USAGIプロジェクトによるIPv6基本ソフトウェアの開発

吉藤　英明[†]　神田　充[††]　高宮　紀明[†††]　関谷　勇司[††††]
江崎　浩[†]　村井　純[†††††]

Development of IPv6 Software System by the USAGI Project

Hideaki YOSHIFUJI[†], Mitsuru KANDA[††], Noriaki TAKAMIYA[†††], Yuji SEKIYA[††††], Hiroshi ESAKI[†], and Jun MURAI[†††††]

あらまし　近年，インターネットにおけるアドレスの不足が深刻化しており，この抜本的解決策として IPv6 が注目されている．本論文では Linux における IPv6 基本ソフトウェアのアーキテクチャ設計提案とその実装評価を行っている．具体的には，新しいタイマ管理と排他制御ポリシの導入，プロトコル処理におけるモジュールやインタフェースの再定義，IPv4 からのより安全かつ円滑な移行性を可能にするための新しいソケットインタフェースの導入を行った．さらに，モジュール性の高い IP セキュリティ機能を実現し，仮想デバイスによる移動透過性の高いモバイル IP の実現に向けたソフトウェアアーキテクチャの設計も行なった．最後に本研究により Linux における IPv6 プロトコルスタックの品質が大幅に改善されたことを定量的に示した．

キーワード　IPv6，Linux，USAGI プロジェクト，オペレーティングシステム，タイマ管理，排他制御，モジュール化

1. まえがき

IPv6(Internet Protocol version 6) [1] は現行のインターネットプロトコルである IPv4(Internet Protocol version 4) が抱える種々の問題，特に通信に必要なアドレス空間の枯渇問題を根本的に解決することを目的に，IETF(Internet Engineering Task Force) で標準化が進められてきた次世代インターネットプロトコルで，128 ビットの事実上無限ともいえる広大なアドレス空間を提供する．

1990 年代前半から IETF において進められてきた IPv6 の仕様はこれまでにほぼ確定した．IPv6 技術は，試用期を経て，種々のネットワーク機器や，オペレーティングシステム（OS）を含むソフトウェアのベンダが対応を推進しており，さらに，ISP(Internet Service Provider) による商用サービスも始まるなど，いよいよプロダクトクォリティな実用段階に入ってきている．

1998 年には日本における Linux IPv6 ユーザが集まって Linux IPv6 Users Group JP が発足し，Linux IPv6 プロトコルスタックの実運用に即した種々の検証が行なわれたが，未実装機能の多さ，準拠仕様の古さと，仕様準拠度の低さなどの多くの不都合があることが指摘された．

特に，以下の 2 つの課題が重要な問題として認識された．

（1） ソケットアドレス構造体（**sockaddr_in6{}**）にスコープ識別子を格納するメンバがないなどの，基本ソケット API におけるスコープの概念の欠如（**sin6_scope_id 問題**）

（2） アドレスの自動設定を含む近隣探索機能（IPv4 でのアドレス解決プロトコル（ARP）に相当す

[†] 東京大学大学院情報理工学系研究科，東京都
Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo, Tokyo, 113-8656 Japan
[††††] 慶応大学大学院政策・メディア研究科，神奈川県
Graduate School of Media and Governance, Keio University, 5322, Endo, Fujisawa, Kanagawa, 252-8520 JAPAN
[††] 株式会社東芝 研究開発センター，神奈川県
Corporate Research & Development Center Toshiba Corporation, 1, Komukai Toshiba-cho, Saiwai, Kawasaki, Kanagawa, 212-8582 JAPAN
[†††] NTT ソフトウェア株式会社，神奈川県
NTT Software Corporation, 223-1, Yamashita-cho, Naka, Yokohama, Kanagawa 231-8851 Japan
[†††††] 慶応大学環境情報学部，神奈川県
Faculty of Environmental Information, Keio University, 5322, Endo, Fujisawa, Kanagawa, 252-5222 JAPAN

る機能を包含する）の不具合（注1）

Linux カーネルのネットワーク関連の開発を担っている netdev [3] に対して様々な提案の行なわれる中，Linux Conference '99 会場に集まった Linux IPv6 Users Group JP の主要メンバである筆者らは，これらの問題を集中的に解決するために，USAGI(Universal Playground for IPv6) プロジェクト [4] を発足させることになった．

USAGI プロジェクトでは，WIDE プロジェクトの支援の下，カーネルから種々のユーザ空間上のライブラリ，アプリケーションにわたる総合的な基本ソフトウェアの研究開発を行なっており，IPv4 からのより安全かつ円滑な移行性の実現と，IETF 規格への高い整合性を実現し，さらに，最新規格に基づいた新たな機能も実装する．

本論文では，筆者ら（すなわち USAGI プロジェクト）が Linux における実装上の問題点を解決するために行なった基本アーキテクチャの提案とその評価を行なっている．具体的には，新しいタイマ管理と排他制御ポリシの導入，プロトコル処理におけるモジュールやインタフェースの再定義，IPv4 からのより安全かつ円滑な移行性を可能にするための新しいソケットインタフェースの導入，さらに，柔軟性の高い IP セキュリティアーキテクチャや，仮想デバイスを用いたより移動透過性の高いモバイル IP の実現に向けたソフトウェアアーキテクチャを議論している．

2. タイマ管理と排他制御ポリシ

近隣探索（Neighbor Discovery）は，IPv6 の最も特徴的な機構の一つで，ルータ探索とアドレス自動設定，リンク層アドレス解決と不到達性検出および，リダイレクトに分類されることはよく知られるところである．ネットワーク上に起こる種々のイベントとタイマによる，多様な状態の正確な管理が必要とされる．

例えば，データリンク層のアドレス解決と不到達検知では，近隣ノードの状態を保持する近隣キャッシュエントリが管理されるが，特に正確な不到達性の検知には，エントリの状態に依存する精緻なタイマ管理が必要である．また，ノードに割り当てられたアドレスの管理も，その有効期間について正確な時間管理が要求される．

（注1）：IPv6 仕様準拠度を評価する TAHI プロジェクトによる適合性試験によって明確になった [2].

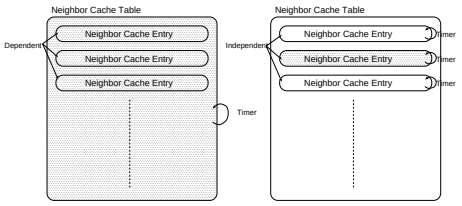


図 1　近隣キャッシュにおけるタイマおよび排他制御の局所性（左:Linux, 右:USAGI）

Fig. 1　Locality of timer and mutual exclusion of the Neighbor Cache Entries (Left: Linux, Right: USAGI)

従来の実装においては，近隣キャッシュエントリの状態遷移やアドレスの有効期間の管理が，それぞれ 30 秒や 120 秒という時間スケールを基準としたタイマによるポーリングで実現されていた．このため，数秒以下の精度でのタスク処理が要求される近隣キャッシュエントリの状態遷移やアドレスの有効期間の管理がきわめて不正確なものとなっていた．

この問題の解決方法としては，状態遷移処理を十分な精度で行なうよう，定期タイマ間隔を短くする方法が知られており，タイマ処理の負荷を最小にできる可能性がある．しかしながら，この状態遷移のための処理は常に十分な頻度で繰り返さなければならず，さらに，個々のエントリを網羅するために，それらを保持するテーブル全体を，全体の処理が終わるまで排他的に扱い，かつ同時に行なえる処理も制限しなければならない．すなわち，排他処理が細分化され，カーネル内で複数の処理が並行して行なわれる状況では，テーブル全体の排他制御は効率の低下を招いてしまう．

そこで，USAGI 実装においては，近隣キャッシュエントリの状態遷移を整理し，エントリ毎にタイマを用意して，タイマによるエントリの状態遷移処理を 1 つに集約し，状態に応じたタイマを実行させることとした．これにより，個々のエントリ相互，およびその他の各種資源との資源独立性が高まり，大域的な排他制御を削減した（図 1）．

一方，アドレス管理に関しては，全アドレスで最も直近に評価すべき時刻を推定して次のタイマを決定することにより，アドレスの非優先化および無効化を確実に実現している．処理タスクの実行頻度は最頻でも 0.5 秒毎とすることにより，アドレス管理においてしばしば発生する，ほぼ同時に処理すべきアドレスが複数あった場合の負荷を軽減することができるようなシステムとした．

3. プロトコル処理におけるモジュールおよびインタフェースの再定義

近隣のノードの探索とその到達性の管理，という処理自体は，IPv6 固有のものではない．実際に通信される内容の詳細は異なるが，IPv4 における ARP も同様の役割をもつ．このため，それらの中核的な処理を統合することが考えられる．事実，Linux においては，IPv6 の近隣キャッシュエントリと，IPv4 の ARP エントリ，DECnet の近隣エントリが，核となる単一の処理部によって扱われていた．

しかしながら，Linux においては，それらの違いは，状態遷移の時間設定程度であったため，近隣通知メッセージ（Neighbor Advertisement）のように，メッセージに含まれるフラグや近隣の状態に依存した複雑な処理ができなかった．

USAGI 実装においては，近隣キャッシュエントリ更新処理を，その原理と前章のタイマ管理・排他制御ポリシにしたがった枠組として再設計した上で，処理関数の引数の意味を拡張して IPv6 近隣通知メッセージなどの特殊処理を派生させ，適切な近隣探索の状態遷移を実現した．

また，近隣探索メッセージ処理については，従来，処理方法が統一的でなく，悪意のある通信からできるだけ保護し，安定運用に不可欠である正当性試験の多くが不十分であった．USAGI 実装においては，近隣探索プロトコルで共通に必要となる処理，特に近隣探索のオプション解析部を共通部品化するとともに，受信処理部の構造を統一化して可読性を向上させ，正確で堅牢な正当性試験を実現している．

4. IPv6 への円滑な移行性の実現

IPv6 では，IPv4からの移行技術の一つとして，IPv4 射影アドレスと呼ばれる機構を用意している．これは，IPv4 空間を広大な IPv6 空間の一部に射影するもので，アプリケーションは，IPv4 の通信相手は IPv6 のそれと同様に扱うことができる [5]．これにより，アプリケーションは IPv6 に対応するだけで IPv4 も簡単にサービスできる利点がある．しかしながら，特にアクセス制御の観点では，通信相手のアドレスの形式が従来の IPv4 と異なるために，従来の制御ファイルをそのまま流用できなかったり，アプリケーションの処理が繁雑になってセキュリティ上の弱点が生じやすいという欠点がある．

表 1　種々の環境における IPv4 射影アドレス対応戦略
Table 1　IPv4 mapped address strategy on various platforms

環境	射影アドレス	ポート共存	仕様適合	潜在安全性
FreeBSD 4.x	◎	○	●	△
NetBSD 1.5	△	◎	△	○
OpenBSD	×	◎	×	◎
Solaris 8	◎	○	●	△
Windows XP	×	◎	×	◎
Linux	◎	×	○	×
USAGI Linux	◎	●	◎	△

◎:非常に優れている／○:優れている:／●:やや優れている／△:やや問題がある／×:問題がある

表 1 に示すように，IPv4 射影アドレスのサポート，TCP や UDP それぞれにおける IPv6 と IPv4 のポート共有の可否などに関して，実装ごとに異なる方法を選択している．Linux においては，IPv6 射影アドレスをサポートし，TCP や UDP のポート空間を共有していた．このため，射影アドレスをサポートしない OpenBSD などを考慮して IPv4 と IPv6 のそれぞれにソケットを作成するようなアプリケーションや全く別のプロセスでサービスするようなアプリケーションを必ずしもうまく動作させることができなかった．これは**double-bind 問題**と呼ばれている．

USAGI プロジェクトではセキュリティ上および利便性，従来の挙動との互換性の観点から，この挙動を整理した．具体的には，IPv4 射影アドレスを無効にするための IPV6_V6ONLY ソケットオプション [6] を導入し，IPv6 ソケットに対してこれが設定されている場合に限り IPv4 ソケットとの共存を許すものとするように拡張した．これにより，当該ソケットオプションを設定しない従来の Linux アプリケーションのなどとの互換性を維持するとともに，特にアクセス制御などを重視するアプリケーションもこのソケットオプションを利用することで実現可能となった．セキュリティを重視するアプリケーションは今後このオプションを利用すると考えられるため，この方式は有効に機能すると期待される．

USAGI 実装では，これに加え，未指定アドレスと指定アドレスが SO_REUSEADDR ソケットオプションを利用して同一ポート上に共存した場合にパケットは指定アドレスでのサービスに優先して配送されることを利用して，未指定アドレス上のサービスへのパケット配送を妨げる攻撃の脅威を低減している．SO_REUSEADDR ソケットオプションの効力を同一

ユーザに限り，さらに，IPV6_V6ONLY オプションについても，同様な対策を適用している．

これらの機能の実現のため，TCP,UDP および IPv4,IPv6 の組合せで 4 つある検査関数を従来の実装で，不可解であった SO_REUSEADDR オプションなどの挙動も含めて原理に立ち戻って整理し，論理的かつ簡明にすべての組合せをつくす方法に改めて，複雑な挙動の実装に際して誤りを起きにくくしている．

5. IP セキュリティの実現

IP セキュリティ（IPsec）[7] の機能は，IPv6 仕様上必須とされているにもかかわらず，現在のところ，カーネル配布物の中に IPsec の実装は含まれていなかった．一方，IPv4 については FreeS/WAN プロジェクト [8] がその実装を独自に提供し，IPv6 についても，この FreeS/WAN プロジェクトのソースコードを一部利用していたものが存在していただけであった．FreeS/WAN プロジェクトの成果物である IPv4 の IPsec 実装を流用して実装をすすめていく方法が考えられるが，以下の理由により本章で提案しているソフトウェア構造での実装を行なった．

FreeS/WAN プロジェクトの実装では IPsec のアーキテクチャーの中で IPv4 や IPv6 に関わらず共通化できる部分である SAD(Secutirty Association Database) や SPD(Security Policy Database) など，強く IPv4 アドレス体系に依存している．さらに，パケット処理部もトンネリングの手法を用いた BITS(Bump In The Stack) と呼ばれる IP 層（ネットワーク層）とデータリンク層の間に実装されているおり，パケット転送効率の低下が発生してしまう．

図 2 に，USAGI プロジェクトによる IPsec 実装の構成を示す．

5.1 IP バージョン非依存なデータベースの実現

IPsec 機能の中で SAD と SPD は IPv4 や IPv6 といった IP バージョンには依存しない．すなわち，USAGI 実装においては，IPv4 と IPv6 で共用が可能な SAD と SPD を実現している．これらのデータベースには，その利用するプロトコルに応じた IPv4 アドレスまたは IPv6 アドレスとポート番号などが格納される．汎用性や IPv6 のスコープの概念への対応の観点から，これらはそれぞれ **sockaddr_in{}** 構造体と**sockaddr_in6{}** 構造体を使用すると都合が良い．このため両者のポータビリティのために定義された**sockaddr_storage{}** 構造体を新たにカーネル内

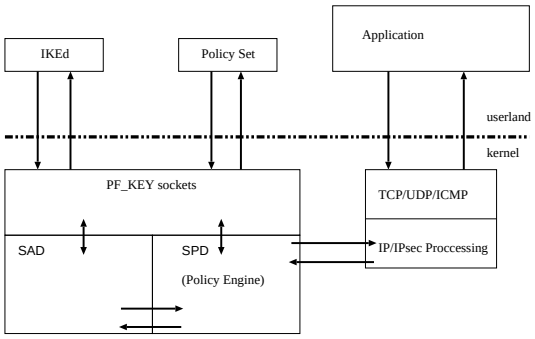


図 2 USAGI IPsec 実装の構造
Fig. 2 Structure of USAGI IPsec

に導入することで，IP バージョンに依存することのないデータベース構造を実現している．

5.2 暗号/認証アルゴリズム

IPsec で使用する暗号化や認証のためアルゴリズムは IPsec 固有のものでなく一般的なものである．そのため，USAGI プロジェクトでは上記アルゴリズムを cryptoapi と呼ばれる実装 [9] を利用している．この実装では暗号化/復号化関数等のインタフェースを抽象化しており，様々なアルゴリズムを容易に追加削除することができる．これにより新しいアルゴリズムに対応することも容易であるし，また，種々の暗号輸出規制などが存在する国においても，アルゴリズムだけを除外して配布ことが容易に実現できる．現在 USAGI プロジェクトの IPsec で利用できる暗号化アルゴリズムは des-cbc と 3des-cbc であり，認証アルゴリズムは hmac-md5 と hmac-sha1 である．

5.3 PF_KEY ソケット仕様

PF_KEY はユーザ空間アプリケーションがカーネルにセキュリティポリシ（Security Policy）を設定するためにインタフェースである．PF_KEY(Version 2) の仕様は RFC2367 [10] に定義されているが．この文書に定義されている仕様は基本的なものだけで十分といえず，各種のセキュリティポリシなどを設定するためには，実装ごとに独自の拡張がなされている．USAGI プロジェクトとしてはこの拡張部分は FreeS/WAN プロジェクトの定義を利用することによってユーザ空間のアプリケーションとの互換性を保つようにしている．

6. モビリティ機能の実現

モバイル IPv6 は，IPv6 で注目されている機能の 1 つである．モバイル IPv6 では IPv6 の機能を活かし，IPv4 のモバイル IP では必要とされている外部エー

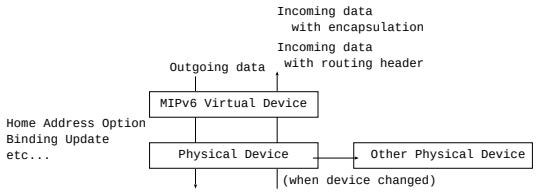


図 3　仮想デバイスを用いたモバイル IP の実装
Fig. 3　Mobile IP implementation using virtual device

ジェント（Foreign Agent）を省略した通信が可能である．

USAGI 実装では，ヘルシンキ工科大学（HUT）によるモバイル IPv6 の実装を取り入れ，USAGI 実装が基礎をおく最新カーネルでの動作が可能となっている．一方で，HUT の実装を取り入れた結果，以下の問題点が顕在化した．

（1） 移動ノードにおけるホームアドレス（Home Address）の設定

（2） ダイナミックホームエージェント探索（Dynamic Home Agent Discovery）

（3） Netfilter および IP セキュリティ機能との相互運用性

（4） モバイル IPv6 起動時の初期化

ホームアドレスの設定は，現在ネットワークに接続している物理デバイスに直接設定するため，デバイスをまたがった移動透過性に欠けている．USAGI ではこの問題点を仮想デバイス（Virtual Device）を用いて解決している．仮想デバイスを導入することによりホームアドレスの管理を物理的なデバイスから分離し，デバイス間で透過的な移動が可能となる（図 3）．

ダイナミックホームエージェント探索については，現在の Linux がエニキャストがサポートされていないことから，HUT 実装では通常のユニキャストアドレスのようにエニキャストアドレスを割り当ててダイナミックホームエージェント探索を実現している．そこで，USAGI ではエニキャストを実装し，この枠組の中でダイナミックホームエージェント探索を実現する．

また，USAGI プロジェクトは Netfilter や IP セキュリティ機能など IPv6 上での重要な機能を実装しているが，現在のモバイル IPv6 の機能ではこれらの機能との共存を前提としていない．Netfilter の機能は，モバイル IPv6 の機能が必要としているが通常のパケットフィルタリング等との機能と併用したときの動作が保証されていない．USAGI プロジェクトではこれらの機能との共存ができるようなモバイル IPv6 の機能の実装を今後行なう．

表 2　TAHI 仕様適合性試験結果（“成功” 率，抜粋）
Table 2　TAHI Conformance Test Result（“PASS” Ratio）

試験	Linux		USAGI
分野	2.2.15	2.4.14	2.4 系 2001/11 末
ND	34%	39%	86%
SAA	4%	77%	98%

7. USAGI プロトコルスタックの評価

上述した USAGI プロトコルスタックにおける基本ソフトウェアの評価を TAHI 仕様適合性評価システムを用いて行なった（表 2）．本論文の論点である近隣探索（Neighbor Discovery），およびルータ探索・アドレス自動設定（Stateless Address Autoconfiguration）に関して，顕著な改善が実現されたことがわかる．なお，USAGI プロジェクトの成果は Linux 配布物にも反映されている．

8. むすび

本論文では，USAGI プロジェクトによる IPv6 基本ソフトウェアについて，その特徴的なアーキテクチャをいくつか取り上げて議論，評価を行なった．USAGI ソフトウェアは，基本的には従来の Linux カーネルのプロトコルスタックを踏襲しているが，近隣探索プロトコル，IP セキュリティ，モビリティといった実現において，IPv6 の機能をより自然，広範かつ安定に動作するための新しいソフトウェアアーキテクチャを提案し実装している．

最後に，今後の課題を挙げる．

（1） 全体構造としては，同様な処理が複数箇所に存在するなど，全体が必ずしも自然かつ効率的なソフトウェア構造とはなっていない部分が未だ数多く存在する．

（2） スコープの扱いの一層の改善．

（3） end-to-end 通信の実現によりますます重要となる鍵交換をはじめとしたセキュリティアーキテクチャの検討．

以上のような基礎・応用両面にわたる検討をもとに，より洗練され完成度の高い実装を広く提供して IPv6 の普及を図り，また標準化活動にも貢献する予定である．

文　　献

[1] S. Deering and R. Hinden, "Internet Protoco, Version 6 (IPv6) Specification," RFC2460, 1998.

[2] 福原一朗, 国武功一, 吉藤英明, 岡村耕二, 楠根雄志, 関谷勇司, "Linux でつなぐ IPv6 の世界," Linux Conference '99, 1999.

[3] Netdev mailing list, <netdev@oss.sgi.com>

[4] USAGI Project, <http://www.linux-ipv6.org>

[5] R. Gilligan, S. Thomson, J. Bound and W. Stevens, "Basic Socket Interface Extensions for IPv6," RFC2553, 1999.

[6] R. Gilligan, S. Thomson, J. Bound and W. Stevens, "Basic Socket Interface Extensions for IPv6," draft-ietf-ipngwg-rfc2553bis-04.txt, 2001.

[7] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol," RFC2401

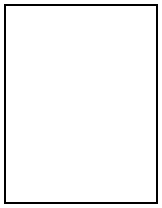
[8] FreeS/WAN Project, <http://www.freeswan.org>

[9] International Kernel Crypto API for GNU/Linux, <http://sourceforge.net/projects/cryptoapi/>

[10] D. McDonald, C. Metz, B. Phan, "PF_KEY key Management API, Version 2," RFC2367

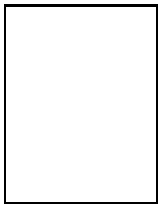
[11] D. Harkings, D. Carrel, "The Internet Key Exchange (IKE)," RFC2409

（平成 x 年 xx 月 xx 日受付）



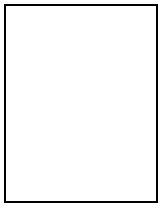
吉藤　英明　（学生員）

1999 年東北大学工学部情報工学科卒. 2001 年同大大学院情報科学研究科修士課程了，東京大学大学院情報理工学研究科博士課程入学. コンピュータネットワークにおける情報伝達，次世代ネットワークの構築・管理に関する研究開発に従事.



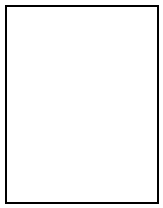
神田　　充

1997 年 東北大学工学部機械電子工学科卒，1999 年 東北大学大学院情報科学研究科修士課程了. 同年株式会社東芝入社. ユービキタスコンピューティング及び次世代ネットワークの研究開発に従事.



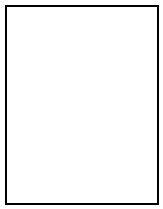
高宮　紀明

1994 年東京工業大学理学部情報科学科卒. 同年ＮＴＴソフトウェア株式会社入社. 現在インターネット技術センター配属. IPv6 ネットワークの構築およびプログラム開発，SE 業務に従事.



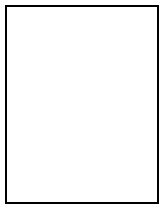
関谷　勇司

1997 年 京都大学総合人間学部基礎科学科数理情報論卒業，1999 年 慶応義塾大学政策・メディア研究科修士課程修了，1999 年 10 月より半年間，ISI(Information Siences Institute) にて訪問研究員を勤める. Lisp 言語に関する研究，次世代ネットワークの開発・構築に関する研究ならびに DNS に関する研究に従事.



江崎　　浩　（正員）

1987 年 九州大学 工学部電子工学科 修士課程 了,（株）東芝 入社. 1990 年より 2 年間 米国ニュージャージ州 ベルコア社, 1994 年より 2 年間 米国ニューヨーク市 コロンビア大学 CTR (Centre for Telecommunications Research) にて客員研究員. 1998 年 10 月より東京大学大型計算機センター助教授，2001 年 4 月より現職（東京大学 情報理工学系研究科 助教授）. ATM ネットワーク制御技術，高速インターネットアーキテクチャ，セルスイッチルータの研究・開発・マーケティングに従事. WIDE プロジェクトボード メンバー. MPLS-JAPAN 代表，IPv6 普及・高度化推進協議会専務理事，通信放送機構ジャパンギガビットネットワーク運営委員. 工学博士（東京大学）.



村井　　純　（正員）

1984 年 慶應義塾大学工学部数理工学博士課程 了. 1987 年 博士号取得. 1984 年 東京工業大学総合情報処理センター助手. 1987 年 東京大学大型計算機センター助手. 1990 年 慶應義塾大学環境情報学部助教授. 1997 年 慶應義塾大学環境情報学部 教授, 1999 年 慶應義塾大学 SFC 研究所所長. WIDE プロジェクト代表，社団法人日本ネットワークインフォメーションセンター理事長，ICANN ボード.