Title: 2003年 WIDE合宿における
仮設ネットワークの設計と構築

Author(s):
宮地利幸 (toshi-m@jaist.ac.jp)
遠藤正仁 (masaxmasa@tahi.org)
村本衛一 (muramoto.eiichi@jp.panasonic.com)
川喜田佑介 (kwkt@sfc.wide.ad.jp)
斉藤賢爾 (ks91@sfc.wide.ad.jp)
島慶一 (keiichi@iij.ad.jp)
高宮紀明 (takamiya@po.ntts.co.jp)
土井一夫 (kazuo@cysols.com)
石山政浩 (masahiro@isl.rdc.toshiba.co.jp)
田坂和之 (kazuyu-t@is.aist-nara.ac.jp)
成瀬大亮 (don@sfc.wide.ad.jp)
廣瀬峻 (tuffy@sfc.wide.ad.jp)
衛藤将史 (masash-e@is.aist-nara.ac.jp)
大江将史 (masa@fumi.org)
中尾嘉宏 (yoshih-n@is.aist-nara.ac.jp)

Date: 平成16年7月30日

第1章　2003 年春合宿ネットワーク

本章では、2003 年 3 月 3 日 (月) から 6 日 (木) まで滋賀県長浜ロイヤルホテルにおいて開催された WIDE プロジェクト春合宿 (以降、本合宿) におけるネットワーク構成および、そのネットワーク上で行われた実験の内容とその結果を報告する。

1.1　ネットワーク構成

図 1.1 に本合宿のネットワークを示す。

図中、点線より上部が奈良先端科学技術大学院大学 (以下 NAIST)、下部が合宿地である。また中央の atmis は京都大学に設置されている。四角はルータまたはホスト (サーバ) を表し、実線はイーサネット、専用線または無線 LAN を表す。二重鎖線は衛星回線を表す。

合宿地と NAIST は地上線として ATM により接続されている。この回線は、まず合宿地に地理的に近い京都大学に 3Mbps の ATM メガリンクを利用して接続されており、京都大学から NAIST には JGN を利用して接続されている。また、合宿地と NAIST は衛星回線 (上り下りとも 768kbps) により接続されている。図左側の 128kbps の INS は設営時や障害時用のバックアップ回線である。合宿ネットワーク外部への、IPv4 による 80 番ポートへのトラフィックは衛星回線に流し、地上線にはそれ以外のトラフィックを流した。

以下の時間帯は太陽雑音のため衛星回線が利用不能になることが予想されていた。このため、衛星回線を通る IPv4 の 80 番ポートへのトラフィックはトランスペアレントプロキシを利用して衛星回線に向け、以下の時間帯については、トランスペアレントプロキシを停止し、地上線を通るようにした。

- 2003/03/03 11:00 - 11:07

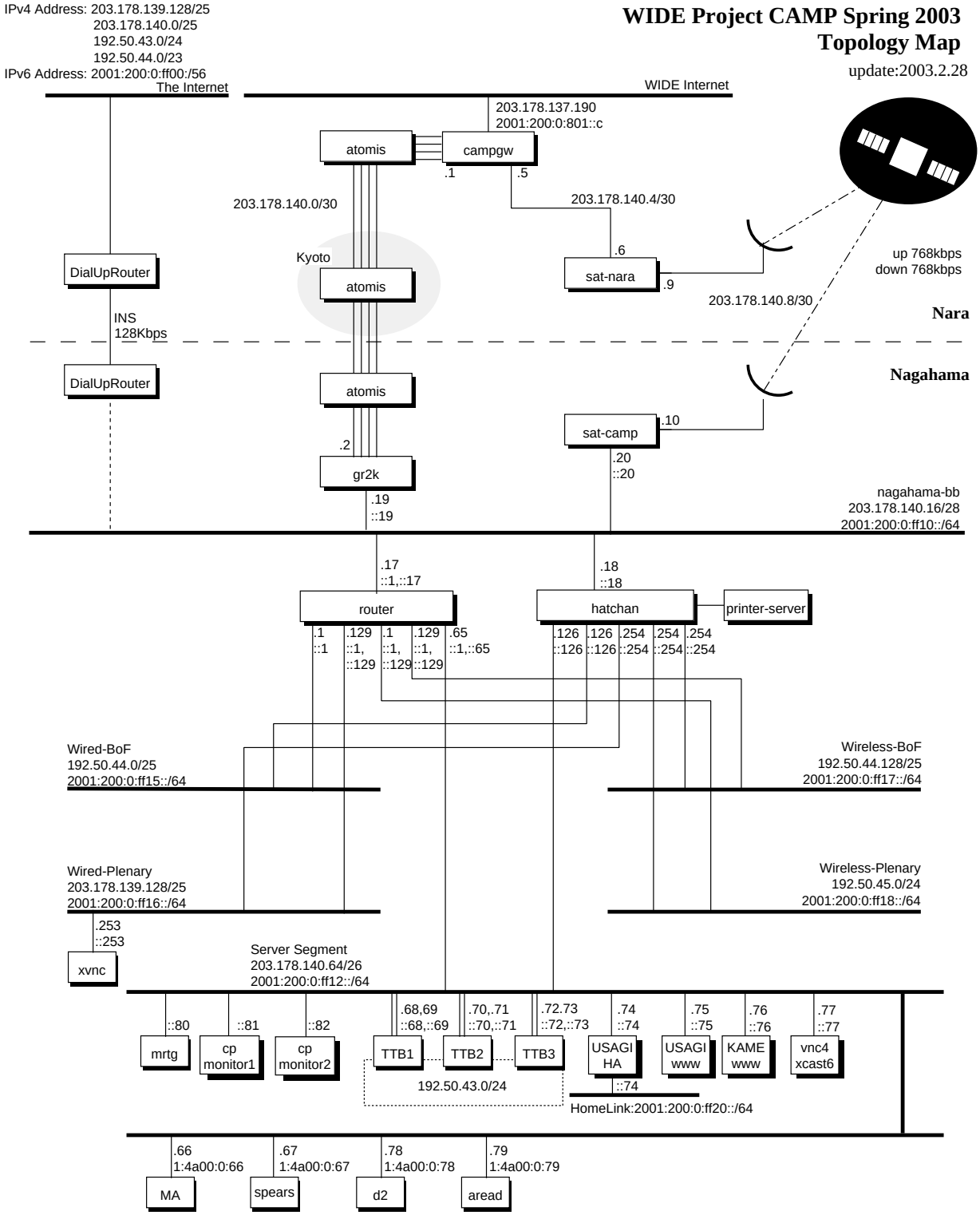

図 1.1: 本合宿のネットワーク構成

- 2003/03/04 10:58 - 11:09
- 2003/03/05 10:57 - 11:10
- 2003/03/06 10:57 - 11:10

合宿会場では有線の LAN としてプレナリ会場と BoF にそれぞれ Wired-Plenary と Wired-BoF を、無線LAN として同様に Wireless-Plenary と Wireless-BoF を提供した。各サブネットはルータ (router) および、様々なサービスを提供しているサーバである hatchan に接続した。

1.2 HotStage

本合宿は、WIDE 合宿初の関西での開催であったため、合宿地でネットワークを設営する前に、別の場所にネットワークを設営しその動作検証をおこなう HotStage にも工夫をした。

従来、関東で合宿を行う際には、慶応義塾大学 湘南藤沢キャンパス (以下 SFC) で HotStage を行うことが多かった。これは、以下のような要因による。

- WIDE 合宿の参加者および実験を行うグループが関東に多い
- HotStage 会場から合宿地へ機材を移動する必要があるため、移動距離が短い関東が良い
- WIDE バックボーンへ接続するため WIDE NOC に近い
- HotStage を設営するだけのスペースがある
- HotStage 会場近くに宿泊施設がある

しかし今回の合宿地が滋賀県であるため、SFC で HotStage を行った場合、機材の輸送や、実験を行うグループは合宿地から遠い HotStage 会場に一度集まらなくてはいけないなどという問題が生じた。

そこで、本合宿 HotStage では HotStage 会場を SFC と NAIST の 2 つに分割した。NAIST 会場にネットワーク全体を構築し、SFC 会場では VLAN を用いてユーザセグメントへの接続を提供した。L2 で接続したことにより、

両会場を同じセグメントで接続することができ、合宿地に近似した環境を構築することができた。これにより、ユーザセグメントのみを利用する実験参加者は、NAIST・SFC を選択して参加することが可能になった。また、両会場を Polycom を利用して常時接続したため、参加者同士のコミュニケーションも円滑に行なうことができた。

HotStage のネットワーク構成を 1.2 に示す。四角はルータおよびスイッチを示し、実線は Ethernet および VLAN を利用した仮想的なセグメントを、2 重線は ATM を表す。また四角の中の数字はルータおよびスイッチのポート番号を、VLAN ID で始まる文字列は VLAN ID を、[PV/VC] は ATM の PV と VC を表している。

VLAN により SFC と NAIST 間を接続するため TWO WG に協力していただいた。

1.3　合宿ネットワークを利用した実験

本合宿では以下の 6 つの実験が行われた。

1. IPv6/IPv4 トランスレータ「TTB」による IPv4→IPv6 トランスレーション実験と、Mobile IPv6 との連携実験
2. VNC for XCAST6 の評価実験
3. SPEARS(ICARS)+LIN6+MIPv6
4. PGP を応用した自由通貨システムの評価実験
5. Mobile IPv6 実環境運用実験
6. パッシブモニタ (CpMonitor) によるトラフィック情報可視化実験

1.3.1　IPv6/IPv4 トランスレータ「TTB」による IPv4→IPv6 トランスレーション実験と，Mobile IPv6 との連携実験

1.3.2　実験の目的

TTB シリーズは，従来の IPv6→IPv4 のトランスレート機能に加えて，新機能として DNS Proxy と連携した IPv4→IPv6 トランスレート機能と，

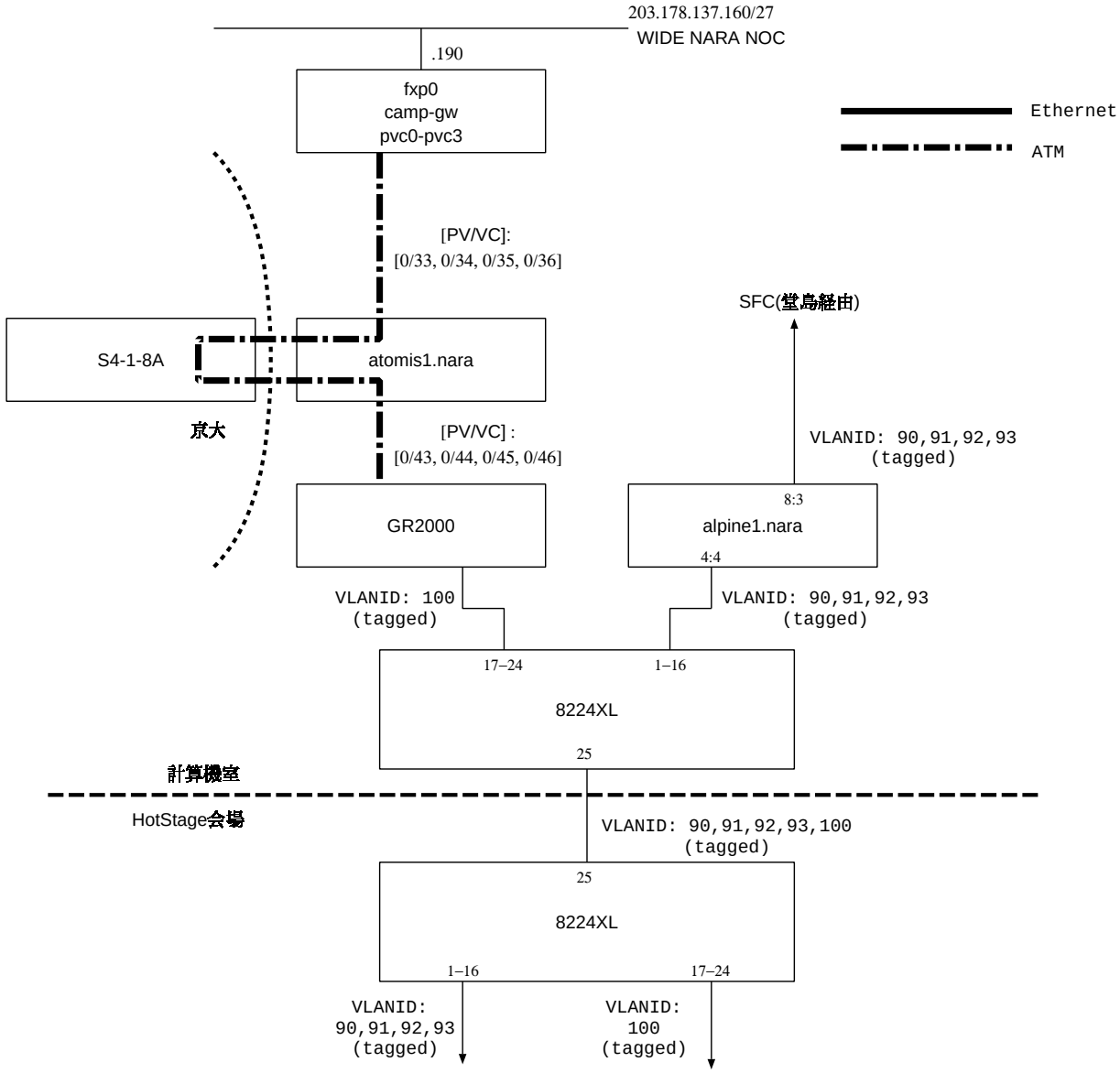

図 1.2: HotStage のネットワーク構成

Mobile IPv6 コレスポンデント機能 (ルートオプティマイゼーション未実装) を搭載した．本実験では，DNS Proxy と TTB が連携することによって，IPv4 クライアントから，IPv6 サーバにアクセスできることを検証し，またモバイルノードから IPv6→IPv4 のトランスレーションができることを検証する．

1.3.3 実験の概要

本実験では、3台の TTB を連続稼動させ，安定したトランスレーション環境を提供する．トランスレータ機能 TTB と，DNS Proxy 機能 TTB をそれぞれ，Server セグメントに設置する．IPv4 クライアントが IPv6 サーバに通信を行なうとき，DNS Proxy 機能搭載 TTB に名前を問い合わせることによって，IPv6 サーバのアドレスと，DNS Proxy 機能搭載 TTB 内にある IPv4 アドレスプールから任意のものを対応させ，IPv4 クライアントにそのアドレスを返す．こうして得られた IPv4 アドレスへの通信をトランスレータ機能搭載の TTB が IPv6 サーバへの通信として変換を行なう．また，TTB はコレスポンデントノードとして動作することができ，モバイルノードから IPv6→IPv4 トランスレーションを使用する場合は，DNS Proxy 機能搭載 TTB に名前を問い合わせることによって可能になる．ただし，経路最適化のサポートはしていないため，モバイルノードはホームエージェントを経由して，TTB を利用することになる．

1.3.4 実験環境

本実験では，トランスレータ機能搭載 TTB を二台と，DNS Proxy 機能搭載 TTB 一台をそれぞれ，サーバセグメントに設置する．DNS Proxy 機能搭載 TTB は，トランスレータ機能搭載 TTB 二台を1対1の割合で負荷を分散させている．また IPv4 アドレスプールとして，192.50.43.0/24 が割り当てられ，その中から，192.50.43.0/25 を ttb2 に割り当て，192.50.43.128/25 を ttb3 に割り当てた．

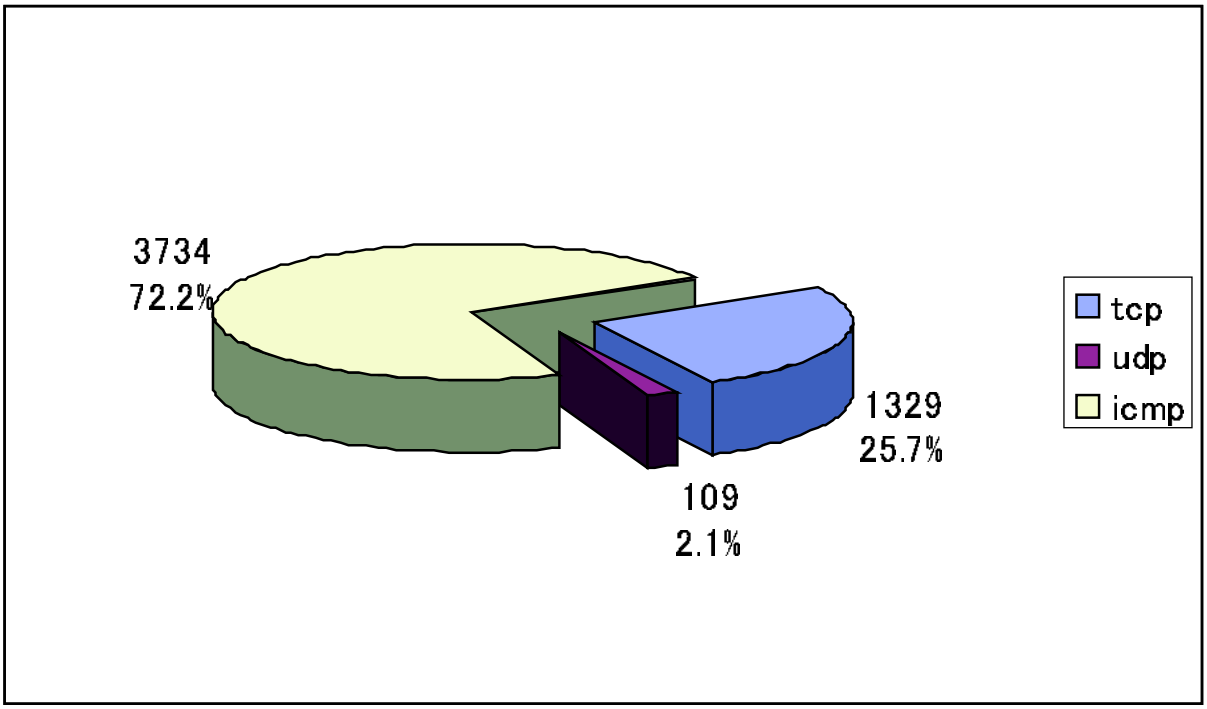

図 1.3: プロトコル別変換比率

1.3.5 結果

合宿期間中，連続稼動運転を行った．その結果 TTB トランスレータの IPv6→IPv4/IPv4→IPv6 変換比率，プロトコル別変換比率，負荷分散比率はそれぞれ図 1〜図 3 となった．ICMP は実装上 Echo Request/Reply で一つのセッションとみなされるため，必然的に多くなっている．それを加味すると TCP の変換が一番多く行なわれた．また，IPv4→IPv6 変換より IPv6→IPv4 変換の方が多く利用されていることがわかる．

IPv6→IPv4 の変換数 4597 セッションの内，MIP6 を用いたものは，1591 セッションあった．IPv6→IPv4 の変換総数を 100% とすると，MIP6 を用いたものは 34.6% となり，MIP6 とも十分に連携できた結果となった．また，TTB トランスレータのコネクション数に基づいた分散状況は，表 1 となった．TTB2 と TTB3 の使用状況が 1:1 の割合だったことがわかる．

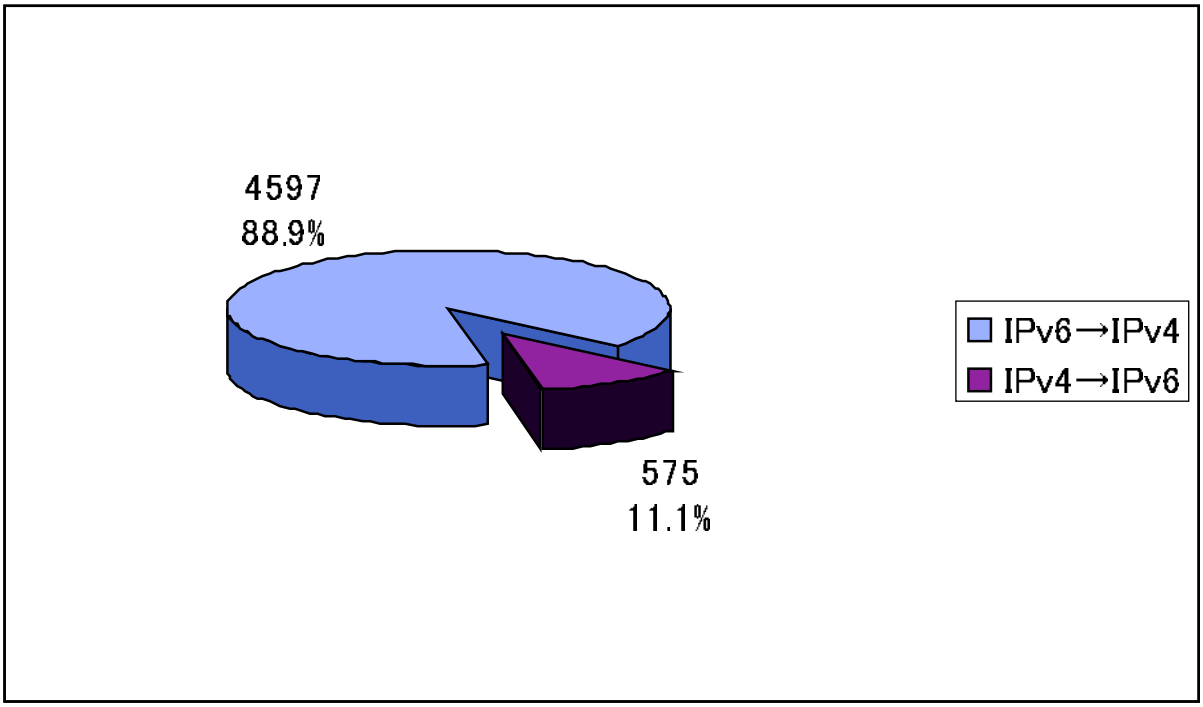

図 1.4: IPv6 → IPv4 と IPv4 → IPv6 の変換比率

TTB トランスレータ	コネクション数
TTB2	2578
TTB3	2594

表 1.1: TTB トランスレータ分散状況

1.3.6 まとめ

今回の実験結果により，大規模な仮説ネットワークベットにおいて TTB を用いた IPv4 → IPv6 変換が安定動作することを確認できた．ただし現状においては IPv6 → IPv4 変換のほうが需要が高いことがわかる．MIP6 コレスポンデント 機能は，トランスレータは十分に連携できることが確認できた.

1.4 VNC for XCAST6 評価実験

1.4.1 VNC for XCAST6 とは

遠隔のコンピュータのデスクトップを共有するアプリケーション VNC(Virtual Network Computing)[1] を XCAST6 に対応させたものである。

VNC のサーバ (Xvnc) とクライアント (vncviewer) との間のプロトコルは、『画面から書き変わった長方形の領域を切り出して転送する』という RFB(Remote Frame Buffer) と呼ばれる簡素な概念に基づいて設計されている。VNC for XCAST6 とは、この画面転送部分を XCAST6 に対応させたものである。クライアントで発生したマウス動作、キーボード操作等のイベントは、TCP の接続を通じてサーバ側に通知され処理される。

1.4.2 VNC for XCAST6 の課題

VNC for XCAST6 を用いたグループ通信をインターネットで実用展開するには、次に挙げる課題がある。

課題 1 RFB の送信速度制御、RFB 間引き転送に対する利用者の使用感の関係の分析

課題 2 Xvnc(VNC サーバ) のセッション広告およびグループ管理方式の確立

課題 3 ヘテロな受信者の能力や利用可能な帯域に応じた適用流量制御の確立

課題 4 他の TCP と公平な帯域共用を実現する輻輳制御の確立

[1] http://www.realvnc.com/

1.4.3　実験の目的

上記課題１、２の解決、および XCAST6 を用いたグループ通信の研究開発の WIDE プロジェクト内での普及促進を目的として、2003 年春合宿で VNC for XCAST6 評価実験を実施した。

1.4.4　実験システムの構成および実験の方法

実験システムは次の構成要件で構成した。

- 複数の受信者に対して VNC サービスを提供する Xvnc サーバ
- 損失率等を記録しながら VNC サーバの RFB を受信し、画面に表示する vncviewer
- 受信者のリストを管理するグループ管理サーバ (xcgroupsrv.cgi)
- グループ管理サーバから受信者のリストを取得し Xvnc サーバに供給する機能 (xcgroup-a)
- 受信者の IP アドレスをグループ管理サーバに登録する機能 (xcgroup-b)
- 損失率等のログデータを収集管理するログ収集サーバ
- vncviewer が記録したログデータをログ収集サーバに自動更新するログ収集クライアント

合宿ネットワーク上のサーバセグメントに、Xvnc サーバ、グループ管理サーバ、グループ管理サーバおよびログ収集サーバを導入・設置した。実験参加者には、XCAST6 が利用できる X 端末を準備してもらい、この端末に vncviewer、xcgroup-{a,b}、ログ収集クライアントを導入し、実験のセッションに参加してもらった。実験のセッションは、Xvnc の送信速度 (1Mbps,10Mbps,100Mbps)、RFB の転送方式（Raw,RRE) を変化させ複数回開催した。参加者には、合宿アンケートシステムで、Xvnc の送信速度と使用感に関するアンケートに返答してもらった。また、合宿中の BOF において、Xvnc サーバのセッションの運用課題に関する議論を行った。

1.4.5 実験結果

実験では、延べ１０台、５名による参加が得られ、次の事実が確認できた。アンケートでは、４名の参加者から使用感に関する有効な回答が得られた。

- インストール大会を合宿初日に開催したが、カーネルの入れ替えを要するため実験参加のコストは高く、多数の参加者を得ることは難しい
- 実験のセッション広告が十分でなかったため、インストールは完了したが、接続に失敗したユーザが存在した
- 1Mbps の速度で 640x480 の画面を Raw 転送方式で伝送した場合でも、使用感に問題を感じない利用者がいた
- 1Mbps の速度で 640x480 の画面を RRE 転送方式で伝送した場合、使用感に問題を感じない利用者が増えた

また、合宿中の BOF において、１台のサーバで複数の Xvnc サーバ機能を同時に立ち上げるためには、現在の MBUS を用いた既存の xcgroup 実装は利用できないことを確認した（課題２）。また、この対策として、Xvnc サーバプロセスが直接グループ管理サーバを通信し、受信者リストを取得する方法を採用すべきであるという結論に至った。

1.4.6 まとめ

本実験では、利用者がグループで画面を見ながら協働作業を行う場合に許容できる画面更新の頻度を得るには、RRE 転送方式で、1Mbps の帯域を用いる必要があることが確認できた。この結果は、流量制御・輻輳制御を実現するトランスポートとして、画面の間引きを行う方式を採用することが可能であることの裏付けとなる。また、Xvnc サーバに代表されるように、１台の端末で複数セッションを同時に参加する際のグループ管理に関するの問題の共有と解決手法の検討ができた。　今後は、課題３、４を解決するトランスポートの設計・実装を進める。また、このトランスポートの評価用アプリケーションとして、VNC for XCAST6 を活用していく計画である。

1.5 実空間ネットワークを利用したコミュニケーション支援

SPEARS WG では、実空間の情報を利用しユーザへのサービスを行うインフラストラクチャを実空間ネットワークとして構築してきた。近年、ユーザ間のコミュニケーション支援を WIDE 合宿における実験の主な目的としているが、本年度は、RFID によるプライバシ保護機構および、構造化困難な ID を扱う名前解決機構に焦点を絞った。2003 年 9 月の本実験に先立ち、2003 年 3 月に予備実験を行った。詳細は 2003 年 9 月の本実験の章に述べる。

1.6 PGP を応用した自由通貨システムの評価実験

1.6.1 目的

この実験の目的は、$(PG)^3A$ (PGP/GPG Applications) WG にて開発した自由通貨プロトコルを実装するアプリケーションを評価することであり、次の評価項目を予定していた。

1. 自由通貨プロトコルの検証
2. アプリケーションのヒューマンインタフェースの評価

1.6.2 概要

自由通貨は、円やドル等の法定通貨と異なり、個人や法人がコミュニティ内における自らの信用に基づいて発行する通貨である。自由通貨には、予算制約に因われずに自由な経済活動を促進できるという利点があり、代表的な応用例として地域の振興を目的とし、地域内で流通させることが多いことから地域通貨とも呼ばれている。

この実験では、自由通貨を電子的に利用可能にするプラグインを組み込んだ Jabber クライアントである Wija を配布し、合宿参加者を中心に利用してもらうことにより、データを収集することを狙った。

自由通貨プロトコルについては IDEON/$(PG)^3A$ WG 共同報告書で詳細に説明しているので参照されたい。

1.6.3 実験環境

自由通貨システムは、継続的に運用してこそ利用価値が生まれるため、合宿ネットワークに依存せず、継続的に利用できるシステム構成を心がけた。

自由通貨プロトコルを実行するエンティティは Jabber のクライアントに置き、Jabber サーバは一般に公開されているものであればどれでも使用可能とした。(ただし、少なくとも合宿当時は jabber.jp とその他のサーバとの間に Jabber プロトコルでの到達性がないことが問題となった。)

実験参加者のコンピュータには次をインストールする必要があった。

- Java 2 Standard Edition Runtime Environment 1.4.1 (1.3.1 以上で可)
- GnuPG 1.2.1 (PGP Freeware は不可)
- Wija 0.1 (実験用 Jabber クライアント)

自由通貨プロトコルでは PGP を利用しているため、利用者は事前に安全な形で鍵交換を行なっている必要がある。それを支援するため、公開鍵のフィンガープリントを貼り付ける掲示板を用意した。

1.6.4 結果

264 名が参加した 4 日間の合宿で、実際に自由通貨を利用できたのは 7 名のみだった。アンケート調査により判明したその理由を表 1.2 に示す。

表 1.2: 自由通貨を利用しなかった理由

利用する機会に恵まれなかった	13.8%
利用するのが面倒に思えた	11.5%
GnuPG, Java2 をインストールしたくなかった	10.3%
説明が不十分で理解できなかった	8.1%
利用する意義が見い出せなかった	6.9%
利用方法を思いつかなかった	4.6%
PGP/GnuPG が不得手	3.5%
クライアントが動作しなかった	2.3%

アプリケーションのヒューマンインタフェースの評価という点では、純粋に Jabber クライアントとして評価されたものも含め、多数の指摘を頂いた。逐次、Wija に反映させる作業を続けている。

1.6.5 まとめ

この実験の結果を受け、次のことが重要だとの認識に基づき、新たに実験用システムの開発を行なうことにした。

1. 利用の機会が提供されること
   - 合宿の参加者全員が関わる活動であることが望ましい
2. 特別なソフトウェアをインストールしなくても利用できること
3. 簡単な原理に基づいており、理解しやすいこと

このことが第 2.6 節で述べる WIDE Hour の実験へとつながった。

1.7 Mobile IPv6 operational experiment in a real environment

1.7.1 Objectives

Recently, the demands for mobility in the Internet is growing, because of the NAT-free address space obtained by IPv6 and many wide area data communication technologies. The world, that we can assign addresses to any devices we have and connect to the Internet using several access technologies from every our belonging such as PDA, is coming. In such an environment, we need mobility support for IP. Unfortunately, no network service provider is providing IP mobility service now. We must convince both Internet users and network service providers that the IP mobility technology is available and operatable. Otherwise, commercial carriers won't provide such a service in the future and users never realize the necessity of mobility. In this experiment, we demonstrate the mobility operation and services to convince participants of the WIDE camp as a first step for the mobility service deployment in a real world.

The experiment has following four objectives.

- Operation of a layer 3 mobility service
  - Discover any operational problems by operating home agents in the WIDE camp network
- Demonstration of a benefit of the Mobile IPv6 route optimization
  - Make users feel that a route optimization mechanism decreases the delay of responses between a mobile node and a correspondent node
- Demonstration of the vertical handover in a hybrid network environment
  - Make users feel the benefit of moving between different media, by designing a hybrid network which consists of Ethernet and wireless LAN
- Actual proof of a combination with other networking technologies and Mobile IPv6 technology

1.7.2 Network configuration

Figure 1.5 describes the network configuration we constructed.

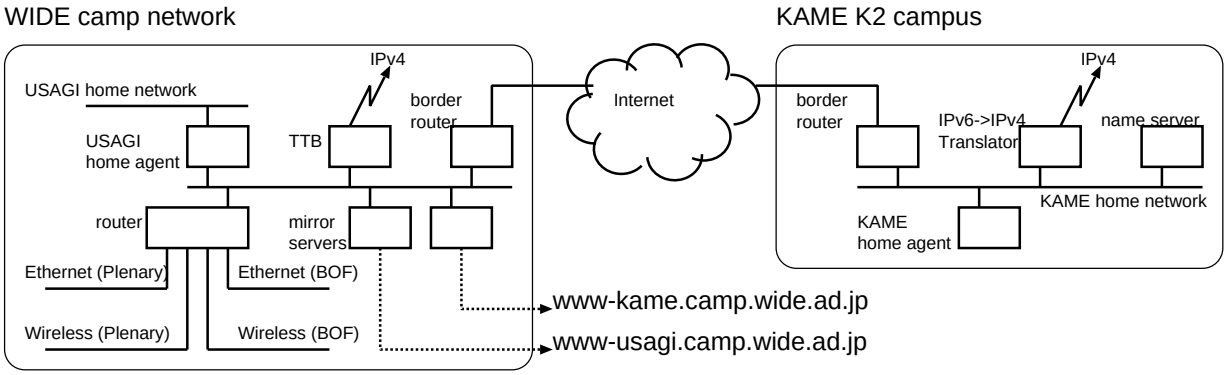

図 1.5: Network topology

In this experiment, we designed the WIDE camp network as a set of 4 networks (2 networks by Ethernet and 2 networks by wireless(IEEE 802.11b)). We operated 2 home agents and 2 home networks as a small layer 3 mobility service provider.

Also, we put mirror servers of the WIDE camp web server, which have Mobile IPv6 correspondent node functions. Any user who have Mobile IPv6 mobile node function can access the mirror servers using route optimization.

As we already mentioned, the WIDE camp network consisted of different access media, which were Ethernet and wireless LAN, each had a different subnet prefix. All users, whose node support Mobile IPv6, can roam from one subnet to another subnet without terminating on-going connections.

1.7.3 Result

Participants

The number of participants is shown in table 1.3. 14 nodes participated in this experiment as mobile nodes.

Operating System	# of participants
FreeBSD + KAME	5
NetBSD + KAME	1
Linux + USAGI (PC)	6
Linux + USAGI (Zaurus)	2
Total	14

表 1.3: The number of participants

Transition of home registration entry

Figure 1.6 shows the transition of the number of home registration entries on the KAME home agent during the camp.

We saw maximum 14 entries during the camp. KAME creates 2 binding cache entries for every one home registration, the home agent served

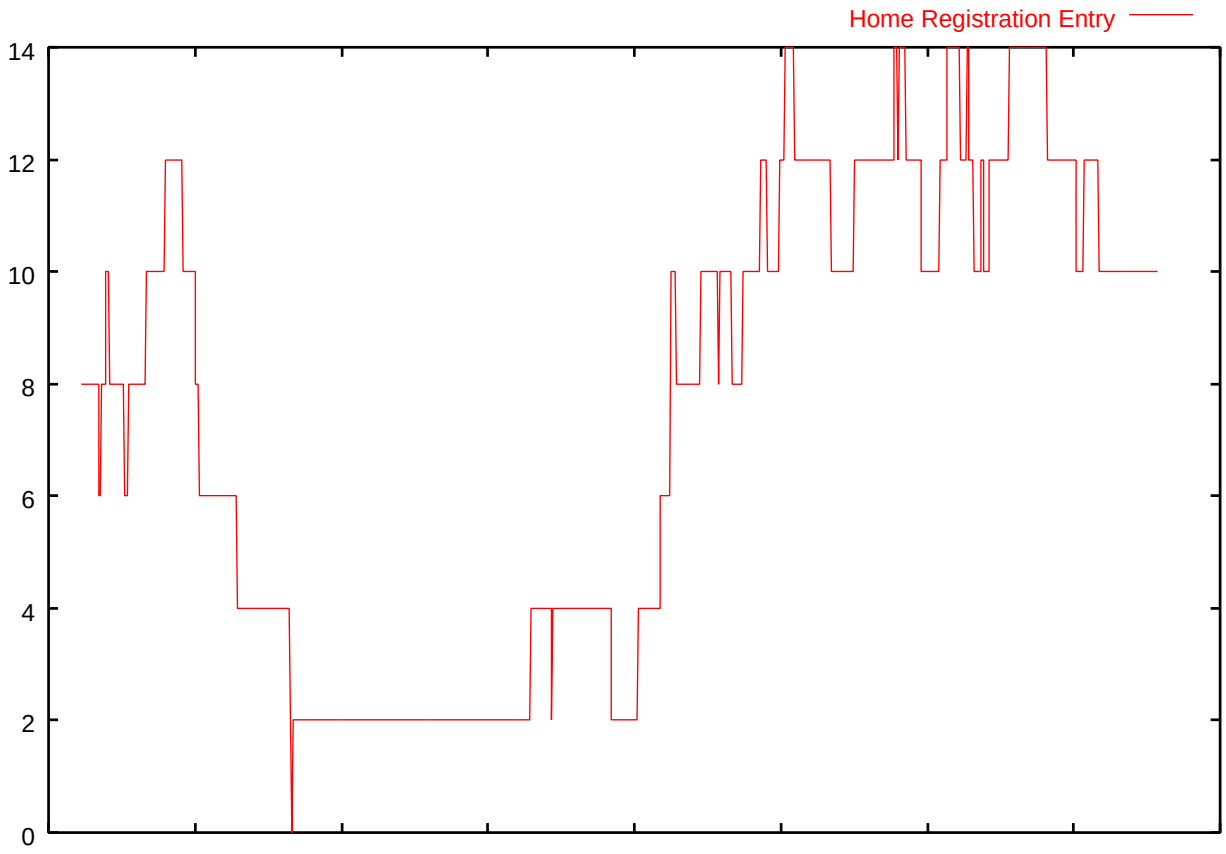

図 1.6: The number of home registration entries

maximum 7 mobile nodes at the same time.

Transition of binding cache entry

Figure 1.7 shows the transition of the number of binding cache entries of the KAME mirror server during the camp.

The mirror server had maximum 3 binding cache entries. Different from home registration, one binding cache entry is created per one mobile node.

Route optimization ratio

Figure 1.8 shows the number of IPv6 packets, Home Address Destination Option (HAO) and Routing Header Type 2 (RTHDR2), which the KAME mirror server sent and received.

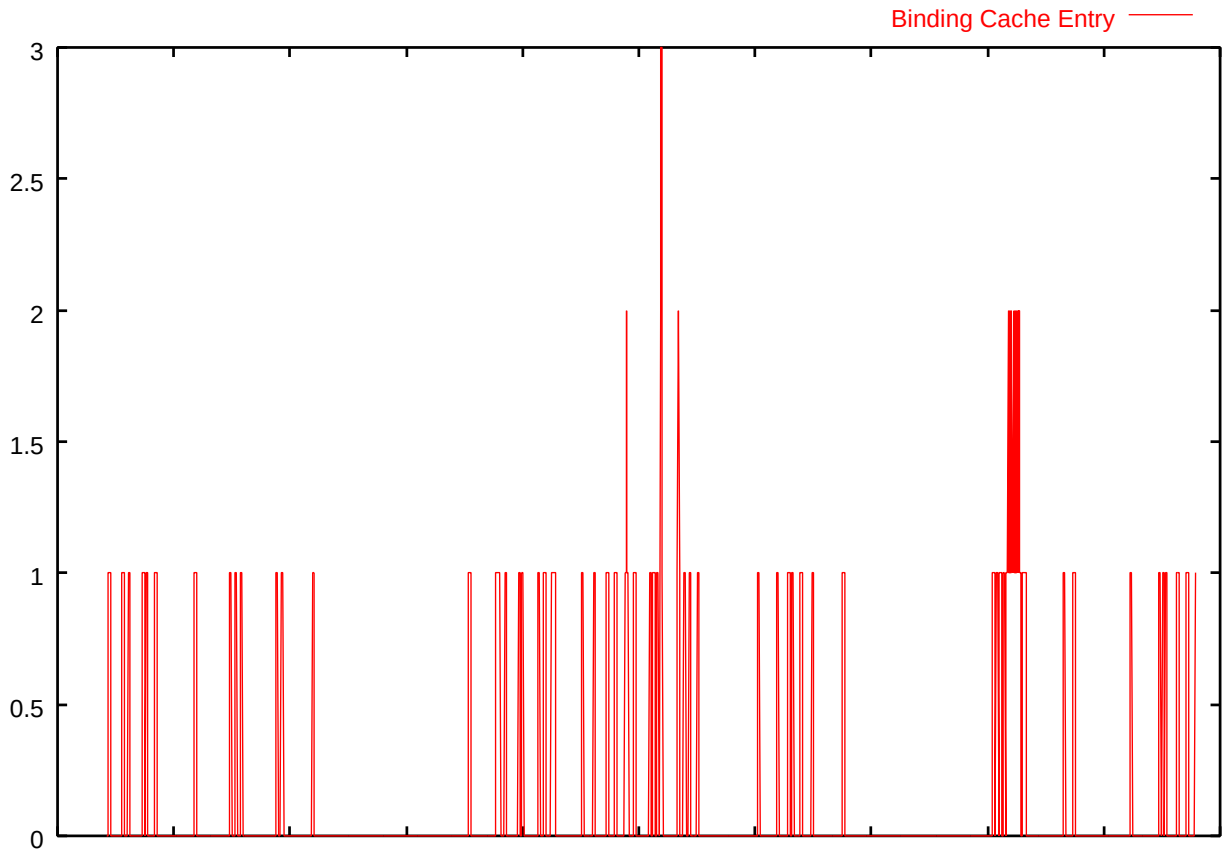

図 1.7: The number of binding cache entries

1.7.4 Combination with other technologies

TTB

TTB is a production quality equipment for IPv6-IPv4 protocol translator. IPv6 only nodes can communicate with IPv4 only nodes using TTB translation mechanism, and vice versa. In this experiment, we use TTB as an IPv6 to IPv4 translator so that Mobile IPv6 nodes can communicate with IPv4 nodes in the Internet. A mobile node specify TTB address as a name server address. When a mobile node try to resolve the IP address of IPv4 only node, TTB returns an IPv6 address which is corresponding to the translation table TTB keeps. A mobile node believes that the node, which is connecting to, has the IPv6 address, in fact it doesn't have. TTB receives all traffic from mobile nodes to IPv4 only node via the IPv6 address which returns as a response of a DNS query from mobile nodes, and trans-

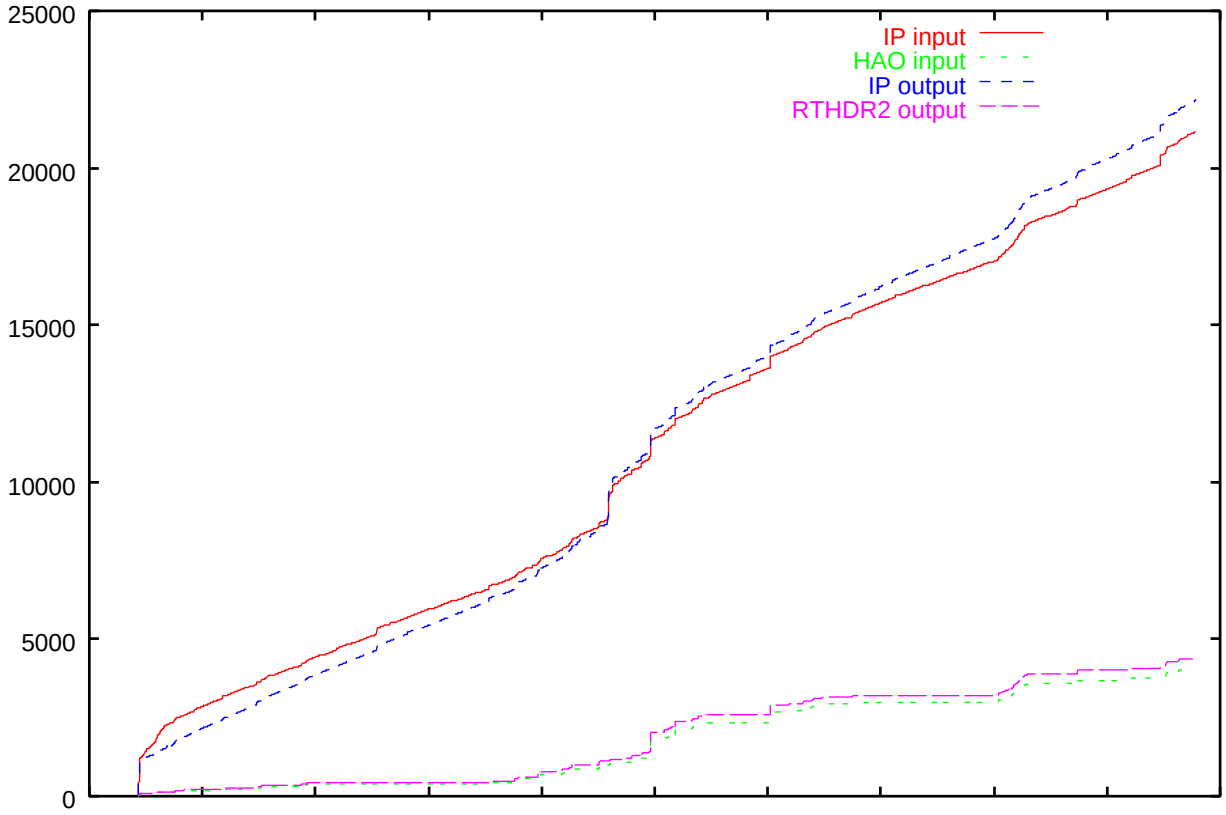

図 1.8: The number of HAO and RTHDR2 packets

late IPv6 packets to IPv4 packets. In a reverse direction, TTB receives all traffic from IPv4 only nodes and translate them to IPv6 packets and forward them to mobile nodes. From a mobile node point of view, IPv4 only nodes can be treated as a IPv6 node which doesn't support Mobile IPv6. Since the Mobile IPv6 specification allows to communicate with IPv6 nodes which doesn't support Mobile IPv6, there are no problem in communication between mobile nodes and IPv4 only nodes via TTB.

We suggested all participants to use TTB as their name server, so that they could communicate with IPv4 nodes. They could transit from IPv4 environment to IPv6 mobility environment without losing IPv4 communication. In addition, this provided not only IPv6 mobility, but also pseudo IPv4 mobility as a side effect.

VoIP

The always asked question when we demonstrate Mobile IPv6 is what is the good application for mobility. In this experiment, the USAGI project demonstrated SIP and VoIP over Mobile IPv6. The application used in this experiment was Linphone 0.10.1. Linphone was prepared for Linux USAGI stack, since we should apply some fixes to the original Linphone because it didn't work well with IPv6 network. The patched Linphone was distributed before the WIDE camp started. Linphone worked well in the camp network. We can communicate between Linphones running on a Linux PC and a Linux Zaurus, and between two Linux Zauruses.

1.7.5 Consideration

We operated two home agents during the camp. The operation itself was quite stable and we saw no problem in providing Mobile IPv6 service to participants. The most complicated task in the experiment is to set up IP security associations between mobile nodes and home agents. We felt we need some automatic mechanisms to reduce the load. Regarding to the number of participants, we had only 14 participants out of over 300 participants at the WIDE camp. We thought the reasons why we couldn't get many participants are as follows:

- Most people don't use UNIX/Linux recently. They cannot participate in our experiment, since we cannot provide more popular operating systems, for instance, Windows and MacOS.
- We requested participants to replace their kernel with the other kernel we provided to add mobility support to the kernel. Usually, people don't want to replace their kernel to relatively unstable one.

The first problem is difficult to solve by ourselves. We cannot add mobility function directly to the commercial operating systems, because of the following two reasons:

- Their networking code is very different from *BSD and Linux which we are using to develop our mobility protocols.
- Some of them don't disclose the source code of their own.

The second problem can be solved by merging Mobile IPv6 parts to the original UNIX/Linux distributions. We are planning to merge our code to the original trees. This work in one of most important tasks in the next year.

Next, though we tried to feel people the merit of route optimization feature specified in the Mobile IPv6 specification, we couldn't show it effectively. Figure 1.8 shows some kind of route optimization ratio, however, it has almost no meaning. There were only 3 clients which used route optimization feature as we can see in Figure 1.7. This number is quite small compared to the number of participants (14 participants). Most participants used the official web server, not our mirror severs. We should have pointed our mirror servers from the official dns name of the web server so that users could use our mirror servers transparently.

Next, we provided Ethernet networks and wireless LAN networks so that participants could move from one communication media to another media. This worked technically, however, most participants seemed to use only wireless LAN networks, since the WIDE camp network provides very small number of Ethernet port to its participants. We think the WIDE camp is not a suitable place for the demonstration of vertical handover. We should prove it more realistic way, for instance, moving between PHS and wireless LAN.

Finally, the combination with other networking technologies worked well. Mobile IPv6 requires no changes to upper layer protocols and applications, since it is a layer 3 protocol. We used TTB as a protocol translator and VoIP as a communication application. These application worked causing no problems with Mobile IPv6.

1.7.6 Conclusion

In the near future, we will see the network environment which we must move from one network to another network. For instance, we have PHS data connection in most of our workplaces, and at the same time we may have wireless LAN services. When we move from PHS only area to wireless LAN area, it is natural for us to change the access technology from PHS to wireless LAN. In such a situation we never want to terminate our on-going connections when switching our access network from PHS to wireless LAN.

To make the above scenario possible, we need following things:

- A layer 3 mobility protocol which doesn't depend on underlying communication technologies and doesn't require any changes to upper layer protocols.
- A layer 3 operator service

In this experiment, we used the WIDE camp network as a model of the future Internet which consists of various communication media and users move among them. We chose Mobile IPv6 as a layer 3 mobility protocol, since it doesn't depends on layer 2 protocols and doesn't require any changes to upper layers. Also, we provided Mobile IPv6 service as an operator. Our operation worked well. We will extend this kind of actual proof of a mobility service in a wide area network.

1.8 パッシブモニタ (CpMonitor) によるトラフィック情報可視化実験

1.8.1 実験の目的

開発中のパッシブ型ネットワークモニタ (CpMonitor) を合宿ネットワークにおいて運用し、基本的なネットワーク観測の有用性・必要性を示すとともに、新機能 (802.1Q VLAN 対応) の実環境における試験を行う。また、今後データ解析を行うためトラフィックデータの収集を行う。

1.8.2 実験の概要

CpMonitor とは、ネットワークを流れる通信パケットを受動的に観測し、これを計測し公開する目的を持つ高機能なネットワークモニタである。IPv4・IPv6 トラフィックに対応し、TCP・UDP・ICMP の別やアプリケーション (ポート番号)、VLAN ID 毎にそれぞれ独立してパケット数・通信量を計測することが可能である。またネットワークタップ装置を介しての接続やスイッチのミラーポート接続に対応し、既存のネットワーク構成に影響を与えることなく組み込みが可能な特徴を持つ。また、情報の公開には SNMP を用い、標準への準拠を行っている (http://www.cysol.co.jp/products/cpmonitorsmart/index_j.html)。

この CpMonitor を合宿ネットワークに組み込み、対外線 (衛星および地上線)、および合宿地内の複数のセグメントと基幹ルータの間のトラフィックを観測した。また合宿地内にマネジャを設置し、SNMP による CpMonitor エージェントからのデータ収集、Web および Java によるトラフィックグラフのリアルタイム公開を行った。

また、各実験チームの要請に応じ、実験トラフィックの観測も随時行った。

1.8.3 実験環境

CpMonitor エージェントを 2 台、うち 1 台はマネジャ兼用として、合宿ネットワーク内に設置した。合宿ネットワークへの接続はネットワークタップ装置を用い、観測対象とするネットワーク接続に影響を与えないよう留意した。観測対象としたのは以下の 3 リンクである。

- 地上線
- 衛星
- 合宿地内ネットワーク

なお、合宿地内では 802.1Q を用いた VLAN ネットワークが構成されていた。

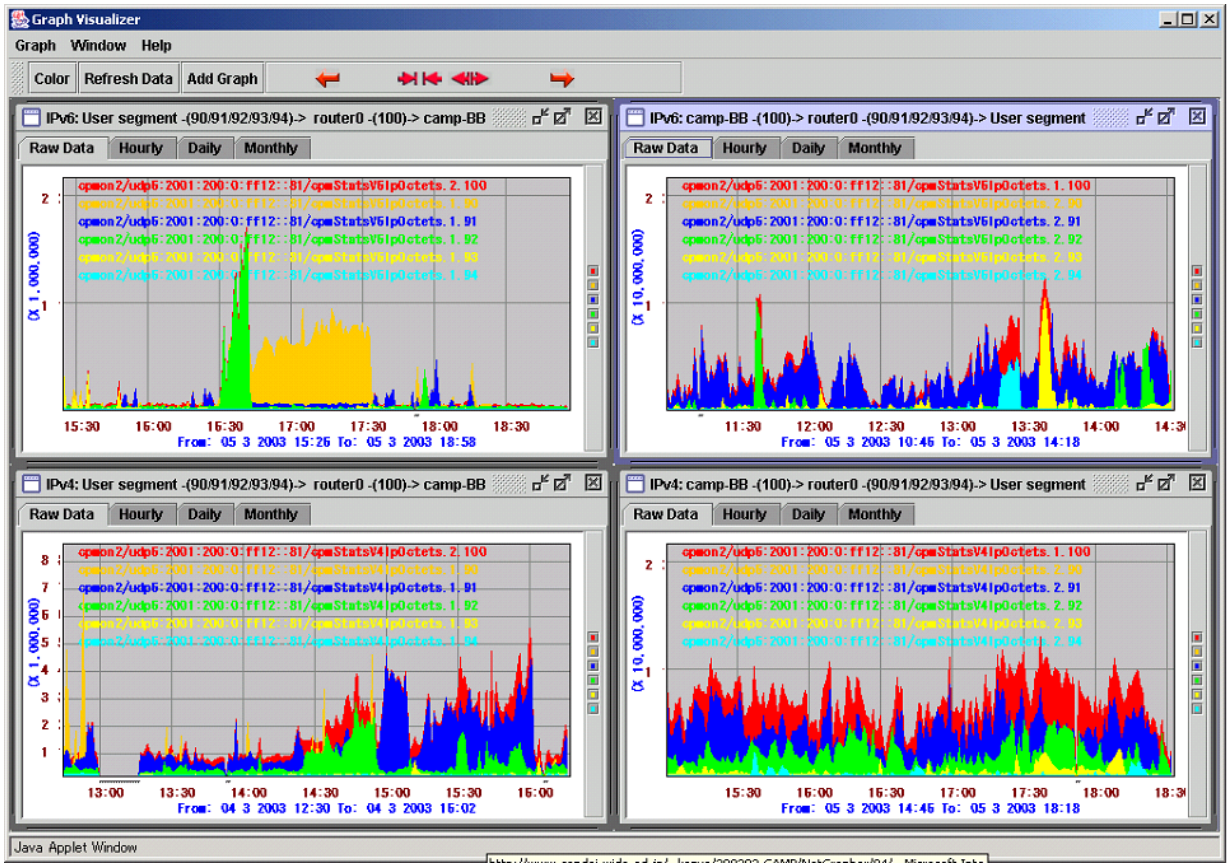

図 1.9: CpMonitor によるトラフィックデータの公開

1.8.4　結果

本システムは概ね合宿 2 日目より安定し、最終日まで連続して稼働した。インフラチームによる MRTG データ収集も問題なく行われた。

参加者への情報公開は、以下のようにあらかじめ分類して行い、わかりやすさを狙った。

- 衛星リンク – 地上リンク利用バランス
- 合宿地各ユーザセグメント – バックボーントラフィック状況
- プレナリ無線・有線利用状況

データ公開の様子を以下に示す。

また、3 日目に起きた通信トラブルをこのシステムによりとらえた様子を以下に示す。明らかに異常な量の、普段観測されない種類のトラフィックが断続的に流れていることがわかる。

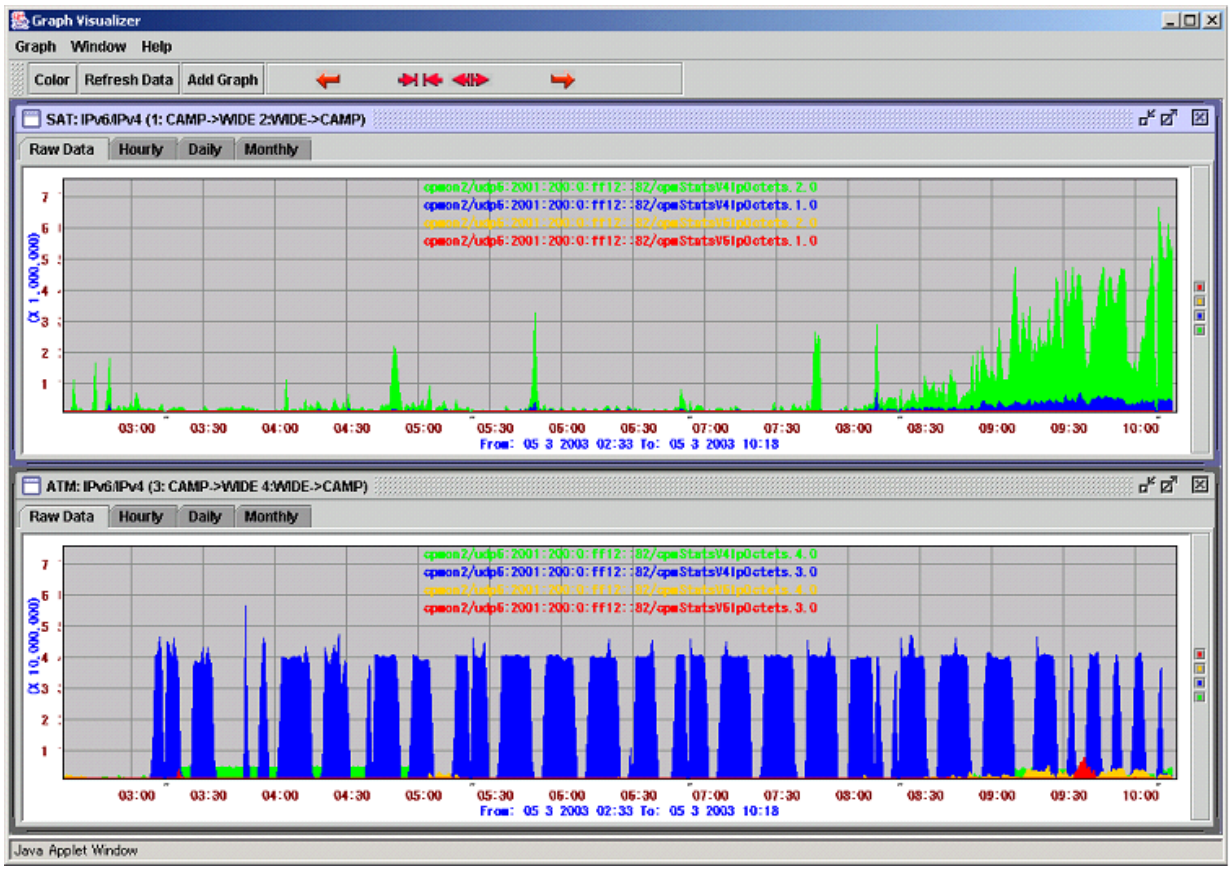

図 1.10: CpMonitor による通信トラブルの発見

また、データ収集パラメータは以下の通りであった。

- 収集箇所
  - 地上線 (gr2k – nagahama–bb 間) 上り・下り
  - 衛星 (sat–camp – nagahama-bb 間) 上り・下り
  - 合宿地内ネットワーク (router – L2 スイッチ間) 上り・下り
  - 802.1Q VLAN 90, 91, 92, 93, 94, 100
- 収集対象以下それぞれの通信量およびパケット数
  - IPv6 – TCP, UDP, ICMP 主要アプリケーション (port 22, 80, 110, 143, 993, 995, 2401, 5999)
  - IPv4 – TCP, UDP, ICMP 主要アプリケーション (port 22, 80, 110, 143, 993, 995, 2401, 5999)
- 収集間隔 60 秒

また期間中、VNC 実験グループのトラフィック観測を併せて行い、これを全うした。

これらにより最終的に蓄積されたデータ量は 164,801 キロバイトとなった。データは以下の URL で現在も自由に閲覧することができる (http://www.sendai.wide.ad.jp/ kazuo/200303.CAMP/NetGrapher/)。

1.8.5　まとめ

今回の実験により、CpMonitor のネットワーク運用における安定性・有用性が確認された。また新機能 (802.1Q VLAN 対応) も問題なく動作することが確認され、今後の運用の幅がいっそう広がることとなった。

一方で、現行の CpMonitor では、実験や事故による突発・偶発的トラフィック、ダイナミックにポートを変化させるアプリケーションのトラフィックなど、あらかじめ想定し設定を行えないトラフィックの観測には対応し辛いことが明らかとなった。

今回の実験で明らかになった問題は次回以降改善の予定である。

第2章　2003 年秋合宿ネットワーク

本章では、2003 年 9 月 8 日 (月) から 11 日 (木) まで静岡県浜名湖ロイヤルホテルにおいて開催された WIDE プロジェクト秋合宿 (以降、本合宿) におけるネットワーク構成および，そのネットワーク上で行われた実験の内容とその結果を報告する.

2.1　ネットワーク構成

図 2.1 に本合宿中のネットワーク構成の一例を示す.

図中，左右の点線より上部が慶應義塾大学湘南藤沢キャンパス (SFC) であり，下部が合宿地である． 四角はルータおよびホスト (サーバ) を表し，線はイーサネット，専用線，無線 LAN を表す． 二点鎖線は衛星回線を表す.

合宿会場とインターネット (SFC) との接続には地上線として ATM(3Mbps) および衛星回線 (上り 512kbps, 下り 1.5Mbps) を用いた．衛星回線には IPv4 の HTTP および HTTPS のみ流すようにした． 合宿会場の commodity network として各部屋などに有線および無線セグメントを配置し，IPv4 および IPv6 の接続性を提供した．IPv4 は DHCP によるアドレス割り当てである． また，この commodity network とは別に，HTTP など合宿会場での各種サービスを提供するサーバおよび各実験で利用するサーバなどを接続するサブネットを用意した.

合宿中運用にかかわる大きな障害は発生しなかったが，合宿地にて MSBLAST に感染したノードによるパケットによって一時的に接続性が失われることがあった．感染ノードは発見次第 switch にて mac address を元に filter することにより，合宿ネットワークへ影響を与えないよう配慮された.

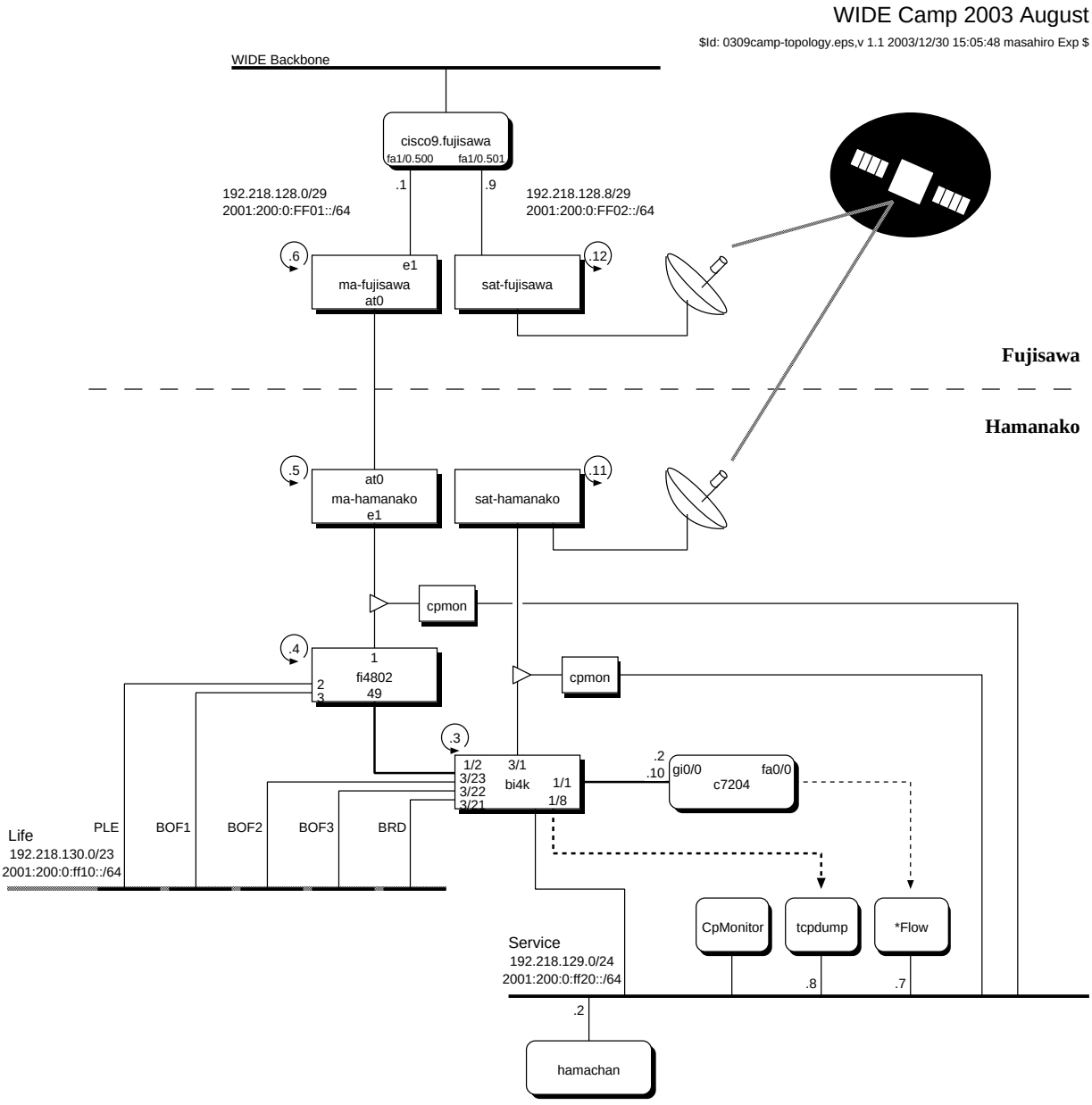

図 2.1: 本合宿のネットワーク構成

2.2 実空間ネットワークを利用したコミュニケーション支援

SPEARS WG では、実空間の情報を利用しユーザへのサービスを行うインフラストラクチャを実空間ネットワークとして構築してきた。近年、ユーザ間のコミュニケーション支援を WIDE 合宿における実験の主な目的としているが、本年度は、RFID によるプライバシ保護機構および、構造化困難な ID を扱う名前解決機構に焦点を絞った。2003 年 9 月の本実験に先立ち、2003 年 3 月に予備実験を行った。

2.2.1 RFID によるプライバシ保護機構の構築

近い将来、RFID を用いて実空間の物体の存在を認識する技術が広く利用されるようになると予想される。しかし、RFID システムは、タグとリーダ間で無線交信を行うため、リーダを持っていれば誰でも RFID タグの中に格納されている ID が取得できてしまう。

SPEARS WG は、WIDE 合宿において従来よりご飯チェック・アプリケーションを構築、運用してきた。このサービスは、来場者に RFID を携帯させることで、食堂の入り口で入退場の可否管理および履歴の記録をしている。従来この RFID には WIDE 番号を格納していた。この運用形態では、RFID を携帯する本人に気づかれることなく WIDE 番号が読み取れてしまう。この結果、WIDE 番号および関連付けられた情報の漏洩が、RFID を携帯する本人のプライバシを侵害する可能性が指摘されていた。

本実験では、このような第三者へのプライバシ漏洩を防止する試みとして RFID に格納する ID を暗号化する機構の試作を行った。

本機構では、プライバシ漏洩防止のアプローチとして、RFID に格納される ID を暗号化し、暗号化された ID を認証されたリーダ経由でのみ復号化可能とすることでプライバシを保護を実現した。具体的には、ネットワーク上に認証・復号化サーバを置き、リーダの認証と ID の復号化を行った。暗号化された ID を読み取ったリーダは、「自身（リーダ）の ID」と「暗号化された ID」をサーバに送信する。サーバではリーダの ID を用いて認証を行い、認証されたリーダに対しては、暗号化された ID を復号化して送信する。以上により、認証されたリーダでは復号化された ID を取得可能となった。

2003 年 9 月の WIDE 合宿では、本機構をご飯チェック・システムおよび、

Introduction Service のアプリケーション・プログラムと連動させ、有効に動作した。

2.2.2　構造化困難な ID に対応した名前解決機構の構築

RFID の普及に伴い、個体認識技術を応用した様々な研究が行われている。その中の一例として、EPC Global、Auto-ID Lab が中心となって研究を進めている Auto-ID システムが挙げられる。Auto-ID システムでは、EPC と呼ばれる ID を格納した RFID タグを製品に貼付し、非接触通信により EPC を取得し、個体認識を行う。

また、Auto-ID システムには、インターネット上に情報が展開して利用されるという特徴がある。RFID タグには EPC のみを格納し、EPC 関連する情報をインターネット上で管理する。インターネット上に展開する EPC による情報サービスは EPC-IS と呼ばれている。

EPC に関連する情報操作をインターネットを介して行うために、EPC-IS の利用者やソフトウェアは、該当する EPC に関連する EPC-IS のインターネット上での位置情報を認知する必要がある。Auto-ID システムでは、EPC とその EPC に該当する EPC-IS の位置情報の対応を管理し提供する名前解決機構として ONS が提案されている。

提案されている ONS は、EPC を FQDN エンコーディングし、EPC-IS の位置情報管理に DNS を用いている。なお、DNS RR として NAPTR RR が持ちりられ、EPC-IS の位置情報は URL 他の情報として記述される。

EPC は、製造者、製品種類、シリアル番号に構造化されている。また、プライバシ保護の観点から、EPC を暗号化するなどして製品情報を推測できないようにする提案がされている。暗号化された EPC や、自家発番した EPC などは従来の DNS を用いた ONS では扱いにくい。Auto-ID が普及することで、構造化されない EPC や、構造化が困難な EPC に対応した名前解決機構が必要となることは容易に予想ができる。

2003 年 9 月の実験では、Auto-ID システムにおいて構造化されない ID を扱う利用場面を想定し、構造化されない ID と製品情報を保持するサーバとの対応付けを行う名前解決機構の実装を行った。具体的には、合宿参加者に配布している RF-CODE 社の Spider Tag に格納される ID (Active Tag なので構造化されない) に対して、関連する情報サーバを想定し、その対応付

けを行う名前解決機構のプロトタイプを実装した。開発の都合でSPEARS WGのサービスの中へ組み込むことはできなかったが、デモを実施した。

本機構は、分散ハッシュテーブルを用いて、RFIDに格納されるIDを検索キーとしハッシュ空間にマッピングすることで、IDの構造と体系に依存しない名前解決を実現した。DNSを用いるONSに対し容易な登録を可能としその有意性を確認できた。

2.3 Implementation of Automatic Detection / Collection System for XSS Vulnerability

2.3.1 Purpose

In this working group we address the design of mechanism for detecting and collecting XSS vulnerability. Our purpose is to not only protect the participants against XSS attacks, but also collect and share the XSS vulnerability information, such as, XSS vulnerable web sites. We also discuss how to warn the XSS vulnerable web sites.

2.3.2 Abstract

Implementation of Automatic Detection / Collection System for XSS Vulnerability

Eliminating XSS vulnerabilities in server side by using security tools and by coding secure applications are not preventing the attackers from exploiting new XSS vulnerabilities. Our approach is detecting XSS attacks in the client side(such as the LAN the client connected to) by using network proxy server as a check point for investigating XSS vulnerabilities. The detection mechanism investigates the HTTP traffic which gets through the proxy server and collects the informations about XSS attacks.

Evaluation of this System

By deploying the system in WIDE camp it is important for us to test the system in a realistic environment. Investigating how the system works,

finding bugs and most importantly, knowing how much XSS vulnerability it covers is our main purpose.

2.3.3 Requirement

In order to investigate the HTTP traffics in WIDE Camp, we would like to ask you to do some configurations in your web browsers. Figure. 2.2 shows how we deploy our system. The detection/collection part which resides on a proxy server and the database are circled in red dash line.

2.3.4 Evaluation

We measure the performance of our proposed approach on the number of detected XSS vulnerable web sites.

Result

During the whole experiment period, the system investigated more than 8000 HTTP requests (HTTP GET requests) and the corresponding response messages. Through the detection mechanism, the system detected 13 XSS vulnerable web sites (note that those are not the XSS attacks, but the vulnerable web sites that could be used to launch the XSS attacks). Except some sample web sites we provided, the system detected a couple of new XSS vulnerable web sites.

Problem and Future Work

From the participants point of view, the system protected them against XSS attacks but it failed to provide a completely stable web environment. For example, the system generated some false alerts and as a result, the corresponding responses were blocked. Also, since every request is required to be checked, the participants felt that the system violated their privacy. Deploying the system directly in users computer by using local proxy might avoid those problem but this approach need to be tested.

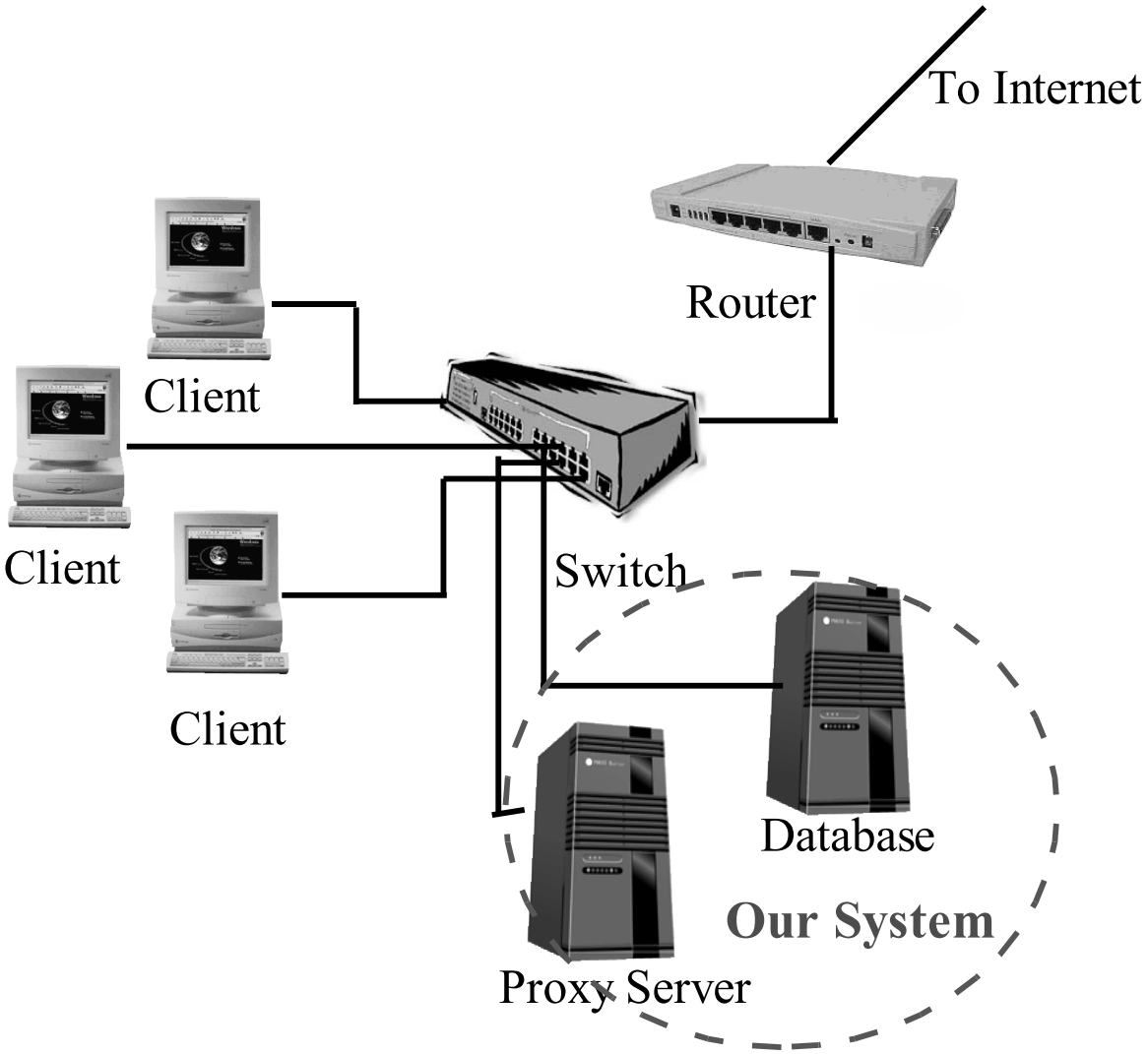

図 2.2: XSS Topology

2.4　無線 LAN 性能検証

802.11 ファミリの無線 LAN 技術の検証を行った。詳細については wlanops の項を参照のこと。

2.5　フローベースによる合宿ネットワーク計測

2.5.1　目的

本研究はネットワークトラフィックの傾向解析を目的としており、ネットワークモデルの一つとして合宿ネットワークをとらえトラフィック計測を行なった。

本研究ではネットワークトラフィックの性質について次の項目を仮定している。

- ネットワークの構成に依存する
- 流行しているアプリケーションに依存する
- 家庭（集合住宅など）や職場（会社や学校、店舗など)、各種イベントのようにネットワークを利用する環境に依存する
- 利用者の傾向に依存する
- ネットワークの規模（利用人数、利用頻度など）に依存する

ネットワークを利用する環境や利用人数、ネットワークの構成などでネットワークをモデル化できれば、ネットワーク構築を含むネットワーク運用へのフィードバックが期待できる。

例として奈良先端大学院大学の学生宿舎ネットワークを分類する。ただし、今回は分類に際してネットワークの構成については対外線の帯域のみに注目し、流行しているアプリケーションは考慮しない。

- 対外線は Ethernet(100Mbps 1 本と 10Mbps 3 本) である
- 奈良先端科学技術大学院大学生のための集合住宅
- 利用者は主に大学院生である

- 500 人程の人間が利用している、そのうち 100Mbps の対外線の利用者数がおよそ 125、10Mbps 3 本の利用者数がそれぞれおよそ 121、160、94 である

このネットワークおよび同様のネットワークの利用傾向を解析しモデル化出来た場合、ネットワークの運用や新たに大学の学生宿舎など同様のネットワーク構築へのフィードバックが期待できる。

本研究においてはモデル化する際に指標となるデータが必要となるため、データの蓄積が重要な要素となる。

今回の実験では合宿ネットワークを以下のように分類した。

- 対外線は地上線として ATM(3Mbps) および衛星回線 (上り 512kbps, 下り 1.5Mbps) がある
- WIDE 合宿というネットワーク技術の研究者が集まるイベント
- 利用者のほとんどはネットワーク技術の研究者である
- 200-300 人ほどの人間が常時利用している

本実験においては、このように分類されたネットワークのサンプルとして合宿ネットワークのトラフィックを計測した。

2.5.2 概要

今回の実験は合宿ネットワークのトラフィックを計測したものである。

本実験ではトラフィックを計測する際にパケットの形式での保存ではなくフローの形式で保存する事によってデータ容量の縮小化を図っている。なおこの計測においてはサンプリングを行なっていない。

計測環境

計測点としては、図 2.3 に示すように合宿ネットワークにおける基幹ルータとなった Cisco 7204 から NetFlow パケット（ver.5）を Capture Server に対して送信し、それを計測しデータベースに格納している。

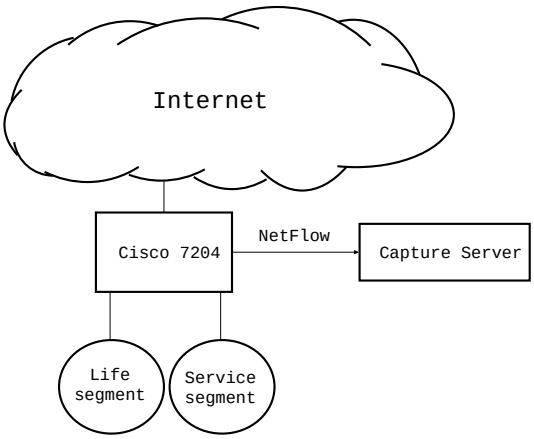

図 2.3: ネットワーク構成図

抽出パラメータ

キャプチャしたNetFlowパケットから、本実験では以下のパラメータを抽出している。

- unix_secs
- source IP address
- destination IP address
- source port number
- destination port number
- Layer 3 protocol
- packet count
- byte count
- input interface ID
- output interface ID

サーバ構成

Capture Server の構成は以下のようになっている。

- OS: FreeBSD 4.7-RELEASE

- CPU: Pentium III 1GHz
- Memory: 512MB
- Hard Disk: 80GB

データ収集の際の注意点

トラフィックにはユーザのプライバシに関する情報やネットワークの運用ポリシに関する情報が大量に含まれている。そのためトラフィックを計測するにあたって以下の3つの点を踏まえている。

- パケットのヘッダ部分の情報のみを利用する
- 収集したトラフィックデータは研究, 運用目的にのみ用いる
- 研究目的に用いる場合でも, データを外部に公開する際はホストやユーザの情報が特定されないようにデータを隠蔽もしくは変換する

2.5.3 結果

トラフィック計測は9月8日の 13時16分から9月11日9時46分まで行なった。

データベースに格納されたフロー数は401739である。この際のデータベースのデータ容量は約108MBである。同じトラフィックをtcpdumpを用いて先頭の68byteを計測したものはデータ容量が約11GBとなった。このことから、この実験ではフローの形式で保存する事によりデータの縮小化に成功しているといえる。

この期間のICPM,TCP,UDP毎の横軸に時刻、縦軸にフローカウントのグラフは図2.4-2.12となる。ただしTCPとUDPのグラフに関してはフロー数の総量のうち利用率の大きい5つのポート番号は利用率の高いものから順にそれぞれに色分けされ表示されている。

今回のWIDE合宿では図2.4-2.6で見られるように、突発的に起こるICMPのトラフィックが目立った。また、図2.7ではTCPでの通信において135番ポートでの通信が80番に次いで2番目にあがっている。これは今年の夏に大流行したW32/BlasterやW32/Welchiaと呼ばれるNetwork Wormが原因

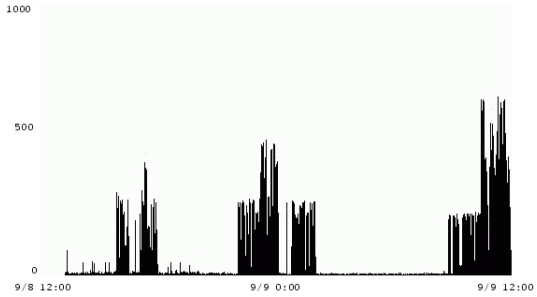

図 2.4: 9/8 12:00-9/9 12:00 ICMP フローカウント

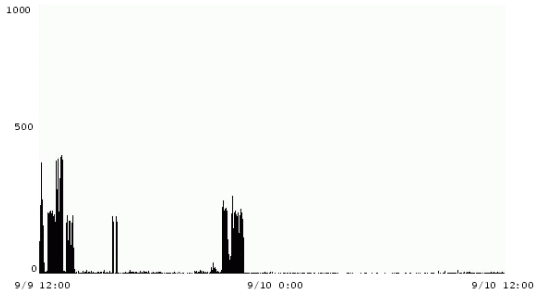

図 2.5: 9/9 12:00-9/10 12:00 ICMP フローカウント

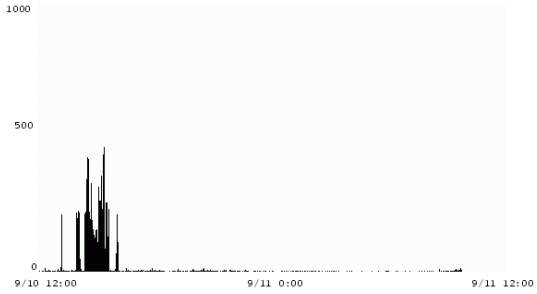

図 2.6: 9/10 12:00-9/11 12:00 ICMP フローカウント

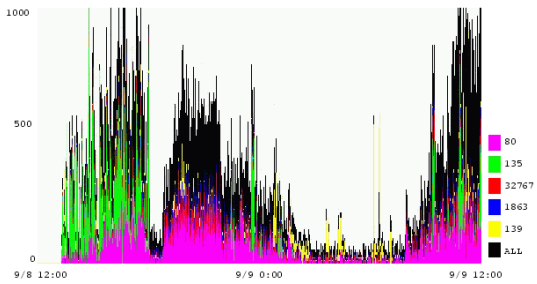

図 2.7: 9/8 12:00-9/9 12:00 TCP フローカウント

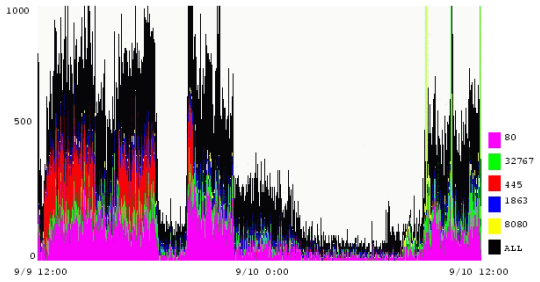

図 2.8: 9/9 12:00-9/10 12:00 TCP フローカウント

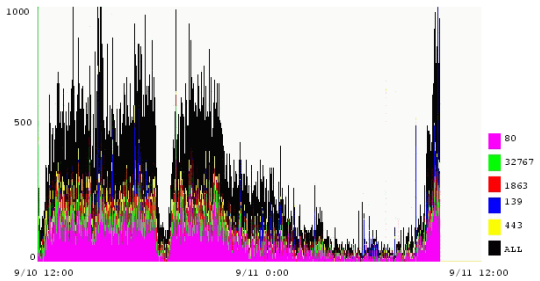

図 2.9: 9/10 12:00-9/11 12:00 TCP フローカウント

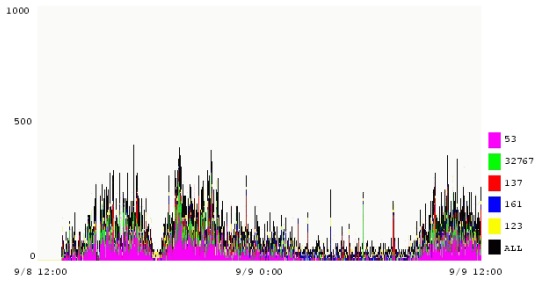

図 2.10: 9/8 12:00-9/9 12:00 UDP フローカウント

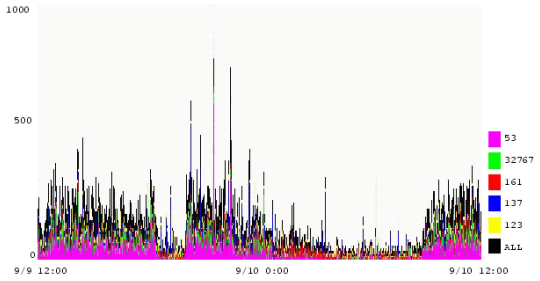

図 2.11: 9/9 12:00-9/10 12:00 UDP フローカウント

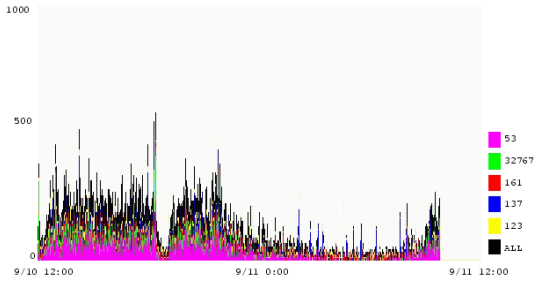

図 2.12: 9/10 12:00-9/11 12:00 UDP フローカウント

であると考えられる。これを受けてルータでの filtering 処置やユーザに対して注意を出したため、図 2.7-2.9 から分かるように TCP/135 番ポートでの通信は初日の夕飯頃を境に減少している。これ以外に図 2.7-2.12 からは深夜から朝にかけて利用数が下がっている事と、11:50〜13:00、18:30〜20:00 の食事時間はネットワークの利用数が急激に下がっている事が分かる。

2.5.4　まとめ

今回の実験では合宿ネットワークを、WIDE 合宿というネットワーク技術の研究者が集まるイベントにおける 200-300 人ほどの人間が常時利用しているネットワークと分類してデータを収集した。

その際に当時流行していた Network worm である W32/Blaster や W32/Welchia による被害が出ていたと考えられる。

今回のネットワークは基幹ルータに Cisco 7204 を採用したが PC ルータのように比較的スペックの低い機材を採用する場合、その PC ルータがスペックが十分かどうか検討の余地がある。

次回 2004 年 3 月に予定されている WIDE 合宿においても、合宿ネットワークは同様のモデルに分類できると思われる。

これを続ける事により、本研究ではネットワーク運用に対するフィードバック可能なモデル化手法の構築を目指す。

2.6 自由通貨を用いた協調メカニズムの検証 (WIDE Hour)

2.6.1 目的

利己的なノードがピアグループを形成する (人間の) ネットワークで、如何にシステム全体の要求仕様を満たすか、という問題に対して、自由通貨を用いて解決するというアプローチを検証する。

そのため、合宿生活を快適にするためのリソースのフェアな分配のために自由通貨 (第 1.6 節参照) による取引を適用し、その効果を計測することを狙った。

2.6.2 概要

WIDE メンバであれば利用できるポイント交換システム「WIDE Hour」を導入した。このポイント交換システムは、ポイントを表す単位である WIDE Hour と、システム内での信用を表す単位である WIDE Power から成る。

WIDE Hour:

「WIDE の活動として 1 時間費やす」労働の価値を表す媒体である。WIDE メンバは、WIDE Hour を互いに交換することができる。

すべての WIDE メンバが WIDE Hour の口座を持ち、その初期値はゼロである。口座は負の残高を許す。

WIDE Power:

次の計算式により求められる。

$$WIDEPower = \log \frac{income \times outlay}{|income - outlay| + 1} - penalty$$

WIDE Hour の意味づけについては IDEON/$(PG)^3$A WG 共同報告書で詳細に説明しているので参照されたい。

今回の実験では、WIDE メンバ間で分配するリソースに WIDE Hour を対応づけることはできなかったが、予約した食事を欠食することに対してペナルティを科した。また、プレナリにおける発言に対して、口座を集中管理する WIDE Hour 事務局から WIDE Hour を振り出すことにした。

2.6.3　実験環境

サーバ

WIDE Hour システムは、http://fran.sfc.wide.ad.jp/ にて、Web アプリケーションとして提供を開始した。

クライアント

参加者は Web ブラウザがあれば WIDE Hour を利用できるが、更に moCA で発行された WIDE メンバ証明書を組み込んだブラウザを利用することで、よりセキュアかつ簡便に WIDE Hour を利用することができる。

実験内容の公開と周知

1. 実験公開用のコンピュータを設置し、WIDE Power の番付を常時表示した。
2. また、合宿終了時点で WIDE Power のピーク値が最大だった参加者に特典を提供することをアナウンスした。
3. BOF におけるログとりなどの労働に対して、WG のチェアから WIDE Hour を贈るなど、使い方の提案を行なった。

他の実験との協力

SPEARS WG の実験と協力し、Web ページに公開された次の情報を取得して処理した。

1. プレナリでの発言者
   - 事務局から 5 WIDE Hours を贈った。
2. 予約した食事の欠食者
   - ペナルティ 1 を科した。

2.6.4 結果

リソースの分配という、本来の目的のためにはならなかったとしても、多数の合宿参加者に WIDE Hour に関わってもらうことができた。

表 2.1 は、合宿終了日である 9 月 11 日に取得した統計データの一部と、この報告書を執筆している 12 月 31 日に取得したデータを併記したものである。

表 2.1: 9 月 11 日 (合宿終了日) 時点と 12 月 31 日 (大晦日) 時点の統計

2003 年 9 月 11 日時点	
何らかの形で参加	161 名
総ログイン回数	530 回
証明書使用	406 回
SSL + パスワード	60 回
平文パスワード	64 回
総取引回数	361 回
無効化された取引	7 回
プレナリでの発言	68 回

2003 年 12 月 31 日時点	
何らかの形で参加	189 名
総ログイン回数	1058 回
証明書使用	872 回
SSL + パスワード	92 回
平文パスワード	94 回
総取引回数	649 回
無効化された取引	23 回

次の 2 点から、合宿にて実験を行なうことの意味があったと考える。

1. 合宿後も継続して取引を行なっている人々がいることから、WIDE Hour の概念を浸透させる上で役立ったと考えられる。

2. 合宿後、100 日以上経った後の数値上の伸びが高々 2 倍程度であるということは、通常なら 100 日以上かかることを、4 日で成し遂げたことを意味する。合宿という空間において集中してシステムを利用してもらうことには意味があり、今後も合宿において実験を行なうことでデータ収集上の効果が期待できる。

2.6.5 まとめ

現在、行なわれている WIDE Hour の取引の多くには次の問題があり、本来の目的である、協調メカニズムの検証を行なうまでには至っていない。

1. 価値が Hour の定義から無関係になっているという問題

2. 取引に実体が伴っていないという問題

継続的に取引機会の提供を行ない、価値観の形成と浸透に努めることにより、これらの問題の解決を図りたい。

2.7 Pluggable CpMonitor 実験

2.7.1 実験の目的

CpMonitor は、誰もが基本的なネットワークトラフィック観測をより手軽に行えることを目指し、我々が開発を続けているパッシブ型ネットワークモニタである。本合宿では、前回実験を踏まえ CpMonitor に今回追加された新機能の実環境における試験、および合宿参加者へのデモンストレーションを行う。また今後データ解析をすすめるためのトラフィックデータの収集も併せて行う。

2.7.2 実験の概要

前回実験で問題となったのは、ネットワーク状況・ユーザの行動によりダイナミックに変化する観測対象を、手動で予測・捕捉することの困難さであった。ここで我々は、スナップショット機能と呼ぶ機能を開発し、実装を行った。

スナップショット機能とは、観測対象 (アプリケーション・VLAN 等) をあらかじめ定めて行う既存の観測とは異なり、1 分間に観測インタフェース上で観測されたトラフィックを自動的に分類し公開する機能である。

この CpMonitor を合宿ネットワークに組み込み、対外線 (衛星および地上線)、および合宿地内の複数のセグメントと基幹ルータの間のトラフィックを観測・公開した。

その公開には 1 分ごとに更新される CGI スクリプトを用いた。

また合宿地内にマネジャを設置し、従来の方法である SNMP による CpMonitor エージェントからのデータ収集、Web および Java によるトラフィックグラフのリアルタイム公開も併せて行った。

2.7.3 実験環境

CpMonitor エージェントを 3 台、うち 1 台はマネジャ兼用として、合宿ネットワーク内に設置した。合宿ネットワークへの接続はタップキットを用い、観測対象とするネットワーク接続に影響を与えないよう留意した。観測対象としたのは以下の 3 リンクである。

- 地上線
- 衛星
- 合宿地内ネットワーク (GbE)

2.7.4 結果

本システムは合宿 1 日目より最終日まで安定して稼働した。ただし GbE インタフェースへの対応が十分でなく、合宿地内のモニタに関しては最終的に稼働させることができなかった。

参加者へのデータ公開の様子を以下に示す。

蓄積されたデータは以下の URL で現在も自由に閲覧することができる (http://www.sendai.wide.ad.jp/ kazuo/200309.CAMP/)。

2.7.5 まとめ

今回の実験により、CpMonitor のスナップショット機能の安定動作が確認された。

今後は、ユーザへの公開 UI、スナップショット機能により取得されるデータの蓄積について、開発を進めていく必要がある。

また、今後の合宿地ネットワークの広帯域化にあわせ、今回適用に失敗した GbE 環境への対応も進めていく必要があると考えられる。

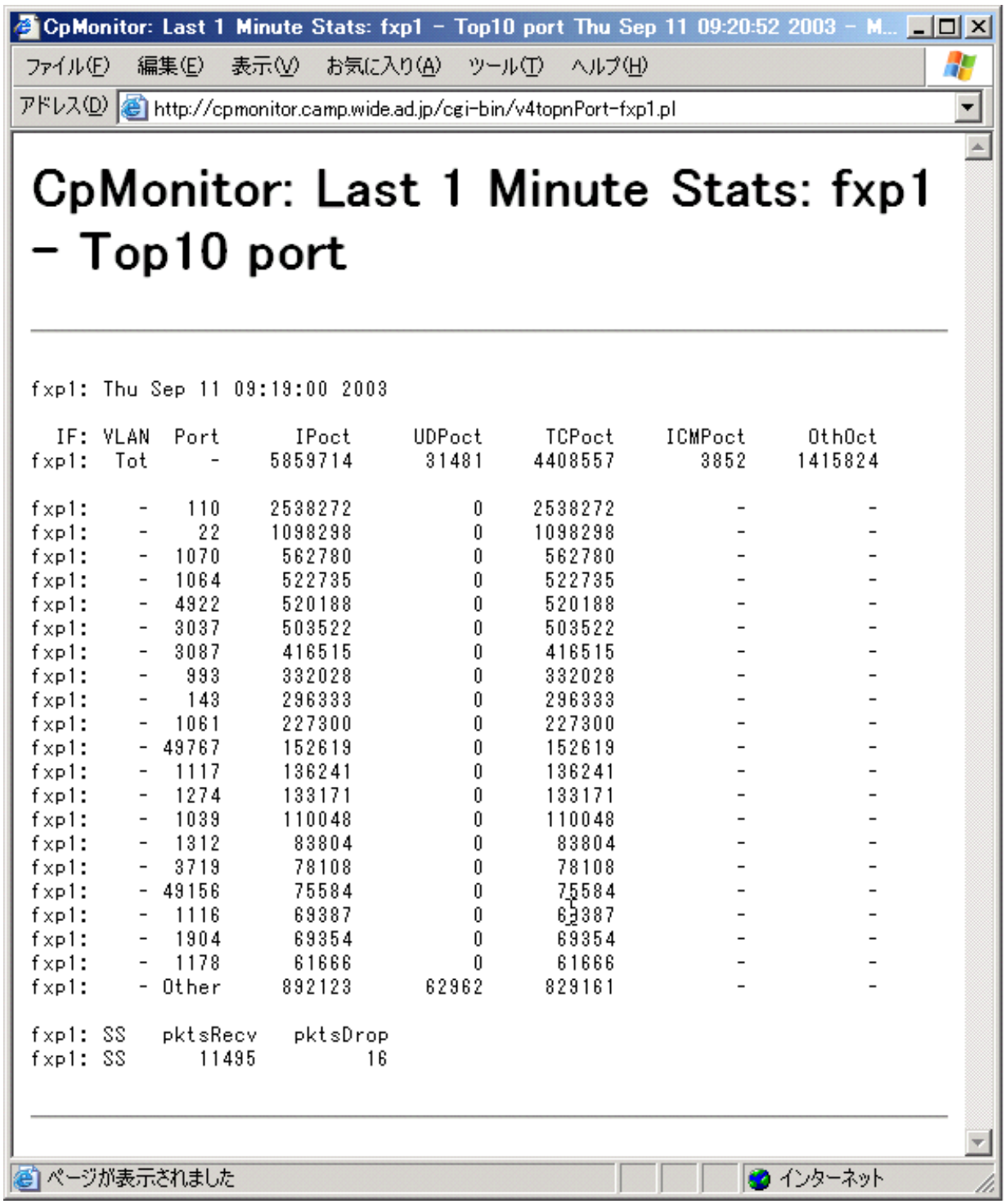

図 2.13: スナップショットの公開

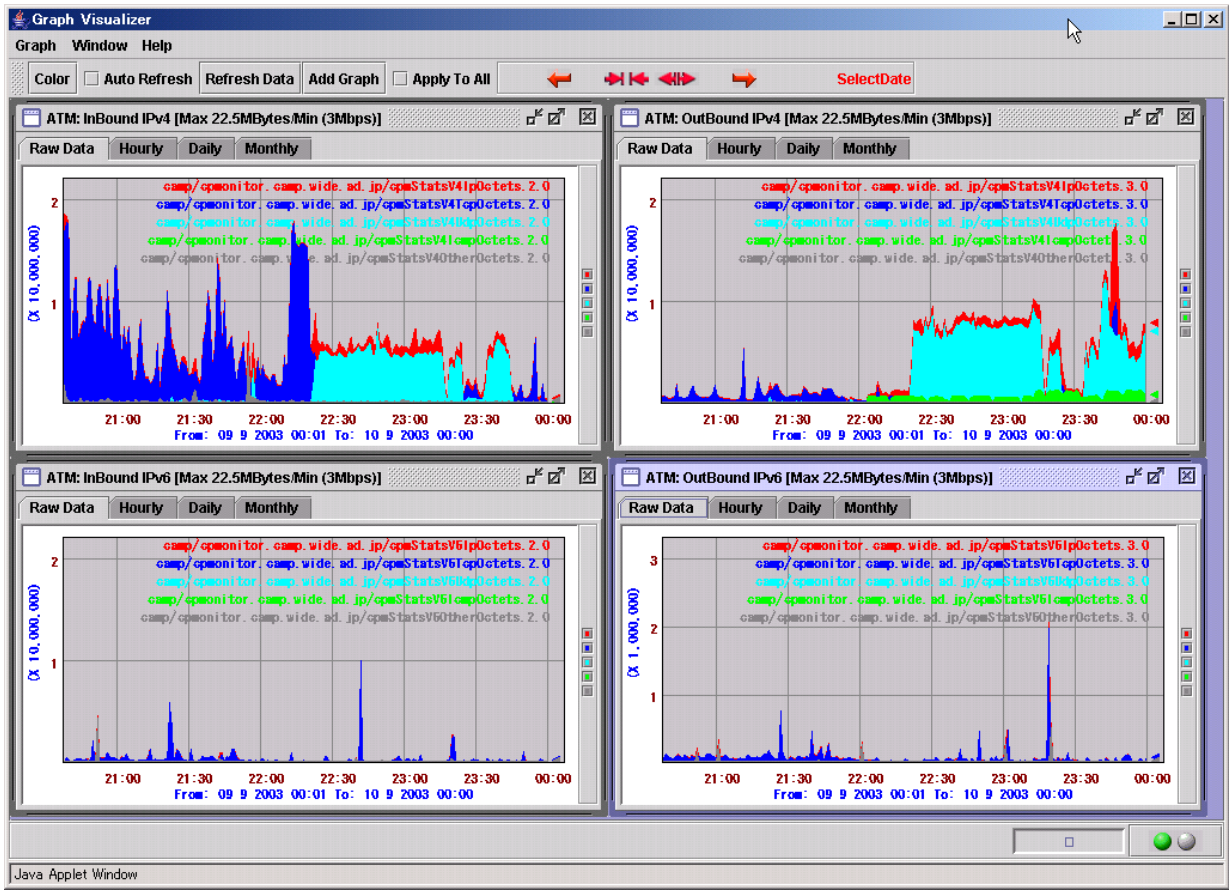

図 2.14: トラフィックグラフの公開