階層的手法を用いた動的ネットワーク設定機構に関する研究[†]

冨永　明宏[‡]
慶應義塾大学大学院
政策・メディア研究科

寺岡　文男[§]
ソニーコンピュータ
サイエンス研究所

村井　純[¶]
慶應義塾大学
環境情報学部

本稿では、ネットワークの自動設定を実現する「動的ネットワーク設定機構」を提案する。大規模化し、複雑で動的に変化するインターネットにおいて、ネットワークの形態に対応する設定・管理の自動化は急務である。また、インターネットは個々の自律システムの集合として定義され、設定の方針は一定の管理構造に基づいて決定される。そこで本システムでは、この管理構造を階層的に定義し、それに基づいた自動設定を可能にするシステムを構築した。

本システムはサーバ・クライアントモデルを用いており、階層構造を自動決定するサーバ木構成部、ネットワークを設定するネットワーク設定部、個々の計算機を設定するホスト設定部の3つから構成される。プロトタイプ実装では簡易ネットワークエミュレータによる仮想LAN上で、サーバ木構成部とネットワーク設定部の基本動作を検証した。ホスト設定部は独立にプロトタイプ実装し、実際のネットワーク上でホスト単位のアドレス付け替えが行なわれることを確認した。これらの結果からネットワークの自動設定機構として、階層化モデルが有効であることを実証した。

1　はじめに

従来、計算機をネットワークへ接続するのに必要な設定は手動で行なわれていた。しかし携帯型計算機の普及と共に設定変更の頻度が急増し、ユーザへの負担となっていた。そのためDHCP[1]などの計算機自動設定機構が実現された。しかし依然としてネットワーク管理者の負担は大きいため、単体の計算機だけでなくネットワークを自動的に設定する機構が求められている。

例えば近い将来に実現されるUbiquitous Computingの環境では、多数の計算機から構成されるネットワークを自動的に設定して、管理者の負担を軽減することが望ましい。また、IPアドレスの枯渇や経路情報の急増によるネットワーク運用障害などの問題を解決するためには、ネットワークアドレスを付け替える必要がある[2]。アドレス付け替えの労力を軽減するためにもネットワークの自動設定機構が必要とされている。

現在まで、このようなネットワーク自動設定機構は実現されていない。またアドレス付け替えについても研究が始まったばかりである。したがって、これらの問題を解決するためのネットワーク設定機構を実現し、ネットワーク管理の負担を軽減することが本研究の目的である。本稿では動的ネットワーク設定機構の設計を行ない、プロトタイプ実装を用いて基本動作を検証した。

2　動的ネットワーク設定機構の概要

本稿ではネットワークの自動設定に必要な機構を初めから全て実現するのではなく、ネットワークアドレスの設定のみに着目する。経路制御の複雑な設定や、他のネットワークサービスの設定については扱わない。

本節では初めにネットワーク管理の手法をモデル化し、次に動的ネットワーク設定機構の設計目標について述べる。これらの目標を達成するには階層構造が適していることを明らかにし、最後に本システムの基本構成を示す。

2.1　ネットワーク管理のモデル

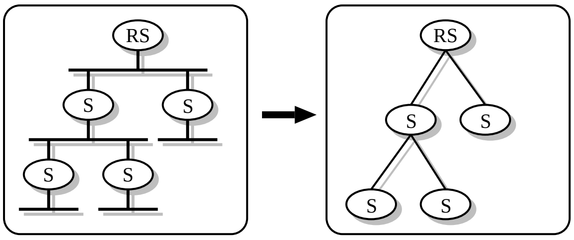
本研究におけるネットワーク管理モデルでは、管理の最小単位は「1つのルータ, そのルータに接続された複数の物理セグメント, それらのセグメント上のホストの集合」から構成される。またネットワーク設置には次のような手順をとる:

[†] Research on Dynamic Network Configuration Mechanism with Hierarchical Scheme

[‡] tomy@sfc.wide.ad.jp

[§] tera@csl.sony.co.jp

[¶] jun@wide.ad.jp



(a) (b)

図 1: ネットワーク階層化の例

1. 既存のネットワークにルータを接続し、既存ネットワーク側のネットワークインタフェースを設定
2. それ以外のネットワークインタフェースを設定
3. その他のホストのネットワークインタフェースを設定
4. ルータで経路制御を開始
5. その他のホストでも経路制御を開始

既存のネットワークとは組織あるいはインターネットプロバイダのバックボーンなどを意味し、そこを始点にネットワークは木のように徐々に成長するものとする。すなわちネットワークの設置とは、あるネットワークを根とする木構造を拡張することであるとモデル化する。

2.2 階層構造の導入

動的ネットワーク設定機構を実現するにあたり「現実のネットワークや Ubiquitous Computing で用いるために負荷分散を実現し、大規模性を確保する」「管理者の負担を減らすために管理者が直接操作する計算機は少数にする」「種々の問題を解決するために必要なアドレス付け替えを支援する」「多様なネットワーク管理ポリシに柔軟に対応するためにポリシと独立な機構にする」「DHCP などの既存プロトコルを利用できるようにモジュール性を高める」などを目標とする。

ネットワーク管理のモデルに対応して、全てのルータ上では本システムが動作する。ルータ群は自律的にネットワークトポロジに基づいた階層構造 (Spanning Tree) を構築する (図 1)。ただし前述の管理モデルではバックボーンなどのネットワークを始点としたが、実際にはネットワーク管理者が指定したルータを始点とする。

階層構造を採用する理由は、第一に大規模性が確保しやすいためである。すなわち、あるルータで要求を処理できない場合にだけ上位へ位置するルータに要求を転送することで、負荷分散を実現できる。第二にネットワーク管理者が操作する計算機を階層構造のトップに位置するルータだけに限定できるためである。

2.3 システム・アーキテクチャ

本システムではサーバ-クライアントモデルを用いる。サーバは全てのルータ上で動作し、クライアントはルータ以外の全ての計算機上で動作する。階層構造の最上位 (木構造の根) に位置するサーバを特にルートサーバと呼んで区別する。したがってシステムはルートサーバ、それ以外のサーバ、クライアントの 3 つのエージェントから構成される。ルートサーバは唯一管理者によって設定管理されるサーバであり、それ以外のサーバは階層構造内の上位、下位のサーバと協調して自律的に動作する。

次にプロトコルの機能別に本システムを分類する:

1. 階層構造を動的に決定するサーバ木構成部
2. ネットワーク単位でのアドレス管理を行なうネットワーク設定部
3. クライアントを自動設定するホスト設定部

システムが起動すると、初めにサーバ木構成部を用いてネットワーク・トポロジに基づく階層構造を自動的に決定する。ネットワーク設定部は階層構造を基にアドレス管理を行ない、各サーバにアドレスの集合を割り当てる。各サーバはホスト設定部を用いて、接続されているセグメント上に存在する各ホストの自動設定を行なう。

3 動的ネットワーク設定機構の設計

初めに本システムが管理の対象とするネットワーク環境の条件を示し、次に本システムを構成する 3 つの機能のそれぞれについて設計を述べる。

3.1 ネットワークと計算機に対する前提

ここでは本システムが対象とするネットワークの条件を示すが、これらの大半は現在の標準的な環境では既に満たされており、十分に現実的なものである。

第一にルータの自動設定という難しい問題を切り離すため、本システムの管理対象となるネットワークでは、(default route, RIP, 単一 AS 内の OSPF などの) 単純な経路制御を用いていなければならない。また効率的にアドレスを利用するためには、サブネットの大きさを自由に設定できる必要がある。したがって全てのルータは Variable Length Subnet Mask をサポートしていなければならない。さらに対象となるネットワークは単一の管理ポリシで運用されていなければならない。もし仮に複数のポリシが混在すると、自律分散システムとして設計された本システムでは矛盾を生じて破綻する可能性が高い。またアドレス付け替えを実現するために、サーバは連続して設置されていなければならない。

アドレス付け替え実現のため、クライアントは一つのネットワークインタフェースに複数アドレスを設定することができるものとする。1 つしかアドレスが設定できないと、アドレスの付け替えと共に通信が切断されてしまう。これを回避するには複数アドレスを用いて緩やかにアドレスを付け替えればよい。緩やかなアドレス付け替えの方法については後述する。

さらに、全計算機でクライアントが動作しなければならないと前述したが、これは非現実的である。したがってネットワークの二重運用モデルというものを用いる。ネットワークの二重運用モデルでは 1 つの物理ネットワークを、従来通りに運用する論理ネットワークと、本システムを用いて運用する論理ネットワークの 2 つに分割する。クライアント機能を持たない計算機は従来通りの論理ネットワークに接続しておけばよい。このネットワークの二重運用を行なうためには、全てのルータで 1 つのネットワークインタフェースに複数ネットワークアドレスを設定できなければならない。

3.2 サーバ木構成部の設計

サーバの階層構造を静的に設定すると耐故障性が低くなる。また誤った静的設定は、階層構造をもとに動作する本システムの誤動作の原因となる。したがって自動的に階層構造を決定するサーバ木構成部が必要となる。ただしルートサーバだけは管理者が静的に指定する。

設計の際には「(IP アドレスが変化するため)IP アドレスや IP ルーティングに依存しない」「起動から階層構造が決定されるまでの時間を短くする」「不必要なアドレス付け替えを防ぐために階層構造の変動は抑える」「少ない計算量で動作する」「任意のサーバの起動が遅延しても階層構造は変化しない」「実装しやすい簡潔な機構にする」などの目標を設けた。

本システムでは経路制御に似た独自の情報交換に基づいて階層構造を決定する。主な経路制御方式には距離ベクトル型とリンク状態型の 2 つがある。一般にリンク状態型の方が収束は早いが、実装は距離ベクトル型の方が容易である。またリンク状態型はネットワークの接続性がわずかに変化しただけで大量の演算を必要とする。「少ない計算量」と「簡潔な機構」という目標を達成するために本システムでは距離ベクトル型を採用する。ただし Triggered Update に類似した手法を取り入れ、階層が深くなっても実用に耐える時間内に収束するようにする。Triggered Update とは、経路が変化した場合に直ちに近接ルータへ更新情報を送る手法である。本システムではルートサーバに関する情報だけを伝達するので、トラフィック量は無視できる。

階層構造の変動を少なくするために、サーバは一定時間が経過した情報だけを (安定したと仮定して) 採用する。また、一つのセグメントに複数のサーバが存在する時には、メッセージを交換してサーバを一つ選択する。この時、サーバの起動が多少遅延しても一定のサーバが選択されるようにした。

情報交換から求められる上位サーバと現在の上位サーバが異なる状態が長期間継続した場合 (上位サーバが故障したり、よりホップ数の小さな上位サーバが発見されたと考えられる) には、上位サーバを変更する必要がある。しかし本システムでは上位サーバ変更はアドレス付け替えなどを引き起こすため、頻繁に行なうべきではない。上位サーバを変更する条件やその時期については今後の研究で検討する必要がある。

3.3 ネットワーク設定部の設計

ネットワーク設定部では決定された階層構造に基づき、ネットワークアドレスの授受を行なってネットワークを設定する。本稿では以後、連続したアドレスの集合をアドレスバルクと呼び、アドレスバルクの集合をアドレスプールと呼ぶ。ネットワーク設定部は本システムの中核であるため設計目標は概ね全体の目標と一致する。

管理ポリシとプロトコルを分離するために、ネットワーク設定部ではアドレスプールの状態と状態遷移、アドレスプールへの操作、使用するメッセージなどについてのみ定義して設計とする。その際にルートサーバに対する要求が適切な下位サーバまで伝達するように状態遷移を定義して、集中操作を実現する。

ネットワーク設定部は「上位サーバから下位サーバへのアドレスバルク割当」と「下位サーバから上位サーバへのアドレスバルク返却」の 2 つの操作を基本として動作する。アドレスの割当 (返却) を要求する側がメッセージの再送を行なう。すべてのネットワーク管理ポリシは、これらの操作を用いて実現される。

本システムは階層構造をとるため、割当 (返却) 要求を上位 (下位) のサーバに要求を転送しなければならない場合がある。階層が深くなると各サーバでの処理が重なって時間がかかるため、処理中を意味する応答メッセージ (Pending メッセージ) を設ける。

各サーバの基本動作の例を以下に示す。

1. アドレスが不足すると上位サーバに割当 (下位サーバに返却) を要求する。
2. 割当 (返却) 要求を受け取った場合、割当 (返却) 可能なら処理し、不可能なら拒否する。
3. 割当 (返却) 要求を処理するのに時間がかかる–例えば上位 (下位) サーバに割当 (返却) を要求しなければならない–場合には、Pending メッセージを返す。

表 1: ネットワーク設定部で用いるメッセージ

名称	用途
AllocRequest	下位サーバから上位サーバへの割当要求
ReleaseRequest	上位サーバから下位サーバへの返却要求
Alloc	上位サーバから下位サーバへの割当
Release	下位サーバから上位サーバへの返却
AllocNack	下位サーバからの割当要求に対する拒否
ReleaseNack	上位サーバからの返却要求に対する拒否
AllocPending	AllocRequest を上位サーバが処理中であるという通知
ReleasePending	ReleaseRequest を下位サーバが処理中であるという通知

アドレス不足であると判定する基準や、割当と返却のどちらを要求するかを選択する基準、それらの要求を拒否する基準などはポリシに深く関わるため、ここでは定義しない。使用するメッセージの一覧を表 1に、アドレスバルクの状態を表 2に、アドレスバルクの状態遷移図を図 2に示す。返却中という状態はアドレス付け替えを支援するために設けた。また不必要なメッセージの再送を防ぐために、割当などを拒否されたアドレスバルクを記憶する状態を設けた。また返却要求メッセージには強制回収フラグが含まれ、下位サーバが返却を拒否できないように指定できる。

3.4 ホスト設定部の設計

ホスト設定部は個々のクライアントを設定する。設計に際して「モジュール化し、他の部分から独立に実現する」「アドレス付け替えのために複数アドレスの設定をサポートする」「アドレス付け替えのためにアドレスの強制回収を可能にする」「ホストは自身がネットワークを移動したことを検知できる」「プロトコルを簡潔にする」などを目標とした。

ホスト設定部は IPv6 の Autoconf[3] のように、サーバが定期的にネットワークに関する情報を Advertise する。新しい内容を持つ Advertise メッセージを受信したクライアントはアドレスの割当を要求する (図 3)。

内容の異なる Advertise メッセージを複数送信すれば、その各々に対してクライアントは Request するので複数アドレスがサポートできる。また Advertise メッセージにフラグをセットすることで、アドレスを強制回収できるようにした。さらに Advertise メッセージは定期送信されるため、一定時間以上メッセージが受信できなかったり、今までと異なる Advertise メッセージを受信した場合にはネットワークを移動したと推測できる。

最後の「プロトコルを簡潔にする」という目標を達成するために、アドレスには有効期限などの違いを設けない。

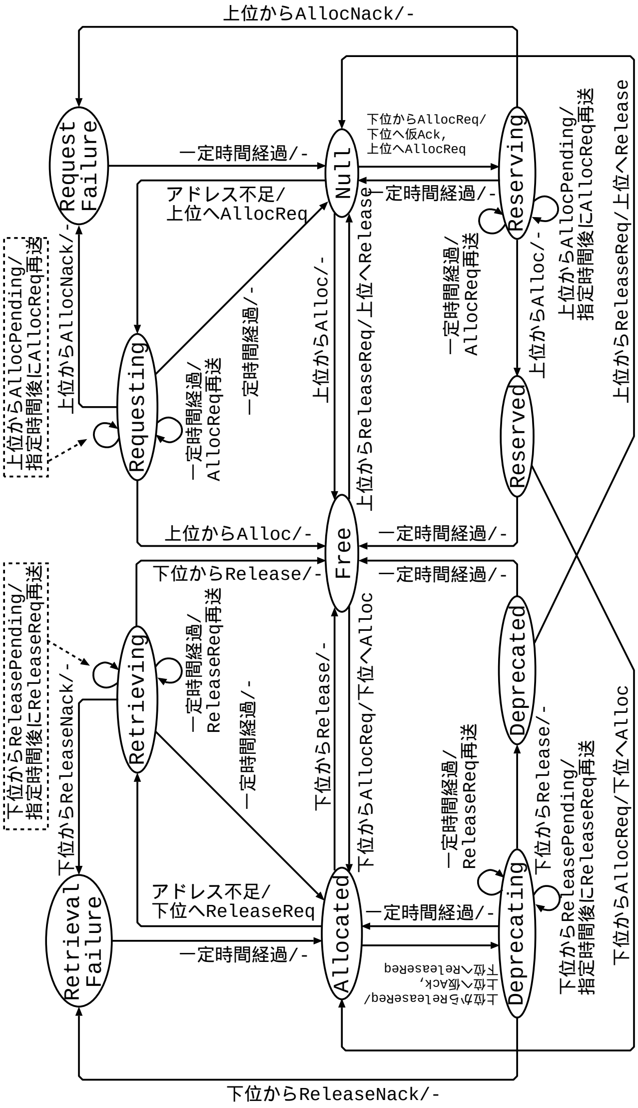


図 2: アドレスプールの状態遷移

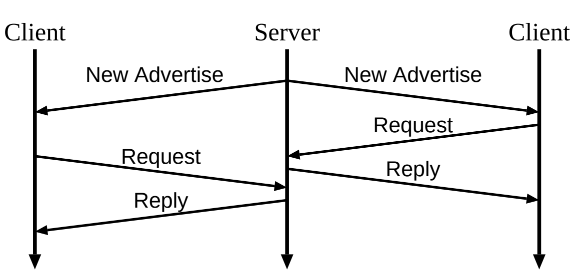


図 3: アドレス割当 (ホスト設定部)

表 2: アドレスバルクの状態

状態	説明
Free	サーバやクライアントに割り当てられていない領域
Allocated	下位サーバあるいはクライアントに割り当てられている領域
Requesting	上位サーバに割当要求中の領域。割当により Free 状態に遷移する
Request Failure	Nack で割当を拒否されたために一定時間、割当要求できない領域
Retrieving	下位サーバに返却要求中の領域。返却により Free 状態に遷移する
Retrieval Failure	Nack で返却を拒否されたために一定時間、返却要求できない領域
Reserving	下位サーバの割当要求により上位サーバへ割当要求中の領域
Reserved	Reserving 状態の領域に対する割当を受け、下位サーバからの割当要求を待っている領域。一定時間後に Free 状態に遷移する
Depricating	上位サーバの返却要求により下位サーバへ返却要求中の領域
Deprecated	Depricating 状態の領域に対する返却を受け、上位サーバからの返却要求を待っている領域。一定時間後に Free 状態に遷移する

また本システムでは論理セグメントとサーバは 1 対 1 に対応する。これによりホスト設定部は、DHCP のように 4 回メッセージを交換するのではなく、Request, Reply の 2 回で処理できる。またクライアントへ通知する情報は IP アドレス, ネットマスク, デフォルトルータの 3 つに限定し、それ以外の情報については Service Location Protocol[4] などの外部機構を利用するものとする。

3.5 アドレス付け替えの概要

アドレス付け替えのために、ホスト設定部は「サーバが任意の時点でクライアントに対してアドレスを返却させる」機能を持っている必要がある。以下に本システムでのアドレス付け替えの手順を、通常のホスト設定処理の流れと共に述べる。

1. サーバはクライアントに新しいアドレスを取得させるために、新しい Advertise を開始する。付け替えの対象となる古いネットワークアドレスについても Advertise を継続するが、これには Deprecating フラグとアドレスの最終有効期限を含める。
2. クライアントは新しい Advertise を受信し、Request を送ってアドレスの割当を要求する。
3. 古いアドレスに関する Advertise を受信したクライアントは、有効期限が過ぎたら古いアドレスの利用を中止する。あるいは自分の属するネットワークアドレスの Advertise を一定時間以上受信できない場合にも利用を中止する。

さらにアドレス付け替えのために、ネットワーク設定部では新しいアドレスの割当、古いアドレスの回収、経路制御の変更の 3 つを実施する必要がある。ネットワーク設定部のアドレス付け替え手順の概要を以下に示す。

1. ReleaseRequest を下位サーバに送る。
2. 受け取った下位サーバは、その領域が Free であれば直ちに上位サーバに返却する。
3. その領域を下位サーバに割り当ててある場合には、下位サーバに ReleaseRequest を送る。同時に上位サーバには ReleasePending メッセージを送る。
4. その領域を自分で使用している (クライアントに割り当てている) 場合にも、上位サーバに ReleasePending メッセージを送る。同時に上位サーバへ新たなアドレスの割当を要求し、ホスト設定部を通じてクライアントのアドレス付け替えを行なう。
5. 期限までに上位サーバへアドレスを返却する。
6. サーバはクライアントへアドレス割当を開始する時に経路情報をアナウンスし始める。また上位サーバへアドレスを返却した時点で、アナウンスを止める。

4 実装と評価

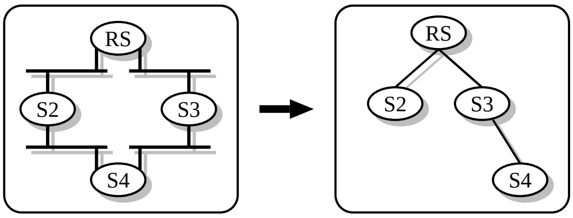
BSD/OS 2.1 上に実装した各部のプロトタイプについて実装と評価を述べる。

4.1 サーバ木構成部の実装と評価

サーバは他のサーバから送られてくるメッセージの受信待ちを行ないつつ、多数のメッセージの送信や再送をする必要がある。このような処理の多重化を実現するために、サーバ自身でスケジューリングすることにした。また本システムはアプリケーション・プロセスとして実装し、通信に UDP を利用するなどして移植性を重視した。

本システムは、ネットワークアドレスが設定されていない状態でも通信を行なわなければならない。またメッセージを受信したネットワークインタフェースを識別したり、メッセージを送信するネットワークインタフェースを指定できなければならない。これらの要求を満たすためには、Berkeley Packet Filter もしくは同等の機能を利用すれば良い。ただしプロトタイプ実装では、サーバ木構成部およびネットワーク設定部には簡易ネットワークエミュレータを用いた。これにより任意のネットワーク・トポロジを用いて実験することが容易になった。

実験の結果、図 4(a) のようなループを持つ仮想ネットワークから、図 4(b) のようなサーバ木が構成されること



(a) (b)

図 4: 仮想ネットワークに構築された階層構造

を確認した。実験では経路表の収束までに要する時間は 2 秒程度であり、収束の速さについても目標を達成できた。その他の目標についても達成できたが、計算量についての定量的な評価については検証できていない。今後は、さらに複雑なトポロジや実際のネットワークでも動作を確認する必要がある。

4.2 ネットワーク設定部の実装と評価

ネットワーク設定部の設計では「ポリシとプロトコルの分離」を目標としたが、プロトタイプでは以下のような仮アルゴリズムを実装した。

- あるセグメントに割り当てたアドレスの総数が、アドレスバルク全体の 50%を超えた時点で、アドレスが不足したとみなす。
- アドレスが不足した場合には、下位サーバへ返却要求するのではなく上位サーバへ割当要求する。下位サーバへ返却要求するのは、ネットワーク管理者がルートサーバに指示した場合だけとする。
- アドレスが過剰であっても、上位サーバから要求されない限りは返却しない。

前述の仮想ネットワークを用いて、ネットワーク設定部の基本動作も検証した。その結果、各サーバに対して必要なアドレスバルクが割り当てられることが確認できた。今後はアドレスの強制回収やアドレス付け替えなどについて動作確認する必要がある。また現実の多様な管理ポリシを実現できることを検証し、あるポリシの実現に必要な機能が不足していないか評価しなければならない。

4.3 ホスト設定部の実装と評価

プロトタイプでは、ホスト設定部をネットワーク設定部と統合せずに単独のプロセスとして実装した。理由として「負荷を分離し、ネットワーク設定部の動作を安定させる」「メンテナンスを容易にする」「他のホスト設定機構を利用できるようにモジュール性を高める」などがある。ホスト設定部も BSD/OS 2.1 上に BPF を用いて実装した。また実装を簡素化するために物理メディアは Ethernet に限定した。

ホスト設定部は現実のネットワークを用いて動作を確認した。その結果、複数アドレスの設定と (計算機単位の) アドレスの付け替えが機能することを確認した。またクライアント自身が別のネットワークへ移動したことを検知できることも確認した。

5 結論

本稿では、ネットワークの自律構成およびアドレス付け替えの支援機構を実現するための手法として階層化モデルを提案した。この階層化モデルに基づいて実装した動的ネットワーク設定機構は、計算機単体の設定のみならずネットワーク単位での設定を可能にする。このようなネットワークの自律構成、ネットワーク単位での自動設定およびアドレス付け替えを実現する機構は前例がない。

本システムはサーバ木構成部、ネットワーク設定部、ホスト設定部の 3 つの機能に分割できる。サーバ木構成部には Triggered Update の手法を用いた距離ベクトル型経路制御アルゴリズムを適用し、階層構造の素早い構築を実現した。ネットワーク設定部は多様なネットワーク管理方針に対応できるようにポリシ独立に設計した。またエミュレータにより構成される仮想実験ネットワーク上で実験を行なった結果、ネットワーク設定部の基本動作を確認できた。ホスト設定部ではサーバがネットワーク情報を定期的に配布する手法を提案し、この手法がアドレスの付け替えや強制回収およびネットワークの移動を検知するのに有効であることを実証した。

参考文献

[1] Tominaga, A., Osamu, N., Teraoka, F., and Murai, J.: *Problems and Solutions of DHCP*, Proceedings of INET'95, Vol. 1, pp. 481–490, June 1995.

[2] Fuller, V., Li, T., Yu, J., and Varadhan, K.: *Classless Inter-Domain Routing(CIDR): an Address Assignment and Aggregation Strategy*, RFC 1519, September 1993.

[3] Thomson, S., and Narten, T.: *IPv6 Stateless Address Autoconfiguration*, draft-ietf-addrconf-ipv6-auto-07.txt, December 1995.

[4] Veizades, J., Kaplan, S., and Guttman, E.: *Service Location Protocol*, draft-ietf-srvloc-protocol-07.txt, December 1995.