IPv6 Everywhere: IPv6 接続のための適応的トンネル構成機構
IPv6 Everywhere: The Adaptive Configuration of Connectivity to IPv6 Networks

中野悦史 [*]
Etsushi Nakano
ets@ubi.cs.ritsumei.ac.jp

河口信夫 [†]
Nobuo Kawaguchi
kawaguti@nagoya-u.jp

西尾信彦 [‡]
Nobuhiko Nishio
nishio@cs.ritsumei.ac.jp

概要

IPv6 のネットワークインフラはまだ不十分であり，ノードが IPv6 の接続性を得るためには IPv6 over IPv4 トンネル接続技術を利用する必要がある．しかし，既存の技術では IPv6 グローバルネットワークへの接続性を得られない環境が存在する．我々は，そのような環境においても利用可能で，ノードおよびネットワークに接続性を提供する IPv6 over TCP/UDP の接続技術 CAULIS を設計した．また，それぞれの接続技術の特徴を踏まえ，最適な接続技術をノードに設定するシステムを設計した．本稿ではそれらの設計とプロトタイプの実装について述べる．

1. はじめに

近年 IPv6 への期待が高まりつつある．IPv4 グローバルアドレスの枯渇 [1] や P2P アプリケーションの需要の増加などが主な理由に挙げられるが，ネットワーク対応の家電/AV 機器の増加やオンラインゲームの増加に伴う，知識の少ないユーザがネットワークを利用する機会が増加していることも，この風潮を後押ししている．

IPv6 は既に多くの OS でサポートされており，対応の比較的遅かった Windows においても，次期バージョンである Windows Vista において IPv6 を IPv4 より優先的に利用することが発表 [3] され，ようやくコンシューマレベルで IPv6 の利用が加速する土台ができあがってきた．

しかし，それは OS での話であり，現在でも IPv6 ネットワークインフラは十分と言えない状況にある．現状，日本国内において，IPv6 グローバルネットワークへの接続性を IPv4 と同様のコストで得られる例は殆ど見られない．ホットスポットや学内 LAN のような公共のネットワークでは尚更である．そのため，IPv4 ネットワークを利用して IPv6 グローバルネットワークへの接続性を得る IPv6 over IPv4 トンネル接続技術を利用する必要がある．しかし，このような技術を利用するためには，豊富なネットワークの知識を有するか，ISP に特別なサービスを受けるためのコストを支払う必要がある．このような問題は IPv6 アプリケーションの開発/利用の上でも障害となる．

我々は既存の代表的なトンネル接続技術それぞれの特徴を考察した結果，それらの接続技術では IPv6 の接続性を得られないネットワーク環境が存在することを明らかとした．本研究ではそのような環境でも利用可能な接続技術「CAULIS」を設計し，プロトタイプを実装した．また同時に，様々なネットワーク環境で，例え知識がなくとも最適な IPv6 の接続性を得ることを可能とする，適応的なトンネル構成機構を設計し開発した．この 2 つの技術により，あらゆる環境から最適な接続技術で IPv6 グローバルネットワークへの接続性を得ることを可能とする．

本稿では，既存の代表的なトンネル接続技術と CAULIS について特徴を述べ，それらの長所短所を考察した上で，それらをどのように効果的に利用するかを明らかにする．また，CAULIS および適応的構成を可能とするシステムについて，その設計とプロトタイプの実装について述べる．

本論文は全 6 章から構成される．2 章では既存の代表的なトンネル接続技術について述べ，考察し，利点と欠点を明らかとする．3 章では CAULIS の設計を述べ，続く 4 章で適応的に接続技術を構成するシステムについて設計を述べる．5 章ではそれらのシステムの実装と評価について述べ，最後に 6 章で本論文をまとめる．

なお，本稿において「ネイティブの IPv6」は非 IPv6 over IPv4 トンネル接続の IPv6 を意味し，「NAT」は差し障りない限り NAPT を意味する．

2. 既存の IPv6 over IPv4 技術

本章では，既存の IPv6 over IPv4 トンネリング技術の中から，基本である静的トンネリングおよび代表的な自動構成トンネリング技術について，その特徴と利点/欠点を述べる．そして，それらが利用できる環境と状況を明らかにし，利用可能なネットワーク環境を明らかにする．

2.1 静的トンネリング

IPv6 パケットに IPv4 のヘッダを付けて送信する，IPv6 over IPv4 の基本的なトンネル接続手法である．

静的トンネリングは，接続するルータ (ノード) 双方からトンネル通信先を互いに指定し，お互いのルータへパケットを転送するトンネルインタフェースを仮想的に定義する．そのインタフェースを利用して，パケットの中継や経路制御を行う．アドレス規則に特に制限はない．

2.1.1 利点

利用するプレフィクスに特に制限がないため，ノードに IPv6 の接続性を与えたり，ルータとして構成可能にしたりと，比較的自由なネットワーク構成が可能である．

[*] 立命館大学大学院理工学研究科
Graduate School of Science and Engineering, Ritsumeikan University
滋賀県草津市野路東 1-1-1
[†] 名古屋大学大学院工学研究科/情報連携基盤センター
Graduate School of Engineering, Nagoya University
名古屋市千種区不老町 1 名古屋大学大学院工学研究科
[‡] 立命館大学情報理工学部
Department of Computer Science, Ritsumeikan University
滋賀県草津市野路東 1-1-1
本研究は総務省戦略的情報通信研究開発推進制度 (SCOPE)「異種スマート環境間をセキュアに動的接続・構成する基盤技術」の支援を受けて実施されている．

2.1.2 欠点

ルーティング先が静的であるため通信効率が良いとはいえない．また，トンネルの構成にあたっては双方から明示的にトンネリング先を指定する必要があり，利用するノードには固定された IPv4 アドレスが必要である．つまり，アドレスが動的な環境では利用が難しい．そのため，DTCP(Dynamic Tunnel Configuration Protocol)[4] のように，ノード間で認証を行いトンネルを自動構成する技術で利用することが望ましい．

なお，一般的な静的トンネリングでは IPv4 プロトコル 41 番 (ipv6) あるいは GRE(47 番) を利用している．これらのプロトコルは NAT の内側から使うことが困難である．

2.2 6to4

6to4 は，自動構成トンネリングを含む IPv4 ネットワークから IPv6 ネットワークへの接続技術である [5]．6to4 を利用する IPv4 ノードは，IPv4/IPv6 ネットワーク両方への接続性を持ったリレールータ (6to4 relay) を利用してネイティブな IPv6 ネットワーク上の IPv6 ホストへ通信を行うことができる．

2.2.1 アドレス規則

6to4 には 2002::/16 の TLA が割り当てられており，各ノードは自身の IPv4 アドレスを埋め込んだ 2002:IPv4ADDR::/48 のプレフィクスを利用する (図 1)．これにより，IPv4 ネットワーク全てを 2002::/16 という巨大な TLA と見なすことができる．このアドレス構成が，自動構成トンネリングを可能とする仕組みである．

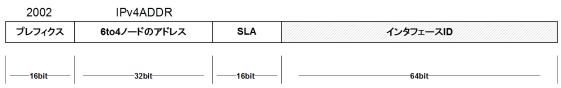

図 1: 6to4 のアドレス

2.2.2 ルーティング

6to4 の通信におけるパケットのルーティングは，大きく 2 つに分けられる．

■6to4 ノードから IPv6 ホストへの通信

6to4 ノード (6to4 利用ノード) から IPv6 ホストへの通信は，行きと帰りで異なる経路を辿る (図 2)．

行きの IPv4 → IPv6 の通信では，指定した 6to4 relay に対するトンネルが構成される．6to4 relay は IPv6 パケットのカプセルを外し，宛先の IPv6 ホストへ転送する．この 6to4 relay の指定において IPv4 エニーキャストアドレス 192.88.99.1 を利用することが可能であり，効率の良い通信経路が選択される．

帰りの IPv6 → IPv4 の通信は，IPv6 ホストのネットワークポリシに基づいて 2002::/16 のルーティングで自然に選ばれた 6to4 relay を経由して行われる．6to4 relay に到達したパケットは，その宛先 IPv6 アドレスに含まれる IPv4 アドレスを基に再び IPv4 パケットにカプセリングされ，自動構成されたトンネルを通じて 6to4 ノードに到達する．

■6to4 ノードから 6to4 ノードへの通信

6to4 ノード間の通信は，より効率的である．

宛先アドレスが 6to4 のアドレスであれば，宛先 IPv6 アドレスに含まれる IPv4 アドレスを基に IPv4 のヘッダを付加し，宛先の 6to4 ノードに送信する．このとき 6to4 relay を経由する必要はないため，IPv4 のルーティングとほぼ同性能の通信が期待できる．

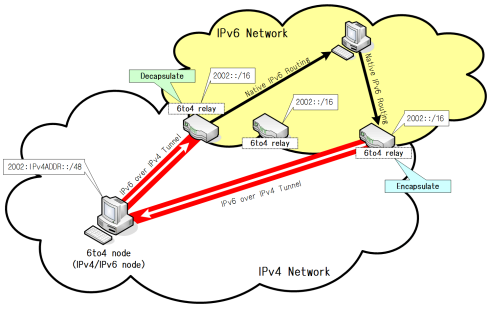

図 2: 6to4 ノードと IPv6 ホストの通信

2.2.3 利点

2002:IPv4ADDR::/48 という広大なアドレス空間を提供することが利点である．イントラネットのエッジノードが 6to4 を設定しルータとなるだけで，そのイントラネットに IPv6 の接続性を提供できる．

また，効率的なルーティングも利点といえる．6to4 relay が分布すればするほど IPv6 ホストとの通信性能は向上する．また，6to4 ノードおよび 6to4 プレフィクスを持つノードとの通信は，IPv4 の直接通信と殆ど変わらないため，非常に効率的である．

2.2.4 欠点

欠点は，IPv4 パケットへのカプセル化に IPv4 プロトコル 41 番 (ipv6) を使用することである．このプロトコルのパケットはポート番号がないため，NAT の内側から同時利用することができない．そもそも 6to4 のアドレス規則も，1 つの IPv4 グローバルアドレスで複数のノードが利用することを可能としていない．さらには，NAT によってはフォワードを設定できないことも多々ある．

これらの問題から，プライベートアドレスしか持たないノードが 6to4 を利用することは難しいといえる．利用にあたっては基本的に IPv4 のグローバルアドレスが必要で，ネットワーク管理者が利用する技術ともいえる．

2.3 ISATAP

ISATAP は，サイト内で IPv6 の接続性のない IPv4/IPv6 デュアルスタックノード間で，自動設定トンネルを構成する技術である [6].

2.3.1 アドレス規則

ISATAP のアドレスは，上位 64bit に ISATAP ルータが所持するグローバルに一意なプレフィクスを埋め込む．インタフェース ID は，基本的には ISATAP の識別子 0000:5efe とノードの IPv4 アドレスである．この IPv4 アドレスはプライベートアドレスでよい．ISATAP ルータが同一サイトに存在し，IPv6 グローバルアドレスを有した IPv6 ルータであれば，ISATAP ノードも IPv6 グローバルアドレスを得ることができる．

2.3.2 ルーティング

ISATAP のルーティングは 2 つに分けられる．(図 3).

■同一のプレフィクスを持つノード宛の通信

同一の ISATAP ルータを利用するノードであるため，同一サイト内のノードであり IPv4 の直接の通信が可能である．したがって宛先ノードの IPv4 アドレス (ISATAP アドレスに含まれている) を宛先とする IPv4 ヘッダを IPv6 パケットに付加し，直接ノードに送信する．

■異なるプレフィクスを持つノード宛の通信

ISATAP ルータの IPv4 アドレスを宛先とした IPv4 ヘッダ

でカプセル化し，ISATAP ルータ宛に送信する．ISATAP ルータはそのパケットを受け取り，IPv4 ヘッダを取り除いて通常の IPv6 パケットとし，宛先に転送する．ISATAP ルータから先の通信方法は，ネイティブでも 6to4 のようなトンネル接続でも良い．

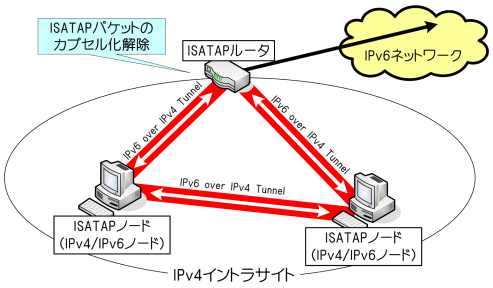

図 3: ISATAP のルーティング

2.3.3 利点

ISATAP は，上位 64bit がいかなるプレフィクスであっても利用できる点に特徴がある．したがって，ISATAP ルータが IPv6 ネイティブの接続性を持たず 6to4 を利用していても，ISATAP ノードは IPv6 グローバルアドレスを得ることが可能であり，グローバルネットワークへの接続が可能である．トンネル接続からネイティブへの移行も，ルータへの変更だけで良い．

また，基本的に ISATAP は自動設定の仕組みを有していることも利点である．この自動設定を有効にするために，”isatap.[domainname]”([domainname] はサイトのドメイン名) という FQDN を ISATAP ルータに割り当てることを推奨している．このホスト名のルックアップにより，自動的に ISATAP ルータと通信が行われ，プレフィクスが取得され，IPv6 のグローバルアドレスを取得することができる．

2.3.4 欠点

エンドユーザがノードに ISATAP を設定したところで，必ずしも IPv6 の接続性を得られるものではないことが欠点である．ノードが ISATAP により IPv6 の接続性を得られるかどうかは，IPv6 グローバルネットワークへ接続している ISATAP ルータが存在するかどうかによる．ISATAP ルータが存在しないネットワーク環境では，IPv6 の接続性を得ることはできない．

つまり ISATAP は，ユーザが各ノードに設定して IPv6 の接続性を得るものというよりは，イントラネットの IPv6 利用を容易に行える技術と捉えるのが正しい．

2.4 Teredo (Shipworm)

Microsoft 社が提唱する，STUN[7] と呼ばれる NAT 越えを含む IPv6 over UDP/IPv4 の技術である [8]．UDP を利用したトンネリング手法は他にも存在するが，運用に至っているものはごく少数であり，最も有力な IPv6 over UDP/IPv4 の標準はこの Teredo である．

この技術は，Windows Vista で標準で有効化されることが発表がされている [9]．

2.4.1 アドレス規則

Teredo クライアントは指定した Teredo サーバと通信を繰り返すことでアドレスを生成する．

Teredo アドレスは，識別子である 2001:0000/32 のプレフィクスを持つ．上位 64bit の後半 32bit には Teredo サーバの IPv4 アドレスが埋め込まれ，インタフェース ID にクライアントの利用ポート番号や IP アドレスといった情報が埋め込まれる．このアドレス構造を図 4 に示す．

図 4: Teredo のアドレス

なお，現在の Windows XP SP1(SP2) の Teredo の実装により構成される IPv6 アドレスは，実験時に利用されていた 3ffe:831f/32 という暫定的なプレフィクスである．

2.4.2 ルーティング

Teredo のルーティングは 6to4 と非常に似ている．しかし，6to4 以上に無駄の少ないルーティングを行っており，UDP を利用することでわずかに劣る転送効率を補っている．

■IPv6 ホストとの通信

Teredo クライアントと IPv6 ホストとの通信においては，宛先 IPv6 ホストごとに異なる Teredo relay(IPv6/IPv4 境界ルータ) を利用する仕組みを有している．

Teredo クライアントが明示的に指定した Teredo サーバを利用するのは，最初の Teredo relay の選択のためだけである．Teredo クライアントは，Teredo サーバを通じて ICMPv6 Echo Request を宛先ノードに送る．その Echo Reply で自然に Teredo relay が選択されるだろう．以後はその Teredo relay を経由して Teredo クライアントと IPv6 ホストは通信を行うことが可能になる (図 5)．IPv6 の自然なルーティングで選ばれた Teredo relay は，多くの場合 IPv6 ホストから最短の距離にある．したがって，この Teredo relay を利用することで通信を最適化する効果を期待できる．時には宛先の IPv6 ホストそのものが Teredo Host-Specific Relay(Teredo クライアントと直接通信できるホスト) であり，効率的な通信を行える可能性もある．

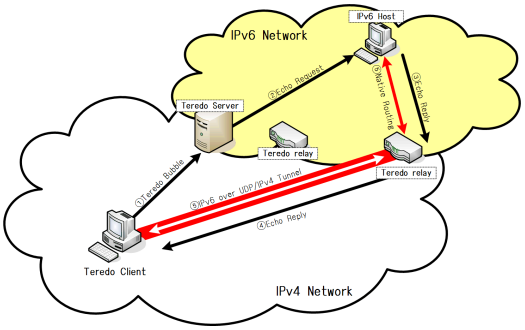

図 5: Teredo クライアントから IPv6 ホストへの通信

■Teredo クライアント間の通信

Teredo は STUN とよく似た UDP Hole Punching の NAT 越え手法を実装しており，クライアント間の通信においては，NAT 越えを行い直接通信を行う．

Teredo は ping のような到達性の確認パケットとして「Teredo バブル」というパケットを利用する．このパケットは，UDP/IPv4 ヘッダと IPv6 ヘッダだけの簡単なパケットである．Teredo クライアントは NAT にマッピングする (外から通過するための「穴」を開ける) ために，この Teredo バブルを Teredo サーバに定期的に送信している．このような穴を両端の NAT に用意することで，NAT を多重に介したとしてもあたかも NAT が存在しないかのように End-to-End の通信が可能となる (図 6)．

■同一リンク上の Teredo クライアントとの通信

同一リンク上の Teredo クライアントと Teredo クライアントの通信は，もっと単純に行われる．

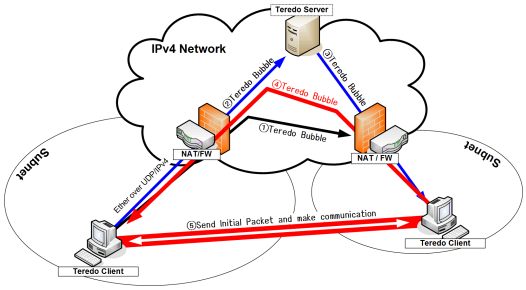

図 6: Teredo クライアント間の通信

Teredo クライアントである送信元ノードは，宛先ノードのアドレスを入れた Teredo バブルの IPv4 マルチキャストを行う．パケットを受け取った宛先ノードは，ユニキャストでバブルを返送し，自身の IPv4 アドレスを通知する．こうして，送信元ノードは宛先ノードの IPv4 アドレスを発見し，通信を行うことが可能になる．

2.4.3 利点

Teredo の最大の利点は，利用するノード単体への設定のみで NAT の内側のノードにさえも IPv6 の接続性を容易に与えられることである．これは 6to4 や ISATAP にはない大きな特徴である．

また，優れた通信効率も利点である．カプセル化に UDP を使っただけの技術ではなく，NAT 越えの技術も併用してクライアント間の直接通信を可能とし，無駄の少ない通信を行う．これは同時に，Teredo relay の負荷を減らすことにもつながっている．

2.4.4 欠点

Teredo はノードにプレフィクスを配る技術ではない点が欠点である．また，LAN の外部のホスト宛の UDP 通信を制限しているネットワークでは当然利用することは出来ない．また，外部のホスト宛に UDP 通信が出来たとしても，一部の NAT(Symmetric NAT) では NAT 越えが出来ないため利用することが出来ない．

なお，現状では利用できない技術である．現在の Teredo は IPv6 グローバルネットワークへの接続方法として十分に機能せず，ネイティブの IPv6 ホストとの通信が不可能な状態にある．その理由は Teredo relay がまだ殆ど存在していないことにある．なお，Teredo サーバに関しても十分な数が分布していない．ISP の協力による Teredo relay/サーバの設置が不可欠である．

最後に，現在の Windows XP SP1/SP2 における実装では，Teredo の他のインタフェースに IPv6 の接続性があるならば Teredo は自動的に Teredo Host-Specific Relay として構成され，アドレスは失われ，他のインタフェースより優先的に利用することが出来なくなる．したがって，Windows で Teredo を優先的に利用するためには，他の接続性をなくす必要があることに注意が必要である．

2.5 考察

本章では，代表的な IPv6 over IPv4 トンネリング技術の特徴と利点・欠点について述べた．結果，これらの技術にはそれぞれ必要な条件と，得手不得手があることがわかる．あるノードがそれらの技術を利用する上で必要な条件と，IPv6 ルータとなることが可能かどうかを表 1 に示す．

表 1 より，エンドユーザは，NAT の内側において LAN に ISATAP ルータが存在せず，UDP が制限されている場合，IPv6 の接続性を得ることは出来ない．また，IPv4 のグローバルアドレスが得られない場合，ノードを IPv6 ルータを構成することは出来ず，周辺のノードに IPv6 の接続性を与えることは出来ない．

表 1: 必要な条件と IPv6 ルータ化

接続技術	必要な条件	ルータ化
静的トンネリング	IPv4 G.A. 双方からの設定 静的なアドレス	○
6to4	IPv4 G.A.	○
ISATAP	LAN 内のルータ	×
Teredo	外部宛の UDP 通信	×

G.A. はグローバルアドレスの略．ルータは IPv6 ルータ．

UDP が制限された環境は制限の厳しいネットワークならば稀な環境ではない．多くの場合 UDP は名前解決に利用できれば十分であるため，制限の厳しい環境では外部ホスト宛の通信は遮断されている事が多い．そのような環境で 1 ノードの IPv6 通信のためにネットワークポリシを変更する (してもらう) のは，現実的ではないだろう．また，エンドユーザがグローバルアドレスを得られない ISP は存在し，今後更に増えることが想像される．そのような ISP を利用している場合，ユーザ個人のネットワークに IPv6 の接続性を与えることは，これまでに挙げた技術では不可能である．

多くの環境でノードやネットワークに IPv6 の接続性を与えるためには，これらの技術では接続性を得られない条件でも利用できる技術が必要である．その要件は以下のようになる．

1. IPv4 グローバルアドレスが不要
   NAT の内側に存在するノードも利用できる．
2. 静的なアドレスが不要
   ネットワークの移動に対応でき，DHCP の環境下でも利用できる．
3. LAN 内に IPv6 ルータが不要
   エンドユーザがノードに設定をするだけで利用できる．
4. UDP が制限されていても利用可能
   FW により制限されたネットワークでも利用できる．
5. IPv6 ルータとして構成可能
   ノードの所属するネットワークに IPv6 の接続性を提供できる．

この要件を満たす技術があれば，既存の技術では接続性を得られないような制限された環境でも，必要とする IPv6 の接続性を得ることが出来るだろう．

3. IPv6 over TCP/UDP の設計

本章では，既存技術では IPv6 の接続性を得られない環境でも，周辺ノードを含めた接続性を得ることが可能な，IPv6 over TCP/UDP のトンネリング手法の設計を述べる．

我々はこの接続技術に，Connect All Unemancipated Lans Into Six の頭文字を取って『**CAULIS**』と名付けた．以下では，CAULIS における IPv6 over UDP を **CAULIS-UDP**，IPv6 over TCP を **CAULIS-TCP** と呼ぶことにする．

3.1 概要

本研究の手法は，エニーキャストや DNS に頼らずに近傍のリレーサーバを探索可能とすることで，トンネル終端双方が動的であっても効果的な静的トンネルを構築可能とするものである．

CAULIS におけるリレーサーバは，既に何らかの IPv6 の

接続性を持っていてルータとなれるノードである．IPv6 パケットは IPv4 の TCP あるいは UDP パケットにカプセリングされ，NAT の内側のノードもトンネルを構成し IPv6 の接続性を得ることが可能である．このとき，IPv6 の接続性を提供するリレーサーバを**ルートノード**と呼び，接続性を得ようとするノードを**ブランチノード**と呼ぶ．

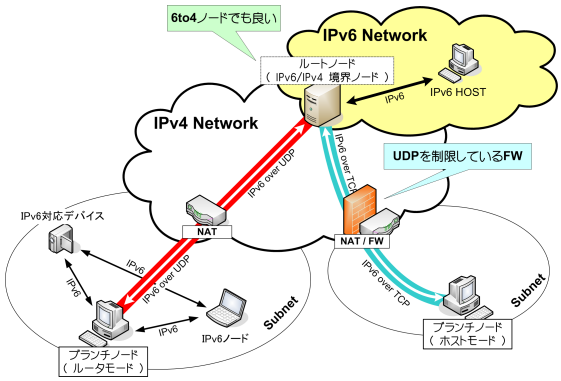

図 7: ルートノードとブランチノード

ルートノードはブランチノードに単純に IPv6 の接続性を提供するだけではなく，自身の所持しているアドレス空間からネットワークプレフィクスを配布し，ブランチノードを IPv6 ルータとして構成することを可能とする．つまり，LAN 内の 1 台のノードが CAULIS を利用し，IPv6 ルータとなることで，周辺ノードには何の変更も加えることなく LAN 全体に IPv6 の接続性を与えることが可能となる (図 7)．

プレフィクスの配布方法およびルートノードの設置における要件の敷居を低く設計することで，ルートノードを容易に設置可能とし，ISP の協力を必要とせずユーザサイドから接続技術の性能を向上可能とする．

3.2 ホストモードとルータモード

ブランチノードは，「ホストモード」と「ルータモード」の 2 つの動作モードを持つ．

ホストモードはブランチノードのみに IPv6 の接続性を与える動作モードである．ルートノードから IPv6 のアドレス 128bit の全体が配られる．ルータモードは IPv6 ルータとして構成可能とする動作モードである．いくつかのネットワークプレフィクス (例えば 63bit 長) を受け取り，周辺ノードにアドレスを配布可能となる．

ルートノードは対応可能なブランチノードを制限できる．つまり，ノードが所有するアドレス空間が広大でなくとも，ホストモードのブランチノードのみを受け入れるルートノードとして構成できる．

3.3 TCP と UDP の併用

TCP と UDP を併用するのは，より効果的な接続手法を選択するためである．

一般に，UDP は送出したパケットについてノードが考慮する必要はなく，トンネリングに向いているとされる．しかし，UDP の通信は名前解決に利用出来れば必要十分とされることがあり，LAN の外部のホストに対する UDP の通信はファイアウォール (以下 FW) で塞がれている可能性がある．そのように UDP が制限された環境でも，外部のホスト宛の TCP 通信ならば利用できる可能性が高い．

ところが，TCP のトンネルはパケットロス時に輻輳が起こり，急激に通信速度が低下することが TCP over TCP 問題として知られている [10]．この問題は，TCP の SACK オプション (WindowsXP ではデフォルトで有効) で抑えることができる [11] が，それでも低速な回線では通信が不安定になりやすい．

つまり，安定性やスループットにおいては TCP よりも UDP の方が性能が高く，利用においては UDP よりも TCP の方が可能性が高い．相互に欠点を補完するために，併用することが効果的である．

3.4 アドレス規則

プレフィクスは，ルートノードの持つアドレス空間からブランチノードに割り当てられる．識別子となるものはない．配布される範囲はブランチノードの振る舞いによって異なり，ホストモードならば 128bit(アドレス全体) が配られ，ルータモードならばネットワークプレフィクス (例えば 63bit 長) が配られる．

ルートノードは，自分自身の持つ IPv6 のアドレス空間の大きさやネットワーク帯域に基づいて，配布する範囲を自由に設定できる．設定はルートノードの管理者の任意である．

3.5 ルーティング

この接続方法は実質的には静的なトンネルであるため，ブランチノードの通信は全てルートノードを経由して行われる．ブランチノードが持つプレフィクスは元々はルートノードの持つプレフィクスであるため，ルートノードのネットワークポリシに基づいてパケットはルーティングされる．

3.6 ルートノードの探索

CAULIS-TCP/UDP におけるリレーサーバであるルートノードは，既存技術のようにエニーキャストや名前解決で選択することが困難である．なぜなら，ユーザが自由にルートノードを設置/撤去することも可能であるからだ．

本研究ではこの問題を解決するため，エニーキャストや名前解決に頼らずに P2P ネットワーク的な手法で近傍のサーバを発見することを可能とする手法を設計している．

3.6.1 近傍ルートノードの発見

本研究で取った近傍探索の手法では，Traceroute とデータベース (ハッシュ表) を利用する．このデータベースはネットワーク上に存在し，CAULIS を利用するノードが共有するものである．局所性の実現に Traceroute を利用する手法は，CDN 技術の Coral[12] においても利用されている．

Traceroute をあるホストに対して行った場合，自分自身が利用しているルータのアドレス (以下，RA) を得ることが出来る．そのため，同一 AS 内でルートノードとブランチノードが同様の行為を行った場合，経路の途中で同一のルータを利用している可能性が十分に高く，同一の RA を得る可能性が高い (図 8)．

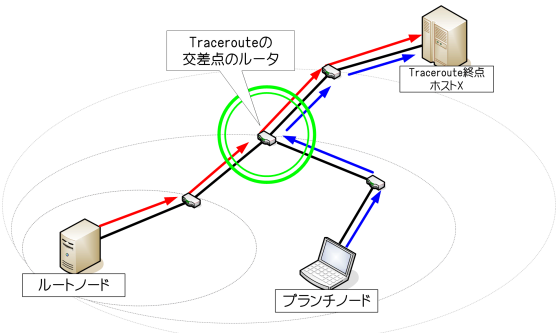

図 8: Traceroute に基づく近傍

そこで，この RA をハッシュ表のキーとして利用し，データベースを介して近傍探索を可能とする．具体的には，ルートノードはあるホスト X 宛の Traceroute の結果で得られた RA をキーとして，自分自身のアドレスなどの情報 (MyInfo) をデータベースに *put(RA, MyInfo)* を行って格納する．もし 10 個のアドレスが得られたならば，10 回この操作が行われる．反対にブランチノードは X 宛の Traceroute の結果で得られた RA をキーとして，データベースから *get(RA)* によって値を引き出す．もしルートノードとブランチノードが X までの経路で同一のルータを利用しているならば，ルートノードの格納した値 (＝ルートノードの情報) が得られる．

このとき，ルートノードが各ルータまでの RTT(Round Trip Time) もデータベースに納めておくことで，ブランチノードから各ルータまでの RTT との和によって，参考程度にルートノードとブランチノードの距離を測定できる．これによって得られる値は正確ではないが，大量のルートノードが発見された場合にフィルタとして利用できるだろう．この仕組みを通じて近傍のサーバを発見することが可能となる．

3.6.2 探索に失敗した場合

この手法ではルータの負荷分散処理やノードの所在によってルートノードを全く発見できないことが想像される．そのような場合に備え，ルータの FQDN を分解したものもキーとして利用する．具体的には，router.ubi.cs.ritsumei.ac.jp というルータがあったとしたならば，(1)ubi.cs.ritsumei.ac.jp (2)cs.ritsumei.ac.jp (3)ritsumei.ac.jp (4)jp というキーを得る．なお「ac.jp」というキーは，「jp」と範囲が同様であるため排除される．同様に，「com」「net」といったものも排除される．この方法で，弱い近傍性でルートノードを発見する．

これでも最終的に一切のルートノードを得ることが出来なかった場合，確実な手段 (CGI やキャッシュなど) でランダムで選択されたルートノードの内から，最も近傍のルートノードを利用する．この仕組みにより，Traceroute を基に発見できなかったルートノードを発見することを可能とし，また，ルートノードを一切発見できない事態を回避する．

3.7 ノードの要件

上記の設計を基に，ルートノードとブランチノードに必要な要件をまとめる．

3.7.1 ルートノードの要件

以下は，ルートノードの要件である．

1. IPv4/IPv6 デュアルスタックのノードである．
2. IPv6 の接続性を有している．
3. IPv6 ルータの要件を満たしている．
4. プレフィクス長 63bit 以下のグローバルアドレス空間を所有する．
5. IPv4 のグローバルアドレスを持っている．あるいはフォワードにより必要なプロトコルの通信を受け入れることが出来る．

要件 4 について補足する．ルートノードがホストモードのブランチノードのみを受け入れる場合，所有するアドレス空間はプレフィクス長 63bit で良い．ルータモードのブランチノードを受け入れる場合，ルートノードはプレフィクスを配るため，より広大なアドレス空間が必要となる．

要件 5 は，ルートノードは不特定のネットワークに所在するブランチノードからの通信を受け入れるパブリックサーバであり，IPv4 のグローバルアドレスを所有していなければフォワードによって通信を受けなければならない事を示している．

ここで挙げた要件は，現在 IPv4 のグローバルアドレスを所有しているノード，あるいは IPv4 のグローバルアドレスを所有しているルータを直接操作可能な環境に位置するノードならば，容易に満たすことが可能である．例えば，6to4 ノードならば既に要件を満たしていることになる．

3.7.2 ブランチノードの要件

以下はブランチノードの要件である．

1. IPv4/IPv6 デュアルスタックのノードである．
2. IPv4 の接続性を持ち，外部のホストに対する TCP あるいは UDP の通信が出来る．

ブランチノードに必要な要件は実に単純である．ルートノードとの TCP あるいは UDP の通信さえ出来れば，問題なく CAULIS の利用が可能である．

3.8 利点

この設計の利点は，UDP と TCP を状況に合わせて選択できることである．例え UDP を制限しているネットワークにおいても，TCP ならば利用できる可能性が高い．

さらにプレフィクスを配ることのできる設計であるため，ブランチノードは IPv6 ルータとして構成することができる．これにより，NAT の内側のノードをブランチノードとして構成し，その周辺のノードにまとめて IPv6 の接続性を与えることができる．また，大きなネットワーク空間を受け取ることで，自由度の高い構成ができるだろう．

また，この手法ではルートノードの設置に関して ISP の協力が必要ない．ルートノードは特別なアドレスを持っているわけでもなく，いわば通常の IPv4/IPv6 のデュアルスタックホストである．所有するアドレス空間に応じて受け入れるブランチノードの種類を選択できるため，わずかなアドレス空間しか持たなくともルートノードとして構成可能である．つまり，このルートノードの要件を満たすことは難しくはなく，エンドユーザですら容易に設置可能である．したがって，ルートノードの拡散が容易である．

3.9 欠点

6to4 や Teredo などの自動構成トンネリングと異なり，全ての通信がルートノードを経由するため通信効率が悪い．特に，ルートノードとのネットワーク距離が遠いときや，ルートノードのスループットが低いとき，TCP によるトンネリングを選択したとき，この影響は顕著になる．なるべく近くの，通信帯域に余裕のあるルートノードを利用する必要がある．

4. 適応的なトンネルの構成

本章では，これまでに挙げてきた接続技術の特徴を踏まえた，最適なトンネルの構成を述べる．また，その構成を自動的に行うためのシステムの設計について述べる．

4.1 接続技術の選択の必要性

2 章，3 章で述べたように，それぞれのトンネル接続技術には長所と短所がある．CAULIS-TCP のようにあらゆる環境で高い確率で接続可能とする技術はあるが，それを常に利用するのは効率が良いとは言えない．それぞれの接続技術を考慮した上で，状況に応じて最適な接続技術を組み合わせて選択することが重要である．

ノードに IPv6 の接続性を与えるのならば，スループットを基準として接続技術を構成するのが効果的である．しかし，エンドユーザがどの接続技術をどう組み合わせれば効果

的か判断することは容易ではない．また，モバイル PC のように様々なネットワーク環境を移動するノードに，その場その場で最適な接続技術を設定するのは煩雑である．

そのような煩雑さを解決するため，最適な接続技術を設定する適応的なトンネル構成技術を設計する．

4.2 接続技術とネットワーク環境

これまでに挙げた IPv6 の接続技術を，パケット構造やルーティングの観点から，総合的に高速な通信を期待できる順を考察し，利用する優先度と組み合わせを決定する．なお，一般的な静的トンネリングは設定が煩雑なため考慮しない．

本稿で挙げた既存の接続技術の中で，ISATAP は特殊である．ISATAP は ISATAP ルータの接続性に依存しており，多くの場合高速であると推測されるが非効率な可能性もある．確実に高速なのは 6to4 である．6to4 はパケット構造がシンプルであり，6to4 ノード間の通信も効率が良い．Teredo も効率の良いルーティングを行うが，パケット構造および通信初期のコストの大きさから，6to4 には劣る．なお，前章で設計した CAULIS は静的なトンネル接続技術であるため，多くの場合これらの接続技術に劣る．

以上を考慮すると，通信速度は (ネイティブ IPv6)，(ISATAP)，6to4，Teredo，CAULIS-UDP，CAULIS-TCP の順となる．このとき，ISATAP と Teredo に関してはノードをルータ化することが出来ないことに注意が必要である．また，ISATAP で接続性を得られたとして，得たプレフィクスが 6to4 のものであった場合は，可能ならば自分自身を 6to4 ノードとした方が最適である．

以上の順を接続技術の優先順序とすると，最適な接続技術を構成するにあたって確認すべきネットワーク環境は以下の項目となる．

1. IPv6 グローバルの接続性を持っている → [ネイティブ]
2. ISATAP ルータがサイトに存在する → [ISATAP]
   ただし，ISATAP で得たプレフィクスが 6to4 のもので，かつ 6to4 を利用できる → [6to4]
3. IPv4 のグローバルアドレスを持っている → [6to4]
4. UDP で NAT 越えを行って通信できる → [Teredo]
5. LAN 外部のホスト宛に UDP で通信できる → [CAULIS-UDP]
6. LAN 外部のホスト宛に TCP で通信できる → [CAULIS-TCP]

これらの条件を順次確認し，Yes となった項目に対応する接続技術を設定することで最適な接続性が得られる．もし接続技術を設定しても IPv6 グローバルの接続性が得られなかったならば，次の Yes となった項目に対応する接続技術が最適となる．

このとき，項目 1-3 のいずれかで接続性を得ることが出来た場合，可能ならば Teredo を Host-Specific Relay として構成し，Teredo クライアントとの通信を効率化すると良い．同様に，項目 1-2 で接続性を得られた場合に 6to4 を併用すると，6to4 ノードとの通信に効果的である．

これらの構成と確認はユーザ自身が行うと煩雑である．システムが肩代わりすることで負担を減らす．

4.3 ノード単体に IPv6 の接続性を与える場合

この場合，前項で示した優先度を基準としてトンネル接続技術を選択するのが効果的である．このとき，通信効率を高めるために 6to4 や Teredo を併用することも考慮する．ISATAP に関しては，得られたプレフィクスを確認する必要がある．

この構成順序を図 9 に示す．

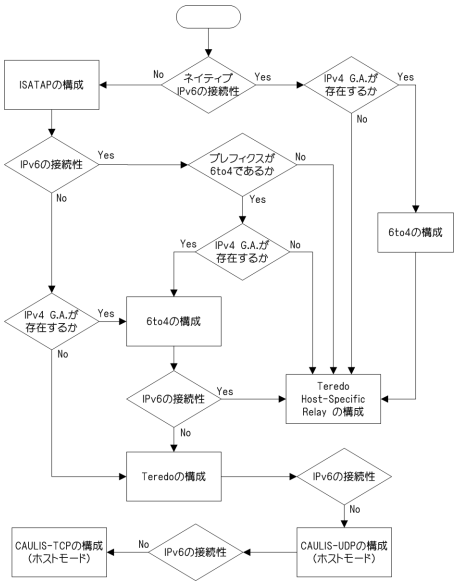

図 9: ノード単体に接続性を与える構成順序

なお，この順序では Teredo で接続性を得られないという現在の Teredo の問題点は考慮していない．したがって，Teredo は Teredo サーバとの初期動作に成功しても，必ず接続性のテストで失敗すると考えられる．その場合無条件に CAULIS-UDP の構成に移ることだろう．CAULIS-UDP で接続性を得られたなら，自然に Teredo は Teredo Host-Specific Relay となる．

4.4 ノードをルータとして構成可能とする場合

ノードをルータとして構成可能にするためには，プレフィクスを得る必要がある．本稿で挙げた既存の接続技術の中で，プレフィクスを得ることが出来るのは 6to4 のみである．6to4 が利用できなかった場合，本研究の CAULIS を利用することが最終手段となる．この場合でも，6to4 で接続性を得られたときに Teredo Host-Specific Relay を構成し，通信効率を向上させる．

その条件判断と構成順序を図 10 に示す．

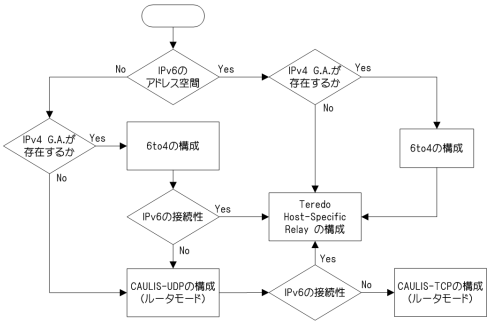

図 10: ノードをルータ化する構成順序

5. 実装と評価

本章では，ノードのネットワーク環境を確認し，最適なトンネル接続技術をノードに対して設定するプログラム「6connect」について，その実装内容と評価を述べる．

6connect は，CAULIS と適応的なトンネル構成技術のプロトタイプ実装である．6connect は OS 非依存になるように設計されており，6connect を使うことでネットワークの知識がなくとも最適な IPv6 の接続性をノードに与えることが可能となる．

5.1 CAULIS の実装

CAULIS の実装においては，IPv6 のパケットを TCP および UDP のパケットにカプセル化する技術が必要である．本実装のプロトタイプにおいては，オープンソースの VPN である OpenVPN[16] を利用してトンネルを構成している．

5.1.1 トンネリングの実装

■OpenVPN によるトンネリング

OpenVPN は Tun/Tap と呼ばれる仮想インタフェースを利用して，サーバとクライアントの間で UDP あるいは TCP のトンネリングを実現する．このとき，VPN サーバとなるのがルートノードであり，VPN クライアントとなるのがブランチノードである．ここに IPv6 のアドレスやプレフィクスを配布/設定する機能を別途用意し組み合わせることで，トンネルを確立する．

本実装では，Java のプログラムで VPN セッションの構成および IPv6 の設定を動的に行っている．このプログラムは，ルートノードとブランチノードの間で対話的にお互いのノードに IPv6 の設定を施し，OpenVPN を操作してトンネルを構成する．このプログラムを通じ，ノードに IPv6 の接続性が与えられる．

■現状の問題

現状では設計通りのトンネルの実装に至っていない．

その主な理由は，Windows において，Tun/Tap デバイスが IPv6 に正式に対応していないことである．このため，IPv6 のパケットをカプセル化することが出来ず，レイヤ 3 のトンネリングを行うことが出来ない．そのため，本実装ではレイヤ 2 のトンネリングを行っている．

したがって，ルートノードとブランチノードが同一リンク上に見えていることに注意が必要となる．本実装では同一リンク上でなければ利用できないプロトコルをノードの構成には利用していないが，不正に送信できてしまう危険性がある．また，Ethernet 上のパケットを送信するため，通信効率の低下が予想される．

これらの点に対策が必要である．

5.1.2 ルートノードの探索技術の実装

近傍のルートノードの探索技術の実装における主要は，データベース (以下 DB) の構成と上位のルータの探索に利用する技術である．

■データベースの構成

6connect の DB は現在単一サーバが担当している．DB に対しては *put(key, value)* と *get(key)* という操作を外部のホストから行うことが可能で，この操作を通じてルートノードは自身のアドレスを登録し，ブランチノードはルートノードのアドレスを取得する．

1 つのキーにはリスト構造で複数のデータが格納される．リストは重複がないように整理されており，新しいデータはリストの最後尾に追加されるため，チャーン (不安定なノード) を区別しやすい．

ルートノードがデータベースに登録するデータには生存期間が設定されており，ルートノードが何らかの原因で消滅した場合，一定期間の後にデータベースから削除される．ルートノードは生存期間が切れる前に自分自身の情報を更新することで，データの生存期間を延ばし，DB に情報を保存し続ける．なお，登録されるデータは XML で構成されているため，将来的に拡張が可能であり，異なる実装でも利用が可能であろう．

このような設計は，P2P ネットワークの探索技術である DHT と似ている．したがって，将来的には静的に存在すると考えられるルートノードで築かれる DHT を利用して，単一障害点のない DB を構築する事が出来るだろう．

■上位のルータの探索

本実装では ICMP を用いた Traceroute を利用して上位のルータを探索する．

終点ホストには，異なる地域に存在するホスト 3 つを指定している．この内 1 つは，終点ホストが近く，十分な数の上位ルータを得ることが出来なかった場合に備えてのアドレスであり，2 つのアドレスに対しての Traceroute で十分な数の上位ルータを得られれば必要ない．2 点に対し Traceroute を行うのは，十分な数のルータを探索するためと，ルータの負荷分散によって経路が一定でない場合にも，他のノードが同一のルータを通過する確率を高めるためである．

このときルータの FQDN も同時に取得し，3.6.1 節の設計のように分解して，弱い近傍性実現のためのキーとする．この場合でもルートノードのアドレスが全く取得できなければ，DB からランダムでルートノードのアドレスを取得する．

なお，この実装には ICMP を利用しているという欠点が存在する．ICMP を制限しているネットワークではこの手法が利用できないため，本来ならば UDP や TCP を利用した Ping を実装すべきであろう．その問題点を補うために，自分の所属するネットワークの DNS Suffix を得ることで，FQDN を取得したときと同様の処理を行い，弱い近傍性でルートノードを探索することを可能としている．

5.1.3 関連研究

■OCN IPv6

NTT コミュニケーションズが一般ユーザ向けサービスとして提供している IPv6 over UDP のトンネル接続サービスで，「どこでも IPv6」と呼ばれている．接続クライアントのみが IPv6 の接続性を持つ”ホストモード”と，プレフィクスを受け取りルータとなる”ルータモード”の 2 つの動作モードを持つ．

このサービスでは，IPv6/IPv4 ネットワークの境界ルータを接続サーバとし，IPv4 ネットワークで孤立しているノード (接続クライアント) との間に静的なトンネルを構成して，接続クライアントおよびその周辺ノードに IPv6 の接続性を提供する．UDP を利用しているため NAT の内側のノードも利用可能である．

この技術におけるトンネルは，厳密には IPv6 over PPP over L2TP over UDP/IPv4 である．つまり，UDP ベースのプロトコルである L2TP 上で PPP の認証を行い，IPv6 のパケットを交換可能とする．サーバとクライアントの間にはレイヤ 2 のトンネルが構成されるため，ケーブルで接続したかのように振る舞い，DHCPv6 を利用して自然な IPv6 のネットワーク設定が行われる．

この技術では TCP のトンネルに関しては考慮されていない．また，サーバとクライアントの間のネットワーク距離が性能に大きく影響するが，サーバの分布は NTT コミュニケーションズに依存している．

■OpenVPN IPv6 Tunnel Broker

JOIN において Christian Strauf 氏が提言している OpenVPN を用いたトンネルブローカ手法である [14]．トンネルブローカとは RFC3053 にて提案されているトンネル構成手法 [15] であり，IPv4 ネットワーク上に存在する IPv6 の接

続性を提供するサーバである．

このシステムの提案では，トンネルブローカがクライアントからの接続要求を受け取ると，データベースにクライアントの情報を保存すると共にクライアントの公開鍵をOpenVPN サーバに渡す．これを鍵としてクライアントとOpenVPN サーバ間で Point-to-Point のトンネルを構築する．クライアントはただ接続性を得るだけでなく，ルータとして構成することも可能である．

この設計は CAULIS の実装と同一のように思われる．しかし，明確にしておきたいのは，CAULIS は「TCP/UDP を用いたトンネル接続技術の実装として現在は OpenVPN を利用している」ということである．なお，CAULIS ではユーザが自由に VPN サーバ (ルートノード) を公開でき，VPN サーバが流動的である点も異なる．

5.2 適応的なトンネル構成技術

6connect は，ホストモードではあらゆる接続技術を試み，ルータモードではノードがルータになれる接続技術のみで自動構成を試みる．優先順序および条件は 4.1 節の考察を踏まえている．最適な接続技術の構成を行っても接続性が得られなかった場合は，次の接続技術の構成を行う．このシステムにより，ノードにインターネットの接続性があれば IPv6 グローバルの接続性を得ることができるだろう．

なお，このシステムは IPv4 ネットワークの変化やルートノードの消滅を検知すると直ちに接続技術の再構成を行うようになっている．したがって，ノードが次々とネットワークを移動したとしても，それぞれのネットワークの障害や制限を吸収し，適応的に IPv6 の接続性を得ることを可能とする．

5.3 評価

5.3.1 CAULIS の性能

本研究の CAULIS のプロトタイプ実装の性能を確認し，有用であるか評価する．

なお，CAULIS のプロトタイプ実装では，設計と異なりレイヤ 2 のトンネリングを行っている．そのため，IPv6 パケットをカプセル化したときに 12byte のフレームヘッダも付加され，通信効率の低下およびフレームヘッダの付与/除去に伴うオーバヘッドが想定される．設計通りの実装がされれば本実装より高速な結果が期待できるため，この評価は本来の性能より劣ることを考慮する．

■実験環境 プロトタイプ実装におけるトンネルのカプセル化のオーバヘッド，および安定性を評価するため，図 11 のようなネットワークを構築した．この実験環境では，ホスト B によって同一リンク上のホスト C に IPv6 の接続性が提供されている．このホスト B は 6to4 ルータであり，Teredo サーバ/relay であり，ルートノードである．それぞれのホストはグローバルアドレスを有している．

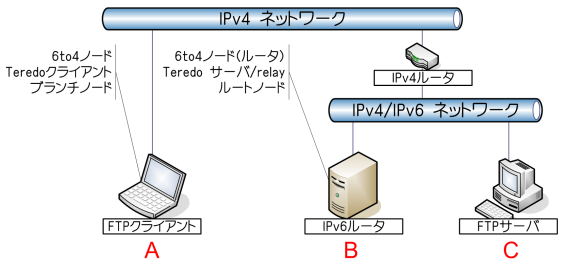

図 11: 実験環境

この実験では，ホスト A はホスト B を利用して IPv6 の接続性を得て，ホスト C から FTP で 100MB のデータをダウンロードする．実験に用いる接続方法は 6to4，Teredo，CAULIS である．なお，ホストおよび経路の性能の指標として IPv4 の通信も行う．

CAULIS は UDP と TCP の接続方法が存在しそれぞれ評価を行っているが，この実験環境では通信路が安定しており TCP over TCP 問題が発生しにくいため，ここでは TCP 接続を比較対象とする．また，本来ならば一般的な静的トンネル，および ISATAP も比較対象とするべきであろうが，同一リンク上でテストした結果 6to4 と性能に殆ど差が見られなかったため，ここでは割愛する．

なお，各ホストの仕様/性能は表 2 の通り．

表 2: ホストの性能

ホスト A	OS: Windows XP Home Edition SP2 CPU: Intel(R) Pentium(R) M 1.20GHz メモリ: 1024MB
ホスト B	OS: Fedora Core 4 CPU: Intel(R) Pentium(R) 4 3.40GHz メモリ: 512MB
ホスト C	OS: Windows XP Professional SP2 CPU: Intel(R) Pentium(R) M 1.73GHz メモリ: 1024MB

■実験結果 結果は図 12 のようになった．グラフ横にはスループットの計測値 (Mbps 単位) と IPv4 を基準としたときの百分率を記載している．

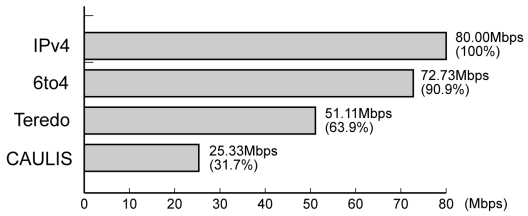

図 12: スループット測定結果

■考察 CAULIS は速度の面で Teredo に大きく劣るものの，通信路として実用出来るレベルは保っていることがわかった．

CAULIS が Teredo に劣るのは，本実装はレイヤ 2 でトンネリングしていることが 1 つの原因である．また，一般にユーザランド VPN は通信速度が遅いと言われており，ユーザランド VPN である OpenVPN は性能面で不利がある．これらはプロトタイプの実装に起因する問題である．

5.3.2 ネットワーク状況に適応した接続

4.2 節で，確認すべきネットワーク環境と，その項目が Yes であったときに優先的に利用すべき接続技術を述べた．

それを踏まえ，(1)IPv6 グローバルの接続性はあるか (2)ISATAP ルータが存在するか，および ISATAP で得たプレフィクスが 6to4 ではないか (3)IPv4 グローバルアドレスを持っているか (4)UDP で NAT 越えできるか (5)UDP で外部と通信できるか (6)TCP で外部と通信できるか という 6 項目の確認動作が正常に行われ，様々な環境で IPv6 の接続性を得られるか確認する．

■結果:ホストモード ノード単体に IPv6 の接続性を与えるホストモードで自動構成を行った結果を表 3 に示す．

Teredo は構成されたとしても必ず接続性の確認で失敗するため，Teredo の構成されるべき順序で CAULIS-UDP が構成されている．そのとき，Teredo は Host-Specific Relay となっている．これは 4.3 節でも述べた動作である．

表 3: ホストモードの自動構成

1	2	3	4	5	6	設定された接続技術
○	*	○	○	○	*	なし (+ 6to4+Teredo)
○	*	○	×	*	*	なし (+ 6to4)
○	*	×	○	○	*	なし (+ Teredo)
○	*	×	×	*	*	なし
×	○	*	○	○	*	ISATAP (+ Teredo)
×	○	*	×	*	*	ISATAP
×	△	○	○	○	*	6to4 (+ Teredo)
×	△	○	×	*	*	6to4
×	△	×	○	○	*	ISATAP (+ Teredo)
×	△	×	×	*	*	ISATAP
×	×	○	○	○	*	6to4 (+ Teredo)
×	×	○	×	*	*	6to4
×	×	×	○	○	*	CAULIS-UDP (+ Teredo)
×	×	×	×	○	*	CAULIS-UDP
×	×	×	×	×	○	CAULIS-TCP
×	×	×	×	×	×	失敗

番号は確認項目と一致.
○は Yes, ×は No. *はどちらでも良いことを表す.
項目 2 のみ, △はプレフィクスが 6to4 の場合を示す.

後半で設定される技術は，より過酷な条件でも動作していることが分かる．最終的に，TCP で LAN の外部のホストと通信できない場合以外，接続技術を構成できることが確認された．

■結果:ルータモード ノードをルータとして構成可能なルータモードで自動構成を行った結果を表 4 に示す．

表 4: ルータモードの自動構成

1	2	3	4	5	6	設定された接続技術
○	*	○	○	○	*	なし (+ 6to4+Teredo)
○	*	○	×	*	*	なし (+ 6to4)
○	*	×	○	○	*	なし (+ Teredo)
○	*	×	×	*	*	なし
×	*	○	○	○	*	6to4 (+ Teredo)
×	*	○	×	*	*	6to4
×	*	×	○	○	*	CAULIS-UDP (+ Teredo)
×	*	×	×	○	*	CAULIS-UDP
×	*	×	×	×	○	CAULIS-TCP
×	*	×	×	×	×	失敗

番号は確認項目と一致.
○は Yes, ×は No. *はどちらでも良いことを表す.
ここでの”ネイティブ”はルータの要件を満たしている必要がある.

ホストモードと異なり，プレフィクスを配る接続技術以外は構成されない．最終的に，LAN の外部のホスト宛の TCP 通信ができない場合を除いて，ルータとなれる接続技術が構成されている．

■考察 設計した通りに動作していることが認められ，LAN 外部のホストとの TCP 通信が制限されているネットワークでなければ，何らかの IPv6 の接続性が得られることが示された．それはノード単体への接続性だけでなく，ノードをルータとして構成し，ノードの所属するネットワーク全体に接続性を与える場合においても，同様の結果であった．その構成順序もスループットを意識したものである．

以上より，CAULIS の実装によって，IPv6 の接続性を得られるネットワーク環境が広範になったことが示された．

6. まとめ

本稿では，制限されたネットワーク環境でも利用可能な IPv6 over TCP/UDP のトンネル接続技術の設計を述べた．また，ネットワーク環境に最適な接続技術の構成を述べた．それらのプロトタイプの実装では，実用できることが示された．

「あらゆるネットワーク環境で適応的に IPv6 の接続性を得る」というこの研究は，IPv6 の研究者/アプリケーション開発者にとって大きな利益をもたらすものと考えている．また，接続設定をシステムが肩代わりすることは，エンドユーザが気軽に IPv6 を体験することを可能とするだろう．そこにセキュリティの懸念は確かに存在するが，容易にネットワークに接続できる利益は大きい．

本研究のトンネル接続技術の設計と実装は，まだ初期段階に過ぎない．トンネリングに問題があるのは本文で述べているが，実際に多くのルートノードを分布させての評価がまだ行われておらず，今後行っていく必要がある．それ以上に，ネットワークのセキュリティポリシを考慮した設計が実用する上で必須である．今後はそのような問題を解決し，実際に利用可能な接続技術として確立していきたい．

参考文献

[1] Geoff Huston, “IPv4 Address Report”, "http://www.potaroo.net/tools/ipv4/", 18th August 2006.

[2] IP Version 6 Working Group (ipv6) Charter, "http://www.ietf.org/html.charters/ipv6-charter.html".

[3] ”The Cable Guy - 2005 年 10 月 : Windows Vista および Windows Server ”Longhorn” の IPv6 への変更”, "http://www.microsoft.com/japan/technet/community/columns/cableguy/cg1005.mspx".

[4] Peter R. Tattam, “Yet another Dynamic Tunnel Configuration Prototocol”, "http://jazz-1.trumpet.com.au/ipv6-draft/dtcp-draft-prt-13-may-1999.htm", May 1999.

[5] B. Carpenter, K. Moore, “Connection of IPv6 Domains via IPv4 Clouds”, IETF RFC3056 [Standards Track], Feb. 2001.

[6] F. Templin, T. Gleeson, M. Talwar, D. Thaler, “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)”, IETF RFC4214 [Experimental], October 2005.

[7] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, IETF RFC3489 [Standards Track], March 2003.

[8] C. Huitema, “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)”, IETF RFC4380 [Standards Track], February 2006.

[9] Changes to IPv6 in Windows Vista and Windows Server ”Longhorn”, http://www.microsoft.com/technet/community/columns/cableguy/cg1005.mspx".

[10] Olaf Titz, “Why TCP Over TCP Is A Bad Idea”, "http://sites.inka.de/sites/bigred/devel/tcp-tcp.html", Apr 23, 2001.

[11] Osamu Honda, Hiroyuki Ohsaki, Makoto Imase, Mika Ishizuka and Junichi Murayama, “Understanding TCP over TCP: Effects of TCP Tunneling on End-to-End Throughput and Latency,” in Proceedings of OpticsEast/ITCom 2005, October 2005.

[12] Michael J. Freedman and David Mazieres, “Sloppy Hashing and Self-Organizing Clusters”, In Proc. 2nd Intl. Workshop on Peer-to-Peer Systems(IPTPS ’03) Berkeley, CA, February 2003.

[13] OCN, "http://www.v6.ntt.net/".

[14] JOIN: OpenVPN IPv6 Tunnel Broker Guide, "http://www.join.uni-muenster.de/Dokumente/

Howtos/Howto_OpenVPN_Tunnelbroker.php?lang=en#clientuserguide"
[15] A. Durand, P. Fasano, I. Guardini, D. Lento, "IPv6 Tunnel Broker", RFC3053 [Informational], January 2001.
[16] OpenVPN, "http://openvpn.net/"