第 XXX 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 30 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　2008 年春合宿ネットワークに関する報告

本ドキュメントでは、2008 年 WIDE 春合宿におけるネットワークと各実験の報告を述べる。

1.1 概要

2008 年 WIDE 春合宿は、2008 年 3 月 3 日から 6 日にかけ静岡県浜松市の浜名湖ロイヤルホテルで開催された。当合宿プログラム委員会では、ネットワークチームを編成し、多数の被験者を用いた実験（ネットワーク研究のテストベット）の舞台かつ利用者のインフラとなる合宿ネットワークを構築した。

当合宿で実施された実験は、下記の通りである。

- おいでよ　ハイエナの森（地球規模 OS 外殻プロトタイプによる融通力の検証）
- Camp Support System on Radio Wave Structured Network
- 実証実験用車載モバイルルータの製作
- 分散 SNS におけるコンテンツの共有・同期

1.2 合宿ネットワーク

1.2.1 方針と設計

今回は、ネットワーク層に密着したような大規模な実験を招致したいとの考えのもと、研究会などを通しネットワーク実験に関するブレインストーミングと招致活動を行った。しかし、残念なことに、ネットワークレイヤに密接したような実験は実施されなかった。

そこで、基盤のネットワークトポロジに関しては、なるべく単純なものを採り、構築コスト（人・時間・機材）を低減することとした。また、会場である浜名湖ロイヤルホテルは過去にも多数の合宿開催経験があることから、電源や対外ネットワーク引き込みなどについては、過去の経験や資料をもとにした容易な設計が可能であった。

インターネットがインフラとなって以降、WIDE プロジェクトにおいても参加後間もない（主に）学生諸氏にとっては、ネットワークを構築し運用するという経験を積む場が少なくなっている。我々は、合宿ネットワークの構築を、このような経験を積む場とも考え、ネットワークやサーバの構築経験の少ない学生等のメンバーを中心に、ネットワーク構築経験を積みネットワーク構築の楽しさや大変さを伝えることを、ネットワークチーム内の一つのテーマと位置づけることとした。

1.2.2 ネットワークトポロジと運用

ネットワークトポロジを図 1.1 に示す。

先にも記したとおり、今回のネットワークトポロジは、ネットワーク層に影響を与える実験の応募が無かったこともあり単純なトポロジとした。対外線としては、物理ネットワークとして NTT 西日本のフレッツ光プレミアム、ISP として OCN を利用した。その上で、WIDE インターネットを延伸するために、WIDE NOC と合宿地の間でトンネル接続を行い、IPv4/IPv6 のネットワークを接続した。

合宿地内には、ユーザ収容セグメントとして 2 セグメント、サーバ類設置用セグメントとして、1 セグメントを用意した。レイヤ 3 ルーティングは、1 台の PC ルータでの集中ルーティング（経路制御プロトコル無し）で行った。

また、合宿地側に、基本的なネットワークサービス（DNS/DHCP/Web 等）を提供するための設置した。このサーバでは、ネットワークの監視や計測、障害のチケッティングなども実施した。

1.2.3 ユーザ収容（無線 LAN）

ほぼ全ての参加者が無線 LAN デバイスを持参するようになり、数年前より、ユーザ収容は、無線 LAN のみで提供している。この方針により、会場内ネットワークの敷設所要時間が非常に削減できている。

従来、臨時で短期間の利用である合宿ネットワークでは、一部の実験的な取り組みを除いて、高度な認証などは行って来なかった。しかし、無線 LAN のセキュリティが注目されている現状を鑑み、今合宿

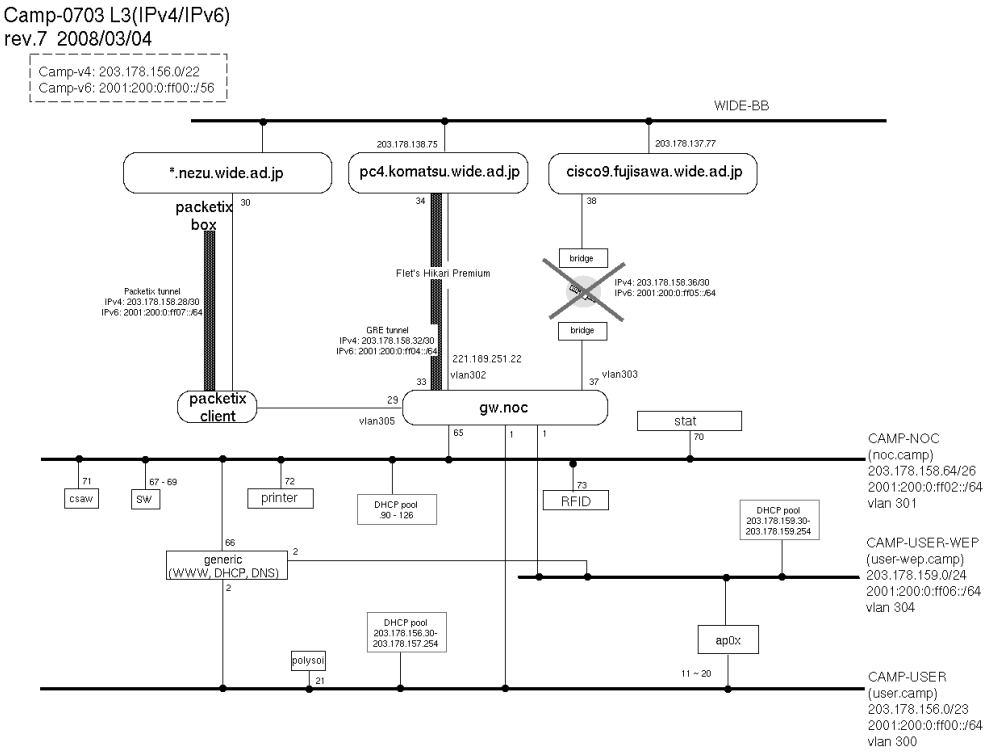


図 1.1.　ネットワーク

では、MoCA 証明書を用いた EAP-TLS 認証による接続を基本サービスとした。

接続用の利用者向けドキュメントを各種用意したこともあり、非常に多くの参加者に EAP-TLS 認証での接続を行って頂けたが、一部の古いハードウェアでは接続ができないという問題が依然存在した。このような方々のために、細々ながら WEP のみでの接続も準備した。

1.3 実験

1.3.1 おいでよ　ハイエナの森（地球規模 OS 外殻プロトタイプによる融通力の検証）

- 実験実施者：IDEON ワーキンググループ、斉藤賢爾

1.3.1.1 実験の概要

地球規模 OS 外殻（シェル）のプロトタイプとして開発しているメッセージングソフトウェア wija を用い、合宿参加者自身や合宿会場に持ち寄られたコンピュータの余剰資源を融通し合い、その有効性を検証した。

具体的には、以下を行った。

1. 携帯機器（DoCoMo、AU、SoftBank の携帯電話や iPod touch など）を通して XMPP インスタントメッセージング・ネットワークに接続し、人間がセンサとなって BoF やプレナリセッションなどの賑わい度を測定し、みんなで共有する。
2. 自分のディスクのデータを（圧縮・暗号化後）他人のコンピュータのディスクにバックアップとして保存する。ストレージ領域の貸し借りの関係の公平性は、SSC（Storage-Standard Currency; ストレージ本位通貨）により担保される。

1.3.1.2 実験の成果

アイデアの検証が行えたことに加え、以下に代表される有用な意見および問題提起を頂けた。

- バックアップされるデータにはフレッシュネスがあり、長くバックアップが存在するわけではないことは理解できる。すると、その限られた時間の中でのデータの広がり具合を計測することで、インターネットがどれだけモバイルになったかということが分かる可能性がある。
- センサとしての自分の履歴という概念と、みんなの情報を集約したものという概念、それに加えてもしかしたら他者の履歴という概念も出てくるかも知れないので、それらの概念がユーザ

インタフェース上、明確に分かれるようにした方がよいのでは？

- ゲーム機で実施する、とかの方が、機種依存性に引っ張られなくてよいのでは？
  - 学生が大学で公共性のある研究をするためにゲーム機開発企業と契約するということができる可能性もある。
  - そうした方法で大学でやることのよさが引き出せていけるのでは。
- iPod touch に期待しており、ランチレスキュー（当実験と類似する、丸の内地区におけるランチの混雑状況を計測する実証実験）のようなものの、災害時の本当の難民を救うためのシステムを作りたい。
- インセンティヴは、ライフログのようなかたちだと参加者自身が面白い。
- ライフログを健康管理に使う研究をしており、記録をとると、健康管理に役立つ、ということはユーザは理解できる。しかし、入力のモティベーションは、また別の話で、別にデザインしていく必要がある。

1.3.1.3 まとめ

今回の実験では、余剰資源の融通のための基本的な機構について、人間の労力やストレージ領域を互いの利益のために利用する簡易な検証を行うことができた。

根本的な課題として、余剰資源の融通の有用性については多くの人が理解を示すものの、実際に人々が資源を提供するに至るまでには、動機づけの大きな壁があることを改めて認識できた。このことに関しては、ユーザインタフェース等、技術で解決できる部分もあれば、人々の目的意識に抜本的な変革を迫らなければならない面もある。

今後は、技術開発のみならず、あらゆるレベルにおいて、この課題に取り組んでいきたい。

1.3.2 Camp Support System on Radio Wave Structured Network

- 実験実施者：安田真悟、井上朋哉

1.3.2.1 実験内容

WIDE 合宿ネットワーク内に設置・運用される Camp Support System を、オーバーレイネットワークによる、多チャンネルスポットキャストを用いて設置・運用を行う実験を行った。この多チャンネルスポットキャストは、Radio Wave Structured Network と P2P@DNS の 2 つの技術アプローチにより実現される。これにより、大規模な仮設ネットワークにおいて多数のノードを用いたサービスを、簡便に設置・運用する技術の実現を目指す。

1.3.2.2 目的と意義

近年の合宿では、WIDE に加入して間もない合宿参加者の為に、CampPC によって合宿内に合宿支援システム（Camp Support System）が構築運用されてきた。この合宿支援システムは、話者紹介システムである、Introduction System、アナウンス事項表示システムである Ticker System、参加者のコミュニケーションの促進を図る SNS である、Portal System により構成されていた。後に、Portal System は、合宿支援システムから独立し、現在は、csaw ワーキンググループにより WIDE メンバ全体のコミュニケーションを促進するための SNS として運用されている。しかし、合宿中にのみ必要とされる Introduction System や Ticker System の設置・運用は必須であり、その構築は煩雑であるため、その設置運用は PC の仕事に大きな負荷をかけている。合宿支援システムの設置には、csaw、Server、RFID Reader、プロジェクタ、PC が必要となり、その電源およびネットワーク配線、ホスト名・IP アドレスの管理・設定も必要となる。そのため、その管理は、ホスト名・IP アドレス等の設定を事前に NetPC と交渉する必要があり、NetPC 側では、DNS 等における設定作業が必要となり、多くの担当者にまたがる仕事を要する。

そこで、オーバーレイネットワークを用い、自律分散的にホストの名前解決を行う P2P@DNS を用いる事で、DNS に関する交渉・設定作業を不要とした。これにより、各合宿支援システムのホストは DHCP による動的な IP アドレスで設置可能となる。

また、合宿内各所に置かれた、プロジェクタに話者情報やアナウンス事項を表示する為には、表示の応答性向上の為に、プッシュ型配信システムが求められているが、プッシュ先ホストの指定等に煩雑なシステムと設定を必要としていた。そこで、オーバーレイネットワークを用いて情報種類（チャネル）を分けスポット的な配信を行う事が出来るミドルウェアである Radio Wave Structured Network を用い

る事で、各表示機器は起動時に必要なチャネルをミドルウェアに設定するだけで、必要な情報を受信でき、且つ Server に送信先を選択する機構・設定を不要とした。

さらに、これまで合宿支援システムで話者を認識する為に利用されていた RFID Reader、表示用 PC は有線 LAN で接続されるように設計されていた為、電源だけでなく、有線 LAN の配線も必要としていた。近年の合宿内ネットワークでは、参加者の生活ネットワークは無線 LAN が主となっていた為、合宿支援システムへの有線 LAN の配線は合宿ネットワークの設計の足かせとなっていた。そこで表示用 PC、RFID Reader を共に無線 LAN でユーザーセグメントに設置する事で、有線 LAN の配線の手間とネットワーク設計の手間をなくす事にした。

1.3.2.3 実際の詳細

合宿支援システムにおいて用意した機材は以下の通りである。

- RFID Reader 4 台
- Mac mini 10 台（サーバ 1 台、プロジェクタ表示用 9 台）

プロジェクタ表示用 Mac mini 9 台は DHCP にて動的な IP アドレスを設定し、ホスト名を P2P@DNS を用いて解決した。P2P@DNS は、Symphony と呼ばれる structured P2P ネットワークを構築し、DHT を用いた名前解決機構を構築する。これにより、ネットワークセグメントをまたいだ名前解決が可能となる。P2P@DNS を構築する各ノードは、DHCP や静的に付与された IP アドレスとホストネームが対となったリソースレコードを、DHT 上へ動的に保持させる。これにより、お互いが DNS を利用することなく任意のホストネームによる通信が可能となる。

RFID Reader は有線 LAN 配線の手間を省くため、無線 LAN を用いた RFID Reader を作成した。合宿支援システムで従来用いていた RFID Reader は、SFC Auto-ID Lab が作成した物で、Armadillo-J と Takaya TR3-C201 RFID リーダ/ライタモジュールを利用した物であった為、有線 LAN のネットワークインターフェースしか持っていなかった。しかし、無線 LAN 対応 Armadillo は高価であった為、我々は FON ソーシャルルーター La Fonera を拡張し、Takaya TR3-C201 を制御する事で無線 LAN RFID Reader を作成した。

サーバ用およびプロジェクタ表示用 PC は P2P@DNS とは別に、P2P オーバーレイネットワークを用いた多チャンネルスポットキャストネットワークである Radio Wave Structured Network（以下 RWSN）と言うデータ転送網を構築している。

RWSN は各参加ノードの設置位置情報を基に、隣接ノード同士が結合し非構造化 P2P オーバーレイネットワークを構成する。そして構成したネットワーク内の特定の位置にスポット的にデータを転送する事が可能なミドルウェアである。各ノードで起動する RWSN 対応アプリケーション（本実験では、Introduction System、Ticker System の各表示アプリケーション）は、起動時に設置場所と受信したいチャネルを RWSN ミドルウェアに設定する。各ノードの RWSN ミドルウェアは転送されてきたパケットのチャネルを確認し、自ホストに該当チャネルを受信するアプリケーションが起動している場合、パケットのペイロードを該当アプリケーションに転送する。これにより、設置エリア、チャネルの組み合わせによりデータを一斉配信することが可能となる。

今回はサーバのみ固定 IP アドレスを用いた。これは時間的制約から、La Fonera 内で動作する P2P@DNS ソフトウェアを作成する事が出来なかった為、RFID Reader のデータ送信先アドレスを指定しなければいけなかったためであり、本来 La Fonera でも P2P@DNS が動作していれば、全装置が DHCP による動的な IP アドレスにて運用する事が可能であった。

話者がマイク脇の RFID Reader にタグをかざした後のデータ処理の流れは以下の通りである。

RFID Reader にかざされた、合宿参加者のタグから読み込まれた WIDE ID は UDP のユニキャスト通信で、Introduction System、Ticker System 供用のサーバ（I/T Server）に送られる。I/T Server は csaw サーバに WIDE ID を問い合わせ、WIDE メンバの氏名、所属などの情報を取得する。取得したデータを RWSN に送信対象エリアとチャネルを設定して送信する。送信されたデータは RWSN により表示用 PC に送られ、該当エリアにあるプロジェクタに現在の話者が表示される。

1.3.2.4 実験結果

実験結果としては、利用者側の意見として、十分動作したという意見があったので本実験は成功したと

言える。合宿内ネットワークが不安定になった場合に、度々 P2P@DNS および RWSN が構成する P2P Network が分断される現象が発生し、サービスが一時停止した。これは、ネットワークの瞬断、構成されたオーバーレイネットワークの分断を検知する機構が無かった事に起因するが、実装とテスト環境において、安定したネットワークである北陸先端科学技術大学院大学の学内無線 LAN ネットワークを利用していた為、洗い出す事が出来なかった。

本結果から、仮設ネットワーク内で実験を行うネットワークアプリケーションは、ネットワークの瞬断等の影響を受ける可能性が高く、不安定なネットワーク上でのアプリケーションの挙動を、十分に検証しておく必要を痛感した。

1.3.3 実証実験用車載モバイルルータの製作

- 実験実施者：Internet CAR ワーキンググループ、Nautilus6 ワーキンググループ

1.3.3.1 はじめに

現在 Internet CAR ワーキンググループ（iCAR ワーキンググループ）で用いている車載モバイルルータはおよそ 2 年前に構築したもので、それ以降大幅なバージョンアップを実施していない。そこで、今後数年の実証実験に耐える実験基盤構築のため、車載ルータの更新を実施する。実際の更新作業を WIDE 合宿の活動の一環として実施することにより、作業の現場を広く知ってもらい、多くの方に興味を持ってもらうことを目指した。

1.3.3.2 実装目標予定機能

以下の機能を実装したモバイルルータを合宿期間中に製作することを目標とした。

- Linux NEPL ベースの基本システム
- PHS と無線 LAN の複数インターフェースのサポート
- NEMO Basic Support
- IPsec サポート（for future use）
- IKEv2 サポート（for future use）
- Nautilus6 ワーキンググループ開発の HAiku ウェブインターフェースによる管理

1.3.3.3 各機能の実装状況報告

1.3.3.3.1 Linux NEPL ベース基本システム

これまで iCAR では NetBSD SHISA ベースのモバイルルータを用いてきたが、SHISA の開発が近年停滞していること、および Linux NEPL 用の IPv4 サポート（DSMIPv6）が進んでいることから、今後数年の研究開発を支える基盤システムとして Linux を採用することとなった。

本合宿では、モバイルルータに利用するハードウェアリソースが貧弱である（Soekris ベースの組み込み用ボード）を鑑み、Debian ベースの Voyage Linux を元に、NEPL システムを構成した。

最終的に一応動作するものは開発できたものの、モバイル機能を管理するプロセスが突然終了するなど、なお動作に不安が残る状態である。後の検討の結果、組み込み用に最適化された Linux を用いたため、こちらで想定した以上の機能が削られており、それらを追加していく過程でシステムが不安定になっているのではないかと判断された。別途 Internet CAR ワーキンググループの活動報告にもある通り、本活動は、この後通常の Debian ベースの Linux を採用する様に方針を変更し、継続している。詳しくは Internet CAR ワーキンググループの部を参照されたい。

1.3.3.3.2 PHS と無線 LAN の複数インターフェースのサポート

合宿期間中には、無線 LAN 経由のモバイル機能の確認ができたものの、PHS を用いたモバイル機能は実装できなかった。PHS を用いる場合、モバイルルータに付与されるアドレスは IPv4 アドレスになるため、DSMIPv6 機能が必要になる。実験実施時点では、まだ DSMIPv6 の動作が確認できていなかったため、本項目は今後の課題として残された。

1.3.3.3.3 NEMO Basic Support

NEMO Basic Support による基本的なモバイルネットワークの機能は、合宿実験期間中に確認できた。ただし、前述の様にシステム自体が不安定な状態であったため、システム安定化作業が課題として残された。

1.3.3.3.4 IPsec サポート（for future use）

IPsec サポートは元になるシステムが標準でサポートしており、機能的な不足はなかった。本合宿期間中は、モバイルルータ、および慶應義塾大学湘南藤沢

キャンパスの NOC 内に設置されたホームエージェントに IPsec の設定を投入し、合宿地から正しく藤沢キャンパスのホームエージェントに登録できることを確認した。

1.3.3.3.5 IKEv2 サポート（for future use）

IKEv2 をサポートすることで、前述の IPsec の設定投入が自動化でき、管理が容易になる。本合宿期間中は、IKEv2 機能を実現する racoon2 ソフトウェアを基本システムに組み込む事ができたものの、その動作を確認するには至らなかった。この機能に関しては今後の課題である。

1.3.3.3.6 Nautilus6 ワーキンググループ開発の HAikuWeb インターフェースによる管理

WIDE 合宿に先立ち、慶應義塾大学湘南藤沢キャンパスにあらかじめホームエージェントを設置する作業を実施した。その際、Nautilus6 ワーキンググループが開発したホームエージェント用の管理インターフェース HAiku を同時に導入し、GUI 画面によるホームエージェント管理体制を整えた。本システムは、以後の iCAR ホームエージェントとして活用されている。

1.3.3.4 まとめ

本合宿では、今後の iCAR ワーキンググループの研究開発の基盤となる車載モバイルルータの更新作業を実施した。当初目標としていた機能を実装するには至らなかったものの、基盤となるオペレーティングシステムを NetBSD から Linux に変更し、一応の動作を確認できたことにより、今後の活動へ繋げることができた。車載モバイルルータ開発は、本合宿以後も継続的に続けられており、2008 年末の時点では EMOBILE（PHS の代わり導入された）を用いた DSMIPv6 機能も一部動作に漕ぎ着けている。今後、本基礎システムを早急に完成させ、実環境を用いた各種実験へと繋げていく予定である。

1.3.4 分散 SNS におけるコンテンツの共有・同期

- 実験実施者：高井一輝、CSAW ワーキンググループ

1.3.4.1 実験の概要

合宿開催地に設置された CSAW サーバと、SFC に設置された CSAW サーバとの間で、コンテンツの連携に関する実験を行った。バックグラウンドで “イベント” の交換を行うことにより、ユーザがどのサーバ上で投稿やコミュニティ参加などの操作を行ったかにかかわらず、最終的には連携している全てのサーバでコンテンツを同期させる。これにより、ネットワークの接続性が安定しない環境や、一時的にネットワークが利用できないような環境であっても、安定的に情報へアクセスしコミュニケーションを行うことが可能となる。

1.3.4.2 実験の背景

csaw ワーキンググループでは、コミュニティの活動を支援するためのアーキテクチャやシステムに関して議論を行うとともに、WIDE 向けコミュニティシステムである “CSAW” の運用を行っている。CSAW のサーバは fujisawa-noc（SFC）に設置されており、通常時はインターネットからのアクセスが可能であるが、WIDE-Camp 時には現地（WIDE-Camp ネットワーク内）にサーバを設置し、運用を行っている。これは、Camp Support System などの他のシステムとの連携を行い、また、Camp-Net の対外線への影響を抑えるためである。その際、両サーバ間でコンテンツの不整合が起きないよう、一時的にサービスを停止し、コンテンツの同期とリダイレクトの設定などを行っている。

また、CSAW や CSAW とソースコードの大部分を共有している ACS（Academic Community System）が様々な組織で利用されることで、情報が異なるサーバ上に分散してしまうと、利用者は複数の SNS（サーバ）を利用しなくてはならない。組織外に出すことができないセキュアな情報がある一方、特定の相手とは共有したい情報も存在するため、各 SNS が単独で動作しつつも必要に応じて緩やかに連携する必要がある。

これらの問題を解決するため、非同期に、かつ部分的に情報を共有する仕組みが必要となる。本実験では CSAW を拡張し、実際に非同期でデータを転送することにより、複数サーバ間でのコンテンツ共有の実証を行った。

1.3.4.3 実験の詳細

1.3.4.3.1 情報の共有・同期手法

本実験では、複数サーバ間で情報を共有・同期するために、既存の CSAW システムを以下のように拡張した。

- 全ての情報に対して、全てのサーバ上で一意な識別子（UUID）を付与
- 情報の状態を変化させる全てのユーザ操作（掲示板への投稿やコミュニティへの参加、メンバ情報の更新など）を “イベント” として記録
- 各サーバはお互いに任意のタイミングで共有・同期対象のサーバにアクセスし、イベントのリストを取得
- 取得したイベントリストを解析し、各サーバ上で処理を実施

イベントの転送はPULL型（受信側が送信側へアクセスし、取得する）とした。受信側からhttpsで送信側サーバに接続し、認証を行う。認証に成功した場合は、送信側サーバがイベントリストをXML形式で受信側に転送し、受信側でXMLを解析して必要なイベントのみ処理を行う。また、各コンテンツに付与されたアクセス権や親子関係などのメタデータを保持するために、受信側に制御対象のオブジェクト（ユーザやコミュニティなど）が存在しない場合は、内容を持たないスタブ・オブジェクトを自動で作成し、メタデータが失われないようにした。

1.3.4.3.2 実験環境

Camp-Net内にCSAWサーバ（csaw.camp.wide.ad.jp）を設置し、fujisawa-noc に常設のサーバ（csaw.wide.ad.jp）との間でイベントの交換によるコンテンツの共有を行った。具体的には、以下の情報について、両サーバ間での双方向の共有・同期を行った。

- コミュニティ掲示板への投稿
- コミュニティ掲示板へのレス
- コミュニティの作成と削除
- ユーザのコミュニティへの参加情報
- コミュニティフォルダへのファイルアップロード

それ以外の情報については、イベントの記録のみを行い、イベントの転送は行わなかった。イベント転送を行わない部分については、csaw.wide.ad.jp側の変更を実験終了後に破棄し、csaw.camp.wide.ad.jp側の変更のみ両サーバに反映させた。

1.3.4.4 実験結果とまとめ

実験期間中には150件以上のイベントがcsaw.camp.wide.ad.jpからcsaw.wide.ad.jpに転送・処理された。また、csaw.wide.ad.jpからcsaw.camp.wide.ad.jpへのイベント転送も十数件あり、双方向でコンテンツの共有・同期を行えることを確認できた。

本実験では2つのサーバ間での共有・同期を行ったが、サーバ数が増加した場合はイベントの転送がさらに複雑になる。複数サーバ間で効率的かつ安定的にイベントを転送するアルゴリズムについて考える必要がある。また、本実験では認証情報の共有は行っていないため、複数サーバに同じユーザがログインする場合でも、管理者が別々にアカウントを作成し、認証情報を個別に設定する必要がある。認証情報が複数サーバ間で共通である方がユーザにとっては利便性が高いが、現実的には、認証情報を共有できる範囲は限定的であると考えられるので、範囲を限定して認証情報を共有する仕組みを考えていく必要がある。

1.4 新規取り組みと課題、反省

1.4.1 ホットステージ未実施の試み

例年、合宿地での素早い立ち上げやトラブル軽減を目的として、合宿期間の数週前に、合宿ネットワークに接続する機器や実験を実施する方々を一堂に集めて接続試験等を行う「ホットステージ」を開催している。

今回の合宿では、

- ネットワーク構成が単純であったこと
- 実験内容としてインフラとなるネットワークへの影響が軽微と考えられること

から、コストの削減などを考慮してホットステージを行わないという試みを行った。つまり、サーバなどは、各拠点で予めセットアップなどをすました上で、合宿会場において最終的な調整を行うこととした。

しかし、ネットワークやサーバの構築経験が少ないメンバーが構築に携わったこともあり、構築や障害の切り分けに時間を要し、合宿初日にはネットワークが不安定な時期も生じ参加者に影響を与えることとなった。ホットステージを実施しない場合は、特に、構築時間の見積もりやベテランメンバーが構築に介入するタイミングなどが今後の課題として残った。

1.4.2 NOCのプレナリ内設置

例年、ネットワーク機器類の設置ならびに操作は、専用に部屋を設けNOCルームとして運用を行ってきた。この形態では、

- 一部のメンバーが NOC 部屋に籠もり合宿自体（セッションなど）への参加が難しい
- 合宿ネットワークがどのように出来ているのか、どのように運用しているのかなどが参加者に見えづらい

などという問題が指摘されていた。

そこで、今回、新たな取り組みとして、プレナリルーム内にオープンな NOC スペースを構築し、ネットワークの構築ならびに運用を行うという試みを行った。本取り組みに対しては、一部、合宿ネットワーク構築メンバーからは「落ち着かない」という感想もあったが、全体としては、

- NOC 内での作業中でも興味のある発表を聞いたり質疑を行ったりできる
- 参加者が行われている実験に興味を持ち、交流が図れる

という概ね好意的な感触が得られた。

1.4.3 技術伝承、連携

今回、衛星リンクの確立に非常に手間取ったという問題が生じた。衛星通信用の機器が老朽化していることもあり、問題発生当初から機器の故障等が疑われ、本来の原因に至るまで非常に長い時間を要してしまったことが問題であった。最終的に、原因は、合宿地に設置したアンテナの直下（階下）に大電流の強電ケーブルが存在し、その影響で通信ができないことと判明し、アンテナの移動で障害は解決した。

この問題は、特に衛星回線のオペレーションという合宿ネットワークチームにも専門家が少ないものであったことから、少数の担当者に完全に依存してしまいチームとしてのフォローアップができなかったものである。このような経験から、前任者からの技術・経験の伝承をより進めることは言うまでもなく、

- メンバー間での意思疎通・状況の確認
- 合宿ネットワークチーム内で全てを解決しようとせず、必要に応じて外の専門家とも連携した障害追跡

をより進める必要があるという認識を得た。

第 2 章　2008 年秋合宿ネットワークに関する報告

2.1 はじめに

本章は、2008 年 9 月 9 日（火）から 12 日（金）にかけて、長野県長野市松代町にて開催された WIDE 合宿ミーティングの活動をまとめたものである。

本合宿ミーティングのプログラム構成に関しては 2.2 節で、ネットワーク構成および合宿ネットワークを活用した実験に関しては 2.3 節にて詳しく報告する。

2.2 プログラム報告

2.2.1 プログラムのねらい

2008 年秋合宿では、「現地に集合したからこそできる活動をサポートする合宿」という目標のもとにプログラムが構成された。

毎回、合宿では数多くの BoF が開催されているが、最近の傾向として、必ずしも合宿で実施する必然性のない報告が中心の構成になる場合が増えているように感じられていた。決して安くない参加費用を払って集まるからには、集まったことによって実現できる成果をだすべきである。

今回、合宿 PC では参加型チュートリアル、参加型実装合宿、参加型プログラミング講習会のような、顔を合わせることによって劇的にその効果を増大させることができるスタイルの活動を最大限支援するべくプログラムを構成した。

目的遂行のために考案され施行されたプログラムは以下の通りである。

- mini ワークショップ。
- 初心者による BoF 総括。

さらに、合宿参加者、特に国外からの参加者（あるいは、日本語を母国語としない参加者）と、小さな子供を同伴する参加者の障壁を下げるため、今回初めて以下の活動を試みた。

- プレナリセッションの英語ログ。
- 子供部屋の設置とプレナリセッションの中継。

各項目に関しては、以後の節で個別に報告する。

なお、通常の合宿と同様に、以下のプログラムも施行された。

- ワーキンググループ BoF
- 研究発表
- チュートリアル

2.2.2 mini ワークショップ

2008 年秋合宿では、合宿地で顔を合わせることで相乗効果をだす企画のひとつとして、mini ワークショップの企画を募集した。mini ワークショップの内容に特に制限は設けず、「合宿地に集まって作業することで、最大限の効果が発揮される」ワークショップを開催することのみを条件とした。例としては、実装大会、デモ大会、密な議論の時間を確保する、などが挙げられる。普段のオンライン活動では埋めきれない部分を、合宿中に埋めて欲しいという期待から企画されたプログラムである。

合計 5 つの mini ワークショップが開催された。それぞれの概要は以下の通りである。

1. Haskell を使って Scheme 処理系を作ろう！

   コーディネータ 山本和彦/株式会社インターネットイニシアティブ

   概要 純粋関数型言語 Haskell で、関数型言語 Scheme を実装します[1]。実装の作業を通じて、Haskell の最も難しい部分であるモナドや、Scheme の内部に対して理解を深めます。これは Haskell や Scheme のチュートリアルではありません。Haskell や Scheme をある程度知っており、理解を深めたい人が参加して下さい。以下の文献ぐらいは、あらかじめ読んでおきましょう。

   Scheme の参考書

   - プログラミング Cauche

   Haskell の参考書

   - 入門 Haskell
   - ふつうの Haskell プログラミング
   - 情報処理の連載[2]

   プログラム 2008 年 9 月 10 日、11 日 13:00–18:20

   - 一日目：6 章までを目指す
   - 二日目：最終章の 12 章までを目指す

2. SCTP/DCCP 時代のネットワークプログラミング

   コーディネータ 小塚真啓/京都大学

   概要 TCP や UDP に代わる新しいトランスポートプロトコルである SCTP と DCCP は、それ自体が研究・開発になるだけの存在から各種環境において、容易に利用可能なものへと変わりつつあります。本ワークショップでは、これまで SCTP ワーキンググループで充分に取り扱ってこられなかった、SCTP や DCCP はどうしたら使えるのか、どのように使えばうれしくなるのかという点に焦点を当て、SCTP/DCCP プログラミングを実践します。ごく簡単な使い方から、TCP や UDP との差異まで幅広く SCTP/DCCP プログラミングを取り扱うので、ネットワークプログラミングの初心者から熟練者まで幅広くお楽しみいただけます。

   プログラム 2008 年 9 月 10 日 13:00–18:20

   (a) SCTP/DCCP の簡単な紹介（20 min.）
   - SCTP/DCCP を使うと何ができるのか
   - SCTP/DCCP を使うにはどうすればいいのか

     各種 OS での対応状況、NAT 問題への対処の情報を紹介

   (b) SCTP/DCCP を使ったプログラミング（30 min.）
   - TCP/UDP 互換ソケット API の使い方

     SCTP を TCP っぽく使うノウハウなど
   - SCTP/DCCP での新しいソケット API の使い方

     Multi-Stream 機能や Partial Reliability 機能など

   (c) SCTP/DCCP プログラミング実践（2 hours）
   - みんなでプログラミングしてみよう

     SCTP/DCCP の機能を使う課題
   - 既存のアプリケーション SCTP/DCCP 対応

     こういう部分を探し出して変えるとかのノウハウをプレゼン

   (d) ディスカッション（1 hours）

3. 今後のインターネットにおいて L3 が果たす役割と可能性を議論し、それを実現するアーキテクチャを提案しようワークショップ

   コーディネータ 関谷勇二、竹井淳/WIDE ボード

1 http://halogen.note.amherst.edu/~jdtang/scheme_in_48/tutorial/overview.html
2 http://www.sampou.org/haskell/ipsj/

概要 単なる評論家の集まりではないかと噂されるボードメンバーが、その持ち得る知識と経験を駆使して、将来（3〜5 年後）に訪れる世界を、あくまでも L3 インターネットを中心とした視点から、ワークショップ参加者とのトークを交え熱く議論する。単なる議論で終わらせるのではなく、これからの L3 インターネットが果たす役割をいくつかの実践的なシナリオを用いて検証し、実現するための具体的なアーキテクチャを示す。本ワークショップのアウトプットは、数年後のネットワークコンピューティングをとりまくアーキテクチャの提案である。

プログラム 2008 年 9 月 11 日 13:00–18:20

(a) 本ワークショップの趣旨説明（15 min）
(b) ボード合宿議論の総括（30 min）
(c) 参加者とのラフな議論［場合によってはグループワーク］（2 hour）
(d) アーキテクチャの構築（1 hour）
(e) まとめ（30 min）

4. iPod touch/iPhone アプリケーション開発

コーディネータ 川本芳久/大阪学院大学

概要 iPhone/iPod touch の SDK が公開されてからしばらく経ちました。開発が進んでいる人も苦労している人もいるでしょう。そこで、iPhone SDK で開発している人を集めて、iPhone SDK を使いこなすための技術を共有したいと思います。この mini ワークショップに参加すれば、開発速度がさらに加速すること間違いなし、です。

参加対象としては、iPhone SDK を使ってアプリケーションを作る環境を持っている人を想定しています。WIDE ネタなら参加できるという人もどうぞご参加ください。（秘密厳守です。）

プログラム 2008 年 9 月 10 日 13:00–18:20

みんなでプログラム。適宜 15 分程度で以下のような内容のプレゼンテーションを行う予定です。

- iPhone SDK、つまずくのはココ。
- ここまでできてる iPhoneStumbler。CoreLocation の使い方。
- WiFi 情報の取り方。
- ハマる！ QTKit。
- Bonjour を使ったネットワークアプリケーションの一例。

5. お手軽 MR 作成ワークショップ

コーディネータ 佐藤雅明/慶應義塾大学

プログラム 2008 年 9 月 11 日 13:00–18:20

- みんなで作る MR（モバイルルータ）

アンケートによれば、mini ワークショップ企画に関して 75%の参加者がよい試みだと答えており、92%の参加者が企画の継続を望んでいることから、一定の成果があったものと結論づけられる。一方で、長い時間を取ることにより、BoF に参加できない、複数の mini ワークショップに参加することが難しいなどの弊害についても考慮すべきであることも指摘された。

2.2.3 初心者による BoF 総括

今回の合宿では、最終日に合宿初参加者による BoF 総括を企画した。初参加者同士が合宿で同室になるように手配してあるので、一緒に BoF へ参加してもらい、その中である特定のテーマを扱っているワーキンググループを取り上げ、実際見て感じたことについて総括してもらった。

本企画の目的は、所属の違う同世代の人達と議論し話しあう中で、横の繋がりを作り、同世代がどのような視点で物事を考えているかを知る機会を、合宿という実際に顔を合わせる環境で提供することである。さらに、それぞれのワーキンググループにとっては、BoF の内容を充実させる指標ともなり、初参加者が参加しやすい環境を作っていくことも期待できる。

合宿後のアンケートによれば、本企画に肯定的な初参加者は 71%、横の繋がりを作ることができたと答えた割合は 90%に登る。その一方で、企画内容は有用だが、個人的にはもうやりたくないと答えた初参加者が 28%も存在し、本企画がワーキンググループの活動に効果的だったと答えた参加者が 52%しかいないことから、やり方の改善が必要であることがうかがえる。

2.2.4 プレナリセッションの英語ログ

近年増えている外国からの WIDE 合宿参加者、および、来日間もなく、日本語にまだ習熟していない留学生などへの配慮として、プレナリセッションのログをリアルタイムに英語に翻訳し、部屋前方のス

クリーンに掲示する試みを実施した。

初めての試みだったこともあり、スクリーンの字が小さい、英語化の際の品質が保証されていない、などの問題が残った。

WIDE および WIDE 合宿の国際化に関しては、ここしばらく熱心に議論されている内容であり、WIDE の活動の質を落とさずに、バランスよく国際化を取り入れる努力が継続されている。

2.2.5 子供部屋の設置とプレナリセッションの中継

家庭の事情などにより、近年合宿に子供を同伴する参加者が現れた。これに伴い、ベビーシッターが不在となる夕方、夜のセッションにおいて、子供同伴の参加者が議論に参加できないという問題が新たに発生している。今回、その救済策として、SAMTK の協力の元、夕方および夜のプレナリセッションの内容を BoF 部屋に中継し、子供同伴参加者が子供の世話をしながら遠隔参加できる環境を実験的に提供した。

初めての試みだったこともあり、中継プログラムの問題、音質、画質の問題、離れた部屋から参加することの疎外感など、実際に実施してみて分かった点も多かった。アンケートによれば、73%の参加者がよい企画であると答えており、今後増えるであろう子供同伴参加者への対応策としての可能性を発見できた。本試みは合宿実験の一環として行われた。詳しい内容は、実験報告の節にて参照されたい。

2.2.6 プログラムまとめ

2008 年秋合宿では、「現地に集合したからこそできる活動をサポートする合宿」という目標のもとにプログラムを構成した。mini ワークショップ、新人総括による初参加者のサポートなど、顔を合わせることで実現できる内容を実現でき、それぞれ成果を挙げることができた。以後のプログラム委員は、今回のアンケートを通じて得られたフィードバックを反映して、より効果的な合宿にとりくんでいってもらいたい。

2.3 ネットワークおよび実験報告

本ドキュメントでは、2008 年 WIDE 秋合宿におけるネットワークと各実験の報告を述べる。

2.3.1 概要

本合宿では、

- 模倣インターネット環境の実インターネット環境への統合実験
- XCAST + SAM-TK
- P2P ネットワークを用いた写真の共有
- IDS の試験とボット検出用に作成したルールの評価
- 映像・音声による遠隔地コミュニケーションの実験

の 5 件の実験が実施された。これらの実験実施者からの、合宿ネットワーク設計に対する要求事項に応えるため、合宿に先立って行われた 2008 年 WIDE 6 月研究会より、PC が実験チームからのヒアリングを開始し、ネットワーク設計を行った。これらの実験は、近年の WIDE 合宿で見かける事が少なくなった、複雑なレイヤ 2 及びレイヤ 3 トポロジを要求するものも含まれた。このため、合宿実施に先立ち、

- 合宿地での設営時間を短縮する事
- 各実験が確実に合宿ネットワークで展開できるようにする事

を目的として、慶応義塾大学湘南藤沢キャンパスセミナーハウスにて、仮設合宿ネットワークを構築するホットステージが開催された。

合宿期間中は、

- 合宿ユーザの利用形態に対し、対外回線の帯域が不足していたこと
- 新規の windows update 項目が発表され、大量の合宿地向けトラフィックが発生したこと
- アナウンス不足により、利用者側で、リソース不足のために解決不能な問題であるか、または解決可能な障害であるか、の判断が困難であったこと

より、合宿参加者にとっては、非常にストレスの大きい環境であっただろうことは、残念である。

一方で、各実験実施者からは、概ね当初計画していた実験計画を満たすことができたとの報告があったことは幸いである。

今後は、今回の反省点を踏まえ、よりよい合宿参加者の環境、および実験環境を構築していく。

2.3.2 ネットワーク構成

2.3.2.1 無線 LAN システム

本合宿において、IEEE802.11n 規格（Draft）に準拠した無線 LAN を運用し、利用する無線帯域における干渉状況や性能、そして、各 STA（クライアント）におけるサポート状況や利用上の問題点を調査した。

2.3.2.1.1 IEEE802.11n の運用構成

今回は、5 GHz/2.4 GHz の 802.11n に対応したアクセスポイント 1 台、802.11agb に対応したアクセスポイントを 9 台用意した。今回の調査では、11n 対応アクセスポイントが 1 台しか準備できなかったため、5 GHz 帯での運用とした。理由は、2.4 GHz 帯では、11bg 基地局が混在する場合における干渉制御が難しいことと、40 MHz 帯域（デュアルチャンネル）での運用は大規模な運用では帯域不足となるからである。

5 GHz 帯での運用は、40 MHz 帯（従来の約 2 倍の周波数帯域）を割り当て、最大 300 Mbps でのリンク接続が可能な構成とした。

2.3.2.1.2 運用結果

運用の結果、無線ネットワークへ接続した STA 数は、最大で 180 台同時接続（1 分毎に各アクセスポイントへ接続している STA の最大値）となり、延べ数で、266 STA の接続があった。

このうち 11a（5 GHz 帯）と 11bg（2.4 GHz 帯）の利用内訳は、約 1 : 1.5 となっており、2007 年 SIGCOMM Kyoto での運用結果では、1 : 2.7 であったことから、Intel Centrino 準拠デバイスセットの普及などにより、2.4 GHz/5 GHz 帯に対応した無線カードの普及が伺える。

また、この 5 GHz 帯を利用した STA の内、約 24% が、11n に対応した機器であった。2007–8 年にかけて、Centrino に含まれる Intel Wireless WiFi Link 4965AGN や、MacBook の普及結果ではないかと推測される。

以上より、802.11n は、決して、ニッチなメディアではなく、運用上考慮すべき通信メディアであるといえる。

次に、干渉状況を確認するために、2.4 GHz 帯および、5 GHz 帯をスペクトラムアナライザーで常時監視し、無線 LAN 以外の広帯域通信の有無などを調査した。その結果、2.4 GHz 帯では、一部 STA が、バーストでデータを伝送する際に、無線帯域幅以上に電波を送出し、複数のチャンネルにまたがって干渉し、他 STA においてアソシエーションが消失したり、通信が不能になったりする現象が発生した。当該 STA 製品と問題発生の相関性が確認されたため、機器の異常があると考えられ、ベンダーへのエスカレーションを行うこととなった。2.4 GHz では、Bluetooth などの存在も確認されたが、ビーコンタグや、コードレス電話といった、広い帯域を干渉する無線機器の影響は観測されなかった。

一方、11n を運用する 5 GHz 帯では、干渉はほぼ見られず、40 MHz 対応 STA と 20 MHz 対応 STA の利用状況に応じた利用率が得られる結果となった。

以上より、11n の運用は、大規模な環境であっても、帯域に余裕がある 5 GHz 帯においては、問題無く運用可能なことが確認できた。

2.3.2.1.3 今後の課題

本合宿では、対外帯域が 802.11n の無線帯域に比べて遙かに小さい上に、オンサイトのコンテンツも無かったことから、帯域を生かし切れる環境ではなかった。したがって、今後は、高い利用率が想定される環境での試験が必要とされる。

2.3.2 レイヤ 2 およびレイヤ 3 の構成

2.3.2.2.1 合宿ネットワークの構成

合宿基幹ネットワーク L3 トポロジーを図 2.1 に示す。

本合宿では、各実験グループの実験実施に伴い、以下の要求事項を満たすよう設計を行った。

- AS レベルで IPv4 を用いたインターネット環境を模倣する実験のため、合宿地内に、プライベート AS を用いた複数 AS を構成すること。
- XCAST6 を用いる実験のため、IPv6 アドレスをもつ複数セグメントを構成すること。

また、トラブル時や構成変更時への柔軟性を持たせるため、基幹ネットワーク上のルーティングは、PC ルータを用いることとした。なお、この PC ルータは、新しい技術への挑戦として、単一 PC 上に、仮想化技術を用いて、仮想 PC を複数台用意し、それぞれの PC 上でソフトウェアルータ機能を用いることで実現させた。具体的には、Dell PowerEdge 2950 上に、Linux（CentOS 5.2）を動作させ、その上に、Xen 3.2 on Linux（CentOS 5.2）を用いて、GW-AS ルータおよび User-AS0 ルータとして動作させた。

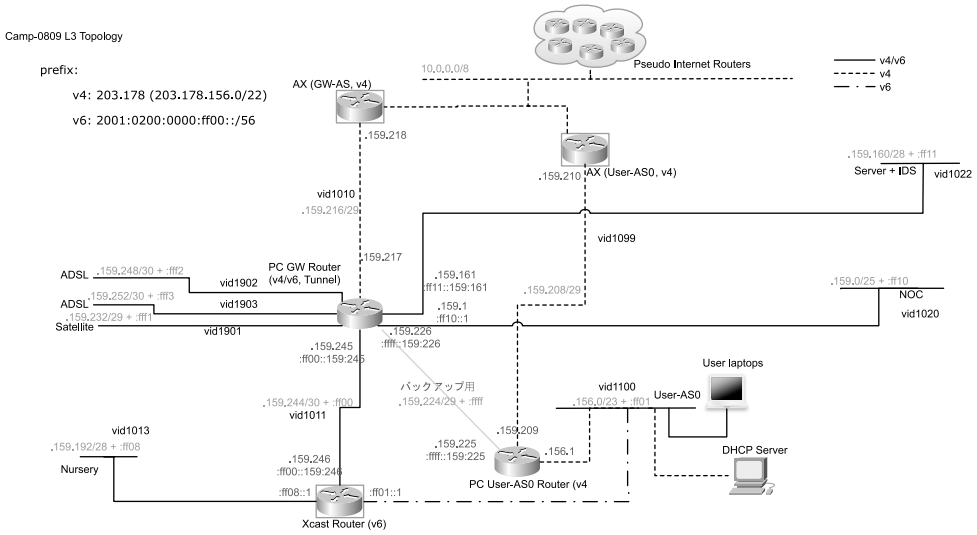


図 2.1. 2008 年秋合宿基幹ネットワーク L3 トポロジー

2.3.2.2.2 合宿ネットワークの運用

合宿ネットワークの運用に関する報告と、合宿期間中に報告された問題点とその解決策について述べる。

ADSL 回線 PPPoE 認証情報の誤りにより、ADSL 回線のうち 1 本の運用開始が合宿初日夜まで遅れた。また、合宿地にて ADSL モデムが NAT をしてしまうことが発覚したため、GW-AS ルータにより、PPPoE 接続を代替した。

GRE トンネル 本合宿では、ADSL 回線を用い、AS-GW ルータにおいて GRE による IP-in-IP トンネルを行ったため、Path MTU Discovery ブラックホール問題（RFC 2923）が発生した。この問題を解決するため、トンネルインターフェイスにおいて MTU の調整を行い、さらにトンネル対地の藤沢 NOC にて経路の変更および DF ビットをクリアする設定を行うことで、ネットワークの安定化を図った。また、藤沢 NOC 内にて duplex mismatch による 30%以上のパケットロスが観測されたため、該当機器の設定変更を行い、解決した。

実験ネットワークとの相互運用 本合宿では、IPv4 ネットワークは模倣インターネット実験、IPv6 ネットワークは XCAST 実験との相互接続を行い、合宿ネットワークとして運用した。

IPv4 ネットワークについては、ホットステージ及び合宿前日での事前準備および確認を十分に行ったため、円滑に運用を行うことができた。

IPv6 ネットワークについては、合宿初日は経路の切り替え作業が円滑に行われなかったことにより、接続性が不安定になってしまったが、それ以降は円滑に運用を行うことができた。

Policy-based routing 本合宿では、対外線として ADSL 回線を 2 本、衛星回線 1 本を利用し、これら 3 本の対外線を有効利用するために、policy-based routing を行った。なお、routing policy は、回線利用率を監視しながら適宜変更を行った。

2.3.2.2.3 今後の課題

本合宿では、ホットステージ中に準備を行ったものに関しては、円滑な運用を行うことができたが、合宿地にて新たに発生したパケットロス等の問題へのデバッグにかなりの時間を要する結果となってしまった。既知の問題に対して迅速に対応し、新たな課題に取り組む時間を取れるよう、発生した問題をドキュメント化し、継承する必要がある。

2.3.2.3 衛星回線

2.3.2.3.1 システム概要

合宿ネットワークにおける対外接続として、慶應義塾大学湘南藤沢キャンパス（SFC）と合宿地（信州松代ロイヤルホテル）を双方向衛星回線で接続し、合宿地から WIDE 藤沢ネットワークを経由したインターネット接続性を提供した。地球局は VSAT 可搬局を使用し、伝送速度はアップリンク・ダウンリンクともに 768 kbps とした。合宿ネットワーク、およ

表 2.1.　合宿コンテンツ閲覧における認証方式

コンテンツ	認証方式
デフォルト	証明書認証 or Basic 認証
アンケートシステム（要 moCA 証明書）	証明書

び WIDE 藤沢ネットワークの基幹ルータでは、衛星回線を Ethernet のデータリンクとして収容し、2 系統の ADSL と併用可能な対外接続回線として運用した。

2.3.2.3.2 本合宿での注意点

衛星回線の運用にあたり、地球局の構築手順や運用情報を整理したマニュアルを作成し、効率的な運用ができるよう工夫した。また、今回の合宿では、設営前日より SFC での回線運用を開始した。合宿地での設営作業において、SFC から送信されたキャリアの受信状況が捕捉対象衛星の選別・確認の参考にできるため、より確実な衛星捕捉と設営作業の効率化が実現できた。

2.3.2.3.3 今後に向けて

今回作成したマニュアルを活用し、未経験者を含めた衛星回線運用のトレーニング環境を充実させる予定である。

2.3.2.4 サーバ構成

本合宿では、以下のサーバを用いて合宿ネットワーク利用者へサービスを提供した。

機種	CPU	メモリ
IBM System x3250	Intel Pentium D 945 3.40 GHz	512 MB

2.3.2.4.1 サービス提供

合宿ネットワーク内での、アドレスアサイン、ドメイン名解決、合宿情報・スケジュール表示、アンケートシステム運営のサービス提供マシン（generic サーバ）を設置した。generic サーバで使用したプログラムパッケージは、`isc-dhcpd-4.0`（DHCP）、`bind-9.5.0-P2`（DNS）、`httpd-2.2.9` + `php-5.2.6` + `perl-5.8.8_1`（WEB）である。

合宿情報・スケジュールは `www.wide.ad.jp` の WIDE CAMP 2008 Autumn のコンテンツと同期させ、`www.camp.wide.ad.jp` のトップコンテンツとして提供した。合宿プログラム表示用の PC や、携帯電話などの証明書を持たない（動作が不明な）端末の使用を想定し、moCA 証明書の使用に加え、Basic 認証を用いてアクセスを制御した（表 2.1 参照）。

2.3.2.4.2 反省点

合宿初日は WEB コンテンツの同期方法が未定のままであり、WEB サービス提供ができず、合宿プログラムの時間、開催部屋を参加者に十分に伝えられなかった。また、アンケートシステム公開後、一部動作しなかったページがあった。ホットステージ実施の段階において運用方法を明確に決定し、利用されているコンテンツが確実に動作することを確認するべきだった。

2.3.2.5 計測システム

本合宿では、ネットワーク機器やサーバの障害に素早く対応するために、Nagios による定期的な生存チェックを行った。また、ネットワークの使用状況を cacti によって計測・グラフ化した。これらの情報は本合宿ウェブページで参加者にも公開した。計測・監視に使用したマシンのスペックは以下の通りである。

機種	CPU	メモリ	消費電力
HP ProLiant DL320 G3	Intel Pentium4 3.2 GHz	4 GB	350 W

2.3.2.5.1 留意点

Nagios による生存チェックは、計測対象機器やサービスがダウンした場合にはメールで通知するように設定した。合宿中は特に問題なく動作し、ネットワーク運用状況を把握することができた。

cacti によるネットワーク使用状況の計測・グラフ化は、各スイッチの全ポートに対するトラフィック量と、アクティブな DHCP クライアント数について行った。トラフィック量については、最初は全ポートのグラフを番号順に単に並べて表示していたため、合宿ネットワーク外やサーバ間に流れるトラフィック量など重要な情報がわかりにくかった。合宿中に、これらの重要だと思われるグラフをラベル付けしてまとめることで見やすさの改善を図った。

2.3.2.5.2 トラフィックの傾向

9/11 12:00–9/12 11:00 における対外線のトラフィック量の推移を図 2.2、2.3、2.4 に示す。深夜から早朝にかけてと夕食時（18:00 ころ）にはほぼ外へのトラフィックは流れていない。しかしそれ以外の時間帯については、ADSL の下り帯域（Inbound）の大部分が消費されている。実際、参加者から対外線を使用する通信が重いといった声が上がっていた。よって、ADSL 2 本では本合宿規模のユーザ数に対して帯域が不足していたと言える。

NOC スイッチと generic サーバ間のトラフィック量の推移を図 2.5 に示す。generic サーバから発生するトラフィック量はせいぜい数 100 kbps ほどで、帯域には十分余裕があった。

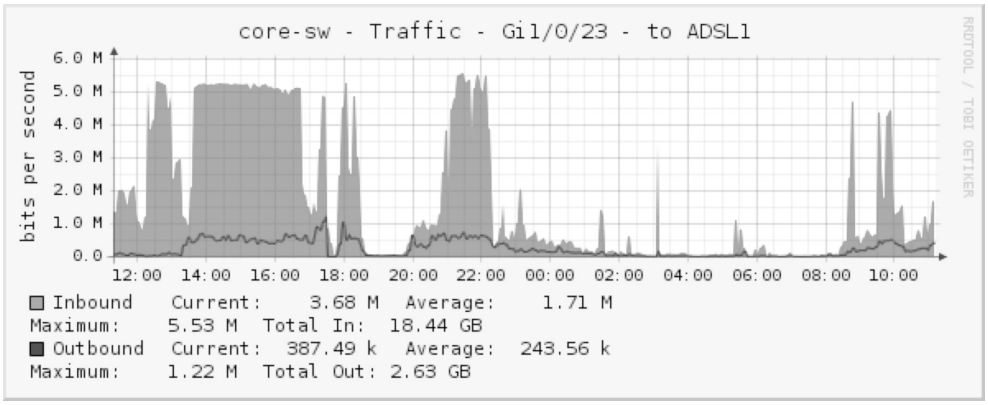


図 2.2. 9/11–12 におけるコアスイッチ–ADSL1 間のトラフィック

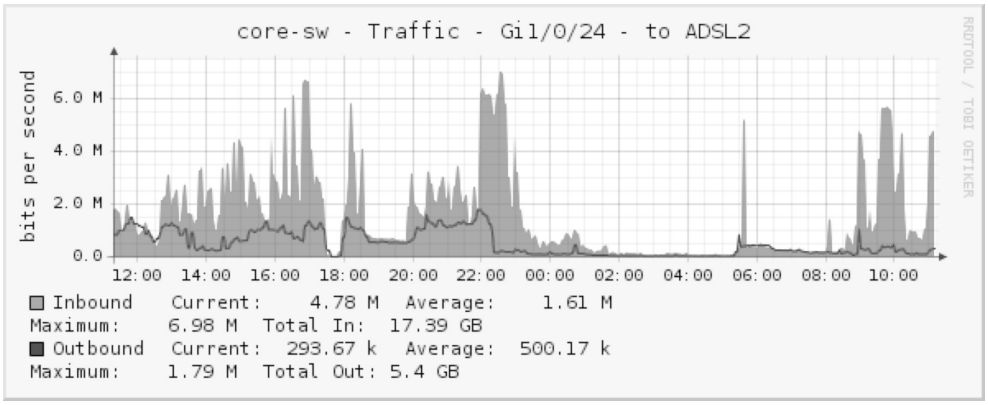


図 2.3. 9/11–12 におけるコアスイッチ–ADSL2 間のトラフィック

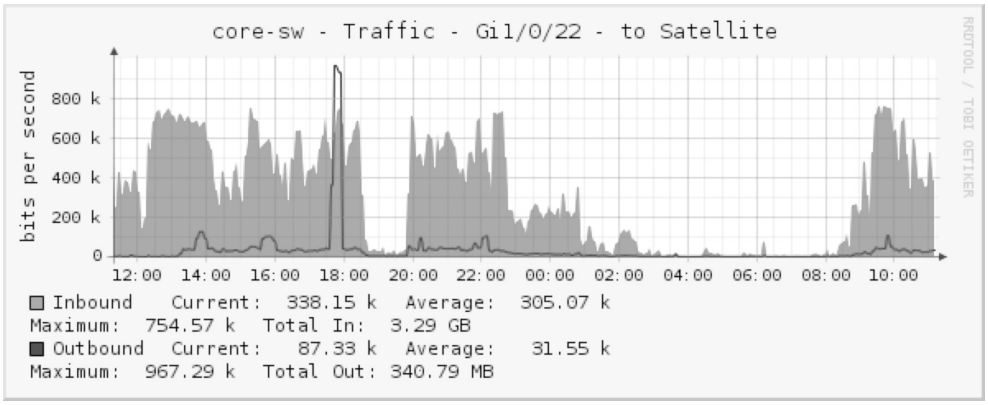


図 2.4. 9/11–12 におけるコアスイッチ–衛星間のトラフィック

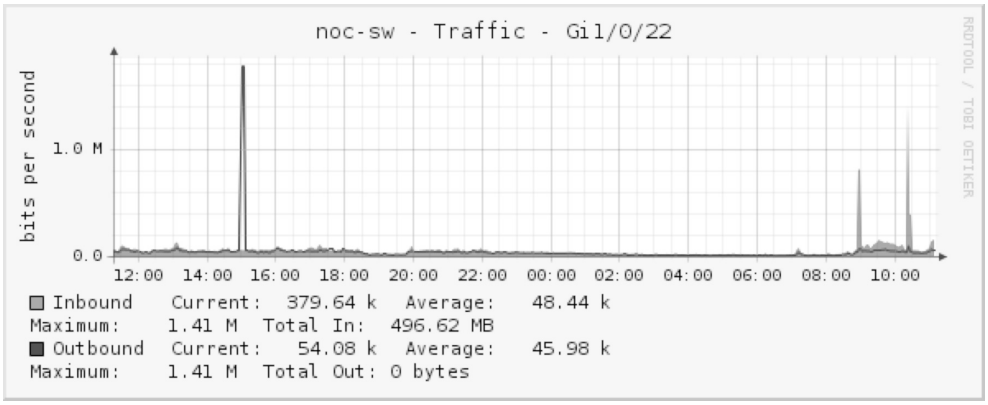


図 2.5.　9/11–12 における NOC スイッチ–generic サーバ間のトラフィック

2.3.2.5.3 アクティブな DHCP クライアント数

9/11 18:00–9/12 10:00 における合宿参加者セグメント（203.178.156.0/24 および 203.178.157.0/24）のアクティブな DHCP クライアント数の推移をそれぞれ図 2.6、2.7 に示す。

これらは合宿ネットワーク利用者に配布したアドレス帯である。割り当てられたアドレスが解放されるまでの時間を考慮する必要があるが、おおよその IPv4 ユーザ数の推移を知ることができる。

2.3.2.5.4 反省点

今回計測した項目以外に、無線のアクセスポイントごとの接続ユーザ数や IPv6 のユーザ数を計測・グラフ化すると、運用状況のより詳細な分析ができると思われる。また、各ポートごとのトラフィック量を表示するだけでなく、ネットワークウェザーマップを用いるなど、どのサーバからどのような方向・量でトラフィックが流れているかをわかりやすく可視化する方法を検討するとよいと思われる。

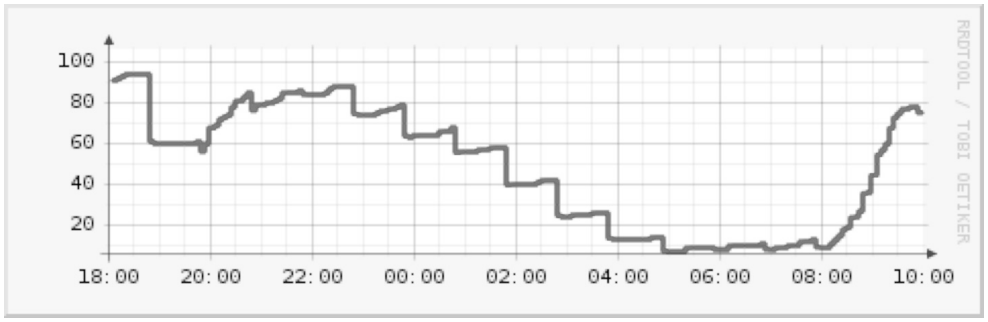


図 2.6.　9/11–12 における 203.178.156.0/24 のアクティブな DHCP クライアント数

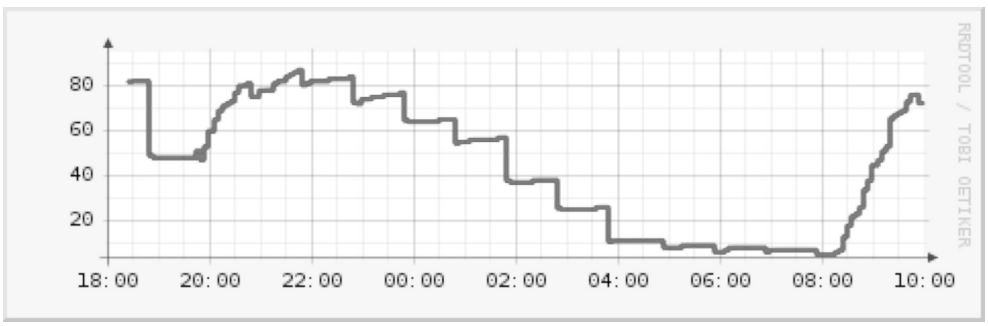


図 2.7.　9/11–12 における 203.178.157.0/24 のアクティブな DHCP クライアント数

2.3.2.6 電源構成

本合宿ではプロジェクタやL2スイッチ、Wirelees APなどの電気供給が停止すると顕著に影響のある機器と、一般ユーザが使用するノートPCを別の電源系統に接続する構成とした。また、NOCや実験では事前に容量を報告していただき、必要容量を算出した。NOCや実験では急遽追加の電源が必要になることが考えられるため、算出した必要容量よりも多くの容量を想定した。その結果、実際に追加の電源容量が必要になったが柔軟に対処することができた。他方、電源タップの配置については、電源系統図に加え各会場の配置する電源タップの設置場所を示した図を作成した。そのため、会場準備者が電源タップの数をその場で考える必要がなくなり、効率的に電源タップを配置することができた。結果として、合宿中は全部屋で電源供給が停止することなく無事に終了することができた。

2.3.2.6.1 Plenaryルーム

Plenaryルームでは電源が7系統ある。そのうち、NOCルームや実験ルームに近接している2系統をNOCルーム、実験ルームの系統として割り当てたためPlenaryルーム用には電源5系統を使用し、加えて追加電源を2系統用意した。ノートPC接続用の電源タップをできるだけ前方に配置し、できるだけ前方に着席していただくことを狙った。しかし、実際には前方で電源タップが余っているにも関わらず多くのノートPC使用者が後方に着席していた場面もあった。また、後方に電源タップが少なかったという指摘もあった。今後は後方にも電源を配置する工夫が必要である。

2.3.2.6.2 BoFルーム × 4部屋

BoFルームの4部屋のうち1部屋のみが広めの部屋で4系統、残りの3部屋は2系統の電源を使用した。Plenaryルームと同様に全ての部屋でノートPC接続用の電源タップを前方に配置した。また、広めの部屋には他の部屋と比べ電源タップを多く配置した。しかし、広い部屋で開催されたBoFの中には人数が少ないものがいくつかあり、反対に狭い部屋で開催されたBoFの中には人数が多く人が溢れているものがいくつかあった。そのため、狭い部屋で開催された人数の多いBoFからは電源が足りないなどの指摘があった。今後は、人が入りそうなBoFは広めの部屋で開催していただく工夫や、狭い部屋にも多くの電源タップを配置するなどの工夫が必要である。

2.3.2.6.3 Boardルーム

Boardルームでは2系統の電源を使用した。Boardルームではできるだけどこに着席しても電源コードを接続できるように電源タップを配置した。概ね問題なく終了できた。

2.3.2.6.4 ホワイエ

ホワイエには空気清浄機とプロジェクタのみに電源タップを使用することを想定し、多くのノートPCが接続できないよう口数の少ない電源タップを配置した。また、ノートPCを接続する可能性もあるため、プロジェクタと空気清浄機で電源系統を分けた。

2.3.2.6.5 NOCルーム（Plenaryルームの後方に設置）

あらかじめNOCルームで使用する容量を計算し、6.9 Aであったため余裕をみて1系統の追加電源を用意した。原則として用意した追加電源を利用する構成とした。電源タップも事前に計算した口数よりも多めに用意した。その結果、急遽電源が必要となった場合にも問題なく対処できた。

2.3.2.6.6 衛星アンテナ設置場所

あらかじめ使用する容量を計算し、10.5 Aであったため余裕をみて1系統の追加電源を用意した。原則として用意した追加電源を利用する構成とした。電源タップも事前に計算した口数よりも多めに用意した。

2.3.2.6.7 実験ルーム（Plenaryルームの後方に設置）

あらかじめそれぞれの実験について使用する容量を計算し、Plenaryルームの電源1系統と追加電源3系統を用意し、各実験を割り当てた。しかし、ホテル側に追加電源注文後に事前に計算した容量よりも多くの容量が必要であることがわかったため、再度割り当てが必要となった。再割り当てを行った結果、追加電源注文前の容量計算時に余裕をもって電源を割り当てていたため、さらに追加電源を注文することなく電源を割り当てることができた。電源タップも事前に計算した口数よりも多めに用意したため、当日は問題なく終了できた。

2.3.3 実験

2.3.3.1 模倣インターネット環境の実インターネット環境への統合実験

ネットワークエミュレーションテストベッドでの実験手法の確立や、各種の仮想化技術の革新によって、実インターネットに規模的にも機能的にも近い環境がテストベッド上で簡単に構築できるようになりつつある。このような模倣インターネット環境構築技術の発展を前提に、模倣インターネット環境と実インターネット環境の結合を想定すると、さまざまな運用上、セキュリティ上の問題が存在すると考えられる。そこで、本実験では合宿ネットワーク内に模倣 eBGP トポロジを構築し、ネットワークの安定運用に必要な知見やセキュリティ上の課題の洗い出しを行った。

2.3.3.1.1 実験の背景と目的

我々は、これまで、実機（物理ノード）によるネットワーク実験環境での実験手法 [119, 176] や仮想化技術を用いたネットワーク実験環境の研究開発を行ってきた。また、インターネット実証実験の代わりに利用できる実験環境を目指して、インターネットを模倣した環境をネットワーク実験環境上に構築する技術 [65, 66, 231] についても、研究開発を行っている。これらの模倣インターネット環境構築技術の進展により、模倣インターネット環境はすでにいくつかの実験で利用されており、模倣インターネット環境を実験に利用する場合や、模倣インターネット環境を実インターネットと結合する場合の運用上、セキュリティ上の問題を明らかにすることが求められている。そこで、模倣インターネット環境を実際に運用されているネットワークに結合し、その性能の評価と、運用上やセキュリティ上の問題を明らかにすることを目的として、実験を行った。

2.3.3.1.2 実験環境

本実験では、合宿ネットワーク内に模倣インターネット網を構築し、実験を行った。

合宿ネットワークの参加者向けの無線 LAN の一つ（ESSID = WIDE）と NOC ネットワークとの間に、構築した模倣インターネット網を挟み込む形で構成した。これにより、合宿参加者がこの無線 LAN を用いて合宿ネットワークより外のインターネットなどと通信する場合には、必ず NOC ネットワークから WIDE-BB を介し、模倣インターネット網を経由することになる。概要を図 2.8 に示す。

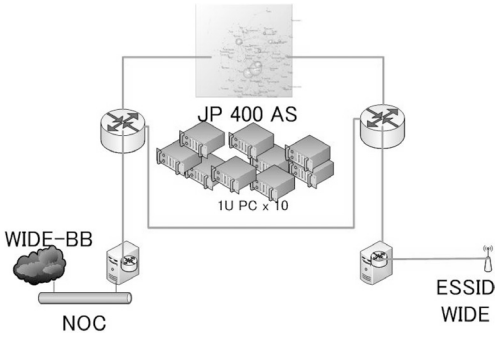


図 2.8.　模倣インターネット実験構成概要図

実験に用いた機器は、以下の通り。

- 実際に模倣インターネット網を構築する Mohou Slave Server（10 台）
- 模倣インターネット網を管理するための Mohou Master Server
- 模倣インターネット網を観測するための Mohou Measurement Server
- 模倣インターネット網を監視および状態の閲覧をするための Mohou bgpd Console
- 模倣インターネット網を通過するパケットの収集をする Packter Capture Engine
- 模倣インターネット網を通過するパケットを可視化する Packter Viewer
- 無線 LAN、NOC ネットワークと接続するための L3 Switch（AX-USER-AS、AX-GW-AS）
- これらをつなぐ L2 Switch

物理接続を図 2.9 に示す。Layer 3 の接続を図 2.10 に示す。

今回用いた模倣インターネット網は、実際のインターネットの観測データに基づく AS 間ネットワーク（eBGP トポロジ）を模倣した模倣 eBGP トポロジである。CAIDA AS Ranking プロジェクト [33] で公開している AS の接続関係を推定したデータセット（以降、AS 接続関係データセット [23]）の 2008 年 7 月 21 日時点のデータのうち、JPNIC の 2008 年 8 月 8 日時点の AS 一覧に記載されていて、接続関係がある 445 AS を、AS 接続関係データセットの接続関係に基づいて模倣した。AS65504 を無線 LAN の AS とし AS23793 に、AS65502 を NOC ネットワークの AS とし AS23632 に接続した。トポロジ図を図 2.11 に示す。

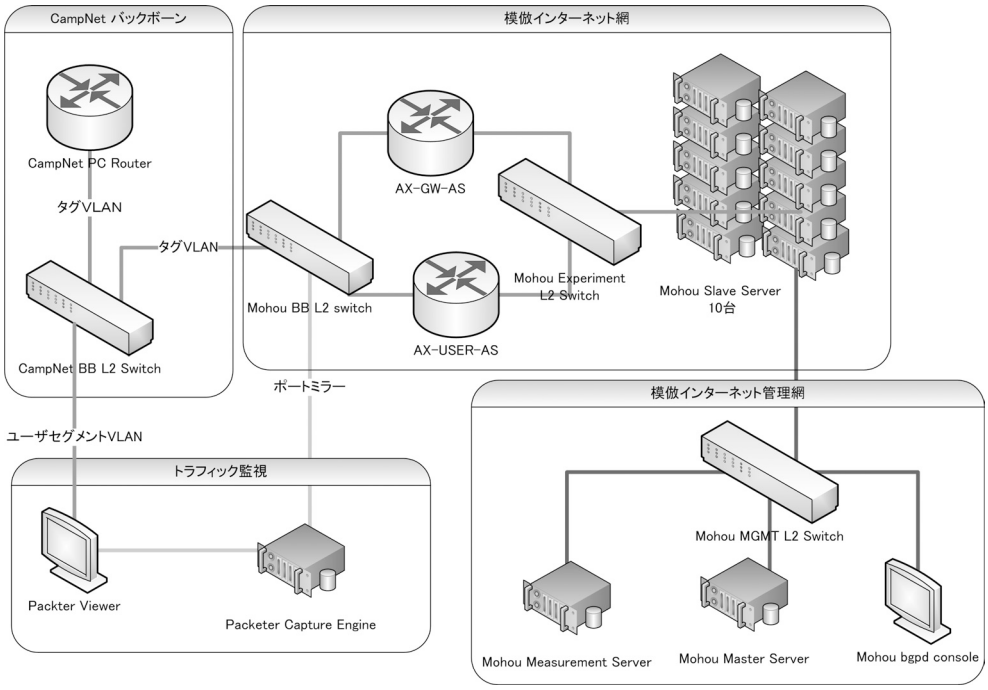


図 2.9. 模倣インターネット網と合宿ネットワークとの物理接続図

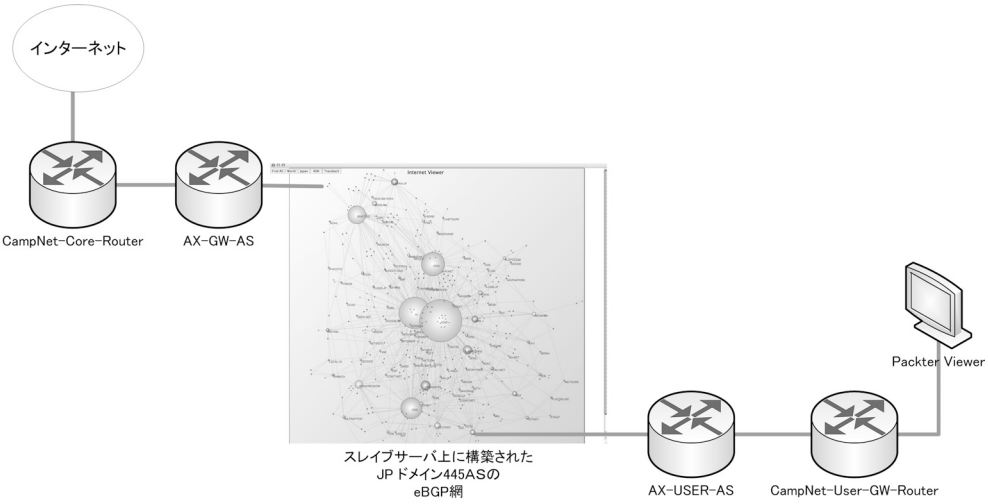


図 2.10. 模倣インターネット網と合宿ネットワークとの Layer 3 接続図

2.3.3.1.3 実験

前節に示した実験環境を用いて実験を行った。

下記の実験を行う予定であった。

1. 模倣インターネット網を経由した場合の性能
2. 模倣インターネット網の可視化
3. 模倣インターネット網を経由するパケットの可視化
4. 模倣インターネット網上の複数の経路を利用した場合のそれぞれの性能
5. 模倣インターネット網の構成を変更した場合の収束までの状態
6. 模倣インターネット網の IPv6 対応確認

実際には、NOC ネットワークから WIDE バックボーンへの接続点が不安定であったため、1〜3 のみを行った。また、6 については、実装上の不具合により、今回は実施しなかった。

模倣インターネット網を経由した場合の性能については、模倣インターネット網の入口（無線 LAN 側、AS0 と表記）と出口（NOC ネットワーク側、GW-AS と表記）で、Throughput や RTT、パケットロス率

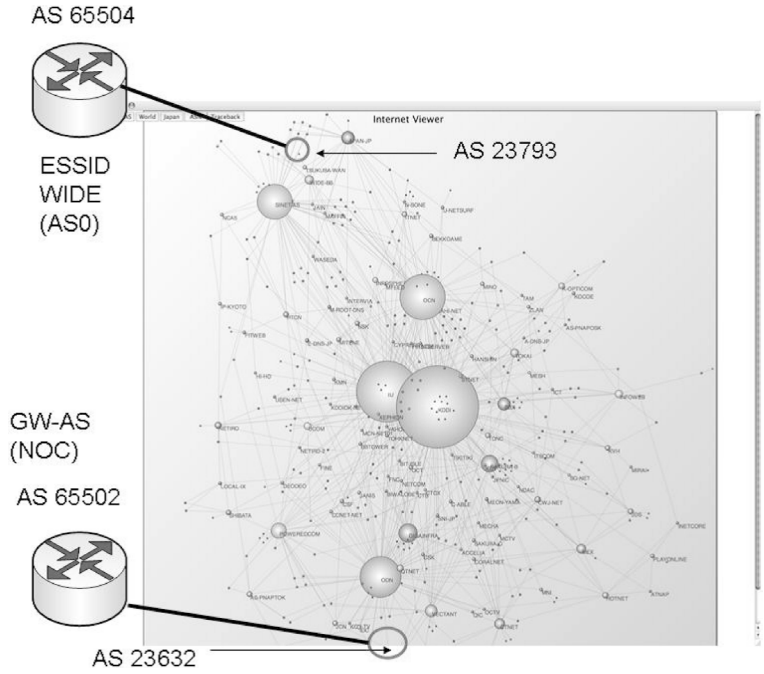


図 2.11. 模倣 eBGP トポロジ（AS 間ネットワーク）図

```
ve1099-pseudo-as0-router.camp.wide.ad.jp \
(203.178.159.210) 1.558 ms  1.337 ms  1.362 ms
10.3.168.1        1.200 ms  1.701 ms  1.130 ms
10.0.227.1        1.485 ms  1.269 ms  1.631 ms
192.168.0.1       1.319 ms  1.562 ms  1.538 ms
192.168.0.11      1.859 ms  2.770 ms  2.630 ms
10.3.250.1        2.083 ms  1.912 ms  2.774 ms
10.2.137.1        2.770 ms  2.086 ms  1.863 ms
10.2.212.202      2.611 ms  2.402 ms  2.776 ms
ve1010-pc-gwas-router.camp.wide.ad.jp \
(203.178.159.217) 6.348 ms  2.368 ms  3.499 ms
```

図 2.12. traceroute: AS0 → GW-AS

```
ve1020-gw \
(203.178.159.1)   1.943 ms  0.894 ms  0.967 ms
ve1010-pseudo-gwas-router \
(203.178.159.218) 2.112 ms  1.444 ms  1.847 ms
10.2.212.1        1.450 ms  1.215 ms  1.119 ms
10.2.137.2        1.433 ms  1.983 ms  1.563 ms
10.3.250.2        1.828 ms  1.538 ms  1.723 ms
10.1.20.1         1.819 ms  1.557 ms  2.262 ms
192.168.0.35      2.273 ms  1.933 ms  2.279 ms
10.0.227.2        2.015 ms  2.414 ms  2.368 ms
10.3.168.204      3.270 ms  2.774 ms  2.608 ms
ve1100-gw \
 (203.178.156.1)  2.327 ms  2.242 ms  3.295 ms
```

図 2.13. traceroute: GW-AS → AS0

を計測した。模倣インターネット網の可視化では、開発した可視化ツールにより、AS 間接続と AS の死活を可視化した。模倣インターネット網を経由するパケットの可視化では、開発した可視化ツールにより、AS0 を出入りするパケットを可視化した。

以下に、模倣インターネット網を経由した場合の性能について示す。Throughput は、`netperf` による計測で 30 から 60 Mbps、`iperf` による計測で 90 から 100 Mbps であった。RTT は、図 2.12 と図 2.13 に `traceroute` の結果を示す。パケットロス率は、`ping` による計測で 0%であった。なお、参考までに、模倣インターネット網の出入口（図 2.14）とトラフィックが実際には通らない AS（図 2.15 左）、通る AS（右）の 3 日間のトラフィックをグラフに示す。

2.3.3.1.4 評価と課題

実験の 3 日間を通して、模倣インターネット網は安定しており、性能も 100 Mbps 程度の網を、パケットロス無し、かつ、特に大きな遅延無しで提供可能であることが確認できた。

今後は、今回実施できなかった、複数の経路を利用する場合や、構成変更を行う場合、IPv6 対応などについて、実験を行いたいと考えている。また、現在は eBGP トポロジのみを模倣しているが、AS 内のネットワークについても模倣手法を確立し、実験を行いたい。

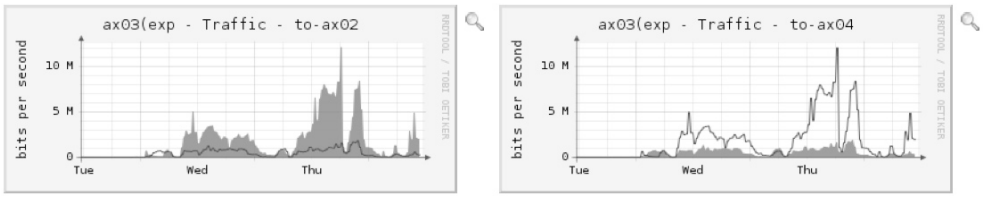


図 2.14. 模倣インターネット網の出入口のトラフィックグラフ（3 日間）

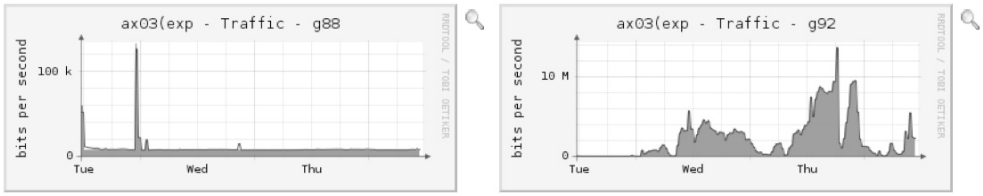


図 2.15. 模倣インターネットのうち、トラフィックが通らない AS（左）と通る AS のトラフィックグラフ（3 日間）

2.3.3.2 XCAST + SAM-TK

XCAST (eXplicit Multi-Unicast) has been proposed as an alternative point-to-multipoint communication protocol to alleviate the scalability problems of conventional group-based IP multicast. Active research is ongoing on the implementation of XCAST in both IPv4 and IPv6. XCAST6 refers to the implementation of XCAST on IPv6. Learning from the challenges encountered in the deployment of XCAST6 version 1.0, a redesign of the XCAST6 headers has resulted in XCAST6 version 2.0. This has in turn prompted for a reconsideration of the routing strategy for this protocol. This experiment was therefore to investigate the feasibility of designing, implementing and evaluating an out-of-the-box solution for routing of XCAST6 packets aimed at simplifying real-world deployment of XCAST6. This out of the box routing solution is what we refer to as XCAST6 Routing Engine.

2.3.3.2.1 Background and Objectives of the experiment

Point-to-multipoint communication has become very vital in multimedia content transmissions in the modern-day applications such as IP telephony, videoconferencing, e-learning, online gaming and real-time distribution of stock quotes in the financial markets. IP multicast has been the protocol of choice in such scenarios and has performed well in terms of scalability where huge multicast groups are needed. However, in scenarios where a temporary but large number of small, distinct groups are needed, IP multicast's scalability is limited by the availability of the number of multicast addresses that can be allocated. XCAST has since been proposed to provide an alternative in scenarios where a scalable number of small multicast sessions are needed.

XCAST (eXplicit Multi-Unicast) is a point-to-multipoint communication protocol which in contrast to the conventional group-based multicast explicitly specifies the destination addresses as a list of unicast addresses embedded in the IP packet header. The implementation of XCAST on IPv6 is referred to as XCAST6. Currently existing is XCAST6 version 1.0 while XCAST6 version 2.0 implementation is underway. XCAST6 version 2.0 currently is under testing on FreeBSD. The main difference between the two versions of XCAST6 is that XCAST6 version 1.0 uses hop-by-hop options header for routing while XCAST6 version 2.0 eliminates the use of hop-by-hop options header. This in turn motivated a re-consideration of an alternative method of routing of XCAST6 packets hence we proposed the XCAST6 Routing Engine.

The objective of this experiment was therefore

to investigate the feasibility of using our proposal to achieve the routing of XCAST6 packets in a situation where the network core router is not XCAST-aware.

2.3.3.2.2 The Experiment and Testbed Description

To implement the XCAST Engine, an XCAST-aware node is connected to the network core router such that upon receiving XCAST6 packets, the core router forwards them to the XCAST Engine for processing after which the engine sends back the packets to the core router for onward transmission. This concept is illustrated in the diagram 2.16.

To conduct this experiment we had the following set up,

- 4 PCs each running FreeBSD 6.2
- XCAST6 version 2.0 for FreeBSD installed in all the 4 PCs
- Juniper Jseries router (J2320) running JUNOS 9.1
- Tcpdump version 3.9.4 with libpcap 0.9.4 installed in all the 4 PCs
- Nagios from the camp-net was also used for performance measurements.

Our Juniper router was the core router for IPv6 network (marked as XCAST-Router) in the camp-network topology. To test the routing of XCAST6 packets each of the 4 PCs were placed in different segments and then used in sending of the XCAST packets. At the same time, tcpdump was being run from the PC designated as the XCAST Engine. The testbed is illustrated in Figure 2.17.

The XCAST6 Engine was also designed by implementing numerous virtual interfaces through the vlan tagging features of FreeBSD and Juniper's JUNOS.

To undertake the measurements, we developed two small programs (Xcastsend and Xcastreceive) for sending and receiveing of XCAST6 packets. We further implemented shell scripts to ensure that the Xcastsend program ran at 1 second interval.

On the Juniper router, we implemented a filter-based-forwarding policy for routing of XCAST6 packets based on the XCAST traffic class and applied the filter to all inbound interfaces of the router.

On the XCAST Engine, we used tcpdump to

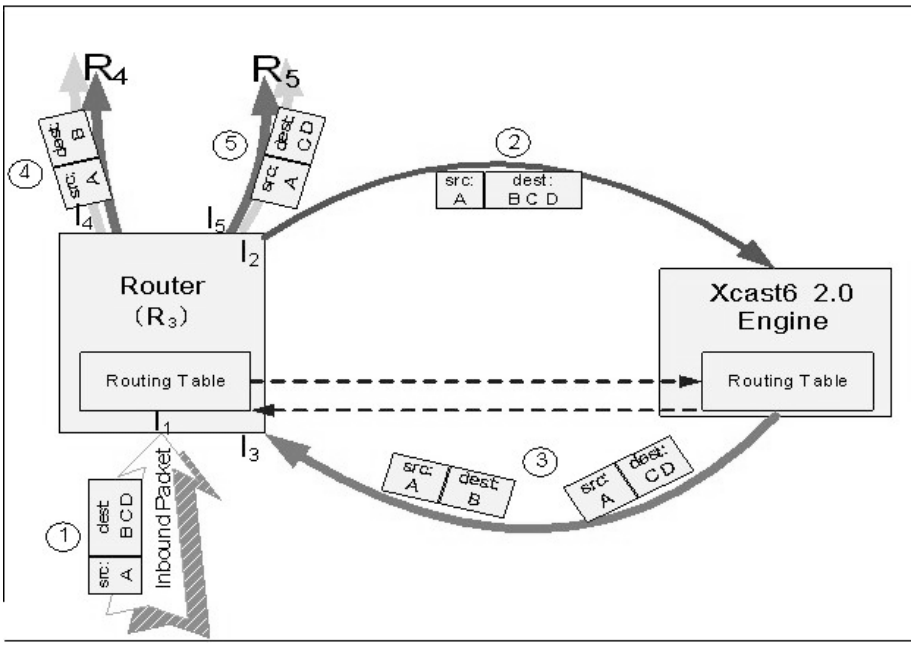


Fig. 2.16. xcast concept

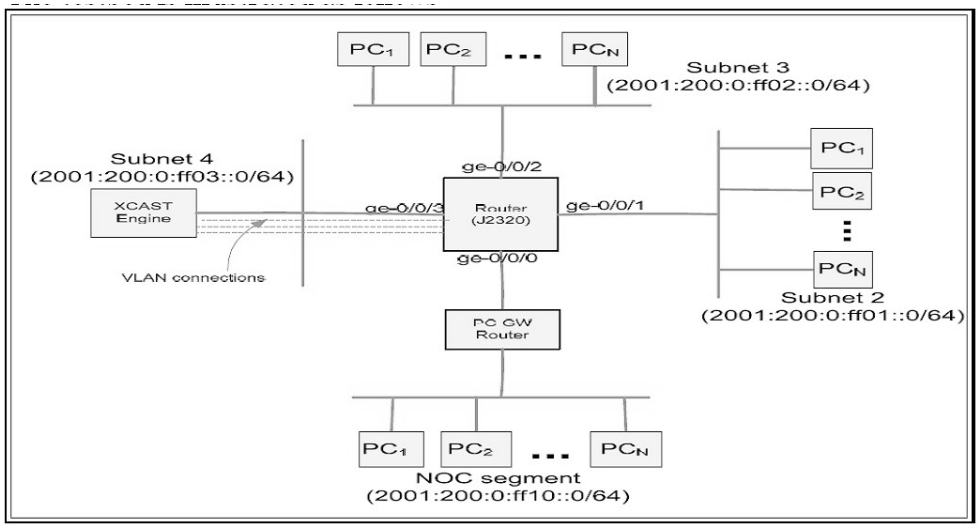
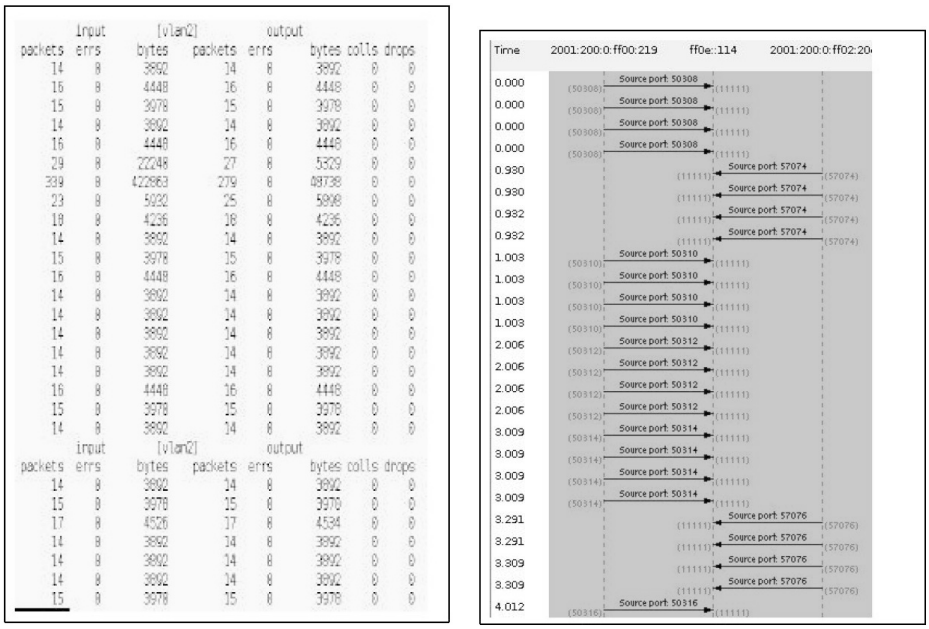


Fig. 2.17. xcast testbed



(a) result1 (b) result2

Fig. 2.18. sample result

capture the XCAST6 packets and write them into a file which we later analyzed to assess metrics of interest in the experiment.

In left hand figure in Figure 2.18 shows the packet loss statistics while the right one shows the conversation list as captured by tcpdump. This conversation list confirms that XCAST6 packets to all host were actually being routed via the XCAST6 Engine. Additionally, Figure 2.19 shows a sample response time from the experiment. We noted that the XCAST Engine realized a mean response time of 34 micro-seconds.

2.3.3.2.3 Conclusion

In this experiment, we investigated the concept of routing of XCAST6 packets through an XCAST6 Routing Engine. We proved that it works using two small programs sent to hosts

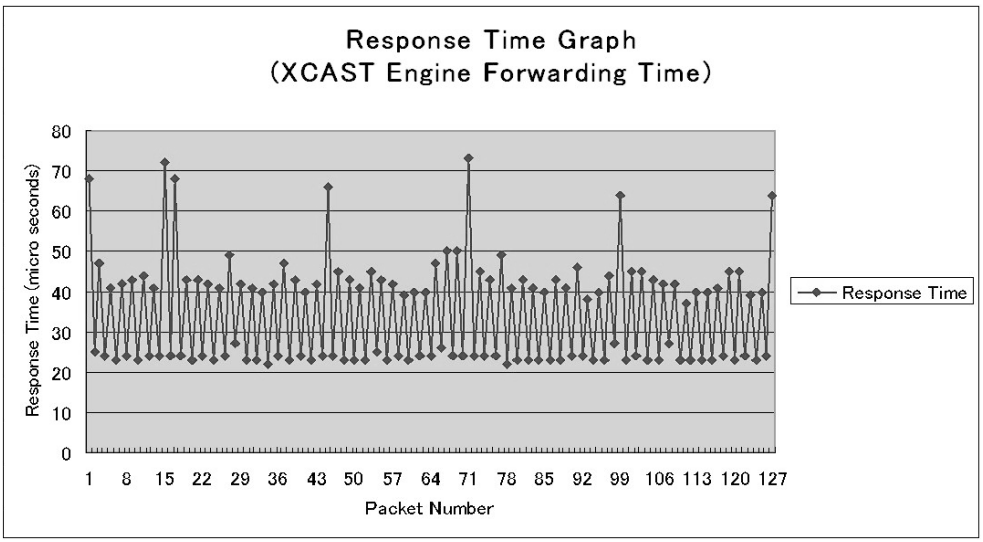


Fig. 2.19. Responce Time Graph

located in disparate segments of the WIDE camp network. We are therefore in the process of developing a real-world application for further testing and we intend to use it to investigate the feasibility of real-world deployment of XCAST protocol.

2.3.3.3 P2P ネットワークを用いた写真の共有

本実験は合宿で撮影された写真を P2P ネットワーク上で共有し、無線 LAN 環境下での P2P ネットワーク構築や、アプリケーションの有用性などを評価した。実験参加者には PC にアプリケーションをインストールしてもらい、各個人が撮影した写真を実験参加者間で共有する。実験中には写真の枚数・トラフィック総量・検索効率・離脱時の状況などを評価した。以下に詳細を述べる。

2.3.3.3.1 実験背景

デジタルカメラの高性能化と低価格化により、高画素・高精細な写真データを身近に利用できるようになってきている。また、カードメディアなどの高容量化により、一度に撮影できる写真の枚数が飛躍的に増加している。さらに現像コストがなくなり、印刷をしないまま写真を閲覧するなど、撮影することに対する障壁が減りつつある。これらの現状を受けてユーザの持つ写真の容量は増加を続けている。しかし、インターネット上におけるサーバ・クライアント型の写真共有サービスでは、こうした容量の増大に伴い管理コストも増大するため、1 枚あたりの容量やアップロード可能な枚数を制限することが行われている。この問題に対処するため、P2P ネットワークを用いて写真を共有するソフト「PSP2P」を開発し、本合宿にて実験を行った。

PSP2P では高画素な写真を取得可能にするため、サムネイル画像と撮影時刻を用いた写真の共有を行う。写真を取得可能にするためには、検索可能にする必要がある。このとき、検索結果として写真の概要が分かるサムネイル画像であっても検索結果を確認する為には問題がない。PSP2P ではこの点に着目し、サムネイル画像のみをあらかじめ共有し、検索結果として表示する。また高画素写真を取得したい画像に関してはその写真を保持しているユーザのコンピュータに対して直接クエリーを送信し、写真を取得する。サムネイル画像を共有することで写真のアップロードと取得を高速に行えるだけでなく、他のユーザにかかる負荷も低く抑えることができる。

また、PSP2P では検索に用いるクエリとして写真の撮影時刻を利用している。従来のファイル名を用いるクエリの方式では、写真ファイルに対して有意義なファイル名が付けられていない。よく用いられるタグによる検索であっても、すべての写真に対してタグが付与されることも期待できない。

これらの理由から、最も検索の基本となる部分には撮影時刻を用いた。

2.3.3.3.2 実験結果

実験では合宿ネットワークに接続している合宿参加者に PSP2P をインストールしてもらい、P2P ネットワークに参加してもらった。また、P2P ネットワーク参加時に接続するためのブートストラップサーバも用意した。本合宿期間中のノード数の推移を図 2.20 に示す。図 2.20 では X 軸が時間、Y 軸がノード数を表す。図のようにそもそもの参加数が少なかったことに加え、夜になると参加者がいなくなっていることが分かる。

また、図 2.21 に共有されている写真の枚数の推移を示す。X 軸が時間、Y 軸が枚数を示す。最初に枚数が飛躍的に増加したのは、保持している写真全てを共有しようとしたユーザが存在したためである。

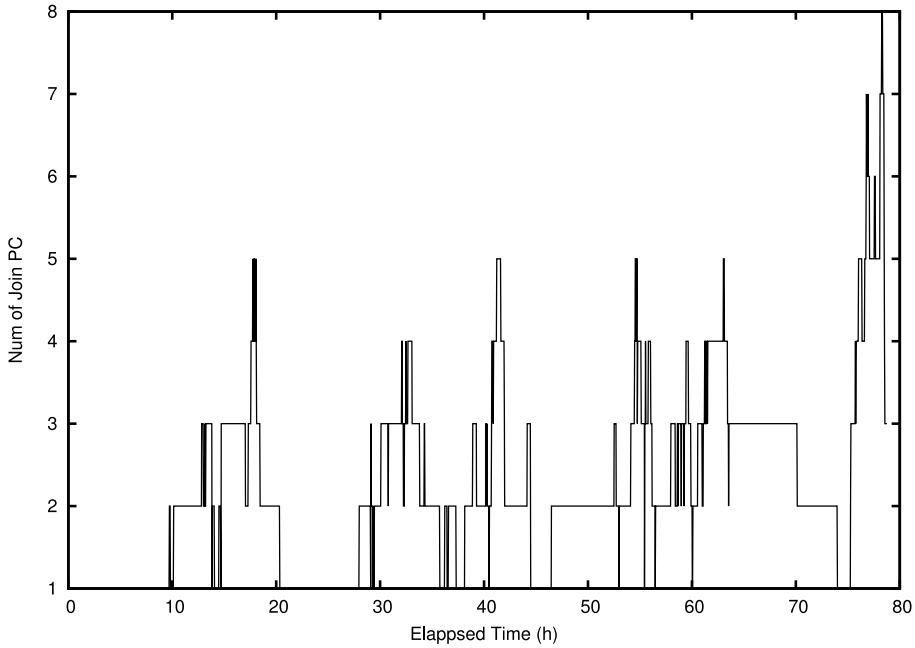


図 2.20. 実験結果：ノード数の変化

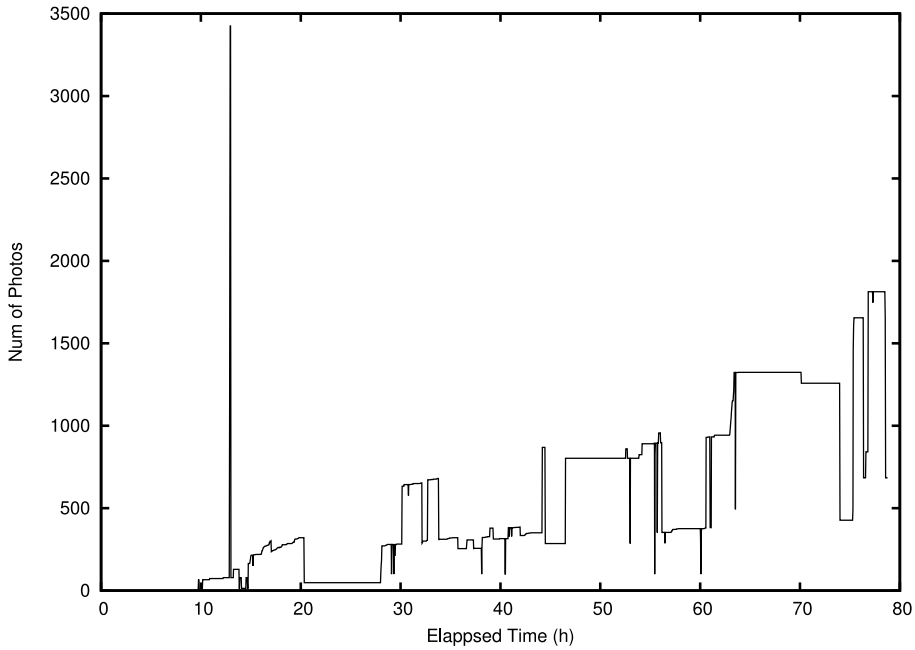


図 2.21. 実験結果：写真の共有枚数

2.3.3.3.3 考察

実験中に様々な問題が発生し、今後の P2P アプリケーション開発のための様々な知見を得ることができた。以下にそれらを列挙する。

P2P 接続不能問題　P2P ネットワークを構成するために、参加者の PC はそれぞれ P2P で接続できる必要がある。しかし、実験中において P2P 接続できない問題が発生した。ここで、接続しようとする PC「A」と接続される PC「B」、更に仲介役の PC「C」を想定する。A は C に対してクエリ等を送信した結果、B という PC に接続する必要が出てくると仮定する。これは例えば元の高画素写真を B が持っている場合などである。A は B に接続しようとするが、何故か接続できず、B が既に離脱したと想定して C に再びクエリする。しかし、C は B と正常に接続できるため、A に対して B と接続するように要請する。これを繰り返し行った結果、P2P ネットワークの安定化部分にまで影響し、P2P への参加や写真の共有に影響が出るようになった。

ここで、A と B が接続出来なかった理由として、ARP による問い合わせが考えられる。参加者は同一セグメントの無線 LAN 環境下にいるため、A と B が P2P 接続をする場合に ARP による問い合わせを行う必要がある。しかし、ARP は IP 層におけるブロードキャストを利用するため、無線 LAN の AP がブロードキャストを抑制した場合、ARP による問い合わせが行われず、結果として A と B が直接接続できないという状況が発生する。

実際にこの事象が発生しているユーザの PC から ARP テーブルを確認したところ、当該 IP アドレスの行を確認することができなかった。そのため、PSP2P に用いている TCP のコネクションはおろか ICMP の ECHO メッセージであっても到達できなかった。尚、デフォルトゲートウェイへの到達性は確保できているため、通常のサーバ・クライアントなサービスを利用する分には問題なかった。これは同一セグメントかつ無線 LAN 環境下で起こる P2P 特有の問題であると考えることが出来る。この問題へ対策するためには異なるセグメントへ複数の PC を配置するなどが考えられるが、根本的に P2P 接続できない PC への対処を考える必要がある。

NAT 問題　P2P ネットワークにとって NAT 下のユーザは大きな問題であるが、本合宿では基本的にユーザがラップトップの PC を利用しているため、NAT 下のノードは存在しないと考えていた。しかし、参加した計算機の幾台かは NAT 下のプライベートアドレスを利用していたため、P2P 接続不能問題と同様に接続できない PC が存在し、問題となった。これは、ユーザが仮想計算機（VM）を PC 上で動作させ、その上で他の OS を動作させるなどし、その OS から P2P ネットワークに参加したために発生した。今後、VM 技術がより一般化すればこうした問題は顕著になっていくと考えられる。

2.3.3.3.4 まとめ

本合宿にて写真の共有を行う P2P アプリケーションの実験を行った。実験の最中に発生した様々な問題のため、少ない PC での写真共有となってしまった。今回の問題は、そもそも P2P 接続できない PC への対処を行っていなかったために発生した。IPv4 アドレスが枯渇していく中で様々な対処法が検討されているが、そのどれもが P2P 接続を阻害する要因になる。例えば、IPv4 アドレスのみを持つ PC と IPv6 アドレスのみを持つ PC が直接通信するためにはトンネリング技術等をユーザの PC に導入する必要がある。また、キャリアグレード NAT が行われれば NAT を越えられない P2P アプリケーションは正常に動作できない。今後のインターネット事情と P2P ネットワーク技術を鑑みながら P2P ネットワークの研究、アプリケーションの開発を P2P ネットワークの研究者は行っていく必要がある。

2.3.3.4 ネットワーク監視による高精度なボット検出手法の実験

2.3.3.4.1 実験概要

本実験では合宿ネットワークのトラフィックを監視し、水谷らが開発したボット検出手法を運用・評価した。これによって水谷らが提案・開発した手法がネットワーク監視によるボット検知において誤検知の発生率が低いことを示した。

近年、コンピュータウィルスに代表される悪意あるプログラム（マルウェア）の活動が深刻化している。特にインターネット上の攻撃者が Command & Control（C&C）サーバを経由して任意のコマンドを多数の感染ホストに実行させるボットネットによる被害が問題となっている。ボットネットを構成するボットは C&C サーバから送信される命令に従い、

分散型サービス妨害攻撃（DDoS）や迷惑メールの送信、個人情報の盗難、他ホストへの感染を実行する。これに対し、各ネットワークのセキュリティ管理者は内部情報の漏洩阻止や対外的な信頼性維持のため、管理ネットワーク内のボットを検知、駆除する必要がある。しかし、ボットは C&C サーバからの指示によってボット自身のソフトウェアを更新する機能を持つため、亜種の発生が頻繁になっており、ボット毎の特徴を示すシグネチャを用いた検知手法では検知できないボットも多い。さらに、ボットは迷惑メール送信や DDoS の請負、あるいは機密情報の売買などによるビジネスモデルが確立している。そのため、ボットの作成者もボットが検知、駆除されるのを防ぐために、特徴的な通信ではなく目立ちにくい通信によって命令の送受信や悪意ある活動を実施する傾向がある。これらの理由により、既存のネットワーク検知手法では十分な精度が期待できない。

本実験では通信の相関関係を利用し、未知のボットの活動を高精度に検知する手法を評価した。侵入検知システム（IDS）は通信のヘッダーおよびペイロードの検査によって、ネットワーク上で発生する様々なセキュリティ侵害に関するイベントを検知している。しかし、通信をパケットや TCP セッション毎に検査しているため、複雑なイベントの検知が困難であった。水谷らは TCP 接続や UDP の送信元ポート、宛先ポートの組合せをセッションとして扱い、セッション間の相関関係を利用することで複雑なイベントを高精度に検知するための手法を提案し、実装している。この手法は特定のソフトウェア（P2P ファイル交換ソフトウェアなど）を高精度に検知するための手法であったが、今回の実験ではボットの検知に着目した。水谷らが収集したボットの検体を実験環境において動作させ、通信データの収集・解析を実施した。その結果、検知に有効であると見られるルールをいくつか作成した。これをボットの通信データに適用した結果、検知の見逃し（False Negative）が発生しにくい事を確認した。このルールを用いて合宿ネットワークを監視し、実運用ネットワークの通信データを用いても誤って検知してしまう False Positive（FP）の発生率を調査した。

2.3.3.4.2 システム構成

本手法はネットワークトラフィックを監視し、リアルタイムでボットの活動を検出するシステムである。システム実装の概要を図 2.22 に示す。本実装は UNIX OS のホスト上で動作するソフトウェアであり、本実験では Debian/GNU Linux 上で動作させた。トラフィック収集には libpcap を利用しており（図中の Traffic Capture 部分）、パケット単位で通信データを収集する。さらに収集したパケットを解析・

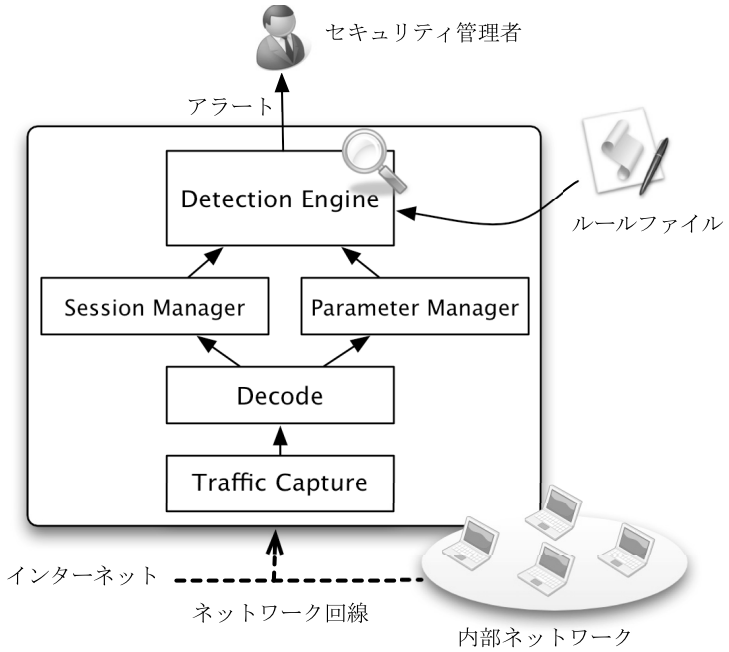


図 2.22. 水谷の実験におけるボット検出手法のシステム構成概要

正規化している（図中の Decode 部分）。ネットワーク上のパケットは通信用のデータ変換やセグメントの分割などにより通信やプロトコルの効率化が実施されている。さらに、URL の UTF8 エンコードなど多様な形式を許容するプロトコルもあるため、検知処理を実行する前の正規化が必須である。その後、通信データ中に出現したセッションを管理する Session Manager やセッション間のメッセージング処理を担うパラメータを管理する Parameter Manager が関連するデータを検索する。これらの情報を元にした上で、読み込まれたルールに従い検知エンジンが動作している。検知した事象はセキュリティ管理者に送付され、インシデント対応に利用される。

今回の実験では、より広範囲のホストに対する通信データの取得を試みたため、対外ルータの直前のトラフィックをミラーリングし通信データを取得した。また、適用したルールは以下の通りである。

1. IRC の通信においてボットで頻繁に利用されるコマンドの文字列が出現しないかを検査する。該当する文字列が出現した場合には、脆弱性を持った Microsoft Windows OS を探索する活動として TCP のポート番号 135、137、445、2967、2968、3127 に対する広域スキャンを発見する。これによってボットの命令と命令に伴う活動を検知する。
2. サービス不能攻撃（DoS）を検知するため、DoS の実行を促すコマンド名を検知し、その直後の大量のデータ転送を検知する。1 つのホストから 1 つのポート番号に対して大量の送信パケットが送信されていた場合、DoS の傾向があるとみて調査する。
3. IRC 中に発生した URL に対し、送信されたホストが HTTP によってファイルのダウンロードを実施したかを検査する。さらにダウンロードされたファイルが Microsoft Windows の実行形式ファイルであった場合、ボットの更新命令が発効されたものとみなし、ボットの活動として検知する。
4. (1) のルールと同様に広域スキャンを検知し、さらに Microsoft Windows の実行形式ファイルの送信を検知する。広域スキャンだけではネットワーク機器によるサービス探索の可能性があるが、スキャンと同時に実行形式ファイルを送信するのは脆弱性を利用した攻略コードが実行されたか、あるいは既にボットに感染しているホストのバックドアを利用した攻撃である可能性がある。
5. ポート 135、137、139、445 に対して攻略コードと見られるデータが送られてきた後に、Microsoft Windows の実行形式ファイルのダウンロードを検知する。多くの攻略コードには x86 系 CPU の NO-OP を意味する 0x90 が連続して含まれる場合が多い。連続する 0x90 を検知するだけでは攻撃発生の根拠として乏しく、攻撃だったとしても攻撃の成否を判断できない。しかし、ボットは感染直後に新しい検体を導入する傾向があるため、発見が可能である。
6. ボット独自のプロトコルを検知する。独自のファイル転送プロトコルや感染ホストを HTTP Proxy として動作させるプロトコルなどが挙げられる。これらのプロトコルは送信データ長やフィールド毎の値を検査することで検知可能であり、連続してパケットを検査することによって高精度に検知ができる。

2.3.3.4.3 実験結果

本実験の結果、合宿ネットワークにおいて誤検知は発生しなかったことを確認した。合宿ネットワークは近年の一般的なネットワークとは異なり、ファイヤウォールが設置されておらず、基本的に通信が制限されていない。そのため、外部からの攻撃は各ホストが個別に防御する必要があり、通常のネットワークよりリスクが高いと言える。監視した際、ルール (5) において内部ネットワークへの攻撃が発生した事実は確認した。しかし、これによってボットを含むマルウェアに感染したと見られる活動は確認されなかった。

この結果から、本手法の誤検知発生率が低い事を確認した。本実験で用いたルールは汎用的なルールとなっており、複数種類のボットを検知できる。しかし、ボットは日々新種が発見されており、水谷らが収集した検体とは全く異なる種類のボットは検知されていない可能性がある。最も著名なオープンソース IDS の 1 つである snort を利用し合宿ネットワークの通信データを追試したところ、マルウェアと疑われる通信が一部発見された。（ただし、これは誤検知が起こりやすいルールであった点と、合宿ネットワークに参加するホストは様々な実験や、一般的に

は利用されないソフトウェアを利用しているなどの点から、マルウェアに感染しているかどうかの判断はできなかったことを補足しておく）そのため、ボットの正確な検知のためには、今後も検体の収集とルールの作成を継続的に実施していく必要がある。

2.3.3.5 映像・音声による遠隔地コミュニケーションの実験

2.3.3.5.1 概要

近年、高速ネットワークの発展によって遠隔地同士をネットワークで結びコミュニケーションを行う遠隔コミュニケーションが盛んに行われている。テレビ電話のように音声対話に映像が加わるだけでなく、遠隔会議、遠隔医療など幅広い分野で利用されている。本実験では、SAMTK（Scalable Adaptive Multicast ToolKit）という、多地点間コミュニケーションを実現するソフトウェアを容易に構築するためのツールキットを用いて開発されたアプリケーションを使用し、プレナリ部屋と子供部屋との通信が円滑に行われるかどうか確かめる。このアプリケーションは、ハイビジョン画質の映像（HDMI キャプチャカードから取り込んだ 1920 × 1080 の HD 画像）と音声による 2 地点間の通信を可能にする。

2.3.3.5.2 システム構成

- プレナリ部屋
  HP 製ワークステーション（xw6600）に SONY 製デジタルビデオカメラ（HDR-HC7）を HDMI ケーブルで接続し、ワークステーションのヘッドホン端子と XLR パッチパネルの入力端子を接続する。また、Polycom 製エコーキャンセラー（Vortex EF2241）に BOSE 製スピーカーを接続する。
- 子供部屋
  DELL 製ワークステーション（T7400）に SONY 製デジタルビデオカメラ（HDR-HC7）を HDMI ケーブルで接続し、ワークステーションの USB 端子に YAMAHA 製スピーカーフォンを接続する。

2.3.3.5.3 結果

プレナリ部屋の様子を子供部屋に中継することに成功した。利用者からは、プレナリ部屋の雰囲気が伝わってきてありがたかったという声がある一方で、技術的に改善点があると感じたとの声もあった。

2.3.3.5.4 反省点

通信のクオリティが悪かったため、合宿中にデバッグしなければならなかった。また、お子さんの年齢、昼間の過ごし方、プログラムのスケジュールとの関係などで、期待していたほど利用が進まなかったため、直接宿泊部屋に中継する仕組みを考えるなど、中継の方法を検討し直す必要がある。

2.4 まとめ

本合宿では、合宿地に実際に集合して顔を合わせる機会を有効に活用できる場の提供を目指し、そのための方策として、mini ワークショップと初参加者による BoF 総括を実施した。合宿終了後のアンケートによれば、これらの活動が一定の成果を挙げた事を確認できたものの、時間配分や総括内容の品質の問題に対する問題点も指摘された。また、近年増えてきた外国人参加者、子供連れ研究者への対応も試みた。

ネットワークに関しては、ホットステージで検証できなかった内容（ADSL セットアップ、MTU 問題、duplex mismatch）で問題が発生していた。今後はホットステージにて必要な検証を十分に完了する計画をたてるとともに、今回の問題点を文書化するなどして次回以後の運用につなげていく。

実験は以下の 5 つが実施された。

- 模倣インターネット環境の実インターネット環境への統合実験
- XCAST + SAM-TK
- P2P ネットワークを用いた写真の共有
- IDS の試験とボット検出用に作成したルールの評価
- 映像・音声による遠隔地コミュニケーションの実験

各実験の詳細に関しては2.3.3項を参照して欲しい。