

## 第 6 部

# IP Version 6



# 第 1 章

## V6 分科会

この章では、IPv6 に取り組んでいる V6 分科会 の活動と成果を報告する。

### 1.1 はじめに

1992 年神戸で開催された ISOC の最初の国際会議 INET'92 において、クラス B アドレスが数年内に枯渇することが報告された。それまでインターネットでは、経路制御表の大きさを抑えるために、小規模なネットワークにもクラス B を割り当てて来た。よって、インターネットの成長に伴って、使用効率の低いクラス B の枯渇という問題に直面したのである。

クラス B の割り当て速度を落とすために、ネットワークの大きさに適した複数のクラス C の割当てが始まった。しかし、一つのネットワークに対して複数の情報を持たなければならないため、今までのクラスに基づいた経路制御表は急激に増加する。もし、2 の冪乗のクラス C のブロックをネットワークとして連続した地域に割り当てているなら、可変長のマスクを用いて経路情報を集約できる。そこで、インターネットは、この割り当てポリシーを世界規模で採用した。また、アドレスとマスクの組みによる経路制御技術は、CIDR(Classless Inter-Domain Routing)[14] という名称で知られ、現在では多くのルータメーカーによって実装されている。

アドレスの使用効率を上げ、かつ、経路制御表の成長を抑える枠組が用意されたとはいえ、IPv4 のアドレス空間は  $2^{32}$  しかない。インターネットがこのまま成長を続ければ、いつかは IP アドレス全体が枯渇することは明らかである。実際に、2010 年ごろには、割り当てられる IPv4 のアドレスはなくなると予想されている。そこで、より大きなアドレス空間を持つ次世代 IP を開発し、少なくともアドレスの枯渇を根本的に解決しなければならない。

次世代 IP の選択の歴史は、さまざまな提案と妥協、そして、合意の繰り返しであった。当初は、OSI の CLNP を用いた TUBA(TCP and UDP over Bigger Addresses)[15] や、TCP の改良にまで取り組んでいた TP/IX[16] などが、次世代 IP の候補として挙がっていた。また、IP に IP をカプセル化するという第三の提案があったが、IPAE(IP Address Encapsulation) と名を変え、SIP(Simple IP) への移行戦略として採用された。

SIP とは、20 年間の IPv4 の運用の中で経験的に使われなかったヘッダを削り、ヘッダを簡略化することに重点を置いた提案である。ポリシーを考慮した経路制御の枠組を提供する PIP[17] とマージされた後、さらに、アドレスを 64 ビットから 128 ビットに拡張して SIPP16[18] となった。

1994 年 7 月の IETF では、TUBA、TP/IX、SIPP16 などの候補のなかから、次世代 IP の初期デザインを選ぶために、選択基準が発表された。この基準に従って、同年 10 月の IETF において、次世代 IP のとして SIPP16 の採用が決定された。SIPP16 を基に改良が加えられた仕様は、バージョン番号を 6 としたため、現在 IPv6 の名で知られている [19]。

他のプロトコルに携わっていた研究者も、IPv6 の研究開発に参加し、アドレスの拡張以外にもさまざまな機能を取り込む努力が行なわれている。IPv6 の概要については、1.2 節で述べる。現在では、基本的な仕様が RFC となっていると同時に、世界的にさまざまな組織が IPv6 の開発を進めており、相互接続性を検証できる段階にきている。

WIDE Project では、IPv6 をインターネットの重要な基礎研究であると位置づけ、V6 分科会 を設立して研究に取り組んでいる。ホストやルータの実装、研究ネットワークの構築、人により優しいネットワーク環境の整備などが研究課題である。

WIDE Project は、以下のように研究の段階をおおまかに 3 つに分け、研究・開発を進めている。また、これに並行して、IPv4 から IPv6 への移行の枠組を作ることに取り組んでいる。

1. ホストとしての機能の実装。
2. ルータとしての機能の実装。実験ネットワークの構築も含まれる。
3. より人に優しい環境の整備。

現在は第一段階にあり、BSD/OS を基にした独立した実装が 3 つ存在する。3 つの実装は、慶應義塾大学の南、奈良先端科学技術大学院大学の島、株式会社日立製作所の渡部らが中心になって開発している。慶應義塾大学の南と奈良先端科学技術大学院大学の島は、スクラッチから実装している。株式会社日立製作所の渡部らは、当初 INRIA が NetBSD 用に開発したコードを、BSD/OS に移植したものであったが、現在では大幅に書換えられている。また、ソニー株式会社の尾上は、BSD/OS のユーザプロセスとして IPv6 のエミュレータを作成している。

我々は、これらの実装に対し、第一回 V6 分科会 ワークショップ、第二回 V6 分科会 ワークショップ、および、ニューハンプシャー大学の InterOperability Lab. で実施された第一回 IOL 相互通信実験においてテストを行なっている。これらの成果については、1.4 節で詳しく説明する。

1.3 節は、これまでの実装経験を通じて得られた IPv6 のためのソフトウェア・アーキテクチャについて議論する。また、第二段階である実験ネットワーク構築の準備を始めているので、この計画について 1.5 節で触れる。

## 1.2 IPv6 の概要

IPv6 は従来のインターネットプロトコルである IPv4 を基に新しくデザインされたプロトコルである。IPv4 から IPv6 への大まかな変更部分を以下に挙げる。

**アドレス空間の拡張** アドレス空間が従来の 32 ビットから 128 ビットに拡張された。また、新しいアドレスの表記法が定義された。前者については 1.2.1 節、後者については 1.2.3 節で詳細に述べる。

**ヘッダフォーマットの簡略化** ヘッダフィールドのうち幾つかの部分が削除されるか、拡張ヘッダとなった。その結果、ヘッダ長が固定となり、ヘッダ解析による計算機およびルータの負荷が軽減した。詳細は 1.2.4 節で述べる。

**フローラベルによるパケット転送** IPv6 ではフローラベルという 24 ビットの識別子を設けている。これは、リアルタイム通信などの回線品質を保証するサービスに利用される。

**認証と秘匿性の枠組** インターネットの利用環境が研究から商用へ比重が移るに従って、通信相手の認証やその内容を保護することが重要になってきている。IPv6 では拡張ヘッダの形式で認証や暗号化の枠組を提供する [20, 21, 22]。

以下では、IPv4 にはない IPv6 の新しい概念、用語等を中心に解説する。

### 1.2.1 IP アドレス

IPv4 から IPv6 への移行の主たる目的はアドレス幅の拡張である。インターネットに接続されている各計算機は IP アドレスという整数値によって一意に識別される。IPv4 ではこの幅が 32 ビット、すなわち 40 億程度しかなく、近い将来には全てのアドレスを使いきってしまうと考えられる。そこで、IPv6 においてはアドレス空間が 128 ビット ( $2^{96} \approx 8 \times 10^{28}$  倍) に拡張された。

### 1.2.2 アドレスアーキテクチャ

アドレス体系については、IPv4 のクラスと同様に上位数ビットで、アドレスのタイプを決定できるようになっている [23]。表 1.1 にアドレスタイプを示し、それぞれの割り当てアドレスの意味を以下で述べる。

**Provider-Based Unicast Address** 経路情報を集約するために、プロバイダごとに割り当てていくためのアドレスである。

**Geographic-Based Unicast Address** 市町村単位でネットワークを組めるように、地域単位で割り当てられるように予約されている。Provider-Based Unicast Address と同じく、経路が集約できるように割り当てて行く。

**Reserved for NSAP Allocation** NSAP アドレスとのマッピングが行えるように、予約されているアドレスである。

**Reserved for IPX Allocation** IPX アドレスとのマッピングが行えるように、予約されているアドレスである。

**Link Local Use Address** 同一リンクでのみ使用されるアドレス。このアドレスでは、ルータを超えて通信することは出来ない。

**Site Local Use Address** 同一サイトでのみ使用されるアドレス。

また、IPv6 アドレスの機能は以下の 3 種類に大別される。

**ユニキャストアドレス** CIDR を利用している IPv4 のユニキャストアドレスと構造は同じである。

**マルチキャストアドレス** IPv4 のそれを拡張した形で提供される。ブロードキャストアドレスは特殊なマルチキャストアドレスと定義される。

**エニーキャストアドレス** 複数の計算機が同一のアドレスを持つとき、それをエニーキャストアドレスという。マルチキャストとは異なり、そのアドレスに対して送信されたパケットは、同じアドレスを持つ計算機のうち経路的に最も近い計算機に配送される。例えば、複数の DNS サーバに対して同一のアドレスを設定することによって、最も近いサーバを指定できる。アドレス体系はユニキャストアドレスをそのまま使用する。

### 1.2.3 アドレス表記法

IPv4 では、オクテットごとにピリオドで区切ったアドレス表記法が使われていた。しかし、IPv6 ではアドレス長が 4 倍になったため、この表記法では長くなりすぎる。そこで、IPv6 のアドレスは 4 桁の 16 進数ごとに “:” で区切ることになった [23]。

```
0800:1234:0000:0000:0000:0000:00AB:CDEF
```

連続した 0000 は “::” で省略することができる。

```
0800:1234::00AB:CDEF
```

また、区切り単位ごとに上位桁の 0 は省略できる。

```
800:1234::AB:CDEF
```

表 1.1: アドレスタイプ

先頭ビット	割り当て
0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP Allocation
0000 010	Reserved for IPX Allocation
0000 001	Unassigned
0000 1	Unassigned
0001	Unassigned
001	Unassigned
010	Provider-Based Unicast Address
011	Unassigned
110	Geographic-Based Unicast Address
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link Local Use Address
1111 1110 11	Site Local Use Address
1111 1111	Multicast Address

ただし、複数箇所を “:” にすることはできない。これはアドレスが曖昧になり、一通りに定まらないからである。例えば、以下のようなアドレスを考える。

```
0123:4567:89ab::cdef::0123
```

これは以下のどちらにも解釈できるので、曖昧なアドレスである。

```
0123:4567:89ab:0000:0000:cdef:0000:0123
```

```
0123:4567:89ab:0000:cdef:0000:0000:0123
```

## 1.2.4 パケットフォーマット

IPv4 の 20 年間の運用経験に基づいて、有効に機能しなかった部分を削ったことで、IPv6 ではヘッダフィールドを大幅に簡略化した。また、使用頻度の少ないヘッダフィールドや、ルーティングヘッダ等のオプションヘッダは拡張ヘッダとして IPv6 ヘッダから追い出した。この結果、IPv6 ヘッダは固定長になり、転送時にヘッダのオプション解析にかかるオーバーヘッドを軽減している。

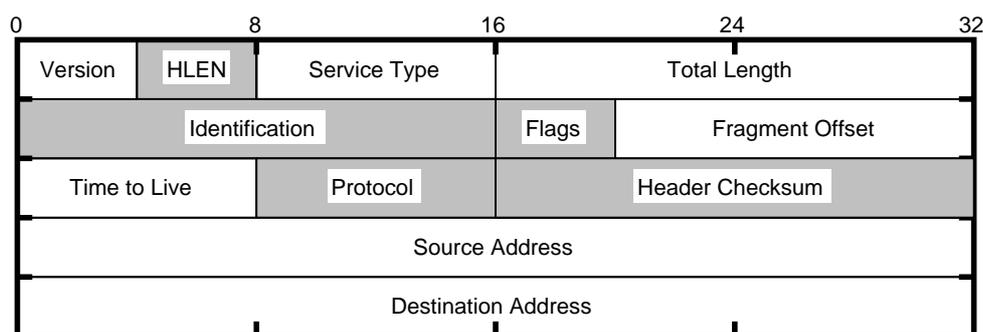


図 1.1: IPv4 のヘッダフォーマット

図 1.1 が従来の IPv4 のヘッダフォーマットである。斜線の部分が削除されたり、拡張ヘッダに追い出された部分である。図 1.2 が IPv6 のヘッダフォーマットである。新しく作られた部分を斜線で示している。なお、それぞれの図の行幅は 32 ビットである。また、幾つかのヘッダフィールドは、同じ目的であるものの、異なる名前に変更されている。

ヘッダを簡略化したことにより、アドレスの大きさが 4 倍になっているにもかかわらず、ヘッダ自体の大きさは 2 倍に抑えられている。

## 1.2.5 数珠つなぎ型のヘッダ構造

IPv6 では、ルーティングヘッダやフラグメンテーションヘッダなどを機能ごとに分割して、それぞれを一つの拡張ヘッダとした。これらの拡張ヘッダを、数珠つなぎ型 (daisy

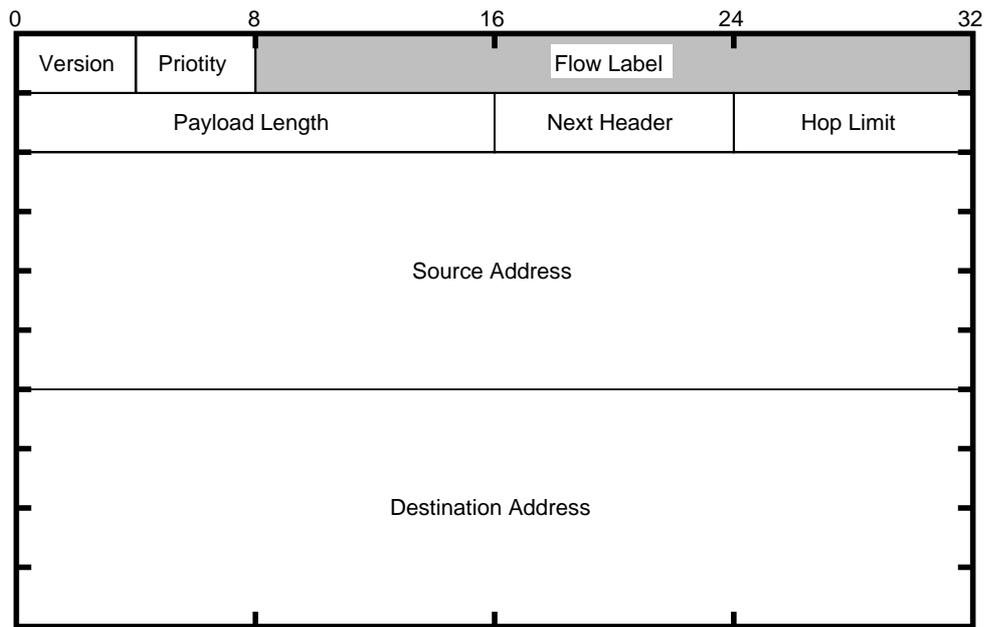


図 1.2: IPv6 のヘッダフォーマット

chain) に並べることによって、任意のヘッダを挿入することができる。この結果、IPv6 では、柔軟なオプションの設定を可能にしている。また、新たな拡張ヘッダを簡単に付加でき、拡張性にも優れている。

拡張ヘッダの使用方法を具体的に説明する。IPv6 ヘッダ、拡張ヘッダ内に、Next Header というヘッダフィールドが設けられている。そこには、次に続くヘッダの内容を規定された 8 ビットの整数値で示している [24]。

図 1.3 に数珠つなぎ型のヘッダ構造の一例を示す。

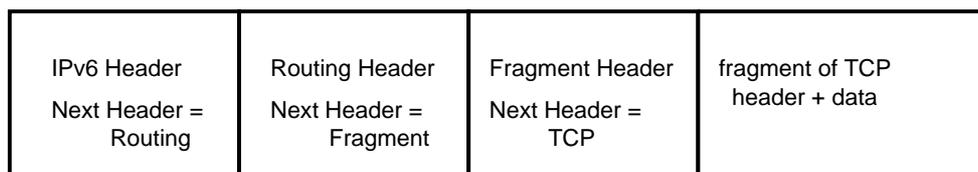


図 1.3: 数珠つなぎ型のヘッダ構造

文献 [19] では、以下の拡張ヘッダが規定されている。また、この順序でそれぞれが多くとも一回だけ現れる事を推奨している。

**Hop-by-Hop Options header** 途中経路を含む各計算機、ルータ上で参照される。

**Destination Options header(1)** 次述のルーティングヘッダで指定された計算機と、最終目的アドレスの計算機において参照される。

**Routing header** 配送経路を明示的に指定するためのヘッダ。基本的には IPv4 の source routing option と同一の働きをするが、よりきめの細かい設定ができる。

**Fragment header** フラグメンテーションされたパケットを再構成するためのヘッダ。IPv4 と違い、パケットは途中経路で分割されない。

**Authentication header** 認証を行うためのヘッダ。

**Encapsulating Security Payload header** 暗号化を行うためのヘッダ。このヘッダが付いていた場合、次のヘッダ以降は暗号化されて、データを盗聴できないようにする。

**Destination Options header(2)** 最終目的アドレスの計算機だけが参照する。

**upper-layer header** 上位層のプロトコルヘッダ。

### 1.2.6 隣接ホスト探索プロトコル

隣接ホスト探索プロトコル (Neighbor discovery protocol、以下 NDP) は、同一リンクに存在する計算機に対して、以下の情報を取得するために定められたプロトコルである。

- (1) データリンク層のアドレス
- (2) 非到達アドレス
- (3) ルータ情報

(1) は IPv4 の ARP と機能は同一であるが、IPv6 において ICMP の一部として実現されている。

(2) はネットワーク的に接続されなくなった計算機を自動的に検出する機能である。対象となる計算機がホストだった場合は、NDP テーブルから消去する。ルータであった場合は、デフォルトルートを変更する。

(3) に関しては、1.2.7 節で説明する。

### 1.2.7 アドレスの自動設定

IPv6 では、Plug & Play が新しく実現された。これは、ネットワークを簡単に構築するための機能である。新しく計算機を導入したとき、従来の IPv4 ネットワークでは手動で IP アドレス、デフォルトルート等の設定を行わなければならない。

しかし、IPv6 ではネットワークに計算機を接続すると、後は世界中で一意的なアドレスが自動的に計算機に付加される。この機能は、NDP の一部である二種類のパケットによって実現されている。

**Router Solicitation message** ルータから情報を送信してもらうように指示するメッセージ

**Router Advertisement message** ルータから送信される情報

Router Advertisement message にはルータの属するネットワークのアドレス等が含まれている。計算機はそれを利用して、自分自身にアドレスを設定する。大まかな動きを以下に示す。

1. 計算機はそのリンクでのみ使えるアドレス (link local address) を自分自身につける。このアドレスはインターフェースの MAC アドレスに基づいて設定されるので、そのリンク内で一意になる。
2. 自分自身が属しているリンクの全ルータに対して、計算機は router solicitation message を発行する。
3. そのメッセージを受け取ったルータは、送信者に対して router advertisement message を返す。
4. router advertisement message にはネットワークアドレスの情報が付帯している。この情報と MAC アドレスを組み合わせることによって、全世界で一意的なアドレスを決定する。

ネットワークのアドレスはルータによって提供されるので、あるリンクから別のリンクへ計算機を移動したときは、アドレスの再設定が自動的に行われる。

また、あるサイトがプロバイダを変更したとする。従来ならばそのサイトに属する計算機全体の IP アドレスを付け変えなければならなかった。しかし、IPv6 ではルータの設定を変更するだけでよい。

### 1.2.8 IPv6 における新機能のまとめ

IPv6 は単にアドレス長を大きくしたというだけではなく、アドレスの自動設定機能や、フローラベルによるパケット送信、セキュリティ機能などを新しく付加した。その結果、IPv6 ではネットワークをより便利に、効率的に、安全に使えるようになったといえる。

## 1.3 IPv6 のソフトウェア・アーキテクチャ

大規模かつ、急速に成長し続けているインターネットにおいて、短期間に IPv4 から IPv6 へ移行するのは不可能であり、また IPv4 は IPv6 の普及後も残留すると予想される。よって、IPv6 は IPv4 との共存を考えながら設計・実装されなければならない。そのためには、IPv4 と IPv6 のどちらの packets も処理できるようなデュアル・スタックを持つ計算機が必要となってくる。この節では、デュアル・スタックをサポートするためのソフトウェア・アーキテクチャについて議論する。

### 1.3.1 別プロトコルとしての IPv6

IPv6 と IPv4 の両者は、IP という名前を持ち、バージョン番号のフィールドを共有しているため、一見同一のプロトコルのように思える。しかし、我々は全くの別プロトコルであると認識している。

その理由は以下の通りである。

- ヘッダフォーマットの大幅な変更
- 拡張ヘッダの概念の導入

これらの結果として IPv6 と IPv4 の packets は、全く異なった手順で処理される。以下、IPv6 と IPv4 は別プロトコルであるという哲学の基に議論を進める。

### 1.3.2 インターフェースとプロトコルキュー

インターフェースはデータリンク層とネットワーク層の間に位置する層である。データの送信時は、ネットワーク層から受け取った packets を出力したいデータリンク層のフレームの形にして出力する。

受信時は、インターフェースはデータリンク層のフレームを受け取り、それを適切なプロトコルのプロトコルキューに振り分ける。プロトコルキューとは、各ネットワーク層がさまざまなインターフェースから packets を受信するために、packets を一時的に待機させておく場所である。

IPv4 と IPv6 のデュアル・スタックにおいて、プロトコルキューを共用するか、それぞれに用意するかで設計が全く異なって来る。

プロトコルキューを共用する場合、インターフェース層は IPv4 と IPv6 の packets を区別することなく、IP 用のキューにつなぐ。また、packets の入口が 1 つであるため、おのずと IPv4 と IPv6 の両方をサポートする IP 層を実装しなければならない。

プロトコルキューを独立に用意する場合、インターフェース層は IPv4 と IPv6 のフレームを区別し、取り出した packets をそれぞれのプロトコルキューにつなぐ。よって IPv4 と IPv6 は独立したネットワーク層の実装となる。

我々は、IPv4 と IPv6 を全く別のプロトコルであると認識しているので、プロトコルキューを分け、インターフェース層で振り分けるべきだと考えている。幸いにイーサネット等では、IPv4 と IPv6 には異なったフレームタイプが定義されているので、インターフェース層でパケットを振り分けられる。

また、プロトコルキューを分ける場合の利点としては、プロトコルキューまわりの実装が理解し易くなる他、既存の IPv4 のコードに影響を与えないことが挙げられる。WIDE Project では全ての実装が後者の実装になっている。

### 1.3.3 IP 層

前述したように、IP 層のアーキテクチャとして、以下の 2 つが考えられる (図 1.4)。

1. IPv4 のパケット処理に IPv6 のパケット処理を追加する。IP 層では、バージョン番号によって処理を変える。
2. IPv6 のパケット処理を独自に追加する。

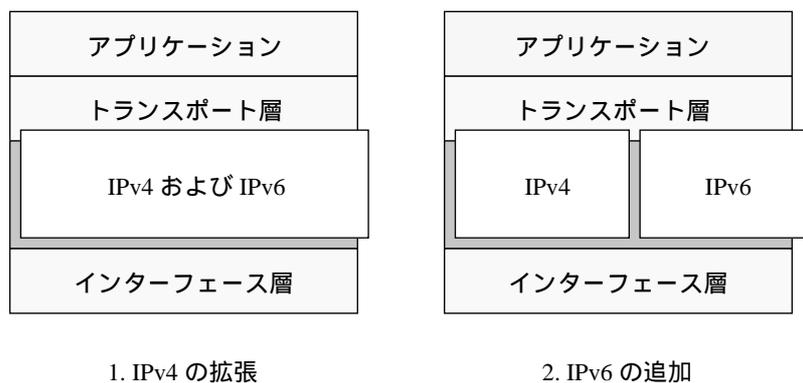


図 1.4: IP の実装方法

前者の利点は IPv4 のパケット処理を拡張するだけで実現できることである。また、IPv4 から IPv6 への移行期間中の協調処理が行ないやすいと思われる。

後者の利点は本来別プロトコルである IPv4 に影響を与えずに IPv6 のパケットを処理できることである。また、将来 IPv6 のネットワークが構成された時点において、IPv4 のコードを切り捨てられる。さらに前節でも述べたようにコードが完全に IPv4 から独立しているので理解が容易であり、デバッグをしやすい。

WIDE Project では、全ての実装が IPv6 を独立して処理するデュアル・スタックを採用している。

### 1.3.4 ICMPv6

ICMPv6 は従来の ICMP にいくつかの新しい機能を加えている。以下に ICMPv6 の主な機能を示す。

ICMPv6 は次の機能を持っている。

1. 転送エラーの報告
2. マルチキャストのグループメンバ管理機能
3. NDP パケットの送受信

ICMPv6 の転送エラーは、IPv4 内の ICMP にはない新しいメッセージである。マルチキャストのグループメンバ管理は IGMP プロトコルに実装されていたが、IPv6 では ICMPv6 に統合された。

NDP は物理アドレスと IPv6 アドレスの対応付けるプロトコルである。IPv4 の ARP (Address Resolution Protocol) に相当する。NDP は ICMPv6 の機能の一部として定義されており、ICMPv6 の次の 4 つのメッセージを利用する。

- Neighbor Solicitation(NS)
- Neighbor Advertisement(NA)
- Router Solicitation(RS)
- Router Advertisement(RA)

これまではホスト間の接続性を確立するのが第一目的だったので NS、NA のみを実装している。マルチキャストのグループ管理と、ルータと送受信を行う RS、RA は実装していない。

IPv4 において、ARP は特定のデータリンクアドレス解決できるデータリンク層のプロトコルであった。したがって、ARP の出力ルーチンは、データリンクに依存したコードを実装すればよかった。

それに対し NDP はデータリンクには依存しないように、ICMPv6 の一部として定義されている。このメッセージは、IPv6 のパケットとして送信される。そこで、NDP を IPv6 の出力ルーチンを用いて送信するのが自然に思えるが、この方法はうまくいかない。なぜなら、NDP を IPv6 の出力ルーチンは、仕様に従ってマルチキャストのアドレスをキーにしてまず経路制御表を引くが、この検索は成功するとは限らないからである。つまり、データリンクに閉じているパケットに対して、経路制御を禁止する必要がある。今回の実装ではこの機能を果たす IPv6 の出力ルーチンを用意して問題を解決している。

### 1.3.5 TCP/UDP

IPv4 と IPv6 用の TCP/UDP に機能の違いはない。ただし、IPv6 ヘッダにはチェックサムが存在しないため、UDP に関してはチェックサムの機能を必ず付け加えなくてはならない。

IPv4 と IPv6 が別々のネットワーク層として実装されていても、TCP/UDP は共有すべきルーチンが多い。実際に我々は、TCP/UDP のコードを若干改良しただけである。しかし、PCB に依存する部分は共有できないため、これは以下に述べる PCB の設計に大きく影響を受ける。

### 1.3.6 PCB

PCB はプロトコルの通信に関する情報を保持するデータ群である。それぞれのプロトコルに対して特定のデータ郡が定義されている。

実装方法として 2 つの方法がある (図 1.5)。

- (1) IPv4 の PCB を拡張する。
- (2) IPv6 の新しい PCB を定義する。

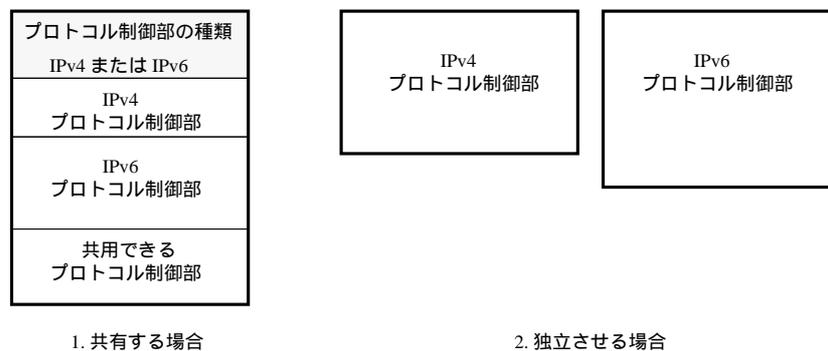


図 1.5: PCB の設計

- (1) の方法には以下の利点がある。
  - (a) PCB を操作するコードを共有できる
  - (b) IPv4 と IPv6 のプロトコル変換を行ないやすい
- (2) の方法には以下の利点がある。
  - (c) コードが理解しやすい。

(d) デバッグが容易である。

BSD の API [BSD-API] で提唱されている IPv4 ソケットと IPv6 ソケットの切替えが簡単に実装できる可能性があるため、(b) は重要である。BSD/OS2.1 の PCB とソケットは双方向のポインタで関連付けられており、一つのソケットが一つのプロトコル制御部に対応している (図 1.6)。つまり、(1) の枠組では、IPv4 で作成したソケットを IPv6 で使いたい時でも、PCB をそのまま利用できる。逆に (2) の方法では IPv4 の PCB をソケットと切り離れた後に、IPv6 のプロトコル制御部と接続しなければならない。

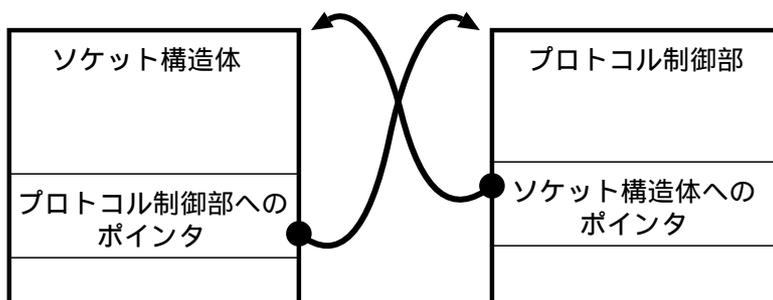


図 1.6: ソケットと PCB の関係

先程にも述べたように IPv6 は研究段階のプロトコルであるので、IPv4 と独立して実装しておく不都合が発生した時に問題の切り分けが容易になり、同様にデバッグも行ないやすい。また、経験上安定しているコードに手を加えると動作が不安定になる恐れがある。

この 2 つの設計はどちらも一長一短であり、どちらの設計方針が良いのかはこれから検討されるべき課題である。WIDE では慶應義塾大学の南と株式会社日立製作所の渡部らが前者の方針で実装し、奈良先端科学技術大学院大学の島とソニー株式会社の尾上が後者の実装を行なっている (表 1.2)。

表 1.2: PCB の実装

実装者		実装レベル	PCB
慶應大学	南 正樹	カーネル (独自)	共用
奈良先端科学技術大学院大学	島 慶一	カーネル (独自)	分離
日立製作所	渡部 謙 (代表)	カーネル (移植+独自)	共用
ソニー株式会社	尾上 淳	アプリケーション (独自)	分離

### 1.3.7 ソケット

アプリケーション層とトランスポート層の間を取り持つ機構としてソケットがある。ソケットは、様々なプロトコルのアドレスを抽象化して表現できるので、新しいプロトコルでも柔軟に対応可能である。

図 1.7に、IPv4のソケットアドレス構造体を示す。また、図 1.8に IPv6 用に新たに定義したソケットアドレス構造体を示す。IPv6 では、アドレスが長くなった他、フローラベルが追加されているのが分かる。

```
struct sockaddr_in {
    u_char  sin_len;           /* struct sockaddr_in の大きさ */
    u_char  sin_family;       /* プロトコル識別子 */
    u_short sin_port;         /* トランスポートポート番号 */
    struct  in_addr sin_addr;  /* 32ビット IPv4 アドレス */
};
```

図 1.7: IPv4 ソケットアドレス

```
struct sockaddr_in6 {
    u_char  sin6_len;         /* struct sockaddr_in6 の大きさ */
    u_char  sin6_family;     /* プロトコル識別子 */
    u_short sin6_port;       /* トランスポートポート番号 */
    u_long  sin6_flowinfo;    /* IPv6 フロー情報 */
    struct  in6_addr sin6_addr; /* 128ビット IPv6 アドレス */
};
```

図 1.8: IPv6 ソケットアドレス

### 1.3.8 ソフトウェア・アーキテクチャのまとめ

我々は、IPv4とIPv6は異なるプロトコルという立場に立って、プロトコルキューとネットワーク層を独立に実装すべきであると提案した。また、インターフェース層は、IPv4とIPv6の受信フレームを区別し、それぞれのプロトコルキューへパケットを振り分けるべきである。

TCP/UDPは、IPv4とIPv6に共通である。TCP/UDPが利用するPCBは、共有する方法と別々に用意する方法の2つが考えられるが、現時点ではどちらが優れているか結論は出ていない。

3つの実装と1つのエミュレータを、独立に開発してきたため、PCBなどの実装は統一されていない。これは、PCBなどの設計方針を議論するには好都合である。今後は、IPv6に関するソフトウェア・アーキテクチャについて、さらに考察していく必要がある。

## 1.4 現状と成果

インターネットの研究では、標準に準拠して実装するだけでなく、実際に他の実装との通信を確認することが重要である。このため我々は、相互通信実験を繰り返し相互接続性を検証してきた。

まずはじめはWIDEプロジェクト内部で相互通信実験を行なった。V6分科会では、参加者がそれぞれ独自に開発しているため、それぞれの実装は異なっており相互通信実験としての意味がある。

### 第一回 V6 分科会 ワークショップ

日程：1995年12月14日～15日

場所：慶応義塾大学 湘南藤沢キャンパス

### 第二回 V6 分科会 ワークショップ

日程：1996年1月29日～30日

場所：慶応義塾大学 湘南藤沢キャンパス

次に、ニューハンプシャー大学のIOLで実施された第一回IOL相互通信実験に参加した。これは、より多くの実装と相互通信実験を行なうと同時に、WIDE ProjectでIPv6を実装開発していることを世界にアピールすることを目的としたものである。

### 第一回 IOL 相互通信実験

日程：1996年2月5日～9日

場所：ニューハンプシャー大学 Research Computing Center 内 InterOperability Lab.

以下では、それぞれの相互通信実験の結果について述べる。

### 1.4.1 第一回 V6 分科会 ワークショップ

1995年12月14日～15日の2日間、慶応義塾大学において第一回V6分科会ワークショップを開催し、相互通信実験を行なった。

第一回の相互通信実験では、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なかったため、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。2つのノード間でIPv6パケットを交換するには次の3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) ネットワーク層アドレスとデータリンク層アドレスが対応づけられること。

このうち (3) については、従来の IPv4 では ARP によって処理されていたが、1.2節で述べたように IPv6 では NDP が処理している。NDP がネットワーク層アドレスからデータリンク層アドレスを得るためには 2 つのノード間で適切な NDP パケットのやりとりが必要となる。第一回の実験では上記の各項目の確認のため、ping コマンドを用いて ICMP Echo Request、Reply パケットの交換を確認することで相互接続性を検証した。

図 1.9 に今回の実験時のネットワーク構成を示す。また、図 1.10 に 2 つのノード間のパケットの送受信シーケンスを示す。

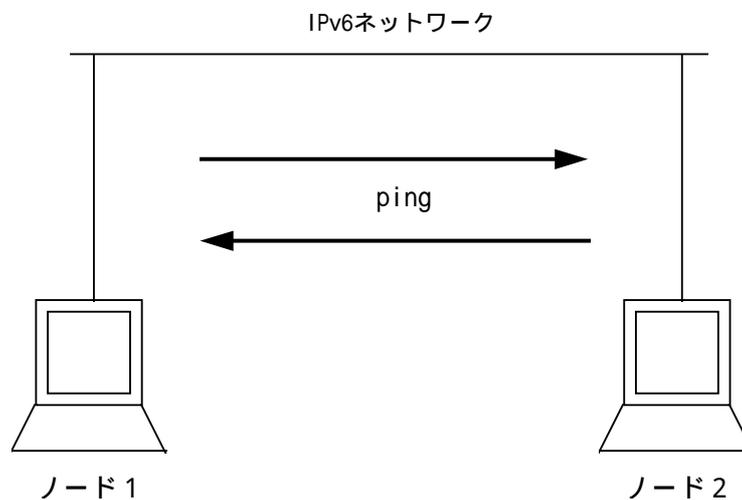


図 1.9: 実験ネットワークの構成

図 1.9 と図 1.10 を用いて 2 つのノード間の IPv6 パケットの交換の実験手順を説明する。

- ノード 1 および ノード 2 において、ndp コマンドにより NDP テーブル内にお互いのエントリが登録されていないことを確認する。もし、エントリが既に登録されていれば、ndp コマンドを用いて削除する。
- ノード 1 または ノード 2 において、tcpdump コマンドによりネットワーク上を流れるパケットを記録する。
- ノード 1 において、ping コマンドを用いて、ノード 2 へ ICMP Echo Request を送信する。

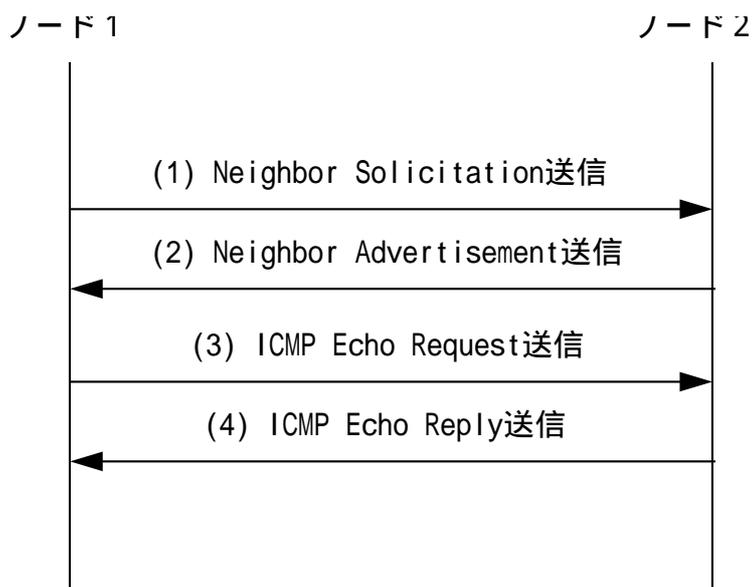


図 1.10: メッセージ交換シーケンス

- ノード 1 において、ping コマンドの結果より、ノード 1 の ICMP Echo Request に対するノード 2 からの ICMP Echo Reply を正常に受信したことを確認する。
- ノード 1 またはノード 2 において、実行していた tcpdump コマンドの出力結果により、以下の項目を確認する。
  - パケットのフォーマットが正しいこと。
  - パケットのシーケンスが図 1.10 に示したシーケンスに従っていること。
- ノード 1 およびノード 2 において、ndp コマンドにより NDP テーブル内にお互いの IPv6 アドレスと対応するデータリンク層アドレスが登録されていることを確認する。

図 1.11 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。例えば、図 1.11 中の奈良先端科学技術大学院大学から慶應義塾大学への矢印は、奈良先端科学技術大学院大学のノードから慶應義塾大学のノードへパケットの送信を開始し、図 1.10 に示したシーケンスに従ってパケットを交換し、IPv6 アドレスからデータリンク層アドレスへの対応づけと ICMP Echo Request、Reply パケットの交換が正常に動作したことを表す。図 1.11 に示す通り、全ての実装の間で ping の送受信ができることを確認した。

第一回 V6 分科会 ワークショップでは、これまで別々に開発が進められていた各実装の間で、IPv6 アドレスからデータリンク層アドレスへの対応づけが問題なく動作した。その結果、IP 層のレベルでの相互接続性が検証された。

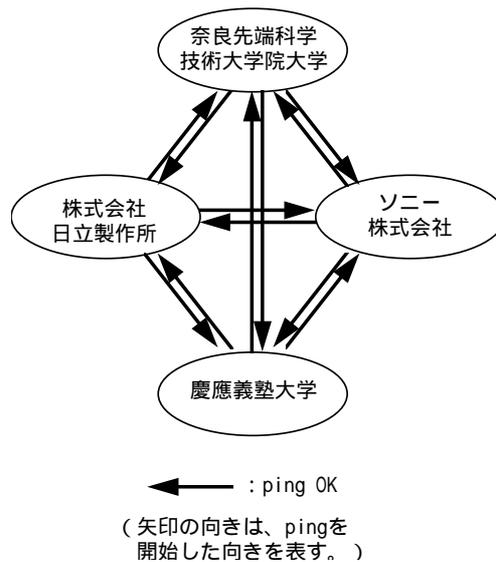


図 1.11: ping の実験結果

また、本ワークショップで、奈良先端科学技術大学院大学の島によってその時点でのドラフトの仕様では NDP がアドレス解決に失敗する可能性が指摘された。この問題を NDP のドラフトの執筆者である Thomas Narten、Erik Nordmark 両氏に報告したところ、仕様の誤りであることが判明し以降のドラフトでは修正された。

#### 1.4.2 第二回 V6 分科会 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 V6 分科会 ワークショップを開催し、相互接続実験を行なった。

第一回 V6 分科会 ワークショップにおける相互接続実験では、同一 LAN 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い、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であることを確認した。そこで今回は、異なる LAN 上に存在するホスト同士の間で 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3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** ホスト 1 から異なる LAN 上に存在するホスト 2 に対して発行した ping コマンドが経路上のルータを介してホスト 2 に送られ、そのままホスト 1 に送り返されることを確認する。

**telnet による TCP の通信テスト** BSD/OS 2.0 の telnet と telnetd 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 2 に対して発行した telnet コマンドがホスト 2 の telnetd によって受け付けられ、ホスト 1 からホスト 2 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** BSD/OS 2.0 の 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 2 の UDP ポート 7 に対して発行した UDP データグラムが、ホスト 2 の inetd によって受け付けられ、そのままホスト 1 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。

図 1.12 ~ 図 1.14 に実験結果を図示する。

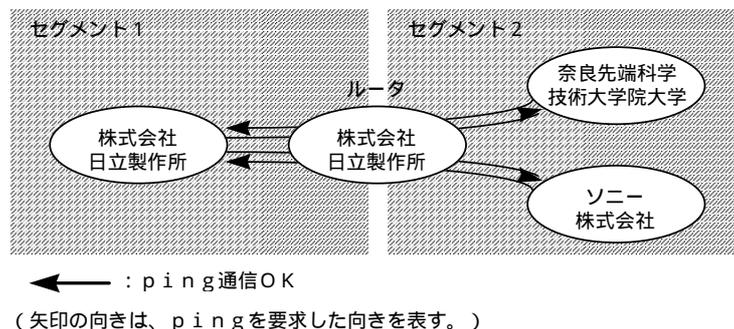


図 1.12: IPv6 パケットの転送実験の結果

- IPv6 パケットのフォワーディング処理を実装していた株式会社日立製作所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科学技術大学院大学、ソニー株式会社の尾上、株式会社日立製作所の実装の間で 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  
奈良先端科学技術大学院大学と株式会社日立製作所の実装の間では、telnet/telnetd

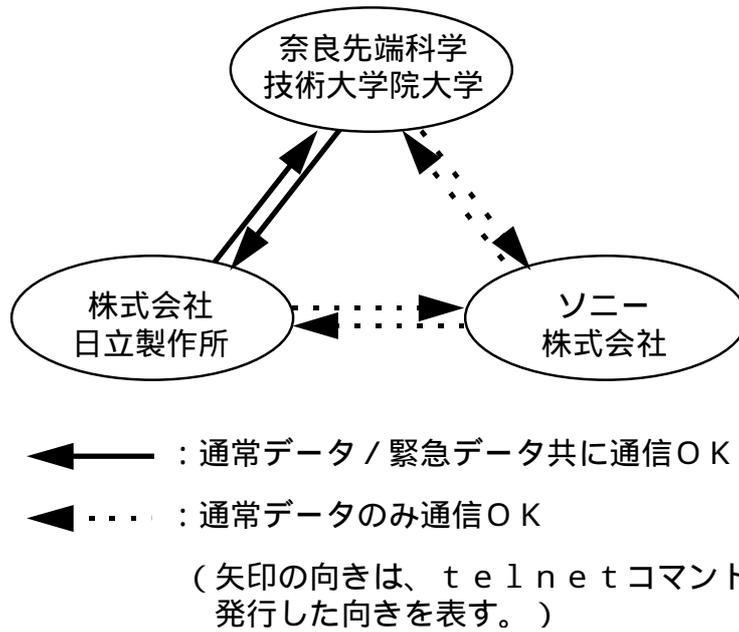


図 1.13: TCP の実験結果

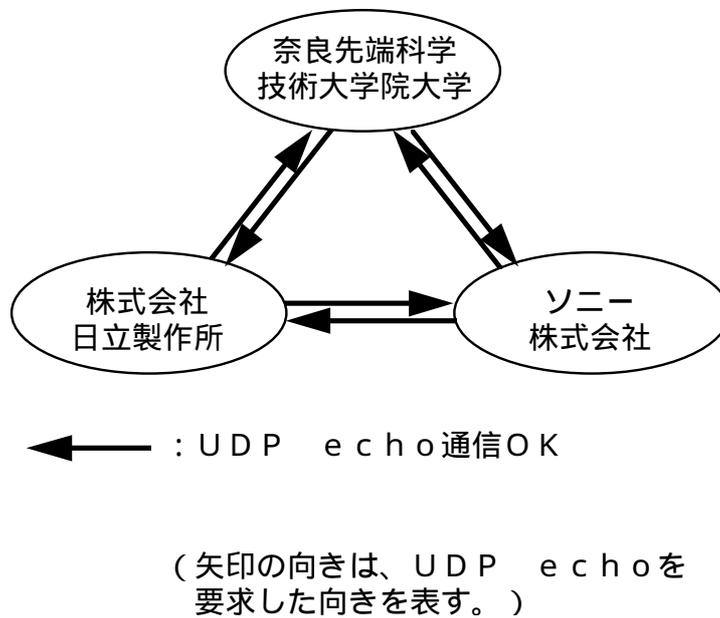


図 1.14: UDP の実験結果

共に問題なく動作し、TCP の通信が確認できた。ソニー株式会社の尾上の実装は、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

奈良先端科学技術大学院大学、ソニー株式会社の尾上、株式会社日立製作所の全ての実装の間で UDP の通信ができることを確認した。

株式会社日立製作所の実装をルータとして用い、IPv6 フォワーディング機能によるネットワーク層の通信が可能であることを確認できた。また、TCP のうち未実装の一部機能を利用する通信ができなかったことを除き、奈良先端科学技術大学院大学、ソニー株式会社の尾上、株式会社日立製作所の独立した 3 つの実装の間で通信できることを確認した。すなわち、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ており、ネットワーク層のプロトコルとして問題がないことを確認した。

### 1.4.3 第一回 IOL 相互通信実験

1996 年 2 月 5 日～9 日の 5 日間、ニューハンプシャー大学の 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。V6 分科会 からも 3 つ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院大学
- 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 Research Center

IOL での通信実験で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、IOL が用意した NDP の動作を試験するプログラムにより、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ため、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)

図 1.15 に実験機器の接続を示す。IOL の基幹 LAN に直接 IPv6 のパケットを中継できるソフトウェアルータを 3 台接続し、その 3 台のルータの配下のセグメントに参加者それぞれの PC/WS を接続した。参加者それぞれの PC/WS には IPv4 コンパチブルアドレスが割り当てられた。また、それぞれのセグメントにはパケットのモニタリングおよびテストパケットの生成のために IOL が用意した WS が 1 台ずつ接続された。

**NDP の動作確認** 図 1.16 に NDP のうちアドレス解決の状態遷移図を示す。図 1.16 に示すようにアドレス解決の状態遷移は遷移の条件、タイミングが細かく規定され複雑である。IOL が用意した NDP のテストプログラムは、ある状態から別の状態へ遷移する際の条件およびタイミングを個々に検証する。また、テストプログラムが動作する WS と被テスト機を同一リンク上に接続してテストパケットを被テスト機に送り、その応答を観測する。

**ping コマンドの動作確認** ping コマンドの動作確認は、IPv4 互換アドレス、all-router アドレス、all-node アドレスに対して各自 ping(Echo Request) パケットを送信し、その応答を観測した。また、テストプログラムが動作する WS から被テスト機に対して ping コマンドを用いてフラグメントしたパケットを送信し、正しく応答を返せるか調べた。

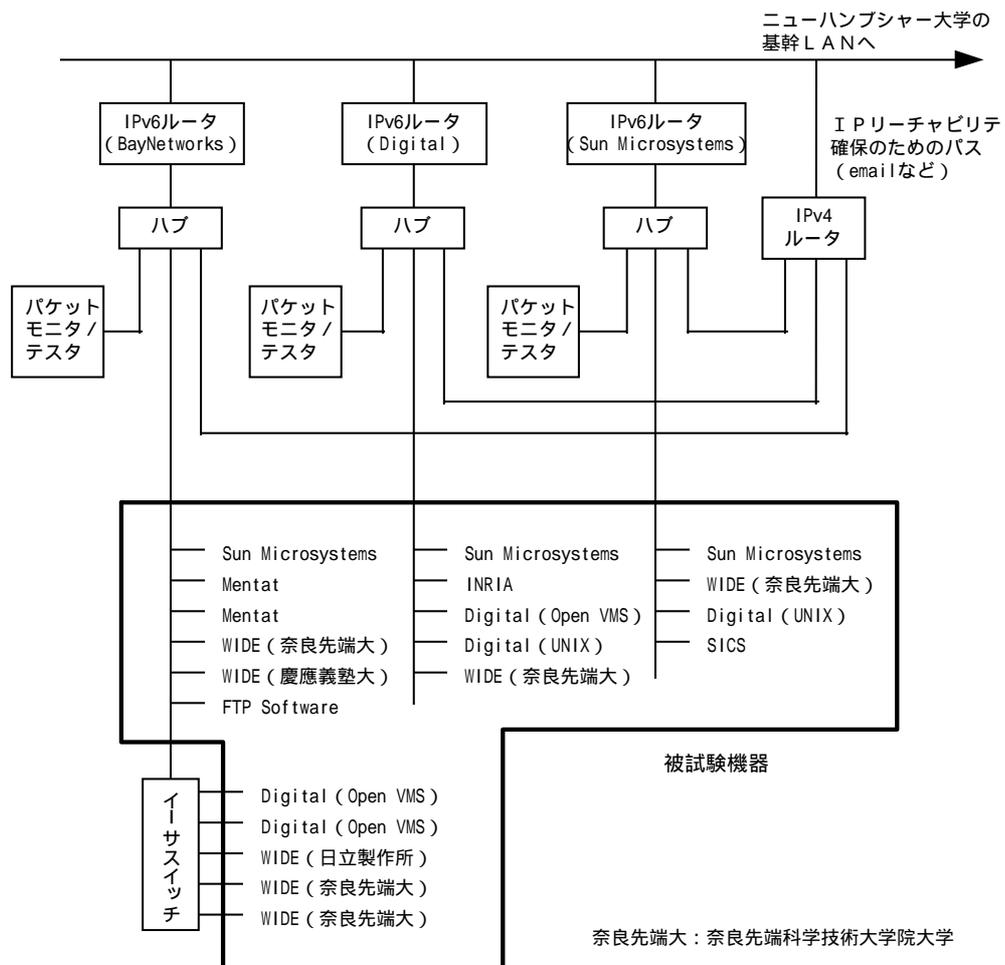


図 1.15: IOL における IPv6 実験機器の接続

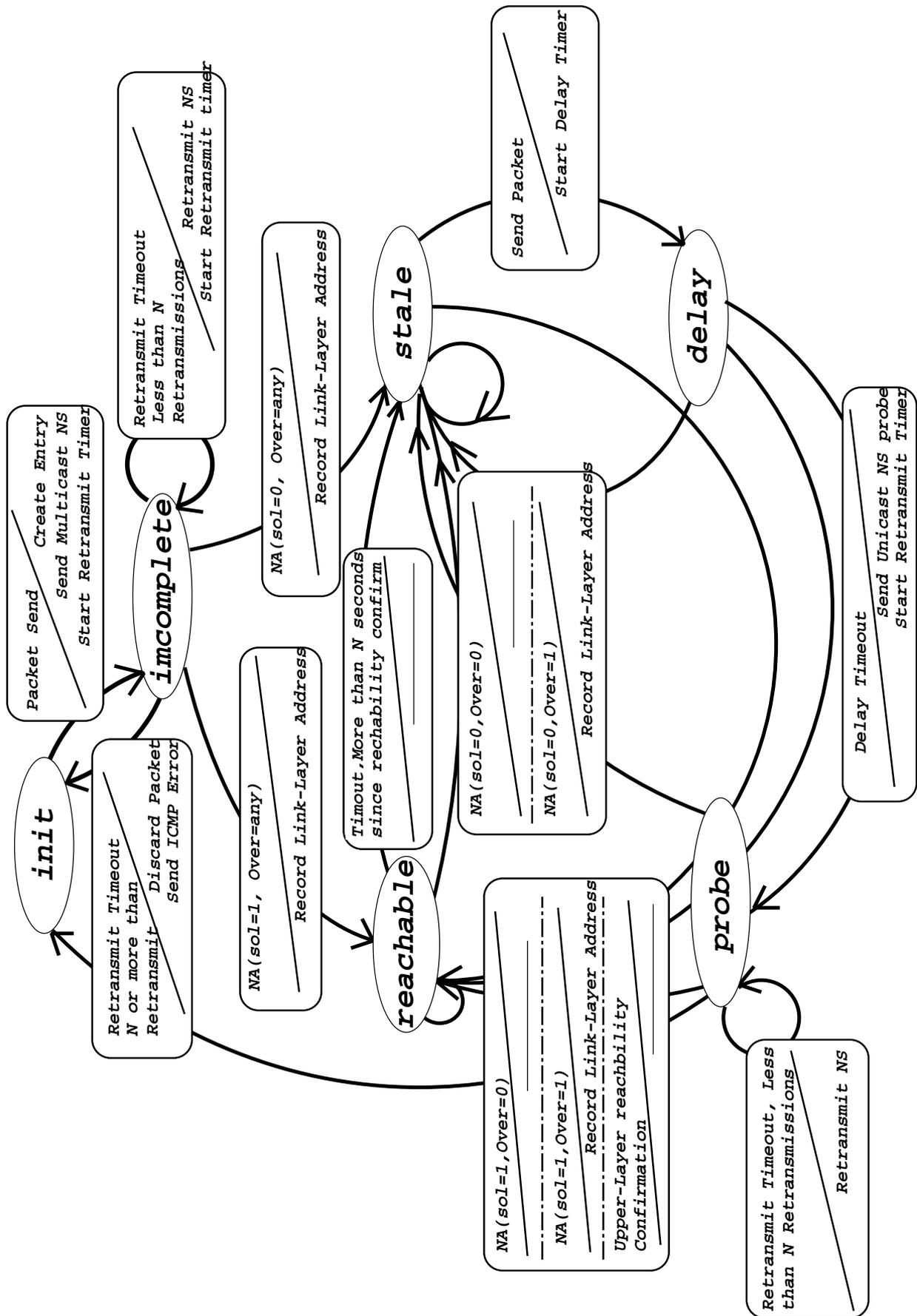


図 1.16: NDP の状態遷移

**telnet コマンドの動作確認** telnet コマンドの動作確認は、IPv4 互換アドレスを使い、各自相手の端末に対して telnet コマンドを実行して通信できるか調べた。

- NDP の動作確認

**慶應義塾大学の実装** 当初、mbuf の扱いを間違えていたため、テストを始めるとパニックが発生してダウンしていた。しかし、その場で修正しテストを実施できた。テストの結果、Host Unreachable Detection と NS (Neighbor Solicitation) メッセージが出るタイミングが間違っていることを指摘されたが、その他は正しく実装されていることがわかった。なお、このタイミング間違いは IOL から帰国後すぐに修正した。

**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、株式会社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。

このため、IOL から帰国後フリーの NDP ソースを使うことをやめ、初めから作り直した。そして、その後の実験でアドレス解決の状態遷移が正しく行なえるようになったことを確認した。

- 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。

NDP の内、アドレス解決機能の実験、ping を用いた基本的な通信機能の実験、telnet を用いた TCP の動作実験を行なった結果、それぞれが仕様を正確に実装できていることを確認できた。さらに、この実験で洗い出した誤り、不正確さを元に修正し、相互接続性を向上できた。

#### 1.4.4 現状と成果のまとめ

3 回の相互通信実験を通じて我々は相互接続性の向上に努めてきた。この結果、個々の実装により進捗に違いはあるものの、次の機能を実装し動作を確認することができた。

- ヘッダ処理を含む基本的なパケット送受信機能
- NDP(アドレス解決機能のみ)
- ICMP
- 転送機能
- TCP/UDP およびソケットインターフェース機能

さらに、国際的な場に我々の実装を持ち込み相互通信実験を行なったことで、多くの課題は残るものの、標準仕様を正確に実装するための足がかりを築くことができたと考える。

### 1.5 6Bone の構築

IPv6 のルータの機能を実装した後は、実際にネットワークを構築し運用を通じて、テストを行ったり、問題点を発見したりしていく必要がある。このネットワークは、広域な IPv4 ネットワークを活用し、世界中の IPv6 研究者が共有する実験ネットワークとなるべきである。マルチキャストの実験ネットワークが MBone と呼ばれているように、IPv6 の研究者間の架け橋となるネットワークを 6Bone と呼ぶ。本章では、6Bone に関する世界的な活動、および、WIDE Project が予定している具体的な活動内容について述べる。

#### 1.5.1 世界規模での協力

Digital 社の Jim Bound 氏が中心となって、世界的な 6Bone を構築することを予定している。6Bone は、アジア、アメリカ、ヨーロッパに分かれ、それぞれの地域を、WIDE Project の村井純、Xerox PARC の Steve Deering 氏、そして、Christian Huitema 氏が取りまとめる。現状では、どの地域も予算を確保する段階である。

#### 1.5.2 WIDE Project の計画

WIDE Project では、次の 3 段階に分けて 6Bone の構築に取り組んでいく予定である。

- (1) IPv4 と独立した IPv6 のみの 6Bone の構築
- (2) トネルを用いた IPv4 上の 6Bone への移行

## (3) 他の地域との接続

(2) および (3) は、Mbone と同じであるから説明は省略する。(1) に関して、現在の予定されている IPv6 のみのネットワークを図 1.17に示す。

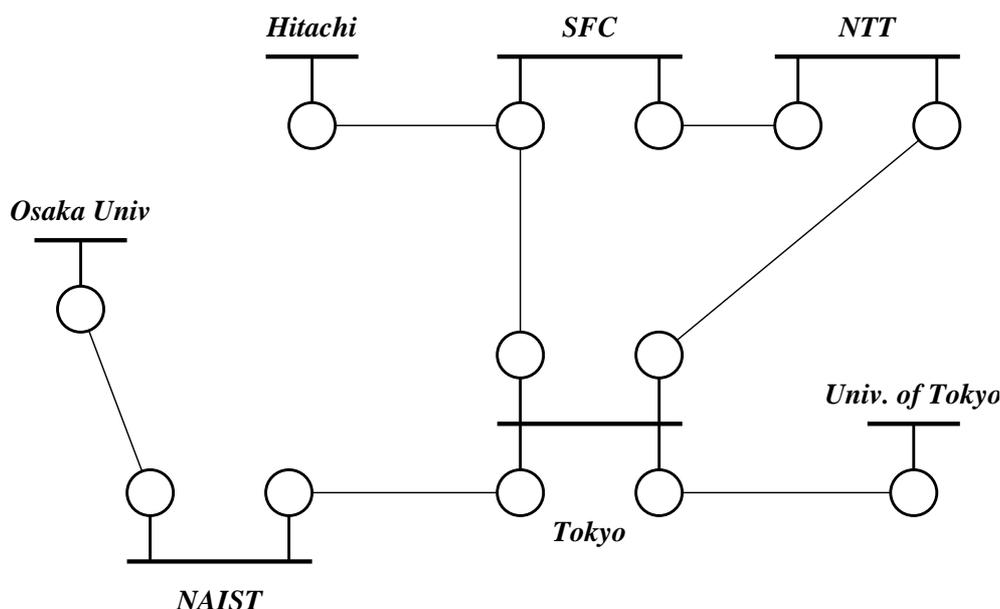


図 1.17: WIDE Project における初期の 6Bone

この初期の 6Bone は、V6 分科会の開発の中心となる奈良先端科学技術大学院大学、慶應義塾大学および株式会社日立製作所を結ぶほか、東京大学や大阪大学、NTT などにも延びる。WIDE Project のバックボーンと重なる部分については、専用線の多重化装置を用いて、論理的に IPv4 とは切り離された回線を用意する。

Riscom などの高速シリアルボードを搭載した PC AT 互換機に BSD/OS を載せて、6Bone のルータとして利用する。この BSD/OS には、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コードを用いて IPv6 の機能が拡張されている。PC AT 互換機には、基本的に複数の高速シリアルボードを挿さず、Ethernet を用いて NOC でのパケットを交換する。これは、高速シリアルボードによって、問題が起こらないようにするためである。

6Bone にループがない時期では、動的な経路制御は必要ないので、静的に経路を制御する。ループを作るまでには、IPv6 用の経路制御デーモンを開発し、動的経路制御に切り替える予定である。また、(2) への移行に備えて、トンネル技術も開発する必要がある。



### 1.5.3 6Bone のまとめ

6Bone は、世界中の IPv6 の研究者にとって必要不可欠であり、できるだけ早く構築に着手する必要がある。世界的な視点で見ると、アジア、アメリカ、ヨーロッパが中心となつて、6Bone の黎明期を支えていく予定である。

アジアでは、WIDE Project が中心となり、6Bone の構築に取り組んでいくが、これは V6 分科会 の第二段階への第一歩でもある。我々は、まず IPv4 とは独立した 6Bone を構築する。そして、徐々にトンネルなどの技術を用いて IPv4 上に移行し、他の地域と接続していく予定である。

## 1.6 V6 分科会 の活動のまとめ

この章では、これまでの V6 分科会 の活動内容と今後の予定について報告した。

我々は、IPv4 と IPv6 は、IP という名とバージョンフィールドを共有しているものの、全く別のプロトコルであると認識している。この哲学の基にソフトウェア・アーキテクチャを議論し、これまでに確立した設計方針について述べた。別のネットワークプロトコルには、データリンク層でそれぞれのフレームタイプを定義して区別しなければならない。また、別々のネットワークプロトコルキューを用意し、ネットワーク層も全く独立させて実装すべきである。TCP や UDP は共通であるため、これらが利用する PCB については、今後さらに議論して性質を明らかにする必要がある。

これまでに、慶應義塾大学の南、奈良先端科学技術大学院大学の島、株式会社日立製作所の渡部が中心となって、BSD/OS 上に IPv6 の機能を独立に実装した。これらは、第一回 V6 分科会 ワークショップ、第二回 V6 分科会 ワークショップ、第一回 IOL 相互通信実験を経て着実に安定化し、また、機能の充実化もはかられている。

我々は、第一段階であるホストとしての機能の実装はほぼ終了したと考えている。今後は、6Bone の構築を足掛かりに、第二段階であるルータとしての機能の実装に着手していく。