

IPv6 対応小型ネットワークサーバの開発(第一報)

坂本佳則^{*1)}

Development of a small IPv6 network server (1st report)

SAKAMOTO Yoshinori^{*1)}

The progress to next-generation network protocol IPv6 from IPv4 is becoming clear by explosive rapid growth of the Internet. IPv6 attracts attention as a protocol dealing with a vast address space, and security and multimedia enhancement. Aim for the IPv6 world, we develop the small and cheap network server which can handle IPv6, in two years. The hardware of the server uses PC architecture as a base, and develops it there combining OS of an open source. In this paper reports, investigation of the technical trend involving an IPv6 protocol, and IPv6-ized implementation of a network of Industrial Institute as maintenance of experiment environment. This is a report about the first year of a two-year developing plan.

1 . はじめに

旧来のインターネットを支える通信プロトコルである IPv4 が物理的な限界に近づきつつある現在、次世代通信プロトコル IPv6 への移行作業が急ピッチで進められている。

そこで、急激に起きると考えられるインターネット環境の IPv6 への移行に備え、中小規模のネットワークで有効活用できる安価な IPv4/IPv6 両対応のサーバを平成 14 年度から 15 年度の 2 年間で試作開発する。

本稿では研究開発の 1 年目として平成 14 年度に行った内容を報告する。

以下、2 章で次世代インターネットプロトコル IPv6 についての概要紹介を行い、3 章で小型サーバの基本アーキテクチャ検討の内容について述べ、4 章で研究開発環境として工業技術センターネットワークの IPv6 化の紹介を行う。5 章でまとめとして、今後の展望と課題を述べる。

2 . IPv6

本開発が対象とする、次世代インターネットプロトコル IPv6(Internet Protocol version 6)とはどのようなものかを、まず紹介する。

2.1 IPv6 策定の経緯

インターネットの技術は IETF (The Internet Engineering Task Force) で検討され、標準化されている。その IETF で従来の枠組みである IPv4 のアドレス枯渇の可能性が指摘されたのは、1991 年ごろであった。ちょうど、それまで研究用途に限定して発展を続けてきたインターネットが

商用に使われるようになる時期である。1992 年春の IETF では、この問題の短期解として、CIDR (Classless Inter-Domain Routing) と Supernetting、長期解として、BiggerAddress が検討されることになった。短期解は IPv4 のアドレスを効率的に活用していく技術であり、延長上にローカル IP の活用や、NAT の開発が行われ、アドレスの節約に大いに役立った。後者が次世代通信プロトコルの検討であり、23 もの RFC(提案書)が提出され、それから 1994 年に至る長期の議論を経て、IP Version 6 が策定された。

2.2 IPv4 から IPv6 へ

32 ビットの数値で表されてきた IPv4 の規格では、もはやインターネットにつながってくる膨大な端末機器を将来にわたって区別し続けることは困難になってきた。また、通信全般のインフラとして機能し始めた今日のインターネットでは、当初想定されていた遠隔地の計算機間でのデータ通信用途のみならず、動画や音声配信といった実時間性を要する通信や、セキュリティ対策のための暗号化技術が必須となりつつある。

NAT 技術や、特定の機器間での帯域保証技術など IPv4 の上でもそれぞれの対策技術は開発され、使われてきたが、その場しのぎの拡張の繰り返しであり、本来の機能の一部を犠牲にしていたり、インターネット全体で等しく利用できる状態ではなかったりという状態となっている。

IPv4 は 1970 年代から 1980 年にかけて仕様が策定されたプロトコルであり、実際にインターネットを地球規模の大規模ネットワークとして稼働させた、優れた規格ではあったが、長年の利用で先に述べたような様々な問題点が明らかとなり、新しい IPv6 プロトコルが提案された。

問題解決のために提案された、IPv6 プロトコルは次の

*1)電子・情報・デザイン技術チーム

ような新しい特徴を有している。

- アドレス空間を 128 ビットに拡張
- ヘッダ情報が整理され単純化
- Plug & Play 機能
- セキュリティ機能
- 実時間通信サポートのための仕様

プロトコルレベルでこれらの機能が明確に制定されたことで、たとえば特定の機関同士、ソフトウェア同士といった制限なしに広くインターネットで共用できる技術として使用可能となる。

2.3 プロトコル仕様

128 ビットに拡張された IP アドレスは 16 ビットずつのグループに分け、コロンで区切って表記される。このとき 16 進表記を用いる。(IPv4 では 8bit ごとのコロン区切りの 10 進表記が一般的だった)。Table 1 に IPv4 と IPv6 のアドレス表記の比較を示す。

Table 1 Notation of an IP address

種別	アドレス表記例
IPv4	202.249.60.1
IPv6	2001:200:168:2:290:feff:fe23:96a
IPv6省略形	2001:200:168:1::3

なお、連続した 0 が続く 1 箇所限り表記上省略できる。Table 1 中に示した、「IPv6 省略形」がその表記である。途中で 0 以外の数値が入り、複数箇所でも 0 が連続しても 1 か所以上は省略できない。

IPv6 プロトコルは、そのプロトコルヘッダ部分を IPv4 より単純化するという改良が行われている。IPv4 で定義はされていたものの、長い運用の歴史の中で現実にはほとんど利用されなかったオプションが取り払われ、パケットサイズを固定長へと制定することにより、パケットを扱う中継装置の負荷が少なくすむような改善が行われた。

新たに、Plug & Play 機能として、クライアントが自らの IP アドレスを自動的に付与できるように工夫が行われている。従来であれば IP アドレスを配布管理するサーバ (DHCP サーバ) の運用が不可欠だったものを、そのクライアントがつながっているルータからの最低限の情報を利用して自動的にアドレス付与が可能となった。

セキュリティ対応としては、IPv6 端末間の認証(鍵機能付き MD5)と暗号化通信(CBS DES)の方法が標準で制定されている。プロトコル仕様の中で標準的な手法が制定されているので、ユーザが必要としたときに、これらを活用していつでも安全な通信を行うことができる。

さらに、実時間通信の機能が IP のプロトコルレベルでサポートされた意義は大きい。IPv4 では IP のレベルでの帯域の保証が難しかったため、ビデオデータなどの通信に

は非常に高速な回線を用意し、専用のアプリケーションを利用する必要があった。帯域予約機構などの手法をプロトコルレベルで制定することにより、より広範囲にネットワーク資源を共有しつつ(必要以上の広帯域ネットワークを構築することなく)、実時間通信をスムーズに行える基礎とすることができる。

2.4 動作環境

さて、IPv6 はどのような OS 上で稼働するかであるが、現在では、「ほぼすべてのプラットフォームの最新バージョン」上で、利用することが可能なまでになっている。IPv6 の OS への実装については、日本の研究グループが着手した、KAME というパッケージが高い完成度をもって、世界的に先行した。これは、BSD 系の UNIX(BSDI, NetBSD, FreeBSD, OpenBSD)への IPv6 対応化パッチとプログラム集として開発されてきている。

この KAME の実装は、今でもリファレンス的な位置づけを維持しており、数々のプラットフォームが KAME の実装を参考に IPv6 対応を行っている。

昨今、幾多の商用プラットフォームとして採用が続いている Linux については、元々 IPv6 対応コードがカーネルに入っていたが、プロトコル仕様を満足できるまでではないまま放置されていた。これも日本の研究グループが USAGI プロジェクトという産学官の共同開発プロジェクトを立ち上げ、KAME のコードを本格的に移植をした結果、現時点では遜色のない仕上がりを見せている。

その他、SUNOS や、Microsoft Windows シリーズなど主要な商用 OS についての対応状況も、実験的対応の段階を経て、現在では本格サポートに向かいつつある状況にまで達してきている。

プロバイダの対応は、まだまだ発展途上であるが、すでに複数の主要プロバイダでは、IPv6 接続が何らかの形で可能となっており、試験接続を行っているところもあれば、すでに商用サービスが始めているところもある。

ルータ機器など、IPv6 対応が必要な機器についても、安価な製品が出現してきたという状況である。

3 IPv6 対応小型ネットワークサーバ

中小規模のネットワークで有効活用できる安価な IPv4/IPv6 両対応のサーバの開発に向け、基本アーキテクチャの検討と、ユーザインタフェースソフトウェアの検討を行った。

3.1 OS とハードウェアの策定

高機能なサーバを安価に独自開発し、サポートメンテナンスも可能とするため、内部情報の公開されたオープンソースの OS の有効活用を前提とした。

IPv6 プロトコルが安定して動作するオープンソースの OS としては、BSD 系と Linux 系のものが存在する。特にネットワークハンドリング部分については、そもそも世界で初めて IPv4 の IP プロトコルが OS 内部に搭載されて普及に貢献した OS であり、本格的な IPv6 対応開発でも先行した BSD 系の完成度の高さが魅力的である。

しかしながら、今回の開発では、1. 小型のハードウェア上で安定稼働する必要があること、2. 中小企業が本研究開発をきっかけとして自力で開発やメンテナンスが継続できること、の2点を重視し、現時点で幅広く普及しているため、ちまたに数え切れない参考文献が存在しているため、数多くのハードウェアへの対応も迅速に行われている、Linux プラットフォームを採用することとした。

なお、小型サーバのハードウェアについては、これも基本アーキテクチャが広く公開され、複数のメーカーが提供している PC 互換アーキテクチャの中で検討することにした。

3.2 ユーザインタフェースの検討

Linux OS の設定は基本的には、各種設定ファイルをカスタマイズしていくことで行う。しかしながら、最終ユーザが Linux 等のエキスパートではなく、一般ユーザであることを想定するには、テキストエディタを駆使した各種設定ファイルの書き換えという複雑な設定作業ができることを前提にはできない。

Linux の各種設定補助ツールとしてのソフトウェアを調査したところ、Webmin と呼ばれる GUI 環境が存在することがわかった。Web ベースのシステム管理ツールとなっており、PC 等から Web ブラウザを通じて GUI の操作で各種システム設定ができるものである。Fig. 1 に初期状態の Webmin の画面を示す。

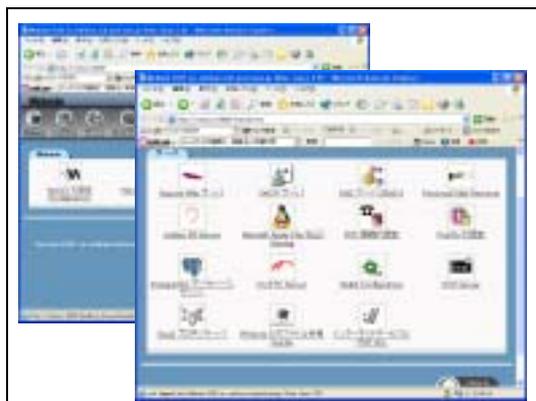


Fig.1 Webmin, web-based interface for system administration

Webmin の初期状態では、システム上のかなりの部分の設定を実現しようとして少々雑然としている。また、IPv6 に特化した設定項目が存在しないなど、このままの利用は難しいが、miniserv という小型の WWW サーバを用いた動

作原理など、基本動作部の応用が可能であり、カスタマイズによって開発期間を大幅に短縮できると考えられる。

3.3 基本ハードウェア構成

PC 互換性のあるアーキテクチャで、大幅な小型化を実現可能とするハードウェアとして、組み込み用基板の VIA Technology 製、EPIA Mini-ITX(EPIA-5000)を中心とした構成で動作状況を検証した。Fig. 2 に ITX 基盤を、Fig. 3 に、それを中心として構成された、動作検証用の試作品を示す。

部品構成には、安価で調達のしやすいものを中心に集めている。また、EPIA-5000 は、CPU がファンレスで動作する基盤であるため、完成品の故障原因が少ない。加えて、外部電源による構成も試してみた。電源部分を筐体外部に追い出せれば、熱対策が容易になるためである。

これらの構成で、Linux の稼働状況をテストした。



Fig.2 VIA Mini-ITX EPIA-5000



Fig.3 Prototype for verification

本構成にはフロッピードライブが存在しないため、CDROM から起動し、Vine Linux のインストールを試みた。ところが初期接続の A 社製 CDROM ドライブでは、ブートプロセスが終了し、インストール用カーネルが起動し

た瞬間に CDROM を認識しなくなる問題が発生し、インストールを先に進めることができなかった。

そこで、インストールイメージを提供する NFS サーバを所内 LAN に仕立てて、特別に用意した USB フロッピーから、ネットワークインストールを起動するという特殊な手法で先に進めた。この方法でインストールを完了することはできたが、インストール完了後にカーネルパニックが発生し、サーバが稼働しないという状況に見舞われた。

Vine 標準インストールで同時にインストールされる旧版カーネル(2.2.20)を試したところ起動した。原因を探ったところ、CPU アーキテクチャとして、Intel Pentium Pro 系のカーネルがインストールされているのが原因であり、VIA の CPU の VIA-C3 をターゲットして、カーネルを作成し直すことによって最新カーネル(2.4.20)で稼働させることができた。

ここに至っても A 社製 CDROM の認識は不可であったため、最終手段として、パーツ変更(B 社製に変更)を行った結果、こちらも認識させることができた。

3.4 フラッシュメモリ対応

容易に交換可能なフラッシュ RAM 上でシステムの展開ができれば、メンテナンス上有利になると考え、IDE インタフェースにフラッシュ RAM を接続できる変換器を用いたものと、USB タイプのフラッシュメモリの動作確認を行った。



Fig. 4 Compact Flash and IDE interface converter

フラッシュ RAM を IDE インタフェースに接続するタイプのものを Fig-4 に示す。この方式では OS の起動イメージそのものを格納することが可能となるため、システム全体を簡単にアップデートできると考え、稼働確認を行ったが、安定した OS の起動には成功しなかった。ファイルシステムのアロケートまでは成功するが、電源再投入で、領域認識が不良になる。

原因として変換器がフラッシュメモリの「IDE 互換モード」を必須としているが、昨今のデジタルカメラ用途では

利用しないモードであるため、最近の製品では低価格化のために省略されているためではないかと思われる。

次に USB インタフェースを持ったフラッシュ RAM の動作実験を行った。外観と接続状態を Fig-5 に示す。

このタイプのフラッシュメモリは、システムの起動こそできないが、非常にコンパクトで、容量も 32Mb から 256M 以上のもので各種入手可能であるため、システムのメンテナンス用途、緊急復旧用に活用できる。

Version 2.4 のカーネルが動作している Vine Linux で実験したところ、Linux 稼働中に USB インタフェースを差し込んでも、その時点で正しく SCSI HDD デバイスの一種として認識され、FAT ファイルシステムでマウントすれば、格納データにアクセスすることができた。

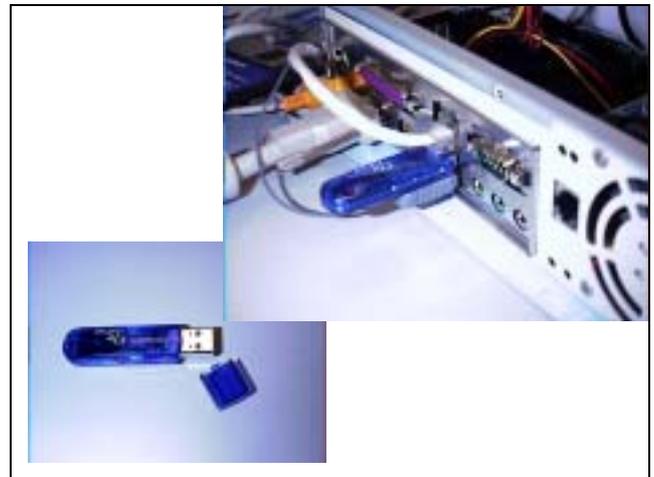


Fig. 5 USB Flash memory

4 . IPv6 実験環境整備

WIDE プロジェクトの有する IPv6 アドレスの割り当てを申請し、2001:200:168::/48 のアドレスブロックを取得した。

WIDE NARA NOC(WIDE Project の奈良における接続ポイント)と工業技術センターとの通信回線は現在 1.5Mbps の専用回線であり、この回線に PPP(Point to Point Protocol)を用いて IPv4 接続が行われていたが、ここに IPv6 の PPP 信号を重層設定し、IPv4/IPv6 デュアルスタック接続を実現した。Fig-6 に概要図を示す。

外部インターネットと IPv6 通信を行うためには、IPv6 の経路情報(ルーティングテーブル)の設定も必要である。WIDE 6bone では、ルーティングプロトコル(ルーティングテーブル交換用の通信プロトコル)として OSPF を用いた運用が行われている。ただ、今回、WIDE NARA NOC とセンター間の接続は専用回線を用いた 1 拠点接続であるため、WIDE NARA NOC のルータ装置において 2001:200:168::/48 への経路をインターネット側に広告(広

く通知)し、NARA NOC と当センター側との経路情報は静的経路設定(Static)にした。

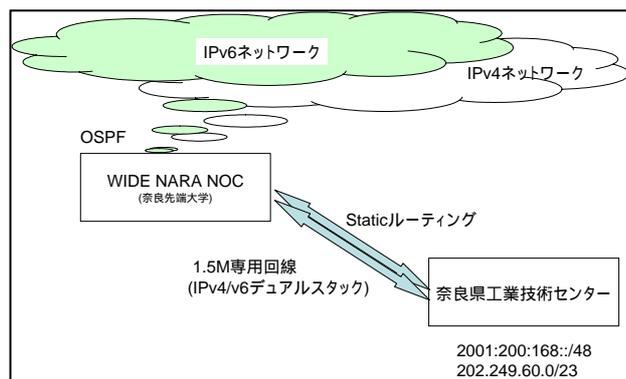


Fig. 6 IPv6/IPv4 Dual stacked connection to the Internet

センター側 LAN のアドレスブロックを 2001:200:168:1::/64, 2001:200:168:2::/64 と定義し、工業技術センターのルータ装置からそれぞれに RA(Router Advertisement)を流すという設定を行った。この後、センター内 LAN において Free BSD, Linux, Windows XP などの IPv6 対応の OS を起動したシステムを接続し、自動的に IPv6 アドレスの設定が行われること、そしてこれらのシステムによりセンター内の LAN から IPv6 通信ができることを確認した。

IPv6 プロトコルの通信確認は、通信確認先として、KAME プロジェクトの WWW サーバ、慶応大学内に存在する、WIDE プロジェクトの実験用サーバなどを対象に、http, telnet, ssh, ftp プロトコルの通信を試みることで行った。

また、IPv6 対応の、DNS 設定、WWW サーバ設定については、IPv6 クライアントシステムを活用して、FreeBSD+KAME による設定と Vine Linux+USAGI による運用実験を行い、それぞれの挙動を確認した。

DNS については、IPv6 アドレスを定義する AAAA レコードの定義を追加することで挙動確認を行った。サーバが DNS クエリーを IPv6 で受けるためにはサーバそのものが IPv6 対応でなければならないが、これに関しては、bind9 をソースからビルドすることで対応した。

WWW については、apache2 が標準で IPv6 対応しており、これも、Vine 標準ではパッケージが存在しないため、ソースからビルドして起動確認を行った。

なおサービス用のサーバでは、自動設定によらない固定 IPv6 アドレス付与が必要となる場合があるため、サーバへの静的アドレス割り当ての実験も行った。

5. まとめ

中規模ネットワークをサポート可能な IPv4/IPv6 両対応のネットワークサーバ開発の一年目の報告を行った。

最初に本研究開発が対象とする、次世代インターネットプロトコル、IPv6 の概要について述べ、次に小型で安価なサーバを開発するためのプラットフォームについて検討した結果を述べた。最後に工業技術センターネットワークの IPv6 対応についての報告を行った。

特にセンターネットワークから、世界に対して IPv6 プロトコルの通信確認が行えることになった進展は大きい。この環境は今後 IPv6 プロトコルに関する試験研究の基盤となるものである。

また、小型サーバの試作に先立って様々なパーツ類の組み合わせによる動作状況調査と行ったことにより、パーツ間の相性や、OS とパーツの相性に関する問題は、かなりの確度で発生し、あらかじめの動作確認が重要であることを再認識した結果となった。

1年目では、主に基礎的な技術調査とサーバのプラットフォームに関する基礎開発を行った。2年目では、試作サーバについて、ネットワークサーバとして基本的にサポートする機能や、各種設定の自動化、ユーザインタフェースの詳細などの設計・開発と同時に、筐体設計・試作を経て、最終試作品の完成を目指す。

また、今年度に完成した、IPv6 通信可能なネットワーク基盤上で、試作完成品のネットワーク動作状況確認、通信性能評価までを行う。

参考文献

- 1) ネットテクノロジーラボ: 入門 IPv6, 技術評論社, (1999)
- 2) 松平直樹: IPv6 ネットワーク実践構築技法, オーム社, (1999)
- 3) 砂原秀樹, 増田康人, 長橋賢吾, 有賀征爾: 使って学ぶ IPv6, 株式会社アスキー, (2002)
- 4) Intel Corporation: micro ATX Motherboard Interface Specification Version 1.1, Intel Corporation, (2002)
- 5) Intel Corporation: ATX Specification Version 2.1, Intel Corporation, (2002)
- 6) Intel Corporation: ATX/ATX12V Power Supply Design Guide Version 1.2, Intel Corporation, (2000)
- 7) KAME Project. <http://www.kame.net/>
- 8) USAGI Project. <http://www.linux-ipv6.org/>
- 9) Deering, S. and Hinden, R.: Internet Protocol Version 6 (IPv6) Specification, Request for Comments 2460, (1999)