション作成時には想定しなかった環境にもある程度適応することが可能となる。

この機構を用いることにより、PFS ではよりの確なモード遷移を行なうことができると考えている。また、携帯型計算機上で用いるアプリケーションの多くは、PFS 同様に通信環境に動的に適応する必要があると考えられる。通信情報取得システムを用いることによ

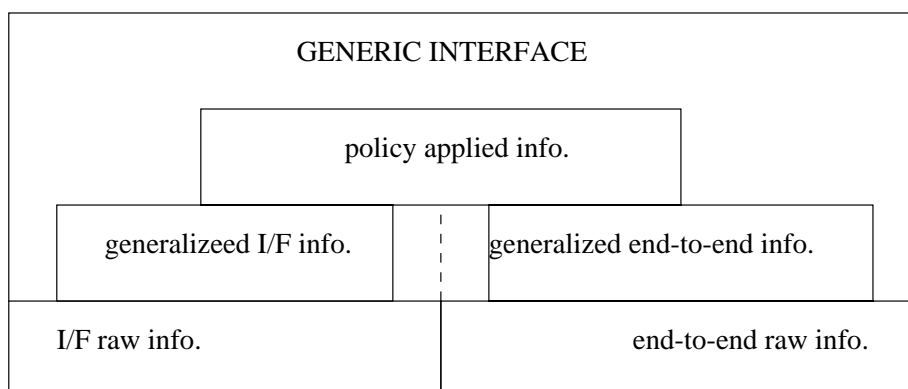


図 2.4: CM1 が扱う情報

り、このような新たなアプリケーションの作成時にも有効に利用できると予想できる。

2.7 今後の予定

現在は現在検討中の通信環境情報取得システムのプロトタイプを実装し、PFS に適用してその有効性の確認作業を行なっている。今後は作業の結果を元に、他のアプリケーションでも利用できる通信環境情報取得システムを構築する予定である。

また、PFS そのものに関しても以下のような検討課題があり、改良を行なっていく予定である。

使用通信バンド幅の制限

標準的な TCP/IP の実装では各アプリケーションの通信に対して公平にバンド幅を割り当てるが、一部の实装では送信時に優先度をつけることにより、アプリケーション毎に使用通信バンド幅を制限することが可能である。そこで、PFS にこのバンド幅の制限を制御する機能を付加することにより、PFS の通信が他のアプリケーションの実行を妨げる事を防ぐことを可能とする。

より最適なキャッシュ更新アルゴリズムの設計

現在の各キャッシュ更新アルゴリズム及びそれらへの遷移アルゴリズムは、必ずしも最善のものとはいえない。そこで、実際に広く運用した結果を用いて、より良いキャッシュ更新アルゴリズムとそれらへの遷移アルゴリズムを設計、実装する。

UNIX 以外のプラットフォームへの移植

現在、PFS は数多くの UNIX 上で動作しているが、原理的には NFS が利用可能な他のオペレーティングシステムでも利用可能である。そこで、JAVA 等のより移植性

の高い言語を用いることにより、他のプラットフォームにおいても PFS を利用可能とする。

第 3 章

移動計算機のトラフィック解析

一般的に、移動計算機は固定計算機と異なったトラフィックパターンを持っていると言われている。しかし、実際にこれを検証した事例は存在しない。そこで我々は、実際に利用されている移動計算機のトラフィックをモニタリングし、上記の仮定が正しいことを検証することにした。

3.1 移動計算機トラフィックモニタリングの手法

通常、バックボーンなどのトラフィックモニタリングは、重要な部分にタップ用の計算機を差し込むことにより実現されている。タップした計算機では、主に、以下の 2 種類の方法によって実際にデータを収集している。

- NNstat による解析
- tcpdump によるキャプチャリング

前者の手法は予め知りたい情報がわかっている場合に、後者は後で様々な解析を行なう場合に取りられる手法である。上記に挙げた手法の他にも SNMP を利用した方法などがある。

現在、我々は、測定すべき対象を明確にできていない。これは、移動計算機のどの部分が特徴的なものが明確にされていない為である。このため、必然的に移動計算機が通信に利用した生のパケットを蓄積し、後で解析する手法を取らざるを得ないことになる。

そこで我々は、tcpdump を利用し、データを蓄積することから始めた。ノート型計算機では、通常、PC-card 型のネットワークデバイスを利用する。そこで、PC-card を管理しているデーモンにフックをかけることにより、ネットワークが利用可能になると自動的に tcpdump を始めるスクリプトを用意し、利用した。現在、6 名の被験者のノート PC に本機構が組み込まれている。

現在、約 3 ヶ月分のデータが蓄積されている。本報告書ではこれらのデータをもとに解析を行なった結果を示す。

3.2 トラフィック解析

rover ではこれまで蓄積したデータとともに幾つかの解析を行なった。本稿ではそれらの解析のうち、各ホストのインターフェイス毎のセッション長 (秒) とセッション毎の通信データ量 (byte) に関してまとめたものを示す。

ここで挙げられているホストは移動計算機のヘビーユーザのものである。日常的に移動計算機を利用して作業を行なっている。また、Mail は、ホスト 4 のユーザを除いて、POP を利用している。

3.2.1 ホスト 1

ホスト 1 は 3 つのインターフェイスを必要に応じて使い分けている。それぞれのインターフェイスの利用形態は以下のようなものである。

無線 LAN

オフィスにいる時に定常的に利用している。Mail、Web、ssh など通常の作業はほぼこのインターフェイスを通じて行なわれる。

Ethernet

自宅でダイアルアップルータを介して利用することがほとんどである。また、テレホーダイ時間の利用が主である。

PHS

会議などで外出した時に、ほぼ Mail を読む為だけに利用している。外出時は 1 時間間隔で利用、または、外出一回につき一度だけ利用する。

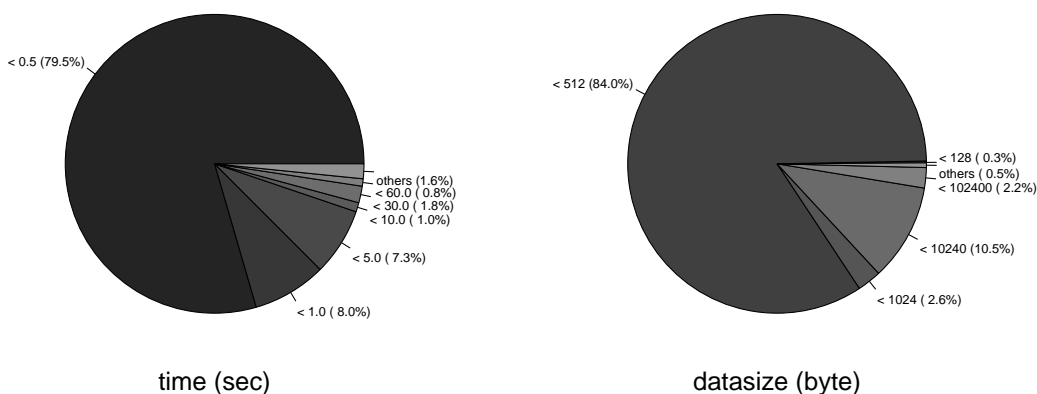


図 3.1: ホスト 1(無線 LAN) のセッションの分布

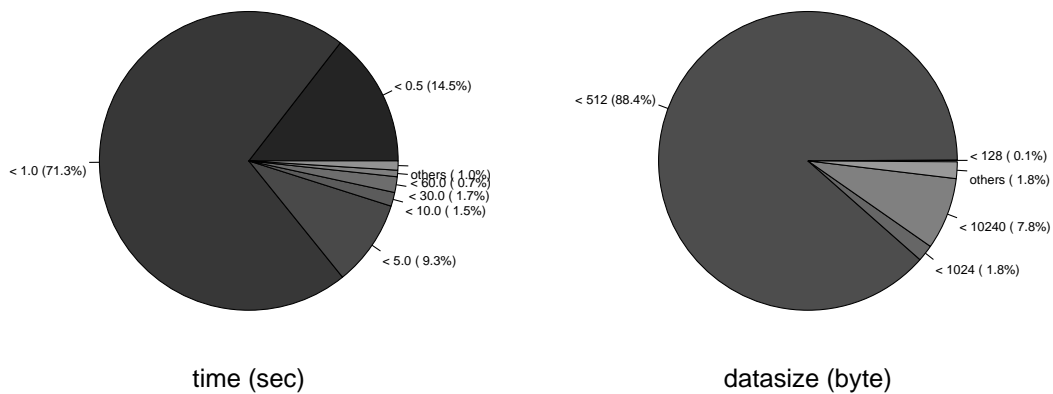


図 3.2: ホスト 1(Ethernet) のセッションの分布

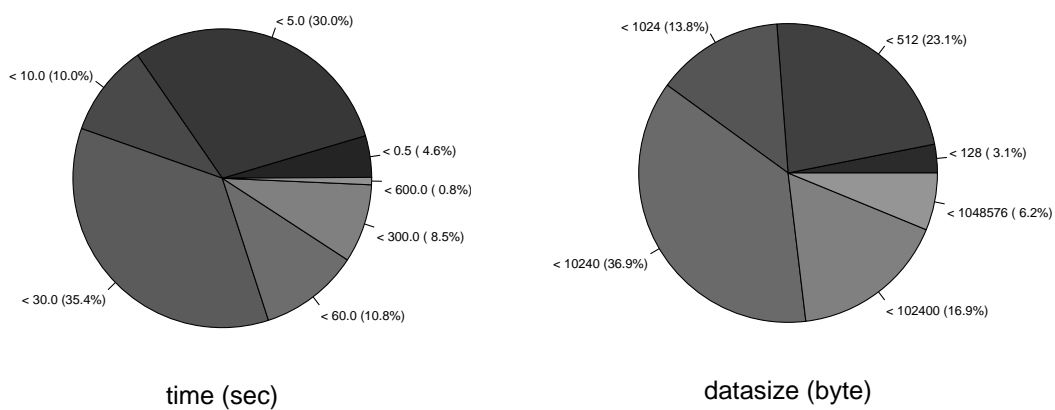


図 3.3: ホスト 1(PHS) のセッションの分布

3.2.2 ホスト 2

ホスト 2 は 3 つのインターフェイスを必要に応じて使い分けている。それぞれのインターフェイスの利用形態は以下のようなものである。

Ethernet

定常的に利用しているインターフェイス。Mail、Web、ssh などの他に NFS などでも利用している。ほぼデスクトップと同じ利用形態である。

無線 LAN

会議室などに移動した時、機器のメンテナンスの為に別室に移動した時などに主に利用する。利用するアプリケーションは Mail および Web。

モデム

自宅からの利用。Mail、Web、ssh が主な利用アプリケーション。

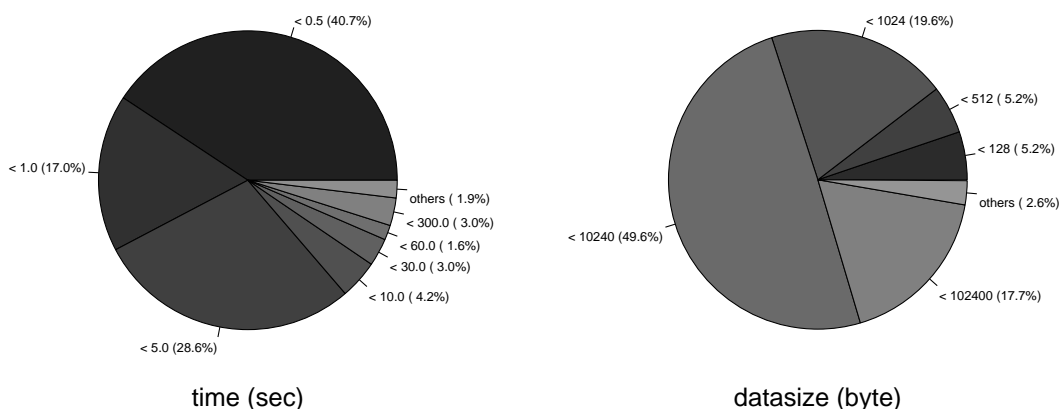


図 3.4: ホスト 2(Ethernet) のセッションの分布

3.2.3 ホスト 3

ホスト 3 は 2 つのインターフェイスを必要に応じて使い分けている。それぞれのインターフェイスの利用形態は以下のようなものである。

Ethernet

主にオフィスのネットワークに接続し、固定ホストからログインして利用している。端末エミュレータやエディタを固定ホストに表示する他、頻繁に Mail の送受信や Web の参照を行っている。

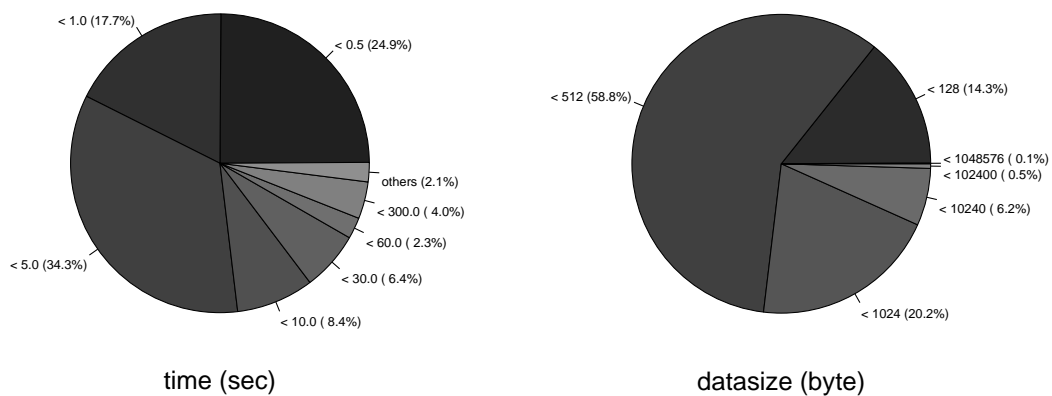


図 3.5: ホスト 2(無線 LAN) のセッションの分布

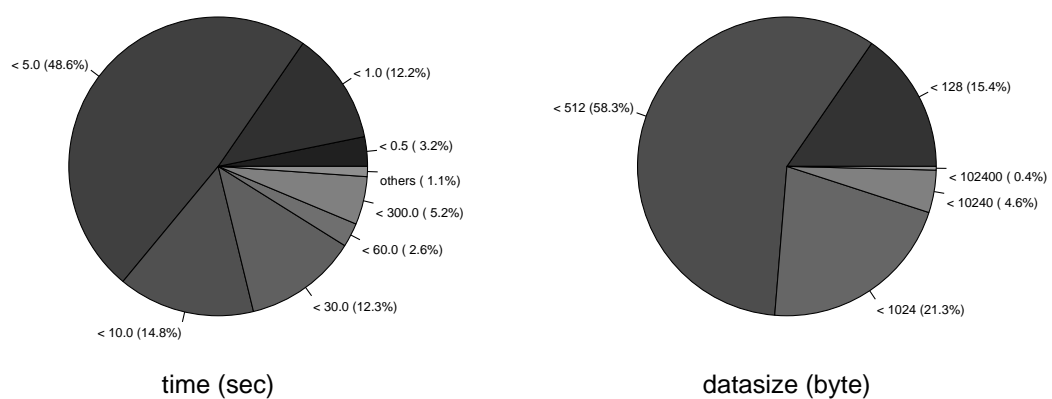


図 3.6: ホスト 2(モデム) のセッションの分布

PHS or モデム

自宅で、もしくは外出先への移動中に、主に Mail の送受信の為に利用している。また、必要に応じて Web の参照や、オフィスのホストへのログインを行っている。移動中は 1 時間程度の間隔で利用する。

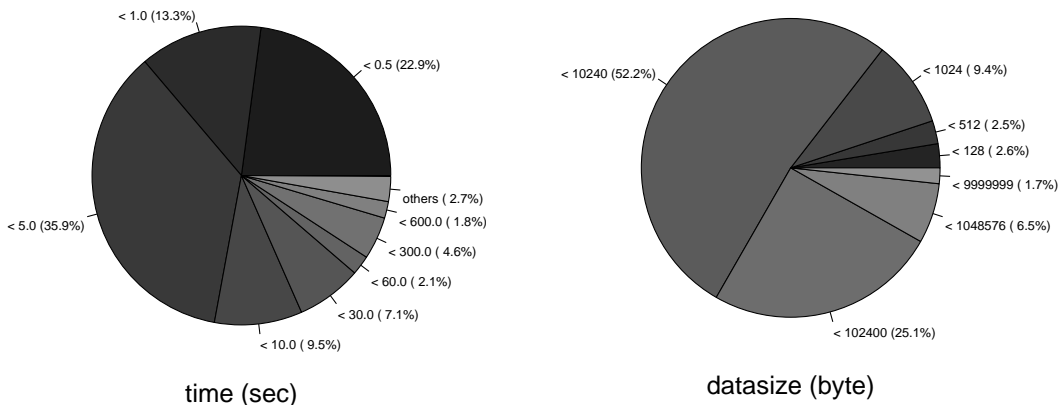


図 3.7: ホスト 3(Ethernet) のセッションの分布

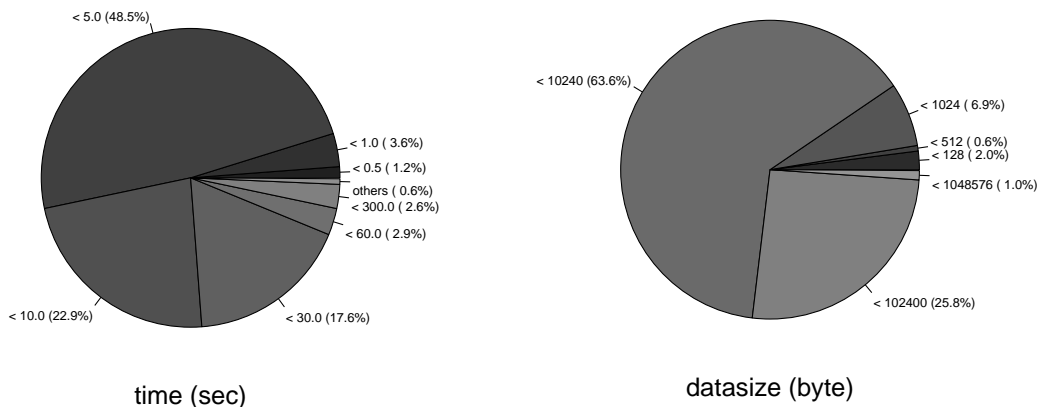


図 3.8: ホスト 3(モデム or PHS) のセッションの分布

3.2.4 ホスト 4

ホスト 4 は 2 つのインターフェイスを必要に応じて使い分けている。それぞれのインターフェイスの利用形態は以下のようなものである。

Ethernet

オフィスでのメイン端末として利用。全ての作業を ssh によりリモートで行なった。また、X の Window をノートに表示することもあった。

モデム

PPP のみで、自宅からの接続に利用する。ssh によりメールを読む。たまに scp でファイルの転送や Web の利用も行なう。33.6k で通信を行なっている。

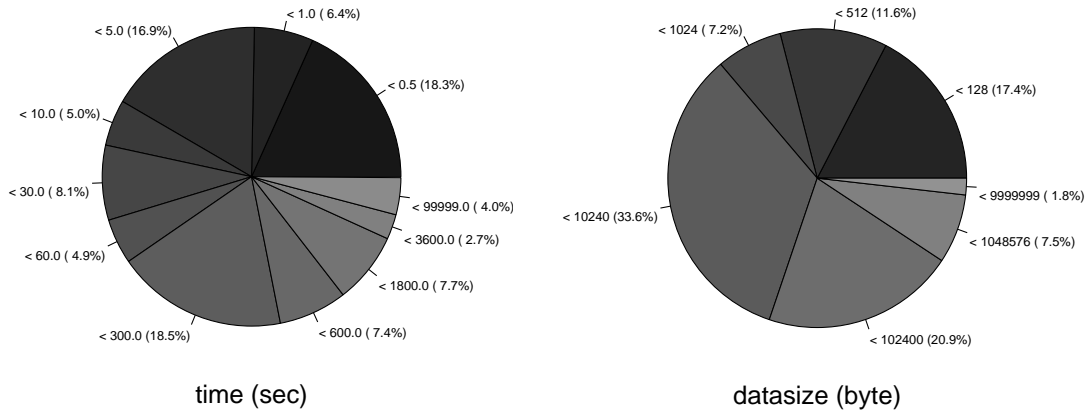


図 3.9: ホスト 4(Ethernet) のセッションの分布

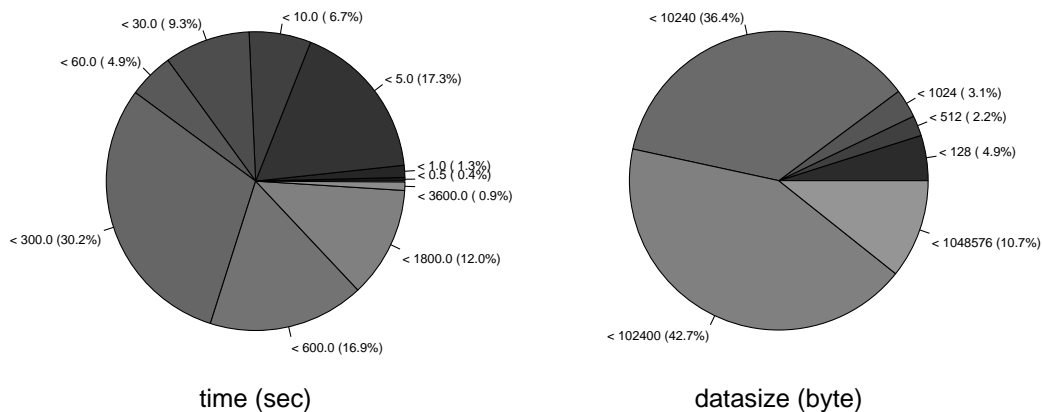


図 3.10: ホスト 4(モデム) のセッションの分布

3.2.5 ホスト 5

ホスト 5 は 1 つのインターフェイス利用している。そのインターフェイスの利用形態は以下のようなものである。

Ethernet

オフィス、自宅の両方で利用している。自宅では、ダイヤルアップルータを利用し、ISDN によって接続している。オフィス、自宅共に別に固定ホストが用意されている為、Mail 以外の用途にはほとんど利用しない。

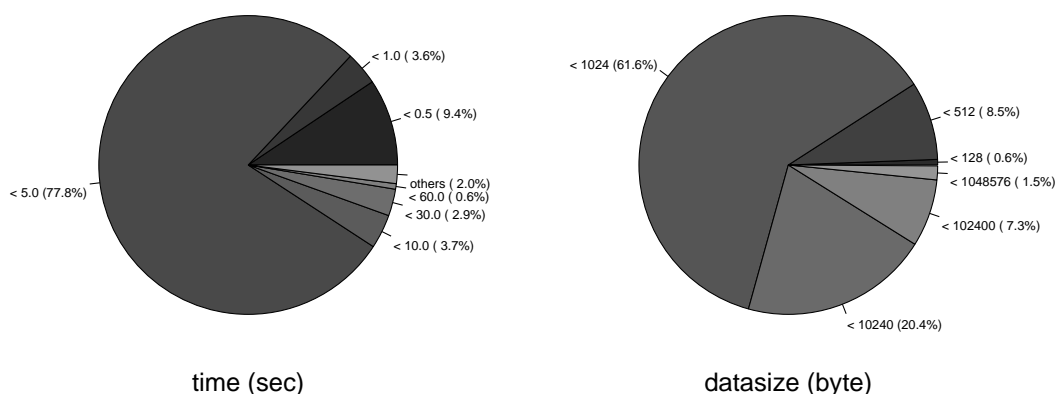


図 3.11: ホスト 5(Ethernet) のセッションの分布

3.2.6 ホスト 6

ホスト 6 は 3 つのインターフェイスを必要に応じて使い分けている。それぞれのインターフェイスの利用形態は以下のようなものである。

Ethernet

主にオフィスおよび外出先のネットワークに接続して使用している。電子メールの送受信や Web へのアクセスなどといった通常のネットワーク利用は、そのほとんどがこのインターフェイスを介して行われている。また、このインターフェイスが使用可能な状況では、ノート PC 上のファイルシステムバックアップ作業も行っている。

無線 LAN

基本的には Ethernet と同様、オフィスや外出先でのネットワーク接続時に利用している。建物内を頻繁に移動する機会が多い時には、もっぱらこちらをよく利用している。

PHS

自宅からダイヤルアップ接続するとき使用する。テレホーダイの時間帯を除けば、メールの送受信以外の利用はほとんど行わない。

3.3 まとめ

現在、rover での移動計算機のトラフィック解析は進行中である。残念ながら、現段階では前節で述べた程度の解析しか行なわれていない。今後、移動計算機のトラフィックの特徴を掴むとともに様々な解析を行なっていく予定である。また、解析ツールの開発、解析手法の検討なども同時に議論していくことになっている。以下にこれからの rover としての仕事をまとめる。

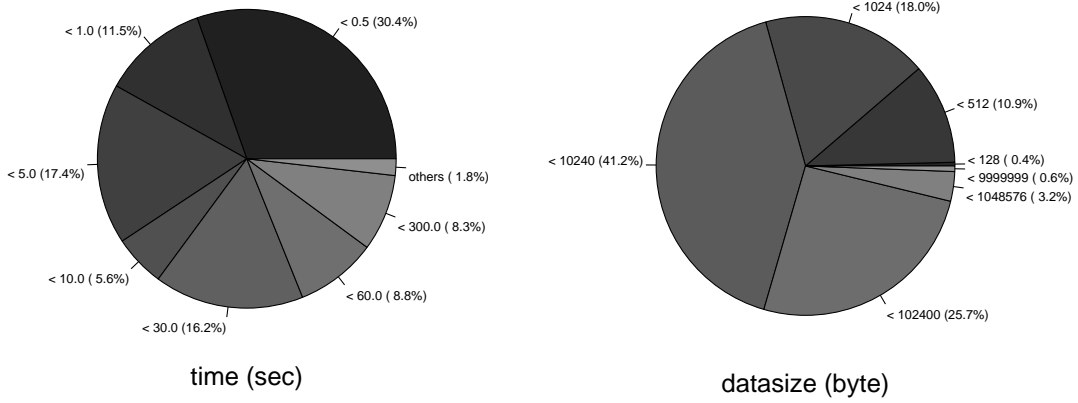


図 3.12: ホスト 6(Ethernet) のセッションの分布

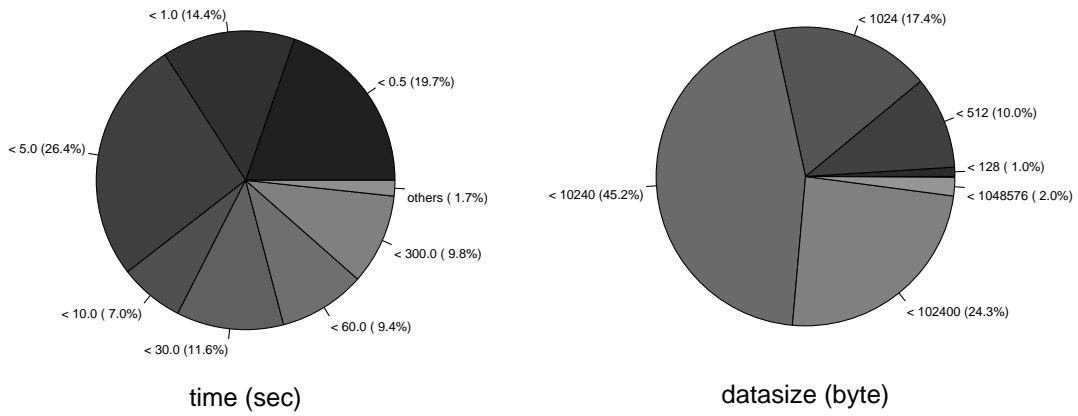


図 3.13: ホスト 6(無線 LAN) のセッションの分布

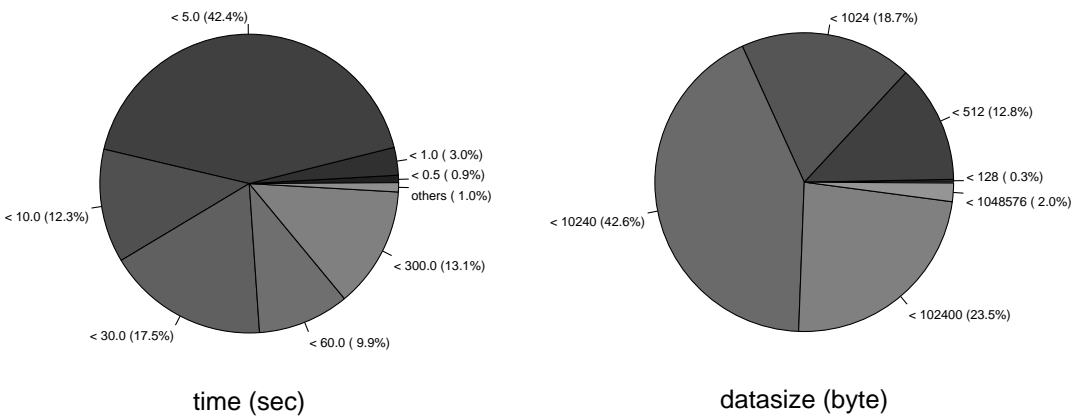


図 3.14: ホスト 6(モデム) のセッションの分布

- 解析項目の検討
- 解析の手法
- 解析ツールの開発
- 移動計算機環境への反映
- 十分なサンプル数 (ホスト数) の確保

第 4 章

おわりに

Rover BOF はおよそ 1 年間の間、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行ってきた。今回は、数ある研究アイテムの中から移動計算機用のファイルシステムである PFS と、移動計算機のトラフィック解析について報告を行った。

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。

