# 第VI部

# IP version 6

# 第 6 部
# IP version 6

## 第 1 章　はじめに

この章では IPv6 分科会の成果について報告する。IPv6 分科会が研究の対象としているのは、IPv6、IPsec、Mobile IPv6 である。これらに関し、実装、標準化活動、普及活動に取り組んでいる。

IPv6 分科会の報告書は以下のように構成される。

第 2 章では、IPv6、IPsec、Mobile IPv6 の実装状況について述べる。標準化活動への関わりについて第 3 章で説明し、その成果物である RFC3142、RFC3146、RFC3178 をそれぞれ第 4 章、第 5 章、第 6 章に掲載する。我々が執筆したインターネットドラフト (以下ドラフトと略記) に関しては、それぞれの要約を第 6 章にまとめる。第 8 章 では、普及活動に関係のある NetWorld+Interop 2001 IPv6 ShowCase と Global IPv6 Summit in Japan に関し報告する。

## 第 2 章　実装状況

この章では、KAME プロジェクトでの IPv6、IPsec、Mobile IPv6 の実装状況について述べる。

### 2.1 IPv6

KAME プロジェクトは 1998 年 4 月から IPv6/IPsec 参照コードの開発実装に注力している。KAME 発の IPv6/IPsec コードは 4 つの BSD、MacOS X、および多くのベンダルータに組み込まれており、IPv6 の普及に大きな役割を果たしている。本年度も継続して開発を進めており、各種ドラフトの仕様変更への追従、バグ修正および BSD への反映、実験的/先進的な機能の実装などを活発に続けている。

- DHCPv6
- Multicast DNS
- ip6.arpa への移行
- スコープ付きアドレスアーキテクチャへのよりよい対応
- 始点アドレス選択アルゴリズムの改善

### 2.2 IPsec

KAME プロジェクトでは IPv6/IPv4 に対応した IPsec を実装している。

2000 年後半に NIST が次世代暗号化アルゴリズムの標準として AES を発表した。また 2001 年には AES の鍵長に相応しいセキュアハッシュ関数として SHA2 を発表した。KAME プロジェクトはこの 2 つの暗号アルゴリズムをいち早く実装した。

現在、IETF の IPsec 分科会は AES と SHA2 を利用するための標準化を進めている。また、AES の鍵長を交換するために相応しい DH 交換方式のグループを 3 つ提案している。KAME プロジェクトでは分科会の提案した AES の変換方式を ESP に実装し、新しい DH 交換方式のグループをすべて IKE に実装した。SHA2 の利用方法は標準化が始まったばかりなので、KAME プロジェクトでは試験的に実装している。

2001 年 6 月の NetWorld+Interop 2001 では、各ベンダーの製品と相互接続できるデモをした。

2001 年 8 月にフィンランドで開催された相互接続テストでは、AES を使った ESP の接続性の検証と新しい DH 交換方式のグループを使って IKE の接続性を検証した。どちらも問題なく継がることを確認した。また IKE に AES を試験的に実装し接続性を検証した。SHA2 に関しては実装しているベンダーが少なかったものの、利用方法の問題点がいくつか明らかになった。

2002 年 1 月には TAHI プロジェクトによる相互接続性実験に参加した。

### 2.3 Mobile IPv6 の状況

Mobile IPv6 の最新仕様に基いた参照コードの提供を目標に、KAME プロジェクトでは KAME の Mobile IPv6 コードを再設計/実装した。元々、KAME

**表 2.1**.　Mobile IPv6 の実装経過

| | |
|---|---|
| 5/初旬 | KAME/Mobile IPv6 スタック作成作業開始 |
| 5/28 | snap-users で Mobile IPv6 担当宣言 |
| | KAME/Mobile IPv6 スタックの作成を承認してもらう |
| 6/10 | KAME から Ericsson/Mobile IPv6 を削除 |
| 8/3 | KAME/Mobile IPv6 パッチを KAME repositry に統合 |
| 10/中 | 基本的な機能の実装終了 |
| 10/15 | KAME/Mobile IPv6 alpha リリース (ID-14 対応) |
| 11/中 | 後方互換のための ID-13 対応 |
| 12/末 | ID-15 対応 |
| 1/23 | TAHI 相互接続完了 |
| 2/20 | KAME/USAGI/SFC 間での認証データ相互接続検証 |

には Ericsson から提供された Mobile IPv6 の実装が組み込まれていたが、その他にも NEC、SFC の実装が存在しており、コードの統合が望まれていた。

5 月に再設計/実装を開始し、8 月には再設計された実装を KAME にマージした。10 月には基本機能の実装を完了し、alpha 版をリリースしている。

また、Mobile IPv6 の相互接続性を検証するため、2 回の相互接続実験の機会を作った。

ひとつは、2002 年 1 月 23 日から 26 日に横浜で開催された、TAHI プロジェクトによる相互接続テストである。KAME の Mobile IPv6 と関係する他の組織として、慶應大学 (SFC)、NEC、USAGI プロジェクトも参加している。

ふたつめは、2002 年 2 月 20 日に慶應大学で実施したもので、認証データによるシグナリングデータの保護機能の相互接続性を検証した。実験には、KAME、USAGI、慶應大学 (SFC) が参加している。この実験で各実装のミスが発見され、ミスを修正することで 3 者の相互接続性が検証された。

現在 Mobile IPv6 は、シングナリングデータ保護に PKI/IPsec や事前認証などの技術を使わない、軽量な認証の仕組みを導入しようとしている。今後も引き続き議論に参加し、仕様策定および、実装による検証を続けていく。

これらの経過を表 2.1 にまとめる。

### 2.4 ISC との共同開発

IPv6 の普及にとって、DNS の IPv6 への対応は重要事項である。利用されている DNS サーバのほとんどが ISC(Internet Software Consortium) の BIND である点を考えると、ISC との共同開発を整えることの意義は大きい。

この度、ISC との交流を深めるため、東芝の神明が ISC へ出向した。期間は、2001 年 3 月 23 日〜3 月 29 日と 2001 年 4 月 9 日〜5 月 10 日の 2 回である。出向期間中に、以下のような仕事を遂行した。

- BIND9 の内容理解
  - KAME だけでもコードを改良できるように
- IPv6 関連での BIND9 への修正
  - A6、IPv6 ACL、mapped アドレス、IPv6 マルチキャスト／エニーキャスト・アドレスの扱い
  - リゾルバライブラリの scope 付きアドレスのサポート
  - mDNS
- bind9 の性能調査
  - root DNS サーバでの負荷にも耐えられる程度の性能向上がゴール
  - ボトルネックがどこかを調べて、改善の目処までつけた
- {ftp,www}.isc.org を IPv6 対応にする

## 第 3 章　標準化活動への関わり

WIDE が実装を進めている分野は、IPv6、IPsec、Mobile IPv6 に大別できる。これらの分野での標準化活動と WIDE の関わりについてまとめる。

IPsec と Mobile IPv6 の分野では、主に実装者の立場にある。ドラフトが公開されると、それを実装

し実際に利用してみて、仕様に不具合があればそれをフィードバックしている。

IPv6 の分野では、このような活動に加えて、ドラフトを執筆し IETF で発表するなど、仕様の策定にも積極的に関わっている。

現在の IETF では、最初のドラフトが公開されてから RFC になるまで 3 年程度かかることが多い。以下の説明では、2001 年だけでなく、長年継続しているものも含まれていることに注意されたい。

2001 年の 6 月に、WIDE のメンバーが書いたドラフトが初めて RFC となった。執筆者は萩野と山本、内容はトランスレータ、番号は 3142 である。このトランスレータを山本が初めて実装したのは 1997 年であるから、仕様策定にいかに時間がかかるか分かるだろう。

続いて、他の 2 つのドラフトが、RFC 3146、RFC 3178 となった。この 3 つの RFC を、それぞれ独立した章として、この章に引き続き掲載する。

ドラフトの中には、個人名で出しているものもあるし、IPv6 分科会のドラフトとして公開されているものもある。

特筆すべきは、WIDE のメンバーのみでの執筆ではなく、他の組織の方と共著になっているドラフトも存在することである。これは、WIDE プロジェクトが IETF で認められ、各分科会に溶け込んでいる象徴ではないかと思う。

たとえば、対向ネットワークにおいて、パケットがループしてしまう問題の解決案は、WIDE の萩野、神明、そしてマイクロソフトの Zill 氏の連名となっている。

また、WIDE が独自に出していたドラフトが、分科会の主要なドラフトにマージされた例もある。たとえば、尾上、神明は、スコープの表記方法を提案するドラフトを出していた。これは現在、スコープのアーキテクチャに関するドラフトに取り込まれており、執筆者として上記二名が加わっている。

また、元々 WIDE のメンバーはドラフトの著者には加わっていなかったが、そのドラフトの責任者に任命されたケースもある。API のドラフトがその例であり、現在神明が責任をもって更新作業にあたっている。

なかなか標準化が進まなかった DNS サーバ探索の分野では、この問題の解決に特化するチームが編成された。WIDE からは、尾上、萩野、神明が参加し、テレコンなどを利用したミーティングに参加しながら、仕様策定に貢献してきた。成果は、マイクロソフトの Thaler 氏と萩野の連名で、ドラフトとして公開されている。

日本の ISP が IPv6 の個人用サービスを開始し、家庭ネットワークを IPv6 インターネットに接続できるようになった現在、ISP からの IPv6 アドレスの自動割り当てが重要な課題となっている。この要求仕様は、現実の問題に直面している日本から提案する予定である。

## 第 4 章　RFC 3142: An IPv6-to-IPv4 Transport Relay Translator

### 4.1 Abstract

The document describes an IPv6-to-IPv4 transport relay translator (TRT). It enables IPv6-only hosts to exchange TCP,UDP traffic with IPv4-only hosts. A TRT system, which locates in the middle, translates TCP,UDP/IPv6 to TCP,UDP/IPv4, or vice versa.

The memo talks about how to implement a TRT system using existing technologies. It does not define any new protocols.

### 4.2 Problem domain

When you deploy an IPv6-only network, you still want to gain access to IPv4-only network resources outside, such as IPv4-only web servers. To solve this problem, many IPv6-to-IPv4 translation technologies are proposed, mainly in the IETF ngtrans working group. The memo describes a translator based on the transport relay technique to solve the same problem.

In this memo, we call this kind of translator “TRT” (transport relay translator). A TRT system locates between IPv6-only hosts and IPv4 hosts and translates TCP,UDP/IPv6 to TCP,UDP/IPv4, vice versa.

Advantages of TRT are as follows:

- TRT is designed to require no extra modification on IPv6-only initiating hosts, nor that

on IPv4-only destination hosts. Some other translation mechanisms need extra modifications on IPv6-only initiating hosts, limiting possibility of deployment.

- The IPv6-to-IPv4 header converters have to take care of path MTU and fragmentation issues. However, TRT is free from this problem.

Disadvantages of TRT are as follows:

- TRT supports bidirectional traffic only. The IPv6-to-IPv4 header converters may be able to support other cases, such as unidirectional multicast datagrams.
- TRT needs a stateful TRT system between the communicating peers, just like NAT systems. While it is possible to place multiple TRT systems in a site (see Appendix A), a transport layer connection goes through particular, a single TRT system. The TRT system thus can be considered a single point of failure, again like NAT systems. Some other mechanisms, such as SIIT [111], use stateless translator systems which can avoid a single point of failure.
- Special code is necessary to relay NAT-unfriendly protocols. Some of NAT-unfriendly protocols, including IPsec, cannot be used across TRT system.

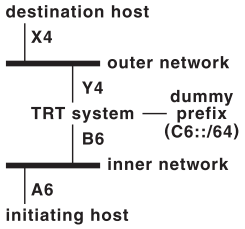
This memo assumes that traffic is initiated by an IPv6-only host destined to an IPv4-only host. The memo can be extended to handle opposite direction, if an appropriate address mapping mechanism is introduced.

### 4.3 IPv4-to-IPv4 transport relay

To help understanding of the proposal in the next section, here we describe the transport relay in general. The transport relay technique itself is not new, as it has been used in many of firewall-related products.

#### 4.3.1 TCP relay

TCP relay systems have been used in firewall-related products. These products are designed to achieve the following goals: (1) disallow forwarding of IP packets across a system, and (2) allow TCP,UDP traffic to go through the system indirectly. For example, consider a network constructed like the following diagram. "TCP relay system" in the diagram does not forward IP packet across the inner network to the outer network, vice versa. It only relays TCP traffic on a specific port, from the inner network to the outer network, vice versa. (Note: The diagram has only two subnets, one for inner and one for outer. Actually both sides can be more complex, and there can be as many subnets and routers as you wish.)




When the initiating host (whose IP address is A) tries to make a TCP connection to the destination host (X), TCP packets are routed toward the TCP relay system based on routing decision. The TCP relay system receives and accepts the packets, even though the TCP relay system does not own the destination IP address (X). The TCP relay system pretends to having IP address X, and establishes TCP connection with the initiating host as X. The TCP relay system then makes a another TCP connection from Y to X, and relays traffic from A to X, and the other way around.

Thus, two TCP connections are established in the picture: from A to B (as X), and from Y to X, like below:

```
TCP/IPv4: the initiating host (A)
    --> the TCP relay system (as X)
    address on IPv4 header: A -> X
TCP/IPv4: the TCP relay system (Y)
    --> the destination host (X)
    address on IPv4 header: Y -> X
```

The TCP relay system needs to capture some of TCP packets that is not destined to its address. The way to do it is implementation dependent and outside the scope of this memo.

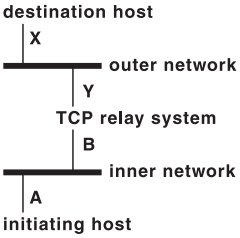
### 4.3.2 UDP relay

If you can recognize UDP inbound and outbound traffic pair in some way, UDP relay can be implemented in similar manner as TCP relay. An implementation can recognize UDP traffic pair like NAT systems does, by recording address/port pairs onto an table and managing table entries with timeouts.

## 4.4 IPv6-to-IPv4 transport relay translator

We propose a transport relay translator for IPv6-to-IPv4 protocol translation, TRT. In the following description, TRT for TCP is described. TRT for UDP can be implemented in similar manner.

For address mapping, we reserve an IPv6 prefix referred to by C6::/64. C6::/64 should be a part of IPv6 unicast address space assigned to the site. Routing information must be configured so that packets to C6::/64 are routed toward the TRT system. The following diagram shows the network configuration. The subnet marked as "dummy prefix" does not actually exist. Also, now we assume that the initiating host to be IPv6-only, and the destination host to be IPv4-only.




When the initiating host (whose IPv6 address is A6) wishes to make a connection to the destination host (whose IPv4 address is X4), it needs to make an TCP/IPv6 connection toward C6::X4. For example, if C6::/64 equals to fec0:0:0:1::/64, and X4 equals to 10.1.1.1, the destination address to be used is fec0:0:0:1::10.1.1.1. The packet is routed toward the TRT system, and is captured by it. The TRT system accepts the TCP/IPv6 connection between A6 and C6::X4, and communicate with the initiating host, using TCP/IPv6. Then, the TRT system investigates the lowermost 32bit of the destination address (IPv6 address C6::X4) to get the real IPv4 destination (IPv4 address X4). It makes an TCP/IPv4 connection from Y4 to X4, and forward traffic across the two TCP connections.

There are two TCP connections. One is TCP/IPv6 and another is TCP/IPv4, in the picture: from A6 to B6 (as C6::X4), and Y4 to X4, like below:

```
TCP/IPv6: the initiating host (A6)
 --> the TRT system (as C6::X4)
 address on IPv6 header: A6 -> C6::X4
TCP/IPv4: the TRT system (Y4)
 --> the destination host (X4)
    address on IPv4 header: Y4 -> X4
```

## 4.5 Address mapping

As seen in the previous section, an initiating host must use a special form of IPv6 address to connect to an IPv4 destination host. The special form can be resolved from a hostname by static address mapping table on the initiating host (like /etc/hosts in UNIX), special DNS server implementation, or modified DNS resolver implementation on initiating host.

## 4.6 Notes to implementers

TRT for UDP must take care of path MTU issues on the UDP/IPv6 side. The good thing is that, as we do not relay IP layer packets between IPv4 and IPv6, we can decide IPv6 path MTU independently from IPv4 traffic. A simple solution would be to always fragment packets from the TRT system to UDP/IPv6 side to IPv6 minimum MTU (1280 octets), to eliminate the need for IPv6 path MTU discovery.

Though the TRT system only relays TCP,UDP traffic, it needs to check ICMPv6 packets destined to C6::X4 as well, so that it can recognize path MTU discovery messages and other notifications between A6 and C6::X4.

When forwarding TCP traffic, a TRT system

needs to handle urgent data [116] carefully.

To relay NAT-unfriendly protocols [58] a TRT system may need to modify data content, just like any translators which modifies the IP addresses.

Scalability issues must carefully be considered when you deploy TRT systems to a large IPv6 site. Scalability parameters would be (1) number of connections the operating system kernel can accept, (2) number of connections a userland process can forward (equals to number of filehandles per process), and (3) number of transport relaying processes on a TRT system. Design decision must be made to use proper number of userland processes to support proper number of connections.

To make TRT for TCP more scalable in a large site, it is possible to have multiple TRT systems in a site. This can be done by taking the following steps: (1) configure multiple TRT systems, (2) configure different dummy prefix to them, (3) and let the initiating host pick a dummy prefix randomly for load-balancing. (3) can be implemented as follows; If you install special DNS server to the site, you may (3a) configure DNS servers differently to return different dummy prefixes and tell initiating hosts of different DNS servers. Or you can (3b) let DNS server pick a dummy prefix randomly for load- balancing. The load-balancing is possible because you will not be changing destination address (hence the TRT system), once a TCP connection is established.

For address mapping, the authors recommend use of a special DNS server for large-scale installation, and static mapping for small- scale installation. It is not always possible to have special resolver on the initiating host, and assuming it would cause deployment problems.

### 4.7 Applicability statement

Combined with a special DNS server implementation (which translates IPv4 addresses into IPv6), TRT systems support IPv6-to-IPv4 translation very well. It requires no change to existing IPv6 clients, nor IPv4 servers, so the TRT system can be installed very easily to existing IPv6-capable networks.

IPv4-to-IPv6 translation is much harder to support with any of the translator techniques [156]. While it is possible to use TRT system for IPv4-to-IPv6 translation, it requires nontrivial mapping between DNS names to temporary IPv4 addresses, as presented in NAT-PT RFC [138].

As presented in the earlier sections, TRT systems use transport layer (TCP/UDP) relay technique to translate IPv6 traffic to IPv4 traffic. It gives two major benefits: (1) the implementation of the TRT system can be done very simple, (2) with the TRT system path MTU discovery issue is easier to deal with, as we can decide IPv6 path MTU independently from IPv4 path MTU. Even with the simplicity, the TRT system can cover most of the daily applications (HTTP, SMTP, SSH, and many other protocols). For NAT-unfriendly protocols, a TRT system may need to modify data content, just like any translators/NATs. As the TRT system reside in transport layer, it is not possible for the TRT system to translate protocols that are not known to the TRT system.

Normally users do not want to translate DNS query/reply traffic using the TRT system. Instead, it makes more sense to run standard DNS server, or special DNS server that helps TRT system, somewhere in the site IPv6 network. There are two reasons to it:

- Transport issue - It is a lot easier to provide recursive DNS server, accessible via IPv6, than to translate DNS queries/replies across the TRT system. If someone tries to ask TRT to translate DNS packets, the person would put C6::X (where C6 is TRT reserved prefix and X is an IPv4 address of a DNS server) into /etc/resolv.conf. The configuration is rather complicated than we normally want.
- Payload issue - In some installation it makes more sense to transmit queries/replies unmodified, across the TRT system. In some installation it makes more sense to translate IPv4 DNS queries (like queries for AAAA

record) into queries for A record, and vice versa, to invite traffic into the TRT system. It depends on the installation/configuration at the user's site.

### 4.8 Security Considerations

Malicious party may try to use TRT systems akin to an SMTP open relay [90] for traffic to IPv4 destinations, which is similar to circumventing ingress filtering [47] , or to achieve some other improper use. TRT systems should implement some sorts of access control to prevent such improper usage.

A careless TRT implementation may be subject to buffer overflow attack, but this kind of issue is implementation dependent and outside the scope of this memo.

Due to the nature of TCP/UDP relaying service, it is not recommended to use TRT for protocols that use authentication based on source IP address (i.e., rsh/rlogin).

A transport relay system intercepts TCP connection between two nodes. This may not be a legitimate behavior for an IP node. The document does not try to claim it to be legitimate.

IPsec cannot be used across a relay.

Use of DNS proxies that modify the RRs will make it impossible for the resolver to verify DNSsec signatures.

## 第 5 章 RFC 3146: Transmission of IPv6 Packets over IEEE 1394 Networks

### 5.1 Abstract

This document describes the frame format for transmission of IPv6 packets and the method of forming IPv6 link-local addresses and statelessly autoconfigured addresses on IEEE1394 networks.

### 5.2 INTRODUCTION

IEEE Std 1394-1995 (and its amendment) is a standard for a High Performance Serial Bus. IETF IP1394 Working Group has standardized the method to carry IPv4 datagrams and ARP packets over IEEE1394 subnetwork [77].

This document describes the frame format for transmission of IPv6 [40] packets and the method of forming IPv6 link-local addresses and statelessly autoconfigured addresses on IEEE1394 networks. It also describes the content of the Source/Target Link-layer Address option used in Neighbor Discovery [109] when the messages are transmitted on an IEEE1394 network.

### 5.3 IPv6-CAPABLE NODES

An IPv6-capable node MUST fulfill the following minimum requirements:

- it MUST implement configuration ROM in the general format specified by ISO/IEC 13213:1994 and MUST implement the bus information block specified by IEEE Std 1394a-2000 [67] and a unit directory specified by this document;
- the max_rec field in its bus information block MUST be at least 8; this indicates an ability to accept block write requests and asynchronous stream packets with data payload of 512 octets. The same ability MUST also apply to read requests; that is, the node MUST be able to transmit a block response packet with a data payload of 512 octets;
- it MUST be isochronous resource manager capable, as specified by IEEE Std 1394a-2000;
- it MUST support both reception and transmission of asynchronous streams as specified by IEEE Std 1394a-2000.

### 5.4 LINK ENCAPSULATION AND FRAGMENTATION

The encapsulation and fragmentation mechanism MUST be the same as "4. LINK ENCAPSULATION AND FRAGMENTATION" of [77].

Note: Since there is an ether_type field to discriminate protocols and MCAP (multicast channel allocation protocol) is used for both IPv4 and IPv6, the version field in GASP (global asyn-

chronous stream packet) header of IPv6 datagrams is the same value (one) as [77].

The ether_type value for IPv6 is 0x86dd.

The default MTU size for IPv6 packets on an IEEE1394 network is 1500 octets. This size may be reduced by a Router Advertisement [109] containing an MTU option which specifies a smaller MTU, or by manual configuration of each node. If a Router Advertisement received on an IEEE1394 interface has an MTU option specifying an MTU larger than 1500, or larger than a manually configured value, that MTU option may be logged to system management but MUST be otherwise ignored. The mechanism to extend MTU size between particular two nodes is for further study.

### 5.5 CONFIGURATION ROM

Configuration ROM for IPv6-capable nodes MUST contain a unit directory in the format specified by [77] except following rules.

- The value for Unit_SW_Version is 0x000002.
- The textual descriptor for the Unit_SW_Version MUST be "IPv6".

Note: A dual-stack (IPv4 and IPv6) node will have two unit directories for IPv4 and IPv6 respectively.

### 5.6 STATELESS AUTOCONFIGURATION

The Interface Identifier [64] for an IEEE1394 interface is formed from the interface's built-in EUI-64 identifier by complementing the "Universal/Local" (U/L) bit, which is the next-to-lowest order bit of the first octet of the EUI-64 identifier. Complementing this bit will generally change a 0 value to a 1, since an interface's built-in EUI-64 identifier is expected to be from a universally administered address space and hence have a globally unique value. A universally administered EUI-64 identifier is signified by a 0 in the U/L bit position, while a globally unique IPv6 Interface Identifier is signified by a 1 in the corresponding position. For further discussion on this point, see [64].

An IPv6 address prefix used for stateless autoconfiguration [136] of an IEEE1394 interface MUST have a length of 64 bits.

### 5.7 LINK-LOCAL ADDRESSES

The IPv6 link-local address [64] for an IEEE1394 interface is formed by appending the Interface Identifier, as defined above, to the prefix FE80::/64.

| 10 bits | 54 bits | 64 bits |
|---|---|---|
| 1111111010 | (zeros) | Interface Identifier |

### 5.8 ADDRESS MAPPING FOR UNICAST

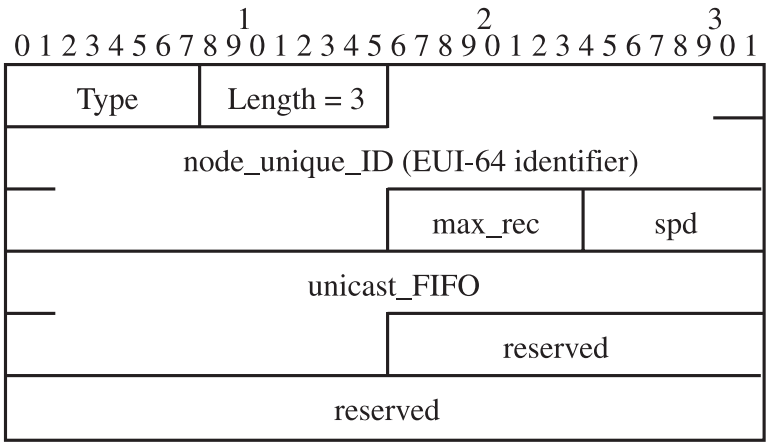
The procedure for mapping IPv6 unicast addresses into IEEE1394 link- layer addresses uses the Neighbor Discovery [109]. Since 1394 link address (node_ID) will not be constant across a 1394 bridge, we have chosen not to put it in the Link-layer Address option. The recipient of the Neighbor Discovery SHOULD use the source_ID (obtained from either the asynchronous packet header or the GASP header) in conjunction with the content of the Source link-layer address. An implementation MAY use some other methods to obtain a node_ID of the sender utilizing a mapping table between node_unique_ID (EUI-64 identifier) and node_ID. The mechanism to make such mapping table is out of scope of this document.

The recipient of an Neighbor Discovery packet MUST ignore it unless the most significant ten bits of the source_ID are equal to either 0x3FF or the most significant ten bits of the recipient's NODE_IDS register.

The Source/Target Link-layer Address option has the following form when the link layer is IEEE1394.

Note that node_ID may change when 1394 bus-reset occurs. The mapping cache held in the node SHOULD be cleared on 1394 bus-reset.

According to [66], the maximum data payload and the transmission speed SHOULD be determined based on the sender's capability, the recipient's capability, and the PHYs of all intervening nodes.




| | |
|---|---|
| Type | 1 for Source Link-layer address.<br>2 for Target Link-layer address. |
| Length | 3 (in units of 8 octets). |
| node_unique_ID | This field contains the node unique ID of the node and MUST be equal to that specified in the node's configuration ROM. |
| max_rec | This field MUST be equal to the value of max_rec in the node's configuration ROM. |
| spd | This field MUST be set to the lesser of the node's link speed and PHY speed. The link speed is the maximum speed at which the link may send or receive packets; the PHY speed is the maximum speed at which the PHY may send, receive or repeat packets. The encoding used for spd is specified in the Table 2 of rfc2734. |
| unicast_FIFO | This field MUST specify the 48-bit offset of the node's FIFO available for the receipt of IPv6 datagrams. The offset of a node's unicast FIFO MUST NOT change, except as the result of a power reset. |
| reserved | This field MUST be set to all zeros by the sender and ignored by the receiver. |

### 5.9 IPv6 MULTICAST

By default, all best-effort IPv6 multicast MUST use asynchronous stream packets whose channel number is equal to the channel field from the BROADCAST_CHANNEL register. In particular, datagrams addressed to all-nodes multicast addresses, all-routers multicast addresses, and solicited-node multicast addresses [64] MUST use the default channel specified by the BROADCAST_CHANNEL register.

Best-effort IPv6 multicast for other multicast group addresses may utilize a different channel number if such a channel number is allocated and advertised prior to use, by the multicast channel allocation protocol (MCAP), as described in [77].

When a node wishes to receive multicast data addressed to other than all-nodes multicast addresses, all-routers multicast addresses, and solicited-node multicast addresses, it MUST confirm if the channel mapping between a multicast group address and a channel number exists using MCAP, as described in "9.3 Multicast Receive" in [77].

The implementation of MCAP is optional for send-only nodes. A node MAY transmit multicast data addressed to any multicast addresses into the default broadcast channel regardless of the existing allocation of the channel. If a node wishes to transmit multicast data on other than the default channel, it MUST first confirm by MCAP whether or not a channel number for the group address has been already allocated. The implementors are encouraged to use this protocol when transmitting

high-rate multicast streams.

The MCAP 'type' value for IPv6 group address descriptor is 2.

### 5.10 IANA CONSIDERATIONS

IANA has assigned a value of 0x000002 for "Unit_SW_Version for IPv6 over IEEE1394" out of the "CSR Protocol Identifiers" name space, as described in section 5. The details of the "CSR Protocol Identifiers" namespace is described in "10. IANA CONSIDERATIONS" of [77].

Section 9.1 of [77] defines MCAP group address descriptors, which include an 8-bit type name space. This document requests that IANA maintain a name space to manage MCAP group address descriptors. The initial assignments for that table are:

| Value | Usage |
|---|---|
| 0 | reserved |
| 1 | IPv4 Multicast Address |
| 2 | IPv6 Multicast Address |
| 255 | reserved |

Additional values from the range 3-254 can be assigned through Standards Action [108].

### 5.11 Security Considerations

IPv6 over IEEE1394 does not introduce any additional security considerations over [77]. The security concerns described in "11. SECURITY CONSIDERATIONS" in [77] apply here as well.

## 第 6 章　RFC 3178: IPv6 Multihoming Support at Site Exit Routers

### 6.1 Abstract

The document describes a mechanism for basic IPv6 multihoming support, and its operational requirements. Unlike currently- practiced IPv4 multihoming, the technique does not impact the worldwide routing table size, nor IGP (Interior Gateway Protocol) routing table size in upstream ISPs. The mechanism can be combined with more sophisticated (or complex) multihoming support mechanisms, and can be used as a foundation for other mechanisms. The document is largely based on RFC 2260 by Tony Bates.

### 6.2 Problem

Routing table size has been a major issue for both IPv4 and IPv6. As IPv6 addresses are 4 times larger in bit width than IPv4, the routing table size issue would have more serious negative effects on router memory usage, as well as routing table lookup performance. To cope with this problem, the IPv6 addressing architecture [64] is designed to take advantage of aggregated routing announcements to reduce the number of routes in default-free zone. Also, 6bone operation guideline [126] (which is the currently-practiced guideline for IPv6 network operation) suggests that ASes not announce non-aggregatable announcements to the default-free zone, if there is no special agreement with the peer.

In IPv4, a multihomed site uses either of the following techniques to achieve better reachability:

- Obtain a portable IPv4 address prefix, and announce it from multiple upstream providers.
- Obtain a single IPv4 address prefix from ISP A, and announce it from multiple upstream providers the site is connected to.

Since the above two methodologies effectively inject additional routes to the worldwide routing table, they have negative impact on the worldwide routing table size issue. They also are not compatible with current IPv6 operational practice.

This document provides a way to configure site exit routers and ISP routers, so that the site can achieve better reachability from multihomed connectivity, without impacting worldwide routing table size issues. The technique uses multiple distinct IPv6 address prefixes, assigned from multiple upstream ISPs. The technique uses an already-defined routing protocol (BGP or RIPng) and tunneling of IPv6 packets; therefore, this document

introduces no new protocol standard (the document describes how to operate the configuration).

This document is largely based on RFC 2260 [17] by Tony Bates.

### 6.3 Goals and non-goals

The goal of this document is to achieve better packet delivery from a site to the outside, or from the outside to the site, even when some of the site exit links are down.

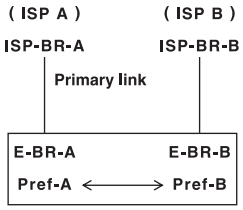
Non goals are:

- Choose the "best" exit link as possible. Note that there can be no common definition of the "best" exit link.
- Achieve load-balancing between multiple exit links.
- Cope with breakage of any of the upstream ISPs.

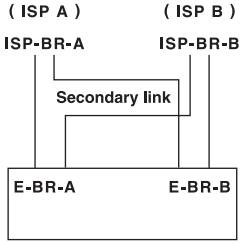
### 6.4 Basic mechanisms

We use the technique described in RFC 2260 section 5.2 in our configuration. To summarize, for IPv4-only networks, RFC 2260 says that:

- We assume that our site is connected to 2 ISPs, ISP-A and ISP-B.
- We are assigned IP address prefixes, Pref-A and Pref-B, from ISP-A and ISP-B respectively. Hosts near ISP-A will get an address from Pref-A, and vice versa.
- In the site, we locally exchange routes for Pref-A and Pref-B, so that hosts in the site can communicate with each other without using external link.
- ISP-A and our site are connected by a "primary link" between ISP router ISP-BR-A and our router E-BR-A. ISP B and our site are connected by a primary link between ISP router ISP-BR-B and our router E-BR-B.




- Establish a secondary link, between ISP-BR-A and E-BR-B, and ISP- BR-B and E-BR-A, respectively. The secondary link usually is an IP-over-IP tunnel. It is important to have the secondary link on top of a different medium than the primary link, so that one of them survives link failure. For example, the secondary link between ISP-BR-A and E-BR-B should go through a different medium than the primary link between ISP-BR-A and E-BR-A. If the secondary link is an IPv4-over-IPv4 tunnel, the tunnel endpoint at E-BR-A needs to be an address in Pref-A, not in Pref-B (tunneled packet needs to travel from ISP-BR-B to E-BR-A, over the primary link between ISP-BR-A and E-BR-A).




- For inbound packets, E-BR-A will advertise (1) Pref-A toward ISP- BR-A with strong preference the over primary link, and (2) Pref-B toward ISP-BR-B with weak preference over the secondary link. Similarly, E-BR-B will advertise (1) Pref-B toward ISP-BR-B with strong preference over the primary link, and (2) Pref-A toward ISP-BR-A with weak preference over the secondary link.

  Note that we always announce Pref-A to ISP-BR-A, and Pref-B to ISP-BR-B.
- For outbound packets, ISP-BR-A will advertise (1) default route (or specific routes) toward E-BR-A with strong preference over the

primary link, and (2) default route (or specific routes) toward E-BR-B with weak preference over the secondary link. Similarly, ISP-BR-B will advertise (1) default route (or specific routes) toward E-BR-B with strong preference over the primary link, and default route (or specific routes) toward E-BR-A with weak preference over the secondary link.

Under this configuration, both inbound and outbound packets can survive link failure on either side. Routing information with weak preference will be available as backup, for both inbound and outbound cases.

## 6.5 Extensions for IPv6

RFC 2260 is written for IPv4 and BGP. With IPv6 and BGP4+, or IPv6 and RIPng, similar results can be achieved, without impacting worldwide IPv6 routing table size.

### 6.5.1 IPv6 rule conformance

In RFC 2260, we announce Pref-A toward ISP-BR-A only, and Pref-B toward ISP-BR-B only. Therefore, there will be no extra routing announcement to the outside of the site. This meets the suggestions in 6bone aggregation guidelines [126]. Also, RFC 2260 does not require portable addresses.

### 6.5.2 Address assignment to the nodes

In IPv4, it is usually assumed that a node will be assigned a single IPv4 address. Therefore, RFC 2260 assumed that addresses from Pref-A will be assigned to nodes near E-BR-A, and vice versa (second bullet in the previous section).

With IPv6, multiple IPv6 addresses can be assigned to a node. So we can assign (1) one address from Pref-A, (2) one address from Pref-B, or (3) addresses from both prefixes, to a single node in the site. This will allow more flexibility in node configuration.

When multiple IPv6 global addresses are assigned to an IPv6 node, source address selection must take place on packet transmissions. Source address selection itself is out of scope of the document. Refer to a separate draft [43] for more discussions.

One simplifying approach is to place the site's Internet hosts on separate subnets, one with addresses in Pref-A and connected to E- BR-A, the other having addresses in Pref-B and connected to E-BR-B. This approach generalizes to having E-BR-A and E-BR-B at different sites, where site A and site B have links to the Internet and to each other.

### 6.5.3 Configuration of links

With IPv6, the primary link can be IPv6 native connectivity, RFC 2893 [119] IPv6-over-IPv4 configured tunnel, 6to4 [26] IPv6-over-IPv4 encapsulation, or some others.

If tunnel-based connectivity is used in some of primary links, administrators may want to avoid IPv6-over-IPv6 tunnels for secondary links. For example, if:

- primary links to ISP-A and ISP-B are RFC 2893 IPv6-over-IPv4 tunnels, and
- ISP-A, ISP-B and the site have IPv4 connectivity with each other.

It makes no sense to configure a secondary link by IPv6-over-IPv6 tunnel, since it will actually be IPv6-over-IPv6-over-IPv4 tunnel. In this case, IPv6-over-IPv4 tunnel should be used for secondary link. IPv6-over-IPv4 configuration has a big advantage against IPv6-over-IPv6-over-IPv4 configuration, as secondary link will be able to have the same path MTU than the primary link.

In the figure, ISP-BR-A and E-BR-A are both single points of failure for inbound traffic to Pref-A. This could be remedied by using different routers for primary vs. backup links.

### 6.5.4 Using RFC 2260 with IPv6 and BGP4+

The RFC 2260 approach on top of IPv6 will work fine as documented in RFC 2260. There

will be no extra twists necessary. Since the multihomed site is not doing transit, variations are possible that do not require it to have a public AS number.

### 6.5.5 Using RFC 2260 with IPv6 and RIPng

It is possible to run an RFC 2260-like configuration with RIPng [94] , with careful control of metric. Routers in the figure need to increase RIPng metric on the secondary link, to make the primary link a preferred path.

If we denote the RIPng metric for route announcement, from router R1 toward router R2, as metric(R1, R2), the invariants that must hold are:

- metric(E-BR-A, ISP-BR-A) < metric(E-BR-B, ISP-BR-A)
- metric(E-BR-B, ISP-BR-B) < metric(E-BR-A, ISP-BR-B)
- metric(ISP-BR-A, E-BR-A) < metric(ISP-BR-A, E-BR-B)
- metric(ISP-BR-B, E-BR-B) < metric(ISP-BR-B, E-BR-A)

Note that smaller metric means stronger route in RIPng.

## 6.6 Issues with ingress filters in ISP

If the upstream ISP imposes ingress filters [47] to outbound traffic, the story becomes much more complex. A packet with source address taken from Pref-A must go out from ISP-BR-A. Similarly, a packet with source address taken from Pref-B must go out from ISP-BR-B. Since none of the routers in the site network will route packets based on source address, packets can easily be routed to incorrect border router.

One possible way is to negotiate with both ISPs, to allow both Pref-B and Pref-A to be used as source address. This approach does not work if upstream ISP of ISP-A imposes ingress filtering. Since there will be multiple levels of ISP on top of ISP-A, it will be hard to understand which upstream ISP imposes the filter. In reality, this problem will be very rare, as ingress filter is not suitable for use in large ISPs where smaller ISPs are connected beneath.

Another possibility is to use source-based routing at E-BR-A and E- BR-B. Here we assume that IPv6-over-IPv6 tunnel is used for secondary links. When an outbound packet arrives to E-BR-A with source address in Pref-B, E-BR-A will forward it to the secondary link (tunnel to ISP-BR-B) based on source-based routing decision. The packet will look like this:

- Outer IPv6 header: source = address of E-BR-A in Pref-A, dest = ISP-BR-B
- Inner IPv6 header: source = address in Pref-B, dest = final dest

  A tunneled packet will travel across ISP-BR-A toward ISP-BR-B. The packet can go through ingress filter at ISP-BR-A, since it has outer IPv6 source address in Pref-A. The packet will reach ISP-BR-B and be decapsulated before ingress filter is applied. Decapsulated packet can go through ingress filter at ISP-BR-B, since it now has source address in Pref-B (from inner IPv6 header). Notice the following facts when configuring this:
- Not every router implements source-based routing.
- The interaction between normal routing and source-based routing at E-BR-A (and/or E-BR-B) varies by router implementations.
- At ISP-BR-B (and/or ISP-BR-A), the interaction between tunnel egress processing and filtering rules varies by router implementations and filter configurations.

## 6.7 Observations

The document discussed the cases where a site has two upstream ISPs. The document can easily be extended to the cases where there are 3 or more upstream ISPs.

If you have many upstream providers, you would not make all ISPs backup each other, as it requires $O(N^2)$ tunnels for N ISPs. Rather, it is better to make N/2 pairs of ISPs, and let each pair of ISPs backup each other. It is important to pick pairs

which are unlikely to be down simultaneously. In this way, number of tunnels will be $O(N)$.

Suppose that the site is very large and it has ISP links in very distant locations, such as in the United States and in Japan. In such a case, it is wiser to use this technique only among ISP links in the US, and only among ISP links in Japan. If you use this technique between ISP link A in the US and ISP link B in Japan, the secondary link makes packets travel a very long path, for example, from a host in the site in the US, to E-BR-B in Japan, to ISP-BR-B (again in Japan), and then to the final destination in the US. This may not make sense for actual use, due to excessive delay.

Similarly, in a large site, addresses must be assigned to end nodes with great care, to minimize delays due to extra path packets may travel. It may be wiser to avoid assigning an address in a prefix assigned from Japanese ISP, to an end node in the US.

If one of the primary links is down for a long time, administrators may want to control source address selection on end hosts so that secondary link is less likely to be used. This can be achieved by marking the unwanted prefix as deprecated. Suppose the primary link toward ISP-A has been down. You will issue router advertisement [109][136] packets from routers, with preferred lifetime set to 0 in prefix information option for Pref-A. End hosts will consider addresses in Pref-A as deprecated, and will not use any of them as source address for future connections. If an end host in the site makes a new connection to outside, the host will use an address in Pref-B as source address, and the reply packet to the end host will travel the primary link from ISP-BR-B toward E-BR-B. A great care must be taken when you try to automate this by using router renumbering protocols [34] , as the approach could lead your site into very unstable state if any of the links flap. The author does not recommend to automate it.

Some of non-goals (such as "best" exit link selection) can be achieved by combining the technique described in this document, with some other techniques. One example of the technique would be the source/destination address selection [43] on the end nodes.

### 6.8 Operational experiences

Hal Snyder has been running the technique, with two upstream ISPs (lava.net and iijlab), using 2 RFC 2893 IPv6-over-IPv4 tunnels to each of them (in total 4 tunnels), and BGP4+ peering over them.

As expected, when the primary links goes down the routing switches to the secondary link within BGP hold time, i.e., we see approximately the relations:

- (hold time - keepalive time) < failover time
- failover time < hold time
- failback time < keepalive time

This has been tested with keepalive and hold times from as low as 3 and 10 seconds respectively, up to 60 and 180 seconds respectively.

The routing change will affect ISP-BR-A (or B) only. Because route instability is not propagated beyond one ISP, it should be feasible to use lower hold and keepalive times than in a conventional IPv4 setting. If primary and backup links terminate on the same router at the ISP, then failover from primary to backup link need not affect reachability information upstream of that router.

Many of the existing IPv6 networks (connected to worldwide 6bone) are assigned multiple IPv6 prefixes from multiple upstreams. In many cases people assign global IPv6 addresses generated from multiple address prefixes. There has been almost no problems raised about complication due to source address selection.

### 6.9 Security Considerations

The configuration described in the document introduces no new security problem.

If primary links toward ISP-A and ISP-B have different security characteristics (like encrypted link and non-encrypted link), administrators need to be careful setting up secondary links tunneled

on them. Packets may travel an unwanted path, if secondary links are configured without care.

## 第7章 ドラフト

この章では、我々が執筆したドラフトの要約を掲載する。

### 7.1 Comparison of AAAA and A6 (do we really need A6?)

draft-ietf-dnsext-aaaa-a6-01.txt

At this moment, there are two DNS resource record types defined for holding IPv6 address in the DNS database; AAAA [135] and A6 [35]. AAAA has been used for IPv6 network operation since 1996. Questions arose whether we really need A6 or not, or whether it is really possible to migrate to A6 or not. Some says AAAA is enough and A6 is not necessary. Some says A6 is necessary and AAAA should get deprecated.

The draft tries to understand pros and cons between these two record types, and makes suggestions on deployment of IPv6 record type.

The draft does not cover the use of bit string label and DNAME resource record (reverse mapping), as it seems that nibble form is well accepted in the community, newer formats have too much deployment costs, thus we see few need/voice that calls for migration. Refer to IETF50 dnsext working group minutes for more details.

### 7.2 Analysis of DNS Server Discovery Mechanisms for IPv6

draft-ietf-ipngwg-dns-discovery-analysis-00.txt

There are any number of ways that IPv6 hosts can discover information required to enable name resolution, in the absence of a DHCP server. This document discusses the issues and provides a taxonomy of possible solutions, and evaluates them against various design criteria. Finally, it provides recommendations as input to the standards process.

### 7.3 Avoiding ping-pong packets on point-to-point links

draft-ietf-ipngwg-p2p-pingpong-00.txt

In IPv6 point-to-point link operation, there is a significant possibility of aberrant behavior in that packets may ping-pong between the two ends of the link. The problem can lead to wasted bandwidth and can possibly be abused by malicious parties. This document provides an analysis and solution to the problem.

### 7.4 Advanced Sockets API for IPv6

draft-ietf-ipngwg-rfc2292bis-06.txt

A separate specification [53] contain changes to the sockets API to support IP version 6. Those changes are for TCP and UDP-based applications and will support most end-user applications in use today: Telnet and FTP clients and servers, HTTP clients and servers, and the like.

But another class of applications exists that will also be run under IPv6. We call these “advanced” applications and today this includes programs such as Ping, Traceroute, routing daemons, multicast routing daemons, router discovery daemons, and the like. The API feature typically used by these programs that make them “advanced” is a raw socket to access ICMPv4, IGMPv4, or IPv4, along with some knowledge of the packet header formats used by these pro-

tocols. To provide portability for applications that use raw sockets under IPv6, some standardization is needed for the advanced API features.

There are other features of IPv6 that some applications will need to access: interface identification (specifying the outgoing interface and determining the incoming interface) and IPv6 extension headers that are not addressed in [53]: The Routing header (source routing), Hop-by-Hop options, and Destination options. This document provides API access to these features too.

### 7.5 IPv6 Scoped Address Architecture

draft-ietf-ipngwg-scoping-arch-03.txt

This document specifies the architectural characteristics, expected behavior, textual representation, and usage of IPv6 addresses of different scopes.

### 7.6 IPv6 Stateless DNS Discovery

draft-ietf-ipv6-dns-discovery-04.txt

This document specifies the steps a host takes in deciding how to autoconfigure the addresses of DNS Servers required for name resolution in IP version 6. The autoconfiguration process includes determining whether such information should be obtained through the stateless mechanism, the stateful mechanism, or both. This document defines the process for acquiring a list of DNS server addresses. Approaches for acquiring a domain search path, and the domain name of the host via a stateless mechanism are included in an appendix for further study. The details of autoconfiguration using the stateful protocol are specified elsewhere.

### 7.7 An overview of the introduction of IPv6 in the Internet

draft-ietf-ngtrans-introduction-to-ipv6-transition-08.txt

This document is a guide to the introduction of IPv6 in the IPv4 based Internet or Intranets. Several general issues to start IPv6 networking in a predominantly IPv4 world are discussed, such as IPv6 addresses, IPv6 DNS and routing issues. Short descriptions are given of the different transition tools and mechanisms that translate between IPv6 and IPv4 and/or tunnel IPv6 over IPv4. The remainder of this document describes how IPv6 can be introduced in various environments, such as ISPs and end user environments. Suggestions are given on the use of the different translation and migration tools in each environment.

### 7.8 Requirements for IPv6 dialup operation

draft-itojun-ipv6-dialup-requirement-02.txt

The memo tries to identify design choices in IPv6 dialup services by ISPs. We also supply a couple of scenarios as design prototypes for ISP IPv6 dialup services.

### 7.9 Socket API for IPv6 flow label field

draft-itojun-ipv6-flowlabel-api-01.txt

The draft outlines a socket API proposal for controlling the flow label field in the IPv6 header. The API uses the sin6_flowinfo member on the IPv6 socket address structure (sockaddr_in6).

The draft is, at this moment, written separately from the IPv6 basic/advanced API RFCs [53][132], as there can be many discussion items. The ultimate goal of the draft is to be a part of the IPv6 basic/advanced API.

### 7.10 Disconnecting TCP connection toward IPv6 anycast address

draft-itojun-ipv6-tcp-to-anycast-01.txt

IPv6 specification implicitly disallows TCP connection toward IPv6 anycast address. However, if such a connection request happens by mistake, currently there is no way to report the incident to the originator of the TCP connection. The document tries to define a way to disconnect TCP connections made toward IPv6 anycast addresses.

## 第 8 章　普及活動

この章では、普及活動の一環として、NetWorld+Interop 2001 IPv6 ShowCase と Global IPv6 Summit in Japan について報告する。

### 8.1 NetWorld+Interop 2001 IPv6 ShowCase

2001 年 6 月 6 日〜6 月 8 日に幕張メッセで開催された NetWorld+Interop 2001 の中で、昨年に引き続き IPv6 ShowCase という IPv6 関連のブースを設置した。WIDE プロジェクトからは、インターネットカーを出展した。下記のページに載っている写真からも分かるように、大盛況だった。

`http://www.kame.net/N+I_2001/`

IPv6 ShowCase は、以下に示す 3 つの柱からなる。

1. 最新の製品による相互接続性のデモ
2. プレゼンテーション
3. ポポちゃんの部屋

#### 8.1.1 最新の製品による相互接続性のデモ

相互接続性のデモで昨年と際立って異なるのは、以下の 2 点である。

- DSL などデータリンクの種類が増えたこと、
- 経路制御プロトコルとして OSPFv3 が実用レベルに達したこと

特に、日立 GR2000、NEC IX 5000、エリクソン AXI 462、Zebra on KAME の 4 つの OSPFv3 の実装を相互につないだことは、意義が大きいと思う。

#### 8.1.2 プレゼンテーション

今年のプレゼンテーションは、マイクロソフト、NEC、日本エリクソン、日立製作所、NTT コミュニケーションズ、KDDI、シスコシステムズ、富士通というスポンサーからの発表と、IPv6 の一般に関する発表があった。どの回も立ち見で通路まで埋め尽くされるほどの盛況ぶりだった。

#### 8.1.3 ポポちゃんの部屋

「ポポちゃん」とは、熊のぬいぐるみの名前だ。今年の目玉は、ポポちゃんの部屋と称し、IPv6 を喋る家電などを集めて展示したことである。

これまで、IPv6 は絵にならないことが悩みだったが、それはもう大した問題ではなくなった。IPv6 はインターネットそのものだから、夢色のインターネットを演出すれば、それは IPv6 を表現したことに他ならない。

ここでは、ポポちゃんの部屋に関し、2 つのデモのみを報告する。

まず、IPv6 を喋る Panasonic のテレビ。チャンネルを変えるなどの操作を携帯電話からができるようにした。我々の夢である、携帯電話で制御できるビデオまで、もう少しといった感じである。

次に、マイクロノードという温度計。これはサーバであり、温度情報を発信する。会場内に配置した多数のマイクロノードから情報を集め、温度分布の図を作成するデモをやった。1 つの情報にはあまり意味がなくとも、たくさん集めると価値がある生まれる例の 1 つだ。

IPv6 ShowCase では、IPv6 の夢と現実性を強くアピールできたと信じる。

### 8.2 Global IPv6 Summit in Japan 2002

2001 年 12 月 3 日からの 2 日間、横浜パシフィコで、「Global IPv6 Summit in Japan2001」(以下 IPv6 Summit) を開催した。WIDE は後援という形で協力した。

IPv6 Summit のホームページは以下の通り。

http://www.jp.ipv6forum.com/

スポンサーは、Cisco Systems、Foundary Networks、Juniper Networks、NEC、富士通、日立製作所 (順不同) の 6 社。

今年は、ISP が正式なサービスを提供し始め、企業や家庭でも IPv6 のコネクティビティを持てるようになった。また Windows XP に (出荷時は利用しないよう設定されているが) IPv6 の機能が搭載され、一般の方にも IPv6 の敷居が低くなった。

このような背景において、IPv6 を家庭や企業に導入することや、それにより引き起こされる社会への影響に関する考察をまとめておくことが、今回の最大の目的となった。

一日目のプログラムは以下の通り。

- 実行委員長挨拶
- 基調講演：IPv6: Making the Dream Real
- 日本の IPv6 に関するビジネス・レポート (1)
- 世界各国の IPv6 に関するレポート
- IPv6 によるユーザアプリケーションとサービス

二日目のプログラム以下の通り。

- 基調講演：IPv6 インターネット社会の構築
- パネル：企業ネットと IPv6
- 日本の IPv6 に関するビジネス・レポート (2)
- パネル：コンシューマへの IPv6 導入課題を探る
- パネル：IPv6 のもたらす社会的インパクト
- 閉会の挨拶

739 名の参加者を得たことや、活発な議論が多々交わされたことから、大成功に終ったと思う。