IPv 4 から IPv 6 への移行技術の検証

Investigation of Transition Technologies from IPv 4 to IPv 6 in Network Environment

小宮山 智之

要 約 昨今，ユビキタスという言葉を耳にする機会が増えている．そしてIPv 6は，ユビキタスネットワークの重要な要素の一つとされている．このIPv 6とは，現在のインターネット/イントラネットの基盤プロトコルであるIPv 4の次期バージョンであり，普及が進むブロードバンド環境や実用化の準備が進むユビキタス環境の展開をにらみ，IPv 4からの移行の必要性が取り沙汰されている．これを踏まえ，本稿では，IPv 4からIPv 6へ移行される過程において途中形成されるIPv 6とIPv 4の混在ネットワークを対象として，両プロトコル共存の環境を実現させる移行技術について紹介する．また実証実験を通した移行技術の検証について触れる．

Abstract Recently, we often hear a term“ ubiquitous". IPv 6 is, then, regarded as one of important elements that comprise a ubiquitous network. IPv 6 has been introduced as the next generation protocol for the Internet and Intranet, and is intended to replace IPv 4, the current protocol infrastructure. The requirements to replace IPv 4 to IPv 6 are stressed being driven by the current trend for rapid expansion of broadband Internet services and for the preparation of ubiquitous environment. During the transition period of protocols, it is unavoidable both IPv 4 and IPv 6 protocol coexist in a same network environment. This paper is prepared to introduce some transition technologies and feasible methods to handle the coexistence of both protocols based on actual proof test results.

1. はじめに

ネットワーク技術の発展と普及は著しく，今日，我々を取り巻く環境が変革の最中にある．特にブロードバンド環境やユビキタス環境の進展は，インターネットに新たな展開をもたらしている．例えば，インターネットのブロードバンドアクセス環境が浸透し，IP電話や動画配信といったリアルタイム通信や双方向通信アプリケーションが頻繁に利用されている．またユビキタスネットワークについても，インターネットを媒体とした様々な適用技術が検討されている．今後のネットワーク環境の方向性について，政府による「e-Japan 重点計画 2002」によると，「すべての機器が端末化する偏在的なネットワークへの進化」を目指し，「ネットワークに接続された多種多様で，極めて多数の端末を安全で，リアルタイムかつ自律的に制御・協調できるネットワーク技術」と「一つの端末にとらわれず，いつでもどこでも接続できる，十分な伝送容量を備えたネットワーク環境」の2005年までの実用化が掲げられている．

こうした今後のインターネットの発展に要求される技術条件に対し，現在のインターネットの基盤をなしているIPv 4（Internet Protocol Version 4）では対応できなくなってきている．例えばユビキタス環境で必要とされるアドレス割り当て数は，IPv 4のアドレス空間では満足しきれなくなる．加えて，IPアドレスの不足はEnd-to-End通信[*1]確保の阻害をもたらし，インターネットの設計思想である「End-to-End通信」によるエンドシステムでの自由なサービスの実現を妨げることに繋がる．更に，あらゆる端末からの通信がインターネットに溢れ出

すと，ルーティング情報の肥大によりルータ負荷が深刻な問題となる．また，誰もが身の回りのものを容易にネットワークへ接続するためには，特に知識がなくてもネットワークに繋ぐだけで利用が可能となる仕組みや，セキュリティを保持させるための仕組みも要求されてくる．この他，いつでもどこでもネットワークに接続できる移動端末に適応した仕組みや，リアルタイム通信を IP レベルで識別し効率的にルーティングさせるための仕組みなど，20 年以上前に定義された IPv 4 の機能では実現困難な要求が重なる．

これらの問題を解決し，IPv 4 に置き換わるインターネットプロトコルとして検討されたのが IPv 6（Internet Protocol Version 6）である．IPv 6 が持つ特徴を整理すると，「アドレス空間の拡大と End-to-End 通信の保持」「階層的アドレス構造によるアドレス管理の効率化と経路情報増加防止の実現」「セキュリティ，モビリティ，マルチキャスト機能の提供」「アドレス自動設定機能などプラグアンドプレイ機能の実装」とまとめられる．インターネットの継続的な発展と成長を支えるためには，この IPv 6 技術の導入と普及が欠かすことのできない要素になると思われるが，現状の IPv 4 によるインターネットが既に広く普及していることから，IPv 6 への移行は段階的に行われる必要がある．そして，しばらくは IPv 6 と IPv 4 が混在する時期が続くと考えられる．同時に，新しいインターネットの波が企業などのイントラネットに押し寄せてくるのも時間の問題であると思われ，企業でも IPv 4 から IPv 6 へスムーズに移行させる技術や両プロトコルを混在させて運用する技術の適切な選択と利用が大変重要な課題となってくる．

以上を踏まえ本稿では，IPv 6 への移行期において，両プロトコルが混在した環境を実現させる移行技術に関して述べる．また IPv 6 への移行技術の有効性を検証した IPv 6 実証実験について報告する．

2. IPv 6 への移行技術

IPv 6 への移行については，現状のインターネットサービス/イントラネットサービスの品質を落とさずに実施されることが重要である．IPv 6 は IPv 4 の基本概念をそのまま引き継いでいるが，今後のインターネットに必要とされる要求を満たすべく再設計されており，IPv 4 との相互接続性は持っていない．このため IPv 6 と IPv 4 を混在させるには何らかの移行技術を用いて，既存 IPv 4 ネットワークへの影響を最小限に抑える必要がある．本章では，IPv 4 から IPv 6 への移行において，ネットワークインフラの移行とアプリケーションの移行に分け，それぞれの移行技術について述べる．

2.1 ネットワークインフラの移行

ネットワークインフラにおいて，IPv 6 と IPv 4 の混在を可能にする技術が各種開発されている．このネットワークインフラの移行技術は，大きく「デュアルスタック」「トンネリング」「トランスレータ」に分類される．本節では，これら 3 種の移行技術について述べる．

2.1.1 デュアルスタック

デュアルスタックは，複数のネットワーク層プロトコルを同時にサポートし，通信対象により使い分ける技術である．

ここで扱うデュアルスタックとは，IPv 6 と IPv 4 を同時にサポートする「IPv 6/IPv 4 デュ

アルスタック」のことである．IPv 6/IPv 4 デュアルスタックホストの通信は，IPv 4 のみをサポートする機器との間では IPv 4 を使用し，IPv 6 をサポートする機器や別のデュアルスタックホストとの間では IPv 6 を使用する（図 1）．

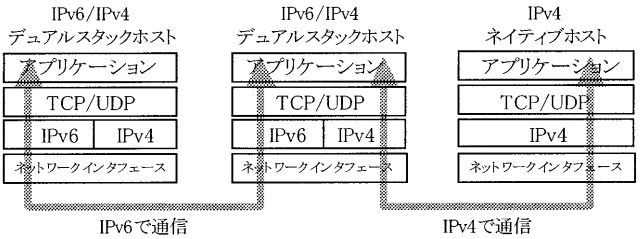

図 1　IPv 6/IPv 4 デュアルスタック

1）利　　点

IPv 6 と IPv 4 との間には互換性がない．この問題に対して，IPv 6 と IPv 4 の両方を同時に扱えるデュアルスタックは，IPv 6 への移行期において有効な対策手段となる．特に IPv 4 で運用している既存のホストやネットワーク機器をデュアルスタック化することで，IPv 4 ネットワークに影響を与えずに新たな機器を追加することなく IPv 6 の利用が可能となる．

2）留　意　点

通信を行うノード間に IPv 4 ルータなど IPv 4 ネイティブなネットワークが存在する環境下では，IPv 6 通信を行うことができない．また IPv 6 と IPv 4 間の相互通信，例えば IPv 6 ホストと IPv 4 Web サーバとの間での通信は行えない．この場合の通信では，後述するトンネリング技術やトランスレータ技術が利用される．なお，トンネリングやトランスレータを行う機器についても，IPv 6 と IPv 4 の両プロトコルの処理が要求されるためデュアルスタックで構成されている必要がある．

3）実装状況

今後，IPv 4 利用目的で出荷される機器であっても，その多くは IPv 6 に対応済みであるか，ある操作により IPv 6 への対応が可能となる機器が増えていくと思われる．なお，現在 IPv 6 に対応する機器のほとんどは IPv 6 ネイティブではなく，同時に IPv 4 もサポートするデュアルスタック機器である．

2.1.2　トンネリング

トンネリングは，あるノード間の通信においてノードが使用するプロトコルとは異なるプロトコルのネットワークインフラを利用する技術である．本項では，トンネリング技術の概要とその種類について述べる．

1）概　　要

IPv 4 から IPv 6 への移行初期段階には，ネットワークの大部分が IPv 4 によって構成される中で，IPv 6 端末が孤立する状況が想定される．IPv 6 ネットワークインフラが整備されるまでの間，IPv 6 端末同士の通信を既存の IPv 4 ネットワークインフラを利用して行えると，資源が有効活用できる．この IPv 4 インフラを利用して IPv 6 トラフィック

の配信を行う機能を提供するのが「IPv 6 over IPv 4 トンネル」である．

IPv 6 over IPv 4 トンネルでは，IPv 6 端末より送信された IPv 6 パケットがトンネル始点において IPv 4 ヘッダでカプセリングされることで，IPv 4 パケットとして IPv 4 インフラ上に配送される．配送されたパケットはトンネル終点でデカプセリングの後，IPv 6 プロトコルにより宛て先 IPv 6 端末へ届けられる（図 2）．カプセリングにおいて，付加される IPv 4 ヘッダのプロトコルフィールドには 41 が設定され，IPv 6 パケットのカプセリングであることが示される．

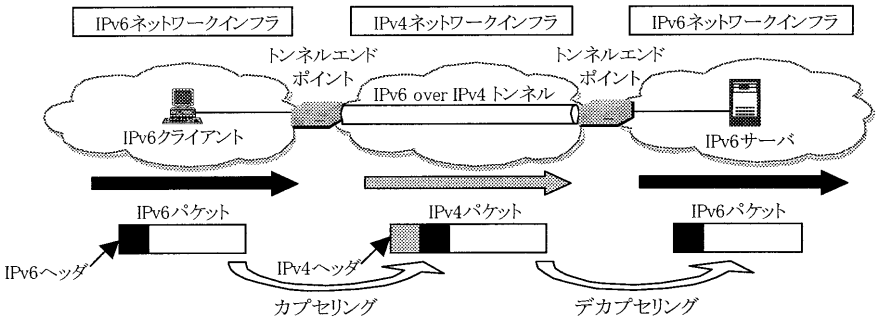

図 2　IPv 6 over IPv 4 トンネルのカプセリングとデカプセリング

IPv 6 over IPv 4 トンネリングにおいて，RFC（Request For Comments）2893 ではトンネルエンドポイント[*2] となる機器により表 1 に示すトンネリングの分類を定義している．

表 1　トンネルエンドポイントによるトンネリングの分類

トンネルエンドポイント	内　　容
ルータ to ルータ	IPv4 インフラを挟んだ二つの IPv6/IPv4 デュアルスタックルータ間でトンネリングするケース．
ホスト to ルータ	IPv4 インフラ内の IPv6/IPv4 デュアルスタックホストが，IPv6 インフラとの境界上にある IPv6/IPv4 デュアルスタックルータとの間でトンネリングするケース．
ホスト to ホスト	IPv4 インフラ内の二つの IPv6/IPv4 デュアルスタックホスト間でトンネリングするケース．
ルータ to ホスト	IPv4 インフラとの境界上にある IPv6/IPv4 デュアルスタックルータが，IPv4 インフラ内の IPv6/IPv4 デュアルスタックホストとの間でトンネリングするケース．

トンネリングにおいて，トンネル終点のアドレス決定方法には，トンネル始点となるノードに明示的に設定しておく方法と動的に決定される方法があり，RFC 2893 ではそれぞれ「手動設定トンネリング（Configured Tunneling）」と「自動設定トンネリング（Automatic Tunneling）」として規定されている．手動設定トンネリングと自動設定トンネリングは，以下の様に考えることができる．

■手動設定トンネリング

　カプセリングを行うノードに予めトンネル終点アドレス指定の設定を施すトンネリング形態．

■自動設定トンネリング

トンネリング終点アドレスの決定が，カプセル化されるIPv 6パケットのアドレスに埋め込まれたIPv 4アドレス情報や，DNSへ登録されたトンネル情報などにより行われるトンネリング形態．（以下，自動設定トンネリングについて本稿では自動トンネルと記述する．）

2） トンネリング技術の種類

トンネリングは，その使用範囲がイントラネット内での利用を想定しているものや，IPv 4インターネットを利用したIPv 6サイト[*3]間の接続を想定しているものなど，各々のトンネリング技術で想定された利用形態を持っている．

以下に主なトンネリング技術をいくつか紹介する．

① IPv 4互換IPv 6アドレスによる自動トンネル

RFC 2893に規定されている自動トンネル技術である．宛て先IPv 6アドレスとしてIPv 4互換IPv 6アドレスを用いることにより，トンネル終点のIPv 4アドレスが関連付けられる．IPv 4互換IPv 6アドレスは，128 bit中下位32 bitにIPv 4アドレスが埋め込まれ，残り上位96 bitが全てゼロとなる形態で表される．例えば，IPv 4アドレスw.x.y.zに対応するIPv 4互換IPv 6アドレスは，：：w.x.y.zと表される．

IPv 4互換IPv 6アドレスによる自動トンネルでは，トンネル始点において宛て先アドレスがIPv 4互換IPv 6アドレス（即ちプレフィックスが“：：/96”）となるIPv 6パケットをIPv 4ヘッダでカプセリングする．この時，IPv 4ヘッダの送信元アドレスにはトンネリング始点のアドレスが設定され，宛て先アドレスには宛て先IPv 4互換IPv 6アドレスに埋め込まれたIPv 4アドレスが設定される（図3）．

ただしIPv 4互換IPv 6アドレスによる運用は，IPv 4ホストにIPv 4アドレスを割り当てた際の運用と変わりがなく，ホストの通信としては単にIPv 4を使用して行うのと同じことである．このためトンネルを活用するケースとして通常は，イントラネット内では後述するISATAPが利用され，インターネットとの接続では同じく後述の6 to 4が利用されるケースが多くなると思われる．従ってIPv 4互換IPv 6アドレスによる自動トンネルを用いる通信は，実験的な用途に限られたものなど，その使用頻度は低いと思われる．

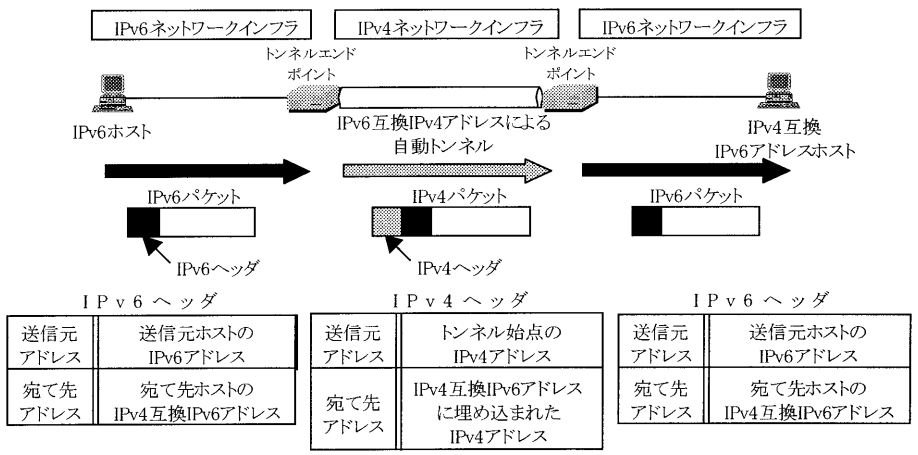

図 3　IPv 4互換IPv 6アドレスによる自動トンネルでのパケットの宛て先アドレス

② 6 to 4 （Connection of IPv 6 Domains via IPv 4 Clouds）

6 to 4 は，IPv 4 インターネットを介して IPv 6 サイト間，あるいは IPv 6 ホストとの接続を実現させる自動トンネル技術で，RFC 3056 で定義されている．6 to 4 を利用するサイトやホストには，グローバル IPv 4 アドレスが割り当てられていることが前提となる．

6 to 4 トンネルを用いた IPv 6 通信では，「6 to 4 アドレス」が用いられる．6 to 4 アドレスは「2002：：/16」で識別されるグローバル IPv 6 アドレスで，ISP から割り当てられたグローバル IPv 4 アドレス w.x.y.z を元に生成されるプレフィックス「2002 : WWXX : YYZZ：：/48」により構成される．（WWXX : YYZZ は，w.x.y.z で与えられる IPv 4 アドレスの 16 進表現を示す．）6 to 4 アドレスが与えられたサイトやホストは，それぞれ「6 to 4 サイト」，「6 to 4 ホスト」と呼ばれる．

6 to 4 トンネルは，通信対象の IPv 6 ホストによりトンネル終点ノードが異なる（表 2）．

表 2　6 to 4 トンネルによる通信対象ホストとトンネル終点ノード

通信対象の IPv6 ホスト	トンネル終点ノード
6to4 サイト内の 6to4 ホスト	6to4 サイトのエッジルータ（6to4 ルータ）
IPv4 インターネット上の 6to4 ホスト	通信対象の 6to4 ホスト
IPv6 インターネット上の IPv6 ネイティブホスト	インターネット上の 6to4 リレールータ

通信対象が 6 to 4 サイト内に存在する 6 to 4 ホストの場合（図 4 中の【a】），送信元からの IPv 6 パケットはトンネル始点にてカプセリングされ，宛て先アドレスには配信先の 6 to 4 サイトを示すグローバル IPv 4 アドレスが与えられる．そして届けられたパケットは 6 to 4 サイトのエッジルータでデカプセリングされる．6 to 4 サイトのエッジルータは「6 to 4 ルータ」と呼ばれ，6 to 4 トンネルにおけるカプセリング・デカプセリング処理を行う．サイトに到着し 6 to 4 ルータでデカプセリングされたパケットは，6 to 4 サイト内を IPv 6 パケットとして配送され，目的の 6 to 4 ホストまで届けられる．

通信対象が IPv 4 インターネット上の 6 to 4 ホストの場合（図 4 中の【b】），トンネル終点はこのホスト自体である．即ち IPv 6 ヘッダの宛て先アドレスはこのホストの 6 to 4 アドレスであり，またカプセリング後の IPv 4 ヘッダの宛て先アドレスには，宛て先ホストの 6 to 4 アドレスに埋め込まれたこのホスト自身を指す IPv 4 グローバルアドレスが設定される．

通信対象が IPv 6 インターネット上の IPv 6 ネイティブホストの場合（図 4 中の【c】）は，カプセリングを施す 6 to 4 ルータに予めトンネル終点となるノードが設定されている必要がある．このノードは「6 to 4 リレールータ」と呼ばれ，IPv 6 インターネットのエッジルータとして IPv 4 インターネットとの境界上に存在する．トンネル始点にてカプセリングされたパケットは IPv 4 インターネット上をルーティングされ，6 to 4 リレールータまで配送される．6 to 4 リレールータまで届けられたパケットはデカプセリングの後，IPv 6 パケットとして目的の IPv 6 ホストまで IPv 6 インターネット上をルーティングされる．このケースでは，IPv 6 ヘッダの宛て先アドレスは目的の IPv 6 ネイティブホストのアドレスであり，またカプセリング後の IPv 4 ヘッダの宛て先アドレ

スには 6 to 4 リレールータの IPv 4 アドレスが設定される．

6 to 4 による IPv 6 ホスト間接続のイメージを図 4 に示す．

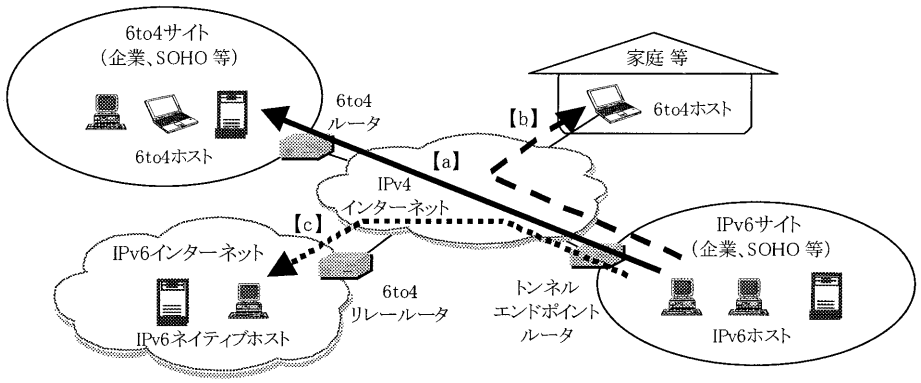

図 4　6 to 4 による IPv 6 ホスト間接続のイメージ

③　ISATAP （Intra-Site Automatic Tunnel Addressing Protocol）

ISATAP は主に IPv 4 ベースのイントラネット内（企業内）で IPv 6 ホスト同士を接続させる自動トンネル技術で，インターネットドラフトで規定されている．6 to 4 トンネルではグローバル IPv 4 アドレスが必要となるのに対し，ISATAP トンネルはプライベート IPv 4 アドレスによる構成も可能である．

ISATAP トンネルを利用した IPv 6 通信では，「ISATAP アドレス」が用いられる．ISATAP アドレスは，IPv 4 アドレス w.x.y.z を元に生成されるインタフェース識別子（128 bit 中　下　位 64 bit）：：0 : 5 efe : w.x.y.z あ　る　い　は：：0 : 5 efe : WWXX : YYZZ（WWXX : YYZZ は，w.x.y.z で与えられる IPv 4 アドレスの 16 進表現）と上位 64 bit の IPv 6 ユニキャストアドレスプレフィックスから構成される IPv 6 アドレスである．ISATAP アドレスは 1 つのプレフィックスで多数の ISATAP ホストの収容が可能である．このため，ISATAP トンネルの利用は大規模イントラネットにおける段階的な IPv 6 への移行においても有効な手法である．

④　Teredo （Tunneling IPv 6 over UDP through NATs）

Teredo は一つ，または複数の NAT（Network Address Translation）環境下の IPv 4 インフラにおいて IPv 6 による接続を可能にする自動トンネル技術で，インターネットドラフトで規定されている．主に SOHO や家庭での使用といった個人ユーザの利用に対応する．

前述のトンネリング技術が IPv 6 パケットを IPv 4 ヘッダでカプセリングするのに対し，Teredo では IPv 4 ヘッダと UDP ヘッダの組でカプセリングする（図 5）．Teredo パケットは，UDP パケットの通過が許可されている NAT を超えることができる．NAT を通過する際，Teredo パケットは従来の IPv 4 ベースの NAT 同様，送信元の IPv 4 アドレスと UDP ポート番号が変換される．

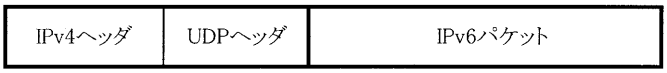

図 5　Teredo パケットフォーマット

⑤　トンネルブローカ　（IPv 6 Tunnel Broker）

トンネルブローカは，クライアントからのトンネル要求を管理する「トンネルブローカ」と呼ばれる専用のサーバによって自動トンネルを実現させる技術で，RFC 3053 に規定されている．トンネルブローカは，IPv 4 インターネット上にて IPv 6 ネットワークへの接続を提供する IPv 6 の仮想 ISP の様に機能する．

IPv 6/IPv 4 クライアントがトンネルブローカに IPv 6 接続を要求すると，トンネルエンドポイントとなる「トンネルサーバ」にクライアントの情報が登録され，トンネルサーバ側のトンネルが設定される．クライアントへはトンネル構築に必要な情報を含むトンネルクライアントスクリプトが配信され，クライアントのトンネル自動設定が行われる．

トンネルブローカモデルに似た技術に「トンネルサーバモデル」がある．トンネルサーバモデルは，クライアントがトンネルブローカを介さず直接トンネルサーバへトンネル要求を行い，トンネルクライアントスクリプトを受け取り，トンネルの構成を行うものである．

⑥　6 over 4　（Transmission of IPv 6 over IPv 4 Domains without Explicit Tunnels）

6 over 4 は，IPv 4 イントラネット上で IPv 6 ノード間の接続を行う自動トンネル技術で，RFC 2529 で定義されている．6 over 4 では，IPv 4 インフラがマルチキャスト機能を持つリンクの一つとして用いられる．

6 over 4 トンネルを利用した IPv 6 通信では，「6 over 4 アドレス」が用いられる．6 over 4 アドレスは，IPv 4 アドレス w.x.y.z を元に生成されるインタフェース識別子：：WWXX : YYZZ と上位 64 bit の IPv 6 ユニキャストアドレスプレフィックスから構成される．（WWXX : YYZZ は，w.x.y.z で与えられる IPv 4 アドレスの 16 進表現を示す．）

6 over 4 で使用される IPv 4 インフラは IPv 4 マルチキャストに対応している必要がある．更に他のトンネリング技術の登場により 6 over 4 の必要性は少なくなっている．このため，6 over 4 が利用されるのは稀であると思われる．

⑦　DSTM　（Dual Stack Transition Mechanism）

DSTM は，前述の他のトンネリング技術とは異なり，IPv 6 インフラを利用して IPv 4 通信を行う IPv 4 over IPv 6 の自動トンネル技術で，インターネットドラフトで規定されている．これまでとは逆に，移行期終盤で IPv 4 ホストが孤立した状況下での利用が想定されている．

DSTM による IPv 4 通信の際には，通信を行う IPv 6 ノード（IPv 6/IPv 4 デュアルスタック）は DHCPv 6 により一時的な IPv 4 アドレスやトンネルエンドポイントとなる DSTM ボーダルータのアドレス情報を取得し，トンネル設定を行う．

⑧　MPLS による IPv 6 パケットの中継

ここまで紹介してきた技術とはやや趣向が異なるが，既存のインフラを利用する IPv 6 ネットワーク構築方法として，MPLS（Multi Protocol Label Switching）バックボー

ンによるIPv 6パケットの中継を紹介する．

離れたIPv 6ネットワーク同士を相互接続させるのに既存のIPv 4ベースのMPLSバックボーンを共用し，IPv 6パケットの中継を行う．MPLS網内ではラベルに基づくフォワーディングが行われるため，IPv 6プロトコルに依存せず高速バックボーンを構築できる．ただし，MPLSバックボーンのエッジルータがIPv 6をサポートする様アップグレードされている必要がある．

3） トンネリング技術の用途

表3に各トンネリング技術の主な用途を示す．

表 3　トンネリング技術の主な用途

トンネリング技術	用　　途
① IPv4 互換 IPv6 による自動トンネル	実用性は低く，実験的な用途など利用範囲は限られる．使用頻度は低いと思われる．
② 6to4	IPv4 インターネットを通じて，IPv6 イントラネット同士やインターネット上のIPv6 ホストへ接続する場合に利用する．使用頻度は高いと思われる．
③ ISATAP	IPv4 プライベートアドレス運用のイントラネット内で，IPv6 ホスト同士を接続する場合に利用する．企業内部のネットワークを段階的に IPv6 へ展開する場合に有効．使用頻度は高いと思われる．
④ Teredo	IPv4 インターネットを通じて IPv6 ホストへ接続する際， NAT を越える必要がある場合に利用する．主に SOHO や個人ユーザの使用が対象とされる．
⑤ トンネルブローカ	IPv4 インターネットを通じて IPv6 ホストへ接続する場合に利用する．インターネット上に設置されているトンネルブローカとトンネルサーバを活用する．使用頻度は低いと思われる．
⑥ 6over4	イントラネット上で IPv6 ホスト同士を接続する場合に利用する．ただし IPv4 マルチキャスト環境のインフラが必要なため，同様のケースで使用される ISATAP の方が活用の機会が多いと推測される．使用頻度は低いと思われる．
⑦ DSTM	IPv6 インフラを利用して IPv4 ホスト同士を接続する場合に利用する．移行期終盤で多く活用されると思われる．
⑧ MPLS による IPv6 パケットの中継	離れた IPv6 ネットワーク同士を相互接続する高速バックボーンとして利用する．MPLS 技術を使用した IP-VPN サービスが，IPv6 接続により実現できる．

これまで述べてきた様に，本項で紹介してきた各トンネリング技術はそれぞれに最適な使用形態が想定されている．これらは推測される使用頻度も様々で，例えば6 over 4の様に，新しく開発されたトンネリング技術により，その有効性が低下しているトンネリング技術もある．そしてこれらのトンネリング技術の利用対象が大規模企業なのか小規模企業なのか，SOHOであるのか個人サイトの移行なのかによっても，その選択は異なってくる．

ここで比較的規模の大きな企業を対象とした場合において，IPv 6への移行時に採用されると思われるトンネリング技術の適用予測を述べる．既存のIPv 4ベースイントラネットはプライベートアドレス運用がされているため，イントラネット内で孤立したIPv 6ホスト同士の接続にはISATAPトンネルの利用が有効であると考えられる．またIPv 6対応のWebサーバへの接続は，IPv 4インターネットを介して6 to 4トンネルにより実施されるケースが増えると推測できる．この他，IPv 6による企業間接続を行う場合には，MPLS網によるIP-VPN通信の利用が進むものと思われる．

2.1.3　トランスレータ

IPv 6ホストとIPv 4ホストとの間で通信を行う場合，IPv 6とIPv 4には相互接続性がない

ため何らかの相互変換を施す必要がある．この様な状況下で利用されるのがトランスレータと呼ばれる技術である．本項では，トランスレータ技術の概要とその分類について述べる．

1)　概　　要

トランスレータは，IPv 6 通信と IPv 4 通信との相互接続において，その境界上でパケットの相互変換機能を提供するものである（図 6）．これにより，IPv 6 ホストと IPv 4 ホストとの通信が可能となる．

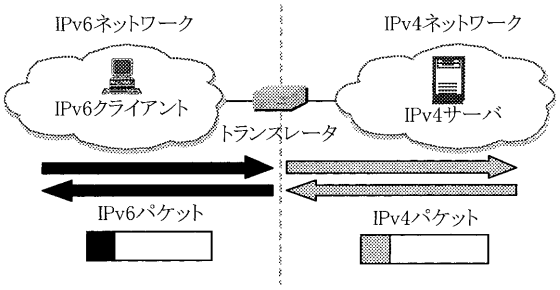

図 6　トランスレータによるパケット変換イメージ

■想定される利用ケース

IPv 6 への移行期には，IPv 6 と IPv 4 で相互変換した通信を実現させたい場面が想定される．つまり IPv 6 ホストから IPv 4 ホストへの接続を行うケース，及び IPv 4 ホストから IPv 6 ホストへの接続を行うケースである．前者の例としては，IPv 6 のみの搭載が予想される PDA や携帯電話が，既存の IPv 4 Web サーバに接続したい状況などが考えられる．また後者の例としては，IPv 4 ホストから IPv 6 対応の情報家電を操作する状況などが考えられる．この様な場面でトランスレータの利用が有効となる．

2)　トランスレータ技術の分類

トランスレータはプロトコル変換を行う階層により，「ヘッダ変換」「トランスポート層リレー」「アプリケーション層ゲートウェイ」の 3 種類に分類され，それぞれ利用範囲が異なる．

以下にこれら三つの分類の概要を述べる．

①　ヘッダ変換

ヘッダ変換は，IP レイヤにおいて IPv 6 ヘッダと IPv 4 ヘッダの置き換えを行うトランスレータである．ヘッダ変換には次の様な特徴がある．

・変換処理によるオーバーヘッドは小さい．
・IPv 6 アドレスと IPv 4 アドレスの対応付けのため，DNS の機能追加が必要である．
・FTP の様にアプリケーションレベルで IP アドレスやポート番号が埋め込まれている通信には適用できない．
・一部を除いて一般に ICMPv 4 から ICMPv 6 への変換には対応していない．

図 7 にヘッダ変換の階層モデルを示す．

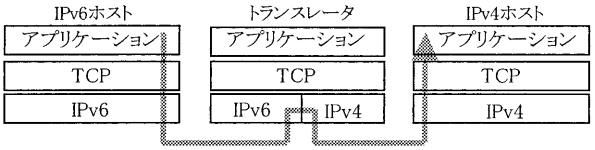

図 7　ヘッダ変換の階層モデル

ヘッダ変換によるトランスレータの実装例を表 4 に示す.

表 4　ヘッダ変換によるトランスレータの例

トランスレータ	概　　要
NAT-PT	Network Address Traslation-Protocol Translation の略. RFC2766 で定義されている. IPv4 で使用される NAT と同様の概念により, IPv6 アドレスと IPv4 アドレスの変換を行う.
SIIT	Stateless IP/ICMP Translation Algorithm の略. RFC2765 で定義されている. IPv4 ヘッダと IPv6 ヘッダの相互変換, または ICMPv4 ヘッダと ICMPv6 ヘッダの相互変換を行う.
TOWNS	TranslatOr With Name Server の略. ヘッダ変換の際, IPv6 と IPv4 の始点アドレス, 終点アドレスの対応付けを保持したネームサーバを利用する.
BIS	Bump In the Stack の略. RFC2767 で定義されている. IPv4 ベースのアプリケーションホスト内において, IPv4 プロトコルスタックとネットワークインタフェースドライバとの間に IPv4 と IPv6 を変換する機能を挿入し, 外部の IPv6 ホストからの IPv4 アプリケーション利用を可能にする技術.

② トランスポート層リレー

トランスポート層リレーは, トランスレータまで届いた TCP コネクションを一旦終端させ, 自身が代理として新たなコネクションを設定しデータの受け渡しを行うトランスレータである. トランスポート層リレーには次の様な特徴がある.

・変換処理によるオーバーヘッドはヘッダ変換に比べて大きい.
・IPv 6 アドレスと IPv 4 アドレスの対応付けのため, DNS の機能追加が必要である.
・コネクションの終了が明確な TCP でのみ利用可能である.

図 8 にトランスポート層リレーの階層モデルを示す.

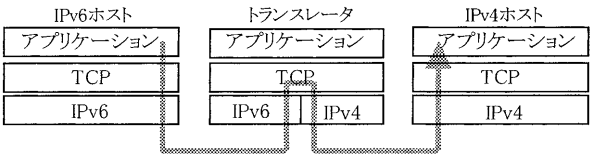

図 8　トランスポート層リレーの階層モデル

トランスポート層リレーによるトランスレータの実装例を表 5 に示す.

表 5　トランスポート層リレーによるトランスレータの例

トランスレータ	概　　要
FAITH	IPv4 ホストからの接続を FAITH ゲートウェイがコネクションを終端させ，ゲートウェイ上の代理サーバにより IPv6 側へコネクションを張りデータの受け渡しを行う．
SOCKS	RFC1928 で定義されている．SOCKS は，クライアントホストからの TCP 接続を SOCKS サーバが受け，接続先ホストとは TCP より接続し，SOCKS サーバの仲介による透過的な接続を行う．
TRT	Transport Relay Translator の略．RFC3142 で定義されている．IPv4 アドレスと IPv6 アドレスのマッピングにダミープレフィックスと呼ばれる擬似 IPv6 プレフィックスが用いられる．

③　アプリケーション層ゲートウェイ

アプリケーション層ゲートウェイは，自身がアプリケーションの代理サーバとして振る舞いデータの受け渡しを行うトランスレータ技術であり，クライアントホストからのコネクションをアプリケーション層で終端させる．クライアントホストは接続したいホストへ接続するのではなく，明示的にアプリケーション層ゲートウェイを接続相手として通信を行う．アプリケーション層ゲートウェイには次の様な特徴がある．

・変換処理によるオーバーヘッドはトランスポート層リレーに比べて大きい．
・アプリケーションのサービス毎にトランスレータが必要となる．
・IPv 6 アドレスと IPv 4 アドレスの対応付けが不要であり，DNS の変更が不要である．

図 9 にアプリケーション層ゲートウェイの階層モデルを示す．

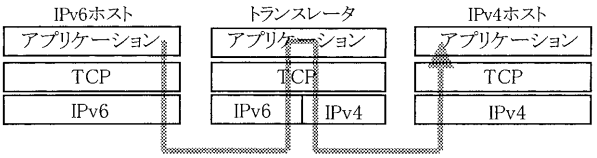

図 9　アプリケーション層ゲートウェイの階層モデル

アプリケーション層ゲートウェイによるトランスレータの実装例を表 6 に示す．

表 6　トランスポート層リレーによるトランスレータの例

トランスレータ	概　　要
SOCKS64	RFC3089 で定義されている．クライアントホストからの SOCKS コネクション要求を受けた SOCKS サーバは，クライアントからの FQDN（完全修飾ドメイン名）を元に接続先ホストへ IPv6 あるいは IPv4 でコネクションの確立を行う．この応答をクライアントホストへ応答する．
BIA	Bump-in-the-API の略．RFC3338 で定義されている．IPv4 ベースのアプリケーションホスト内において IPv4API と IPv6API を変換する機能を挿入し，外部の IPv6 ホストからの IPv4 アプリケーション利用を可能にする技術．
Squid	IPv6 対応の Web プロキシサーバ．

3)　トランスレータ技術のまとめ

表 7 に各トランスレータ技術の分類における主な特徴を示す．

表 7　トランスレータ技術の主な特徴

トランスレータ技術	変換処理	処理負荷	特　　徴
ヘッダ変換	IP レイヤでのヘッダ変換.	小さい.	DNS に変更が必要. 一般に ICMP の変換は対応しない. アプリケーション層で IP アドレスを意識するプロトコルには対応しない.
トランスポート層リレー	トランスポート層でクライアントからの TCP コネクションを終端し，新たにコネクションを設定する.	大きい.	DNS に変更が必要. TCP コネクションのみ適用可能.
アプリケーション層ゲートウェイ	アプリケーション層でクライアントからのコネクションを終端し，代理サーバとして動作する.	大きい.	適用するアプリケーションのサービス毎に実装が必要.

トランスレータはその分類によりそれぞれ一長一短があり，適用するネットワークやアプリケーション，端末の特性などを考慮した選択と構築が必要である．例えば，トランスレータがボトルネックとなることを軽減させたい状況においてはヘッダ変換によるトランスレータを採用し，また特定のアプリケーションに最適化した用途としてトランスレータを適用させるケースではアプリケーション層ゲートウェイによるトランスレータを採用することが考えられる．

2.1.4　ネットワークインフラの移行技術のまとめ

ここまで紹介した「デュアルスタック」「トンネリング」「トランスレータ」についてのまとめを表 8 に示す．

表 8　ネットワークインフラの移行技術のまとめ

移行技術	概　　要
デュアルスタック	既存ネットワークを IPv6 へ移行する際の基本となる技術である．移行期には，多くのホストやネットワーク機器が IPv6/IPv4 デュアルスタック化されていると思われる．ただし，両プロトコルはそれぞれ独立して利用され，相互変換性は持っていない.
トンネリング	IPv6 ネットワークを構築する上で有効となる技術で，既存の IPv4 インフラを活用できる．移行期には，多くのトンネリングが利用されると思われる.
トランスレータ	トラフィックを IPv6 と IPv4 で変換させたい状況で利用される技術である．デュアルスタックの実装が困難なホストが異なるプロトコルによるホストとの通信を実現させたいケースで有効となる.

IPv 6 への移行時には，これらの移行技術が組み合わされて IPv 6,IPv 4 混在のネットワークが構成されると考えられる．この内，デュアルスタック技術とトンネリング技術は既存の IPv 4 ネットワークへ影響をあまり与えずに移行を進める上で有効であると思われる．ただしこれらの技術は，IPv 6 と IPv 4 の両ネットワーク間を相互変換して利用するものではなく，それぞれ独立のプロトコルとしてサービスを活用することになる．IPv 6 と IPv 4 との間でプロトコルの相互変換を行う際はトランスレータ技術が利用される．特にデュアルスタック化が困難な状況にある既存のホストや，携帯電話，情報家電など IPv 6 シングルスタックが実装されると思われるホストが,異なるプロトコルで運用されているホストと接続したい状況においては,

トランスレータによるプロトコル変換の実施が有効と思われる．

2.2 アプリケーションの移行

IPv 6 を扱う環境へ移行させるのに伴い，アプリケーションについても IPv 6 化への対応が必要となる場合がある．本節では，各 IPv 4 ホストを IPv 6 化させる際に行われるプラットフォームとアプリケーションの IPv 6 対応について述べる．なお，ここで述べるアプリケーションの IPv 6 対応とは，ソースコードを持つアプリケーションに対して行う IPv 6 対応のことである．プロダクトアプリケーションの様なバイナリコード形式のアプリケーションに対する IPv 6 対応については，開発元へ確認する必要がある．

2.2.1 プラットフォームの IPv 6 対応

IPv 6 環境の構築においては，ネットワークインフラの IPv 6 化と並行して各ホストも IPv 6 を扱える状態にする必要がある．また，IPv 6 対応のアプリケーションを使用するにあたっても，動作環境であるプラットフォームが IPv 6 を処理する仕組みを持ち，IPv 6 を扱えることが前提となる．従ってアプリケーションの IPv 6 化を行う前には，プラットフォームの IPv 6 化を実施することになる．

■実装状況

現在，各 OS の最新版には IPv 6 スタックが組み込まれており，IPv 6 に対応済みである．ただしその多くはデフォルトで IPv 6 が無効化されている．従って OS が IPv 6 を扱える状態とするためには，IPv 6 を有効化させる何らかの処置が必要となる場合が多い．表 9 に主な OS の IPv 6 実装状況を示す．なお現在は，ほとんどの OS が IPv 6 を実装させることにより IPv 6/IPv 4 デュアルスタックホストとして機能し，IPv 6 シングルスタックホストの構成が可能となる OS は少ない．

表 9　主な OS の IPv 6 実装状況

OS	IPv6 実装状況
Linux	Kernel 2.2 より IPv6 スタックが組み込まれている． コマンドにより IPv6 を有効化する．コマンドはディストリビューションにより異なる． （あるいは，USAGI Project より提供されている IPv6 スタックを適用する．）
Solaris	Solaris7 より IPv6 スタックに対応している． インストール時の設定で使用可能．（インストール後に IPv6 化する場合は，touch コマンドによりインタフェース名のファイルを作成し IPv6 を有効化する．）
BSD 系 UNIX	KAME Project の IPv6 スタックが組み込まれている．デフォルトで有効．
HP-UX	デフォルトでは IPv6 がサポートされていない． HP 社より提供されている IPv6 パッチを適用する．
WindowsXP	IPv6 スタックが組み込まれている． コマンドプロンプトから ipv6 install コマンドを実行して IPv6 機能を有効にする．
Windows2000	Microsoft 社より提供されている IPv6 スタックを導入することで使用可能．

2.2.2 アプリケーションの IPv 6 対応

プラットフォームを IPv 6 化しても，既存のアプリケーションは，そのままでは IPv 6 を利用することができない．IPv 4 と IPv 6 ではソケット仕様が異なるためで，IPv 4 対応のアプリケーションを IPv 6 対応とさせるには，既存の IPv 4 プロトコルによるソケット API（Application Program Interface）から IPv 6 プロトコルによるソケット API を使用する様にソースコードを修正する必要がある．

1） IPv 6 のソケット API 定義

IPv 6 のソケット API について，RFC 3493 と RFC 2292 で定義されている．RFC 3493 では IPv 6 の基本的なソケットインタフェースの拡張について記述され，RFC 2292 では IPv 6 に特有な機能を使用するためのソケット API について記述されている．RFC 3493 により規定されるソケット API を使用するアプリケーションは，IP レベルより下層のプロトコルに依存せずに IPv 6 と IPv 4 の両方をサポートする．

2） IPv 4 から IPv 6 へ対応するためのソースコードの修正

アプリケーションのソースコードにおけるソケット API の呼び出しは，一般に IPv 4 と IPv 6 で対応関係が存在する．このため IPv 4 から IPv 6 へ対応するためのソースコード修正は，多くの場合ソケット API の呼び出しが行われている箇所に的が絞られる．

また，ソースコードより修正箇所を判別する「修正支援ツール」が提供されており，ソースコードを IPv 6 化修正する際はこの様なツールの利用も有効となる．このツールは，ソースコードをスキャンしソケット API の呼び出しを行っているモジュールを判別するものである．

以下にツールの例を示す．

・**Checkv 4.exe**　Windows 環境で実行するアプリケーションのソースコードをスキャンし，IPv 6 対応への修正が必要な行番号と変更内容を出力する．Microsoft 社より提供されており，Windows Platform SDK に含まれている．

・**Socket Code Scrubber**　Solaris 環境で実行するアプリケーションのソースコードをスキャンし，IPv 6 対応への修正が必要な行番号とその行を出力する．Sun Microsystems 社より提供されている．

ホストの IPv 6 対応の流れを図 10 に示す．

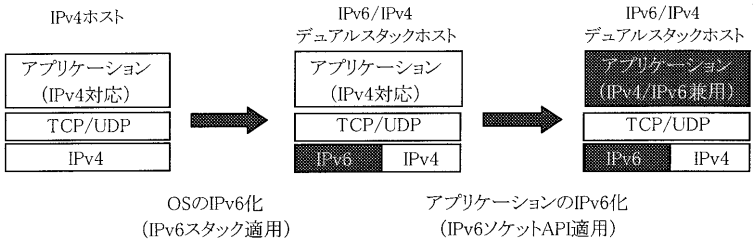

図 10　ホストの IPv 6 対応の流れ

3.　IPv 6 実証実験報告

本章では，2002 年度に日本ユニシス（以下，当社）内で実施した IPv 6 に関する実証実験について報告する．

3.1　概　　要

本実験は 2002 年度に当社内で行われたユビキタス実証実験の一環として，IPv 6 に関する技術検証をテーマに行われたものである．本実験の目的とテーマについて述べる．

1） 目　　的

本実験は，IPv 6 と IPv 4 の混在ネットワーク環境を想定した移行技術の検証，及び IPv

6 ネットワーク環境の構築を見据えた技術の検証を実施することにより，これら技術のまとめと課題・問題点の洗い出しを目的として行われた．

2） テーマ

本実験ではテーマを次の四つに分け，それぞれのテーマ毎に検証を実施した．

（a） IPv 6 のアドレッシングに関する検証

（b） IPv 4 から IPv 6 への移行技術に関する検証（6 to 4 トンネリング実験）

（c） IPv 6 Socket プログラミングに関する検証

（d） IPv 6 上の Xcast 6[*4] プロトコルを使用した電子会議システムに関する検証

3.2 実 験 報 告

本節では実験の環境構築，実験方法，考察について述べる．ただし，ここでは実施した四つのテーマの内，本稿の主題と関係する「（b） IPv 4 から IPv 6 への移行技術に関する検証」と「（c） IPv 6 Socket プログラミングに関する検証」を扱うこととする．

3.2.1 環 境 構 築

本実験で行った環境の構築について主な実施項目を述べる．（環境構成については，図 11 を参照．）

- 各ホスト（Solaris 8,Linux,WindowsXP）に IPv 6 化を施す．（デュアルスタックホスト化）
- ルータ（Cisco 2611）に IPv 6 化を施す．（デュアルスタックルータ化）
- 実験のベースとなる基本環境を構築する．
- IPv 4 インターネット接続環境を構築する．
- トンネルエンドポイントノードとするデュアルスタックホストに 6 to 4 トンネル設定を施す．
- トンネルエンドポイントノードとするデュアルスタックルータに 6 to 4 トンネル設定を施す．

3.2.2 実 験 方 法

本項では，それぞれの実験方法とその結果について述べる．初めに「IPv 4 から IPv 6 への移行技術に関する検証」について述べ，続いて「IPv 6 Socket プログラミングに関する検証」について述べる．

1） IPv 4 から IPv 6 への移行技術に関する検証（6 to 4 トンネリング実験）

本実験では，6 to 4 トンネリングを構築することにより，IPv 4 インフラを通した IPv 6 ホスト同士の接続性を検証した．接続性の確認は，ping 6（IPv 6 対応の ping ツール），telnet,http アクセスを実施することにより行った．

なお，本実験は接続対象の IPv 6 ホストにより，トンネル構成を次の 3 種類に分けて検証した．

① IPv 4 ネットワーク上のデュアルスタックホストと IPv 6 ネットワーク上の IPv 6 ホストの接続

この場合，トンネルエンドポイントとなるのは，デュアルスタックルータと IPv 4 ネットワーク上のデュアルスタックホストである．この構成は，ホスト to ルータ，及び

ルータ to ホストとなる．

② IPv 4 ネットワーク上のデュアルスタックホストと IPv 6 インターネット上の IPv 6 ホストの接続

この場合，トンネルエンドポイントとなるのは，IPv 4 ネットワーク上のデュアルスタックホストと IPv 4 インターネット上の 6 to 4 リレールータである．この構成は，ホスト to ルータ，及びルータ to ホストとなる．なお，本実験はインターネット上に公開されている 6 to 4 リレールータ[*5] を利用して行った．

③ IPv 6 ネットワーク上の IPv 6 ホストと IPv 6 インターネット上の IPv 6 ホストの接続

この場合，トンネルエンドポイントとなるのは，デュアルスタックルータと IPv 4 インターネット上の 6 to 4 リレールータである．この構成は，ルータ to ルータとなる．なお，本実験はインターネット上に公開されている 6 to 4 リレールータ[*6] を利用して行った．

①〜③の実験構成概要を図 11 に示す．

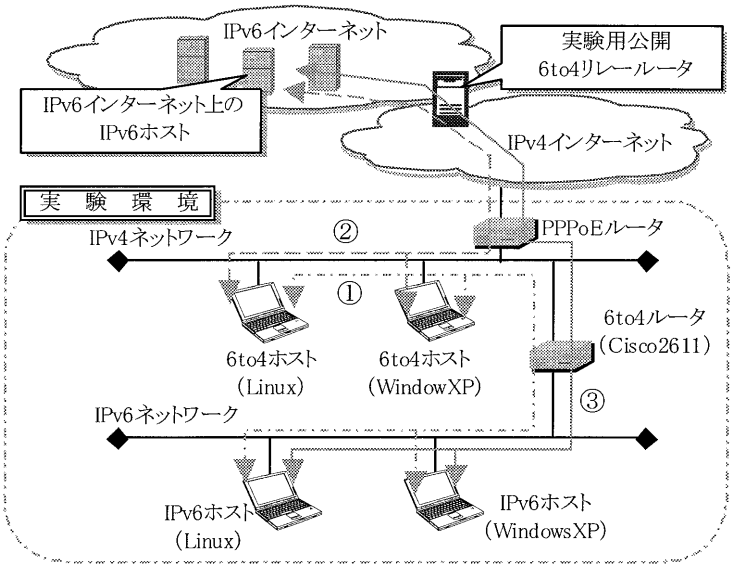

図 11　6 to 4 トンネリング実験構成概要

■結　　果

①〜③の各トンネル構成における実験の結果，それぞれ IPv 6 ホスト同士の接続が正常に行える事を確認した．本実験から，実験で用いた Linux，WindowsXP，Cisco ルータ（Cisco 2611）について，現状で既に 6 to 4 トンネルの実装に対応済みであることが検証できた．

2） IPv 6 Socket プログラミングに関する検証

本実験は，C 言語，及び JAVA による Socket プログラミングにおいて，IPv 6 対応と IPv 4 対応でのプログラミングの違いの検証と，その違いを吸収させる処置の有効性についての検証を実施した．なお，実験は次の 2 項目について行った．

① C 言語におけるプロトコル間相違の検証

本実験は，C 言語により WindowsXP 環境で動作する HTTP クライアントプログラムを作成し，その IPv 4 及び IPv 6 対応時のプログラミング上の差異について検証した．検証は以下の通り実施した．

- IPv 4 で動作する HTTP クライアントプログラムを C 言語により作成した．（実行環境は WindowsXP）
- IPv 4 対応プログラムから IPv 6 対応へ修正する際，ソースコードから必要な修正箇所を判別するツールが提供されている．このツールを用いて，作成した C プログラムの IPv 6 と IPv 4 でのソースコードの違いを検証した．（本実験では，当ツールとして Microsoft 社より提供されている「Checkv 4.exe」を使用した．）
- ツールにより判明したソースコードの違いを修正し，IPv 6 対応とした．
- IPv 6 対応とした HTTP クライアントプログラムを起動させ，予め構築しておいた IPv 6 対応の Web サーバへ接続し，その動作よりツールによるソースコード修正の精度や実用性を検証した．

② JAVA におけるプロトコル間相違の検証

本実験は，JAVA により Solaris 8 環境で動作する HTTP プロキシサーバプログラムを作成し，その IPv 4 及び IPv 6 対応時のプログラミング上の差異について検証した．検証は以下の通り実施した．

- IPv 4 で動作する HTTP プロキシサーバプログラムを JAVA により作成した．（実行環境は Solaris 8）
- ①同様，修正支援ツールにより作成した JAVA プログラムの IPv 6 と IPv 4 でのソースコードの違いを検証した．（本実験では，当ツールとして Sun　Microsystems 社より提供されている「Socket Code Scrubber」を使用した．）
- ツールにより判明したソースコードの違いを修正し，IPv 6 対応とした．
- IPv 6 対応とした HTTP プロキシサーバプログラムを起動させ，HTTP クライアントからの Web サーバへのアクセス要求に対し，取得した HTML ファイルをクライアントへ返す動作により，ツールによるソースコード修正の精度や実用性を検証した．なお，HTTP クライアントとして①で作成したプログラムを使用した．

①，②の実験構成概要を図 12 に示す．

■結　　果

①，②の IPv 6　Socket プログラミング実験の結果，本実験で作成したプログラミングに限られるが，IPv 4 対応プログラミングを IPv 6 対応とする際，修正箇所の判別ツールの利用が有効であることが確認できた．本ツールによるプログラミング修正は，IPv 6 による使用において想定どおり動作するものであった．

3.2.3 考　　察

本項では，実験実施により IPv 6 環境の構築において気づいた事項，及び IPv 6 と IPv 4 との混在環境を見据えた際の特筆事項を述べる．

- 各ホストの IPv 6 化（デュアルスタック化）は，各種 OS 共に比較的新しいバージョンであれば，何らかの形でサポートされている．そして特に難しい操作を要するものでも

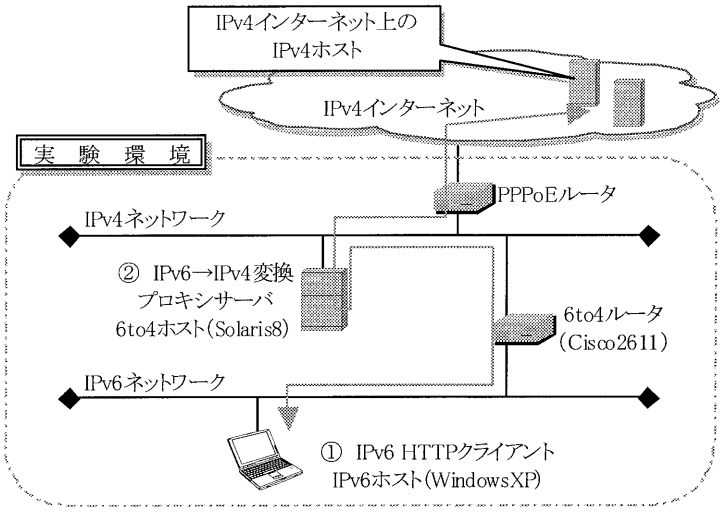

図 12　IPv 6 Socket プログラミング実験構成概要

ない．ただし，操作において OS 毎の基礎知識は必要である．

- ネットワーク機器の IPv 6 化（デュアルスタック化）を実施する際には，機種，OS の種類とバージョン，搭載メモリ容量が関わるため注意が必要である．また実現させる IPv 6 関連機能（例えばトンネリング機能やトランスレータ機能など）によっても要求スペックが異なるのため，使用する機器について十分な調査が必要である．
- 本実験で実施した 6 to 4 トンネリング機能は，IPv 4 との混在環境においても実用可能なレベルであった．
- トランスレータ機能等，一部の IPv 6 関連技術については DNS に特殊な機能を要求するものもあるため，注意が必要である．
- アプリケーションの IPv 6 対応については，既存の IPv 4 対応アプリケーションのソースコードから必要な修正箇所を判別するツールが公開されている．実験では，ツールの利用が有効であった．（本実験では，Windows 環境用の「Checkv 4.exe」と Solaris 環境用の「Socket Code Scrubber」を使用した．）
- HTTP サーバ，DNS サーバ，SMTP サーバなどインターネットサービスに関連するアプリケーションの多くは，IPv 6 への対応が比較的進んでいる．

4. お わ り に

IPv 6 は，標準化が開始されてから既に 10 年以上が経ち，技術的な設計も終えつつある．IPv 4 からの移行についても技術的な問題はほぼ解消されてきており，移行準備も整備が進んでいる．

ところが企業内ネットワークにおいては，現在，ほとんどの企業で IPv 6 化が進んでいない．その理由の一つには，現状の IPv 4 環境で何ら不自由なく運用しており，むしろ IPv 6 へ移行する際に付随するコストやリスクといったデメリットがあるため，IPv 6 移行に対する緊迫性が少ないという点が考えられる．確かにプライベートアドレスの使用により実用上はアドレス不足に駆られることは稀であり，またインターネットの利用もその多くが Web アクセスやメ

ール利用によるものである場合，IPv 4 でも運用が可能と思われる．

多くの企業が社内ネットワークを IPv 6 へ移行し始めるのは，インターネットの IPv 6 化が十分整備されユビキタス環境が一般でも実用され始める時期になると考えられる．この時期には企業でもインターネットの IPv 6 リソースを利用する有用性が増し，企業内で各端末のデュアルスタック化が進むものと思われる．更にユビキタス環境の社内における適用も進み，例えば自律的なセンサー同士のネットワークに IPv 6 が適用されるなど，IPv 6 ネットワークが既存の IPv 4 ネットワークと併用されていると推測される．その頃は，本稿で述べてきた移行技術・共存技術が広く使われているであろう．そして IPv 4 の利用は次第に縮小していくとされる．

総務省の情報通信審議会は，平成 14 年 8 月付けの答申資料“「21 世紀におけるインターネット政策の在り方」についての第 2 次中間答申”にて，IPv 6 の本格普及時期が 2004 年頃より始まるというシナリオを掲げている．そこで，本格的に IPv 6 へ移行が始まるまでの今後の主な課題を挙げると，「IPv 6 対応アプリケーションの充実」，「IPv 6 機能を十分活かしたアプリケーションや適用技術・サービスの開発」，「ネットワーク管理製品やセキュリティ管理製品の IPv 6 対応」，「IPv 6 ネットワークに適応した運用ポリシー/セキュリティーポリシーの確立」などであると思われる．これらを検討し解消することが，スムーズな IPv 6 への移行と IPv 6 ネットワークの発展に必要な要素になると考える．

＊ 1　End-to-End 通信：通信する末端の機器同士間で完結する通信のこと．
＊ 2　トンネルエンドポイント：カプセリングやデカプセリングを行うノードのこと．トンネル始点やトンネル終点にあたる．
＊ 3　IPv 6 サイト：ここでは IPv 4 インターネット，または IPv 6 インターネット上の IPv 6 イントラネットを意味する．
＊ 4　Xcast 6：インターネットドラフトにて規定中の新しいマルチキャスト方式．送信対象となる複数の宛て先を明示的に IPv 6 拡張ヘッダに格納し配信する．
＊ 5, ＊6　公開されている 6 to 4 リレールータ：株式会社 KDDI 研究所が実験目的で 6 to 4 リレールータを公開している．公開期間は 2002 年 3 月 1 日から 2004 年 3 月 31 日．（http : //www.6 to 4.jp/）

参考文献
［1］ 萩原吉弘，IPv 6 の詳細解説と実践導入法，株式会社ソフト・リサーチ・センター，2002.9
［2］ 江崎浩，関谷勇司，吉藤英明，石原知洋，詳説図解　IPv 6　エキスパートガイド，株式会社秀和システム，2002.5
［3］ Christian Huitema, IPv 6　次世代インターネット・プロトコル, 株式会社ピアソン・エデュケーション，1997.2
［4］ IPv 6 magazine 編集委員会，IPv 6 magazine No.4 Winter 2003，株式会社インプレス，2003.2
［5］ IETF, http://www.ietf.org/home.html
［6］ ipv 6.org,http://www.ipv 6.org/
［7］ jp. ipv 6.org, http://www.jp.ipv 6.org/
［8］ e-Japan 重点計画 2002,
http://www.kantei.go.jp/jp/singi/it 2/kettei/020618 honbun.pdf
［9］ 総務省，http://www.soumu.go.jp/

執筆者紹介　小宮山智之（Tomoyuki Komiyama）
1999 年北里大学理学部物理学科卒業．同年日本ユニシス(株)入社．ネットワーク構築業務やネットワークセキュリティ関連業務に従事．現在，アドバンスト・テクノロジ本部　IT 統括部　ユビキタスコンピューティング部に所属．