WIDE プロジェクトにおける IPv6 の実現

南 政樹 (慶應義塾大学)、島 慶一 (株式会社シャープ)、
渡部 謙 (株式会社日立製作所)、山本 和彦 (奈良先端科学技術大学院大学)、
尾上 淳 (株式会社ソニー)、角川 宗近 (奈良先端科学技術大学院大学)、
新 善文 (株式会社日立製作所)、村井 純 (慶應義塾大学)、
宮川 晋 (NTT ソフトウェア研究所)

Abstract : IPv6 はインターネットが抱える問題点を解決する為に開発が進められている、次世代のインターネットプロトコルである。WIDE プロジェクトでは、この IPv6 を、新しいインターネットの研究基盤として位置づけ、研究活動をおこなっている。本稿では、WIDE プロジェクトが進めている実装状況を報告すると共に、その経験に基づいた、IPv6 のソフトウェアアーキテクチャを示す。また、WIDE プロジェクトが今後、どのように IPv6 研究を進めているか述べる。

1 はじめに

インターネットの技術的基盤であるインターネットプロトコルは現在のような巨大なネットワーク環境を想定して設計されたわけではない。このため、アドレス空間の枯渇や経路制御情報の増大のような規模に関する問題が指摘されている。また、インターネットプロトコルが開発された当時では考えられなかった移動ホスト通信、実時間通信、セキュリティとプライバシなどの新しい利用形態への適応が望まれている。

これらの問題はいずれもインターネットプロトコルのプロトコル仕様に起因しているため、インターネットプロトコル自身の変更を行なわない限り解決できない。そこで、新しい次世代インターネットプロトコルを開発することで問題を解決する必要があった。

このような背景の中で、現在のインターネットプロトコルである IPv4(Internet Porotocol Version 4) [1] をベースに再設計した、IPv6(Internet Protocol Version 6) [2] が、いくつかの候補の中から次世代インターネットプロトコルに選ばれた。

WIDE プロジェクトでは、IPv6 を次世代インターネット環境の最も重要な研究課題と位置づけし、IPv6 の開発・実験に取り組んでいる。

本稿では、WIDE プロジェクトがこれまで取り組んだ IPv6 研究の報告と、今後の IPv6 研究活動の展開をまとめる。第 2章では IPv6 の実現に向けた WIDE プロジェクトの指針を示す。第 3章では、WIDE プロジェクトの四つの実装経験から考えられた IPv6 のソフトウェアアーキテクチャを解説する。第 4章では、現在の IPv6 の実装状況と WIDE プロジェクトおよびニューハンプシャー大学相互接続研究所で行なわれた相互接続試験を報告する。第 5章では、IPv6 の実装した上で発見された問題点の整理とその解決方法を提案する。第 6章では、IPv6 研究の今後の展開を示す。特に、IPv6 研究実験ネットワーク 6Bone[3] の構築について述べる。

2 方針

WIDE プロジェクトは、IPv6 を次世代インターネット研究の重要な要素として捉え、IPv6 ワーキンググループを設立し、次世代インターネット研究基盤の構築を目標に研究を進めている。

現在、IPv6 ワーキンググループ内では三つの独立したオペレーティングシステムへの実装が進められている。これらの実装は、以下に示す三段階の計画に従って進められている。

2.1 ホストに注目した開発 (第一段階)

ホストに注目した開発とは、計算機をホストとして IPv6 ネットワークに接続するために必要な機能を実現することである。

例えば、NDP(Neighbor Discovery Protocol)[4] の実装や拡張ヘッダ解析の実装、また、これまでに詳細が決定されていない部分に対し、今後容易に新たなモジュールを組み込める仕組みの提供などである。

また、OS レベルだけではなく、ライブラリ群やコマンド群などアプリケーションレベルの IPv6 環境開発基盤を整備する。

2.2 ネットワークに注目した開発 (第二段階)

ネットワークに注目した開発とは、ネットワークとネットワークを結ぶために必要な機能を実現することである。具体的には、IPv6 パケットの経路制御、IPv4 による IPv6 パケットのカプセル化 (トンネリング)、IPv6 対応の PPP(Point to Point Protocol)[5] によるシリアルライン上の通信などである。

また、ここで開発された技術を利用し、IPv6 の実験ネットワーク “6Bone” を構築し運用する。ここで得られた経験や運用技術は実装にフィードバックされる。6Bone は、まず WIDE プロジェクト IPv6 ワーキンググループ内で構築し、徐々にアジア・太平洋地域、アメリカ、ヨーロッパなどの研究機関と接続し規模を拡大させる。

2.3 快適なユーザ環境に注目した開発 (第三段階)

ユーザが容易にインターネットを利用できる環境を、快適なユーザ環境という。IPv6 は、以下に示した快適なユーザ環境を提供している。ここではこれらの実現を目指す。

- プラグ & プレイ
  ホストのアドレス取得、ルータのアドレス提供、DNS(Domain Name Service) の更新 [11] などの機能を実現する。
- セキュリティ
  ペイロードの暗号化、鍵交換、認証などの機能を実現する。暗号など社会的に規制のある機能は、モジュール化を行なって必要に応じて利用できる枠組を提供する。
- リアルタイム
  ヘッダ中の Priority フィールドと Flow ID フィールドを利用して、実時間性を有するパケットを優先的に処理する機構を実装する。

3 ソフトウェアアーキテクチャ

IPv6 ワーキンググループでは、独立した三組織の実装から得られた経験をもとに比較、検討し、IPv6 のソフトウェアアーキテクチャについて考察してきた。以下ではそれらについて述べる。

3.1 プロトコル処理の分割

プロトコル処理とは、ヘッダ部の解析・生成、上位層または下位層へのデータの受渡しのようにプロトコルとして規定されている処理を意味する。

IPv6 は、IPv4 の問題点を解決しそれに置き換わる次世代のインターネットプロトコルとして設計されている。しかし、IPv4 と IPv6 のヘッダフォーマットを比較 (図 3.1、図 1) すると、バージョンを示す先頭の 4bit 以外共通点はない。これは全く異なるプロトコル処理が必要なことを示す。ヘッダフォーマットだけではなく、オプション機能にも注目すると、拡張ヘッダによる IPv6 のオプション機能と、可変長による IPv4 のオプション機能では、その処理方法が大きく異なる。

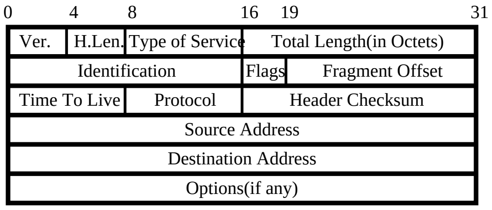


図 1: IPv4 ヘッダフォーマット

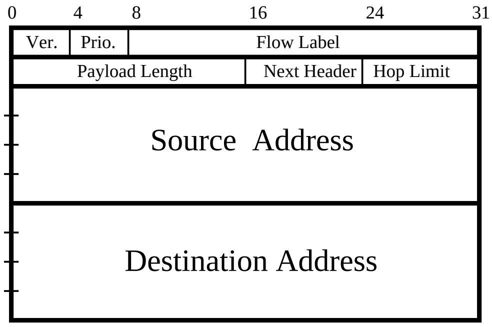


図 2: IPv6 ヘッダフォーマット

そこで、WIDE プロジェクトでは IPv4 のプロトコルスタックは変更を加えず、IPv6 のプロトコルスタックを独立して組み込む方法をとった。

3.2 デュアルスタック

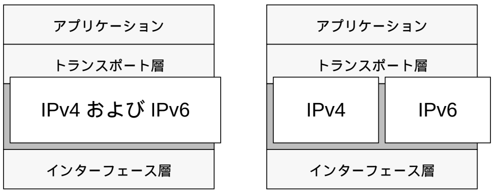
同一ホストに二つの異なるプロトコルスタックが存在することを、デュアルスタックと言う。デュアルスタックを持つホストは IPv4 から IPv6 への移行期間中に必要不可欠である。なぜなら、IPv4 から IPv6 への移行はインターネット上で利用される通信プロトコルに応じて、次のような三つの期間に分けて行なうのが自然だからである。

- IPv4 のみで通信する期間 (現在)
- IPv4 と IPv6 の混在期間 (移行期)
- IPv6 のみで通信する期間 (移行期間後)

現在のように大規模なインターネット環境では、同時に全てのホストを IPv4 から IPv6 へ置き換え

ることは難しい。そこで、移行期にはIPv4とIPv6の両方で通信できるホストを徐々に増やし、インターネット全体で緩やかに移行することが重要だと考える。

前節でも述べた通り、IPv4とIPv6が別の処理系で処理されることを生かし、どちらか一方に限定するのではなく、図3のようにデュアルスタックとして実装することで、同一ホストがIPv4とIPv6の両方で通信を可能にする。



1. IPv4 の拡張　　2. IPv6 の追加

図3: IPv4とIPv6のデュアルスタック

3.3 プロトコルキュー

プロトコルキューとは、データリンク層からネットワーク層に渡されるパケットを一時的に保持するための待ち行列である。

IPv4とIPv6をデュアルスタックで実装する場合、プロトコルキューを共用するかそれぞれ別に用意するかで設計は大きく異なる。確かに、IPヘッダの中のバージョン部分によってIPv4かIPv6かを判断できるので、プロトコルキューを共用できる。実際に、この方法で実装している例もある。

WIDEプロジェクトではIPv4とIPv6は全く異なるプロトコルという解釈と以下のような理由から、プロトコルキューをIPv4とIPv6のそれぞれに用意する方針をとった。

- 理解しやすさの追及
  理解しやすさの追及とは、WIDEプロジェクトのIPv6のソースコードを後から読んだ者が、混乱することなく理解できることへの配慮である。
- データリンク層のフレームタイプ
  データリンクメディアには、多くの場合、データリンクフレームのヘッダにフレームタイプと呼ばれるネットワーク層がどのプロトコルであるかを示す値を含む場合が多い。例えば、Ethernetの場合IPv4は0x0800であり、IPv6は0x86ddである。これによって、データリンク層からネットワーク層へパケットの内容を渡す時に違うプロトコルに分配できる。

3.4 拡張ヘッダの解析

拡張ヘッダはIPv6ヘッダと上位層プロトコルヘッダの間にはさまれる形で、IPv6パケットに含まれる(図4参照)。

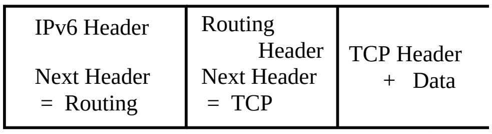


図4: IPv6ヘッダ、拡張ヘッダ、上位層ヘッダの関係

あるヘッダAの直後に来るヘッダBがどのオプションヘッダなのかを知る為には、ヘッダAに書かれている“Next Header” フィールドを参照する。この処理はIPv4で上位層プロトコルを識別する為に用いられているプロトコルスイッチに類似している。そこで、IPv6でオプションヘッダを解析するためにこのプロトコルスイッチを拡張し、それぞれから適当なオプションヘッダ解析関数群にヘッダ部を渡す方法で実装した。

3.5 ICMPv6/NDP

ICMP(Internet Control Message Protocol)[6]は、エラー通知の機構である。

IPv6で用いられるICMPv6(Internet Control Message Protocol for IPv6)[7]は、エラーの報告だけではなく、NDPメッセージの送受信やマルチキャストグループのメンバ管理[8]もできるように拡張されている。

NDPは、ネットワーク層のアドレスとデータリンク層のアドレスを関連付けるプロトコルである。ここでネットワーク層のアドレスとデータリンク層のアドレスを関連付けることをアドレス解決と呼び、その仕組みをアドレス解決機構という。IPv4では、アドレス解決機構をARP(Address Resolution Protocol)[9]というデータリンク層のプロトコルで実現していた。

ARPとNDPの相違点は、ARPはデータリンク層のプロトコルとして設計されているのに対し、NDPはICMPv6を利用したネットワーク層のプロトコルとして設計されている点である。

ネットワーク層にアドレス解決機構を組み込むことで、これまでデータリンクメディア毎に必要だったアドレス解決機構のソフトウェアは統合される。

4 実装状況と検証

WIDEプロジェクトIPv6ワーキンググループには四つの実装がある。

四つの実装のうち三つは、慶應義塾大学、奈良先端科学技術大学院大学、株式会社日立製作所のそ

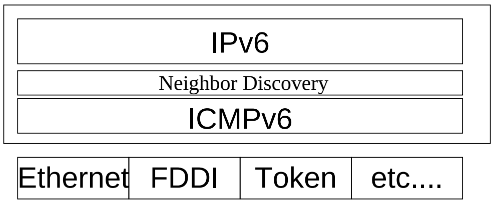


図5: NDP、ICMPv6、データリンクメディアの関係

れぞれグループが進めている、オペレーティングシステムへの実装である。また、残りの一つはソニー株式会社の尾上による、オペレーティングシステムへではなくIPv6デーモンの実装である。これは、BSD UNIXのBPF(Berkley Packet Filter)[10]を利用し実装されている。

いずれも開発の対象のオペレーティングシステムは“4.4BSD-Lite”をベースにしているBSDI社のBSD/OSである。このオペレーティングシステムは市販されているもので安定しており、ソースコードが入手可能だからである。

現在のところ、オペレーティングシステムへの実装は三組織が独立して行なっている。この三つには、実装をはじめた時期や既にある実装をベースにしたかどうかなどの相違点がある。表1にその一覧を示す。

表1: WIDEプロジェクトIPv6ワーキンググループOSへの実装一覧

実装組織	実装開始時期	ベース
慶應大	1995年6月	なし
奈良先端大	1995年9月	なし
日立	1995年9月	INRIA

これらの実装は、第3章で提案したソフトウェアアーキテクチャには準拠しているものの、それぞれが独立している。

一般的に、新たなプロトコルを実装する場合、プロトコル仕様に準拠し実装するだけではなく、実際に他の実装との相互接続できることの検証する必要がある。

そこで、WIDEプロジェクト内で二回のワークショップを行ない相互接続性を検証した。また、ニューハンプシャー大学相互接続研究所で行なわれた、相互接続試験にも参加した。ここではWIDEプロジェクト以外の参加組織の実装と試験を行なった。

4.1 第一回IPv6ワーキンググループワークショップ

1995年12月14日~15日の2日間、慶應義塾大学で開催し相互通信実験を行なった。

4.1.1 実験概要

この時点では、それぞれの組織の実装状況がまちまちであったことと、これまでに各実装の間での相互通信の実績が全く無かったため、各実装間でIPv6パケットが正しく交換されることの確認を目的とした。二つのノード間でIPv6パケットを交換するには次の三項目が満たされる必要がある。

(1) ネットワークへ送信されるパケットのフォーマットが正常であること
(2) ネットワーク上の正常パケットを正しく受信すること
(3) NDPによるアドレス解決機構が正常であること

4.1.2 実験結果

チェックにはpingコマンドを利用し、それぞれの実装間で正しく動作するか実験した。

図6にpingによる実験の結果を示す。図中の矢印はある実装から矢の先が示す実装へのpingが成功したことを示している。これからも分かるように全ての実装の間でpingの送受信ができることを確認した。

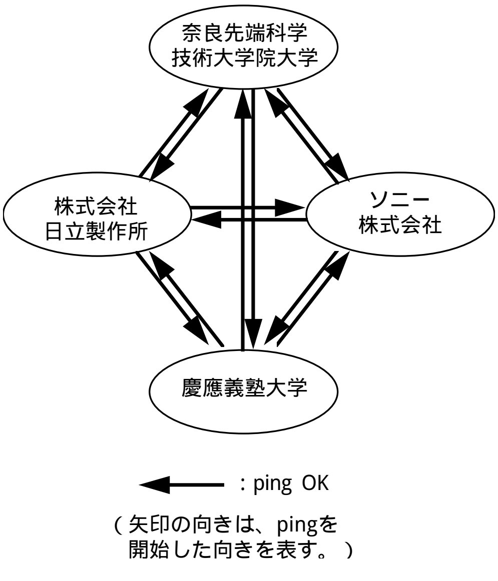


図6: pingの実験結果

4.2 第二回 IPv6 ワーキンググループワークショップ

1996 年 1 月 29 日～30 日の 2 日間、慶応義塾大学において第二回 IPv6 ワーキンググループワークショップを開催し、相互接続実験を行なった。

4.2.1 実験概要

第一回 IPv6 ワーキンググループワークショップでは、同一ネットワーク上に存在するホスト同士の間で ping コマンドによる通信テストを行なった。これにより、NDP を含めた IPv6 基本機能による、ネットワーク層の通信が可能なことを確認した。そこで今回は、異なるネットワーク上に存在するホスト間で ping コマンドによるルータを介した通信テストと、IPv6 上で動作する TCP および UDP の通信テストを行ない、次の三点を確認することを目的とした。

- IPv6 パケットの転送機能によるネットワーク層の通信が可能であること
- 実装した IPv6 がトランスポート層に対してネットワーク層のサービスを問題なく提供できること
- トランスポート層から IPv6 を見た場合にネットワーク層のプロトコルとして問題がないこと

次の方法で各通信を検証した。

ping によるルータを介した通信テスト あるホスト A から異なる LAN 上に存在する別のホスト B に対して ping コマンドを実行する。経路上のルータを介して正しく ICMPv6 メッセージが交換されることを確認する。

telnet による TCP の通信テスト telnet と telnetd を IPv6 を用いるように変更、再構築する。そして、ホスト A から同一 LAN 上に存在するホスト B に対して telnet コマンドを実行する。これがホスト B の telnetd によって受け付けられ、ホスト A からホスト B に対して正常にログイン、ログオフできることを確認する。

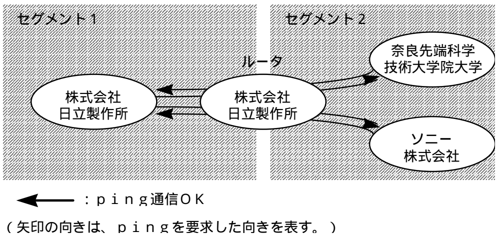
ポート 7 のエコーサービスによる UDP の確認 inetd のソースコードを、IPv6 を用いるように変更、再構築する。そして、ホスト 1 から同一 LAN 上に存在するホスト B の UDP ポート 7 に対して送信された UDP データグラムが、ホスト B の inetd によって受け付けられ、そのままホスト A に送り返されることを確認する。UDP ポート 7 は、送られてきたデータをその送信ホストにそのまま送り返す、エコーサービスを提供するように作られている。

4.2.2 実験結果

実験をもとに、以下のような結果が得られた。

- IPv6 パケットのフォワーディング処理を実装していた日立の実装をルータとして用いテストした結果
  異なる LAN 上に存在する奈良先端大、ソニーの尾上、日立の実装の間で ping コマンドによるルータを介した通信ができることを確認した。
- TCP の通信テスト結果
  奈良先端大 と日立の実装の間では、telnet/telnetd 共に問題なく動作し、TCP の通信が確認できた。ソニーの尾上の実装は、TCP の緊急データに対する処理が未実装であったため、緊急データを用いる通信ができないことを除くと、TCP の通信ができることを確認した。
- UDP の通信テスト結果
  奈良先端大、ソニーの尾上、日立の全ての実装の間で UDP の通信ができることを確認した。

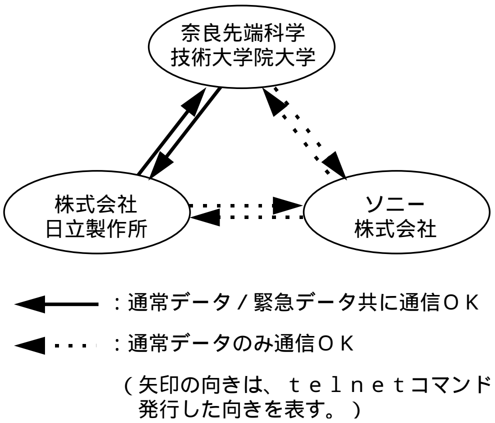
図 7～図 9 に実験結果を図示する。



：ｐｉｎｇ通信ＯＫ

（矢印の向きは、ｐｉｎｇを要求した向きを表す。）

図 7: IPv6 パケットの転送実験の結果



：通常データ／緊急データ共に通信ＯＫ

：通常データのみ通信ＯＫ

（矢印の向きは、ｔｅｌｎｅｔコマンド
発行した向きを表す。）

図 8: TCP の実験結果

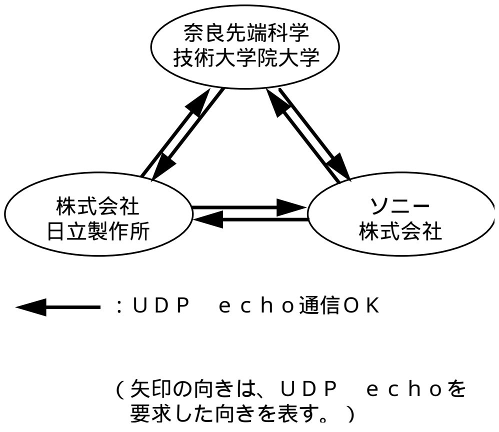


図 9: UDP の実験結果

4.3 第一回ニューハンプシャー大学相互接続研究所接続実験

1996 年 2 月 5 日 ~ 9 日の 5 日間、ニューハンプシャー大学 の相互接続研究所 (InterOperability Laboratry:IOL) において世界各国の主要な IPv6 開発者が参加して第一回相互通信実験が行なわれた。IPv6 ワーキンググループからも三つのオペレーティングシステムへの実装の代表者が実験に参加した。

参加組織は次の通りである。

- WIDE プロジェクト
  - 慶応義塾大学
  - 奈良先端科学技術大学大学
  - 株式会社日立製作所
- BayNetworks、Inc.
- Digital Equipment Corp.
- FTPsoftware、Inc.
- INRIA(Institut National de Recherche en Informatique et en Automatique)
- Mentat、Inc.
- ProcessSoftware Corp.
- SICS(Swedish Institute of Computer Science)
- Sun Microsystems、Inc.
- Xerox / Palo Alto Reasearch Center

4.3.1 実験概要

通信実験は仕様に従って独立に実装された IPv6 が互いに正しく通信できることを確認すると同時に、相互接続研究所が用意した NDP の動作確認プログラムにより、標準仕様に従って実装されているか確認することを目的とした。実験項目は次の通りである。

- NDP の動作確認
  - アドレス解決 (Neighbor Solicitation / Advertisement)
  - Router and Prefix Discovery (Router Solicitation / Advertisement)
  - Redirect

ただし、WIDE プロジェクトの実装はいずれも Router and Prefix Discovery および Redirect を実装していなかったので、アドレス解決の動作のみを確認した。

- ネットワーク層より上位層の動作確認
  - ping(ICMP メッセージの送受信)
  - telnet(TCP の動作確認)

4.3.2 実験結果

確認した内容とそれぞれの実装に対する結果を以下に示す。

- NDP の動作確認

  慶應義塾大学の実装 Host Unreachable Detection と NS (Neighbor Solicitation) メッセージのタイミングを間違えていることを指摘された。その他は正しく実装されていることがわかった。

  奈良先端科学技術大学院大学の実装 上位レイヤから NDP キャッシュにアクセスする場合の状態遷移を除いて、全ての状態遷移が正しく行なわれていることを確認した。

  株式会社日立製作所の実装 本通信実験に参加した時点では、日立製作所の実装はフリーに公開されている NDP のソースプログラムをほぼそのまま用いていた。実験の結果、この実装はアドレス解決は正常にできるものの、次のような数々の不具合が発見された。
  - 状態遷移のタイミングが早すぎる箇所が複数ある。
  - NS (Neighbor Solicitation) メッセージに対する応答として返す NA (Neighbor Advertisement) メッセージの数が多すぎる。
  - probe 状態でのリトライ間隔が短すぎる。

- ping コマンドの動作確認
  WIDE プロジェクトのいずれの実装においても ping コマンドは正しく動作した。また、フラグメントしたパケットを使った ping でも正しく処理できることも確認した。

- telnet コマンドの動作確認
  WIDE プロジェクトのいずれの実装においても telnet コマンドは正しく動作し、TCP が正常に動作することを確認した。

5　問題点とその解決策

WIDE プロジェクト IPv6 ワーキンググループでは、実装によって明らかになった IPv6 の仕様上の問題を指摘する。

問題となったのは、以下の二点である。

- Path MTU Discovery(ICMPv6)
- データリンク層のアドレス解決のための仕様の不備 (NDP)

5.1　Path MTU Discorvery(ICMPv6)

MTU(Maximum Transmission Unit) とは、データリンク層の通信媒体が一度に送信できるフレームの最大長である。Path MTU Discovery とは、通信している二つのホスト間の最小 MTU を求める技術であり、IPv6 では必ず実装しなければならない。

実際には、通信を行なう度に Path MTU Discovery を用いて最小 MTU を調べるのは効率的ではないので、あるホストに対して Path MTU Discovery で求めた MTU はキャッシュしておく必要がある。4.4BSD-Lite では調べた MTU を経路情報と一緒に経路制御表に持っている。しかし、この方法ではデフォルトの経路が経路表に登録されている場合、経路表に含まれる MTU の値が正しくない可能性がある。そこで、WIDE プロジェクト IPv6 ワーキンググループでは経路制御表とは別に通信先のホスト毎に最小 MTU の表を持つ方法を提案した。

5.2　データリンク層のアドレス解決

NDP は IPv6 はある IPv6 アドレスに対応するデータリンク層のアドレスを見付けるために、Neighbor Solicitation メッセージと呼ばれる ICMPv6 パケットをマルチキャストする。Neighbor Solicitation メッセージを受け取ったホストは、送信者に対して自分のデータリンク層のアドレスをユニキャストで返す。このメッセージを Neighbor Advertisement という。ところが、Neighbor Solicitation のパケットフォーマットには、送信者のデータリンク層のアドレスがオプションでしか規定されておらず、必ずしも含まなくてもプロトコルには反していないことになる。

問題はここで発生する。もし受け取った Neighbor Solicitation メッセージに、データリンク層のアドレスが含まれていないと、受け取った側は同じようにアドレス解決に Neighbor Solicitation メッセージを送ることになる。この Neighbor Solicitation メッセージに、もし、またデータリンク層のアドレスが含まれていないと、この二者間では永久にアドレス解決が終わらなくなる。

この問題を NDP の仕様設計者である、Thomas Narten、Erik Nordmark 両氏に報告したところ、仕様の間違いであることが判明した。現在の仕様書では、Neighbor Solicitation メッセージには、データリンク層のアドレスを含めなければならない、と修正されている。

なお、WIDE プロジェクト IPv6 ワーキンググループの実装では、Neighbot Solicitation メッセージを送信する際に、必ず送信者のデータリンク層のアドレスを含めている。

6　今後の展開

WIDE プロジェクトにおける IPv6 研究は、現在、先述の第一段階を終えようとしている。今後はルータの実装を中心に第二段階へ進展する予定である。

6.1　IPv6 ルータ

ルータはネットワークを構成する要所に位置するため、ネットワーク全体での情報を反映させるには、非常に都合が良い。例えば、DNS のエントリを動的に更新したり、移動ホストの検知を行なったりなどはルータを利用する方が効率がよいかも知れない。また、これまでは無かった新しいサービスを組み込むにも、ルータが鍵となる。WIDE プロジェクトでは、新たなネットワーク環境を構築する際に、ルータの役割について議論し、新しいサービスの提供などを考えたい。

ルータのもう一つ大切な役割は、IPv4 から IPv6 への移行期間に、IPv4 と IPv6 のパケットの交換を行なうことである。このため、デュアルスタックを実装したルータが必要となる。また、IPv4 ·IPv6 の相互のカプセル化を実現しなければならない。

6.2　6Bone の構築

6Bone の構築は 2 段階に渡って行なわれる。まず、現在のインターネットとは独立の IPv6 のみが運用されるネットワークを構築する。次に、Mbone のように、トンネリング技術を用いて、IPv4 上に IPv6 の仮想ネットワークを整備する。

現在計画中の 6Bone では、WIDE インターネットの NOC(Network Operation Center) と参加組織のうち、東京 3 か所、藤沢、奈良、大阪を拠点に 6Bone の構築を行う予定である。

6.3　移行メカニズム

IPv4 から IPv6 への移行は，現在も議論が進められている．基本的にはデュアルスタックのホストを利用し，徐々に IPv4 のネットワークから IPv6 のネットワークへ移行するという方針で進められている (NG-Trans[12])．

IPv4 から IPv6 へ移行する際の問題点を完全に予想することは難しい。しかし、実際にネットワークを構築して、運用実験を行なえば、問題点のいくつかは明らかになるであろう。今後、WIDE プロジェクトでは、アジア ·パシフィック地域における

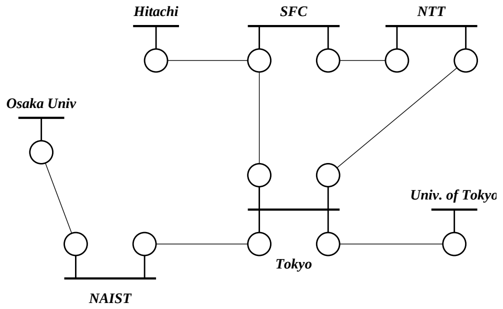


図 10: WIDEプロジェクト6boneのトポロジ

6Boneを構築し、IPv4からIPv6への移行をシミュレーションし問題点を予測していく。

7 まとめ

WIDEプロジェクトは、IPv6を次世代インターネット研究の最も重要な基礎研究と位置づけ、IPv6ワーキンググループ設立した。IPv6ワーキンググループは、実装を通じてIPv6のプロトコル仕様に問題があることや、IPv6のソフトウェアアーキテクチャについて考察し、それを実装に反映させてきた。今後は、引続きIPv6ワーキンググループを中心にIPv6によるインターネットの実現に向け、実験ネットワークである6Boneの構築を行なっていく。

参考文献

[1] J. Postel, “Internet Protocol”, RFC791, 09/01/1981.

[2] S. Deering, R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, RFC1883, 01/04/1996.

[3] Bob Fink, Jim Bound, “6bone Home Page”, http://www-6bone.lbl.gov/6bone.

[4] Thomas Narten, Erik Nordmark, W A Simpson, “Neighbor Discovery for IP Version 6 (IPv6)”, Internet-Draft: draft-ietf-ipngwg-discovery-06, 03/14/1996

[5] W. Simpson, “The Point-to-Point Protocol (PPP)”, RFC1661, STD 51, 07/21/1994.

[6] J. Postel, “Internet Control Message Protocol ”, RFC792, 09/01/1981.

[7] A. Conta, S. Deering, “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)”, RFC1885, 01/04/1996.

[8] S. Deering, “Host extetions for IP Multicasting” RFC1112, 08/01/1989.

[9] D. Plummer, “Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware”, RFC826, 11/01/1982.

[10] Gary R. Wright, W. Richard Stevens, “TCP/IP Illustrated, Volume 2, The Implementation” Addison-Wesley Publishing Company, ISBN 0-201-63354-X.

[11] S. Thomson, C. Huitema, “DNS Extensions to support IP version 6”, RFC1886, 01/04/1996.

[12] R. Gilligan, E. Nordmark, “Transition Mechanisms for IPv6 Hosts and Routers”, RFC1933, 04/08/1996.