マルチプロトコル環境における
仮想リンク制御機構の実装と評価

卒業論文 **1998** 年度 (平成 **10** 年度)

指導教員

村井 純
徳田 英幸
楠本 博之
中村 修

慶應義塾大学 環境情報学部

増田 康人

卒業論文要旨

マルチプロトコル環境における
動的仮想リンク制御機構の実装と評価

本研究では，マルチプロトコル環境における動的仮想リンク制御機構であるTACS(Tunnel Auto Configuration System)の実装と評価を行なった．

本研究で実現した機構によって，IPv4とIPv6の共存するネットワーク環境において，トンネル管理コストの大幅な低減を実現した．また，本機構によって，IPv6をより容易に利用できる環境を実現した．

現在，インターネットにおいて，IPv4(Internet Protocol versoin 4)からIPv6(Internet Protocol version 6)への移行が進行している．移行期のインターネットは，IPv4ネットワークとIPv6ネットワークが共存している．その中で重要な技術となるのがトンネルである．しかし，現状の方式ではIPアドレスが頻繁に変化する状況に対応できず，また，自動的に通信障害を回避することができない．なぜなら，管理コストが非常に高くなってしまうか，もしくは管理ポリシを反映したトンネル設定ができないからである．

そこで本研究では，まずトンネルの管理コストが高くなる環境を明確に定義した．そして，その環境下で発生する問題を解決するため，動的仮想リンク制御機構であるTACSを設計，実装した．TACSは自動的にトンネルを設定することにより，ネットワーク管理コストを軽減する．また，トンネル認証機能により，管理ポリシを反映したトンネル設定を実現している．TACSを実験ネットワーク上で運用し，管理コストについての評価を行なった．その結果，TACSはネットワーク管理コストを軽減しつつ，管理ポリシを反映したトンネルを設定できる機構であることが実証された．TACSの導入によって，IPv6インターネットへの接続が容易になり，その結果，IPv6の普及はより促進される．また，より障害に強い，安定したIPv6ネットワークを構築できる．

Abstract of The Thesis

Implementation and Evaluation of
Dynamic Virtual Link Control System on
Multi Protocol Enviroment

In Internet, the transition from IPv4(Internet Protocol versoin 4) to IPv6(Internet Protocol version 6)has been progressing. The Internet of the transition period is a multi protocol environment, in which IPv4 network and IPv6 network exist together. The Tunnel would be an important technology in such an enviroment, the management cost, however, will rise enormously in a current method, or tunnel cannot be configured under management policy. Thus, in this reserch, I, firstly, clarified the enviroment where the control cost of the Tunnel rises.

More over, to solve the problem which occurs under the circumstances, I designed and implemented TACS(Tunnel Auto Configration System) that is the dynamic virtual link control system. I also estimated the cost by operating TACS on the experiment network.

As a result, it is proven that TACS has reduced the network control cost of setting the tunnel.

目次

図 目 次

表 目 次

第1章

「序論」

本章では，まず本研究の背景を述べ，さらに本研究の目的を述べる．また，本論文の構成について述べる．

1.1 背景

1.1.1 マルチプロトコルネットワークの発生

現在，IPv6[1]の実験ネットワークが世界規模で構築され，IPv6への移行の準備が除々に進んでいる．その結果IPv4[2] とIPv6のネットワークが共存する状況となっている．

インターネット全体のIPv6への移行は，緩やかな段階を経て進行することが望ましい．全世界のルータやホストを一斉にIPv4からIPv6に切替えると，非常に大きな混乱を招く結果となり現実的ではない．まず組織単位でIPv6への移行を行ない，そのIPv6化したネットワーク同士が接続され，段階的に大きくなっていくモデルが適切である．また，現存するIPv4インターネット全てがIPv6に対応することも現実的ではない．IETFでの議論の結果，IPv6への移行は，新たなネットワーク構築をIPv6によって行ない，主要なIPv4ホストをIPv6に対応させる，という方針で移行を進める事に決まった．よって最終的には，IPv4とIPv6のネットワークが共存し，主要なサービスがIPv6によって提供されることが想定される．将来，移行が完了した時もインターネットがIPv6のみによって運用されることは想定されていない．

1.1.2 トンネル

トンネルは複数のプロトコルが共存するネットワークを構築する上で必要不可欠な技術である．トンネルとは，同種のプロトコルを持つホスト同士が他のプロトコルを仮想リンクとして利用し通信を行なう手法である．これにより，既存のネットワークを利用して，その上に仮想的なネットワークを構築できる．トンネルを用いたネットワークの例を図 1.1に示す．

図1.1では，protocol A で構築されている既存のネットワークを仮想リンクとして利用してprotocol Bのネットワークを仮想的に構築している．トンネルはデュアルスタックホスト間に設定している．トンネルについての詳しい説明は第 2章で述べる．

1.2 本論文の目的

トンネルはIPv4とIPv6が共存するマルチプロトコル環境において，非常に重要な技術である．しかし，現在用いられている設定方式では，トポロジが頻繁に変化する状況に対応できない．また，中継ルータの停止などによる通信障害を自動的に回避することができない．その結果，トン

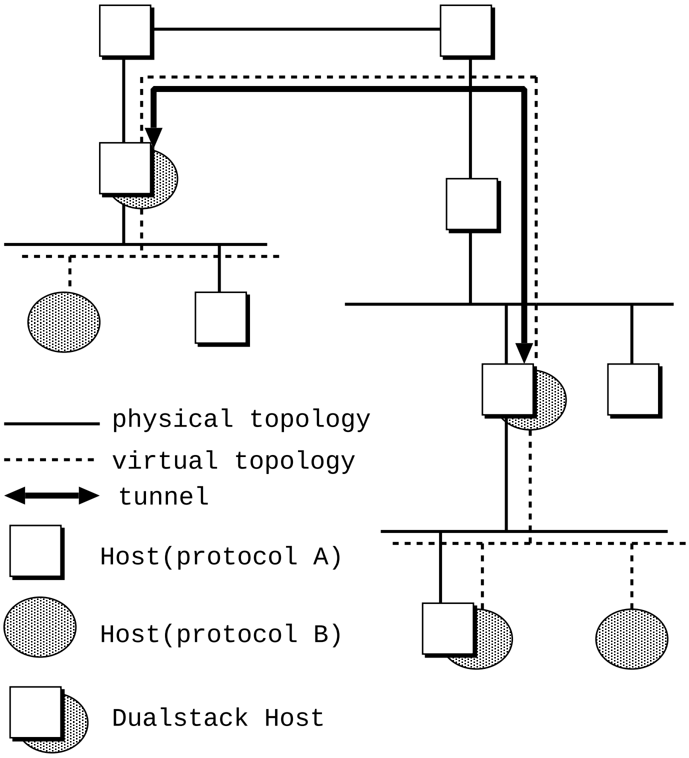

図 1.1: トンネルを用いたネットワーク

ネルの設定によって，ネットワークを管理するためのコストが高くなってしまう．そこで本研究では，以下に挙げる項目を目的とした，トンネルの自動制御機構の実装と評価を行なう．

片端トンネル設定

現状の手動トンネル設定は両端のノードからそれぞれ行なわなければならず，管理コストが高い．この問題を解決するため片端のホストからの操作のみでトンネル設定を可能にする．

トポロジの変化に対応した動的なトンネルの設定

トンネルの設定を頻繁に変えなければならない場合に自動で設定を変化させる．

通信障害発生時の代替経路の提供

通信障害が発生し通信不能に陥った時に，自動的にトンネルの設定を変更することによって，代替経路を提供する．この機能によって，より障害に強いネットワークを構築できる．

管理ポリシを反映したトンネル設定

現状の自動設定トンネルは，管理ポリシを反映したトンネル設定ができない．したがって，不適切な経路情報を流布する危険があるので導入は困難である．本研究では，管理ポリシを反映した自動設定を行なえる機構を提案する．

本研究によって実現される成果を以下に挙げる．

- トンネルの設定が容易になり，IPv6 ネットワークの管理コストが軽減される
- IPv6 の導入が容易になり，IPv6 の普及が支援される

1.3 本論文の構成

第2章では，トンネルの設定方法の問題点を明確にするために，現状の設定方法を示す．そして，そこで発生する問題について考察する．

第3章では，その解決方法として，動的トンネル制御システムであるTACSを提案する．第4章では，システムの詳細な設計を述べる．第5章では，TACSの効果を実証するために設計に基づいて実装したプロトタイプについて述べる．

TACSを実際にネットワーク上で運用し，TACSを導入しない場合との比較評価を行なった．評価については第6章で述べる．

最後に，第7章で結論を述べ，問題点を指摘し，今後の展望を示す．

1.3.1 用語の定義

本論文中で用いる用語を表1.1に定義する．

表 1.1: 用語の定義

用語	本論文中で用いられる意味
デュアルスタックホスト	IPv4，IPv6両方のプロトコルスタックを持つホスト
IPv4ネットワーク	IPv4によって構築されているネットワーク
IPv6ネットワーク	IPv6によって構築されているネットワーク
トンネル設定アドレス	トンネルの始点，終点となるアドレス
トンネル設定ホスト	トンネルの始点，終点となるホスト
上位プロトコル	トンネルを設定して，論理的なネットワークを構成するプロトコル
下位プロトコル	トンネルが利用する物理的なネットワークを構成するプロトコル

第2章

「トンネル設定の現状と問題点」

本章では現状のトンネル設定方法と，その問題について述べる．

2.1　現状のトンネル設定

2.1.1　トンネルの詳細

トンネルとは物理的なネットワークを仮想リンクとして利用する技術である．本研究におけるトンネルの定義を示す．

- あるプロトコルをデータリンク層と見做し，異種，または同種のネットワーク層プロトコルの通信を行なう．
- トンネルの両端でパケットのカプセル化と脱カプセル化を行なう．
- トンネルを形成する両端のホストでトンネルの設定を行なう．
- トンネルで接続されたホストは，トンネルを設定しているプロトコルのネットワークにおいて，隣接ホストと見做される．

IPv4とIPv6の共存ネットワークで用いられるトンネルは，以下の2種類に分類できる[3][4]．

IPv6 over IPv4トンネル

IPv4の物理ネットワーク上に，IPv6論理ネットワークを構築するためのトンネルである．IPv6パケットをIPv4ヘッダでカプセル化することによって，独立したIPv6ネットワークをIPv4ネットワークの接続性を利用して相互に接続する技術である．カプセル化と脱カプセル化の手順を，図2.1に示す．

トンネルを用て，IPv6ホスト間にIPv4ネットワークが存在していてもIPv6で通信を行なうことができる．従って，既存のIPv4ネットワークに変更を加える事なくIPv6ネットワークを新たに構築できる．IPv6 over IPv4トンネルを用いたIPv6ネットワーク構成を，図2.2に示す．

IPv4 over IPv6トンネル

IPv6の物理ネットワーク上にIPv4論理ネットワークを構築するためのトンネルである．IPv6 over IPv4トンネルとは逆に，独立したIPv4ネットワークをIPv6ネットワークの接続性を利用して相互に接続する技術である．このトンネルを用いて，IPv6ネットワーク上にIPv4論理ネットワークを構築することができる．

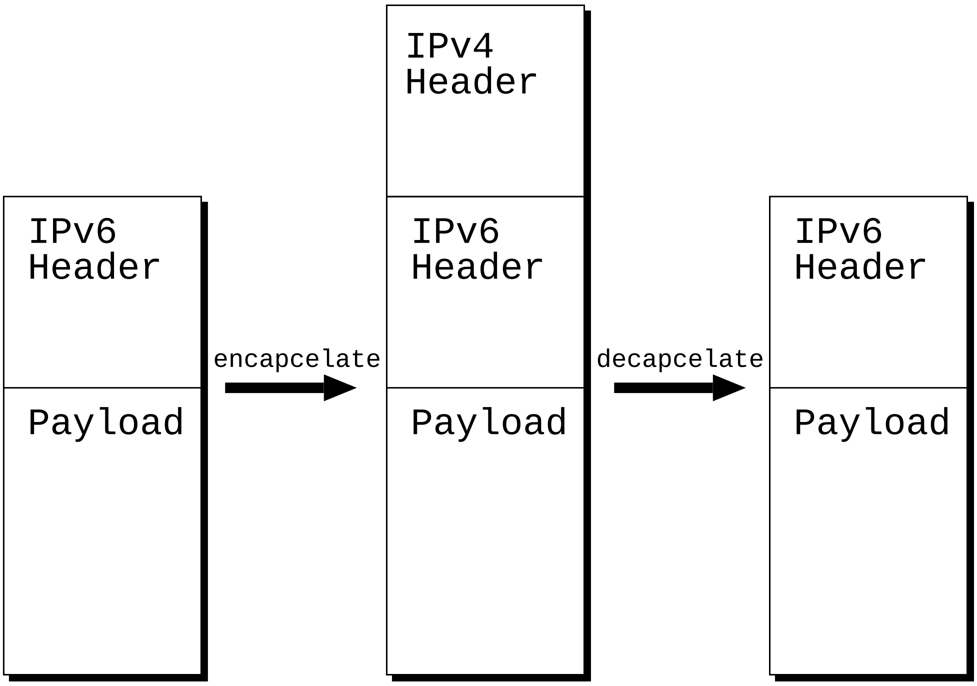

図 2.1: カプセル化と脱カプセル化

IPv4 から IPv6 への移行において，移行の初期段階では IPv6 のホストは少数であるので，IPv6 の物理ネットワークは小規模で数も少ない．したがって，IPv6 ネットワーク同士を IPv6 over IPv4 トンネルで接続することによって，IPv6 インターネットが形成される．移行の後期段階では，逆に IPv4 に対応するホストが少数になる．この場合は IPv4 over IPv6 トンネルを用いて，IPv4 インターネットを構築することになる．

2.1.2 トンネル設定方式

IPv6 への移行に用いられるトンネルの設定方式は現在 2 種類ある．

手動設定方式

両端のトンネル設定ホストから静的に設定されるトンネルである．設定や，設定の解除は管理者が直接行なう．

自動設定方式

通信を行なう際，DNS(Domain Name System)[5] に設定された IPv4 互換 IPv6 アドレス [6] をもとに，自動的にトンネルを設定する方式 [7] である．

2.2 問題が発生する状況

IPv6 が普及するに従い，トンネルは様々な状況で利用されることが想定される．その中で，現状の設定方式では対処できない状況が発生する．本節ではその状況について論じる．

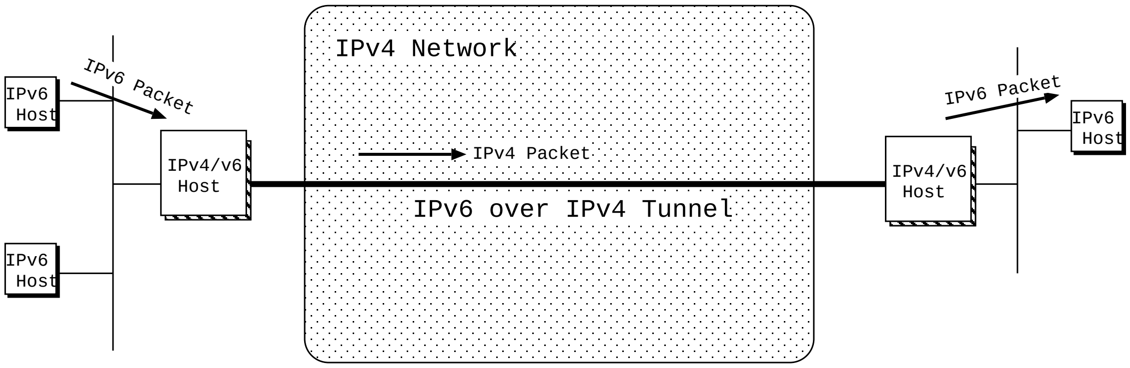

図 2.2: IPv6 over IPv4 トンネルを用いたIPv6 ネットワーク

2.2.1 IP アドレスが頻繁に変化する場合

IPv6 over IPv4 トンネルを設定するホストのIPv4 アドレスが頻繁に変化する場合，その度にトンネルの設定を変更しなければならない．具体的な例として，ダイアルアップPPP[8] で割り当てられたIPv4 アドレスでIPv6 over IPv4 トンネルを設定する場合が挙げられる．将来，IPv6 でLAN を構築し，IPv4 のダイアルアップでインターネットと接続し，IPv6 over IPv4 トンネルを設定してIPv6 の接続性を確保するような環境が発生する．IPv4 アドレスは現在枯渇状態に近付きつつあり，近い将来LAN 内のホスト全てにグローバルなIPv4 アドレスを割り振るのは不可能になるからである．この時のトポロジを図2.3に示す．

このような環境下において，手動設定方式ではダイアルアップで接続する度にトンネルを設定しなければならない．また，トンネルの両端のホストからトンネルを設定しなければならないので非常に管理コストが高い．また，自動設定方式の場合はIPv6 による通信は不可能である．その理由には，ダイアルアップPPP によって取得するIPv4 アドレスが不定なので，DNS にIPv4 互換IPv6 アドレスを登録できないこと，通信するホストが存在するネットワークと直接トンネルを設定するので管理ポリシを反映したトンネル設定ができないことが挙げられる．

また，図2.4のように，トンネルを設定するホストが移動して，別のネットワークに接続される場合も考えられる．例として，デュアルスタックのラップトップPC を様々な場所のIPv4 ネットワークに接続して，IPv4DHCP[9] によって取得したアドレスによってIPv6 over IPv4 トンネルを設定して，IPv6で通信を行なう場合が挙げられる．この場合，接続されるネットワークによってトンネル設定アドレスは変化する．手動設定方式の場合，移動する度にトンネルの設定を変更しなければならない．自動設定方式では，DHCP によって取得されるIP アドレスをDNS に登録できないので対処できない．また，管理ポリシを反映したトンネル設定ができない．

現状のトンネル設定方法で，こういった状況に対処するのは不可能である．以上のような，同一ホストにおいてトンネル設定アドレスが頻繁に変化する状況を，本論文ではアドレス変化発生状況と定義する．また，本論文では，アドレスが変化する側のホストをアドレス変化ホスト，変化しない側のホストをアドレス固定ホストと呼ぶ．

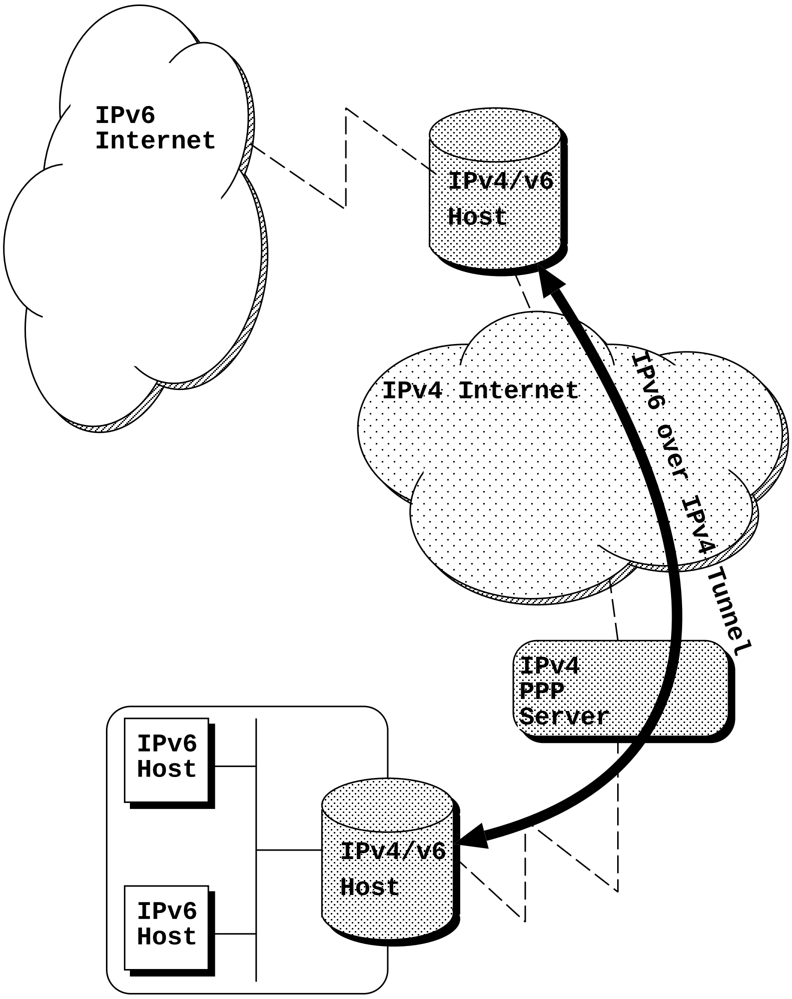

図 2.3: IPv4 ダイアルアップ PPP とトンネル

2.2.2 通信障害が発生する場合

中継ルータが停止するなどの障害が原因で外部ネットワークへの接続性がなくなった場合，トンネルを設定することで接続性を確保することができる．具体的には IPv6 ネットワーク内で発生した通信障害を，IPv6 over IPv4 トンネルを設定することで回避する状況が考えられる．この状況を図 2.5 に示す．

しかし，手動設定方式の場合，ネットワークの管理者が障害を検知してトンネルを設定するまで接続性は回復しない．したがって，通信不能状態が長時間続いてしまう危険がある．また，障害が復旧すれば，管理者は設定したトンネルの設定を解除する必要がある．しかし，管理者が直接設定を変更しなければならいので管理コストが高い．そのため現状の設定方法でトンネルを用いて障害を回避することは困難である．また，自動設定方式の場合，通信障害を回避する事は可能だが，ネットワークの管理ポリシを反映したトンネルの設定ができない．したがって，通信障害を自動設定方式のトンネルで回避する方法は望ましくない．

このような，通信障害発生時に問題を回避できない状況を本論文では障害回避不能状況と定義

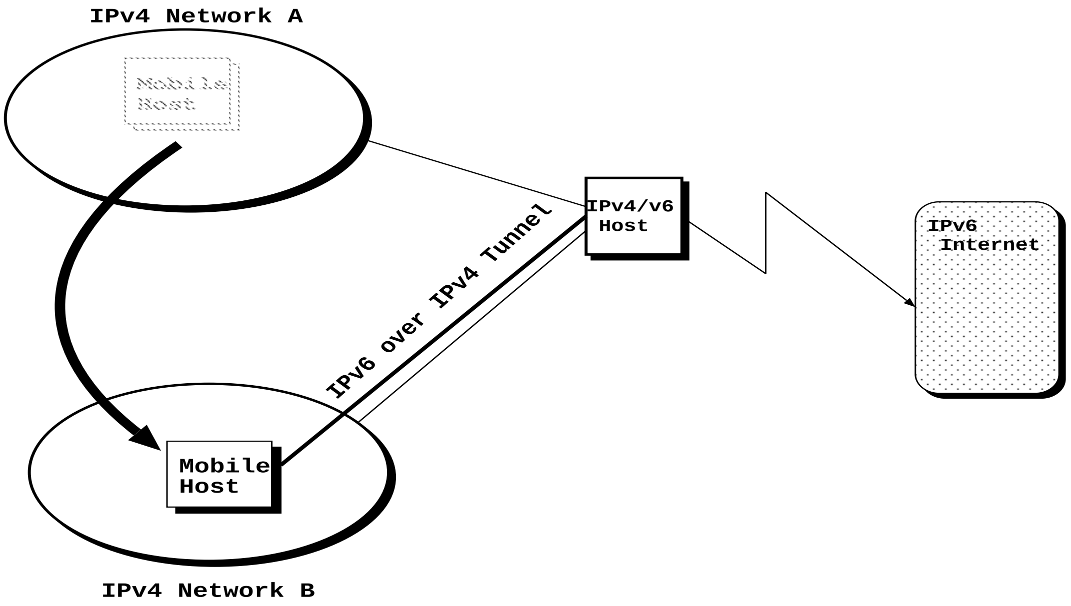

図 2.4: ネットワーク的に移動した場合

する．また，本論文では，通信障害時に備え通信状態を監視し，障害発生時にトンネルを設定するホストを通信障害発生ホスト，通信障害発生ホストのトンネルの設定を受け付けるホストを通信障害回避ホストと呼ぶ．

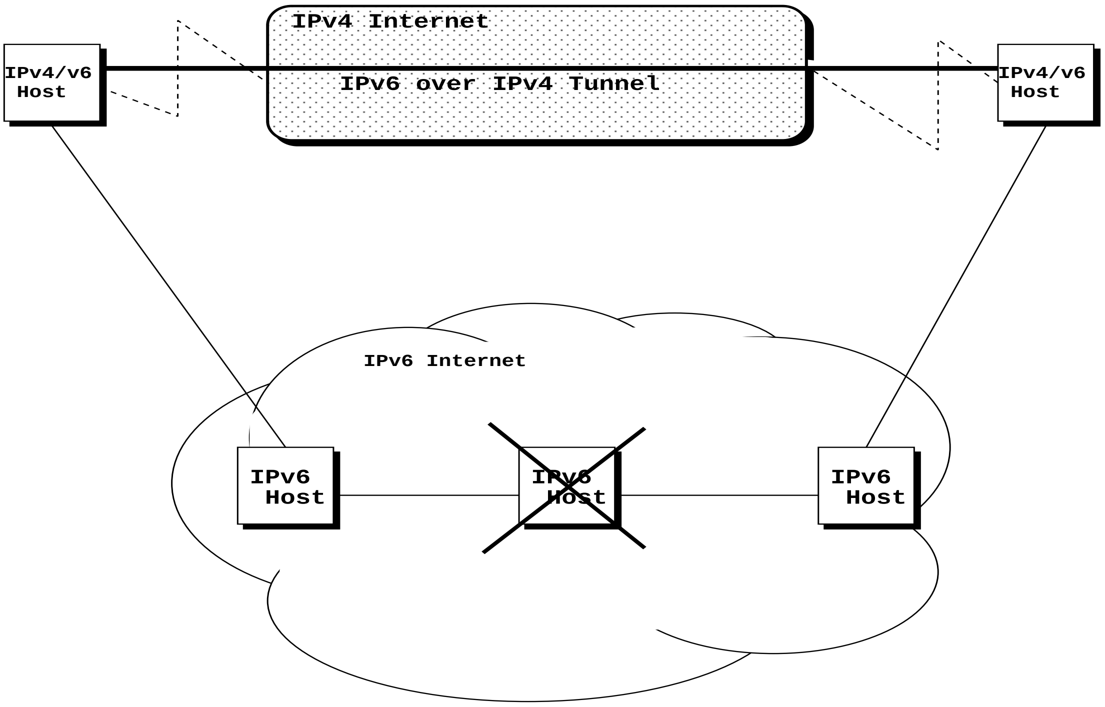

図 2.5: 通信障害を回避する場合

第3章

「解決方法」

本章では、「アドレス変化発生状況」と「障害回避不能状況」を解決するためのトンネル自動設定システムを提案する。本システムにより、ネットワーク管理者の負担を軽減させると共に、安定したネットワーク運用を提供できる。

3.1 問題の解決手法

現状のトンネル設定方式を用いた場合に発生する、2つの状況の解決手法を示す。

アドレス変化発生状況

頻繁にトンネル設定アドレスが変化する状況は2.2.1で述べた。この状況を解決するためには、トンネル設定ホストがIPv4インターネットに接続した際自動的にIPv6 over IPv4トンネルが設定され、接続を切断した際トンネルを解除する必要がある。この時に必要な機能の要件を以下に示す。

1. IPv4での通信状況を監視する
2. 通信が可能になると同時にトンネルを設定する
3. 通信が不可能になると同時にトンネルの設定を解除する

障害回避不能状況

通信障害発生時にトンネルによって通信路を確保できることは2.2.2で述べた。しかし、通信障害をネットワーク管理者が検知してから手動で設定していては時間がかかり、管理コストも非常に高くなってしまう。この問題の解決のためには、障害発生時でも通信路を確保したいホストとの通信を可能にすればよい。この時に必要な機能の要件を以下に示す。

1. IPv6での通信状態を常に監視する
2. 通信が途絶すると同時にトンネルを設定する
3. 通信が復旧すると同時にトンネルの設定を解除する

3.2 トンネル自動設定システムの提案

3.1で論じた問題解決方法の実現のため、本研究ではトンネルの設定設定システム(Tunnel Auto Configuration System:TACS)を提案する。TACSが実現する機能を以下に示す。

片端トンネル設定機能

トンネル設定ホスト間でトンネル設定のためのメッセージを交換する。これによって、片端

のトンネル設定ホストからの操作のみでトンネルが設定、または設定解除される。手動設定方式は両端から設定しなければならないことと比較して、トンネルの設定の負担が軽減される。

トンネル自動設定機能
: 通信状態を監視して、状態の変化に応じて動的にトンネル設定と設定解除を行なうことができる。この機能と片端トンネル設定機能を組み合わせて、頻繁なトンネルの設定変更に対応する。

トンネル認証機能
: 不適切な経路を流さないために、管理者が許可しないトンネルは設定できないようにする。この機能によって、管理ポリシを反映したトンネル設定ができる。

TACSによって、アドレス変化発生状況と障害回避不能状況が解決できる。それぞれの状況下でのTACSの動作を以下に示す。

アドレス変化発生状況
: アドレス変化ホストがもう一端への下位プロトコルの通信状態を監視し、通信可能になれば、片端トンネル設定機能によってトンネルを設定する。通信不可能になれば、トンネルの設定を解除する。

障害回避不能状況
: 通信障害監視ホストが上位プロトコルの通信状態を監視し、通信不可能になれば、片端トンネル設定機能によってトンネルを設定する。通信可能になれば、トンネルの設定を解除する。また、不適切な経路を流さないために、許可されていないトンネルは、認証機構によって設定できないようにする。

TACSを導入することによって、トンネルを手動で設定することなく自動的に管理ポリシを反映したトンネルを設定できる。従って、

- 頻繁なトンネルの設定変更
- 通信障害発生時のトンネルによる障害回避

を行なう場合に、管理コストを削減できる。以上の理由により、トポロジの変化に対応した動的なトンネルの設定と、通信障害発生時の代替経路の自動提供が可能になる。

第4章

「トンネル自動設定システムの設計」

本章では，「頻繁なIPv4アドレスの変化への対応」と「障害発生時における自動的障害回避」を実現するトンネル自動設定システム(Tunnel Auto Configuration System:TACS)の設計について述べる．まずシステムの設計目標を示し，次に具体的な設計について論じる．

4.1 システム設計目標

トンネルの設定を自動化し，管理コストを軽減させるためのシステムには，以下に挙げる条件が必要である．

- トンネルの自動設定，インターフェースの自動管理
- 管理ポリシを反映したトンネル設定

この条件を満たすシステムとして，トンネル自動設定システム(Tunnel Auto Configuration System:TACS)を提案する．これは，動的にトンネルの設定を制御したい2ホスト間上で動作し，ネットワークの状態に応じてトンネルの設定と設定の解除を行なうものである．

本システムの設計目標として，以下の項目を設定した．

トンネルインターフェースの自動制御

ネットワークの状態を監視して，トンネルの自動設定および自動解除を行なう．

管理ポリシを反映したトンネル設定

悪意をもった利用者が権限のないホストにトンネルを設定すると，不適切な経路情報を流されるなどの望ましくない状況が発生する．従って，管理ポリシを反映したトンネル設定をする必要がある．

4.2 システム設計の概要

第4.1節で述べた目標にしたがって，TACSを設計した．本節では，システムの設計の概要について述べる．

4.2.1 動作環境

TACSは，IPv6 over IPv4トンネルとIPv4 over IPv6トンネルを自動的に制御する．サーバとクライアントは共にデュアルスタックホストである必要がある．

4.2.2　システムの概要

本システムは，サーバ，動的トポロジ対応クライアント，自動障害回避クライアントの3つで構成される．サーバはクライアントの要求に応じてトンネルを設定する．サーバと動的トポロジ対応クライアントで構成するシステムを動的トポロジ対応システムと呼ぶ．サーバと自動障害回避クライアントで構成されるシステムを自動障害回避システムと呼ぶ．

アドレス変化発生状況に対応するためには，動的トポロジ対応システムを用いる．アドレス固定ホストがサーバとなり，アドレス変化ホストが動的トポロジ対応クライアントとなる．動的トポロジ対応システムの動作概要を図 4.1に示す．

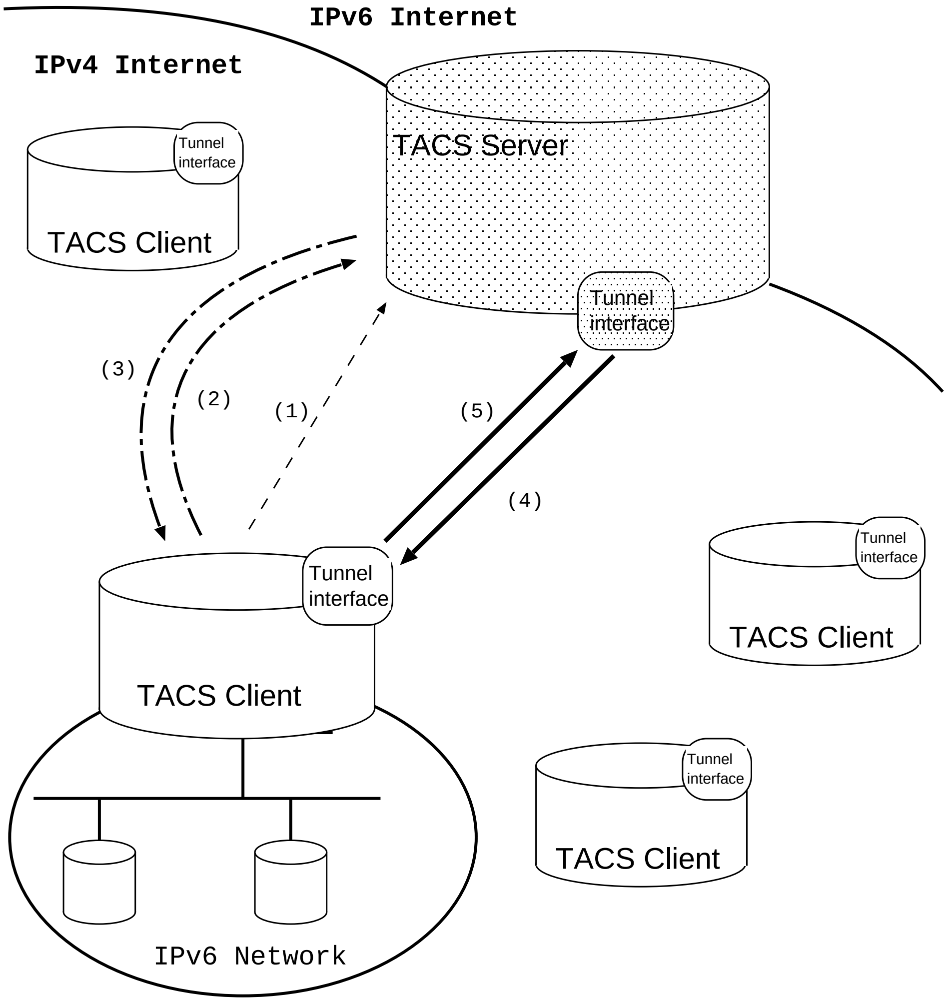

図 4.1: 動的トポロジ対応システムの概要

1. クライアントが，サーバへの下位プロトコルの通信が可能であることを検知する．
2. クライアントがサーバにトンネル設定要求メッセージを送信する．
3. サーバがクライアントにトンネル設定許可メッセージを送信する．

4. クライアントがトンネルを設定する．

5. サーバがトンネルを設定する．

障害回避不能状況に対応するためには，自動障害回避システムを用いる．通信障害復旧ホストがサーバとなり，通信障害監視ホストが自動障害回避クライアントとなる．自動障害回避システムの動作概要を図 4.2に示す．

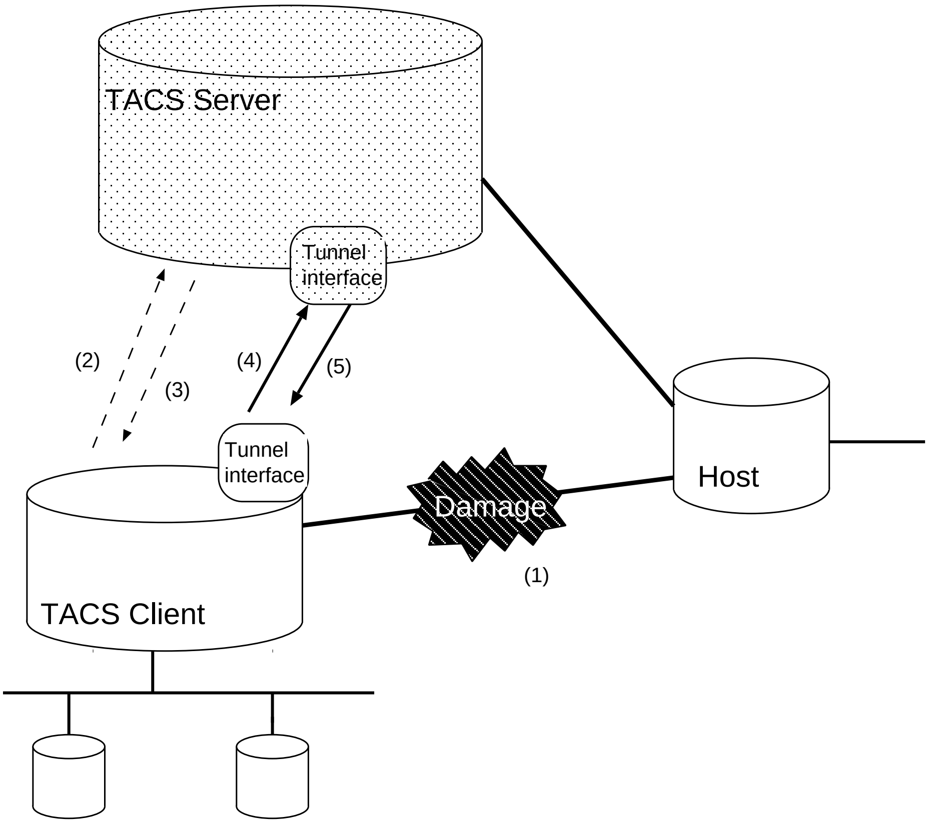

図 4.2: 自動障害回避システム概要

1. クライアントが，上位プロトコルにおいて通信障害が発生したことを検知する．
2. クライアントがサーバにトンネル設定要求メッセージを送信する．
3. サーバがクライアントにトンネル設定許可メッセージを送信する．
4. クライアントがトンネルを設定する．
5. サーバがトンネルを設定する．

4.3 TACSを導入したIPv6ネットワーク管理モデル

本節では，TACSを用いたIPv6ネットワークモデルの定義を行なう．

4.3.1 動的トポロジ対応システムのモデル

動的トポロジ対応システムを導入したネットワークのモデルを図 4.3に示す．

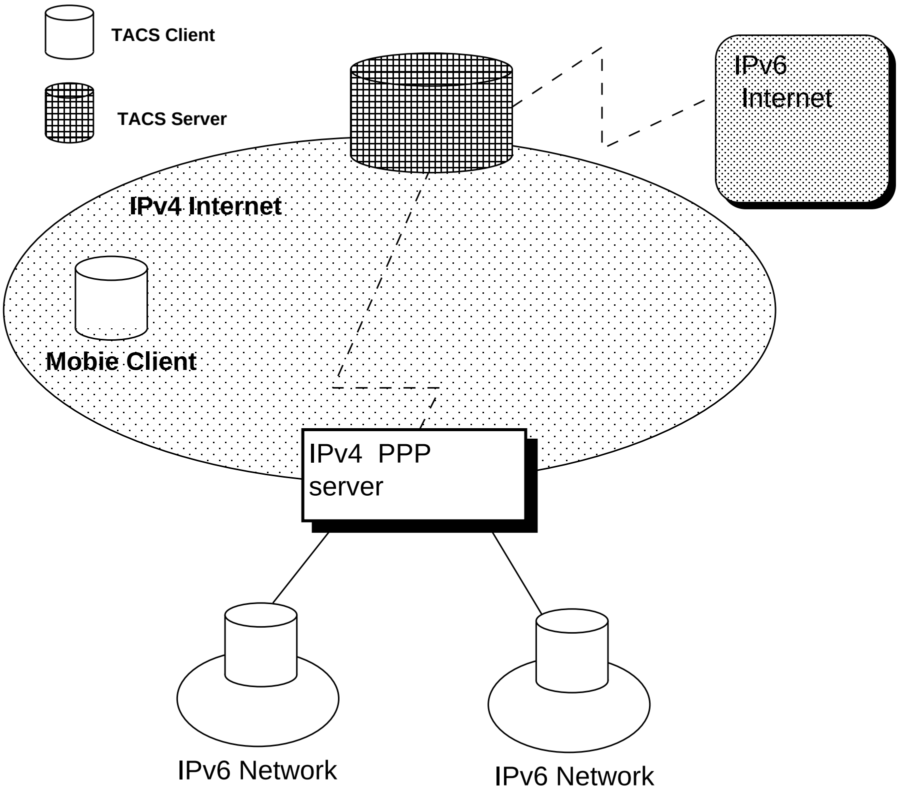

図 4.3: 動的トポロジ対応システムのモデル

多数のクライアントが一つのサーバに接続される．クライアントは容易にIPv6 に接続することができる．

クライアントが多数になると手動の設定ではサーバの設定が困難だが，TACS を導入することによって多数の，小規模なネットワーク，もしくは単一のホストを接続することができる．

4.3.2 自動障害回避システムのモデル

自動障害回避システムを導入したネットワークのモデルを図 4.4に示す．

このモデルは自動障害回避クライアントがIPv4 とIPv6 の両方の物理的な接続性を有している必要がある．

全くプレフィックスの異なるホストに対してトンネルを設定すると，不適切な経路を流してしまう危険がある．したがって，経路の集約[11] が可能なホストをサーバとする必要がある．

TACS を用いない場合，障害が発生してから管理者がそれを検知してトンネルを設定するまで通信は回復しない．TACS を用いれば，障害が発生しにた直後にトンネルを自動で設定するので，通信不能な時間が短い．

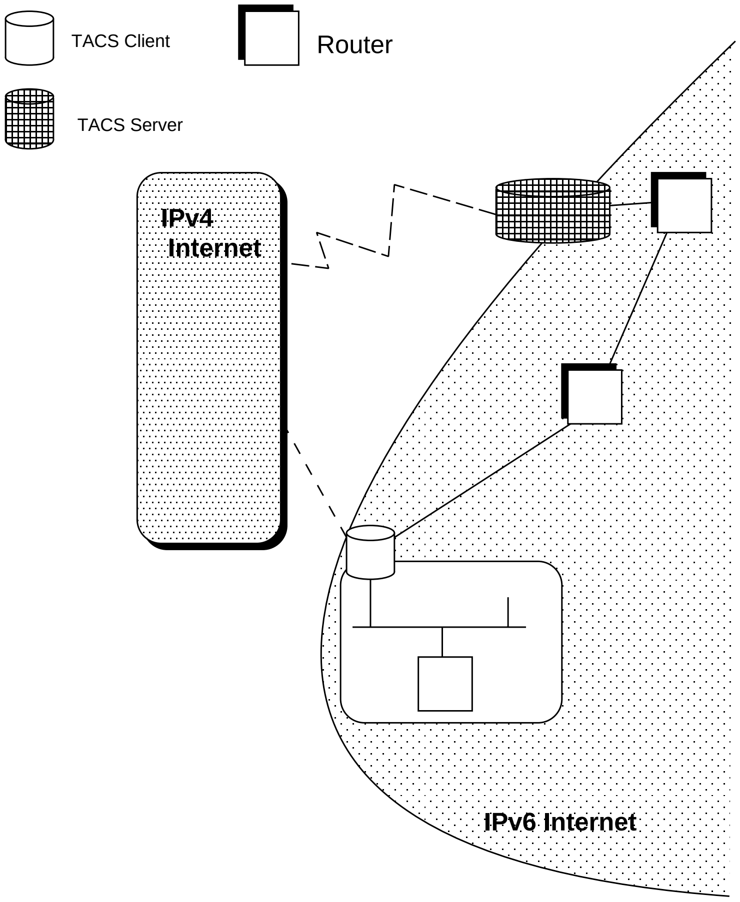

図 4.4: 自動障害回避システムのモデル

4.4 トンネル自動制御プロトコル

片端トンネル設定機能を実現するために，トンネル自動制御プロトコル(Tunnel Auto Configuration Protocol:TACP)を定義する．

TACPの設計目標を以下に示す．

片端ホストからのみの操作によるトンネル設定

サーバ/クライアント間のUDP[10]を用いたメッセージ交換により，片端ホストからの操作のみでトンネルを設定する．具体的な動作は4.4.2で述べる．

4.4.1 TACPメッセージ

TACPメッセージフォーマットを図 4.5に示す．

TACPメッセージは，4bitのメッセージタイプ領域，4bitのメッセージコード領域，32bitのIPアドレス領域，16bitのインターフェース番号領域で構成される．インターフェース番号領域はオプションである．

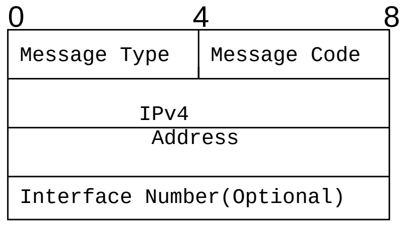

図 4.5: TACP メッセージフォーマット

表 4.1: 設定要求のメッセージコード

message type	message code	説明
トンネル設定要求	0	通常要求
	1	インタフェース指定要求

トンネル設定要求:通常要求

自ホストのIPv4 アドレスのみを設定先に通知する．

トンネル設定要求 :インターフェース指定要求

要求ホストが自ホストのIPv4 アドレスと，相手ホストのトンネルのインターフェースを指定し，送信する．

表 4.2: 設定応答のメッセージコード

message type	message code	説明
トンネル設定応答	0	設定許可
	1	設定拒否（認証失敗）
	2	設定拒否（インタフェース割当失敗）

トンネル設定応答 :設定許可

要求を受けたトンネル設定ホストが，設定を許可する場合に送信する．送信した後に，要求を送信したホストへのトンネルを設定する．要求したホストがこのメッセージを受信し，応答したホストにトンネルを設定し，通信路が確保される．

トンネル設定応答 :設定拒否（認証失敗）

要求メッセージを送信したホストが認証に失敗した場合に，受信したホストが送信する．トンネルは設定されない．

トンネル設定応答 :設定拒否（インターフェース割当失敗）

相手先ホストのトンネルインタフェースが全て使用されている場合，または指定したインターフェースが使用されていた場合に，相手ホストが送信する．トンネルは設定されない．

表 4.3: 切断要求のメッセージコード

message type	message code	説明
トンネル切断要求	0	発信元要求
	1	タイムアウト

トンネル切断要求　:発信元要求

トンネルの切断を相手ホストに要求する．要求するホストは，送信したと同時にトンネルを切断する．

トンネル切断要求　:タイムアウト

一定時間以上通信が行なわれい場合に送信される．要求するホストは送信と同時にトンネルを切断する．

表 4.4: 切断応答のメッセージコード

message type	message code	説明
トンネル切断応答	0	通常応答

トンネル切断応答

トンネル切断要求を受信したホストが相手ホストに送信する．送信した後にトンネルを切断する．

4.4.2　TACP の動作

TACP サーバの状態遷移を図 4.6に示す．

サーバはクライアントからの要求を待ち，要求メッセージを受信すると，応答メッセージを送信する．設定拒否メッセージを送信した場合，初期状態に戻り，再び要求を待つ．設定許可メッセージを送信した場合，トンネルを設定する．切断要求メッセージを送信した場合，クライアントからの応答を待ってトンネルを解除する．切断要求メッセージを受信した場合，応答メッセージを送信しトンネルの設定を解除する．

TACP クライアントの状態遷移を図 4.7に示す．

クライアントは設定要求メッセージを送信し，サーバからの応答を待つ．設定許可メッセージを受信した場合，トンネルを設定する．設定拒否メッセージを受信した場合，初期状態に戻る．切断要求メッセージを送信した場合，サーバからの応答を待ってトンネルを解除する．切断要求メッセージを受信した場合，応答メッセージを送信し，トンネルの設定を解除する．

4.5　システムの構成要素

TACS サーバとクライアントは，それぞれ，以下のモジュールによって構成される．

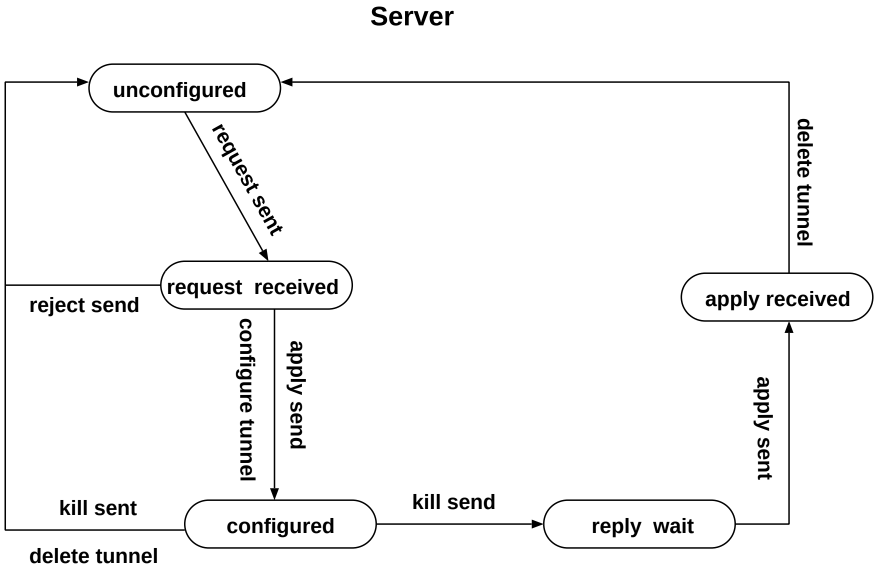

図 4.6: サーバの状態遷移

- TACP モジュール
- ネットワーク監視モジュール
- トンネル設定モジュール

4.5.1 TACS サーバ

サーバの設計目標を以下に示す．

2 種類のクライアントに対応する
サーバは自動設定クライアント，障害復旧クライアントの両方の要求に対応する．

認証機構の導入
サーバの許可しないホストはトンネルを設定できないようにする．

複数のトンネルインターフェースの管理
複数のクライアントからの接続を可能にする．またトンネルインターフェースを効率的に割り当てる．

TACS サーバの構成図を図 4.8に示す．

TACP モジュール
TACP に基づいてクライアントとメッセージを交換する．
要求メッセージを受信し認証する．その結果に応じた応答メッセージをクライアントに送信する．設定を許可する場合，トンネル設定の命令をトンネル設定モジュールに送る．

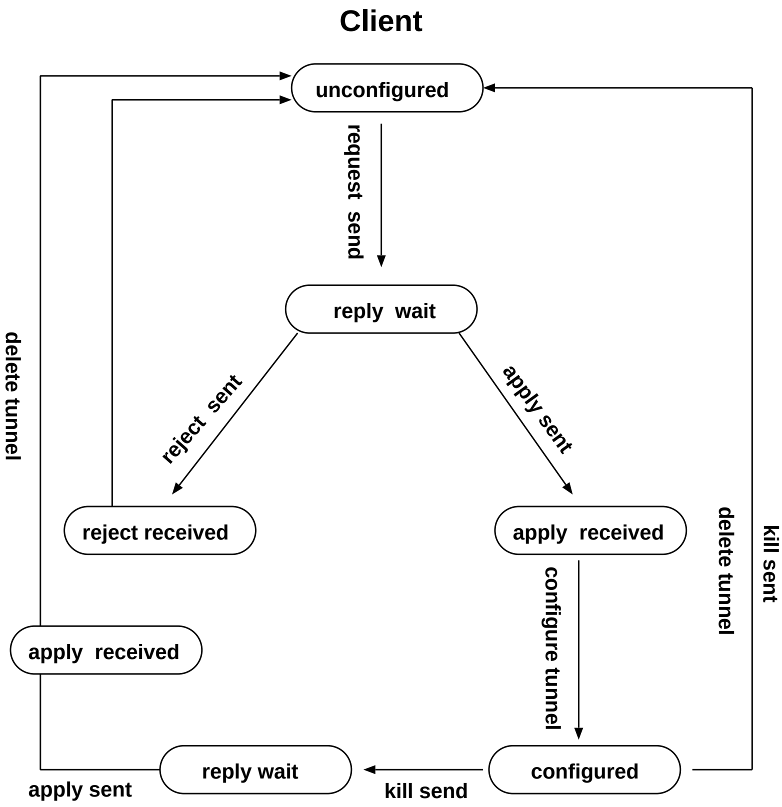

図 4.7: クライアントの状態遷移

ネットワーク監視モジュール
トンネルが設定されたクライアントとのIPv4のネットワーク状態を監視する．ネットワークが切断された場合，トンネル設定解除の命令をトンネル設定モジュールに送る．

トンネル設定モジュール
TACPモジュールとネットワーク監視モジュールからの命令にしたがってトンネルの設定と設定解除を行なう．

4.5.2 動的トポロジ対応クライアントの設計

動的トポロジ対応クライアントは，ダイアルアップPPPを用いてトンネルを設定する場合や，ネットワーク的に移動するホストでトンネルを設定したい場合に用いる．サーバと通信が可能になった場合に，片端トンネル設定機能によってトンネルを設定する．サーバとの通信が途絶した場合，トンネルの設定を解除する．動的トポロジ対応クライアントの構成図を図 4.9に示す．

TACPモジュール
TACPに基づいてサーバとメッセージを交換する．ネットワーク監視モジュールから要求メッセージ送信命令を受け取り，サーバに送信する．サーバから設定許可メッセージを受信すると，トンネル設定命令をトンネル設定モジュールに送る．認証失敗メッセージを受信す

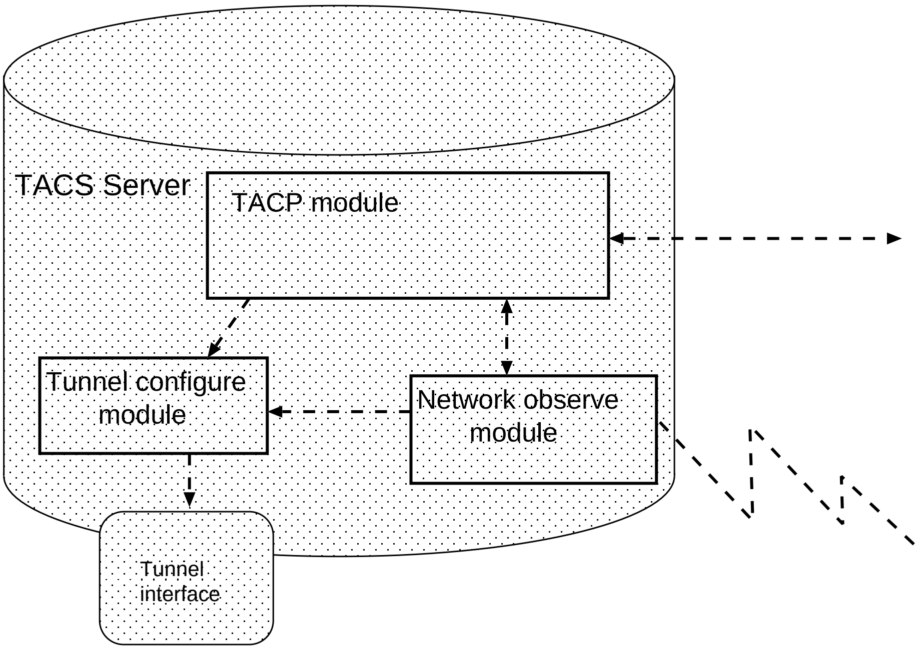

図 4.8: TACS サーバの設計

ると，プログラムを終了させる．インターフェース割当失敗メッセージを受信すると，一定時間後に要求メッセージを再送する．

ネットワーク監視モジュール

サーバとのIPv4での通信状態を恒常的に監視する．通信不能な状態から可能な状態に変化すると，要求メッセージ送信命令をTACPモジュールに送る．通信可能な状態から不可能な状態に変化すると，トンネル設定解除命令をトンネル設定モジュールに送る．

トンネル設定モジュール

TACPモジュールからトンネル設定命令を受け取ると，サーバにトンネルを設定する．ネットワーク監視モジュールからトンネル設定解除命令を受け取ると，サーバとのトンネルの設定を解除する．

4.5.3 自動障害回避クライアントの設計

自動障害回避クライアントは，IPv6の通信を途絶した場合に自動的にトンネルを設定して，障害の回避を図る場合に用いる．通信障害発生時も通信を続けたいマシンとの通信状態を定期的監視し，通信が不可能になった場合にサーバと片端トンネル設定機能を用いてトンネルを設定する．通信が回復した場合，トンネルの設定を解除する．自動障害回避クライアントの構成を図 4.10に示す．

TACPモジュール

動的トポロジ対応クライアントのTACPモジュールと動作は同じである．

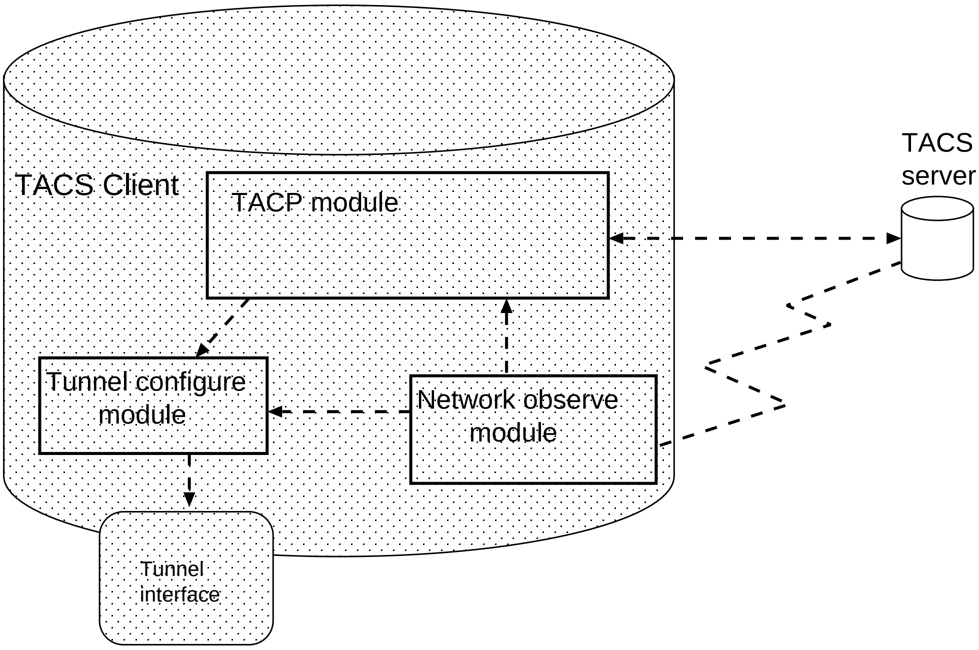

図 4.9: 動的トポロジ対応クライアントの概要

ネットワーク監視モジュール
特定のホストとのIPv6での通信状態を恒常的に監視する．通信可能な状態から不可能な状態に変化すると，要求メッセージ送信命令をTACPモジュールに送る．通信不可能な状態から可能な状態に変化すると，トンネル設定解除命令をトンネル設定モジュールに送る．

トンネル設定モジュール
TACPモジュールからトンネル設定命令を受け取ると，サーバにトンネルを設定する．ネットワーク監視モジュールからトンネル設定解除命令を受け取ると，サーバとのトンネル設定を解除する．

4.5.4　まとめ

TACSのによって提供される利点を以下にまとめる．

IPv6での接続性を容易に提供できる
IPv4のダイアルアップPPPやDHCPでIPv6ネットワークに接続できるので，IPv6ネットワークへの接続が容易になる．

IPv6のネットワークをより安定したものにできる
自動障害回避システムにより通信障害発生時にも接続性を保つことができる．従って，IPv6ネットワークはより障害に強くなり，安定する．

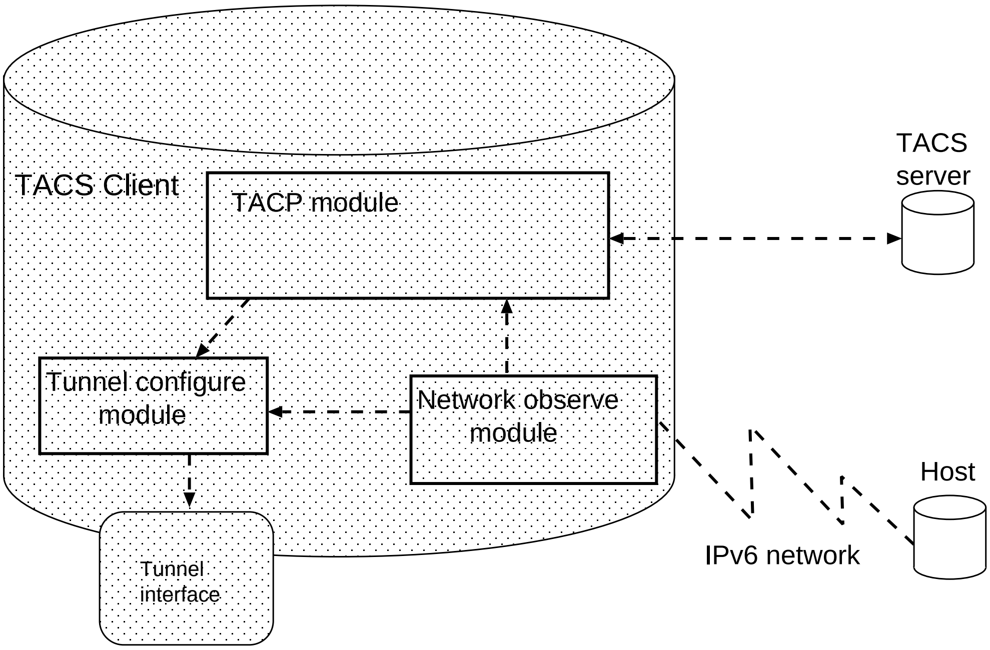

図 4.10: 自動障害回避クライアントの概要

第5章

「トンネル自動設定システムの実装」

第4章で述べた設計に基づき，TACSのプロトタイプ実装を行なった．本章ではサーバ，動的トポロジ対応クライアント，自動障害回避クライアントの実装について述べる．

5.1 プロトタイプの機能

TACSの導入によりネットワークの管理コストが軽減されることを実証するため，TACSプロトタイプを実装した．

TACSプロトタイプは以下の機能を持つ．

IPv6 over IPv4 トンネル自動設定，および自動設定解除

トポロジの変化に対応した動的なトンネル設定と，通信障害発生時の代替経路の提供が実現できるかを評価するために，IPv6 over IPv4 トンネルの自動設定と自動設定解除機能を実装した．

複数トンネルインターフェース管理

TACSサーバが複数のクライアントの要求に対応するために，複数トンネルインターフェースの管理を可能にした．

トンネル要求の認証(自動障害回避システムのみ)

サーバは，クライアントのトンネル設定アドレスを検査し，許可しないクライアントからの要求は拒否する．この方式は動的トポロジ対応システムには対応できないので，自動障害回避システムのみに採用した．

通信はUDPを用いて行なわれる．

5.2 動作環境

以下の環境でプロトタイプの実装を行なった．

OS FreeBSD 2.2.7-Release

IPv6 スタック KAME*-981102-snapshot

トンネル設定モジュールはKAME-981102-snapshotに依存した実装になっている．

---

*http://www.kame.net/

5.3 構成情報

サーバ/クライアント間の通信に用いられるtacs_msg構造体について説明する．

5.3.1 TACPメッセージ

構造体 tacs_msg を図 5.1に示す．

```
struct tacs_msg {
  u_char msg_type;
  u_char msg_code;
  union{
    struct configure_request_msg c_request;
    struct configure_reply_msg c_reply;
    struct kill_request_msg k_request;
    struct kill_reply_msg k_reply;
  }type;
};
```

図 5.1: tacs_msg構造体

tacs_msg構造体の説明を表 5.1に示す．

表 5.1: 設定要求のメッセージコード

メンバ名	変数型	説明
msg_type	u_char	メッセージタイプを指定．
msg_code	u_char	メッセージコードを指定．
c_request	struct tacs_data_plus	設定要求メッセージのデータ．自ホストのIPv4アドレスと相手のトンネルインターフェース番号を指定する．相手ホストはこの情報に基づきトンネルの設定を行なう．
c_reply	struct tacs_data	要求応答メッセージのデータ． 自ホストのIPv4アドレスを指定する．
k_request	struct tacs_data	切断要求メッセージのデータ． 自ホストのIPv4アドレスを指定する．
k_reply	struct tacs_data	切断応答メッセージのデータ． 自ホストのIPv4アドレスを指定する．

tacs_data構造体を図 5.2に示す．

```
struct tacs_data{
  struct in_addr myaddr;
};
```

図 5.2: tacs_data 構造体

tacs_data 構造体は，トンネル設定応答メッセージ，トンネル切断要求メッセージ，トンネル切断応答メッセージのデータ部分である．データ部はアドレスを含むので，in_addr 構造体が含まれている．

tacs_data_plus 構造体を図 5.3に示す．

```
struct tacs_data_plus{
  struct in_addr myaddr;
  int interface;
};
```

図 5.3: tacs_data_plus 構造体

tacs_data_plus 構造体は，トンネル設定要求メッセージのデータ部分である．トンネル設定要求メッセージは，相手のインターフェースを指定することができるので，in_addr 構造体の他にインターフェース番号を構造体の中に含んでいる．

5.4 TACS サーバの実装

4.5.1で述べた設計に基づき，TACS サーバを実装した．

5.4.1 設定情報の読込

トンネルの自動設定のために必要な情報を tacs.conf ファイルに登録する．登録する情報を表 5.2に示す．

登録例を図 5.4に示す．

この例では，サーバのトンネル設定アドレスが203.178.140.47 で，トンネルインターフェースはgif0 ~ gif3 を用いる．また，クライアントは203.178.140.44，203.178.140.56，133.27.4.145 からの要求を許可する設定である．この設定は readconffile() 関数によって，メモリに格納される．

5.4.2 TACP モジュールの実装

サーバのTACP モジュールは起動されると，ポート番号8744 番においてクライアントからの要求に備え待機する．設定要求メッセージを受信すると子プロセスをfork する．fork した子プロセ

表 5.2: tacs.confに登録する情報

server address	サーバ側のトンネル設定アドレス
interface name	自動設定に用いるトンネルインターフェース名
interface number	自動設定に用いるトンネルインターフェースナンバ
client address	設定を許可するクライアントのIPv4アドレス．何も登録しない場合は，全ての要求を許可する．

```
server address
203.178.140.47

interface name
gif

interface number
0
1
2
3

client address
203.178.140.44
203.178.140.56
133.27.4.145
```

図 5.4: tacs.conf登録例

スで認証モジュールを呼び出し，認証する．そして，その結果に基づいた応答メッセージの送信を行なう．

TACPモジュールを構成する関数のうち，主要なものを以下に挙げる．

authenticate() 受信したメッセージを認証する関数．引数としてクライアントのアドレスを含むin_addr構造体を渡す．クライアントのアドレスと，readconffile()で読み込んだ設定を許可するアドレスのリストを比較する．また，使われているトンネルンターフェースを検査して設定が可能かどうか調べる．authenticate()の返り値を表5.3に示す．

make_send_msg() 送信する応答メッセージを作成する関数．引数として，認証の結果と，受信したメッセージのtacs_msg構造体を渡す．成功すると応答メッセージを返す．失敗すると1を返す．

send_msg() メッセージをUDPとして送信する関数．引数としてmake_send_msgが作成したtac-

表 5.3: authenticate()の返り値

返り値	説明
0	設定許可
1	認証失敗
2	インタフェース割当失敗

s_msg 構造体と，受信したメッセージの中の in_addr 構造体を渡す．

5.4.3 ネットワーク監視モジュールの実装

サーバのネットワーク監視モジュールは，クライアントとのIPv4のネットワーク状態を恒常的に監視する．クライアントとトンネルが設定された時点で起動される．一定時間毎にクライアントにICMP echo requestを送信することによって，IPv4のネットワークの状態を監視する．通信が途絶した場合，トンネル設定を解除する命令をトンネル設定モジュールに送る．

ネットワーク監視モジュールを構成する関数のうち，主要なものを以下に挙げる．

observe() ICMP echo requestを送信し，ICMP echo replyを受信する関数．クライアントのIPアドレスが入った in_addr 構造体を引数としてわたす．

受信に成功すれば0，失敗すれば1を返す．

ck_status4() IPv4ネットワーク状態の変化をチェックする関数．

observe()の結果を引数に渡す．前回のobserve()の結果と比較して値を返す．返り値は表 5.4 のとおりである．

表 5.4: ck_status4()の返り値

返り値	説明
0	変化なし
1	通信不可能→通信可能
2	通信可能→通信不可能

5.4.4 トンネル設定モジュールの実装

TACPモジュール，ネットワーク監視モジュールからの命令によって，実際にトンネルの設定と設定の解除を行なうモジュールである．トンネル設定モジュールを構成する関数のうち，主要なものを以下に挙げる．

tun_configure() TACPモジュールからのトンネル設定命令を受けて，トンネルを設定する関数．readconffile()から読み込んだ自ホストのIPアドレスと，クライアントのIPアドレスと，設定するトンネルインターフェースを引数に渡す．

tun_kill() ネットワーク監視モジュールからのトンネル設定解除命令を受けて，トンネルの設定を解除する関数．設定を解除するトンネルインターフェースを引数に渡す．

5.5 動的トポロジ対応クライアントの実装

4.5.2で述べた設計に基づき動的トポロジ対応クライアントの実装を行なった．本節で動的トポロジ対応クライアントのTACPモジュール，ネットワーク監視モジュール，トンネル設定モジュールの実装について述べる．

5.5.1 設定情報の読込

トンネルの自動設定のために必要な情報をtacc.confファイルに登録する．登録する情報を表5.5に示す．

表 5.5: tacc.confに登録する情報

server address	サーバ側のトンネル設定アドレス
interface name	自動設定に用いるトンネルインターフェース名
interface number	自動設定に用いるトンネルインターフェースナンバ
client address	トンネル設定アドレス，または，トンネル設定アドレスが設定されているネットワークインターフェース名

登録例を図5.5に示す．

```
server address
203.178.140.47

interface name
gif

interface number
0

client address
ep0
```

図 5.5: tacc.conf登録例

この例では，サーバのトンネル設定アドレスに203.178.140.47，トンネルインターフェースはgif0を用いる．ep0に設定されているアドレスを用いてトンネルを設定する．この設定はreadconfflie関数によって，メモリに格納される．

5.5.2 ネットワーク監視モジュールの実装

動的トポロジ対応クライアントのネットワーク監視モジュールは，サーバのIPv4のネットワーク状態を監視する．クライアント起動と同時に起動される．一定時間毎にサーバにICMP[12] echo request を送信することによって，IPv4 のネットワークの状態を監視する．通信が可能になった場合，設定要求メッセージ送信命令をTACP モジュールに送る．サーバとの通信が途絶した場合，トンネル設定解除命令をトンネル設定モジュールに送る．

ネットワーク監視モジュールを構成する関数のうち，主要なものを以下に挙げる．

observe() ICMP echo request を送信し，ICMP echo reply を受信する関数．クライアントのIPアドレスが入ったin_addr構造体を引数としてわたす．受信に成功すれば0，失敗すれば1を返す．

kill_order() トンネル設定モジュールに，トンネル設定解除命令を出す関数．observe()が受信に失敗するとトンネル解除命令を出す．observe()関数の結果を引数に渡す．

5.5.3 TACP モジュールの実装

動的トポロジ対応クライアントのTACP モジュールは，IPv4 ネットワークが通信不可能から可能になった場合に起動される．サーバに設定要求メッセージを送信し，応答メッセージを受信する．受信したメッセージを処理して，終了する．TACP モジュールを構成する関数のうち，主要なものを以下に挙げる．

make_send_msg() 送信する応答メッセージを作成する関数．引数として自ホストのIP アドレスと，インタフェースを指定する場合はインターフェースナンバを渡す．成功すると，応答メッセージを返す．失敗すると1を返す．

send_msg() メッセージをUDP として送信する関数．引数としてmake_send_msg() が作成したtacs_msg 構造体と，サーバのアドレスを渡す．

送信に成功すれば0，失敗すれば1を返す．

recv_msg() サーバからの応答メッセージを受信する関数．受信するメッセージコードと，それによって行なう処理を表 5.6に示す．

表 5.6: 受信したメッセージコードとその処理

設定許可	トンネル設定モジュールにトンネル設定命令を出す．
設定拒否(認証失敗)	自動設定クライアントを終了する．
設定拒否(インターフェース割当失敗)	要求メッセージを再送する

5.5.4 トンネル設定モジュール

TACP モジュールとネットワーク監視モジュールからの命令によって，実際にトンネルの設定と設定の解除を行なうモジュール．動作はサーバのトンネル設定モジュールと同じである．

5.6 自動障害回避クライアントの実装

4.5.3で述べた設計に基づき，障害自動回避クライアントの実装を行なった．本節で障害自動回避クライアントのTACPモジュール，ネットワーク監視モジュール，トンネル設定モジュールの実装について述べる．

5.6.1 設定情報の読込

設定ファイル tacc.conf の書式は自動設定クライアントと同一である．自ホストのアドレスを指定する点が自動設定クライアントとは異なる．

登録する情報を表 5.7に示す．

表 5.7: tacc.conf に登録する情報

server address	サーバ側のトンネル設定アドレス
interface name	自動設定に用いるトンネルインターフェース名
interface number	自動設定に用いるトンネルインターフェースナンバ
client address	自動設定に用いるIPv4アドレス

5.6.2 ネットワーク監視モジュールの実装

障害自動回避クライアントのネットワーク監視モジュールは，プログラム起動と同時に起動され，IPv6のネットワーク状態を監視する．目的ホストへの通信が途絶した場合，設定要求メッセージ送信命令をTACPクライアントに送る．通信が回復した場合，トンネル設定解除命令をトンネル設定モジュールに送る．ネットワーク監視モジュールを構成する，主要な関数を以下に挙げる．

observe6() ICMPv6 echo requestを送信し，ICMPv6 echo replyを受信する関数．クライアントのIPアドレスが入ったin_addr構造体を引数としてわたす．受信に成功すれば0，失敗すれば1を返す．

ck_status6() IPv4ネットワーク状態の変化を検査する関数．observe6()の結果を引数に渡す．前回のobserve6()の結果と比較して，値を返す．返り値はck_status4()と同じである．

5.6.3 TACPモジュールの実装

自動障害回避クライアントのTACPモジュールは，ck_status6()の返り値が2の場合に起動される．その他の部分は，動的トポロジ対応クライアントのTACPモジュールと同じである．

5.6.4 トンネル設定モジュールの実装

TACPモジュール，ネットワーク監視モジュールからの命令によって，実際にトンネルの設定，設定の解除を行なうモジュールである．動作はサーバのトンネル設定モジュールと同じである．

第6章

「評価」

TACSがネットワーク管理コストを軽減することを実証するため，実験ネットワークを構築し評価を行なった．本章では，評価について述べる．

6.1　現状の方式の評価

アドレス変化発生状況と，障害回避不能状況に現状のトンネル設定方式で対処した結果を表6.1にまとめる．

表6.1: 現状のトンネル設定方式での効果

設定方式	アドレス変化発生状況	障害回避不能状況
手動設定方式	△	△
自動設定方式	×	△

手動設定方式は2つの状況に対応できたが，全て手動で設定を変更しなければならないので管理コストが高くなった．自動設定方式はDNSにトンネル設定アドレスを登録できず，アドレス変化発生状況には対応できなかった．障害回避不能状況には対応できたが，管理ポリシを反映したトンネル設定ができないので不適切な経路情報を流してしまった．

6.2　評価方法

6.2.1　実験環境

TACSを評価するため，第4.3節で述べたモデルに即した実験ネットワークを構築した．ネットワークは，既存の6Bone JP* ネットワークの上にTACSを導入する形で構築した．

動的トポロジ対応システムの評価のために，6Bone JP内にTACSを導入したアドレス固定ホストを1台設置した．また，アドレス変化ホストとして，TACSを導入したIPv4ダイアルアップクライアントを2台，IPv4DHCPクライアントを1台用意した．自動障害回避システムの評価のために，6Bone JP内にTACSを導入した通信障害回避ホストと，自動障害監視ホストを1台づつ設置した．

- 動的トポロジ対応システム

---

*http://www.v6.sfc.wide.ad.jp/6bone

  - アドレス固定ホスト
    設置場所 :WIDE 6Bone 藤沢
    ホスト名 :elt.v6.sfc.wide.ad.jp
    IPv4 アドレス :203.178.140.44
    IPv6 アドレス :3ffe:501:0:100c:e320:260:67ff:fe2e:2289

  - アドレス変化ホストA
    IPv4アドレス :ダイアルアップPPPにより取得するため不定
    IPv6アドレス: 3ffe:501:0:100c:e321:200:86ff:fe10:334a

  - アドレス変化ホストB
    IPv4アドレス :ダイアルアップPPPにより取得するため不定
    IPv6アドレス: 3ffe:501:0:100c:e322:260:97ff:fe8b:f122

  - アドレス変化ホストC
    IPv4アドレス :DHCPにより取得するため不定
    IPv6アドレス :3ffe:501:0:100c:e322:260:8ff:fe25:55b0

- 自動障害回避システム
  - 通信障害回避ホスト
    設置場所 :WIDE 6Bone 藤沢
    ホスト名 :globe.v6.sfc.wide.ad.jp
    IPv4 アドレス :203.178.141.9
    IPv6 アドレス :3ffe:501:0:1000:2a0:24ff:fe48:7a3c

  - 通信障害監視ホスト
    設置場所 :WIDE 6Bone 藤沢
    ホスト名 :misia.v6.sfc.wide.ad.jp
    IPv4 アドレス :203.178.140.47
    IPv6 アドレス :3ffe:501:100c:e320:260:67ff:fe2e:2288

6.2.2 評価方法

動的トポロジ対応システムと自動障害回避システムについて，以下の方法で評価を行なった．

動作実験
設計に準じた動作によってトンネルが設定されるかを検証した．

現状の設定方式との管理コストの比較
トンネルの管理コストを，以下の3つに分類した．

  - 初期コスト

- トンネル設定
- トンネル設定解除

そして，同一のネットワークで手動設定方式と，TACSを用いた場合との管理コストを比較した．自動設定方式は，アドレス変化発生状況に対応できないので比較対象としなかった．

6.3 動的トポロジ対応システムの評価

6.3.1 動作実験

ダイアルアップPPPによってIPv4アドレスを取得する状況

ダイアルアップPPPによって取得される，毎回異なったIPv4アドレスによってトンネルを設定する場合の通信実験を行なった．実験環境を図6.1に示す．

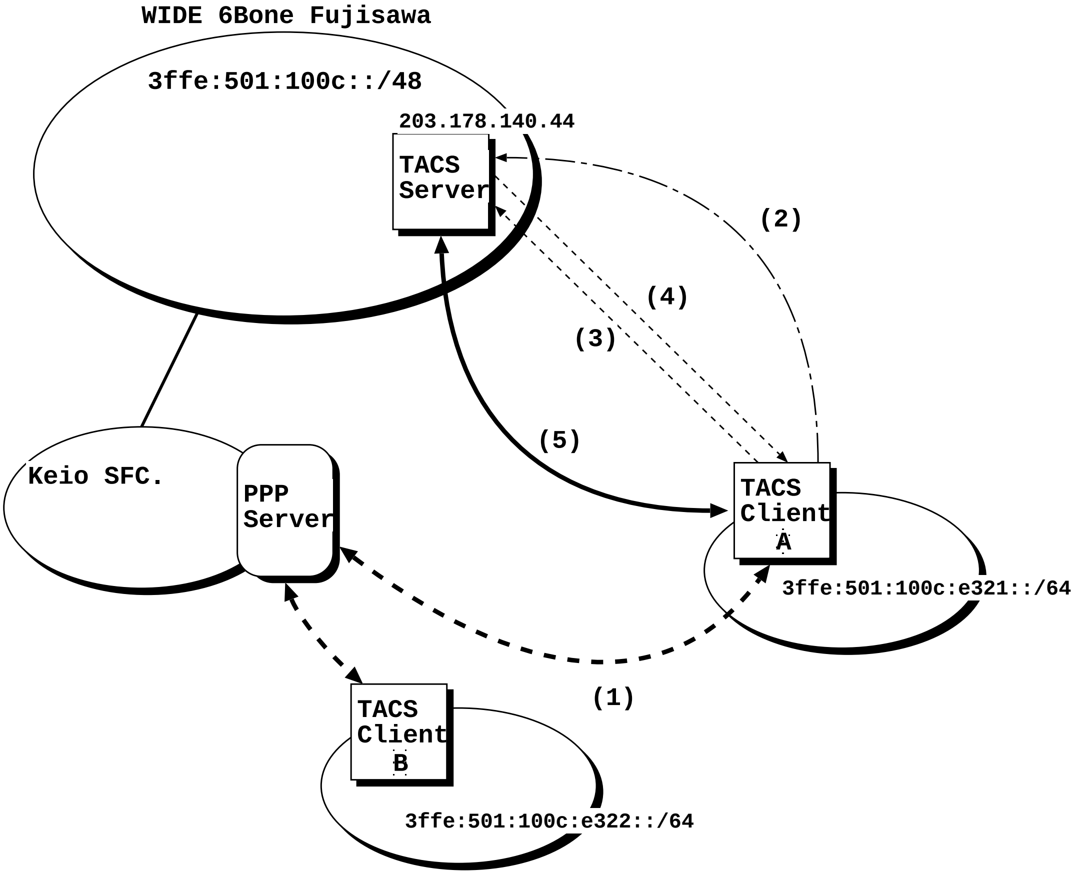

図6.1: ダイアルアップPPPを用いた実験環境

動的トポロジ対応クライアントを，IPv4ダイアルアップPPPによってインターネットに接続する環境に2台用意した．これらを慶應義塾大学湘南藤沢キャンパスのPPPサーバに接続して動作実験をした．

サーバの設定を図6.2に示す．

```
server address
203.178.140.44

interface name
gif

interface number
0
1
2
3

client address
```

図 6.2: ダイアルアップ PPP を用いた場合のサーバの設定

この設定では，TACS に用いるトンネルインターフェースはgif0～gif3である．また，全てのクライアントの要求を受け付ける．

クライアントの設定を図 6.3に示す．

```
server address
203.178.140.44

interface name
gif

interface number
0

client address
ep0
```

図 6.3: ダイアルアップ PPP を用いた場合のクライアントの設定

この設定では，サーバに203.178.140.44 を指定して，インターフェースはgif0 を用いる．トンネル設定アドレスはep0のアドレスを使用する．

この状況でTACS は以下のように動作した．

1. TACS クライアントがPPP サーバからIPv4 アドレスを取得し，PPP のリンクが形成される

2. TACS クライアントが，TACS サーバとの通信が確立されたことを検知する
3. TACS クライアントが，トンネル設定要求メッセージを送信する
4. TACS サーバが，トンネル設定応答(設定許可)メッセージを送信する
5. 相互にトンネルが設定され，通信ができる

クライアントがネットワーク的に移動する状況

アドレス変化ホストがネットワーク的に移動することによって，トンネル設定アドレスが変化する場合の通信実験を行なった．実験環境を図 6.4に示す．

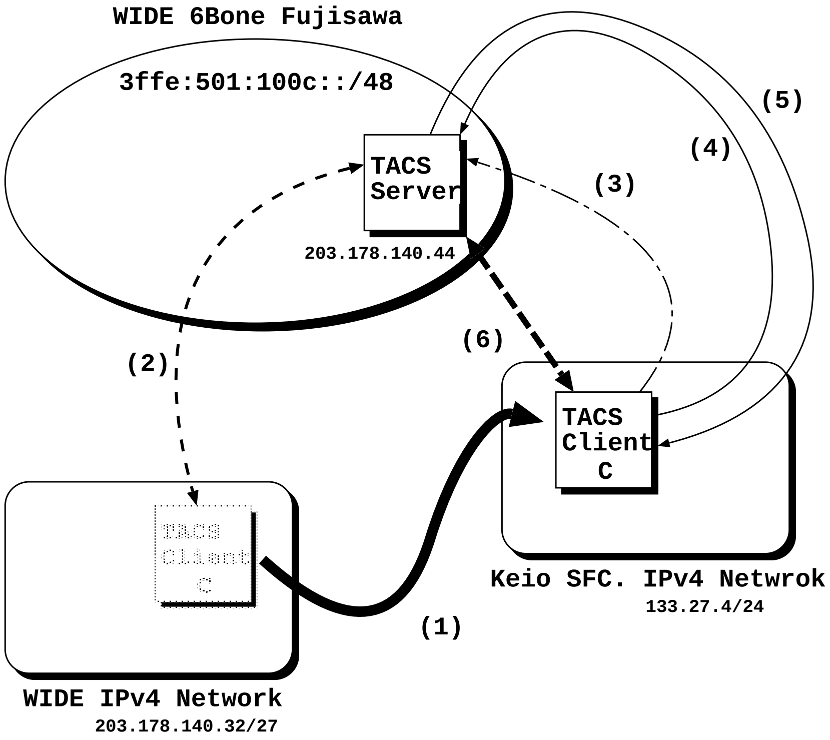

図 6.4: ネットワーク的に移動する状況の実験環境

1. TACS クライアントがWIDE IPv4 ネットワークから Keio SFC. IPv4 ネットワークに移動する．
2. 通信が不可能になったことを検知して，サーバとクライアントが相互にトンネルの設定を解除する．
3. DHCP で新たにIPv4 アドレスを取得したTACS クライアントが，再びサーバと通信可能になったことを検知する．
4. TACS クライアントが，TACS サーバに対して，トンネル設定要求メッセージを送信する．
5. TACS サーバが，トンネル設定応答メッセージを返信する．
6. 相互にトンネルを設定する．

表 6.2: 動的トポロジ対応システムの管理コスト比較実験結果

アドレス固定ホスト	手動設定	TACS
設定コスト	$5 * 2$	0
解除コスト	$5 * 2$	0
アドレス変化ホスト		
設定コスト	5	0
解除コスト	5	0

6.3.2 管理コストの比較

管理コストの比較をする為に，PPPで接続して，切断する動作を2台のアドレス変化ホストで5回づつ行なった．この状況下で，手動設定方式とTACSでのトンネル設定にかかる管理コストを比較した．比較結果を表6.2にまとめる．表6.2では，設定もしくは解除のための手動設定1回をコスト1としている．

アドレス固定ホストにおいてもアドレス変化ホストにおいても管理コストを削減できていることが実証された．特に，手動設定方式でアドレス固定ホストを設定した場合，接続ホストが増加する度に管理コストが増加する．TACSを用いた場合は，自動的にトンネルの設定を変更するので，設定コスト，管理コストの両方ともに0である．

以上の実験結果に基づき，アドレス変化発生状況における管理コストの評価結果を一般化し，まとめた．表中の式の変数と関数を表 6.3に示す．

表 6.3: 変数と関数の意味

h	1サーバに接続するクライアントの数を表す変数
n	クライアントを表す変数
$f(n)$	nがトンネル設定を要求する回数を求める関数
$g(n)$	nがトンネル設定解除を要求する回数を求める関数

動的トポロジ対応システムの管理コストの評価結果を表 6.4に示す．

手動設定で対応した場合初期コストはかからない．しかし，クライアントから要求がある度にトンネルの設定を修正しなければならない．クライアントが増加すれば，それだけ管理コストは高くなる．TACSを用いた場合，初期コストとして設定ファイルを記述しなければならない．しかし，それ以後はクライアントの数に関係なくコストがかからない．

クライアントも，サーバと同様手動設定で対応した場合初期コストはかからないが，トンネルを設定しようとする度にトンネルの設定を修正しなければならない．しかも，サーバ側のトンネルの設定がされなければ通信できない．TACSを用いた場合，設定ファイルを記述しなければな

表 6.4: 動的トポロジ対応システム管理コストの評価

トンネル設定方式	初期コスト	トンネル設定コスト	トンネル設定解除コスト
サーバ			
TACS	設定ファイルの記述	なし	なし
手動設定	なし	$\sum_{n=1}^{h} f(n)$	$\sum_{n=1}^{h} g(n)$
クライアント			
TACS	設定ファイルの記述	なし	なし
手動設定	なし	$f(n)$	$g(n)$

らないが，それ以上の管理コストはかからない．

6.4 障害自動回避システムの評価

6.4.1 動作実験

TACS の障害自動回避システムを用いた通信実験を行なった．実験環境を図 6.5に示す．

WIDE 6Bone 内のルータを TACS サーバとし，SFC 6Bone 内のルータを自動障害回避クライアントとした．そして，中継ルータを意図的に停止させ動作実験をした．また，管理ポリシを反映したトンネル設定について評価するため，トンネル設定を許可していないホストからトンネル設定を要求した．サーバの設定を図 6.6に示す．

この設定では，トンネル設定アドレスとして203.178.141.9 を使用し，トンネルインターフェースはgif10 を用いる．また，203.178.140.47 からの設定要求のみを受け付ける．

クライアントの設定を図6.7に示す．

この設定では，サーバに 203.178.141.9 を指定し，トンネルインターフェースはgif0 を用いる．また，トンネル設定アドレスは203.178.140.47 である．

以上の状況下で，TACS は以下のように動作した．

1. 自動障害回避クライアントがIPv6 ネットワークで通信障害が発生していることを検知する．
2. 自動障害回避クライアントがトンネル設定要求メッセージを送信する．
3. TACS サーバがトンネル設定応答メッセージを返信する．
4. 相互にトンネルを設定する．
    (a) TACS サーバが許可していない自動障害回避クライアントがトンネル設定要求メッセージを送信する

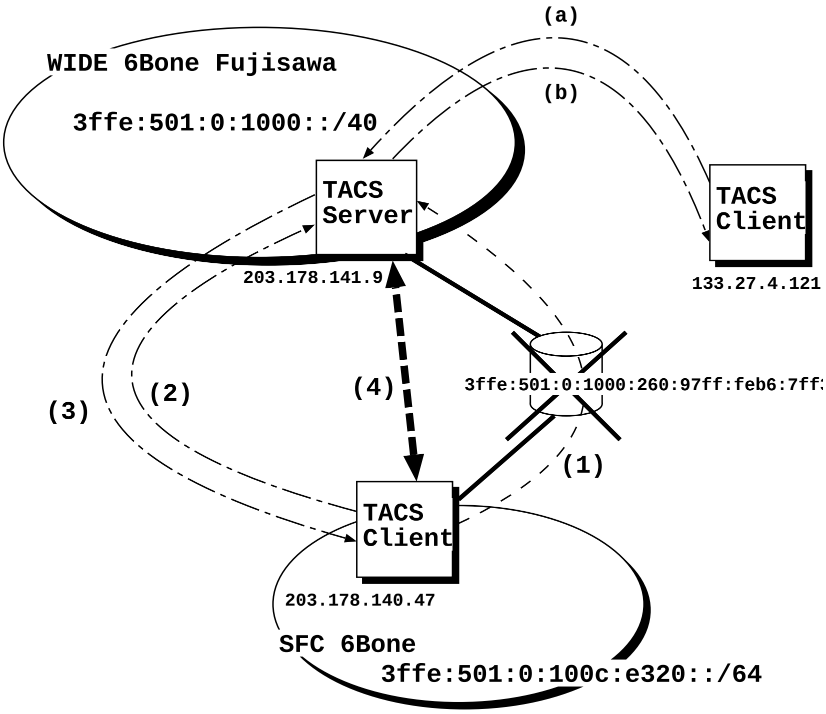

図 6.5: 自動障害回避システムの実験環境

(b) TACS サーバはトンネル設定拒否メッセージを送信する

自動障害回避システムによって，通信障害が自動的に回避できることが実証された．また，サーバが許可しないマシンからのトンネルの設定は拒否した．この結果から，管理ポリシを反映したトンネル設定ができることが実証された．

6.4.2 管理コストの比較

自動障害回避システムの管理コストの評価を行なうため，中継ルータを意図的に停止させ，その後復旧させることを3回繰り返した．この状況下で手動設定方式とTACSでのトンネル設定にかかる管理コストを比較した．比較結果を表6.5にまとめる．表6.5では，設定もしくは解除のための手動設定1回をコスト1としている．

通信障害復旧ホストと通信障害発生ホストの両方で管理コストが削減されていることが実証された．

アドレス変化発生状況における管理コストの評価結果を一般化し，まとめた．表中の式の変数と関数は表 6.3の通りである．サーバの評価結果を表 6.6に示す．

TACS自動障害回避システムも，初期設定を行なえばそれ以後に設定を変更する必要がない．また手動設定の場合，通信障害の発生を管理者が発見するまで，通信は回復しなかった．しかし，TACSは障害を検知し，自動的にトンネルを設定し，通信を回復させることができた．

```
server address
203.178.141.9

interface name
gif

interface number
10

client address
203.178.140.47
```

図 6.6: 自動障害回避システムの為のサーバの設定

```
server address
203.178.141.9

interface name
gif

interface number
0

client address
203.178.140.47
```

図 6.7: 自動障害回避クライアントの設定

また，サーバは設定を許可しないホストにはトンネルを設定しなかった．このことから，管理ポリシを反映したトンネル設定ができたと言える．

6.5 まとめ

手動設定方式と自動設定方式とTACSの比較を表 6.7に示す．

- ネットワーク管理者が直接操作する機会が減少し，管理コストは軽減された．
- トンネル設定認証機能により，手動設定と同様に管理ポリシを反映したトンネル設定が行なえた．

これらの結果から，TACSによってトンネルの管理コストが減少し，さらにネットワーク管理ポリシに則ったトンネルの設定が行なえることが実証された．

表 6.5: 自動障害回避システムの管理コスト比較実験結果

通信障害復旧ホスト	手動設定	TACS
設定コスト	3	0
解除コスト	3	0
通信障害発生ホスト		
設定コスト	3	0
解除コスト	3	0

表 6.6: 自動障害回避システムの評価

トンネル設定方式	初期コスト	トンネル設定コスト	トンネル設定解除コスト
サーバ			
TACS	設定ファイルの記述	なし	なし
手動設定	なし	$f(n)$	$g(n)$
クライアント			
TACS	設定ファイルの記述	なし	なし
手動設定	なし	$f(n)$	$g(n)$

表 6.7: 評価のまとめ

	手動設定方式	自動設定方式	TACS
管理コスト	×	○	○
管理ポリシの反映	○	×	○

第7章

結論」

本章では，本研究のまとめについて述べ，研究成果と今後の課題について述べる．

7.1 まとめ

本研究では，IPv4とIPv6のマルチプロトコル環境において，ネットワーク管理コストを軽減するため，動的仮想リンク制御機構であるTACSの実装と評価を行なった．

インターネットのIPv6への移行がさらに進行すると，インターネットはIPv4とIPv6が共存する環境となる．その中で，トンネルは独立したネットワーク同士を接続する非常に重要な技術である，トンネルはIPv6への移行が進行するにつれて様々な状況下で用いられると考えられる．しかし，現状のトンネルの設定方式では，トンネル設定アドレスが頻繁に変化する状況や，通信障害を自動で回避したい状況に対応できない．現状の設定方式ではネットワーク管理コストが高い，もしくは，ネットワーク管理ポリシを反映したトンネルを設定できないからである．

そこで，本研究では，現状のトンネルで対応できない状況に対応するため，動的仮想リンク制御機構であるTACSを提案した．TACSは，片端トンネル設定機能とトンネル自動設定機能を持ち，トンネルの設定を容易に変更できる．また，トンネル認証機能により，管理ポリシを反映したトンネル設定を行なえる．そして，TACSのプロトタイプを実装し，実験ネットワークを構築して管理コストが軽減できるかを評価した．

その結果，TACSはトンネル設定によるネットワーク管理コストを軽減させつつ，管理ポリシを反映したトンネル設定が行なえるシステムであることを実証した．また，手軽にIPv4ダイアルアップPPPでIPv6を利用できることから，IPv6の普及にも寄与できる．

7.2 本研究の成果

本研究の成果として，以下のものが挙げられる．

ネットワーク管理コストの軽減

動的仮想リンク制御機構を設計・実装したことによって，ネットワーク管理ポリシを反映しながら，ネットワークの管理コストを軽減できた．

IPv6インターネットへの容易な接続

自動設定システムにより，新たにIPv6用の回線を用意したり，手動設定方式のトンネルによって，他組織とトンネルで接続することなく，容易にIPv6インターネットに接続できるようになった．

IPv6への移行支援

容易にIPv6インターネットに接続できるようになれば，IPv6への移行は進行する．従って，

本システムはIPv6への移行を支援することができると言える．

安定したネットワークの構築

自動障害回避システムにより，通信が途絶してしまう時間が短縮され，ネットワークの安定度が増加した．

7.3 今後の課題

本研究における今後の課題には，以下のようなものが挙げられる．

TACSの実用ネットワークでの運用

TACSは，実験ネットワーク上ではネットワーク管理コストを軽減できた．今後，実用ネットワークで運用し，そこで浮上した問題点を改善する．

他OSへの移植

現在，TACSは，OSはFreeBSD，IPv6スタックはKAMEのみで動作する．今後，TACSを普及させるために，他OSに移植する必要がある．

IPv4 over IPv6トンネル自動制御

プロトタイプに実装されなかった，IPv4 over IPv6トンネルをTACSに実装する．今後，IPv6への移行が進み，IPv4 over IPv6トンネルが用いられる場合に対応し，長期間有用性を保つ．

認証システムの強化

プロトタイプの認証システムでは，動的トポロジ対応システムに対応できない．従って，パスフレーズ等による認証機能を導入して動的トポロジ対応システムの認証を実現する．

謝辞

本研究を進めるにあたり，暖かい御指導を頂いた，慶應義塾大学環境情報学部の村井純氏，徳田英幸氏，中村修氏，楠本博之氏に感謝します．

また，日頃の研究活動から様々な御指導を頂いた，株式会社東芝研究開発センターの南政樹氏に感謝の意を表します．

また，本研究に対して，細部にわたる指導と助言を頂いた慶應義塾大学政策・メディア研究科の関谷勇司氏，西田視磨氏，土井裕介氏に感謝します．

そして，様々な激励と助言を頂いた，WIDEプロジェクトIPv6 WGの皆様，慶應義塾大学村井・徳田・楠本・中村研究会のIPng WGの皆様と研究会の諸氏に感謝します．

最後に，バスケットボールと，Steve Kerr[*] を初めとする全ての3ポイントシューター達に深い敬意と感謝の意を表します．

---

[*] http://www.nba.com/playerflie/

参考文献

[1] R.Hinden S.Deering. Internet Protocol Version 6 (IPv6) Specification. RFC 2460. December 1998.

[2] Jon Postel. Internet Protocol. RFC 791. September 1981.

[3] A. Conta S. Deering. Generic Packet Tunneling in IPv6 Specification. RFC 2473. December 1998.

[4] Chistian Huitema. IPv6 The New Internet Protocol. Prentice Hall, second editon. 1997.

[5] R.Elz R.Bush. Clarifications to the DNS Specification. RFC 2181 July 1997.

[6] R.Gilligan E.Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. RFC1933. April 1996.

[7] R.Gilligan Transition Mechanisms for IPv6 Hosts and Routers. Work in progress. August 1996.

[8] W.Simpson. The Point-to-Pint Protocol(PPP). RFC 1661. July 1994.

[9] DHCP Options and BOOTP Vendor Extensions. S Alexander R.Droms. RFC 2132. March 1997.

[10] Jon Postel. User Datagram Protocol. RFC 768. August 1980.

[11] R.Hinden M.O'Dell S.Deering. An IPv6 Aggregatable Unicast Address Format RFC 2374. July 1998.

[12] Internet Standard Subnetting Procedure. J.C.Mogul J.Postel. RFC 950. August 1985.