

ホスト情報を考慮したDHCPv6アドレス割当て機能の実装

門田美由紀[†] 小原 泰弘^{††} 南 政樹^{†††} 村井 純^{†††}

[†] 慶應義塾大学総合政策学部 〒252-8520 神奈川県藤沢市遠藤 5322

^{††} 慶應義塾大学院政策・メディア研究科 〒252-8520 神奈川県藤沢市遠藤 5322

^{†††} 慶應義塾大学環境情報学部 〒252-8520 神奈川県藤沢市遠藤 5322

E-mail: {miyu,yasu,minami,jun}@sfc.wide.ad.jp

あらまし IP アドレスやサービス資源を管理者がそのポリシーに応じて割当てたい、またその割当てた資源を管理者が把握したいという stateful configuration への要求が高まっている。現在、IPv6 環境でのアドレス割当て機構として NDP の RA メッセージが一般的に使用されているが、stateless configuration という特徴ゆえに問題点が挙げられる。本研究では、IPv6 stateful configuration の利点を挙げ、DHCPv6 を用いることで実現が可能であることを示す。また、ホスト情報を定義し、実際に DHCPv6 を用いてホスト情報に応じた IP アドレス、サービス資源の割当て、管理機構を設計、実装する。

キーワード DHCPv6, stateful configuration, IPv6 リンクローカルアドレス

An Implementation of DHCPv6 Address Allocation with Consideration for Host Identity

Miyuki KADOTA[†], Yasuhiro OHARA^{††}, Masaki MINAMI^{†††}, and Jun MURAI^{†††}

[†] Keio University, Faculty of Policy Management 5322 Endo, Fujisawa-city, Kanagawa, 252-8520, Japan

^{††} Keio University, Graduate School of Media and Governance 5322 Endo, Fujisawa-city, Kanagawa, 252-8520, Japan

^{†††} Keio University, Faculty of Environmental Information 5322 Endo, Fujisawa-city, Kanagawa, 252-8520, Japan

E-mail: {miyu,yasu,minami,jun}@sfc.wide.ad.jp

Abstract Router Advertisement message defined in the Neighbor Discovery Protocol is the most commonly used method for IP address allocation in IPv6 networks. However, its use of the stateless configuration brings problems for administrators when allocating IP address and services resources based on a policy or when maintaining the status of the allocated resources.

In this study, we first discuss the merits of IPv6 stateful configuration, and show the feasibility of using DHCPv6. We then define the needed host information, and design and implement a system for allocating IP address and service resources based on DHCPv6.

Key words DHCPv6, stateful address configuration, IPv6 link-local address

1. 背景

1.1 IPv6 と Plug & Play

IPv4 アドレスの枯渇により、次世代インターネット基盤プロトコルである IPv6 [1] が普及してきている。IP アドレスを 128bit へ拡張することにより広大なアドレス空間が提供され、コンピュータだけではなく、家電、自動車、マイクロノードなど様々な機器が IP によってインターネットへつながるように

なる。そのため、IPv6 では物理的にネットワークに接続しただけで IP アドレスを始め主要なネットワークサービスが使用できるようになる Plug & Play 機能が重視された。これにより、高度な技能を持たない人でも「つないだだけでインターネット」が可能となる。

1.2 IPv6 のアドレス自動設定機能

現在、Plug & Play の実装方法として、Neighbor Discovery Protocol (NDP) の Router Advertisement Message (RA) [2] を

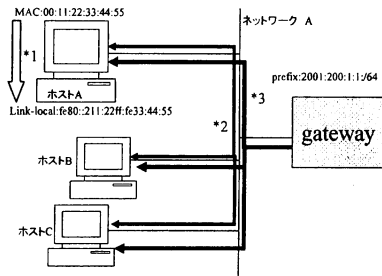


図1 RAメッセージの動作

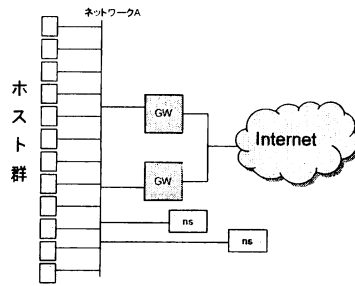


図2 stateful configuration が必要な例

利用して IP アドレスを自動設定している。RA メッセージによる設定は、サーバが IP アドレスを配布したホストの情報をもたない stateless(状態不保持) モデルである。

NDP の RA メッセージによるホストへの IP アドレス割当方法を図 1 を用いて説明する。

図 1 の場合、一つのネットワーク A の中にホスト A,B,C が存在するものとする。ネットワーク A に新たに接続したホスト A は、まず、fe80:0000:0000:0000 の 64bit を前半とし、自身の 48bit の Ethernet アドレスを EUI-64 フォーマットに変換し [3]、それを後半に組み合わせて 128bit のリンクローカルアドレスを作成する(*1)。次に、ホスト A のリンクローカルアドレスがネットワーク A の中で重複していないかを、NDP の neighbor Solicitation, Neighbor Advertisement メッセージをホスト B、C と交換することによって確認し、重複していない場合はホスト A が生成したリンクローカルアドレスが確定する(*2)。ネットワーク A の gateway は、ネットワーク A のプレフィックス情報を含んだ RA メッセージ (RA) を一定間隔で広告する(*3)。ルータの RA を受け取ったホスト A は、RA のプレフィックス情報と EUI-64 フォーマットに変換した自身の Ethernet アドレスを組み合わせて IPv6 アドレスを作成する。

以上が、NDP の RA メッセージを用いた IPv6 グローバルアドレスの自動設定方法である。

1.3 NDP の RA メッセージを利用する場合の問題点

NDP の RA メッセージを利用して IP アドレスを割当てる方法は、ルータが各ホストの割当状況を知る必要がなく、Plug & Play を実現させている点で非常に有用である。

しかしこの方法は、いくつかの問題点があげられる。

- 全ての RA を全てのホストが受信、信用してしまう
- セグメント分割でしかプレフィックスを変更できない
- パケットが固定長である
- IP アドレスとそれを使用するホスト情報を RA を配るルータが保持できない

このため、全てのホストに同じプレフィックスの IP アドレスを配布し接続性を確保するというトポロジの場合、この方法は非常に良いが、複雑なトポロジで運用をする場合は非常に不便となる。また、ネットワーク内に接続性のないプレフィックスを持つ RA を流すホストがいた場合、それを防ぐ手段がな

いことも問題となる。また、RA のパケットは固定長でありオプションとして他のパラメータを入れることができない。その為、Domain Name System(DNS) [4] [5] サーバなどサービス資源の IP アドレスを配布することができない。これについては後述する。

2. 管理者の要求

1 章に挙げた問題点により、特に管理者にとって NDP の RA メッセージを使用する場合、トポロジに制限が多くなる。特に、ネットワークを使用しているホストとそのサービス資源の情報を管理者が把握し、それにに応じて動的に振り分けをしたいという要求がある。

具体的な例を挙げる。

図 2 では、1 つの分割されていないネットワーク A 上にホストが多数いるものとする。ゲートウェイ (gw) は 2 台、また DNS サービスを行うネームサーバ (ns) も 2 台存在する。このときの管理者の要求を挙げる。

- 2 台のゲートウェイを使用する

IP アドレスまたはプレフィックスで通るゲートウェイを決めることで、IP アドレスを識別子としてゲートウェイでトラフィックを分けることができる

- 2 台のネームサーバのどちらかをホストに通知する
- どのホストがネットワークを使用しているか管理者が把握する

管理者がネットワークを使用するホストと、そのホストが使用しているサービス資源を把握する機構が存在することで、現在のネットワーク使用状況が確認できる

- トラブルがあった場合、ホスト個々の IP アドレス、サービス資源を管理者が動的に設定しなおす

この状況の場合、IPv6 の状況下でも管理者がホスト単位で細かい設定ができ、管理者が現在ネットワークを使用しているホスト情報を管理できるというモデルが必要であるといえる。

3. 目的

本研究では、管理者側がホストごとに IP アドレス、サービス資源の配布、割り振り、把握ができるシステムを実装することを目的とする。

具体的には以下の2点の解決を目的として、設計・実装を行う。

- IP アドレス・サービス資源をホスト情報に応じて割当てる機構

- ホスト情報とそのホストの使用サービス資源を把握できるデータベース

これにより、管理者がホストごとに資源割当ての設定することが可能となる。また、IP アドレスの使用状況、サービス資源の使用状況等、管理者が把握したい情報を入手でき、その結果を、ネットワークの運用に生かすこともできる。このため、ネットワークの負荷分散、状況監視等、管理者側にとっての運用上の利点が大い。

4. DHCPv6

stateful(状態保持)のモデルを使用した例として Dynamic Host Configuration Protocol for IPv6(DHCPv6) [6] があげられる。

stateful configuration のモデルを利用し、DHCPv6 を使用した場合の利点を述べる。

- 4 回のメッセージ交換によりサーバを一つに選択する
- サーバの設定により、ホスト情報に応じてアドレス、サービス資源の割当てを変えることができる
- メッセージが可変長のため、オプションを追加することが可能である
- リースしたアドレスのデータベースを持たせることで、ネットワークを使用するホスト情報、使用しているサービス資源の情報を見ることができる

この利点により、本研究では DHCPv6 を用いることで、前章に挙げた目的を達成できると考え、これを用いて IP アドレス、サービス資源の割当て、管理機構を実装する。

5. ホスト情報の定義

今回使用するホスト情報は、「DHCPv6 が IP アドレス、サービス資源を与えるネットワーク範囲上の識別子」と定義する。ネクストホップルータより先のホスト、ルータには資源は割当てないものとする。

管理者がそのポリシーに応じて対象とするネットワーク上の IP アドレス、サービス資源を割当てるには、定義されたネットワーク上でホスト情報が一意の識別子で表されなければならない。

そのため今回は、IPv6 リンクローカルアドレスを識別子として定義した。これを識別子とした理由を述べる。

- ホストが2つ以上のネットワークインターフェースカード(NIC)を使用している場合、それを区別して定義することが必要となる。
- リンクローカルアドレスは、クライアントが明示的に変更しない限りの NIC の Ethernet アドレスから生成される。Ethernet アドレスは IEEE の EUI-48 フォーマットに基づき一意の識別子だと定義されている。
- もし Ethernet アドレスが重複した場合リンクローカル

アドレスは変更されるため、Ethernet アドレスより一意である可能性は高い。

上記の理由より、リンクローカルアドレスは一意であると定義した。また、DHCPv6 のメッセージ交換の際、送信元または送信先アドレスとしてリンクローカルアドレスが使用されるため、サーバでの取得が容易であることも挙げられる。これにより、リンクローカルアドレスをホスト識別子として定義し利用することにした。

6. 管理者ポリシーの表現の定義

管理者が反映したいと思うポリシーは千差万別であり、その表現方法も様々である。

本研究では5章であげたホスト情報によって、割当てる IP アドレス、サービス資源の情報を区別することで、ポリシーを表現することにした。その理由は、このモデルではホスト情報がネットワーク上で識別される情報の中で最小単位となるからである。

7. サービス資源の割当て

前述のとおり、NDP の RA メッセージではサービス資源の割当てを行うことができない。そのため、特に DNS のサービスロケーションが問題となり、現在 IETF にて標準化の議論が行われている。Anycast address [7] や DHCPv6 Service Location Protocol [8] などの方式が提案されているが、本研究では DHCPv6 を使用するため、サービス資源の割当ても DHCPv6 で行うことが可能である。

また、今回は DNS をサービス資源として想定し、実装を行う。しかし、サービスの IP アドレスを通知することでそのサービスが使用できるようになるものは本研究のスコープとしてみなすことができる。

8. 設 計

8.1 IPv6 アドレス割り当て機能

IPv6 アドレスの割当てに関しては、すでに linux 上などで開発が行われている。そのため、他の実装との互換性を計るため、RFC に沿った DHCPv6 のメッセージング、パケットフォーマットを使用することが必須となる。

図3に DHCPv6 のメッセージフローを載せる。

ゲートウェイ情報など RFC のパケットフォーマットに載っていない情報の送信は、IP アドレスを送る IA Address Option の IAaddr-options field にいれることで解決を図った。

DNS などのサービス資源情報については、RFC にオプションとそのフォーマットが規定されているのでそれに従った。

8.2 ポリシーに応じた IP アドレス、サービス資源割当て設定

本システムでは、管理者のポリシーに応じて、IP アドレスやサービス資源を個別に割当てる必要がある。

ホスト識別子であるリンクローカルアドレスと IP アドレス、サービス資源を結びつけるという方法でポリシーの実現を可能とする。サーバにホスト識別子とそのホストに割当てたい情報をサーバの設定ファイルに書くという方法で実現させた。

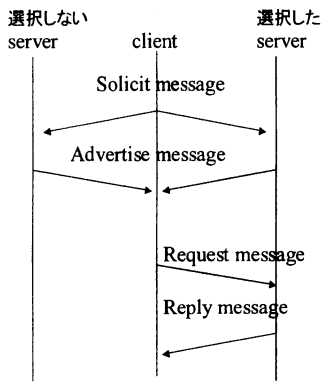


図3 DHCPv6 メッセージ交換

```
interface interface_name {
    range from IPv6 address to IPv6 address;
    prefixlen /64;
    vlttime 300;
    pltime 200;
    gateway gateway_address;
    option domain-name-servers name_server;
};
host hostname1
link-local link-local_address;
fixed-addr IPv6 address;
Gateway gateway_address;
};
```

図4 サーバ側設定ファイル

図4にサーバ側の設定ファイルの例を挙げる。

図4の場合、interfaceのセクションはホスト群全体の設定である。ここで、配るアドレスの範囲(range)、プレフィックス(prefix)などのアドレス情報、優先生存時間(pltime)、可能生存時間(vlttime)、ゲートウェイサーバのアドレス(gateway)、ネームサーバ(domain-name-servers)などのサービス資源情報を設定する。

hostのセクションは個別ホストに対応するセクションである。ホスト識別子であるリンクローカルアドレス(link-local)とそのホストに配りたい情報(fixed-addr,gateway)を書くことにより、ホストごとのサービス資源割当てを可能とする。hostセクションに書かれていない情報は、interfaceセクションに書かれている情報を使用することにより、管理者の過度な負担と設定ファイルの増大化を防ぐ。

8.3 ホスト情報、使用サービス資源データベース

IPアドレスをリースしているホスト情報と、そのホストが使用しているサービス資源を、管理者が把握できることが必要である。

今回は、サーバに、貸し出しているホスト情報と使用しているサービス資源をdouble linked listでつなぎ、それを整形してファイルに書き出す。IPアドレス貸出期限がタイムアウトになったもの、Releaseメッセージがサーバに送られてきたものに関しては、リストから抜き、ファイルも上書きされる。

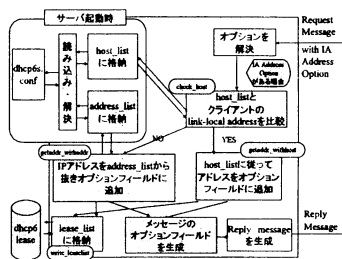


図5 サーバ側処理概要図

9. 実装

前節をふまえ、KAME Project [9] のDHCPv6クライアント、サーバに機能を付加する形で実装を行った。

現在、BSDシステム用に、KAME ProjectでもDHCPv6の開発が進められているが、現在はIPアドレス割当ての機能が実装されていない。しかし、これはサーバとして利用されることも多いBSDシステムにとって必要な機能であり、使用できることが望ましい。そこで実装は、KAME ProjectのDHCPv6にアドレス割当て機能を付加する形で実装を行った。

実装環境は以下のとおりである。

- FreeBSD 4.9-RELEASE
- KAME DHCPv6
- C言語

9.1 DHCPv6 server

サーバにはhost_list,address_list,lease_listという3つのアドレス、サービス資源情報のリストを持たせた。host_list,lease_listは同じデータ構造とし、double linked listとして実装した。

データ構造は以下の通りに定義した。

```
struct dhcp6_addrinfo{
    struct sockaddr *linklocal;
    struct in6_addr *in6;
    int pltime;
    int vlttime;
    int prefixlen;
    struct in6_addr gateway;
    struct in6_addr dns-server;
    struct dhcp6_addrinfo *prev;
    struct dhcp6_addrinfo *next;
};
```

また、address_listのデータ構造は以下の通りに定義した。

```
struct dhcp6_addrlist{
    struct in6_addr in6;
    struct dhcp6_addrlist *prev;
    struct dhcp6_addrlist *next;
};
```

図5にサーバ側のRequest、Replyメッセージ間の処理概要図をあげる。

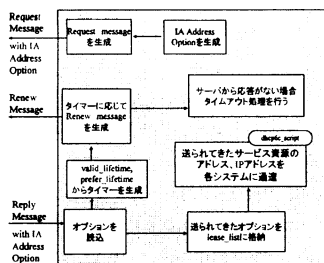


図 6 クライアント側処理概要図

サーバは起動時に、dhcp6s.conf という設定ファイルを読み込む。設定ファイルの書式は図 4 の通りである。設定ファイルの interface セクション中 range に書かれているアドレスを一つずつリストにして address_list に格納し、host セクションの情報を host_list に格納する。また、起動時に dhcp6.lease を読み込んで lease_list に格納し、格納した IP アドレスを address_list より抜く。これにより、クライアントはサーバが再起動した場合も前回と同じサービス資源を割当てられることになる。

クライアントの Request メッセージの中に IA Address Option が含まれていた場合、まず host_list に入っているリンクローカルアドレスとクライアントのリンクローカルアドレスを照合する。二つが適合した場合、host_list に従って IP アドレス、サービス資源をオプションフィールドに追加すると同時に、lease_list に追加する。適合しない場合は、address_list から一つを抜き出して Reply メッセージのオプションフィールドと、lease_list に追加する。

lease_list は更新される度に dhcp6.lease に整形して書き出す。今回は更新される度にテキストファイルに上書きされることで実現させた。

IP アドレス、サービス資源等必要な情報をオプションにいれると、それを Reply メッセージとしてクライアントに送信する。

また、クライアントから Renew メッセージが来た場合は、lease_list, host_list と照合を行う。host_list と照合する理由は、サーバが再起動した場合、新たなホスト情報とサービス情報の組み合わせが追加されている可能性があるからである。

また、新たにポリシーを追加したい場合は、設定ファイルに追加し、DHCP サーバを再起動することで可能となる。設定ファイルが新たに読み込まれることで、次の Renew メッセージが該当ホストから来た場合、新たな設定をオプションフィールドに入れた Reply メッセージが送られる。

9.2 DHCPv6 client

図 6 にクライアントの Request メッセージと Reply, Renew メッセージの処理概要図を載せる。

Reply メッセージから送られてきたサービス資源を読み込むと、クライアントはオプションの情報を自分の lease_list に入れ、該当するシステムにサービス資源のアドレスを通知し、インターフェースにアドレスをつける。それと同時に、IP アド

レスの貸出時間 (valid_lifetime, prefer_lifetime) を計算し、タイマーに応じてサーバに Renew メッセージを送る。Renew メッセージに対応した Reply メッセージが来た場合はそのまま貸出が継続され、もし Reply メッセージでサービス資源の変更が求められた場合はそれに従う。数回に渡り Renew メッセージを送っても Reply メッセージが来ない場合は、タイムアウト処理を行い、サービス資源、IP アドレスを解放する。

10. 評価

この実装により、サーバによって IP アドレス、サービス資源の割当てを振り分けることが可能となり、管理者がポリシーによってアドレスを割当てることが可能となった。また、サーバの持っている dhcp6.lease というデータベースにより、管理者がホストが現在使っているサービス資源を監視できるようになった。

今回、DHCPv6 のプロトコルを使用してこの実装を行った。既存の枠組みを使用して目的を達成することによって互換性ができ、stateful という観点からも良い選択だったと評価できる。

また、Plug & Play については、本システムを用いることで可能ではないが、ネットワークインターフェースを up したときに本システムを起動するように設定を行うことで可能になると考えている。

11. 今後の課題

11.1 運用

本システムを実際のネットワーク上で運用することが必要となる。現在はテストベッド上で実験するに留まっているが、今後、最大割当て数 60 台程度の研究室内のネットワークでの運用を予定している。運用上の問題点、ユーザビリティを洗いだし、実装に反映させる。

また、ホスト数が 100 台、1000 台と増加したときの運用負荷実験を行うことも必要である。特に、IP アドレス、ホスト情報等が入るサーバのデータベースに関しては、今回は linked-list を使って実装したため、台数が増えるにつれて処理性能が著しく遅くなることが予想される。そのため、データ構造のつなぎ方に関しては再設計を行い、radix tree 型の検索アルゴリズムに実装しなおすことを予定している。

11.2 他実装との互換性

今回は FreeBSD 上で実装を行ったが、アドレス割当てを伴った DHCPv6 クライアントの実装は他の OS でもあるため、どのクライアントでも今回実装した DHCPv6 サーバとメッセージのやりとりが可能であることが望ましい。そのため、他の DHCP クライアントの実装とも互換性のテストを行うことが必要である。

また、KAME Project は FreeBSD だけではなく他の BSD システム用のシステムもあるため、本システムをそちら移植することも課題として考える。

11.3 DHCP サーバが複数台あったときの挙動の考察

DHCP サーバが 2 台以上存在した場合、クライアント側が DHCP サーバを選び 1 台のみ使用すると RFC で規定されてい

る。そのため、もし悪意のある DHCP サーバが管理するネットワーク上に置かれても、クライアントに信用できるサーバを選ばせることで意図しないネットワークにクライアントを接続させることを防ぐ。どれが信用できるサーバであるかという選択の基準は、本システムにより提供できると考える。Advertise メッセージにサーバ自身の情報や、そのサーバが信用できるといった情報を含めることで Request メッセージをサーバに送信するときのクライアントの判断材料となる。

また、2 台以上の DHCPv6 サーバを管理者が運用し、負荷分散を図る場合の考察も必要である。その場合、両方も信用できるサーバであるため、どちらのサーバを選択するかはサーバが判断するべきである。ホスト情報と照らし合わせることで判断は可能と推測されるが、この点については更なる考察が必要だと考える。また、複数台のサーバのデータベースを統合することで、管理者が管理しやすくなるなどの利点もあると考えられ、その点についても今後の課題として挙げられる。

11.4 ポリシ定義、表現

今回は、ホスト情報をリンクローカルアドレスとして定義し、それに基づいて最小単位でポリシを反映した。しかし、この方法はホスト 1 つ 1 つに関して情報を設定ファイルに記入することが必要となり、運用上のオーバーヘッドが大きいと考える。そのため、ポリシが同じホストに対して適切なグループ化を行い、それに応じたプレフィックスやサービス資源を割当てる仕組みが重要となる。このグループ化の方法についても今後考察が必要である。

また、ポリシの定義は、管理者によって異なることがほとんどであり、ポリシの判断材料も異なる。しかし、システム側で判断材料の目安となるものを入れることで、管理者の判断が楽になり、自動化が可能になると考える。そのため、管理者がそのポリシを適用したいと考える判断材料についての考察が必要であると考える。具体的には

- メッセージのバケット中にあるビットを立てることで判断
- OS によって判断
- あるアプリケーションで検知した場合に判断

などが挙げられる。

11.5 他のアプリケーションとの連携

dhcpv6 のデータベースや設定ファイルと他のアプリケーションを絡めることにより、よりダイナミックなサービス資源の割り振りや管理が可能になると考える。そのために、他のアプリケーションが使用しやすい形態を考えることが重要である。

11.6 サービス資源割当て

今回は、サービス資源として DNS サーバに関しての割当てを実装した。しかし、サーバの IP アドレスを通知することでサービスが使用できるシステムは多い。本システムの実装では、データ構造を定義したため、その構造体中に通知したいアドレスを定義することで、他のサービスも割当てが可能であると考える。

12. ま と め

本研究では、stateless モデルでの IP アドレスの割当ての問題点を挙げ、それと比較する形で stateful アドレスの有用性を示した。また、ホスト情報によって管理者がサービスを振り分け、ホストが使用しているサービスを把握できることへの利点を示した。そして実際に、DHCPv6 を使用した IP アドレス、サービス資源の割当て・管理方法を提案し、識別子としてリンクローカルアドレスを利用することで設計、実装を行った。

これにより、管理者がポリシに応じて自分の管理するネットワークにおいて、配布したい IP アドレス、サービス資源を、ホスト情報に応じて振り分け、場合によっては制限をすることが可能となる。また、本システムが持つデータベースにより、ネットワークを使用しているホスト情報とそのホストが使用しているサービスを管理者が把握できるようになる。

これにより、図 2 であげたような状況の場合の管理者ニーズを満たすことができ、ネットワークの負荷分散や状況把握等に効果が期待できる。また、何らかの理由から 1 本の物理的な線路上でセグメントを複数もち、動的にサービス資源を変更したい場合にもこれは効力を発揮する。例えば、ウィルス等に感染したホストが出た場合、そのホストへのサービス資源割当てを動的に切断することでネットワーク上からの切り離しが可能になるなどの具体的な成果が期待できる。

文 献

- [1] S.Deering,R.Hinden, Internet Protocol, Version 6 (IPv6) Specification, <http://www.ietf.org/rfc2460.txt>, December,1998
- [2] S.Thomson,T.Narten,IPv6 Stateless Address Autoconfiguration, <http://www.ietf.org/rfc2462.txt>, December,199
- [3] R.Hinden,S.Deering, Internet Protocol Version 6 (IPv6) Addressing Architecture, <http://www.ietf.org/rfc3513.txt>, April 2003
- [4] P.Mockapetris, DOMAIN NAMES-CONCEPTS AND FACILITIES, <http://www.ietf.org/rfc1034.txt>, November 1987
- [5] P.Mockapetris, DOMAIN NAMES-IMPLEMENTATION AND SPECIFICATION <http://www.ietf.org/rfc1035.txt>, November 1987
- [6] R.Droms,Ed. ,J.Bound,B.Volz,T.Lemon,C.Perkins,M.Carney, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), <http://www.ietf.org/rfc/rfc3315.txt>, July 2003
- [7] Jun-ichiro itojun Hagino,K.Ettikan.An analysis of IPv6 anycast, <http://www.ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-anycast-analysis-02.txt>, June.2003
- [8] E.Guttman,C.Perkins,J.Verizades.M.Day, Service Location Protocol Version 2, <http://www.ietf.org/rfc/rfc2608.txt>, June.1999
- [9] KAME Project,<http://www.kame.net/>
- [10] IETF DHC working group,<http://www.ietf.org/html.charters/dhc-charter.html>
- [11] DHCPv6 Home Page, <http://www.dhcpv6.org/>
- [12] 増田康人,長橋賢吾,有賀征爾,使って学ぶ IPv6, ASCII, 東京,2002