第 XIX 部

大規模な仮設ネットワークテストベッドの設計・構築とその運用

第 19 部
大規模な仮設ネットワークテストベッドの設計・構築とその運用

第 1 章　2000 年秋合宿実験ネットワーク

この章では、2000 年 9 月に行われた WIDE プロジェクト秋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結果を報告する。

1.1 ネットワーク構成

この章では、2000 年秋合宿のネットワークの全体構成を説明する。

図 1.1 は、合宿ネットワークのトポロジを示している。四角は、ルータあるいはホストを示す。線は、イーサネット、衛星回線、ATM のいずれかを示している。

図の上側から説明する。図の上側で WIDE インターネットとの接続している。「Nara」および「Fujisawa」と書かれている四角の波線は、WIDE 奈良 NOC および WIDE 藤沢 NOC に置かれている機器である。それ以外は、合宿地に置かれている。WIDE 奈良 NOC と合宿地は、衛星回線および ATM で接続した。WIDE 藤沢 NOC と合宿地は、衛星回線が 1 本と ATM が 2 本で接続した。複数の回線で WIDE NOC と接続している理由は、冗長構成を持

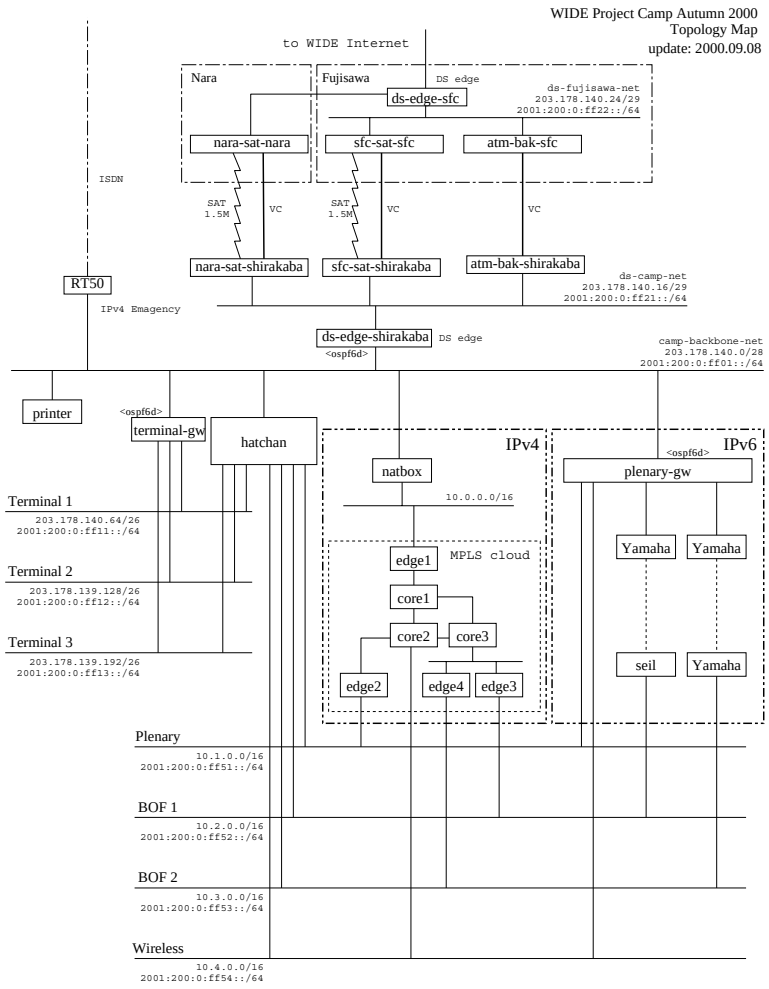

図 1.1.　2000 年秋合宿ネットワークトポロジ

つためである。これにより、ある実験が失敗した時でも、合宿地でのネットワークを提供できる。この冗長構成では、OSPF(IPv4 および IPv6) を利用して自動的にバックアップ回線を利用できるようになっている。合宿地と WIDE NOC の間では、「衛星ネットワークを含んだ首振り」実験と「ユーザからの要求による帯域割り当ての実現」実験が行われた。実験結果については、後の章で述べる。

IPv4 と書かれている波線四角内では、「MPLS スタック (AYAME) による MPLS の運用」実験が行われた。この実験は、IPv4 で行われた。この四角の下の横棒 (Plenary1, BOF1, BOF2, Wireless) は、イーサセグメントであり、合宿参加者が利用するセグメントである。

IPv6 と書かれている波線四角内では、「IPv6 対応 ISDN ルータの評価」、「PRI インターフェースつき IPv6 ルータの負荷試験」、「OSPFv3 の運用」が行われた。OSPFv3 の運用については、この四角内部だけではなく、図中で ospf6d と書かれている機器に載せられている。

「IPv6 での Mobility Support Protocol (LIN/6) 運用実験」実験は、ユーザセグメントでの実験が行われた。「NATPT (v6/v4 translator)」実験は、camp-backbone-net で実験が行われた。

1.2 合宿ネットワークを利用した実験項目

2000 年秋合宿では、以下の 8 つの実験が行われた。

- 衛星ネットワークを含んだ首振り
- ユーザからの要求による帯域割り当ての実現
- MPLS スタック”AYAME” による MPLS の運用
- IPv6 対応 ISDN ルータの評価
- PRI インターフェースつき IPv6 ルータの負荷試験
- IPv6 での Mobility Support Protocol (LIN/6) 運用実験
- OSPFv3 の運用
- NATPT (v6/v4 translator)

これら 8 つのうち、2001 年春合宿で実験を行っている「ユーザからの要求による帯域割り当ての実現」、「MPLS スタック”AYAME” による MPLS の運用」、「IPv6 での Mobility Support Protocol (LIN/6) 運用実験」については、後述の 2001 年春合宿報告でまとめて報告する。以下では、上記 3 つ以外の実験について報告する。

1.3 衛星ネットワークを含んだ首振りの実験

実験者: 安田歩

所属: 慶應義塾大学

1.3.1 目的

今回の合宿でも、対外線に衛星回線を利用した。衛星回線の運用は、衛星回線を利用して合宿地へネットワークの接続性を提供すると共に、普段小型可搬地球局の運用を行う機会が少ないため、合宿を通して衛星回線の運用技術の維持・向上を計ることを目的としている。

また、合宿地から対外線へのルータ、SFC から合宿地へのルータで、衛星回線と地上回線との経路選択を行う首振りの機能を実装し、動作評価を行った。

1.3.2 概要

衛星回線は、他のデータリンクと異なり片方向通信路である。今回は、片方向 2 波を組み合わせて双方向とし、帯域は、上り/下りとも 512Kbps で使用した。

衛星回線の部分のネットワーク図を、構築するための機材と共に図 1.2 示す。

1.3.3 衛星回線の構築

衛星回線を用いて接続性を提供するためには、合宿地側で小型可搬地球局を組み立てる必要がある。この組み立てには、アンテナの設置場所の選定から衛星捕捉して link up させるまでの作業が伴う。こ

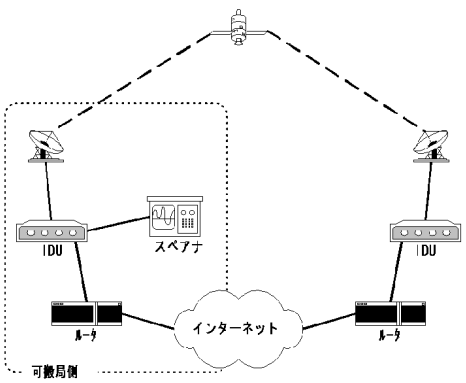

図 1.2.　衛星回線部分のトポロジ図

の作業には、細かい設置手順やトラブル対応能力など、経験が必要となる。

今回の合宿を通じて、衛星ネットワーク構築のためのノウハウを蓄積した。この結果は、次の URL で公開している。

```
http://pontoon.sfc.wide.ad.jp/vsat/
  manual/manual.html
```

1.3.4 首振りの実験と評価

本実験は、Diffserv の実験と協調して行った。Diffserv によりラベル付けされたパケットを、地上回線と約 500ms の遅延が発生する衛星回線の２つの回線の間での経路選択を行う。

首振りの機能は地上回線と衛星回線の双方へのインターフェースを持ったルータに実装した。これは、合宿地側、SFC 側一台づつである。

今回は、v4 のパケットのみを首振りの対象とした。v4 パケットの TOS フィールドを利用してラベル付けされたパケットを、ラベルに応じて出力インターフェースを選択する。この機能は、kernel 内の ip_output.c に実装した。結果、ラベルに応じてパケットを振り分けることができた。

1.4 IPv6 対応 ISDN ルータの評価

実験者: 木村俊洋

所属: ヤマハ (株) AV・IT 事業本部 技術開発本部 通信機器開発室

1.4.1 実験の目的

家庭や小規模オフィスで広く用いられている ISDN ルータに，IPv6 に対応させたファームウェアを実装し，動作の妥当性や安定性を検証する．

1999 年度の合宿では，合宿地と SFC を ISDN 回線で接続し，合宿ネットワークの対外接続環境を提供した．しかし，この実装においては，RIPng と IPV6CP の実装が不完全であったため，他の実装との接続が制限されるという運用上の問題があった．

そこで，今回の合宿では，RIPng を用いた動的な経路制御を可能にするとともに，IPV6CP の実装を整備して PPP によって他の実装と接続できるようにする．また，新しい試みとして，IPv6 に対応した IPsec の実装を追加する．これらの機能を搭載した ISDN ルータを実験ネットワークに組み入れることで，諸機能が良好に動作することを確認する．

1.4.2 実験の概要

RIPng と IPV6CP については，他実装との接続によって相互接続性を検証する．IPsec については，相互接続性の検証が可能な段階まで実装が成熟していないものと判断し，同じ実装を相互に接続することにする．IPsec の基本的な機能はすべて IPv6 に対応しているが，今回の実験では，IKE とトンネルモードの ESP を用いることにする．トランスポートモードと AH の評価については，ネットワーク構成と評価の煩雑さを避けるため，実施しないことにする．

1.4.3 実験環境

RIPng と IPV6CP の実験については，ヤマハの RT300i を使用し，IIJ の SEIL-T1 との間を 1.5Mbit/s の PRI 専用線のエミュレータで接続した．さらに，IPV6CP を起動し，PPP リンクの上で IPv6 パケットを送受信できるようにした．経路制御に関しては，静的な経路を設定せずに，完全に RIPng で運用することにした．

また，IPsec の実験については，ヤマハの RTA50i を 2 台用意した．これらのルータを，ISDN 回線のエミュレータによって接続し，トラフィックが存在するときのみ回線を接続するように設定した．また，Multilink-PPP により，トラフィックの負荷に応じて 64kbit/s と 128kbit/s のいずれかの回線速度を取るように設定した．

IPsec に関する設定としては，300 秒の間隔で鍵を交換するように設定し，IKE を除くすべての IPv6 パケットをトンネルモードの ESP で処理するように設定した．

各機材の配置については，RT300i と SEIL-T1 を，NOC と BOF 部屋 1 を接続する位置に配置した．また，2 台の RTA50i を，NOC と BOF 部屋 2 を接続する位置に配置した．

1.4.4 結果

3 日間の連続的な運用において，おおよそ良好に動作した．発見した問題としては，RIPng の動作において，記憶領域を正しく解放しないことが挙げられる．しかし，この問題については，期間中に修正して正常化することができた．

IPV6CP については，正しくインターフェース識

別子 (Interface Identifier) を交換できることを確認した．また，その識別子にもとづいて，IPv6 アドレスを自動生成し，正しくインターフェースに付与することを確認した．

RIPng については，1 つの UDP パケットに過剰な数の経路エントリを載せて送信したために，パケットがフラグメントされる問題があったが，これについても期間中に修正することができた．

なお，3 日間を通じて，ルータが処理した IPv6 のトラフィックは，ISDN 回線や PRI 専用線の帯域を埋めるほど多くはなかったため，より高負荷な環境で評価をする必要がある．

1.4.5 まとめ

ISDN ルータに IPv6 に対応させたファームウェアを実装し，RIPng，IPV6CP，IPsec などの機能について，良好に動作することを確認した．今後の課題としては，使いやすさや安定性を主眼として作り込むことが挙げられる．また，既存の IPv4 ネットワークに対して容易に導入するためには，トランスレータなどの移行技術を開発する必要があると考えられる．

1.5 IPv6 対応 PRI ルータによる対外接続実験

実験者: 萩野純一郎

所属: IIJ 技術研究所/KAME プロジェクト

IIJ では、KAME IPv6/IPsec スタックを搭載したルータ、SEIL-T1 を開発している。参考までに、SEIL-T1 のハードウェア諸元は以下のとおりである。

CPU: Hitachi SH-3 7709A、CPU クロック 133MHz

RAM: 32Mbytes

ROM: 2Mbytes
(実際には 4Mbytes モデルもある)

OS: NetBSD 1.5

インターフェース: 1.5Mbps 専用線
(PRI インターフェース)、
10Base-T

今回の合宿では、SEIL-T1 を利用して合宿地内の IPv6 パケット配送およびルーティングをし、以下の検証を行った。なお、IPv4 のルーティングは別のルータが担当している。

- IPv6CP および IPv6 over PPP の相互接続試験
  PRI で接続された対向の Yamaha RT300i との間で、IPv6 PPP 接続を確認する。
- RIPng の相互接続試験
  Yamaha RT300i との間で、RIPng による経路交換を確認する。
- SEIL-T1 の安定度検証
  SEIL-T1 は多数の IPv6 ユーザが居る環境でも安定して機能するか検証する。

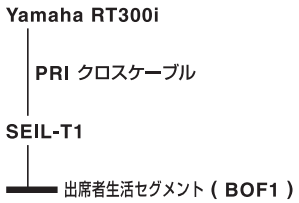

図 1.3.　SEIL-T1 周辺のネットワーク図

ネットワーク図を図 1.3 に示す。SEIL-T1 と Yamaha RT300i は PRI クロスケーブルで接続されている。なお、PRI インターフェースの結線およびコネクタ種別は機種によって異なっており統一されていないので、クロスケーブル作成の際には注意を要する。

SEIL-T1 は合宿期間中一度も再起動せず稼働し続けた。ただし、再設定などのために手動で再起動した場合は除く。RT300i との相互接続性も layer 2、layer 3 ともに全て良好であった。経路制御には RIPng を使用したが、こちらも良好に動作した。

T1 実回線での試験、IPv4/v6 dual stack のルーティングができなかったのが残念である。今回の試験では、トラフィック量などの統計情報を十分に記録できなかった。次回以降は統計情報を残すよう気をつけたい。

1.6 OSPFv3

実験者: 小原泰弘

所属: 慶應義塾大学

1.6.1 概要

IPv6 の AS 内経路制御をより安定したものにするため、WIDE Project では IPv6 をサポートした OSPF の開発と運用を行っている。

2000 年 9 月に行われた WIDE 合宿では、運用経験の蓄積、問題点の発見と、開発へのフィードバックを目的として、OSPFv3 を利用した IPv6 ネット

ワークの運用実験を行った。

合宿内に構築されるすべてのネットワークにおいて、OSPFv3 を利用した IPv6 到達性を提供した。合宿のネットワークは実験的要素が強いため、通常のネットワークと比べると特殊な構成になっていることが多く、比較的、安定した経路制御を提供することが難しい環境であると言える。そのような環境で OSPFv3 の運用実験を行うことによって、実装上の問題点・運用上の問題点などを発見し、解決した。

結果として、3 日間に渡って安定した良好な通信環境をユーザに提供することができた。

1.6.2 実験環境

WIDE 合宿での実験ネットワーク構成を図 1.1 に示す。図 1.1 中、「to WIDE Internet」から、WIDE Backbone に接続されている。WIDE Backbone では 1999 年に IPv6 IGP として OSPFv3 を採用し、定常運用を続けている。WIDE の IPv6 Backbone は WIDE 6Bone と呼ばれ、最大 13 ホップ、33 台のルータから構成されるネットワークである。今回の実験では、WIDE 6Bone で定常運用している OSPFv3 ドメインに接続するかたちで、合宿ネットワークの IPv6 経路制御を行った。OSPF システムとしては、WIDE 6Bone と合宿ネットワークを併せて一つのシステムであるため、以下の評価はすべて 合宿ネットワークを含む WIDE 6Bone 全体に対して行った。

表 1.1. 統計環境

統計採取時間	72 時間
制御された経路数	143 経路

表 1.2. LSA 更新時間

	平均
Router-LSA	1560 秒
Network-LSA	1723 秒
Intra-Area-Prefix-LSA	1706 秒

表 1.3. SPF 計算時間・間隔

最大 SPF 計算時間	66 ミリ秒
平均 SPF 計算時間	25 ミリ秒
最大 SPF 計算間隔	15 時間
平均 SPF 計算間隔	13 分

1.6.3 運用時の統計

統計は、OSPFv3 の実装にプログラムを埋め込んで採取した。

1.6.4 ネットワークの安定度に関する考察

LSA 更新時間の平均 (表 1.2) を見ると、OSPFv3 プロトコルで決められている LSA 更新時間の最大である 1800 秒に十分に近くなっている。

ここから、

1. 本実装が無駄に LSA を生成しない、プロトコル仕様に沿った実装であること
2. 実験ネットワークという、構成の変化が頻繁に起きるネットワークでもプロトコル仕様が効率的であること

がわかる。

また、SPF 計算間隔 (表 1.3) から、

1. ネットワーク構成の変化があるまで無駄に SPF 計算をしないこと
2. 構成変化が頻繁に起きる環境でも、平均 13 分に 一度 SPF 計算しなおすという良好な安定度がプロトコル仕様によって得られること

がわかる。

1.6.5 CPU load に関する考察

OSPF プロトコルで一番負荷が高い処理は、SPF 計算だとされる。SPF 計算時間と SPF 計算間隔の統計 (表 1.3) から、平均 13 分毎に 25 ミリ秒計算時間に費すだけなので、OSPFv3 のプロトコル仕様および本実装が十分に負荷が低いことがわかる。

1.7 NATPT (v6/v4 translator)

実験者: 藤澤慎一

所属: 横河電機

1.7.1 目的

今実装を進めている NAT-PT は IPv6 → IPv4 のプロトコル変換機能に加えて、IPv4 → IPv4 の通常の NAT としての機能も持っている。 両方の機能は同時に使うことが可能だが、普段大量のトラフィックが流れる環境で動かしていない。そこで 大量のトラフィックが流れる WIDE 合宿の場を借りて、トランスレータの耐久性と、IPv6/IPv4 混在の環境でトランスレータとして機能することを確認する。

1.7.2 概要

トランスレータに対して 内側を IPv6/IPv4 混在のセグメントとし、外側をインターネット側とする。内側からの IPv6 のパケットは 基本的に何もしない。必要があれば IPv4 パケットに変換する。 内側からの IPv4 パケットは無条件に NAT (含むポート番号変換) をする。

IPv6 パケットを IPv4 パケットに変換する/しない の判断は、IPv6 パケットの持つデスティネーションアドレスの先頭 96 ビットをみて行う。 ある特定の 96 ビットの値を持っていれば IPv4 パケットに変換する。

1.7.3 構築

今回はこちらの事情で実験のトポロジーを決める会議にほとんど参加できず、結果として IPv4 → IPv4 NAT を主とするトランスレータとしての動作確認となった。

キャンプネットバックボーンに NAT-PT 箱を置く。 その下に MPLS 群があり、ユーザーセグメントが繋っている。 このトポロジーでは ユーザーセグメントからのパケットは IPv4 パケットしか流れないので、NAT-PT 箱と MPLS との間に小さなセグメントを儲けてそこに IPv6 host を置き、IPv6 → IPv4 変換の試験を行った。

1.7.4 実験と評価

前述したように 今回の実験は IPv4 → IPv4 の大量のトラフィックを流した時の動作確認が主となってしまった。 こちらの動作では 若干のメモリリークがあったが、合宿期間中はなんとか持ち堪えてくれた。 このメモリリークは合宿終了後に修正されている。

IPv6 → IPv4 のプロトコル変換はネットワークの構成の制約からほとんど使われなかったが、小規模に実験した分については 大量の IPv4 → IPv4 トラフィックがある時に IPv6 → IPv4 のプロトコル変換がちゃんと動作することが確認できた。 ただし この実験では IPv6 のアドレスをどう解決するかという問題を避けて、直接 IPv6 アドレスを使って実験を行った。

この合宿期間中に実装を追加した機能が二つある。

- この合宿を始めるまで IPv4 のフラグメント化されたパケットは処理していなかった。 しかし合宿の途中でフラグメント化されたパケットを中に通す必要が出てきたので、その機能を追加した。
- MPLS の実験との絡みで、NAT の内側の MTU を 1500 より小さくする必要があった。そこで コマンドラインから MTU を設定し、その値より大きなパケットが来ると ICMP_UNREACH_NEEDFRAG を返すようにした。

両方とも合宿期間中は意図したとおりに動いてくれた。

ただし、MTU に関しては IP ヘッダの DF bit を立てて送ってきているにも関らず、ICMP_UNREACH_NEEDFRAG を返してもパケットの大きさを小さくせず送ってくる サイトがあり、結果として そのサイトと MPLS の内側のマシンと通信ができなかった。

第 2 章　2001 年春合宿実験ネットワーク

この章では、2001 年 3 月に行われた WIDE プロジェクト秋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結果を報告する。

2.1 ネットワーク構成

この章では、2001 年春合宿のネットワークの全体構成を説明する。

図 2.1 は、2001 年春合宿ネットワークトポロジを示している。この図において、中程の点線より上側は WIDE 藤沢 NOC (Fujisawa) に置かれた機器、下側は合宿地 (Nishiura) に設置された機器を表している。

この合宿では、「安定した通信環境」というポリシを持ってネットワーク構築を行った。「安定した通信環境」は、実際にはネットワークの冗長性と、トラブルチケットの運用によって実現した。

図 2.1 中、特に記述の無い点線は、通常時は利用しない非常用の回線として用意されたものである。ま

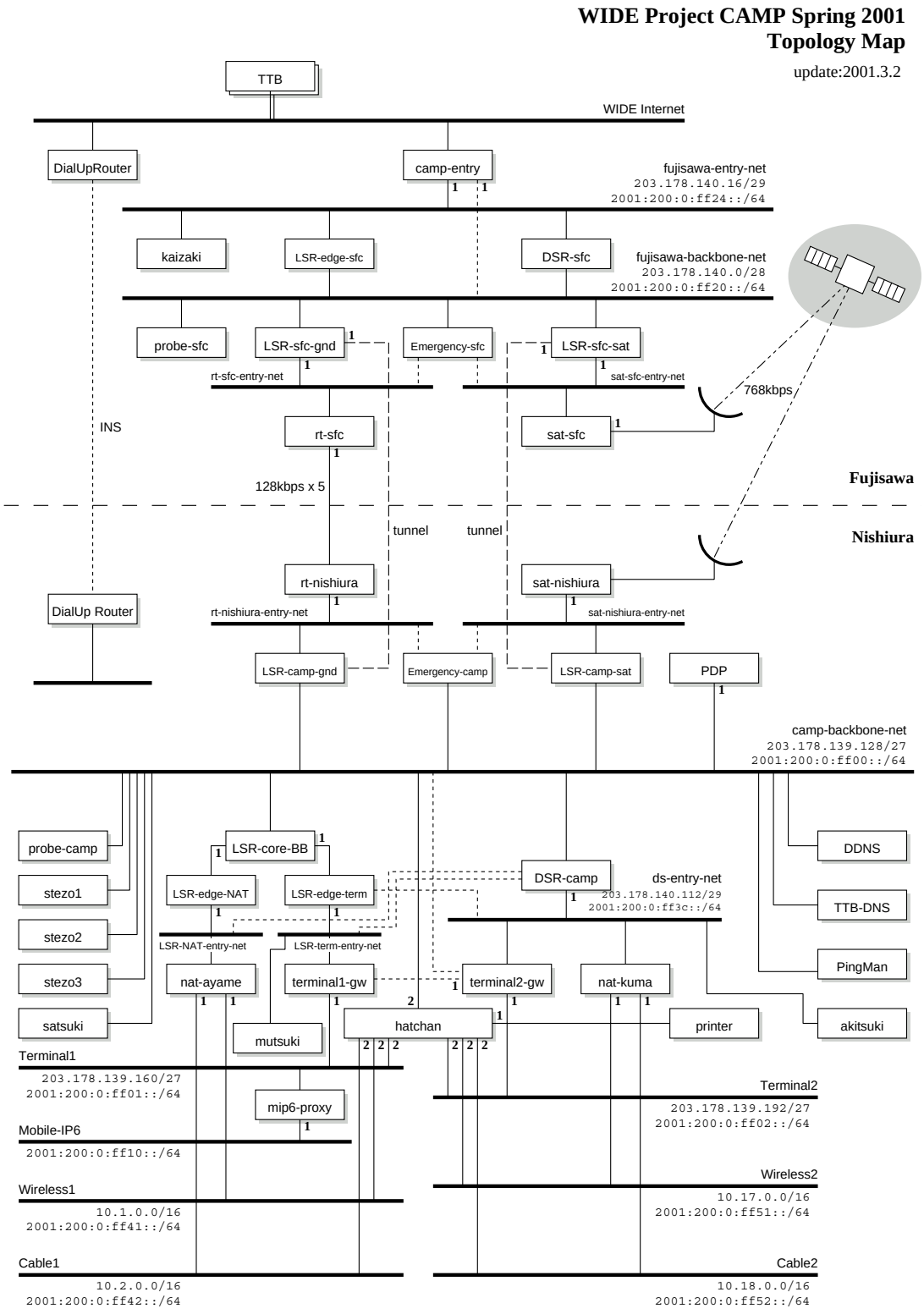

図 2.1. 2001 年春合宿ネットワークトポロジ

た、図 2.1 中左の 「INS」と記述された点線および両端の「DialUpRouter」も、対外線が全く接続できなかった場合のために用意した。合宿中、これらの回線を利用するような障害は起きなかった。

合宿ネットワークの藤沢側と合宿地側は、768Kbps 双方向の衛星回線および、128Kbps の専用線 5 回線を用いて接続した。実験をするための要望として、「対外線は各ルータから一つの回線として見える必要」があったため、専用線 5 回線は Yamaha RT300i を利用して PPP/MP[146] を行い、さらにその両端で IP over IP tunneling 技術 [53] を用いることにより、一つの回線として提供した。

DNS、DHCP など、出来るだけ実験に影響されないサービスを提供するため、図中「hatchan」から全てのユーザセグメントに接続し、これらのサービスを提供した。

2.2 合宿ネットワークを利用した実験項目

2001 年春合宿では、以下の実験が行われた。

- Dynamic Update を利用した名前登録
- MIP6/LIN6
- 分散型ネットワーク管理情報収集システムおよびネットワーク管理情報の可視化システムの試験提供
- IPv6/IPv4 トランスレータ「TTB」による IPv6/IPv4 トランスレーション実験
- 分散監視のためのメカニズムと視覚化の有効性の確認
- MPLS/L3TE/Diffserv/ネットワーク制御

2.3 Dynamic Update を利用した名前登録

実験者: 石原知洋
所属: 慶應義塾大学

2.3.1 実験の目的

独自の名前更新クライアントを利用して、DNS Dynamic Update を実際に運用する。その統計的情報を集め、Dynamic Update の実用的な運用のために必要な事柄の分析を行う。

2.3.2 実験の概要

DNS 実装の一つである bind9 における Dynamic Update の機能を利用し、合宿参加者の PC に割振られた IP アドレスの正引きおよび逆引きレコードの登録を行う。登録は独自の名前更新クライアントで行い、サーバ上で名前の登録および問い合わせに関する詳細なログを取る。

2.3.3 実験環境

合宿ネットワーク内に Dynamic Update を行うサーバを設置する。そして、通常の合宿ネットワークのドメインの下に Dynamic Update によって各 PC のレコードを登録するドメインを作成する。正引きは IPv4 と IPv6 のアドレス両方を登録し、逆引きに関しては IPv6 アドレスのみ登録を行う。

2.3.4 実験結果

17 台の PC がこのこのサービスにより名前の登録を行った。各レコードの更新は短期間に集中して行われる傾向にあった。また、登録されたレコードの問い合わせも短期間に集中して行われる傾向にあった。

名前登録を行った数人にヒアリングをしてみたところ、アドレスが変わっても登録されるかどうかを何回か試してみたという人が多かった。また問い合わせに関しても、実際に登録されているかを何度か試すという人が多かった。

2.3.5 考察

以上の結果から、実際に観測された Burst 的な登録および問い合わせはユーザーが試験的に使ってみた結果の産物で、実際の運用でこのような傾向が得られるとは考えづらいものであった。

また今回の合宿ネットワークは無線のセグメントを大部分の部屋で同一にしていたため、あまりアドレスがつけ変わるという状況にならず、これによる有り難みは薄れていた感があった。

これらの問題はあったものの、実際の運用では大きな問題が発生することなく、正常なサービスを提供することができた。

2.3.6 今後の課題

今回使用したクライアントは FreeBSD、OpenBSD、NetBSD で動作したが、Windows など、さらに多くの OS で動作するようにしなければならない。また、さらに厳密な運用のため TSIG の導入も行う必要がある。加えて、今回は十数人程度の

ユーザーしか利用しなかったため、スケーラビリティの問題が表面化しなかったが、今後もっと大規模な環境で使用した場合の実験および考察は必須事項である。

2.4 MIP6/LIN6

実験者: 國司光宣, 小柴晋, 渡里雅史
所属: 慶應義塾大学
実験者: 松岡保静
所属: 東京工業大学

2.4.1 実験の目的

本実験では, IETF で標準化が進められている Mobile IPv6 (MIP6), また MIP6 での問題点を解決することが期待されている Location Independent Networking for IPv6 (LIN6) という 2 つの移動透過保証プロトコルを実際のネットワークにおいて実用性を評価した．また，本実験では MIP6/LIN6 それぞれの通信性能に関する比較をするとともに，これらのプロトコルが必要となるアプリケーションについて考え，移動透過保証プロトコルの普及を目指した．

2.4.2 実験の概要

以下に示すようなアプリケーションを MIP6/LIN6 が実装されている計算機上で動作させデモを行った．

- IPtel
  BoF 部屋を移動する移動ノードとターミナルルーム内に設置した通信ノード間で VoIP の会話を行う．BoF 部屋を移動した際，有線の場合にはネットワークセグメントが代わるため IP アドレスの変化が生じるが，移動透過プロトコルにより会話を継続できることを確かめる．
- ページャー
  IPtel と同様に，BoF 部屋間を移動する移動ノードとターミナルルーム内の通信ノード間で文字ベースのチャットを行う．
- Mobile Web server
  BoF 部屋間を移動している移動ノード上で apache Web server を起動し，通信ノードから移動する Web server に対してアクセスする．Web server から提供されるデータは，VAIO C1 のカメラを利用して撮った画像である．この画像を定期的に更新することにより，移動ノードが Web 中継を行うことが可能となる．

本実験を体験するためには MIP6/LIN6 が実装がなされていなければならなかったため，IPtel とページャーに関しては限られたホストでしか動作しなかった．しかし，Mobile Web server に関しては MIP6/LIN6 が実装されていないホストのために Web proxy server を用意し，proxy server を介して接続することにより移動ノード上の Web server で提供される画像を閲覧可能とした．

2.4.3 実験環境

本実験の実験環境を図 2.2 に示す．

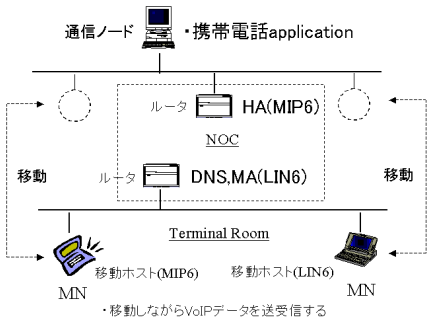

図 2.2. MIP6/LIN6 の実験環境

図 2.2 では，MIP6/LIN6 の実験環境を模式的に表しており，各ノードがどのように配置されていたかを示している．MIP6 の Home Agent (HA) や LIN6 の Mapping Agent (MA) は NOC 内に配置し，閲覧用の Web client や IPtel などの通信ノードはターミナルルームに配置した．移動ノードは BoF 部屋間を移動し，Web server や IPtel 等のアプリケーションサービスを提供した．

2.4.4 考察

本実験に関係する今回の合宿ネットワークトポロジの特徴として，有線は BoF 部屋ごとに違うサブネットであったのに対し，無線は 1 つの BoF 部屋を除き無線のローミング機能を用いることにより，全て同じサブネットとして運営を行っていた．MIP6/LIN6 の利点を体験するためにはサブネット間の移動が不可欠であり，今回の合宿ネットワークトポロジは，サブネット間の移動は無線でネットワークに接続している限りほぼ起こらなかったと考えられる．また，有線を利用する人よりも無線を利用する人が多いこと

は事前に予想されていたため，実際にサブネット間移動をした人の割合はごく少数であった．したがって，今回の実験ではサブネット間を移動した際の問題をあまり感じる人は少なかったはずである．

しかしながら，アンケート結果では MIP6/LIN6 を利用したい人々の割合は利用したいと思わない人の 3 倍程度であり，将来の普及が期待されていることが分かった．今後の課題としては，より MIP6/LIN6 が必要となるような環境で実験をし，より多くのプラットホームでの開発をする必要があると思われる．さらに，MIP6/LIN6 に関するキラーアプリケーションを深く考えることも必要である．

2.5 分散型ネットワーク管理情報収集システムおよびネットワーク管理情報の可視化システムの試験提供

実験者: 土井一夫
所属: 株式会社サイバー・ソリューションズ

2.5.1 実験の概要

近年のネットワーク環境は大規模化・複雑化が進んでいる。このことはすなわち、既存の一極集中型管理モデルの限界が近くなってきていることを示している。この状況に対応するべく今回データ収集プログラム自体の高機能化とともに、分散型ネットワーク管理情報収集システムのプロトタイプを構築し、 2001 年春合宿ネットワークにおいて実証実験を行った。

また、情報公開インターフェース上で利用者にまで分散を意識させてしまうことがないよう、統一的インターフェースも用意して公開した。

2.5.2 ネットワーク情報収集

1. 分散型情報収集
   今回用意した機器は表 2.1、表 2.2、表 2.3 の三台である。
   この三台に表 2.4、図 2.3 で示す範囲のネットワーク管理情報収集を行わせた。
   観測には SNMP を用い、その対象は以下の通りである。

   - MIB-II (interfaces, ip, tcp, udp, snmp)
   - IPv6MIB (ipv6IfStatsTable)

表 2.1. プローブ (A) 兼 indexServer の機器構成

CPU	PentiumIII 850MHz
メモリ	256M
OS	Solaris 8

表 2.2. プローブ (B) の機器構成

CPU	Geode 300MHz
メモリ	128M
OS	FreeBSD 4.2-RELEASE

表 2.3. プローブ (C) の機器構成

CPU	Geode 300MHz
メモリ	128M
OS	FreeBSD 4.2-RELEASE

表 2.4. ネットワーク管理情報収集の範囲

プローブ (A)	SFC・NOC
プローブ (B)	LSR 側
プローブ (C)	DSR 側

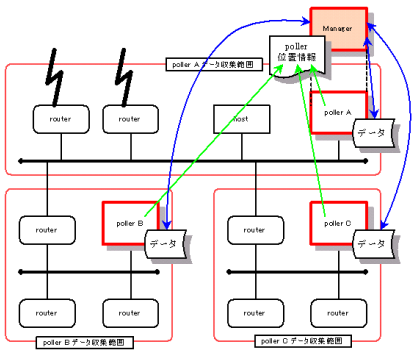

図 2.3. 分散自律型データ収集システムの概要

観測結果を表 2.5、表 2.6、表 2.7、表 2.8 に示す。期間中ほぼ安定して取得が行えた。
分散をしたことの利点として、負荷・トラフィックが分散されていることがここに示されている。また、プローブを対象の比較的近くに存在させられるようになったことから、途中のネットワーク状況に左右されにくいデータ収集が可能になり、安定取得に一役買ったと考えられる。
取得されたデータの例として、合宿地からの対外線、衛星側と地上側のトラフィック比較を図 2.4、図 2.5 に示す。

表 2.5. 観測期間、観測間隔

観測期間	3/4 16:00 – 3/7 11:00
観測間隔	60 秒

表 2.6. プローブ (A) の収集結果

取得対象ホスト数	13 個
1 回あたり取得インスタンス数	692 個
総取得回数	1,043,917 回
総データ量	17,070,080 バイト

表 2.7. プローブ (B) の収集結果

取得対象ホスト数	8 個
1 回あたり取得インスタンス数	344 個
総取得回数	474,239 回
総データ量	7,587,840 バイト

表 2.8. プローブ (C) の収集結果

取得対象ホスト数	5 個
1 回あたり取得インスタンス数	250 個
総取得回数	389,979 回
総データ量	6,154,240 バイト

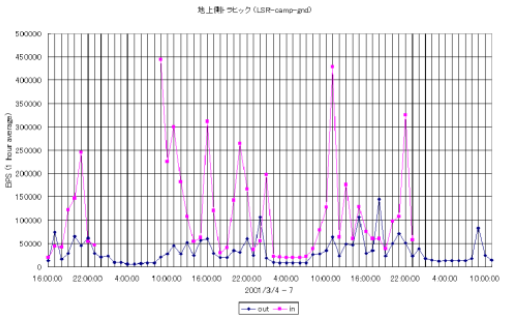

図 2.4. 地上側トラフィック (LSR-camp-gnd)

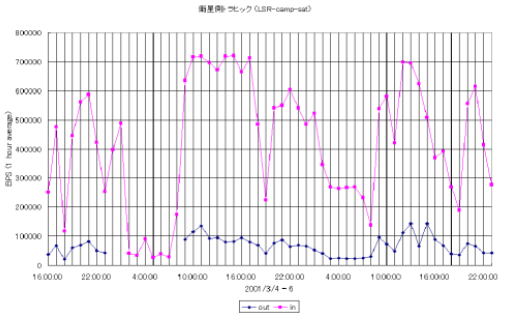

図 2.5. 衛星側トラフィック (LSR-camp-sat)

2. 高効率型情報収集

データ収集に関し、以前から引き続き取り組んでいる管理トラフィック自体の低減であるが、今回は以下の三つの面からの機能を実装した。

- 分散データ収集手法によるトラフィックの局地化

  前述の通りである。データは遠くまで様々なネットワーク機器に影響を与えながら移動することはなく、すぐ近くのプローブに保存される。
- パケット数の削減

  ペイロードの総量が同じであるなら、パケットの数を少なくしてペイロードの割合を増すことにより、ヘッダの分通信量を圧縮できることになる。この考えに基き、1 パケット中に多くのリクエストを詰め込めるようにした。プローブの発生した SNMP パケットの数のグラフを図 2.6 以下に示す。前項の取得データ量と併せて見れば、そのパケット数がかなり少なく抑えられていることがわかる。
- 追加トラフィック無しの情報取得 (RTT)

  データ取得をした際、そのラウンドトリップタイム (RTT) を記録するようにした。これにより、追加トラフィック一切無しでそのリーチャビリティや途中のネットワーク状況を知るために貴重な情報を取得できるようになり、間接的にトラフィックあたりの情報量を増加させることができた。

  合宿ネットワークから仙台 NOC までの RTT グラフを図 2.7、図 2.8、図 2.9 に示す。

3. 自律型情報収集

増加・複雑化するネットワーク管理対象に対し、以前のように手動もしくはそれに近い形での監視対象の設定をすることは非常なコストを伴い、事実上不可能になりつつあると言える。

今回は Web 上に公開されたアドレス一覧をプローブが各々で取得し、それぞれの監視対象を自動的に選びだし、SNMP アクセスの有無を確認した上でデータ取得設定を最終的に生成するシステムを構築した。

また、その設定情報を後述する「indexServer」に通知することで、統一的公開のために必要な

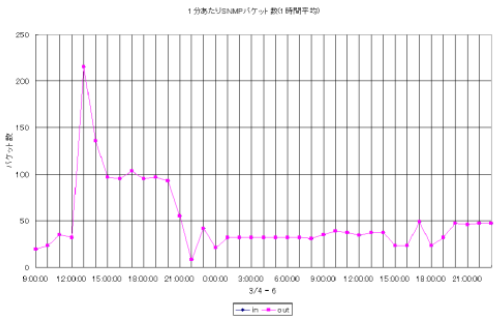

図 2.6.　1 分あたり SNMP パケット数 (1 時間平均)

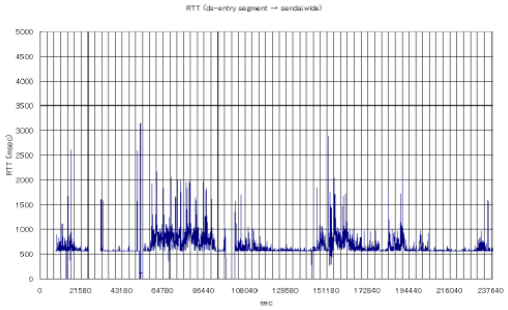

図 2.7.　RTT (ds-entry segment → sendai.wide)

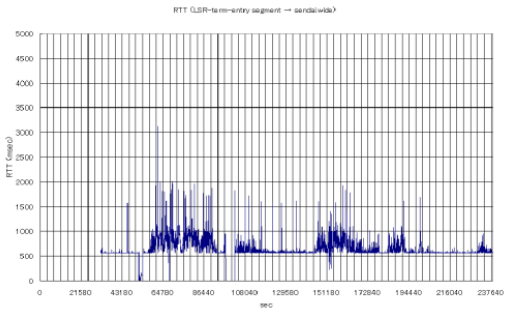

図 2.8.　RTT (LSR-term-entry segment → sendai.wide)

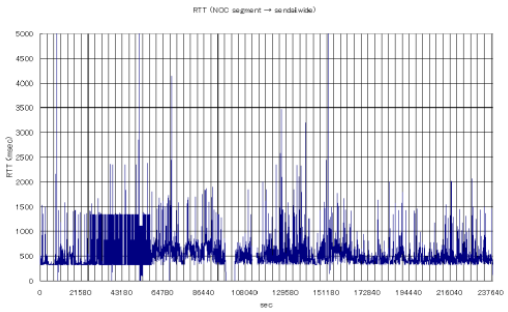

図 2.9.　RTT (NOC segment → sendai.wide)

情報 (データがどこに存在するかという位置情報) を自律的に構成することを可能にした。

2.5.3 ネットワーク情報公開

以上で説明したように、今回のシステムは情報を各所に分散した形で保持するシステムである。このシステムを使って情報公開を行う際の主な問題点は以下の二点である。

- 情報がどこに保存されているかをどうやって調べるか
- 分散することによってユーザに対し新しい知識や操作を要求することにならないか

これらの問題点を克服するためにとった方法を以下に示す。

1. 情報の位置把握のために
   公開された分散情報の位置管理のみを司るサーバ (indexServer) を用意し、各プローブは自身の取得・管理している情報の一覧としての情報収集設定を indexServer に通知する。
   ユーザの要求に応じて起動される情報公開用ユーザインターフェースは indexServer に蓄えられた情報を取得し、情報をどこから得られるか把握することができる。
2. 統一的公開のために
   ユーザの要求に応じて起動される情報公開用ユーザインターフェースは、前述の indexServer の働きにより直接プローブと通信し、データの供与を受けることができる。
   これによりユーザは分散環境を意識することなく、情報を取得することが可能になる。

2.5.4 評価

ユーザインターフェースの実証的評価のため、合宿期間中にアンケートを実施した。

好意的意見は「見やすい」「おもしろい」「是非使いたい」というものがあり、否定的意見としては「動作が重い」「説明が足りない」「トラフィックグラフから何がわかるかわからない」「Java 環境が無い」ということであった。

主眼とした「分散を意識させないインターフェース」は受け入れられていることが示されている。

データ取得は大規模に、しかも総じて安定して行え、分散システムとしての基礎的な考え方の有効性

はこの実験を通して実証されたと考えている。

2.5.5 今後の課題

データの取得自体は安定して行える仕組みが整いつつある。これからはアンケートでも指摘されているように、MIB–II による基礎的なトラフィック情報のみに拘ることなく、必要な情報とは何かということを充分吟味し、それを何如に公開すればよいかということに主眼を移すべきであると考えられる。

また、分散させたことによるメリットは以上に挙げたものだけではなく、ユーザ環境により近いところでのきめ細かいネットワーク情報収集とその提供などもある。これらの利点を充分生かしていく必要があるだろう。

2.6 IPv6/IPv4 トランスレータ「TTB」による IPv6/IPv4 トランスレーション実験

実験者: 尾添靖通

所属: 横河電機株式会社 IT 事業部 IT プロダクト事業センター IPv6 グループ

2.6.1 実験の目的

TTB (The Transition Box) は、IPv6 から IPv4 への移行期において必要な IPv6/IPv4 トランスレーション環境を構築することが可能である。今回の大規模な仮説ネットワークベットにおいて TTB のトランスレーション機能、TTB の DNS サーバ機能が有効である事を実証する。また TTB の DNS サーバ機能により複数の TTB トランスレーション・エンジンを監視し、トランスレーション環境全体に冗長性を持たせる事が可能である事を実証する。

2.6.2 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。1 台の TTB DNS サーバに 2 台の TTB トランスレーション・エンジンを監視させ、TTB トランスレータの稼動状況により、DNS クライアントに返すダミー・プレフィックスを変更させる。クライアントが使用する TTB トランスレータが振り分けられた事を確認するためにログを収集するログサーバを用意し、TTB トランスレーション・エンジンのログを収集する。

TTB トランスレータ、TTB DNS サーバの説明

- TTB トランスレータ
  IPv6/IPv4 トランスレーションを行う。Transport Relay Translator 方式を採用している。トランスレーションのエンジンは複数並べることが可能である。
- TTB DNS サーバ
  DNS Proxy のような働きを持つ。TTB DNS サーバはクライアントからの名前解決に対し、DNS サーバに問い合わせ A レコードが返ってきた場合は、ダミー・プレフィックスを付けてクライアントに返し、クライアントに TTB トランスレータ経由で通信を行わせる。DNS サーバから AAAA レコードで返ってきた場合は、そのままクライアントに AAAA レコードを返し、TTB トランスレータを経由せずに通信をさせる。
  また TTB の持つ DNS サーバの付加機能で上記の TTB トランスレータを 5 台まで監視することが可能で、有効なトランスレータにアサインされているダミー・プレフィックスを付けて DNS クライアントに返す。その結果、IPv6 クライアントからの通信は、複数のトランスレータに分散される。

2.6.3 実験環境

図 2.10 のように TTB トランスレータ (TTB1、TTB1) を SFC 側、TTB DNS サーバ (TTB-DNS) を合宿側に配置し、ログサーバで TTB トランスレータのログ収集を行う。TTB-DNS は、TTB1、TTB2 を 1:2 の割合で監視する。それによって、名前解決時にそれぞれに割当てられたダミー・プレフィックスが、1:2 の割合でクライアントに返され通信が分散される。

2.6.4 結果

合宿期間中、連続稼動運転を行った。その結果 TTB トランスレータのプロトコル別の使用状況と転送データ量はそれぞれ図 2.11、図 2.12 となった。プロトコルの特性から、リクエスト数は HTTP が一番多

表 2.9. TTB トランスレータ分散状況

TTB トランスレータ	コネクション数
TTB1	1540
TTB2	3116

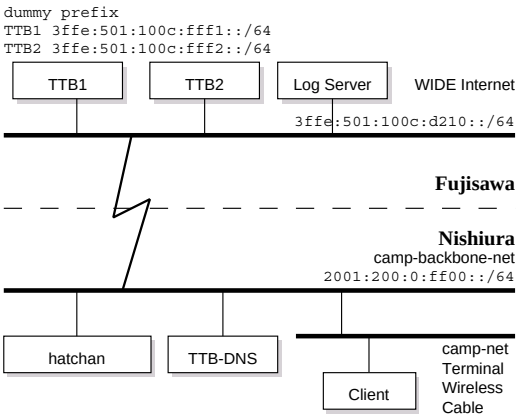

図 2.10.　TTB ネットワーク構成図

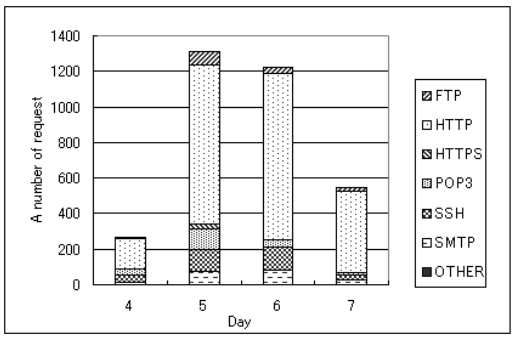

図 2.11.　TTB トランスレータプロトコル別使用状況

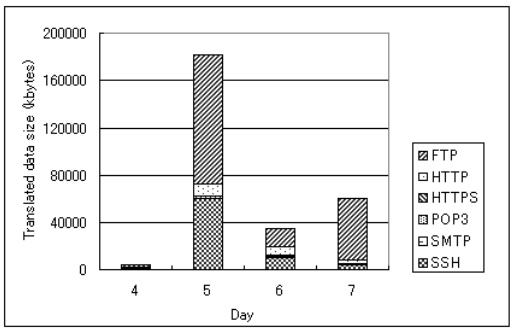

図 2.12.　TTB トランスレータ転送データ量

く、データ転送量は FTP、SSH が多い結果となっている。

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 2.9 となった。TTB1 と TTB2 の使用状況が 1 : 2 の割合だったことがわかる。

2.6.5 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 を用いたシステムが安定動作することを確認できた。また、表 2.9 の結果から TTB DNS サーバのトランスレーション・エンジン監視機能により、トランスレーション環境に冗長性を持たせることが可能であることも確認できた。

2.7 分散監視のためのメカニズムと視覚化の有効性の確認

実験者: 藤原敏樹，鈴木茂哉
所属: 株式会社フォア・チューン

2.7.1 目的

大規模ネットワークを分散監視する手法を実装し、実際の WIDE 合宿における、計測・監視に応用してみることで、その有効性を確認する。

また、観測結果を視覚化 (Visualize) することで、観測データの持つ傾向が把握しやすくなることを実証する。

2.7.2 システムの特徴

本システムは、以下のような特徴を持つ。

- 分散、自律・自立型プローブによる、継続的モニタリング
- 各種データ収集: アクティブ (SNMP poll、DNS、HTTP、POP)、パッシブ (EGP/IGP)、 イベント (syslog, SNMP trap)
- 収集された複数のデータへの、時間をキーとした、統一的アクセス
- 複数のデータを時系列を合わせて重ね合わせ
- Management Resource Locator (MRL) による管理対象の指示の簡易化。スカラ値を、簡単に指定

```
// Node @ Region / ItemSpecifier / Index
```

- 複数のインターフェースの値を配列として指示
- 目的に応じてページのカスタマイズが可能
- Web ベースのビジュアライザ — Static グラフ (要素間の演算が可能) の動的生成と、JavaApplet による Dynamic グラフを用意。
- 自律分散型 Alert
- MRL による値の指定
- プローブ間が分断されていても警告できるメカニズム

2.7.3 構成

自律形プローブを SFC 側と Camp 側に配置。Camp 側の Visualizer で表示 (図 2.13)

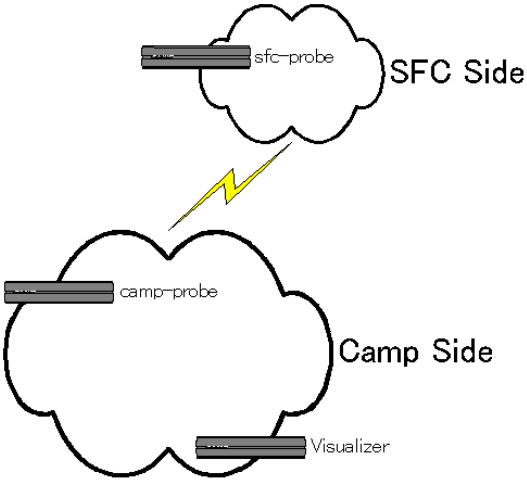

図 2.13. システム構成図

2.7.4 実験の考察

- データを保存できて、後々視覚化できることが有効であることが確認できた。
  データは今でも残っているので、たとえば、地上回線と衛星回線流量の振り分けの加減を検証できる。
- プローブを分散配置し、監視を中央集約型で行うことが有効であることが確認できた。
  しかし、実験中は FC と合宿サイト間の接続が切れることはなかったので、プローブが自立であることの有効性は発揮できなかった。

2.8 MPLS/L3TE/Diffserv/ネットワーク制御

実験者: 穎原桂二郎、小川晃通
所属: 慶應義塾大学
実験者: 森島直人、横山輝明
所属: 奈良先端科学技術大学院大学
実験者: 宇夫陽次朗、宇多仁、小柏伸夫、
末永洋樹、宮地利幸
所属: 北陸先端科学技術大学院大学

2.8.1 実験の背景

インターネットはもともとは構成要素が独立して緩やかな協調分散を行う系として設計・運用されてきた。しかしその成功によるユーザ数の増加や多種多様なアプリケーションの登場は、インターネット自体の変化を要求する圧力ともなっている。特に、リアルタイム性や帯域制御といった設計時に考慮されていないネットワーク特性を求めるネットワークアプリケーションや、DDoS といったネットワークの一部だけでの処理では対応しきれない攻撃の登場によって、広域ネットワークを線または面で捉えた『包括的な協調分散制御』の必要性が考察されつつある。

ネットワークを包括的に制御するためのネットワークモデルとして転送層と制御層の分離モデルが提案されている。転送層はパケット転送を実際におこなう部分で Diffserv (Differentiated Services)[16] や MPLS (MultiProtocol Label Switching)[140]、L3TE (Layer 3 Traffic Engineering) などがこれに含まれる。制御層は管理ドメイン内でのポリシに応じて転送層の制御を行う。

2.8.2 実験の目的

このような背景のもとで、今回の合宿では包括的な広域ネットワーク制御の検討を目的とした実証実験として、ユーザからの動的な要求に対するネットワーク資源予約システムを構築した。このような細かな粒度でのネットワーク制御は、既存の第 3 層での (主に経路制御技術によって行われる) 制御では扱いづらく、さまざまな技術的チャレンジが必要である。実験ネットワークの管理ドメインは、合宿ネットワーク内部だけでなく対外接続点である SFC に設置されたルータを含んでおり、広域ネットワークにおける制御に関しても考察している。

本実験は、WIDE 内でアクティビティを持っている『あやめプロジェクト』[70] と『くまプロジェクト』[73] による共同プロジェクトである『くまあやめプロジェクト』によって行われた。あやめプロジェクトは、第 3 層以下技術に着目したネットワーク制御に関する議論を背景に、MPLS の研究と開発を行っている。また、研究開発用 MPLS 環境として、MPLS ルータ (LSR) および MPLS シグナリング機構である LDP (Label Distribution Protocol)[8]/CR-LDP (ConstraintRouting LDP) の設計および実装を行っている。くまプロジェクトは、多様なポリシを多様な方法で実現するためのアーキテクチャ、コンポーネント間のインターフェース、ポリシ履行状況確認のための計測、ネットワークプロビジョニングのため

の計測など、広域ネットワーク制御を実現するためのフレームワークの研究を行っている。また、制御層として COPS(Common Open Policy Service)[17] を、転送層として L3TE を実装し、実証実験を行っている。

2.8.3 実験の概要

本システムは、さまざまな技術が相互に関連して動作している。以下に利用している技術とその説明を示す。

制御層

- COPS/資源分配/課金/認証
  ユーザからの動的な予約受付および、ネットワーク資源の分配、課金、認証に従ったアドミッション制御を行う。課金には『WIDEUnit』と呼ばれる仮想通貨を導入し、予約の制御および平等性を実現する。さらに、アドミッション制御の結果にしたがってネットワークの制御パラメータを決定する。このパラメータは COPS によって制御の対象となる機器に送信される。
  今回のネットワークは転送層に L3TE と MPLS といった異なる技術で実現される複数の転送層を利用している。このため単一の制御層から特性に合わせて複数の転送層を制御する必要がある。
- NAT-friendly End-to-End communication
  NAT を利用したネットワークでは、アドレス変換の前後で同一のフローに対して観測されるフロースペックが異なり、End-to-End 性が阻害される。このため、Diffserv や MPLS などのフローの識別を本質とするサービスの提供が困難となる場合がある。そこで、NAT の変換表を動的に取得し、アドレス変換の前後のフロースペックを対応づけることによって、単一のアドレスに変換された複数のフローを識別する。

転送層

- MPLS/CR-LDP
  合宿ネットワークからインターネットまで LSP (Label Switching Path) を確立し、要求に応じた通信品質を保証する。MPLS のシグナリング機構として代表的なプロトコルは LDP であるが、LDP は IP 層の経路を MPLS 層に写像するために用いられるため、トラフィックエンジニアリングには向かない。そのため、特定の制約に従った LSP を確立するために CR-LDP プロトコルを用いた。
- L3TE
  現在の IP 層の次ホップ検索機構では、フローごとに異なる経路を与えることができない。そこで、送信元アドレスやポート番号等を次ホップの検索メトリックとして利用できるように拡張し、ユーザの要求に応じた経路を選択できるようにする。
- Diffserv
  IP 層での Diffserv および MPLS/Diffserv を利用して、合宿ネットワークから対外線を含んだ帯域的な DS ドメインを構成した。各構成ルータはパケット中の DSCP (Diffserv Code-Point)[113] に応じて、あらかじめ設定されている挙動制御を行う。

図 2.14 にシステムの概要を示す。

ユーザは予約を発行する際、Web サーバにアクセスする。このとき、User DB に登録されているユーザ情報によって認証する。予約は (src, dst) などのパラメータを Web ページのフォームに投入することによって行う。

投入された予約は PDP (Policy Decision Point) により、課金や資源の状況にしたがってアドミッション制御される。予約が受理された場合、各構成ルータの

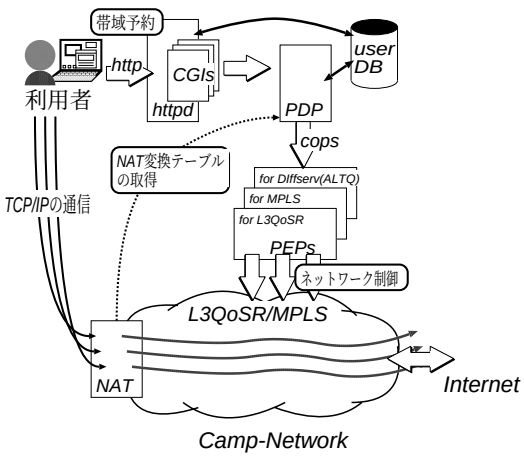

図 2.14.　システムの概要

制御パラメータを決定し、PEP (Policy Enforcement Point) に送信する。PEP では、転送層にしたがって制御パラメータを変換し、機器の設定を行う。

MPLS 網では、PEP の指示に従い、CR-LDP をもちいて LSP を確立する。また、L3TE では PEP によって経路を設定する。どちらの場合もユーザからのトラフィックは Diffserv 的に識別されて処理される。

2.8.4 結果

このように多数の技術を導入して生成されたネットワークであったが、合宿期間中特に問題なく動作し、実験の狙いである『広域ネットワークにおける柔軟な制御』が実現可能であることを実証できた。

2.8.5 まとめ

本実験を通してネットワーク制御に関するさまざまな知見が得られた。これらは論文としてまとめられる予定である。