

卒業制作 2004 年度 (平成 16 年度)

DHCP を利用した動的な資源割当手法に関する研究

指導教員

徳田 英幸

村井 純

楠本 博之

中村 修

南 政樹

慶應義塾大学 総合政策学部

門田美由紀

s01245mk@sfc.keio.ac.jp

平成 16 年 12 月 29 日

概要

現在、マイクロノードや情報家電などが普及しつつあり、日常生活におけるネットワークの利用形態が多様化してきた。そのため、現状のように利用するサービス資源情報をユーザが把握した上で設定を行うのではなく、機器同士が協調動作を行うことで資源情報を設定する機構が必要となってきた。

これらの背景を踏まえ、本研究では家庭内 LAN において、ユーザの設定負担を軽減した資源割り当てモデルを策定した。まず、ユーザのニーズを満たすためのモデル策定要件を挙げ、資源割り当てのために必要な資源情報の定義を行った。その後、要件に基づき資源情報を集中して管理するモデルが要件を満たすと考え、サービス資源の登録、IP アドレスの割り当て、サービス資源の配布、ポリシーによる振り分け、動的な解除の 5 つを必要な機能として設定した。その後、DHCPv6 を拡張した資源情報集中管理システムについて設計、実装、評価を行った。これにより、ユーザに負担をかけることのない資源割り当て手法を確立した。

キーワード

1, サービス探索 2, 資源割り当て 3, DHCPv6 4, 家庭内 LAN

慶應義塾大学 総合政策学部
門田 美由紀

abstract

Recently, popularization of micronodes and information appliances changes the use form of network in daily life. To use those appliances, users must know the service resource information and set them up by themselves. Therefore mechanism in which micronodes and appliances collaborate each other to set resource information by itself is needed.

In this research, the resource allocation model which reduces user's cost to set up the nodes are designed.

First, the requirements which meet user's demands is defined. Then five functions, registration of service resource, allocation of IP address, distribution of service resources, policy based sorting and dynamic release of resources, those are necessary to meet the requirements are designed. In addition, centralized resource management system which is extended from DHCPv6 is implemented and evaluated in this research.

Through this research, the method to allocate resources without increasing cost is created.

Keywords

1, Service Discovery 2, Resource allocation 3, DHCPv6 4, Home Network

Faculty of Policy Management, Keio University
Miyuki Kadota

目次

第 1 章	序論	1
1.1	本研究の背景	1
1.1.1	コビキタスコンピューティング環境	1
1.1.2	家庭内 LAN とノードの多様化	1
1.2	ネットワークの利用対象	2
1.3	研究目的	3
1.4	本論文の構成	4
第 2 章	問題点	5
2.1	現状の資源設定	5
2.2	家庭内ネットワークにおける管理者とユーザ	7
2.3	今後の家庭内ネットワークの資源設定における問題点	7
2.3.1	ユーザによる資源割り当ての負担	7
2.3.2	管理者におけるポリシーの設定と反映	7
第 3 章	家庭内 LAN における資源割当モデル	9
3.1	家庭内 LAN 環境下での資源設定の考察	9
3.2	モデル策定のための要件	9
3.3	集中管理型モデル	9
3.3.1	サービス資源の登録	10
3.3.2	IP アドレスの割り当て	11
3.3.3	サービス資源の配布	11
3.3.4	ポリシーによる振り分け	12
3.3.5	動的な解除	12
3.4	DHCP を利用した資源割当	13
第 4 章	関連研究	15
4.1	Jini	15
4.2	Rendezvous	16
4.3	UPnP	16
4.4	SLP	17
第 5 章	設計	19
5.1	概要	19
5.2	設計要件	19
5.2.1	IP アドレス貸し出し	19

5.2.2	サービス資源情報登録	20
5.2.3	資源情報設定機能	21
5.2.4	資源情報探索機能	21
5.2.5	資源返却機能	22
5.2.6	資源情報参照機能	23
5.3	考慮する点	23
5.3.1	ホスト識別子	23
5.3.2	各アプリケーションへの登録	24
5.3.3	サービス資源表記	24
第 6 章	実装	25
6.1	実装概要	25
6.2	実装環境	25
6.3	DHCPv6 サーバ	25
6.3.1	設定ファイルとポリシーの設定	25
6.3.2	IPv6 アドレス割り当て	26
6.3.3	サービス資源登録	27
6.3.4	サービス資源探索	28
6.3.5	貸し出しリストからの解除	28
6.3.6	ログへの書き込み	29
6.4	DHCPv6 クライアント	29
6.5	クライアント側設定ファイル	29
6.5.1	サービス資源情報の登録	29
6.5.2	サービス資源の探索	30
6.5.3	資源のアプリケーションへの追加	30
第 7 章	評価	32
7.1	モデルとの合致	32
7.1.1	自動設定	32
7.1.2	ポリシーの反映	32
7.2	ブートストラップまでの性能評価	33
第 8 章	結論	34
8.1	今後の課題	34
8.1.1	ポリシーの表現	34
8.1.2	資源情報	34
8.1.3	セキュリティ	34
8.2	まとめ	34

目 次

1.1	家庭内 LAN の構成例	2
1.2	将来的な家庭内 LAN の構成例	3
2.1	資源設定の手順	5
2.2	サービス資源利用の現状	6
3.1	サービス資源の登録	10
3.2	IP アドレスの振り分け	11
3.3	サービス資源の配布	12
3.4	ポリシーによる振り分け	13
3.5	動的な解除	14
4.1	Jini の動作概要	15
4.2	Directory Agent を利用した SLP	17
4.3	サービス資源情報のフォーマット	17
4.4	SLP フォーマットの例	17
5.1	DHCPv6 のメッセージ交換	19
5.2	DHCPv6 DNS Configuration Option	20
5.3	Service Option	20
5.4	Service Option を利用したサービス登録	21
5.5	Service Option を使ったサービス探索	22
5.6	IP アドレスの返却	23
6.1	DHCPv6 サーバ側設定ファイルの例	26
6.2	address_list データ構造	27
6.3	Request メッセージ受信後のアドレス割り当て動作	27
6.4	IA Address Option	28
6.5	service_list データ構造	28
6.6	ログ出力結果	29
6.7	DHCPv6 クライアント側設定ファイルの例	30
6.8	DHCPv6 クライアントスクリプトの例	31

表 目 次

4.1	Rendezvous で使用されるプロトコル	16
5.1	サービス資源表記の例	24
6.1	本システムの実装環境	25
7.1	モデルとの合致	32
7.2	総パケット交換数	33

第1章 序論

本章では、本研究の背景、および研究の目的について述べる。また、本論文の構成について述べる。

1.1 本研究の背景

1.1.1 コビキタスコンピューティング環境

近年,DSL,FTTH,CATV インターネットなど広帯域な回線が安価に提供されるようになり,学校,企業のみならず,店舗や街頭,家庭など様々な場所でネットワークの接続性が提供されるようになった。平成 16 年度版 情報通信白書 [1] によると,ブロードバンド契約数は約 1,500 万に達し,利用人口は 2,607 万人と推計される。これは今後も更に増加する見込みであり,様々な場所・範囲・規模で接続性が提供されると考えられる。これを利用し,将来的に,誰でも,いつでも,どこでも,コンピュータを意識せずに用途に応じてサービスを楽しむことができる,コビキタスコンピューティング環境の実現が現実味を帯びてくる。これにより,パーソナルコンピュータだけではなく,電話,家庭用電化製品など様々な機器が相互に接続し,協調した動作が可能となる。すでに, UbiLab[2], Ubiquitous Open Platform Forum[3] など多くの研究や標準化が行われている。

1.1.2 家庭内 LAN とノードの多様化

広帯域回線の普及に伴い,ブロードバンド対応ルータ・モデムなどネットワーク接続機器も安価に提供されるようになってきている。

現在,家庭内 LAN の対外接続は ADSL や FTTH など差はあるが,ほとんどが ISP より IPv4 アドレスを動的に 1 つ割り当てられている。図 1.1 に,家庭内ネットワークの一般的な構成図を載せる。

複数台の機器を接続する時は,ブロードバンドルータの NAT(Network Address Translation)[4] 機能を利用してプライベート IP アドレスを機器に割り当てる方法が一般的である。IP アドレスは, Dynamic Host Configuration Protocol(DHCP)[5] によって各機器に動的に配布する。また,無線 LAN アクセスポイントを利用することで,無線 LAN ネットワークを作る家庭も多くなっている。

IPv6(Internet Protocol version 6)[6] の普及により,家庭内のネットワーク環境は大きく変容する。例えば,IPv6 はアドレス空間が広大なため,従来のように NAT 機能を使わなくても, Prefix delegation[7] を利用して,家庭内にグローバル IP アドレスでのネットワークを構築することが可能となる。

また,現在は家庭内 LAN に接続されている機器の大部分がパーソナルコンピュータであるが,今後はセンサーが取得した情報を一定時間毎に送信する,特定の制御を行うなど,限られた機能

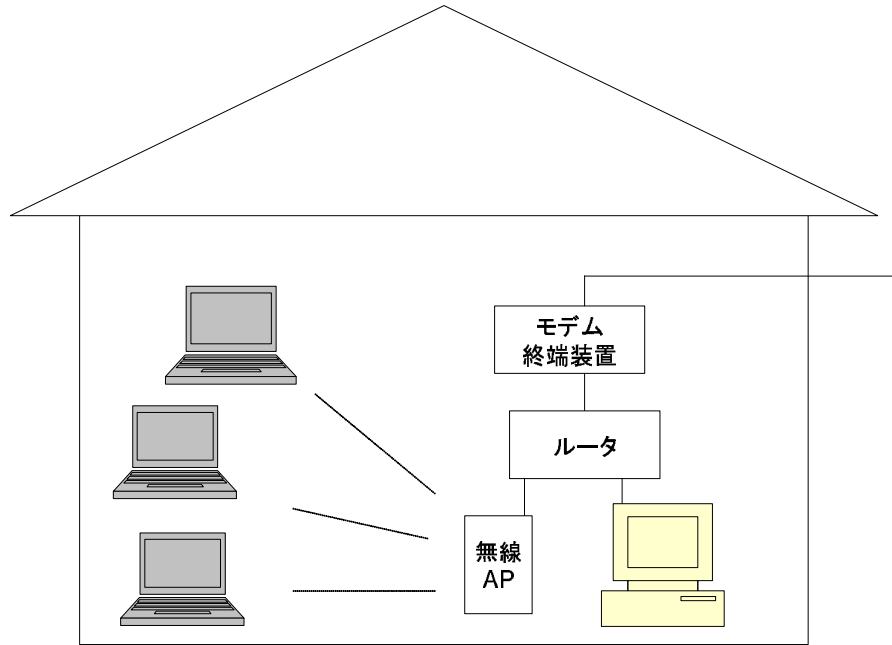


図 1.1: 家庭内 LAN の構成例

しか持たない小型化されたノード (マイクロノード) や、情報家電、組み込み機器が増加する。センサで取得した情報を用いて機器同士が協調動作を行ったり、遠隔から情報を取得したりと、これまでよりダイナミックなネットワークの利用方法が可能となる。

将来的に家庭内 LAN に接続される機器のイメージを図 1.2 に示す。

このように、多くの機器がネットワークに接続されることで、サービスは細分化され、サービスを提供するノード、利用するノードの数も飛躍的に増加する。さらにノードは一つのネットワークで使用されるわけではなく、持ち運ばれた先々で利用される可能性もある。

1.2 ネットワークの利用対象

従来、ネットワークを利用するためにはコンピュータを利用する必要があったため、コンピュータを操作することが大前提となってきた。前述したマイクロノード等の普及により、コンピュータを持たなくても、ネットワークを利用して情報を送信することも可能となってくる。

今後ネットワークを活用するモデルを、高齢者世帯を例として挙げる。

高齢者夫婦世帯の場合

血圧情報を 1 日 1 回自宅のホームサーバに蓄積している。その情報は医者にも月 1 回送信されている。また、ネットワーク対応ポットを利用し、遠隔に住む息子夫婦にポットの使用情報を送信している。ポットが 1 日以上使用されなかったら、息子夫婦から連絡が入る約束をしている。今度、血圧の他に、毎日の歩数もグラフで見たいと思い、新たに万歩計を購入した。夜になると、

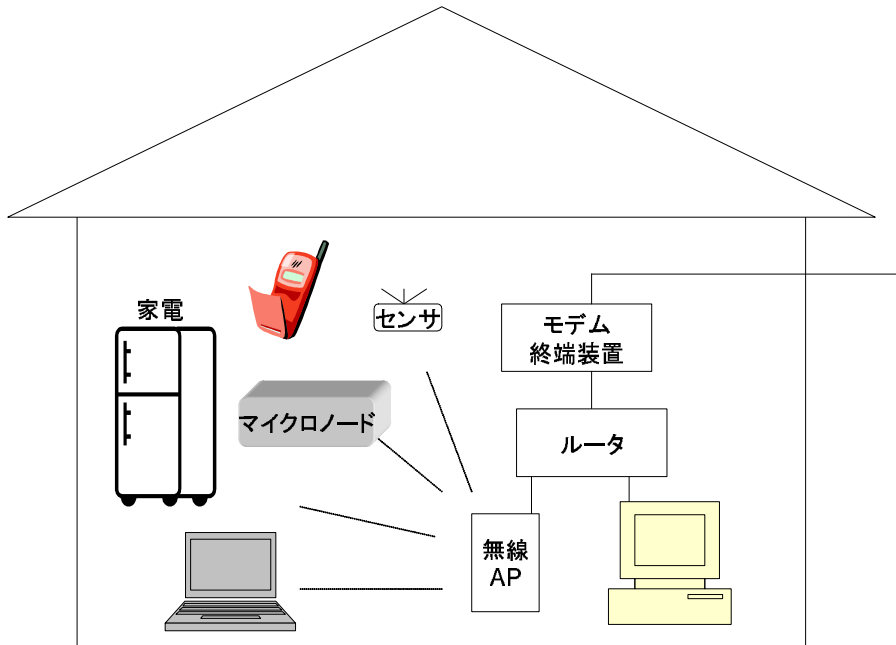


図 1.2: 将来的な家庭内 LAN の構成例

今日の歩数がグラフになってホームサーバのディスプレイに表示される。

このように、血圧計、ポット、万歩計などセンサーで取得された情報を送信することで、医療、介護分野に情報が活用され、自らの生活にフィードバックされる。

しかし、高齢者に限らず、ネットワーク・コンピュータの知識がない人が、センサーを LAN 上で利用するための設定を行うのは難しい。できる限り、購入してきた状態のまま利用したいのが実情である。さらに、「血圧はホームサーバ」「ポットの使用情報は息子の家」など、情報によって送信先は変わってくる。

このように、自らの生活に密接に関わるネットワーク活用方法は、今後増加する。しかし、各ノードがプレインストールされた状態で動作するわけではない。ネットワークの状態に応じてそのネットワークの利用情報、サービス資源の利用情報を設定する必要がある。これらを特に高齢者のような知識のないユーザが正確に 1 台ずつ設定していくことは、利便性を阻害する結果となる。

1.3 研究目的

今後、家庭内での LAN はさらに普及し、接続される機器は増加し、提供されるサービス資源は細分化されていく。このとき、各機器が利用するサービス資源の利用情報、ネットワークの設定情報をどのように設定していくかを考える必要がある。前述したように、ユーザが手動で行う方法では利便性を阻害する。そのため、機器同士が相互に情報を交換し設定を行うことが望ましい。

また、情報の送信先は、機器同士が決定する方が良い場合もあれば、ユーザが設定したい場合

も存在する。

本研究では、家庭内 LAN において動的な資源割当を行うことで、機器同士が相互に情報を交換するモデルを策定することを目的とする。また、ユーザのポリシーに基づいて、情報を設定できる機能を持たせる。

そのため、機器のサービス資源の情報、接続性の情報等を一元的に管理し、必要に応じて再配布することでこれを解決する。

1.4 本論文の構成

本論文は以下のように構成される。

第 2 章では、家庭内 LAN の現状と、資源設定の現状を述べ、現状での問題点を挙げる。第 4 章では、現状で用いられている家庭ネットワーク設定手法について述べ、その問題点について整理する。第 3 章では、問題点を踏まえて家庭内ネットワークで設定を行う際の要求事項を挙げ、考えられるシステムのモデルを提案する。第 5 章では、前章で挙げたモデルに基づいてシステムを設計し、その詳細を述べる。第 6 章では、本システムでの実装を述べる。第 7 章では、本研究のモデルとそれに基づいたシステムについて評価を行う。第 8 章では本研究のまとめと今後の課題を述べ、本研究での結論とする。

第2章 問題点

本章では、現状の家庭内でのネットワーク運用の行われ方を整理し、その上で問題点を列挙する。

2.1 現状の資源設定

現在、アプリケーションがネットワーク上で提供されているサービス資源を利用するためには、図 2.1 の手順を、下から上に、段階を追って踏む必要がある。各手順は段階を飛ばすことが出来ず、全て手動で設定を行う場合でも、必ず行わなければならない。

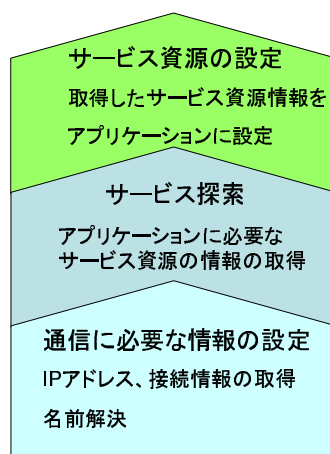


図 2.1: 資源設定の手順

1. 通信に必要な情報の設定

通信に必要な情報としては、IP アドレスの他、サブネットマスクやデフォルトゲートウェイなどが挙げられる。この情報を設定する方法として、IPv4 では DHCP を利用する方法、IPv6 では Neighbor Discovery Protocol[8] の Router Advertisement メッセージを利用する方法が一般的である。上記の機構を用いることで、ユーザが手動で IP アドレスを設定する必要はない。また、名前解決に必要な DNS サーバを設定することも必要である。

2. サービス探索

ネットワークへの接続が完了した後、アプリケーションに必要な情報を取得することが必要である。必要な情報とは、アプリケーションが利用するサーバの資源名、IP アドレスである。

3. サービス資源の設定

2 で取得した情報を元に、アプリケーションが利用するサーバ情報をアプリケーションに

設定することで、サーバにアプリケーションを通しての通信が可能となる。

本研究では、あるノードが他のノードに提供できるサービスを「サービス資源」と定義する。機器に必要なサービス資源を把握し、設定する方法はまだ手動に頼っているのが事実である。現状でのサービス資源の設定を、図 2.2のネットワークを用い、「提供されているサービス資源を他のノードに設定する場合」「サーバノードのサービスの継続が不可能になった場合」「同じサービスを提供するノードが2つ以上ある場合」(振り分け)の設定に関する代表的な3点について説明する。

ネットワーク A にはノード A,B,C の3台が接続されていて、それぞれにサービス資源を稼働させている。

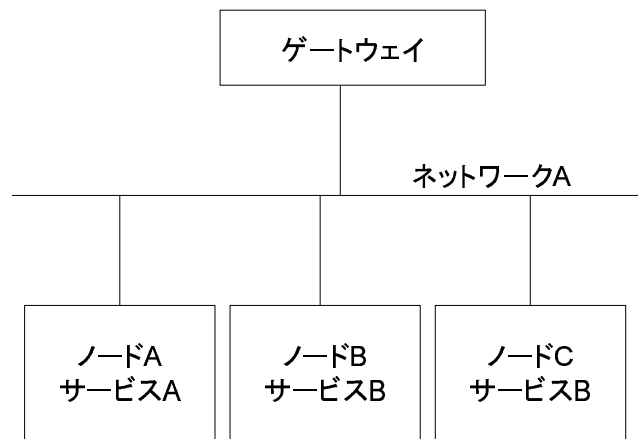


図 2.2: サービス資源利用の現状

提供されているサービス資源を他のノードに設定する場合

サーバノード A が提供しているサービス A をクライアントノード B に設定する場合を考える。ユーザは、ネットワーク上に接続されたノード A、ノード B に対してノード A の IP アドレスやサービス資源名、他の必要なデバイス情報等を具体的に把握し、ノード B に、ノード A の資源情報を設定する。

このように現状では、ユーザはサービス資源の内容を詳細に把握している必要があり、それを元にサービス資源を利用したいノードに設定しなければならない。

サーバノードのサービスの継続が不可能になった場合

サービスを提供しているノード A が、移動、障害など何らかの理由でネットワーク上から切断された場合や、サービス A のアプリケーション障害などでサービスの継続が不可能になる場合が想定される。しかし、ノード B はノード A の状態に関わらずアクションを起こし続ける。ノード B にノード A がサービスを提供していないことを通知する方法は存在せず、ユーザが手動でサービスの提供先を変更することが必要となる。

ノードのサービス稼働を管理者の意向で振り分ける場合

同一のサービス資源 B を、異なるノードが提供している場合、どちらのノードを利用するかは、管理者がノードの状態を把握し、手動で設定する。また、一時的にサービス資源の配布を停止することも管理者が手動で行う。このような、ネットワーク管理者が主体となって稼働状況を決定することを「ポリシー」と呼ぶ。ポリシーを反映したい場合は、管理者がノードの稼働状態を把握した上で、ノードに対して固有のサービス資源を手動で割り当てる必要がある。

2.2 家庭内ネットワークにおける管理者とユーザ

従来のネットワークは専門家のみが利用するものであり、ネットワークに関する専門的な知識を持ち、接続されている機器に対して適切な処置を行う「管理者」が存在し、自らの管理するネットワークへの責任を負っていた。

しかし、現状の家庭内 LAN の管理者は、ネットワークや使用する機器、サービス資源に対して知識のない、利用している家庭のユーザが兼ね備える場合が多い。そのため、機器の管理者を別に設置し、外部から機器に対して管理を行う、監視を行う、などのモデルも考えられる。

2.3 今後の家庭内ネットワークの資源設定における問題点

マイクロノードや組み込み機器の将来的な普及、IPv6 の普及により、家庭内 LAN に接続される機器数は飛躍的に増加すると想定される。また、提供されるサービス資源の内容も細分化される。この状況下で、今後起こる家庭内ネットワークの問題点を挙げる。

2.3.1 ユーザによる資源割り当ての負担

今後、ユーザが個別の機器に対して稼働しているサービス資源、利用すべきサービスの種類を把握し、各デバイスに対して手動で資源割り当てを行う場合、設定に伴う人的コストが上昇すると予想される。また、ユーザへの設定インターフェースが作られていない機器の場合、コンピュータやネットワークに詳しくないユーザが設定するのは困難である。IP アドレスのように、本来全ての資源設定は自動化されるべきである。

2.3.2 管理者におけるポリシーの設定と反映

サービス資源が複数存在することが既知である場合に、その資源を使用する各ノードへ振り分けを行う場合や、一時的にノードへの資源配布を行わなかったりという選択を管理者の側から行いたい場合がある。ポリシーは、同一ネットワーク内だけに限るわけではなく、外部のサービス資源を利用したい場合なども考えられる。

この場合、ユーザは、資源の設定を考えることなく、サービス資源を利用することができ、管理者は設定に関するポリシーを、なるべく容易に反映できる方法が必要である。それには、現状のように個別の機器に手動でサービス資源を割り当てを行うのではなく、ノードと割り当てたいサービス資源の組合せを設定することで、その設定に沿ったサービス資源の配布を行えることが望ましい。

これより、サービス資源の把握をする機構を設置し、ノードから機構へのサービス資源の動的な登録、再配布を行うことが必要とされる。

第3章 家庭内LANにおける資源割当モデル

本研究のアプローチとして、問題点に基づき、家庭内ネットワークにおける資源割当モデルについて考察する。

3.1 家庭内LAN環境下での資源設定の考察

今後の家庭内LANの利用モデルとして、高齢者夫婦世帯のシナリオを1.2章で例として述べた。このシナリオの場合、ユーザは機器をネットワークに接続するということしか行っていない。血圧計はホームサーバを探し、ホームサーバに情報を送るように自動設定する。ポットの送信先は、このネットワークの管理者が息子夫婦の対応アプリケーションに送信するように設定をすることで可能となる。また、新たに加わった万歩計も、ユーザは設定を行わず、万歩計はホームサーバとグラフ作成アプリケーションを利用する。

このように、ユーザは機器を物理的に接続するだけで、ネットワークの管理者と機器が自動的に資源情報を設定し、ユーザは利用するだけという状態が理想である。また、「血圧はホームサーバ」「ポットの使用情報は息子の家」など、ユーザが利用したいサービス資源を設定できることも必要となってくる。

3.2 モデル策定のための要件

前節を踏まえ、家庭内ネットワーク環境資源割り当てモデルを構築するための要件を挙げる。

- 自動設定が可能であること
ユーザに負担をかけないため、ユーザは物理的な接続など必要最低限の動作をし、機器同士が協調動作を行うことで資源情報をやりとりすることが必要である。
- ユーザのポリシーを反映できること
ユーザが手動以外の簡単な方法で、機器の設定に関する割り当て状態を見ることで、ユーザは機器の状態を知ることができる。また個々の設定について、機器に一台ずつ設定を行うのではなく、中央にある機器が振り分けるほうがユーザにとって楽であり、把握する方法も簡単である。

3.3 集中管理型モデル

前述の要件を満たすためには、ソフトウェアがネットワーク上のノードの資源情報を把握し、その情報を再配布する中央管理型のモデルが考えられる。

そのためには、資源情報をすべて集中的に管理し、何らかのイベントがあるごとに対処を行うアプリケーションがネットワーク上に存在することが考えられる。

サービス資源は「あるノードが他のノードに提供できるサービス」と定義した。IPアドレスは有限の資源であるが、他のノードに提供できるサービスではない。また、ホストを識別するための情報も、個々の資源として定義することが出来る。本研究では、同一ネットワーク上に存在する機器のネットワーク範囲上の識別子を「ホスト識別子」と呼ぶ。

集中的に管理を行うために必要な情報は、以下を定義する。

- ノードが提供できるサービス資源
- 利用しているサービス資源
- ホスト識別子
- IPアドレス

これらの情報をノードに対して再配布することで、ノードが必要な情報を網羅出来る。本研究では、以上を総称して「資源情報」と呼ぶ。

以下に、ノード A,B,C を用いて、資源情報集中管理システムが行う機能について述べる。

3.3.1 サービス資源の登録

図 3.1では、各ノードが持っているサービス資源を、資源情報集中管理システムに渡す。渡した情報は、ノードごとにシステム内の資源情報データベースに蓄えられる。

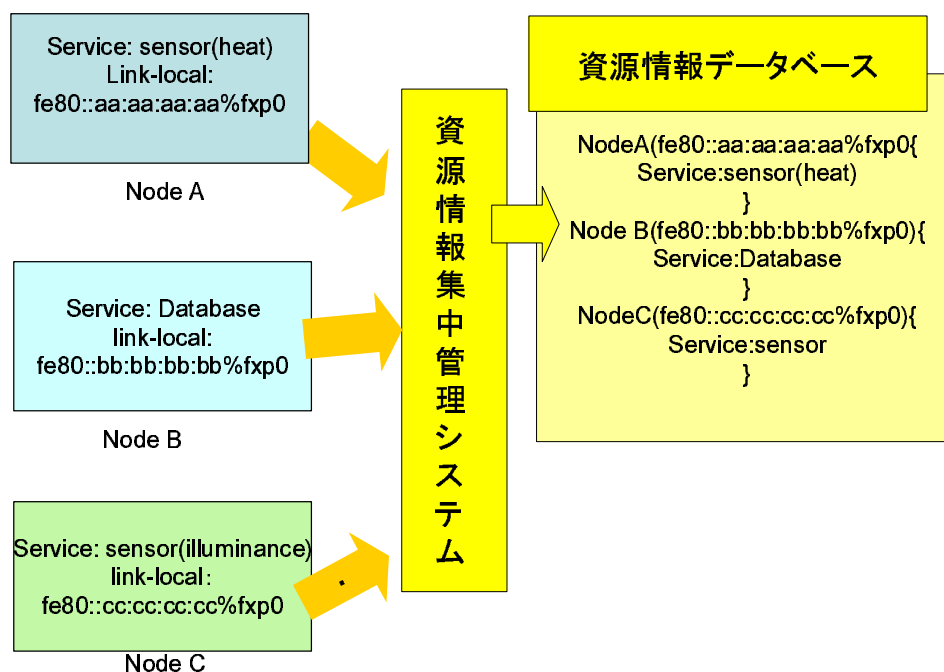


図 3.1: サービス資源の登録

3.3.2 IP アドレスの割り当て

IP アドレスがどのノードに対して割り当てられているかを資源情報集中管理システムが知るためには、以下の3通りの方法が考えられる。

- ノードが割り当てられている IP アドレスを申告する
- 割り当てているアプリケーションが資源情報集中管理システムに情報を渡す
- 資源情報集中管理システムが IP アドレスを割り当てる

前者の2点の場合、IP アドレスを割り当てるためのアプリケーションが必要となって来るため、効率が悪い。

そのため、本モデルでは、3.2のように、資源情報集中管理システムが IP アドレスを割り当てるモデルを採用する。割り当てられた IP アドレスは資源情報に追加される。

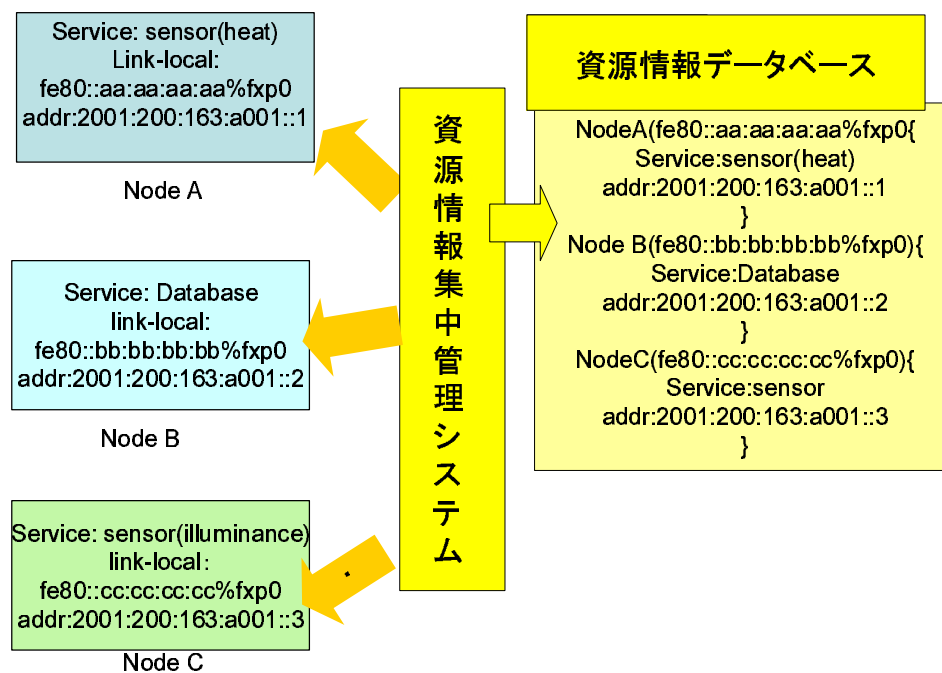


図 3.2: IP アドレスの振り分け

3.3.3 サービス資源の配布

図 3.3は、Node C がサービス資源を利用したい場合を表している。利用したいサービス名を資源情報集中管理システムに問い合わせることで、適切なサービスが利用できるだけの資源情報が問い合わせたノードに送られて来る。

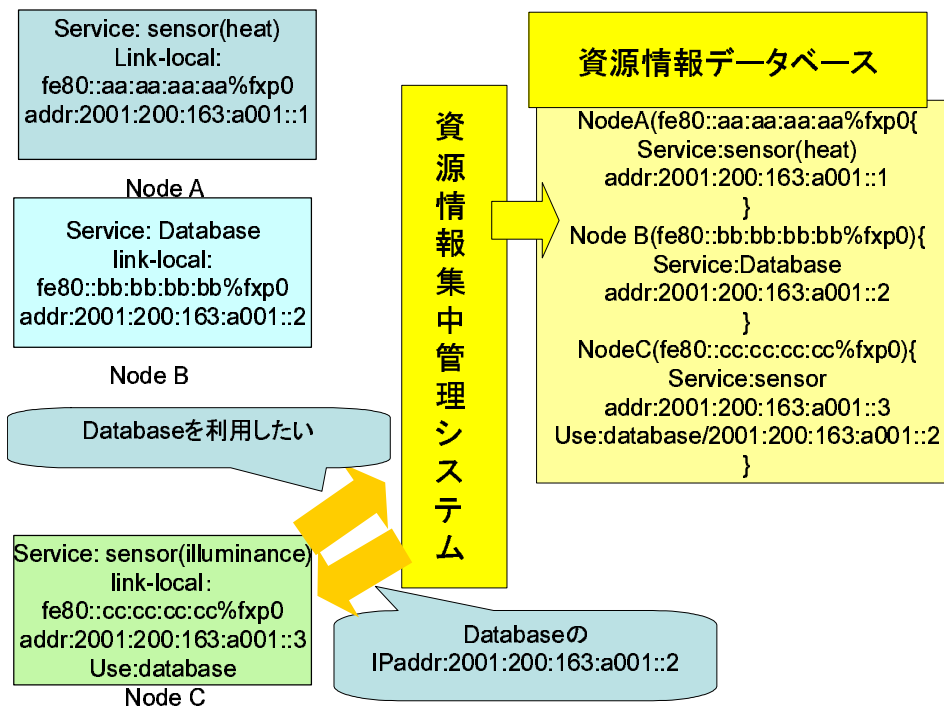


図 3.3: サービス資源の配布

3.3.4 ポリシによる振り分け

管理者がポリシーを反映できることが必要だと前節で述べた。ポリシーを反映する場合、管理者が資源情報集中管理システムに通知することで可能となる。図 3.4では、スコープによってサービス資源を使い分けたり、特定のノードにサービス資源を利用できなくするなどの、手動による制御が可能である。

ポリシーによる振り分けは、以下が考えられる。

- IP アドレス貸し出しの振り分け
「自分の家のノードにのみ IP アドレスを貸し出す」などのポリシーを設定する。
- サービス資源の振り分け
同じ機能を持つサービス資源が 2 個以上あった場合の振り分けや、外部のサービス資源を使用する場合などが考えられる。「血圧センサの情報はクリニックのデータベースに送信する」などのポリシーを設定することが可能となる。

これらについて、ノードごとの資源の振り分けを可能にする。

3.3.5 動的な解除

図 3.5では、Node C が何らかの理由でネットワークから離れる、またはサービス資源を提供できなくなる状態を表している。この場合、資源情報集中管理システムは資源情報から Node C の情報を消去する。これは、Node C が自身がサービス資源を提供できないことを資源情報集中管

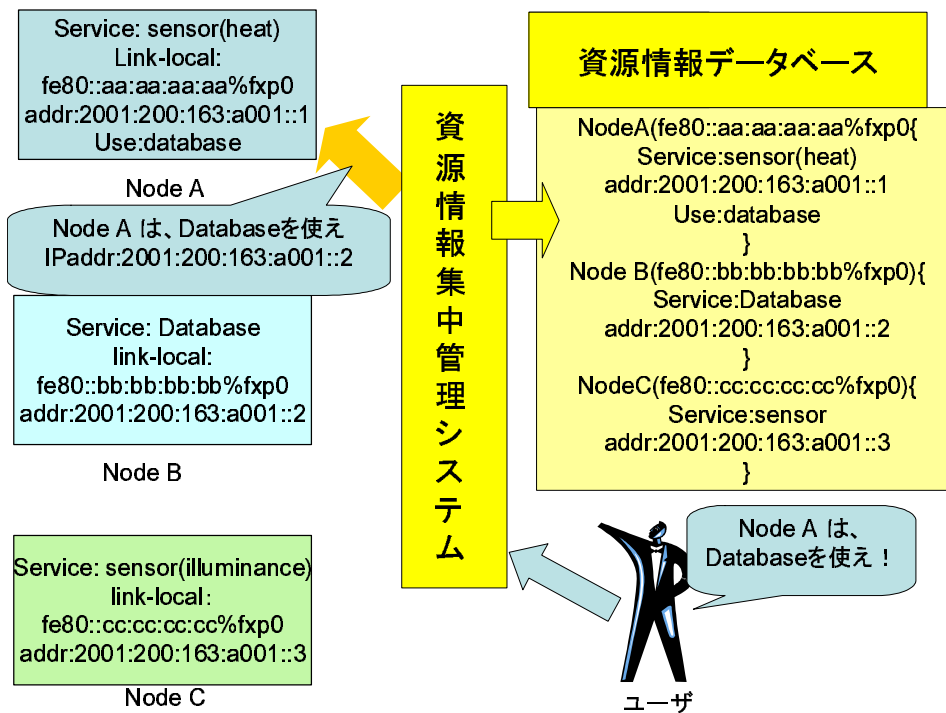


図 3.4: ポリシによる振り分け

理システムに報告するか、資源情報集中管理システムがノードに対して生存しているかどうかを一定時間毎に確認することが必要である。

これにより、現状に存在するノード情報、提供できるサービス情報を把握することができる。

また、今後、IPv6 が普及することを第 2 章で述べたが、そのためにプロトコル非依存のシステムであることも重要である。

3.4 DHCP を利用した資源割当

DHCP は、各ホストにネットワークを利用するための情報を設定するためのプロトコルである。DHCP で行える各種設定を以下に述べる。

- IP アドレスの貸し出し
IP アドレスを DHCP サーバによって各ホストに動的に割り当て、終了時に回収をすることができる。このとき、DHCP サーバは貸し出しているホストのホスト識別子を把握し、定期的にホストの生存情報を確認している。
- ホスト識別子によるサービスの制限
MAC アドレスをホスト識別子として利用し、あるノードにのみ特定の資源を与えることが可能である。また、NetBIOS 等のホスト名も識別子として利用可能である。
- 貸し出しされた資源情報の閲覧
貸し出された IP アドレスは各ホストのホスト識別子と共にログとして書き出されるため、これを利用してホスト情報を把握することができる。

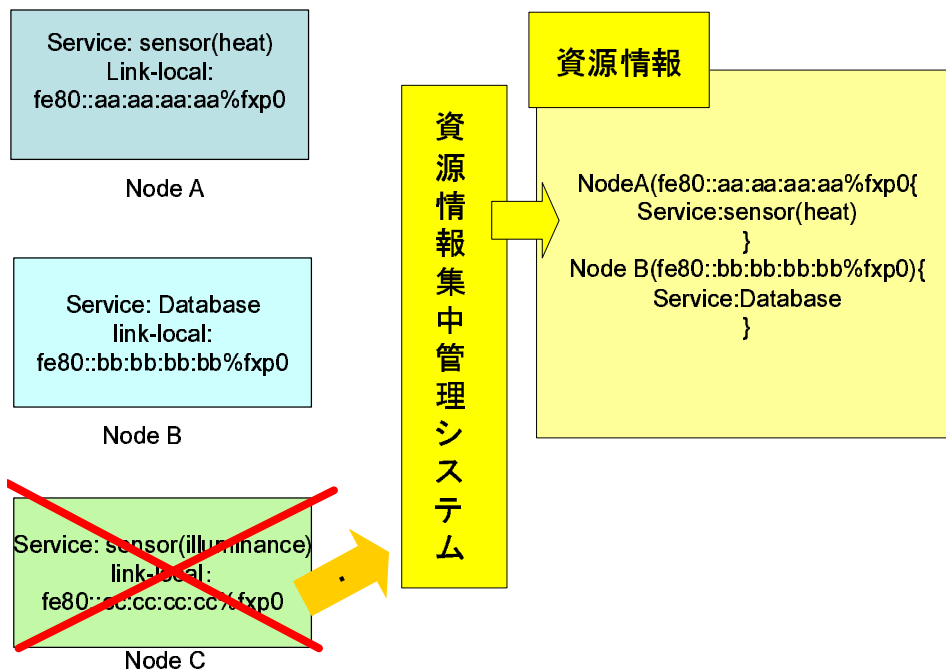


図 3.5: 動的な解除

- サービス資源の静的な登録・配布
 管理者がサービス資源を登録し,DHCP サーバから各ノードに対してサービス資源の登録,配布を行うことは可能である. これは,DNS サーバやゲートウェイ情報の提供に利用されている.
- IPv6 Prefix Delegation
 DHCPv6 では,IPv6 Prefix delegation を利用出来る. これは,上位組織から Prefix を貸し出され,自ホストの IP アドレス貸し出しに利用できるというものである.

また,IPv6 に対しては DHCPv6[9] がすでに RFC3315 で規定されており, IP アドレスの貸し出しに関してはほぼ変わらない機能を持つ.

これより, サービス資源を動的に DHCP サーバに登録・再配布できるようになれば, 前節の条件を満たすことができる. それは DHCP の Request Message, Reply Message を拡張することで対応が可能となる. このため,本研究では DHCP を利用して, サービス資源を含めた情報の割り当てモデルを実装する.

第4章 関連研究

関連研究では,分散型の資源情報設定機構として用いられる Jini,Rendezvous,UPnP を挙げ,それぞれの特徴と,3章で述べた資源情報集中管理システムとの差異を述べる.

4.1 Jini

Jini[10] は,Sun Microsystem 社が中心となって開発している自律的に機器をネットワークに接続するための技術である. Java VM が動いている環境で動作するため,プラットフォームに依存しないことが特徴である.

Jini は,Discovery,Join,Lookup という 3 つの機能を中心に,各デバイスを自律的に動作する. 資源の情報はそのネットワークサービスを管理する Lookup サーバが保持している. サーバは複数台の設置が可能であり初期設定の必要はほぼない. Jini の動作概要を図 4.1を用いて説明する.

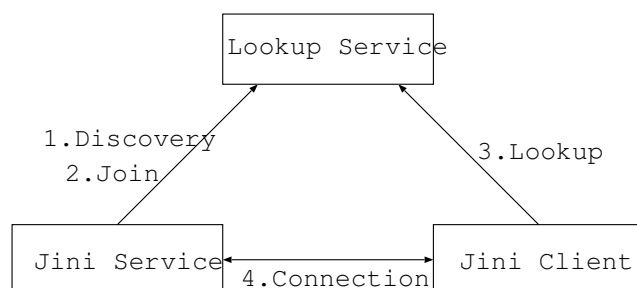


図 4.1: Jini の動作概要

1. Discovery

Jini Service はネットワークに接続されるとそのネットワークで Jini が動作しているかを調査する.

2. Join

Lookup Server からの応答があり,ネットワーク内で Jini が動作していることが判ると Lookup Service に対して自分自身を登録する.

3. Lookup

クライアント (Jini Client) が Jini Service で提供されているサービス資源を使用したい場合クライアントは Lookup Service に対して自分の求めているサービスが登録されているかを照会する. Lookup Service は提供できるサービスのサービスオブジェクトをクライアントに渡す. クライアントは Lookup Service より渡された情報を元に Jini Service と直接やり取りを行う.

Jini は Lookup Server が中央に存在するため, 本研究の考える, 中央管理機構と手法は似ている. しかし, Lookup サーバを管理者が管理する術がないため, ポリシの反映が行えないことが難点である.

また, Java VM を利用するため実装が大きくなってしまう. そのため, マイクロノードなどハードウェアの容量に制限のある機器に載せることが難しい.

4.2 Rendezvous

Rendezvous[11] は, IETF の zeroconf wg[12] が標準化を進めている設定が不要な (Zero Configuration) ネットワークの確立を目指した技術である. Rendezvous は TCP/IP ネットワークを使用し, サーバのないネットワーク環境での IP アドレス割り当て, IP アドレスと名前の変換, サービス検索を行う. Rendezvous が使うプロトコルを表 4.1 に載せる.

表 4.1: Rendezvous で使用されるプロトコル

Addressing(IPv6)	link-local address
Naming	mDNS or Dynamic DNS
イベントング告知	同上
サービスの検索	DNS-SD

このように, Rendezvous は DNS をベースとして, LAN 内での zero configuration を可能としている. しかし, 中央でのサービスの制御や, ポリシの反映など, DNS 上での資源の制限は現状では難しい.

4.3 UPnP

Universal Plug and Play[13] は, Microsoft 社が中心となって開発している, 利用者による設定無しに自律的にネットワークを構築, 協調してサービスを提供する技術仕様である. 家庭など小規模のネットワークを対象とし, UPnP 対応しているデバイスは機器を物理的に接続するだけで家庭内の機器全てをネットワークに接続させ, サービス提供を行うことを目的としている.

UPnP は家庭内の機器全てをネットワークに接続させるための解決方法として, デバイスを中心とした接続性を提供する. そのため, Device Control Protocol(DCP) を機器ごとに定める必要がある.

そのため, 本研究がサービス資源と IP アドレスという資源情報を中心とすると, UPnP はデバイスを中心とした接続方法を提供している. そのため, ポリシに応じたサービス資源の配布は難しい.

また, これらの技術はデバイスの対応が普及への焦点となるが, UPnP, 前述した Rendezvous, Jini は全て業界団体が違い, それぞれにハードウェアベンダーがバックアップしているため, 統一化等の道は模索されていない.

4.4 SLP

Service Location Protocol(SLP)[14] は,IETF svrloc wg で制定されたサービス資源の提供を行う機器を発見するためのプロトコルである. SLP は, サービス資源を提供する側 (Service Agent) と, 利用する側 (User Agent) が直接通信を行う場合と, Directory Agent が仲介する場合がある.

Directory Agent が仲介する場合の動作概要を, 図 4.2に載せる.

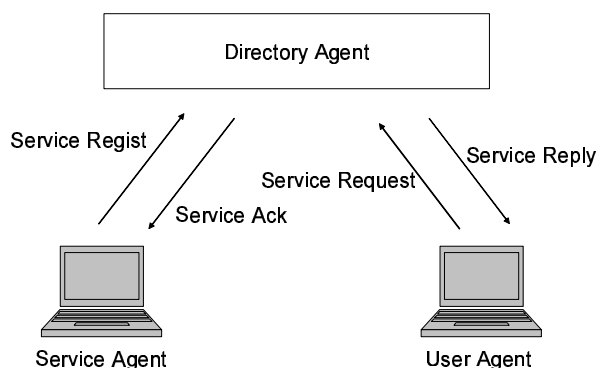


図 4.2: Directory Agent を利用した SLP

Service Agent は Service Register パケットで, 自身の提供できるサービス資源を Directory Agent に通知し, Service Ack パケットで登録が完了される. User Agent は Service Agent に利用したいサービス資源を Service Request パケットを用いて問い合わせ, Service Reply パケットが返答する.

このサービスの問い合わせや返答に用いられるサービス資源情報は図 4.3のフォーマットを用いる.

```
service:<srvtype>://<addrspec>
```

図 4.3: サービス資源情報のフォーマット

例えば, ホスト a.example.com のポート 8080 番で ftp サーバの資源が稼働している場合には図 4.4のようになる.

```
‘service:tftp://a.example.com:8080’
```

図 4.4: SLP フォーマットの例

このフォーマットを利用することにより、アプリケーションベースでのサービス資源情報を配布することができる。しかし、Directory Agent をサービス資源に通知する方法が確立されておらず、通知方法として DHCP など他のプロトコルを利用するか、もしくはマルチキャストを利用する方法が考えられている。また、サービス資源のみを配布するプロトコルであるため、IP アドレスなどの資源情報を把握することが出来ない。

第5章 設計

5.1 概要

本章では、前章で述べたモデルに従って、設計を行う。本機構では、管理者が情報を把握するため、DHCP を用いて情報を管理する。まず、設計に必要な情報と機能について述べ、それらに対する考慮点を述べる。

5.2 設計要件

前節のモデルを踏まえ、以下の機能を設計要件として挙げる。

5.2.1 IP アドレス貸し出し

IP アドレスの貸し出しは、DHCP の 4 つのメッセージ交換によって行われる。図 5.1 に、登録の場合のメッセージ交換を示す。

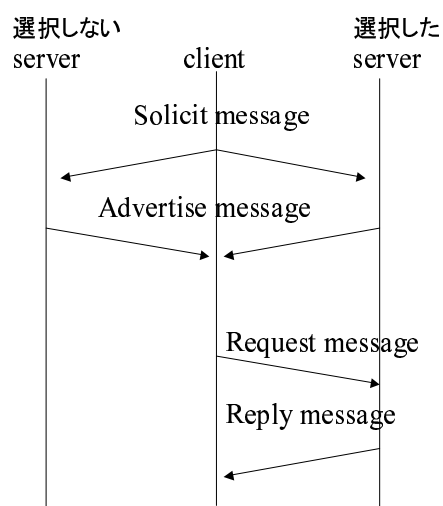


図 5.1: DHCPv6 のメッセージ交換

Solicit Message, Advertise Message によって、DHCP サーバが存在するかどうかを確認される。サーバが複数台存在した場合は、クライアントによって選択されたサーバに Request Message が送られ、サービス資源の貸し出しが許可された場合、Reply Message が送られる。

IP アドレスの貸し出しは IA Address Option が付加される。IA Address Option には、貸し出し IP アドレス、生存時間情報が付加されることが規定されている。

5.2.2 サービス資源情報登録

サービス資源の貸し出しは,Request Message,Reply Message の二つのメッセージ交換によって行う.

現在,DHCPv6 でのサービス資源情報の通知は, サービスごとに option-code を割り当てる形になっている. 例として,DHCPv6 DNS Option??で規定されている DNS サーバのパケットフォーマットを図 5.2に示す.

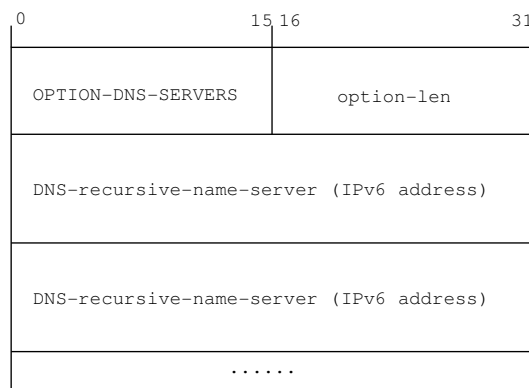


図 5.2: DHCPv6 DNS Configuration Option

この場合,OPTION-DNS-SERVERS にはオプションコード 25 が割り当てられている. しかし,このようにサービスの種類ごとにオプションコードを割り当てる状態を続けると, サービス資源が増加ときのオプションコードの割り当てが追いつかない場合がある.

そのため,本研究では,Service Option を新たに定義し, サービス資源情報を Option に直に記述することによって, 各サービス資源毎のオプションコードの割り当ての必要性を無くす. Service Option パケットフォーマットは図 5.3のように定義する.

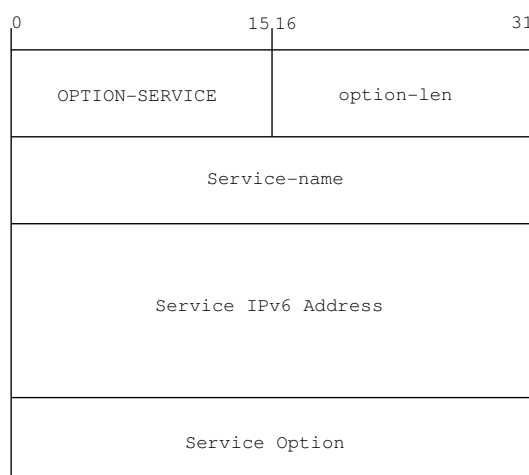


図 5.3: Service Option

サービス資源名は Service Name というフィールドを用意し, そこにサービス資源名を含める.

これにより、サービス資源ごとにオプションコードを定義しなくても独自サービスも登録が可能となる。

図 5.4に、このオプションを利用した場合のサービス登録の方法を例として温度センサを利用して示す。

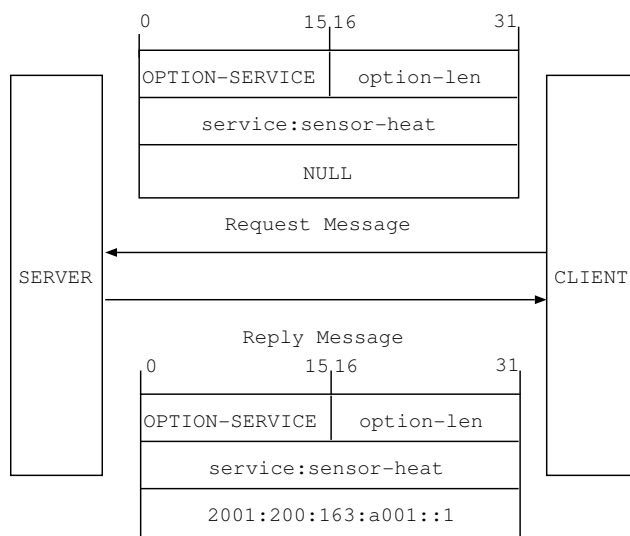


図 5.4: Service Option を利用したサービス登録

クライアントは Request Message に使用できるサービス資源情報を、Service Option の Service Name フィールドに”service:***” の状態で含める。サーバは資源情報を登録し、Reply Message の Service Option に IP アドレスを載せる。

5.2.3 資源情報設定機能

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動的に割り当てる必要がある。

資源をポリシーに応じて動的に割り当てるには、最小単位を設定することが必要である。本研究では、ホスト識別子を最小単位と設定し、これに応じて割り当てる IP アドレス、サービス資源を区別する方法で割り当てに関するポリシーの実現を可能とする。具体的には、サーバ側にポリシーを設定するためのファイルを用意する。それをを用いて、ホスト識別子ごとに割り当てる資源を設定する。ホスト識別子については、次節で述べる。

5.2.4 資源情報探索機能

ノードがサービス資源を利用したい場合、サービス資源が稼働しているノードの IP アドレスを貸し出す。これにより、他のノードに動的なサービス資源配布を行うことができる。これは、Request Message, Reply Message に Service Option を載せることによって可能とする。

図 5.4に、このオプションを利用した場合のサービス探索の例として DNS の検索を示す。

サービスを利用したい側は Service Option の Service Name フィールドに,”search:***” と

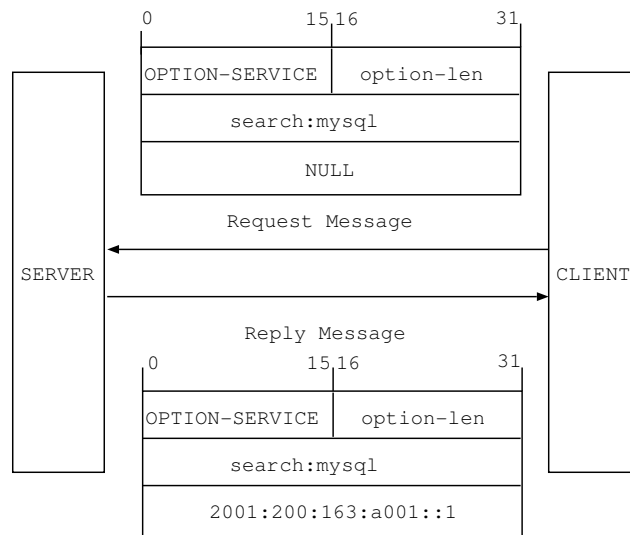


図 5.5: Service Option を使ったサービス探索

いう書式で埋め, Request Message として DHCP サーバに送信する. DHCP サーバは, Service Name から利用可能なサービスを検索し, Service Option の Service IPv6 address フィールドに IP アドレスを設定した Reply Message を送信する. これにより, サービス資源の種類が 256 種類を越える場合であっても, 資源を提供する側とされる側の Service Name フィールドを同一名に設定しておけば, 資源の種類がスムーズに受渡しできる. また, Option フィールドには IP アドレスだけでは設定できない固有情報をいれることでノードのサービス設定を可能とする. また, 必要な情報が OS 依存の場合は, IP アドレスのみを伝え, 他の設定はクライアントとサービス資源とのやりとりに任せるのが良い. もし, 探索されたサービスがない場合は, IPv6 address フィールドを NULL にして Reply メッセージを返す.

5.2.5 資源返却機能

貸し出された資源を返却する場合, 図 5.6 のように, Release Message と Reply Message を利用することでデータベースからの抹消と, IP アドレスの返却を行う.

また, 接続性の障害などで Release Message が通知されないケースを考慮し, 一定間隔で Renew Message と Reply Message の交換を行う. Renew Message の間隔は, IP アドレスの貸し出し時に, サーバ側が通知する. Renew Message が何回か続けて来ない機器はすでにネットワーク上に存在しないものとして, データベースからの抹消と IP アドレスの返却を行う.

このとき, IP アドレスを返却するノードが提供するサービスを利用していた場合の対策が必要となる. もし, 同一ネットワーク上で利用可能な代替サービスが存在する場合には, Service Option に新しいサービス資源の IP アドレスを追加し, Reply Message を送信する. 代替サービスが存在しない場合は, Service Option の IP アドレスフィールドを NULL にして送信する.

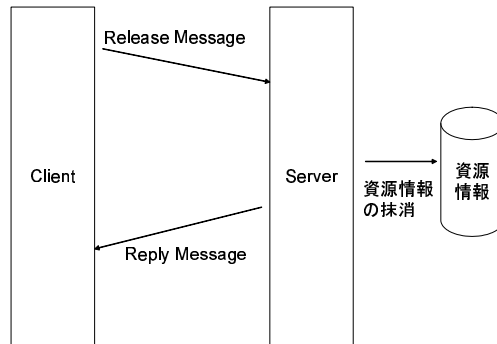


図 5.6: IP アドレスの返却

5.2.6 資源情報参照機能

管理者が、ネットワーク上で利用されている資源情報を把握できると、トラブルシュートを行う時などに効率化が可能である。リアルタイムに資源情報を参照する方法として、ログを出力する。本研究では、IP アドレス貸し出し、サービス資源情報登録、資源情報設定、資源情報探索、資源返却の 5 つの機能のいずれかを利用した場合に、現在の資源情報をリストとしてログを書き出すことによって、資源情報の把握を可能とする。

5.3 考慮する点

5.3.1 ホスト識別子

ホスト識別子は、「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の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ホスト識別子として定義する。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指定しない限り 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。

上記の理由により、リンクローカルアドレスは一意であるといえる。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。これより、リンクローカルアドレスをホスト識別子として利用する。

5.3.2 各アプリケーションへの登録

サービス資源探索のオプションを利用して送信されたサービス資源、また、貸し出された IP アドレスは、各アプリケーション上で適切に設定を行う必要がある。通知された設定情報を利用してアプリケーション毎に設定を行う場合、クライアントは全てのサービス資源に対して情報を保持しなければならなくなり、意味がない。そのため、今回は通知された資源情報を出力できる場所を作り、そこからアプリケーションの適切な場所に通知する方法をとる。これにより、クライアントは資源情報をクライアントの定めた場所に通知するだけでよくなり、各アプリケーションは通知された場所を利用して設定を行う。

5.3.3 サービス資源表記

サービス資源表記はに対して揺らぎがあった場合、サービスを提供しているにも関わらず、データベース上では別の資源として扱われてしまう場合がある。そのため、well-known なサービス資源については、できる限り名前を統一しておくべきである。本研究では、IANA のプロトコル名に準ずることとし、プロトコル名として存在しない場合は、重複しない名前を設定する。

表 5.1 は、サービス資源の名前の一例である。

Printer	プリンターサーバ
DNS	DNS サーバ
NTP	NTP サーバ
SMTP	SMTP サーバ
POP3	POP3 サーバ
WWW	WWW サーバ
LDAP	LDAP サーバ
sensor-light	照度センサ
sensor-heat	温度センサ

表 5.1: サービス資源表記の例

また、独自プロトコルなどは、他のサービス資源と重複しないように、32bit のフィールドの範囲内での独自の名前設定を行うことが可能である。

第6章 実装

本章では,DHCP を利用した資源管理システムの実装について述べる.

6.1 実装概要

本研究では,DHCPv6 のサーバを改良することによって,資源管理を可能とした.

本システムは BSD システム用に IPv6 スタックを実装している KAME Project の DHCPv6 に追加実装する形で実装した. これには,DHCPv6 の機能である Prefix Delegation と DNS の サービス探知が実装されている. そのため,以下の点について実装を行った.

- IP アドレスの割り当て
- サービス資源の割り当て
- 貸し出しリストへの登録
- 貸し出しリストからの解除
- サービス資源の探知
- ポリシ設定
- ログの出力

6.2 実装環境

実装環境は表 6.1 の通りである.

表 6.1: 本システムの実装環境

OS	FreeBSD 4.10-RELEASE kame-20040906-freebsd410 パッチを適用
使用言語	C 言語
コンパイラ	gcc3.2.3

6.3 DHCPv6 サーバ

6.3.1 設定ファイルとポリシの設定

DHCPv6 サーバは起動時に,dhcp6s.conf という設定ファイルを読み込む. 設定ファイルの例を図 6.1に示す.


```

interface fxp0{
    range from 2001:200:163:b001::1 to 2001:200:163:b001::ff;
    prefix /56;
    vlttime 200;
    pltime 300;
    gateway 3ffe:501:1085:8200::100;
};

host hoge{
    linklocal fe80::206:29ff:fe8f:b034%fxp0
    fixed-addr 3ffe:501:1000:100::1;
};

```

図 6.1: DHCPv6 サーバ側設定ファイルの例

interface セクションには、そのネットワーク上で貸し出す IP アドレスを設定する。host セクションでは、個々の機器についての IP アドレスを設定する。そのため、ポリシーの反映は host セクションで行う。

6.3.2 IPv6 アドレス割り当て

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ンクドリフトにして address_list に格納する。host セクションの情報は host_list に格納する。address_list, host_list のデータ構造を図 6.2 に示す。

pltime には、prefer lifetime(希望生存時間) が、vlttime には valid lifetime(有効生存時間) が入り、ノードの生存情報に利用される。prefix, gateway の情報は、接続性の確保に必要なため、サービス資源とは別に保持をする。図 6.3 に、Request メッセージ受信後、Reply メッセージを送信するまでの IPv6 アドレス割り当ての一連の流れを示す。

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、check_conf 関数で 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、lease_hostinfo 関数にて、host_list の IP アドレスをオプションフィールドに追加する。適合しない場合は、address_list から一つを抜き出してリストから削除すると共に 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。その後、IA オプションでアドレスやサービス資源を送信する場合に、write_leaselist 関数にて lease_list に追加を行う。set_option 関数では、IA address Option に貸し出しの IP アドレスを割り当てる。IA Address Option のパケットフォーマットを図 6.4 に示す。

このオプションでは、IPv6 アドレスの貸し出しは行えるが、デフォルトゲートウェイや IP アドレスのプレフィックスなど通信に必要な情報が送信できない。そこで、そのような情報は IAaddr-options フィールドに保存することで解決する。IPv6 アドレスの貸し出しは、Request Message に、フィールドが空の IA Address Option を付加して送信することにより貸し出しを希望する

```

struct dhcp6_addrinfo{
    struct in6_addr *in6;
    struct sockaddr *from;
    int pltime;
    int vltime;
    int prefix;
    struct in6_addr *gateway;
    struct service list *use_service;
    struct service list *give_service;
    struct dhcp6_addrinfo *next;
    struct dhcp6_addrinfo *prev;
};

```

図 6.2: address_list データ構造

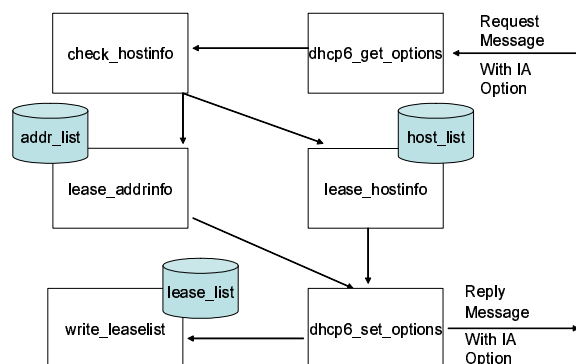


図 6.3: Request メッセージ受信後のアドレス割り当て動作

とみなす。

6.3.3 サービス資源登録

サービス情報は、IP アドレス貸し出し要請である IA Address Option ではなく、Service Option によって送られてくる。IA Address Option と同じパケットで送られてくるとは限らず、Request Message に含まれるため、アドレス貸し出しの時に一緒に登録するのではなく、別のルーチンとして分けるべきである。

そのため、Service Option がクライアントから送られてくると、一度 service_list 構造体に保存する。service_list 構造体を 6.5 に示す。

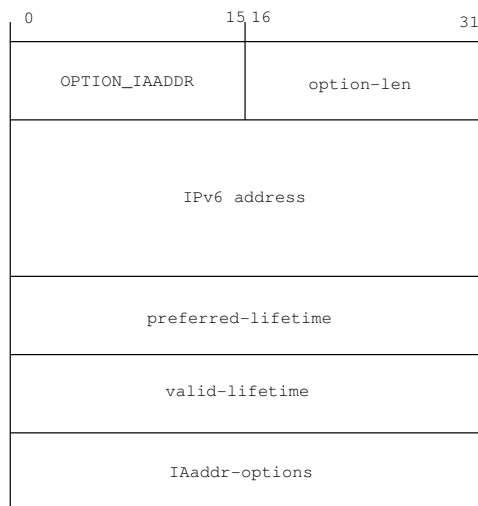


図 6.4: IA Address Option

```

struct dhcp6_service{
    struct sockaddr *from;
    char *servicename;
};

```

図 6.5: service_list データ構造

service_list を一度保存した後, lease_list のホスト識別子をキーとして検索を行う。同一の識別子が見付かった場合, 当該部分の take_service 構造体にコピーをし, write_leaselist 関数を呼び出す。

6.3.4 サービス資源探索

サービス資源探索の Service Option が送信された場合, 探索したいサービス内容を service_list に保存し, servicename をキーとして lease_list と照らし合わせる。照らし合わせた結果, 合致した場合は, その lease_list の情報を Service Option にいれて送信する。また, ホスト識別子をキーとして lease_list を検索し, 一致した場合, use_service に service_list をコピーし, service_list の中身を消去する。

検索方法は, リンクドリストの先頭から順に検索する方法をとった。

6.3.5 貸し出しリストからの解除

クライアントは prefer lifetime の時間毎に Renew Message をサーバに送信し, それに対し, サーバはクライアントに Reply Message を交換することで, ノードの生存情報を確認する。そのため, valid lifetime を超過しても Renew Message がサーバに送信されない場合, もしくは, ク

クライアントから Release Message が送信された場合には、ネットワーク上から離れたと解釈し、貸し出しリストからの削除を行う。

この場合、lease_list から該当部分を抜き出して、リンクドリストからの削除を行う。IP アドレスは、address_list の最後に追加し、貸し出しプールとする。

6.3.6 ログへの書き込み

write_leaselist 関数が呼び出される毎に、ログが出力される。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。

ログファイルは、図 6.6の形式で出力される。

```
host fe80::206:29ff:fe8f:b034%fxp0{
lease 2001:200:163:b001::1
provide-service phone
}

host fe80::20a:e6ff:fe93:69b0%r10{
lease 2001:200:163:b001::1
provide-service heat-sensor
require-service database
}
```

図 6.6: ログ出力結果

6.4 DHCPv6 クライアント

6.5 クライアント側設定ファイル

クライアント側は、図 6.7のように設定ファイルを記述する。

“request v6address” では、IPv6 アドレスの割り当てを希望する。“script “/sbin/dhcp6-script”” では、資源情報を受信した場合に利用するスクリプトファイル名を記述する。これについては後述する。また、提供できるサービス資源、利用を希望するサービス資源も設定ファイルに記述する。

6.5.1 サービス資源情報の登録

ノードが提供できるサービス資源をサーバに登録する場合、設定ファイルに “service *****” のように記述をする。

DHCPv6 クライアントは Service Option の servicename フィールドに”service:サービス資

```
interface fxp0{
    request v6address;
    script "/sbin/dhcp6c-script";
    service sensor-light;
    request service database;
};
```

図 6.7: DHCPv6 クライアント側設定ファイルの例

源名”のように記述し Service Option を付加した Request Message を DHCPv6 サーバに送信する。正常にサービス資源情報が DHCPv6 サーバに登録された場合、クライアントに Reply Message が返信される。

6.5.2 サービス資源の探索

サービス資源の探索は、設定ファイルに “request service ****” と記述することによって行われる。

DHCPv6 クライアントは Service Option の servicename フィールドに “search:サービス資源名” のように記述し、Service Option を負荷した Request Message を送信する。サービスが発見された場合は Reply Message に付加された Service Option に利用できるサービス資源の IP アドレスが通知される。これを元にアプリケーションへの追加を行う。

6.5.3 資源のアプリケーションへの追加

資源情報は最終的に、設定ファイルで記述したスクリプトファイルに渡される。各アプリケーションは、スクリプトファイルにアプリケーションへの追加方法を書くことによって、資源情報の引渡しを行う。図 6.8に、IPv6 アドレスとゲートウェイ情報をアプリケーションに引渡すシェルスクリプトの例を示す。

```
#!/bin/sh
if [$address]; then
echo /sbin/ifconfig $ifname inet6 $address prefixlen $prefixlen
fi
if [$gateway]; then
/sbin/route add default inet6 $gateway
fi
```

図 6.8: DHCPv6 クライアントスクリプトの例

第7章 評価

本章では, 本システムの評価について述べる.

7.1 モデルとの合致

本システムの評価として, 関連研究で挙げたシステムとの比較を行う. 評価項目は, 4.1 章で挙げた要件に基づき, 以下のように設定する.

- 自動設定が可能であること
- TCP/IP ベースであること
- ユーザのポリシーを反映できること

7.1.1 自動設定

Rendezvous, UPnP などの IPv4 アドレス取得は DHCP を利用するため, アプリケーションを動かすという面では DHCP をベースとして用いる本システムはひけをとらない. しかし, DHCP が存在しなかった場合, Rendezvous, UPnP は AutoIP を利用して自動設定を行うが, 本システムは DHCP サーバが存在しなければ自動設定ができない. Jini と比較した場合, Jini サーバが存在するだけで設定が可能なことから自動設定が可能である.

7.1.2 ポリシの反映

分散型の自動設定機構では, 機構を利用してポリシーを反映することは難しく, 手動で行うしか方法がない. それに対して本システムでは, ポリシーを設定ファイルに記述することで, 最小単位でのポリシーを管理者が設定することが出来る.

表 7.1 に, 上記であげた評価を表として示す.

	UPnP	Rendezvous	Jini	SLP	DHCP
自動設定					
ポリシーの反映	×	×	×	×	

表 7.1: モデルとの合致

7.2 ブートストラップまでの性能評価

サービス資源獲得までのパケットメッセージ数を比較することで、ノードがサービスを獲得するまでのオーバーヘッドを知ることが出来る。

表 7.2に,UPnP,Rendezvous,Jini,DHCP,SLP+ICMP の RA メッセージのそれぞれを用いた場合の IP アドレス取得からサービス資源獲得までの最低総パケット交換数を示す。

UPnP	5
Rendezvous	
Jini	6
SLP+RA	6
DHCP	4

表 7.2: 総パケット交換数

これより,DHCP を利用した場合のパケット交換数が一番少ないことが分かる。また,DHCP の場合,Solicit Message 以外は全てサーバとクライアントのやりとりで済むため,交換に対してのオーバーヘッドは少ない。

第8章 結論

8.1 今後の課題

8.1.1 ポリシの表現

今回は、ポリシによる資源情報の設定方法を管理者側が定めたノードごとに行った。しかし、資源情報をサービス資源ごとに設定を行うことや、ある一定の規則に従ってグループを設定することも可能である。また、クライアントからの申告によって、資源を割り当てていくことも可能である。ポリシの表現方法について、熟考が必要である。

8.1.2 資源情報

今回、資源情報はこれが最低限必要なものであると判断し、IP アドレス、サービス資源、ホスト識別子のみを扱った。しかし、資源情報はその他にも使用している OS, NetBIOS で利用されるホスト名等、多くのものを取得することが出来る。取得することで、サービス資源振り分けの可能性が広がることもあり、今後、どのような資源情報が必要なかを考慮しなければならない。

8.1.3 セキュリティ

本システムでは、セキュリティ上の機能を考慮していない。しかし、実際には DHCP サーバのなりすましなどの危険性が考えられる。クライアントとサーバの認証方法等については今後の課題とする。

また、本研究で述べたモデルに基づき、実運用を行い、スケーラビリティ、機器情報を実際に計測することが一番の課題である。

8.2 まとめ

本研究では、今後家庭内 LAN に接続される機器が飛躍的に増加することに着目し、機器に対しての資源設定方法の現状をまとめた。その上で、現状ではサービス資源を管理者が把握し、設定を手動で行われているが、管理者はほとんどの場合、利用している家庭内のユーザが兼ね備える場合が多いため、負担が増加することを問題点として述べた。また、資源の振り分けを、管理者にとって簡易に反映することが必要と述べた。

その上で、家庭内 LAN における資源割り当てモデルとしての要件を述べ、集中型の管理モデルを用いて資源情報を把握し、クライアントからの動的な登録、解除を受け付けるモデルを提案した。DHCP を拡張することで、提案したモデルを設計・実装を行った。

謝辞

本論文の執筆にあたって御指導頂きました慶應義塾大学環境情報学部教授 村井純博士, 並びに同学部教授 徳田英幸博士, 同学部助教授 中村修博士, 同学部助教授 楠本博之博士同学部専任講師 南政樹氏に感謝致します.

また, 絶えず御指導, 御助言を頂いた慶應義塾大学政策・メディア研究科博士課程小原泰弘氏, 今泉英明氏, 同修士課程 白畑真氏, 橋本和樹氏, 佐川昭宏氏に感謝します.

本論文執筆にあたり, 資料収集など多くの面で手伝っていただいた慶應義塾大学環境情報学部 金井瑛氏, 空閑洋平氏, 遠峰隆史氏に感謝します.

徳田・村井・楠本・中村・南合同研究会 の皆様, 特に本論文の執筆にあたり御協力くださった SING,IA*研究グループの皆様感謝致します.

自分のやりたいことを貫き通した私を最後まで見守って下さった父と母, 兄弟, 祖父母に感謝します.

参考文献

- [1] 総務省. 平成 16 年度版 情報通信白書, July 2004.
- [2] Ubiquitous Computing Laboratory . <http://www.ubi-lab.org/>.
- [3] Ubiquitous Open Platform Forum . <http://www.uopf.org/>.
- [4] P. Srisuresh, K. Egevang. *RFC3022 Traditional IP Network Address Translator (Traditional NAT)* , January 2001.
- [5] R. Droms. *RFC 2131 Dynamic Host Configuration Protocol*, March 1997.
- [6] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*, December 1998. RFC 2460.
- [7] S. Miyakawa, R. Droms. *RFC3769 Requirements for IPv6 Prefix Delegation*, June 2004.
- [8] T. Narten, E. Nordmark, W. Simpson. *RFC2461 Neighbor Discovery for IP Version 6 (IPv6)*, December 1998.
- [9] R.Droms J.Bound B.Volz T.Lemon C.Perkins and M.Carney. *Dynamic Host Configuration Protocol for IPv6 (IPv6)*, 2003. RFC 3315.
- [10] Sun microsoft. Jini Technology Architectural Overview. <http://www.sun.com/software/jini/whitepapers/architecture.html>, January 1999.
- [11] Rendezvous. <http://developer.apple.com/macosx/rendezvous/index.html>.
- [12] A. Williams. Internet-Draft Zeroconf Requirements. <http://files.zeroconf.org/draft-ietf-zeroconf-reqts-12.txt>, pages 1–22, 2002.
- [13] UPnP Forum. <http://www.upnp.org/>.
- [14] E. Guttman, C. Perkins, J. Veizades, M. Day. *RFC2608 Service Location Protocol, Version 2*, June 1999.
- [15] e-care town project. <http://www.e-care-project.jp/>.
- [16] P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound. *RFC2136 Dynamic Updates in the Domain Name System (DNS UPDATE)*, April 1997.

[17] R. Droms. *RFC3646 DNS Configuration Options for DHCPv6*, December 2003.