第 XXXVI 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 36 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　2004 年春合宿ネットワーク

本章では 2004 年 3 月 15 日(月)から 18 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 Project 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 1.1 中の左右に引かれた 2 本の太い点線のうち、上側の点線よりも上部が対地として用いた慶應義塾大学湘南藤沢キャンパス（以降、SFC)、下側の太い点線よりも下部が合宿地である。四角はルータおよ

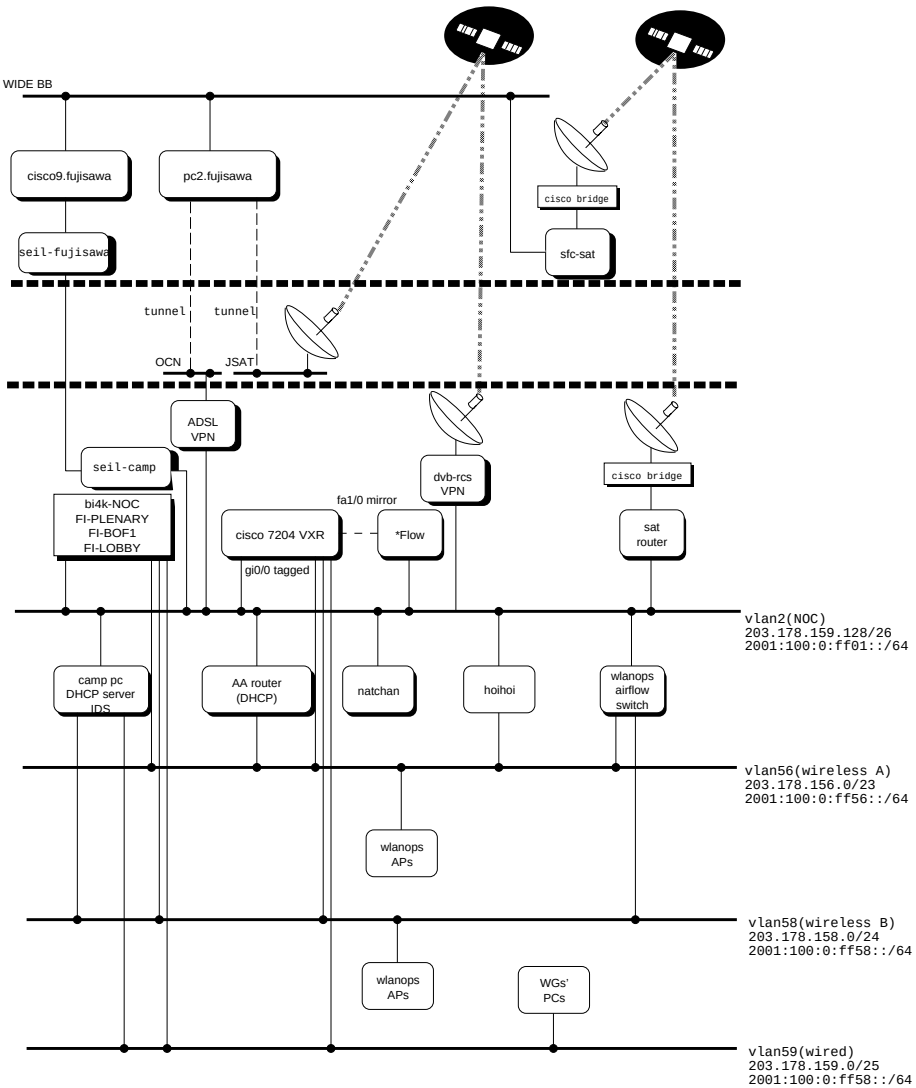

図 1.1.　2004 年春合宿のネットワーク構成

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 2 種類および衛星回線 2 種類の計 4 種類の回線を用いた。地上線として 1.5 Mbps の専用線（デジタルリーチ 1500）および 12 Mbps サービスの ADSL（フレッツ ADSL モア）を用い、衛星回線として Ku バンド（上り 512 kbps、下り 1.5 Mbps）および DVB-RCS を用いた。ADSL に関しては、NTT 局舎から合宿地までの距離が 5.2 km で伝送損失が 45 dB であった。

図中点線は IP-IP トンネルによる接続を示している。フレッツ ADSL のプロバイダには OCN を用いた。wide-bb との接続のために pc2.fujisawa.wide.ad.jp とトンネル接続を行った。また DVB-RCS は基地局が JSAT にあるために同様にトンネル接続を行った。

4 種類の対外線の使い分けは以下の通りになる。

- 3/15（合宿初日）
  ADSL はリンクが定常的に上がるかわからなかったため、ステートレスのトラフィックを流す方針とし、pop、imap、imaps を流した。衛星回線は DVB-RCS を積極的に用いる方針から、Ku を用いず DVB-RCS のみを用いる方針とした。DVB-RCS も実効性能がわからなかったため遅延の大きさがその使用感にあまり関係しない http、https のみを流した。それ以外のトラフィックに関しては、専用線を用いた。
- 3/16
  前日のポリシですでに専用線および ADSL の利用可能帯域を使いきってしまった一方で、DVB-RCS の実効帯域が広いことがわかったので、遅延が大きく影響する ssh のみ専用線に流し、ADSL を pop、imap のみにし、それ以外の専用線に回していたトラフィックを DVB-RCS に回した。
- 3/17〜3/18（最終日）
  前日のポリシにおいて http、https が DVB-RCS を回っていたが、トンネル接続をしているために MTU サイズが小さくなり、一部のホームページが閲覧できない状態になったため、Ku に http、https を回した。

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。

本合宿では ADSL と DVB-RCS の 2 つの新規に利用した対外線の実効性能が、実際に利用をしてみなければ分からなかったため、合宿ネットワークサービスとして計測を行いその情報を合宿ネットワーク利用者に公開した。この計測は、対外線のほかに合宿ネットワークのアドレス利用率などの計測を行った。

また、本合宿ではネットワークにとって有害となるパケットを送出するノードの検出を目的として IDS を導入した。2003 年 9 月に行われた前回の合宿では合宿参加者の利用ホストが MSBLAST に代表される無意味なトラフィックを大量に発生させるコンピュータウィルスに感染し、一時的に接続を失うことがあったためで、IDS の導入はそれらの事故を未然に防ぐための措置である。IDS の詳細に関しては次節に記す。

1.2 IDS の運用

近年、OS のセキュリティホールを利用し、メールなどを介せず直接マシンに感染するコンピュータウィルスが相次いで出現している。これらのウィルスはマシンに感染した後、他ホストへ感染するための通信を開始する。個人の所有するマシンの高性能化にともない、ウィルスの感染活動のためのトラフィックは無視できないものとなり、LAN 内のネットワークをダウンさせてしまう場合がある。また、WIDE の管理下であるネットワークからウィルスの感染活動が外部へ行われることは、組織のもつ信頼性の問題へとつながる。

表 1.1.　対外線の使い分け

	専用線	ADSL	DVB-RCS	Ku
3/15	その他	pop、imap、imaps	http、https	使用せず
3/16	ssh	pop、imap	その他	使用せず
3/17、18	ssh	pop、imap	その他	http、https

本合宿では、IDS を用いて合宿ネットワーク内においてウィルス感染活動を行うホストを監視した。

感染者発見後は迅速な対応を可能にするため、自動的に対処が働くよう設定した。IDS はウィルス以外にも多数の攻撃を検知することが可能であるが、誤検知の可能性が少なくないため、定型的な通信を行うウィルスに限定した。本合宿において検知するよう設定したウィルスは、CodeRed、Nimda、Slammer、MSBlaster（Nachi）の計 4 種である。これらのウィルスが内部から内部、あるいは外部に向けた送信を検知するようにした。

1.2.1 感染者発見後の対処

感染者が現れた場合、対外接続あるいは無線の基地局との接続を切るという処置もできる。しかし、感染ホストの使用者が感染による切断ではなく、ソフトウェア、あるいはハードウェアの障害だと考えると、トラブルシュートのためにネットワークへの接続を試み続ける。その場合、長時間ネットワークに接続されることとなり、膨大なトラフィックの送信によってネットワークが不安定になり、最悪の場合、ダウンしてしまう可能性がある。

この問題に対処すべく、本合宿では慶応義塾大学村井研究室で運用されている「悪い子ほいほい」というシステムを導入した。IDS によりウィルスによる感染活動を発見した後、当該ホストを「悪い子」として、dhcp サーバの設定ファイルに、その MAC アドレスを記録する。設定ファイルには当該ホストに対してアドレスを割り当てるが、default gateway は通常使用されるものではなく、camp-net で用意した「hoihoi マシン」のアドレスに設定する。（図 1.1 の “hoihoi” がこれにあたる）このサーバは全ての通信の転送を許可しないため、悪い子として登録されたホストは同一セグメント以外には接続できなくなる。ただし、TCP の port 80 番については transparent proxy を使用し、全ての http request を hoihoi マシンの http サーバに向ける。hoihoi マシンではすべてのリクエストに対して、「あなたのマシンはウィルスに感染している恐れがあります。至急ネットワークから切断し、camp-pc、あるいは camp-net にご相談ください」という旨のメッセージを表示する。これにより、ウィルスが感染した恐れのあるホストに対して、明示的にそれを伝え、早急にネットワークから切り離してもらうことで、ネットワークにかける負荷を少なくすることが期待される。

1.2.2 IDS 監視結果報告

本合宿において、IDS は初日の合宿開始時から camp-net の撤収まで監視を行った。しかし、今回において監視対象としたウィルスに感染したという検知結果は発見されなかった。合宿中、過度のトラフィックにより通信が阻害された、などの報告がなかったこともこの結果を裏付けている。

この要因としては、参加者の意識の向上、および各組織における指導の徹底がなされたためと考えられる。過去の WIDE 合宿において数回にわたり合宿中にウィルスに感染するというインシデントが発生したため、それによる教訓からパッチの適用、Personal Firewall の導入が行われるようになったことが推測される。また、最近発表されたソフトウェアの脆弱性を利用したウィルスが発生していなかったことも挙げられる。通常のウィルス対策としては Personal Firewall の導入およびパッチの適用を並行して実施するべきだが、パッチの適用のみである PC も実際は少なくない。パッチの適用をしていれば以前に発生したウィルスの感染は防げるが、新種のウィルスに対してはその限りではない。幸い、合宿近辺で、新しい強力な感染力をもつウィルスは発生しなかったため、感染者がいなかった可能性が高い。

1.2.3 まとめ

今回の WIDE 合宿においては IDS を運用し、ウィルスの感染者を即時的にネットワークから切り離すシステム「悪い子ほいほい」を導入した。本合宿においては、ウィルスの感染者は発見されなかったためその有効性を示すには至らなかった。しかしこれは新種のウィルスが発生していない、といった時期的な要因が絡んでいたと推測されるため、今後もこのようなシステムを導入することを提案する。

1.3 MAC アドレス認証を利用した個人認証とアクセス管理についての実験

1.3.1 目的

MAC アドレス認証を利用しアクセス管理を行う、という提案システムの有効性を確認する。また、他の認証技術、運用技術との連携を含めたシステムの汎用性を検証する。

1.3.2 概要ならびに実験環境

AA の MAC アドレス認証つき DHCP サーバは 203.178.156.0/23 の IPv4 アドレス空間に対して、主に証明書を用いた認証つき IPv4 アドレス割り当てを行った。

1.4 2004 年春合宿での ENUM/SIP デモンストレーション報告

ENUM WG では、2003 年 9 月 WIDE 合宿での ENUM/SIP デモンストレーションに引き続き、2004 年春合宿での公開実験として、SIP/VoIP/ENUM の理解・普及促進を目的に、ENUM/SIP、インターネット電話についてのデモンストレーションを行った。

本実験の詳細と結果は、ENUM WG 報告書を参照されたい。

1.5 GLI システムデモ

1.5.1 目的

GLI システムは、インターネット自動車 WG などにおいて車両の地理位置情報の管理に使用しながら、システムの開発を行ってきた。GLI システムを今後普及させるための手段の 1 つとして WIDE Project 春合宿でシステムとサンプルアプリケーションのデモンストレーションを行うこととした。今回のデモンストレーションにより、WIDE Project において GLI システムを認知向上させ、開発・研究やサーバの運用実験に協力してくれる方を増やして、今後の WG 設立の一助とする。

1.5.2 概要

本デモンストレーションでは、実車を使用した GLI システムお試しキット、GLI システムのアプリケーションとして横浜市営バスを使用したプローブ情報システムの展示を行う。

お試しキット

GLI システムを体験できるキットとしてお試しキットを作成した（図 1.2）。お試しキットでは、FreeBSD4.x が導入されインターネットに接続可能な PC と市販されている GPS を使って、GLI に参加して自位置を登録・検索できる。GPS は、NMEA フォーマットでデータを出力し、RS232C で接続できるものならどれでも対応している。ソフトウェアは GLI システムの登録クライアントと検索クライアントを同梱する。検索クライアントでは、自位置が画面上にプロットされ確認することができる。お試しキットを利用するほかのユーザの位置も検索できる。デモンストレーションでは、後述する車両の LAN に PC を接続し IPv6 で GLI の登録送信を行い、合宿現地では検索クライアントを動作させて、車両の位置を表示させた。

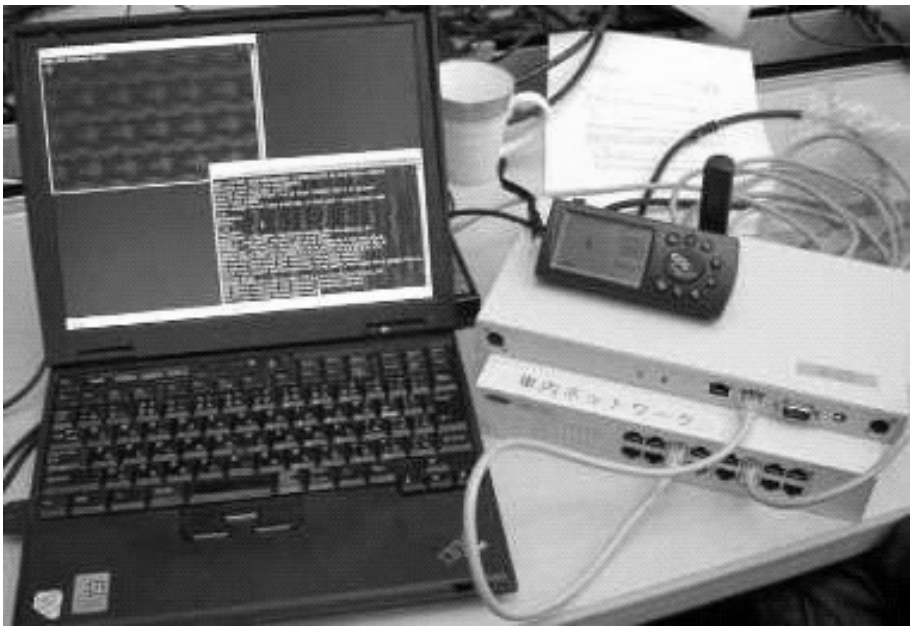

図 1.2.　お試しキット

横浜市営バスを使用したプローブ情報システム

GLI システムを利用したアプリケーションとして、Java と Flash を使用した Web アプリケーションを作成した。このアプリケーションでは、実際に走行しているバスを GLI システムで検索して表示し、その画面上のバスアイコンをクリックすることで、バスの現在のプローブ情報を取得することができる。プローブ情報は、バスの前方に設置してあるカメラからの画像、速度、ウィンカー、地理位置情報である。

システム構成を図 1.3 に示す。バスからは定期的に GLI サーバにバスの識別子と地理位置情報が、車両情報サーバにプローブ情報が送信される。Web クライアントから Web サーバにアクセスがあり検索要求があると、Web サーバは GLI サーバを検索してバスを地図上に表示し、バスのプローブ情報へのリクエストがあると、Web サーバは車両情報サーバに問い合わせてプローブ情報を取得して表示する。実行画面を図 1.4 に示す。

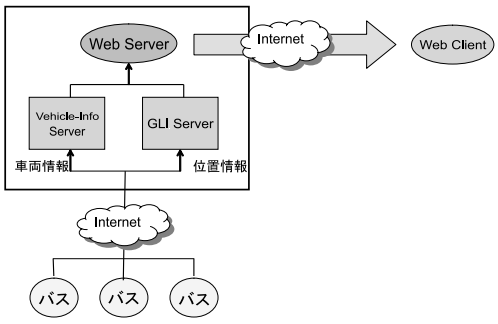

図 1.3. システム構成

1.5.3 実験環境

実験は、インターネット自動車 WG で開発、使用されている NEMO を実装したモバイルルータを使用する。実験ネットワーク環境を図 1.5 に示す。これを実験車両およびバスに搭載し、車両稼動時は b-mobile という PHS データ通信メディアを経由して、128 Kbps の通信速度で常に IPv6 で接続されている。バスには、他に Web カメラとデンソー製車載機が搭載されており、速度センサ、ウィンカーと接続され、これらによる情報を SNMP で取得可能である。

1.5.4 結果

実際の車両を使用してのお試しキットのデモンストレーションは正しく動作し、その間もサーバは安定して動作していた。また Web アプリケーションは作りこみが足りなかったために、多数のクライアントからのリクエストに対応できていなかった。

1.5.5 まとめ

多くの方に説明を聞いてデモンストレーションを見ていただいた。またプライバシ保護、分散管理、普及についてコメント・意見を頂いた。そのほか、お

図 1.4. 実行画面

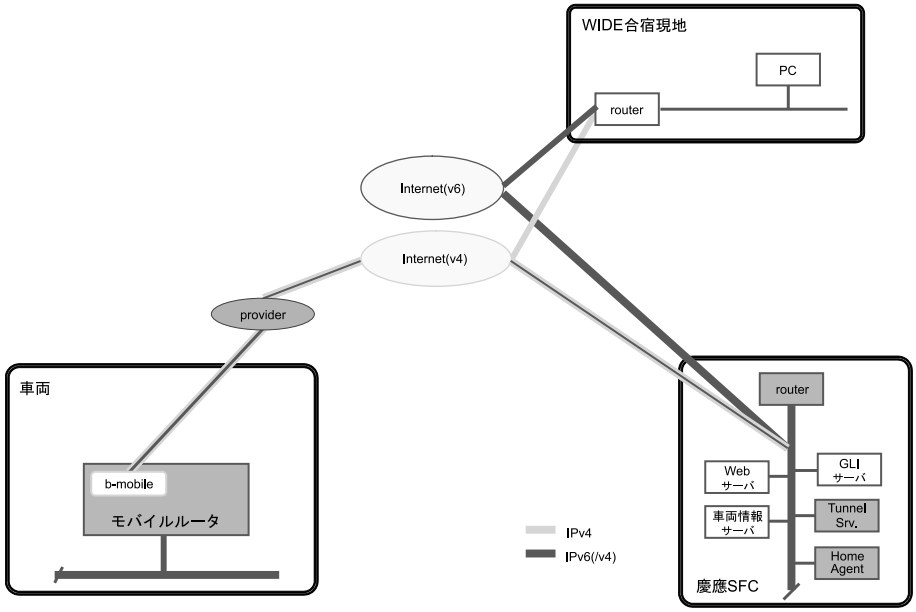

図 1.5.　実験ネットワーク環境

試しキットの利用要望や GLI サーバの分散管理への参加要望があった。今後は、ソフトウェアリリース、アプリケーション開発環境の構築、GPS 以外の位置情報入力手法の取り込み（携帯電話、IC タグ、無線 LAN、etc.）、地理位置マルチキャスト・位置情報自体のマスキングによる個人情報保護などについても取り組んでいく。

1.6 DHT による LIN6 の MA 探索に関する実験

1.6.1 目的

現在の LIN6 では、移動ノードの位置情報管理を移動ノードの識別子に対応する位置管理サーバで管理している。この位置管理サーバを Mapping Agent と呼び、移動ノードの識別子と Mapping Agent の IP アドレスの対応は DNS によって管理している。我々はこのしくみに以下の問題があると考えている。

- 移動ノードの識別子は非構造であり、非構造な識別子に対応する情報の管理に DNS は元来向いていない。
- 従来のしくみでは、通信ノードが移動ノードの位置情報を取得するため Mapping Agent の IP アドレスを DNS から解決しなければならないが、本来は移動ノードの識別子から移動ノードの位置情報を直接取得できるべきである。

以上の問題を解決するため、非構造な識別子を扱うのに適している DHT を用いて、移動ノードの識別子から移動ノードの位置情報を直接取得するしくみを考案しようと我々は考えている。このように DHT を用いて移動ノードの位置情報を管理するしくみを用いた LIN6 を neoLIN6 と呼ぶ。

今回の合宿ではこの neoLIN6 を設計・実装するための予備実験として、DNS の代わりに DHT によって移動ノードの識別子と Mapping Agent の IP アドレスの対応を管理するしくみを実装し、構築した。これを DHT+LIN6 と呼び、DHT+LIN6 を WIDE 合宿ネットワークで動作させ、

- LIN6 の通信において、DNS の代わりに DHT を実際に用いた場合、どのような問題があるか
- DHT の実装として、DHT+LIN6 で利用している pychord にスケーラビリティおよび運用に耐える堅牢性があるか

を確認することを目的とした。

1.6.2 概要

我々は pychord のパッケージと LIN6 のパッケージを用意し、実験参加者には DHT のネットワークへの参加をしていただいたり、LIN6 ノードとして DHT のネットワークを利用していただいたりした。

1.6.3 実験環境

実験は DHT のネットワークを構成するノードと LIN6 の移動ノードで構成される。

DHT のネットワークは慶應義塾大学矢上キャンパス寺岡研究室内に設置されたノード（203.178.135.134）と WIDE 合宿 NOC 内に設置されたノード（203.178.158.159）で構成され、参加者のノードがこのネットワークに参加することによって pychord のスケーラビリティおよび堅牢性を確認することができる。WIDE 合宿ネットワークと寺岡研内に設置されたノード間の通信は DVB-RCS 経由で行い、遅延の大きい DHT のネットワークを構築した。

また、実験に参加する LIN6 ノードはこの DHT を利用して通信ノードの Mapping Agent の解決を行うことで、DHT を利用した場合の問題点を探ることができる。

1.6.4 結果

実験の結果、DHT を用いた LIN6 は正常に動作し、移動透過性を保証した通信を行うことができた。しかし、DHT から移動ノードの Mapping Agent を発見する部分に要する処理時間にはばらつきがあり、処理時間が多くかかる場合（数秒単位の処理時間がかかる場合）には、通信が開始できないという問題が発生した。

DHT の問い合わせ処理時間の原因は、DVB-RCS を往復するパスが生成されたことや、実装上の問題により競合が発生している可能性などがある。今回の実験では、寺岡研に置かれた 1 ノードと合宿地とのアクセスを DVB-RCS 経由にしたため、そのノードに割り振られたデータのアクセスに時間がかかっていた。なお、4〜10 ノードでの実験の結果、ほとんどの問い合わせの所要時間は 1 秒〜1.5 秒の範囲に収まることが確認されている。

DHT のネットワークは最大で 10 ノードで構成され、正常に動作した。実験によって得られたデータは

```
http://member.wide.ad.jp/~doi/
camp0403-result/
```

で公開している。

1.6.5 まとめ

実験を行ったことにより、pychord や pychord のリゾルバの実装のバグを発見することができた。バグは合宿期間中に修正され、合宿の後半では、定常的に DHT の運用ができた。修正後の pychord は安定して動作しており、今後も DHT の実装として使いたいと考えている。

1.7 レイヤ 2 ネットワークにおけるホストの位置特定技術についての実験

1.7.1 目的

本節では、奈良先端科学技術大学院大学の櫨山によって開発された、レイヤ 2 ネットワークにおけるホストの位置特定ツール（L2traceman）の詳細と 2004 年 WIDE Project 春合宿における運用実験の内容およびその結果を報告する。

1.7.2 概要

2003 年に流行した Slammer[195]、Blaster[194] などのネットワークワーム、ネットワークウイルスによるネットワーク崩壊の被害は記憶に新しい。これらのネットワークワームはサブネットのレイヤ 3 ゲートウェイでフィルタリング処置を施したとしても、サブネット内での蔓延は防げない。感染ホストをサブネット内に放置しておくと ARP リクエストや感染用のパケットによりレイヤ 2・レイヤ 3 ネットワーク機器の負荷が上昇しネットワークが崩壊する危険性がある。また、ノート PC などの移動端末に感染した場合、移動することによってほかのネットワークにネットワークワーム、ネットワークウイルスの感染を広げてしまう危険性がある。感染の拡大を最小限に食い止めるには、レイヤ 3 ゲートウェイでのフィルタリングだけでは不十分であり、レイヤ 2 スイッチ上でのフィルタリングなどによる感染ホストの隔離、および感染ホストの物理的な位置と感染ホストの管理者を把握し、ネットワークワーム・ネットワークウイルスの除去、ソフトウェアへの適切なパッチ適用が必要となる。

ネットワークワームやネットワークウイルスに感染したホストをネットワーク上から隔離する場合、レイヤ 2 ネットワーク上における位置、すなわち感染ホストが接続する最初のレイヤ 2 機器およびホストが接続しているレイヤ 2 スイッチ側の物理インタフェース、VLAN などの論理インタフェースを把握することは重要である。感染ホストが接続するレイヤ 2 スイッチのインタフェースまで特定できることによってより細やかなフィルタリングによる感染ホストの隔離を行うことができ、さらには物理的な位置とマッピングすることによって管理者がパッチ当ての作業や連絡に向かわなければいけない相手のいる場所を特定できるからである。

過去 WIDE 合宿の実験ネットワークにおいて、無線 LAN 環境については大江ら [344] によるホストの位置特定および隔離システムの実験運用が行われ、感染ホストの特定と安全な実験ネットワーク運用に大いに役立った。しかし、大江ら [344] はラディウス認証と無線 LAN 機器の情報を利用しているため、無線 LAN のレイヤ 2 ネットワーク内でのみ有効なものである。Blaster[194] が流行した 2003 年 9 月合宿では、大江らのシステム [344] は導入されていたが、有線 LAN ネットワーク上で感染したホストまたは問題のあるホストを大江らのシステムで特定することはできないため、問題のあるホストを特定するために非常に困難を極めた。困難であった理由の 1 つには、合宿ネットワークのトポロジがフラットなレイヤ 2 ネットワーク構成であったため、IP アドレスによる捜索範囲、つまり位置の絞込みが行いにくいことがあった。物理的に広範なレイヤ 2 ネットワーク上に発生したセキュリティインシデントに対し迅速に対応するには、対象ホストのレイヤ 2 ネットワーク上、および物理的な位置を特定できる技術が必要である。

そこで、レイヤ 2 スイッチのフォワーディングデータベース（FDB）の情報を利用したレイヤ 2 ネットワーク上での追跡を行うツール（L2traceman）を試作し、2004 年 3 月の WIDE 合宿の実験ネットワーク上で実験を行った。本報告では、L2traceman の追跡方式、2004 年 3 月での実験方法および結果について述べる。

1.7.3 追跡方式

現在提案されているトレースバック方式の多くはルータ上に特別な実装が必要であったり、専用の機器を設置する必要がある。これは IP アドレスの偽装に対応することを前提として、その上でパケット、またはトラフィックを基にしたレイヤ 3 ネットワーク上での追跡を行うためである。そのため、細やかな追跡ができるほど、導入コストが高いというトレードオフが発生する。

今回試作した L2traceman は IP アドレス、MAC アドレスが偽装されていないことを前提にし、IP アドレス単位、または MAC アドレス単位という粒度の荒い追跡しか行えないが、既存のレイヤ 2・レイヤ 3 ネットワーク機器自身がフレーム転送またはパケット転送に用いるデータベース情報を追跡に用いるため、特別な機器を用意する必要がなく、導入コストが低いことが特徴である。

追跡における基本的概念は文献 [123] で述べられているレイヤ 2 トレースの概念を基にしている。

L2traceman では、レイヤ 3 ネットワーク機器上の IP アドレスと MAC アドレスのマッピング情報とレイヤ 2 スイッチ上の FDB 情報を SNMP 経由で取得し、あらかじめ用意しておいた、サブネット内部のレイヤ 2・レイヤ 3 ネットワーク機器のトポロジ（物理的な接続情報）を基にしてサブネット下における IP アドレスまたは MAC アドレス単位での追跡を行う。追跡の結果、SNMP での情報が取得できるネットワーク機器までではあるが、追跡対象ホストが接続するレイヤ 2 ネットワーク上で対象ホストに最も近いレイヤ 2 スイッチおよび対象ホストと通信を行っている物理インタフェースの割り出しが行える。

IP アドレスと MAC アドレスの変換

MAC（Media Access Control）アドレスは IEEE（米国電気電子技術者協会：Institute of Electrical and Electronic Engineers）で管理されており、単一レイヤ 2 ネットワーク上で送信先ノードのネットワークインタフェースを識別するために設定されるハードウェアアドレスである。イーサネットネットワークでは各ネットワークカードに固有の MAC アドレスを割り当て、イーサネットネットワーク上でのアドレス解決を行いフレームを転送している。

あるサブネットにおいて、宛先 MAC アドレスはネットワーク機器側でフレームが流入したネットワークインタフェースに付けられた MAC アドレスである。つまり、宛先 MAC アドレスから追跡対象ホストと通信を行っているネットワークインタフェースがわかり、そのインタフェースが接続するレイヤ 2 ネットワークが判明する。また、送信元 MAC アドレスは追跡対象ホストのネットワークインタフェースに付いている MAC アドレスであり、固有に識別されるものである。つまり、送信元 MAC アドレスはノード自身を固有に識別できる情報である。

IP アドレスと MAC アドレス（物理アドレス）の対応を扱う MIB は RFC で標準 MIB として、IPv4 と MAC アドレスの対応は SNMPv2 MIB[188] で、IPv6 と MAC アドレスの対応は IPv6 MIB[122] でそれぞれ定義されている。こ

れらの MIB に対応しているレイヤ 3 ネットワーク機器からは、IPv4 アドレスと MAC アドレスの対応は ipNetToMediaPhysAddress（OID は .1.3.6.1.2.1.4.22.1.2）に対しインタフェース番号と IPv4 アドレスをインデックスの値として参照することで、IPv6 アドレスと MAC アドレスの対応は ipv6NetToMediaPhysAddress（OID は .1.3.6.1.2.1.55.12.1.2）についてインタフェース番号と IPv6 アドレスを 10 進数表記した値を用いて参照することで取得できる。

これらの MIB を参照することで、レイヤ 3 ゲートウェイから追跡対象ホストの IP アドレスを基にして、追跡対象ホストの MAC アドレスと追跡対象ホストと通信を行っているレイヤ 3 ゲートウェイ上のインタフェースが特定できる。

特定したレイヤ 3 ゲートウェイ上のインタフェースをあらかじめ用意しているサブネットのトポロジ図と照らし合わせて、次に SNMP の問い合わせを行うべきレイヤ 2 スイッチを特定する。

MAC アドレスを用いたレイヤ 2 ネットワーク上の追跡

レイヤ 2 ネットワーク上の追跡はレイヤ 2 スイッチがフレームの転送を行うときに用いる FDB を利用して行う。

レイヤ 2 スイッチでは、MAC アドレスと送信先ポートの対応表は FDB に記録されている。この対応表は、レイヤ 2 スイッチにフレームを受信したときに、フレームが到着したポートとそのフレームに含まれている送信元 MAC アドレスを元にしてレイヤ 2 スイッチ上で自動的に生成される。フレームを転送する際に今度は記憶されている送信元 MAC アドレスを宛先 MAC アドレスとして、一致する MAC アドレスに対応するインタフェースに対してのみフレームを転送する。

よって、流入パケットの送信元 MAC アドレスを元にレイヤ 2 スイッチの FDB に対し追跡対象ホストの MAC アドレスを検索することでレイヤ 2 スイッチが追跡対象ホストと通信を行っているインタフェースの番号がわかる。このインタフェース番号と MAC アドレスの対応表はレイヤ 2 スイッチのブリッジ MIB[55] から取得できる。

ブリッジ MIB から取得できるインタフェースの特性はレイヤ 2 スイッチの FDB の作り方によって異なる。Foundry 社や Allide-Telesis 社製のレイヤ 2 スイッチの場合はブリッジ MIB から取得できるインタフェース番号は IfMIB[189] と対応した物理インタフェースの番号と対応しているが、Extreme 社製のレイヤ 2 スイッチの場合は取得できるインタフェースは仮想インタフェースの番号となる。

このような場合、ベンダ拡張 MIB にアクセスすることでより詳しい FDB の情報を取得できる場合がある。Extreme 社製や Cisco 社製のレイヤ 2 スイッチの場合は、ファームウェアのバージョンによるが、ベンダ拡張 MIB に対してアクセスすることで物理インタフェース、さらには VLAN インタフェースの番号を取得できる。

こうして得た物理インタフェース番号とサブネットのトポロジーとを照らし合わせて、次に問い合わせを行うレイヤ 2 スイッチを特定する。この作業を再帰的に行っていくことで、最終的には追跡対象ホストに最も近いレイヤ 2 スイッチと追跡対象ホストが接続している物理インタフェースを特定でき、追跡対象ホストの物理的位置の範囲を絞り込める。

無線 LAN ネットワークにおいても、同様に、アクセスポイント、または無線 LAN スイッチからブリッジ MIB の情報またはベンダ拡張 MIB の FDB 情報を SNMP で取得できるならば、追跡対象ホストが接続しているアクセスポイントまでを特定できる。

1.7.4 実装

今回 L2traceman の実装は Perl5.6.2 で行い、SNMP モジュールとして Net::SNMP を用いた。実装環境としての OS は FreeBSD4.8 を用いた。

ifDescr、ipv6IfDescr、ipNetToMedhiaPhysAddress、ipv6NetToMediaPhysAddress、dot1dTpFdbPort といった MIB を基本的に参照して追跡を行う。ifDescr、ipv6Descr はインタフェース番号からインタフェース名（レイヤ 2 スイッチのインタフェース名、モジュール番号、スロット番号、ポート番号）を取得するために用いる。それ以外の MIB は前項で述べたように MAC アドレスやインタフェース番号の取得に用いる。

しかしながら、ベンダによって FDB およびブリッジ MIB の実装方式は異なるため、上記の標準 MIB だけでは十分な情報を取得できない場合や、ブリッジ MIB を出力させることでレイヤ 2 スイッチに高い負荷をかけてしまう場合がある。L2traceman の

実装において、Foundry 社製、Allide-Telesis 社製、Extreme 社製、Cisco 社製のレイヤ 2 スイッチに対して、参照する必要のある標準 MIB および各ベンダの拡張 MIB に関する調査を行った。その結果、Extreme 社製、Cisco 社製のレイヤ 2 スイッチに関しては標準 MIB だけでは正確な物理インタフェース名の特定までを行えないが、ベンダ拡張 MIB にアクセスすることによってブリッジ MIB のみでは不十分であった情報を取得し、追跡が行えることがわかった。

Extreme 社製レイヤ 2 スイッチへの対応

Extreme 社製の場合は、ブリッジ MIB から取得できるインタフェースは FDB を作成するための論理インタフェースである。これは Extreme 社製のスイッチが VLAN グループごとに FDB を作成しているためである。また、Extreme 社製のスイッチにおいては、ブリッジ MIB を作成する設定を有効にしておかないと取得できず、また、そうして作られたブリッジ MIB へのアクセスはスイッチに対して高い負荷を与え、SNMP による反応が非常に遅い。そのためブリッジ MIB ではなく、ベンダ拡張 MIB に対して SNMP によるアクセスを行い、FDB の情報を取得する。ブリッジ MIB の dot1dTpFdbTable に対応する情報は Extreme ware version 6.2.0 から取り入れられているベンダ拡張 MIB である extremeFdbMac、extremeFdbPort にアクセスすることで、ブリッジ MIB と同様に MAC アドレスと記憶している物理インタフェース番号を取得する。まず extremeFdbMac にアクセスして追跡対象ホストの MAC アドレスをマッピングしている VLAN インタフェースに割り当てられたインデックス番号を取得する。取得した VLAN インタフェースのインデックス番号を extremeFdbPort から検索し、物理インタフェースに割り当てられたインデックス番号を取得する。こうして取得した物理インタフェースに割り当てられたインデックス番号は ifDescr、ipv6IfDescr で物理インタフェースに対し用いられているインデックス番号と同一のものである。

Cisco 社製への対応

Cisco 社製のレイヤ 2 スイッチの場合、ブリッジ MIB にアクセスすることで ifDescr、ipv6IfDescr で用いられている物理インタフェースに割り当てられたインデックス番号を取得できるが、製品の型番によって ifDescr、ipv6IfDescr で取得できるインタフェース名は管理者が普段使用する {モジュール番号/スロット番号/ポート番号} の様式とは異なっている場合がある。そこでベンダ拡張 MIB にアクセスすることで {モジュール番号/スロット番号/ポート番号} の様式の物理インタフェース名を獲得することができる。

1.7.5 実験

実験環境

L2traceman を利用した実験を 2004 年 3 月に行われた WIDE 合宿の実験ネットワーク上において行った（図 1.6）。L2traceman は Camp PC 提供の DHCP サーバ兼ネットワーク IDS である PC–DHCP–Serv 上で稼動した IDS によってアラートの上がったパケットの送信元 IP アドレスに対して追跡を行った。L2traceman が実行された場合、基幹ルータである c7204 とレイヤ 2 ネットワークを構成する bi4k、fi4802、fi-bof1、fi-lobby に対して SNMP リクエストを発行し、対象ホストの MAC アドレスの記録を追うことで追跡を行った。

IDS で検知対象としたパケットは Blaster などのウイルスに関するシグニチャに当てはまるものとした。Blaster は 2003 年 9 月の WIDE 合宿にて合宿ネットワーク上で蔓延したため、今回の合宿ネットワークにおいて明確な監視対象であったので、L2traceman の追跡対象とした。

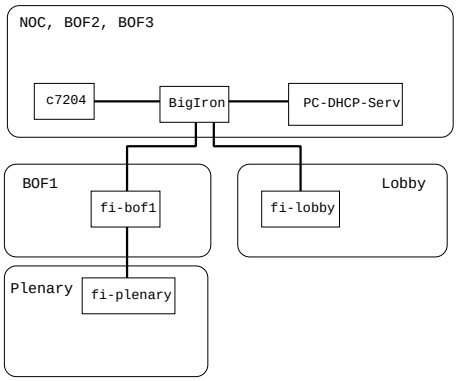

図 1.6.　合宿トポロジ

実験結果

実験結果としては、IDS でウイルスに関するアラートが発生しなかったため、十分な実験結果を得ることができなかった。ただし、手動実行による検証で

は、wlanopsの結果と比較した結果、同じIPアドレスの検索に対して、同一のMACアドレスと同一のアクセスポイントの名前を得ることができた。

1.7.6 まとめ

今回の実験では、十分なデータが集まらなかったため、L2tracemanの有効性を示すまでには到らなかった。ただ、半年間の間に合宿参加者の間に最低限のネットワークセキュリティを保つ意識が生まれ、再びワームが蔓延する悲劇が起こらなかったという現実は非常に喜ばしいことである。

今回の実験で得られた知見とアンケートからいただいた意見を参考にし、引き続きL2tracemanの開発を行っている。

1.8 2004年春合宿でのDVTS Splitterデモンストレーション報告

NP WGでは、2004年春合宿において奈良先端科学技術大学院大学で開発したDVTS Splitterのデモンストレーション実験を行った。

詳細については、NP WGの報告書を参照されたい。

1.9 フローベースによる合宿ネットワーク計測

roft WG（`http://www.roft.org/`）は、フローを用いたトラフィック観測システムを提案し、研究開発を行っている。2004年WIDE春合宿において、roft WGはflow実験チームとして実験を行った。当該実験では、フローと呼ばれる観測単位で合宿のネットワークトラフィックを観測することで、合宿ネットワークの安定運用への貢献を目指した。また、提案システムの有用性のアピール、改善案の募集を行った。本実験の詳細と結果は、roft WG報告書を参照されたい。

1.10 イベント定義可能な実空間ミドルウェアの実現

Spears WGは、WIDE合宿における参加者およびプログラム委員の利便性を向上するために、2004年3月および9月のWIDE合宿において合宿運営を支援するシステムの研究開発を行った。

本実験の詳細な内容については、Spears WG報告書を参照されたい。

1.11 WIDE Hourオーバレイ相互接続実験

1.11.1 目的

WIDE Hourは、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、いか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、補完通貨（地域通貨）を用いて解決するというアプローチをとる。

WIDE合宿における一連の実験では、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補完通貨による取引を適用し、その効果を計測することを狙っている。

今回の実験では特に、補完通貨の取引を行うための2種類のオーバレイネットワーク（Web+SSLおよびIM[1]+PGP）を提供し、その相互接続性を検証することを目的とした。

1.11.2 概要

Web+SSLベースのシステムとして運用しているWIDEメンバ間用ポイント交換システムWIDE Hourについて、XMPP[2]/Jabberを用いるIM+PGPベースの別実装を用意し、相互に接続した。これによりWIDEに近接するコミュニティでもWIDE Hourを用いることができ、仕事を進める上での協調メカニズムを実現しやすくなると期待できるが、そのことをライヴネス、フェアネス、ユーザビリティなどの評価により検証することを試みた。

WIDE Hourとは? WIDE Hourは、「WIDEのために1時間労働する価値」を表す交換媒体である。将来的に、WIDE Project内の補完通貨システムとして用いられることを目指している。

WIDE Hourは、時間を単位としたポイントだと考えることができるが、ポイントのほかに以下の尺度を導入し、取引の頻度を上げたり、対象を広げることに対してインセンティヴを設けている。

WIDE Power: ポイントの収入・支出の累積が大きく、現在のポイントの残高（の絶対値）が小さいほど高い値となる（取引の頻度と収入・支出のバランスを表す）。

WIDE Variety: 取引先が多様であるほど高い値となる（取引の多様性を表す）。

1 IM: Instant Messaging.
2 XMPP: Extensible Messaging and Presence Protocol.

WIDE Across: 電子手形として抽象化された WIDE Hour に対する裏書の連鎖が長いほど高い値となる（信用を表す）。

1.11.3 実験環境

補完通貨システムは、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。

サーバ類は、合宿前後を含み長期的な運用を行うため SFC に設置し、合宿会場では各自のコンピュータにてクライアントプログラム（Web ブラウザおよび XMPP/Jabber クライアント）のみを動かすことにした。

Web サーバ

Web ベースの WIDE Hour システムは、`http://fran.sfc.wide.ad.jp/`にて提供した。

Web ブラウザ

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ようにした。

XMPP/Jabber サーバ

XMPP/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。

XMPP/Jabber クライアント

XMPP/Jabber クライアントプログラム wija を提供した。

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 1.4.2（1.3.1 以上で可）
- GnuPG 1.2.4（PGP Freeware は不可）
- wija 0.02（実験用 XMPP/Jabber クライアント）

バージョンはいずれも当時の値である。現行の wija およびその前提となるソフトウェアのバージョンについては、`http://www.media-art-online.org/wija/` を参照されたい。

補完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っている必要がある。今回から、wija には鍵交換を行う機能を盛り込んだ。

1.11.4 結果

統計

表 1.2 は、前回（2003 年秋合宿）の実験の統計をまとめたものである。

対して、今回の実験の統計を表 1.3 にまとめた。

表 1.2. 2003 年秋合宿での統計

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
	うち、無効化された取引	7 回

表 1.3. 2004 年春合宿での統計

何らかの形で参加		75 名
総ログイン回数		228 回
	証明書使用	206 回
	SSL＋パスワード	1 回
	平文パスワード	21 回
総取引回数		127 回
	うち、無効化された取引	2 回
	XMPP/Jabber を利用	15 回

考察

前回と比較し、規模が約半分に縮小した。目新しさがなかったことが原因ではないかと推測する。

オーバレイ接続実験については、十分なデータを取得できるほどには XMPP/Jabber の利用はされなかった。これには、次の原因があったと考える。

- システムの提供が遅れた（初日に提供できなかった）。
- 鍵交換は簡単にしたが、そもそもソフトウェアのインストールなど、使うまでの手順が多過ぎた。

前回の実験では、相手を決めて空取引を繰り返すことにより WIDE Power を不当に増大させる傾向があり、フェアなシステムを実現する上では問題だったが、このことについては、今回、WIDE Variety という尺度を導入することにより、空取引の評価を

抑える効果があったと考える。しかし、評価式は暫定的なものだったため、今後の検討が必要である。

1.11.5 まとめ

今回の実験は、補完通貨に関する最初の実験（2003 年春合宿；IM+PGP システムのみ提供）と 2 回目の実験（2003 年秋合宿；Web+SSL システムのみ提供）の結果を踏まえ、その両者を統合する環境の提供を行った。

結果、システムが機能することは検証できたが、それはソフトウェアのテストの範疇に留まる程度の成果であり、WIDE Hour が本来の目的を達成するためには、今後、次のことを行っていかなければならないと考える。

- 広報
  - 知ってもらうことに力を入れる。
    - より多くの人に使ってもらう。
    - メカニズムについて知った上で行動してもらうことが、メカニズムデザイン上は重要である。
  - 視覚化に力を入れる。
- ユーザビリティの向上
  - ユーザ数の多い Windows 利用者に快適な環境を提供することを心がける。
  - エンジニアよりも、補完通貨（地域通貨）に興味のある人々に使ってもらい、フィードバックを得る。

実際にこれらのことを推進した結果（詳細な報告は別の機会に譲りたい）、合宿後の話ではあるが、2005 年 1 月現在、補完通貨システム（WIDE Hour ではないが、それをより一般化したシステム）のユーザは着実に増加している[3]。

2005 年の WIDE 合宿では、2004 年春合宿以降に得られた成果を活かし、改善された WIDE Hour システムを提供し、本来の目的にさらに近づけるよう、努力したい。

第 2 章　2004 年秋合宿ネットワーク

本章では、2004 年 9 月 6 日（月）から 9 日（木）まで、長野県信州松代ロイヤルホテルにおいて開催された WIDE Project 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を示す。

図 2.1 中の左右に引かれた点線の内、上部が対地として用いた慶應義塾大学湘南藤沢キャンパス（以降、SFC）、下部が合宿地である。

合宿会場とインターネット（SFC）との接続には地上線 1 種類および衛星回線 2 種類の計 3 種類の回線を用いた。地上線としてフレッツ ADSL モア（12 Mbps 契約）を用い、衛星回線として Ku バンド、および DVB-RCS を用いた。各回線の最大帯域を図 2.2 に示す。尚、図中の ‘上り’ は合宿地からインターネットへ向かう方向、‘下り’ はインターネットから合宿地へ向かう方向を示し、値の単位は Mbps である。

実際の環境の制限、また各実験グループからの希望などにより、合宿ネットワークへの要求条件は以下となった。

- ADSL 回線環境値

  NTT 局舎から合宿地までの距離が 1.8 km で伝送損失が 26 dB であった[4]。しかし、実際の利用可能帯域は、この予想値から算出される値（8 Mbps）より大幅に値が小さく、最大 1 Mbps 程度であった。これより、ホテル側に了承を得て ADSL モデムから IDF 室間の配線調整を継続的に行ったが、性能を向上させることは出来なかったため、IDF 室から先の環境に問題があったと思われる。
- Streaming WG による実験

  Streaming WG による実験のため、時間帯に応じて、富士通研究所向けのトラフィックを遅延

3　本質的に peer-to-peer システムであるため、具体的な利用者数を把握できない。
4　NTT 東日本　線路情報開示システム（http://www.ntt-east.co.jp/line-info/）による。

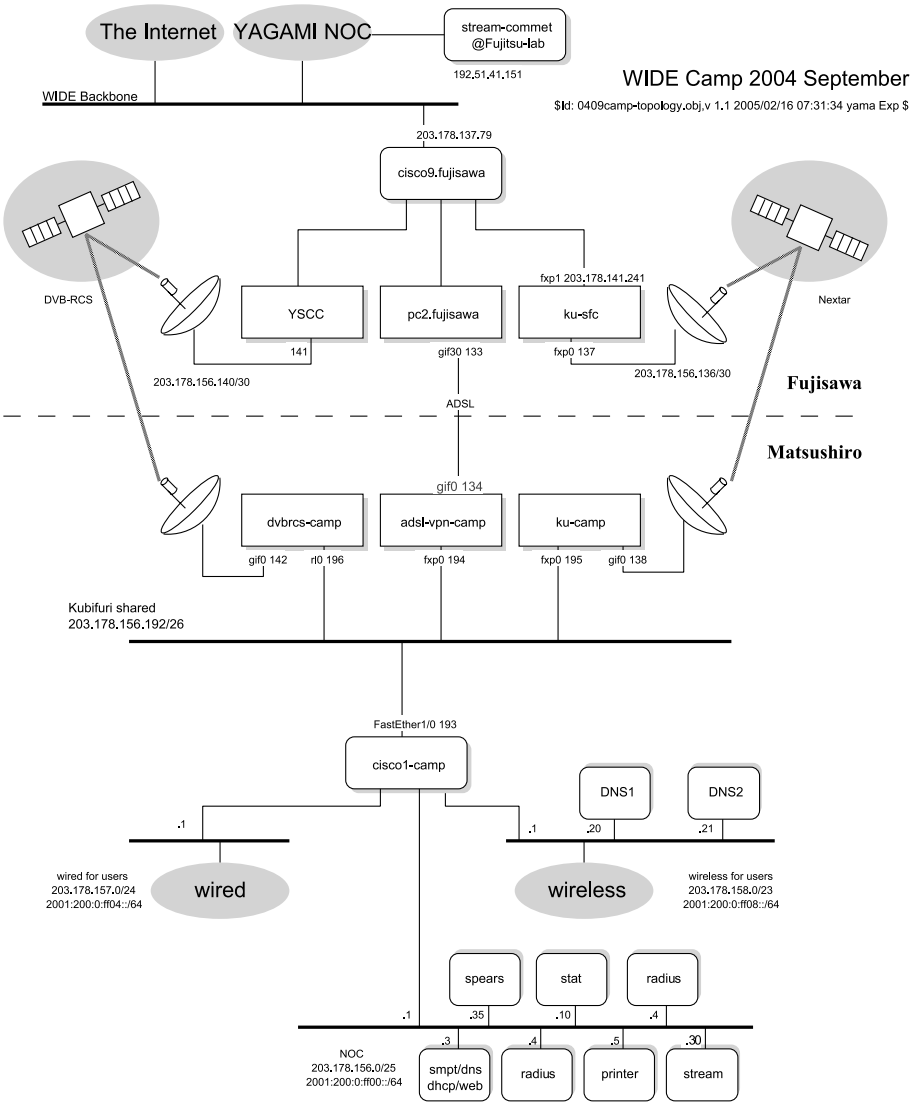

図 2.1.　2004 年秋合宿のネットワーク構成

		上り	下り
地上回線	ADSL	1	12
衛星回線	DVB-RCS	2	10
	ku バンド	0.5	1.5

図 2.2.　対外接続線の種類（単位は Mbps）

の大きい対外線（DVB-RCS 経由、Ku バンド経由）に振り分けること。

- DNS WG による実験
  実験用 PC を無線セグメントに設置し、有線インタフェースによる接続を行うこと。
- Spears WG による実験
  センサ用機器を無線セグメント内に設置し、無線インタフェースによる接続を行うこと。
- 無線接続環境
  近年では、合宿参加者のほとんどが有線インタフェースは利用せず、無線インタフェースのみを利用している状況であること。なお、会場には既設の無線基地局は存在せず、802.11 ワイヤレスシステム運用は、持ち込む機器のみを考慮すれば十分であることを確認した。

以上より、下記のポリシ決定、および結果が得られた。

- 3 種類の対外線の使い分けポリシ、および消費帯域
  基本的な対外接続線を、ADSL 経由とし、ADSL の調整を行う際は、DVB-RCS 経由、富士通研究所向けトラフィックを随時流動的なも

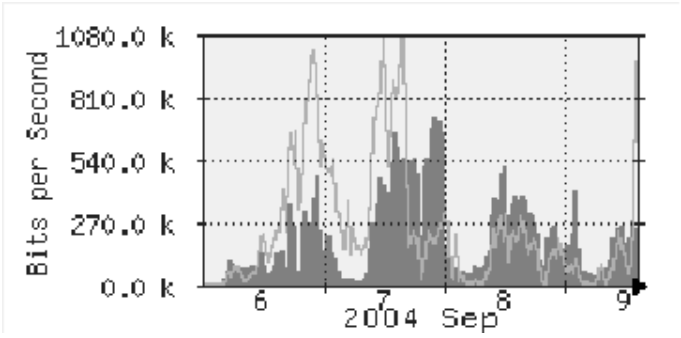

図 2.3. ADSL 回線の利用状況

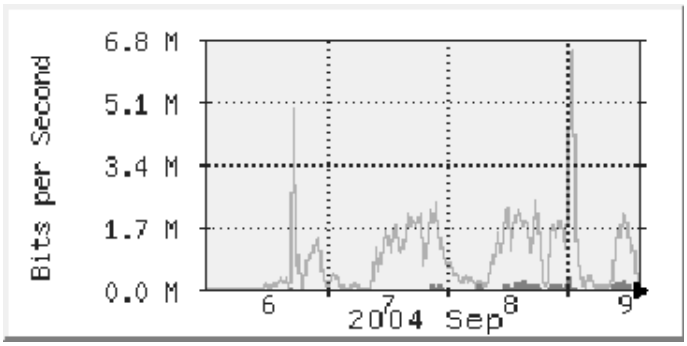

図 2.4. DVB-RCS 回線の利用状況

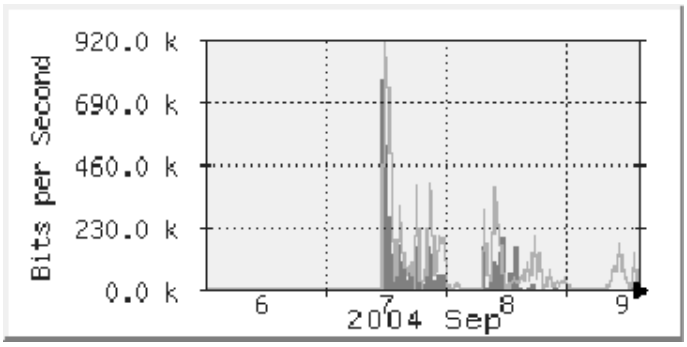

図 2.5. Ku バンド回線の利用状況

のとして扱った。図 2.3、図 2.4、図 2.5 にその消費帯域を示す。

- 会場内部のトポロジ、アドレスアサイン

　会場環境による制約、および、各実験グループからの希望からは複雑な要求が発生しなかった。このため、運用上のコスト削減を考慮し、できる限り単純な構成とした。

　これより、一般参加者向けの、有線用、無線用および管理用の 3 種類のセグメントを用意し、無線到達範囲が会場全体をカバーできるよう、無線基地局の物理的配置を考慮した。

　図 2.6 に各セグメントのアドレス範囲を、図 2.7 に無線基地局を含むネットワーク構成機器の物理的配置、および配線を示す。

用途	IPv4	IPv6
管理用	/25	/64
有線	/24	/64
無線	/23	/64

図 2.6. 各セグメントでのアドレス範囲

- 不正トラフィックへの対応

　本合宿ネットワークのほどんどのトラフィックが、無線利用者によるという現状から、2003 年 WIDE 秋合宿から利用されている wlanops WG

による 802.11 ワイヤレスネットワーク管理システムを利用した。

これにより、オペレーション上、不正なトラフィックを送出している端末が確認できた場合は、その端末の無線接続を禁止し、参加者のネットワーク環境を維持することができた。

• hotstage の実施

合宿前に合宿ネットワークの事前検証、および設定を実施する hotstage では、本番と同様の機器、セグメント、アドレスアサインを用いて、SFC セミナーハウスに仮設ネットワークを構築し、各実験グループによる実験の、最終調整および、事前検証を行った。

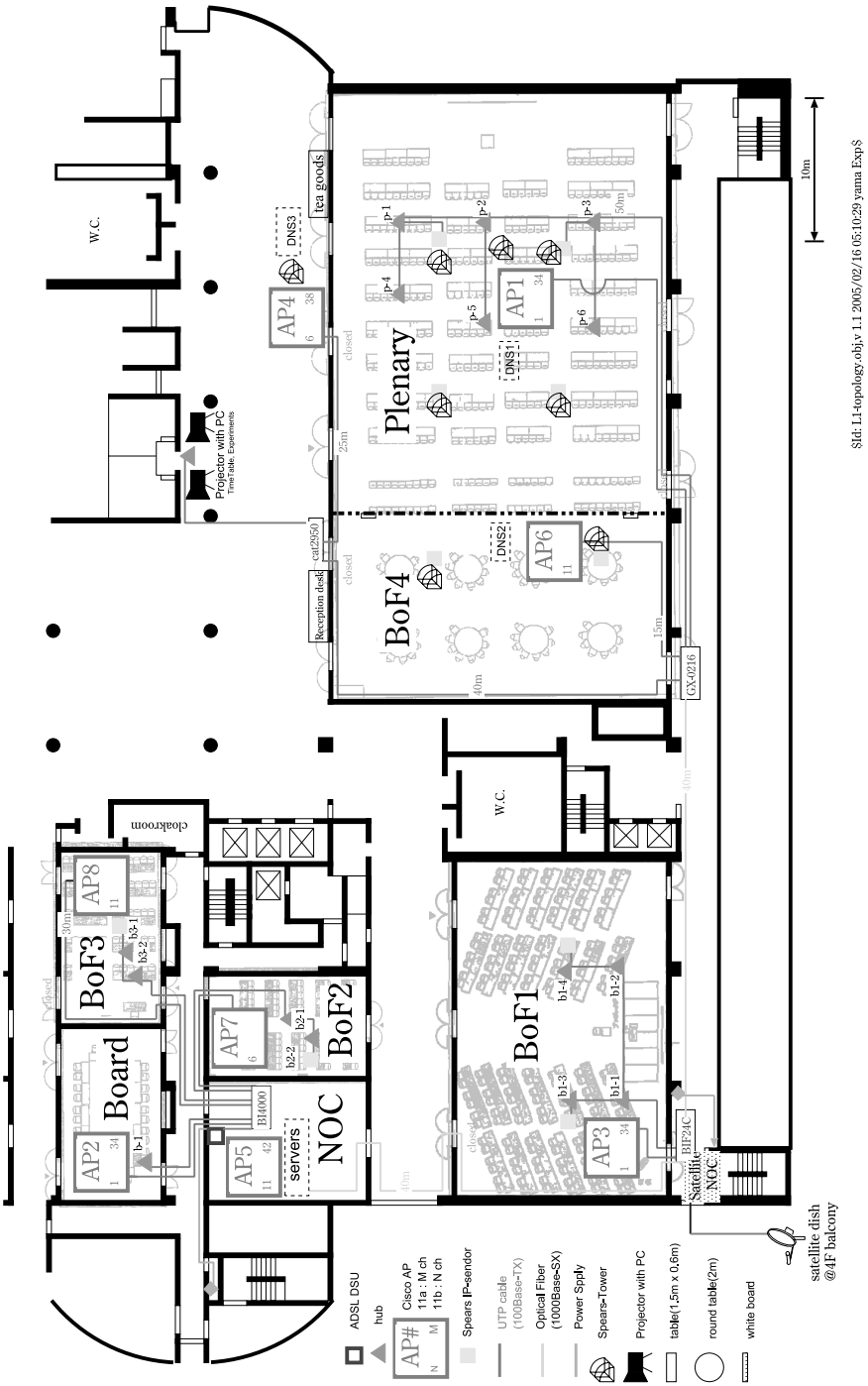

図 2.7.　本合宿の物理的ネットワーク構成

これにより、合宿地での円滑なネットワーク構築、および運用が可能となった。

2.1.1 公開したネットワーク情報

- 対外線利用状況
  ADSL 回線に問題があったこと、衛星回線も含め対外線帯域には制限があることの理解を求めるため、プレナリ対外線利用状況を Web 上およびプロジェクタ投影により、参加者に公開した。
- 各 AP 利用状況
  参加者の無線 LAN 利用の利便性向上を図るため、各無線 LAN アクセスポイントの利用者数、および配置を上記の対外線利用状況とともに公開した。
- トラブルチケット
  参加者がネットワーク利用の際に生じた問題の状態を共有しつつ追跡するため、紙ベースのトラブルチケット利用を行った。しかし、トラブルチケットの存在、およびコンタクト先などの情報をはじめ、参加者へ十分に広報できていなかった点があったため、有用に活用できたとはいえない。

2.1.2 合宿ネットワークを利用した実験

本合宿では以下の 3 つの実験が行われた。

1. Comet TCP と帯域制御（Streaming WG）
2. DNS man in the middle attack の検証（DNS WG）
3. イベント定義可能な実空間ミドルウェアの実現（Spears WG）

その詳細は次節以降で示す。

2.1.3 その他

- 電源配置および利用量の計測
  プロジェクタなど、突然の停電からの保護が必要な機器を保護するため、過去の利用例を参照しつつ、部分的な電源回線増設工事を行った。また、部分的にではあるが、定期的な電流利用量を計測したため、今後の合宿への参考情報を残す事ができた。
- 衛星機材設置
  初めて利用する会場であったが、ホテル側担当者の親切な対応があり、また、衛星アンテナ設置、および回線引き込みに対する調査を入念に行う事が出来たため、会期中の暴風雨に対しても問題無く運用することができた。
- ご飯チェック
  食事会場入口での ID チェックは、Spears WG の協力により、システム全体をチェック場所付近に移動させることで、チェック場所まで回線敷設するコストが削減できた。また、食事の予約変更 Web インタフェースを提供することにより、事務局の食事利用者数把握のコスト削減もできた。

2.1.4 まとめ

松代ロイヤルホテルは、初めて利用する会場であったが、ホテル側担当者の親切な対応が得られ、PC 側での入念な調査が出来たことにより、合宿参加者へ提供する施設としては非常に良い環境であった。ただし、PC 側からの情報公開において、多少不足があったため、参加者と PC との間に少々溝が出来てしまったことが悔やまれる。

今後は、参加者との明示的なインタフェースを設置し、参加者と PC が一体となった合宿が開催される事に期待する。

2.2 Comet TCP と帯域制御

2.2.1 概要

長距離回線において TCP は実効通信速度が低いことが知られている。Comet TCP 通信技術は長距離高速回線（long fat pipe）での高速化を念頭に研究された。2004 年度の秋の WIDE 研究会では、Comet TCP を 1 Gbps を超えるような高速回線ではなく、衛星回線に適用してその有用性を実験した。通常の TCP 通信では帯域の上限を探るために定期的にパケット欠落と帯域縮小を常にくりかえすのに対し、本通信方式は割り当てられた帯域を 10 μsec 程度の精度で保持するという特徴を持つ。この特徴によって衛星回線のように長距離だが数 Mbps〜数 10 Mbps の低速な帯域においても、TCP に比較して、有用であることを実証するのが目的である。

実験の結果、iperf を使用した定量的実験では 1.5 Mbps の細い回線でも RTT 500 msec という長い遅延のおかげで Comet TCP の効果が認められた。しかし、Web の閲覧ではその他の要素も多く利用者の主観的な観測では効果は見られなかった。また、ストリーミングではパケットロスが生じると

TCP 層が RTT の長さ分の時間パケットを止めてしまい、アプリケーションレベルで見た改善はできなかった。

この実験の詳細については Streaming WG の報告書を参照されたい。

2.3 DNS man in the middle attack の検証

本実験は、DNS に対する攻撃をサーベイ、研究し、その防御方法について考察するために行った。

本実験の詳細な内容については、DNS WG 報告書を参照されたい。

2.4 イベント定義可能な実空間ミドルウェアの実現

Spears WG は、WIDE 合宿における参加者およびプログラム委員の利便性を向上するために、2004 年 3 月および 9 月の WIDE 合宿において合宿運営を支援するシステムの研究開発を行った。

本実験の詳細な内容については、Spears WG 報告書を参照されたい。