

仮想ローカルノード化手法による スマート環境間接続モデルの実装と評価

島谷 宙伸^{†1} 新井 イスマイル^{†2} 西尾 信彦^{†1}

本論文では, スマート環境間の連携を可能とするために, PIM(Pooling nodes for Inter-smart-space collaboration with Minimal impact) と呼ばれるスマート環境接続モデルの導入を行う. 本モデルによれば, ユーザは, 仮想的なアドレス空間にノードを共有することができる. また, 共有されたノードは, 遠隔のユーザからあたかもリンクローカルに存在するようにアクセスされる. 我々の PIM の実装は, PeerPool と呼ばれる我々のネットワーキング技術に基づく. PeerPool は, 互いに遠隔なプライベートネットワークに在する任意のノード群の間に IP 層の接続性を与える. しかし, 機能の制約から UPnP と Bonjour を含むいくつかのアプリケーションが使えない問題があった. そこで, ブロードキャストおよびマルチキャストへの対応, プライベートドメイン名の解決, およびペイロード変換の 3 機能を追加することにより, 完全な機能を持つ PIM の実装を実現した.

Realizing Distributed Smart Spaces with Virtual Localization Method

MICHINOBU SHIMATANI,^{†1} ISMAIL ARAI ^{†2}
and NOBUHIKO NISHIO^{†1}

This paper introduces a communication model called PIM (Pooling nodes for Inter-smart-space collaboration with Minimal impact) allowing users to share nodes across distributed smart spaces. According to this model, a user puts a node into a shared virtual address realm, so that trusted remote users can get access to it as if it were link-local to them. Our implementation of PIM is based upon PeerPool, a networking technology that we have been developing for the last a few years. PeerPool provides connectivity for an arbitrary set of nodes in separate private networks. This technology has functional limitations, leaving some popular applications including UPnP and Bonjour unsupported. Thus we have added three new features to PeerPool, namely, broadcast and multicast support, private domain name resolution and payload translation, so that it makes a full-featured implementation of the model.

1. はじめに

近年ユビキタスコンピューティングの研究の進展により, スマート環境と呼ばれる IP ネットワーク環境が家庭やオフィスに普及することが期待される.^{1)–3)} スマート環境とは空間内に遍在したセンサやデバイスがネットワークを介して連携し, ユーザに有益なサービスを提供するような環境を指す. 一般に, スマート環境は, 高いセキュリティ性や潤沢な IP アドレスの使用などを要求するため, プライベートネットワーク上に構築され, NAT(Network Address Translation) やファイアウォールによって外部からの通信を制限されているものと考えられる.

一方, それぞれのユーザが互いに遠隔であるスマート環境同士では, それらの間の通信を伴うユーザの協調作業を支援できるサービスの需要が考えられる. 例えば, 共同研究している研究室空間のビデオ会議を支援するアプリケーションなどが想定される. こうしたサービスの実現のため, 何らかの技術を用いて, 遠隔なプライベートネットワーク間の接続を確保する必要がある. またこうした何らかの接続技術を用いて互いに接続しあうスマート環境の集合のことを本稿では分散スマート環境と呼ぶ.

遠隔のプライベートネットワーク間を接続するために, 今日までに様々な技術が研究・開発されてきた. 一般的な VPN は, ネットワーク構成の変更や個々の末端ノードへのソフトウェア導入を要するなど, 導入の敷居が高い. 一方, RNSplicer⁴⁾ や Shepherd⁵⁾ などの技術は, 利用できるアプリケーションが限定されるなど, 機能面に問題がある. 導入の敷居を低く抑えつつ高い機能性を実現した技術は, 過去になかった.

そこで我々は, ネットワーク的に異なる環境間を, それらを構成する機器に影響を与えることなく接続するための技術として, PeerPool などを研究してきた⁶⁾.

本研究では, PeerPool を実装の一部とするスマート環境間接続モデル PIM(Pooling nodes for Inter-smart-space collaboration with Minimal impact) を導入する. 本モデルでは, Gateway なるノードを各スマート環境に 1 台ずつ導入し, それらをオーバーレイネットワークで接続し, このオーバーレイネットワーク上で 1 つのアドレス空間を共有する. また, ユーザは任意のノードをこのアドレス空間に共有することができる. 共有されたノードは遠隔の

^{†1} 立命館大学情報理工学部

^{†2} 立命館大学総合理工学研究機構

ユーザからの接続要求に対し、接続要求をしたノードのアドレス空間に仮想的に生成され、あたかもリンクローカルに存在するようにアクセスされる。さらに、PeerPool での課題であるブロードキャスト通信とマルチキャスト通信の中継、プライベートドメイン名の別名解決さらに転送パケットのペイロード変換の 3 つの機能を追加実装を行うことにより、分散スマート環境においてネットワークの構成や各々のノードのソフトウェアへの変更を抑えつつ、Web アプリケーションや UPnP⁷⁾ および Bonjour⁸⁾ を含む広範のアプリケーションを利用できるようになる。

2. スマート環境間接続モデル PIM における課題

本章ではまず、本研究が想定する想定環境を述べ、それにより本研究が満たすべき要件を明らかにする。また、我々が研究を行っているスマート環境間接続モデル PIM について説明し、要件を満たすための残された課題を明らかにし、それについて解決案を提案する。

2.1 想定環境

本研究は、互いに一定の信頼を置いたユーザらが、それぞれの家庭やオフィスにスマート環境を構築していることを想定する。我々が想定するスマート環境は、以下に示す特徴を持つ。

- プライベートネットワーク上に構築されるプライベートネットワークは、潤沢な IP アドレスが使用できることや、外部ネットワークからの接続を遮断することによるセキュリティの高さを理由に、家庭やオフィスのネットワーク環境として広く用いられている。スマート環境も同様の理由から一般的にプライベートネットワーク上に構築されるものと考えられる。
- UPnP や Bonjour などのアーキテクチャに基づいてサービスを提供するノードを含む UPnP と Bonjour は、iTunes⁷⁾ といったホームネットワークの代表的な基盤プロトコルである。多くのスマート環境にもこれらのプロトコルに基づいたサービスを提供するノードが存在すると考えられる。
- ソフトウェアの変更が不可能または困難なノードを含むスマート環境には多数の組み込み機器が存在する。組み込み機器は一般的に PC などのようにユーザの手でソフトウェアを書き換えることは困難である。

- 外部ネットワークとの接続性を利用するノードと利用しないノードが混在する遠隔のスマート環境下のノードと直接通信する必要があるのは、ユーザ端末やサーバなどの一部のノードだけである。

2.2 本研究が必要とする要件

本研究の目的は、2.1 節で述べた想定環境に基づく分散スマート環境間の接続を可能とするシステムの実現である。想定環境を考慮した上で、本研究が必要とする要件を以下に述べる。

- (1) IP 上の任意のプロトコルを利用可能にする。
- (2) 通信ノードに特殊な技術を実装しない。
- (3) 接続性が必要なノードにのみ接続性を提供する。
- (4) 接続性が必要ないノードへ影響を与えない。
- (5) 適時かつ容易に接続性を変更可能とする。

2.3 分散スマート環境に関する研究の現状

2.3.1 伝統的な通信技術

遠隔のプライベートネットワーク間で NAT やファイアウォールを超えた通信を実現するために、ポートフォワーディングや VPN(Virtual Private Network) が一般的に利用されてきた。しかし、ポートフォワーディングでは、固定ポート番号を使用するアプリケーションの同時利用などの場合、問題が生じることがある。特に組み込み機器においては、プログラムにハードコードされるなどしてポート番号の変更が不可能である可能性が懸念される。また VPN ソフトウェアは、原則としてネットワーク内の全ノードに一律に接続を与えるため、接続性を利用しないノードが混在する環境には不満である。

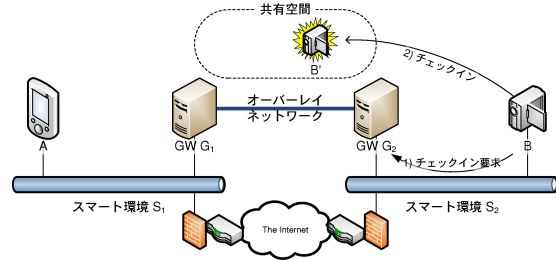
2.3.2 RNSplicer

RNSplicer は、遠隔のスマート環境間で Bonjour ベースのサービスを相互に利用することを可能にする。さらに、実装次第で他のサービスアーキテクチャにも適用可能なモデルである。ただしサービス広告の機構を持たないアプリケーションには適用できない。

2.3.3 Shepherd

Shepherd は、各プライベートネットワークに 1 台ずつ設置されたシェパードノードと呼ばれるノードが Proxy ARP を用いてパケットを集約し、他のプライベートネットワークのシェパードノードに転送する。ユーザがシェパードノード上の GUI を操作してノードを

(a) チェックイン



(b) 仮想ローカルノード化

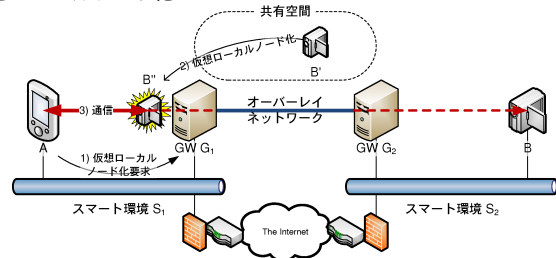


図 1 PIM: Pooling nodes for Inter-smart-space collaboration with Minimal impact

登録することにより、プライベートネットワーク内の一部のノード群に選択的に接続性を与えることができる。

2.4 スマート環境間接続モデル PIM

本節はスマート環境間接続モデル PIM(Pooling nodes for Inter-smart-space collaboration with Minimal impact) の導入を行う。

本モデルでは、GW(Gateway) を各スマート環境内に 1 台ずつ導入し、それらをオーバーレイネットワークで接続する。互いに接続された GW 群は、オーバーレイネットワーク上で 1 つのアドレス空間を共有する。この共有されたアドレス空間を共有空間と呼ぶ。スマート環境内の任意のノードは、同一のスマート環境内に配置された GW に対して以下に定義する操作を要求することができる。

● チェックイン

チェックインは、共有空間に実ノードの写像を生成する操作である(図 1.(a))。本操作

により共有空間に生成された写像を共有ノードと呼ぶ。共有ノードは、接続されたスマート環境群にのみ共有され、グローバルネットワーク上の第三者からは不可視である。また、共有ノードには共有空間内において、それを一意に同定するために固有の識別子が付与される。

● チェックアウト

チェックアウトは、共有ノードを削除する操作である。本操作要求を出したノードの写像が共有空間に存在する場合はそれを削除する。それが存在しない場合は無効である。

● 仮想ローカルノード化

仮想ローカルノード化は、与えられた識別子を持つ共有ノードの写像を自スマート環境のアドレス空間上に生成する(図 1.(b))。本操作により生成された写像を仮想ローカルノードと呼ぶ。本操作要求を発行したノードは、本操作によって生成された仮想ローカルノードに対してリンクローカルな通信が可能になる。仮想ローカルノードは、それが参照する共有ノードがチェックアウトされると消滅する。

すなわち、各スマート環境内のユーザは、他のスマート環境からの接続を許可するノードをチェックイン・チェックアウト要求により、任意に選択することができ、また仮想ローカルノード化要求を行うことで、他のスマート環境内のノードとリンクローカルに通信を行うことができる。

本モデルによれば、チェックイン要求および仮想ローカルノード化要求を行ったノード間のみ接続性が与えられる。したがって、本モデルは要件 3 および要件 4 を満たす。また操作要求を行うだけで、任意のノードの接続性を変更可能であることから、要件 5 を満たすと言える。要件 1 と要件 2 については、実装依存である。我々が従来行ってきた実装では、要件 2 は満たすが、要件 1 は十分に満たされていない。本研究では、これを満足することを目指す。

2.5 スマート環境間接続技術 PeerPool

本節では、我々が研究及び開発をしてきたスマート環境間接続技術である PeerPool⁽⁶⁾ について説明を行う。また、PeerPool は、PIM の一実装と考えることができる。PeerPool における GW は、特に PoolGW と呼ばれる。

2.5.1 DNS クエリによる操作要求

PeerPool におけるノードのチェックインや仮想ローカルノード化要求などの操作要求は、

DNS クエリにより実装されている。これはネットワークソフトウェアでは一般的に DNS をサポートされているため、ノードに特別なソフトウェアをインストールすることなく要求を発行できる。ただし、通信ノードからの DNS クエリは PoolGW へ送信されるよう、各ノードまたはルータの設定を変更しなくてはならない。また、DNS クエリを送信するためのインタフェースを備えていないノードについては、登録されたノードから代理要求を発行することにより、操作要求を発行することができる。

操作要求の方法だが、チェックイン・チェックアウトは、それぞれ当該ノードにおいて、ドメイン名 *node-name.in.pool,node-name.out.pool* に対するアドレス解決を PoolGW に対して要求する。接頭辞 *node-name* は、当該要求によって共有空間に生成されるノードに与えられる名前（識別子）であり、既存の名前と衝突しない限り定められた文字セットの中で自由に定義することができる。

また、共有ノードを仮想ローカルノード化するには、PoolGW に対しドメイン名、(仮想ローカルノード化したい共有ノード名).local.pool に対するアドレス解決要求を発行する。

2.5.2 Proxy ARP による仮想ローカルノード

PeerPool では仮想ローカルノードは、PoolGW が LAN 内の未使用アドレスを利用して構成する Proxy ARP により実装されている。仮想ローカルノード宛の packets は PoolGW によりアドレス変換され、オーバーレイネットワークの通信路を通り、共有ノードの実体へと転送される。

2.5.3 未解決の問題

PeerPool は、IP 層で接続性を提供するため、原則として IP 上の任意のプロトコルに対応する。ただし、以下に述べる制約がある。

まず PeerPool はブロードキャスト通信およびマルチキャスト通信の転送を行う設計はなされていない。一般にブロードキャスト通信およびリンクローカルスコープのマルチキャスト通信は、ルータを超えないが、通信ノードから見て仮想ローカルノードは同一ネットワーク上に存在するため、これらの通信が行えることが望ましい。

また、Bonjour のサービス広告など、ペイロードにプライベートドメイン名または IP アドレスを含むプロトコルは、利用できない場合がある。前者については、送信者と受信者の在るスマート環境の名前空間が異なるため、両者がドメイン名に対する認識に齟齬が生じることが原因である。後者では、PoolGW がパケットを転送する際に IP ヘッダに書かれたアドレスを書き換えるため、送信者と受信者で IP アドレスに対する認識に齟齬が生じるこ

とが原因である。これらの制約により、UPnP や Bonjour などのサービス発見にブロードキャスト通信を用いるプロトコルを使ったサービスで、問題が発生する。また、Web サービスにおいても、データ中にプライベートドメイン名および IP アドレスを含む場合、問題が生じる。

2.6 PeerPool への追加機能の提案

本研究では 2.5.3 節で述べた未解決の問題を解決するため、提案システムに以下の 3 つの機能を追加実装する。

- ブロードキャスト通信とマルチキャスト通信を中継する機能
2 つのスマート環境間 S_1, S_2 において、 S_1 内にあるノード A がこれらのパケットを送信しているとすると、ノード A が S_2 内にある共有ノード B と仮想的にリンクローカルな場合、 S_1 内の PoolGW はノード A が送信しているパケットを実ノード B へと中継する。
- プライベートドメイン名を仮想ローカルノードの別名に解決する機能
2 つのスマート環境間 S_1, S_2 において、 S_1 内のノード A が S_2 内のノード B のプライベートドメイン名の名前解決要求を行った時、 S_1 内の PoolGW は、それがノード B における仮想ローカルノードの別名であるかのように返答する。
- スマート環境間を転送されるパケットのペイロードを変換する機能
ペイロードに IP アドレスを含む場合、PoolGW がパケットを転送する際に、ペイロードに埋め込まれた IP アドレスを、送信者と受信者として IP アドレスに対する認識に齟齬が生じないように変換する。

これらの機能により、分散スマート環境間において IP 上の任意のプロトコルを使用することができると考えられる。

3. 設 計

3.1 ブロードキャスト通信とマルチキャスト通信の中継

本節では、PeerPool を利用する分散スマート環境間でブロードキャスト通信とマルチキャスト通信を実現する手法を説明する (図 2)。なお、簡単のために、以降はブロードキャストとリンクローカルスコープのマルチキャストを統合して B/Mcast と呼ぶ。

本研究の想定環境では、分散スマート環境間での接続性を得ているノードとそうでない

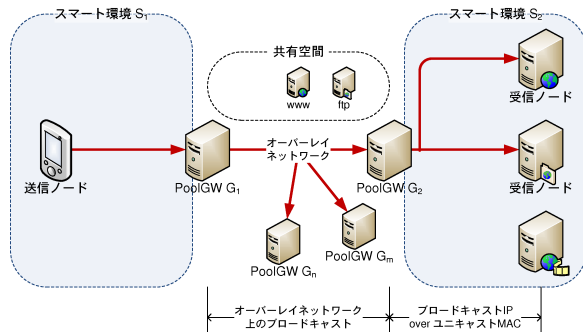


図2 ブロードキャスト通信とマルチキャスト通信の中継

ノードが存在する．よって，単に PoolGW が他の PoolGW に対し，B/Mcast パケットを転送すると，接続性を得ていないノードにまでこれらのパケットが到達することになる．これはユーザが意図する結果ではない．したがって，パケットが到達する範囲を適切に制限する必要がある．

パケットの到達範囲だが，B/Mcast パケットの送信者と物理的および仮想的にリンクローカルであるノードとするのが自然である．任意のノード A,B に対し，A と B が仮想的にリンクローカルなのは，以下の条件の少なくとも 1 つを満たす時である．

- A が共有空間にチェックインされており，B が A を仮想ローカルノード化している．
- B が共有空間にチェックインされており，A が B を仮想ローカルノード化している．

LAN 内の任意のノードへの B/Mcast パケットの配送は，データリンク層レベルのユニキャストを受信者の数だけ繰り返すことによる．

3.2 プライベートドメイン名の別名解決

ペイロードにドメイン名を含む場合でも，それがグローバルに解決可能なドメイン名ならば問題にならない．名前空間の違いのためにドメイン名に対する齟齬が生じる場合は，それがプライベートドメイン名の場合である．本提案では，通信ノードが発行する DNS クエリを PoolGW がフックできることを利用してこの齟齬を解消する．つまり，プライベートドメイン名に関する DNS クエリを PoolGW が受信した場合，PoolGW はそれを仮想ローカルノード名の別名（エイリアス）であるかのように返答を行う．

具体的な例を図 3 に示す．図 3 では，スマート環境 S_2 上には，Web サーバ W と FTP

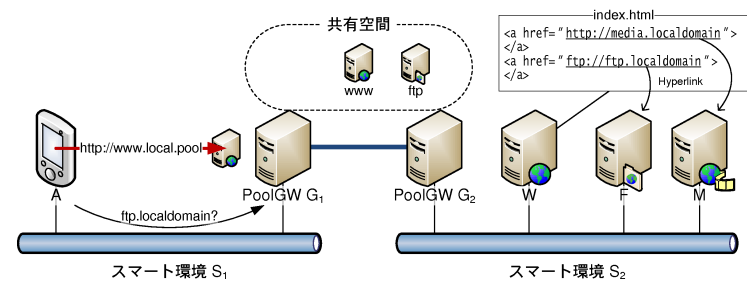


図3 プライベートドメイン名の別名解決が必要な場合の一例

サーバ F およびメディアサーバ M が存在する．これらのサーバは，全てグローバルアドレスを持たず，付与されたドメイン名 (www,ftp,media).localdomain はローカルネットワーク内でのみ名前解決される．また，共有空間上には Web サーバ W (図中 www) と FTP サーバ F (図中 ftp) がチェックインしている．

今スマート環境 S_1 内にあるノード A が共有ノード W に対し，仮想ローカルノード化要求を発行し，通信を行った．W から送られてきたパケットには，F のスマート環境 S_2 におけるプライベートドメイン名 ftp.localdomain が記述されていた．一般的に A がこのプライベートドメイン名に対しアクセスしても，スマート環境 S_1 の DNS サーバが名前解決できないため通信を行うことはできない．

そこで，A からの DNS クエリを PoolGW G_1 でフックできることを利用し，ftp.localdomain が ftp.local.pool の別名であるかのように返答する．これにより，ftp.localdomain の名前解決を行うと同時に F の仮想ローカルノードが生成され，透過的な通信が実現される．

3.3 転送パケットのペイロード変換

ペイロードに IP アドレスを含むプロトコルに関しては，プロトコルに専用のペイロード変換モジュールを実装し，PoolGW においてアプリケーションレベルゲートウェイを構成することによって対応する．これは，ドメイン名の場合と異なり，パケット転送時以外にフックポイントがないためである．

4. 実装

4.1 オーバレイネットワークの実装

PoolGW 間のオーバレイネットワークは、OpenVPN による L2VPN で構築した。OpenVPN は、1 台の VPN サーバを中心に 1 台以上の VPN クライアントが接続するスター型のトポロジを形成する。本実装では、グローバルネットワークに VPN サーバを 1 台設置し、3 拠点のプライベートネットワークにそれぞれ VPN クライアント (PoolGW) を 1 台ずつ設置した。各 VPN クライアントは、VPN サーバ上に設置された独自の認証局によって公開鍵方式で認証される。

4.2 PoolGW の実装

前章までに述べた 3 機能を持つ PoolGW デモンプログラムを Java SE6 で実装した。それぞれの機能について以下に述べる。

4.2.1 ブロードキャスト・マルチキャスト中継機構

PeerPool における分散スマート環境間のユニキャスト通信は、PoolGW がルーティングするさいにオーバレイネットワークのアドレス空間を利用して、iptables¹¹⁾ を用いたアドレス変換を適用することによって実現されている (図 4.(a))。図中 Addr-A と Addr-W はそれぞれ通信ノード A と共有ノード W の実アドレス、Addr-A' と Addr-W' はそれぞれオーバレイネットワークのアドレス空間内の Addr-A と Addr-W の写像、Addr-A' と Addr-W' はそれぞれ A と W の仮想ローカルノードのアドレスである。

ブロードキャスト通信およびマルチキャスト通信も同様に PoolGW がアドレス変換を適用することによって実現する (図 4.(b))。ブロードキャストとマルチキャストのいずれの場合も、送信者アドレスはユニキャストの場合とまったく同様に変換する。宛先アドレスの変換は、ブロードキャストの場合とマルチキャストの場合で異なる。ブロードキャストの場合は、送信者側の PoolGW (図中 PoolGW G₁) が自スマート環境のブロードキャストアドレス (図中 Addr-S₁) 宛の packets をオーバレイネットワークのブロードキャストアドレス (図中 Addr-B_x) 宛に変換する。さらに、受信者側の PoolGW (図中 PoolGW G₂) がオーバレイネットワークのブロードキャストアドレス宛の packets を自スマート環境のブロードキャストアドレス宛に変換する。マルチキャストの場合は、宛先アドレス (図中 Addr-M_x) を変換しない。

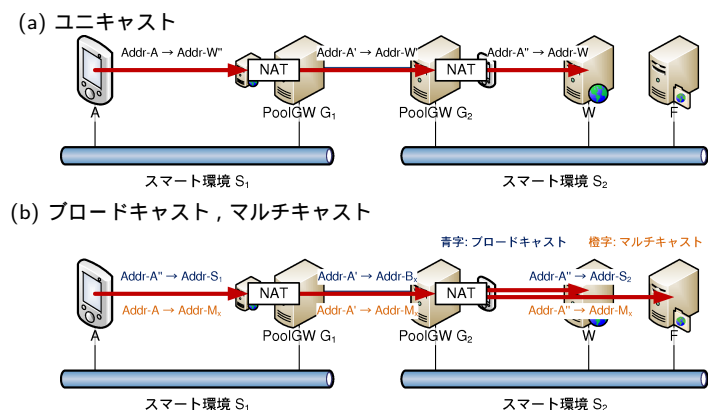


図 4 ブロードキャスト通信およびマルチキャスト通信のルーティングとアドレス変換

受信側の PoolGW から受信者への B/Mcast パケットの配送は、前章で述べたとおり、データリンク層レベルのユニキャストを反復することによる。すなわち、送信者と仮想的にリンクローカルなノードの数だけ B/Mcast IP パケットを複製し、ユニキャストの MAC フレームに納めてそれぞれの宛先に転送する。IP 層より下層で packets を処理するためにパケットキャプチャライブラリ Jpcap⁹⁾ を利用した。

4.2.2 プライベートドメイン名の別名解決機構

プライベートドメイン名の別名解決機構は、PoolGW における名前解決シーケンスを変更することにより実現する。PoolGW は、接尾辞 .pool を持つドメイン名に対する解決要求を PIM の操作要求 (ノードのチェックイン、仮想ローカルノード化など) と解釈し、処理する。また、それ以外の名前解決要求を既定ネームサーバ (NS) に転送することにより、通常の DNS サーバと同等の機能を提供する。本提案では、接尾辞が .pool 以外のときの名前解決シーケンスを以下のように変更することによって、遠隔のスマート環境の名前空間に属するプライベートドメイン名を仮想ローカルノードの別名として解決する。

- (1) 自スマート環境の既定 NS に問合せを転送する。
- (2) 成功返答を得れば、
 - (a) 当該成功返答をクライアントに返す。
- (3) 違えば、

- (a) オーバーレイネットワーク上のブロードキャストにより他の PoolGW に問合せを転送する。
- (b) 成功返答（共有ノードのアドレス）を得れば，
 - (i) 当該共有ノードが仮想ローカルノード化されていない場合は，仮想ローカルノードを生成する。
 - (ii) 仮想ローカルノードのアドレスを返答する。
- (c) 違えば，失敗（NXDOMAIN）を返答する。

この過程においてエラーが発生した場合は，直ちに失敗を返答する。

上の手順 3a で転送を受けた PoolGW は，以下の手順に従う。

- I. 自スマート環境の既定 NS に問合せを転送する。
- II. 成功返答を得れば，
 - (a) 得たアドレスを共有ノードのアドレスのリストと比較する。
 - (b) いずれかの要素と一致すれば，
 - (i) 当該共有ノードのアドレスを返答する。

この過程においてエラーが発生した場合および当該レコードが存在しない場合は，帯域の消費を抑制するために何も返答しない。失敗返答をしない設計のため，手順 3a でクエリを転送した PoolGW が当該レコードが存在しないことを判断する唯一の根拠は，一定時間内に返答がないためにタイムアウトすることである。したがって，この一定時間は，PoolGW 間の RTT (Round Trip Time) に対して十分に大きな値にする必要がある。本実装では 1000 ms とした。

図 5 は，プライベートドメイン名が仮想ローカルノードの別名として解決される場合の処理の流れを示す。図中 Addr-F, Addr-F', Addr-F'' はそれぞれ，ノード F の実アドレス，オーバーレイネットワークのアドレス空間内の Addr-F の写像，F の仮想ローカルノードのアドレスである。

4.2.3 ペイロード変換機構

ペイロード変換モジュールは，ユーザの利用形態に応じて任意のプロトコルに対応できるようにするため，プラグインの形態をとる。プラグインは PoolGW のファイルシステム上に配置し，PoolGW の起動に伴い，読み込まれるものとする。読み込まれたプラグインは当該プラグインが扱うプロトコルを識別するためのパラメータをプラグイン管理部に渡す。本実装では，実行速度を重視してプラグインが渡すパラメータを単純にポート番号とした。よって，ポート番号によってプロトコルが一意に決まらない場合，実装を変更する必要がある。

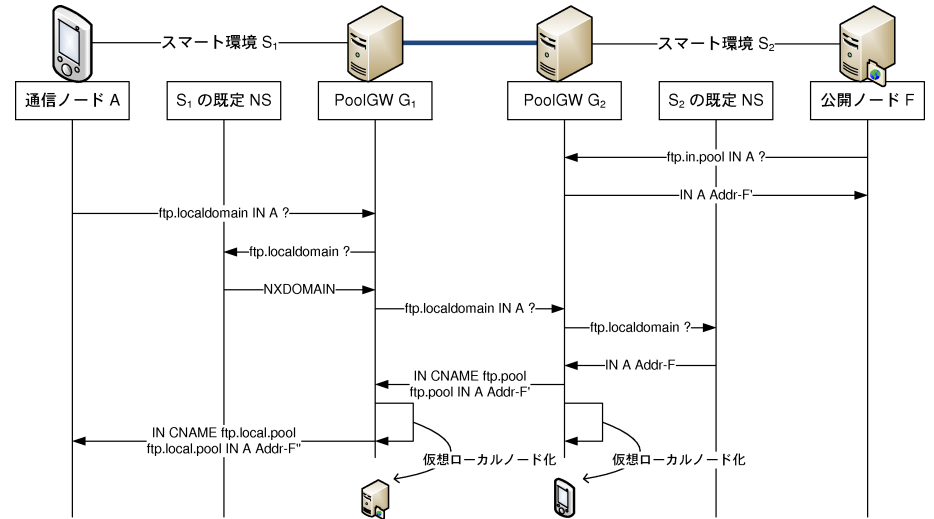


図 5 プライベートドメイン名から仮想ローカルノードへの別名解決のシーケンス

る。また，各プラグインは与えられたパケットのペイロードをアドレス変換による矛盾が起きないように書き換えるために，ペイロード変換ルーチンを実装する。

本稿では，ペイロード変換機構として Bonjour のサービス発見に使用される mDNS プロトコルのペイロード変換プラグインを実装した (図 6)。図中 Addr-B は通信ノード B の実アドレス，Addr-B'' はスマート環境 S₁ 内での B の仮想ローカルノードのアドレスである。またスマート環境 S₁ 内にある PoolGW G₁ には，ペイロード変換プラグインが実装されており，ノード A とノード B は，B/Mcast 通信が届く関係性 (3.1 節) であるとする。

まず，B が mDNS により，自身の Bonjour サービスの IP アドレスをペイロードに含むパケットをスマート環境 S₂ 上にマルチキャスト通信として送信する。PoolGW G₂ の B/Mcast 中継機構により，それらのパケットはスマート環境 S₁ 内にある PoolGW G₁ に到達する。G₁ は，このパケットが使用するポート番号と，読み込んだプラグインから与えられたパラメータと一致するかを判断する。一致した場合，G₁ はペイロード中に記述された B の実アドレス Addr-B を，B の仮想ローカルノードのアドレス Addr-B'' に変換し，共有空間に共有ノードとして存在するノードに対し書き換えたパケットを転送する。

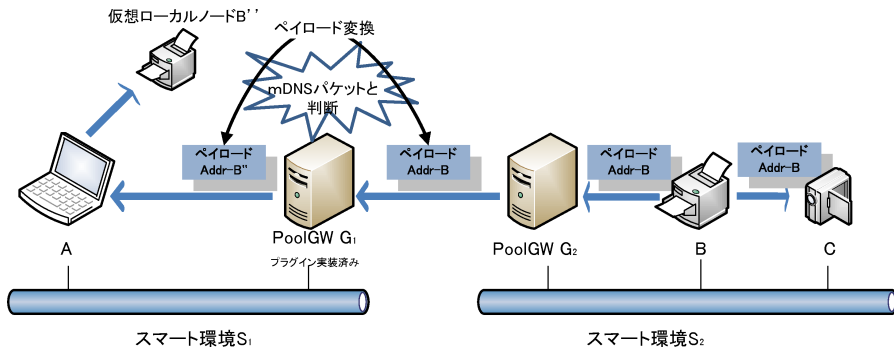


図 6 プラグインを実装した PoolGW によるペイロード変換

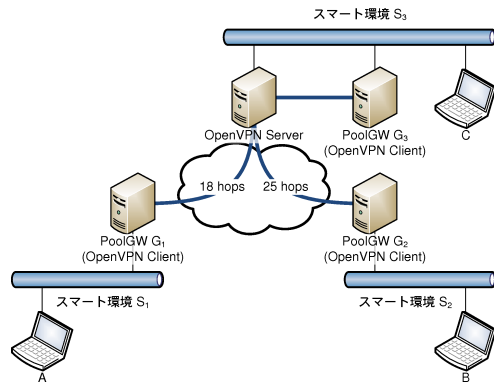


図 7 実験環境

これにより A は、サービスが存在するアドレスは、Addr-B'' であると認識し、続いての通信のやりとりは Addr-B'' に対して行う。Addr-B'' 宛てのパケットは PoolGW G₁ が受け取り、PoolGW G₂ を介し、実ノード B へと届けられる。

5. 評価

5.1 定量評価

図 7 に示した環境下で、本実装のボトルネックになると考えられる部分の性能を測定した。

	レイテンシ (Min., Max., Avg.)			スループット
A-B	72 ms	295 ms	82 ms	909 Kbps
A-C	36 ms	59 ms	38 ms	1360 Kbps
B-C	35 ms	111 ms	39 ms	922 Kbps

5.1.1 三角経路によるレイテンシとスループットへの影響

本実装における PoolGW 間のオーバーレイネットワークは、OpenVPN で構成している。OpenVPN は、1 台の VPN サーバを中心に 1 台以上の VPN クライアントが接続するスター型トポロジを形成する。複数の PoolGW を VPN クライアントで動作させる場合、それらが属するスマート環境間の通信に三角経路を生ずることによる遅延の増加および通信効率の低下が懸念される。そこで、PoolGW 間に三角経路を含む場合と含まない場合についてレイテンシとスループットを測定し、三角経路が性能に及ぼす影響を考察する。

図 7 に示したノード A, B, C のすべての 2 者間でレイテンシとスループットを測定した。レイテンシについては、64 バイトの ping (ICMP Echo Request) を 30 回送信し、RTT の平均、最小値および最大値を求めた。スループットについては、TCP のスループットを 60 秒間測定した。測定結果を表 1 に示す。

A-B 間のレイテンシは、おおよそ A-C 間のそれと B-C 間のその和に等しい。A-B 間のスループットは、おおよそ A-C 間のそれと B-C 間のその最小値に等しい。以降の実験については、三角経路の影響を分離するため、三角経路を含まない 2 点間で測定する。

5.1.2 プライベートドメイン名の別名解決の応答時間

従来 PeerPool における DNS インタフェースは、実用的な応答時間で動作することが実験により確認されている⁶⁾。しかし、提案したプライベートドメイン名の別名解決は、従来からのどの処理よりも問合せの転送回数が多いことから、応答時間に悪影響を与えることが懸念される。そこで、仮想ローカルノード名への別名解決の応答時間を測定する。

図 7 のノード C を PeerPool の共有空間にチェックインした上で、ノード A において C のプライベートドメイン名 c.s3.localdomain を解決する試行を 100 回繰り返した。このとき、97 回の成功返答と 3 回の失敗返答 (NXDOMAIN) を得た。失敗返答はいずれも、PoolGW G₁ から PoolGW G₃ への問合せがタイムアウトしたことが原因であった。また、失敗返答の直後 (約 1 秒後) にクエリを再送しているが、いずれも成功返答であった。成功返答の 97 回について、応答時間を表 2 に示す。

ところで、代表的な DNS クライアントプログラムである nslookup と dig のタイムア

表 2 プライベートドメイン名の別名解決の応答時間

Min.	Max.	Avg.
102 ms	1658 ms	315 ms

ウト値は、デフォルトでそれぞれ 2 秒と 5 秒である。また、Microsoft Windows 2000 の TCP/IP サービスにおける DNS クライアントの実装では、1 秒以内にサーバからの返答がない場合、最終的に同 17 秒後にタイムアウトを報告するまでに数回にわたって問合せを再送する¹⁰⁾。以上のことから、本実装の応答時間は、実用上問題ないと考えられる。

5.1.3 B/Mcast 中継機構のスケラビリティ

物理的に受信ノードを増加させる代わりに、ノード C が Proxy ARP によってすべてのパケットの複製を収集し、現実的な受信ノード数で B/Mcast の中継が機能するか否かを実験により確認した。

図 7 に示したノード A からのブロードキャストパケットを中継し、スマート環境 S_3 内の受信ノード数と送信ビットレートを変化させてパケット損失率を測定した。なお、ノード C が Proxy ARP によってすべてのパケットの複製を収集するため、パケット損失率が 0 ならば、C が受信するパケット数は、

$$A \text{ が送信したパケット数} \times G_3 \text{ のパケット複製数}$$

である。したがって、パケット損失率は次式で算出した。

$$1 - \frac{C \text{ が受信したパケット数}}{A \text{ が送信したパケット数} \times G_3 \text{ のパケット複製数}}$$

測定結果を図 8 に示す（図中 n は G_3 のパケット複製数）。

測定結果より、受信ノード数が 2^5 以下で送信ビットレートが 128 Kbps 以下ならば、ほとんどパケットを損失しない。受信ノード数または送信ビットレートのいずれがこの水準を越えてもパケット損失率が増大する。スループットは、映像のストリーミング配信には不足だが、UPnP や Bonjour などのサービス発見のためには十分であると考えられる。一方、受信ノード数が 2^5 を超える場合が現実的に考えられるため、より多くの受信ノード数に耐えられるよう改善が必要である

5.1.4 ベイロード変換による通信への影響

本節では、ベイロード変換が行われている状態と行われていない状態のレイテンシおよびスループットを計測することで、ベイロード変換がどれほどの影響を与えるかを考察した。

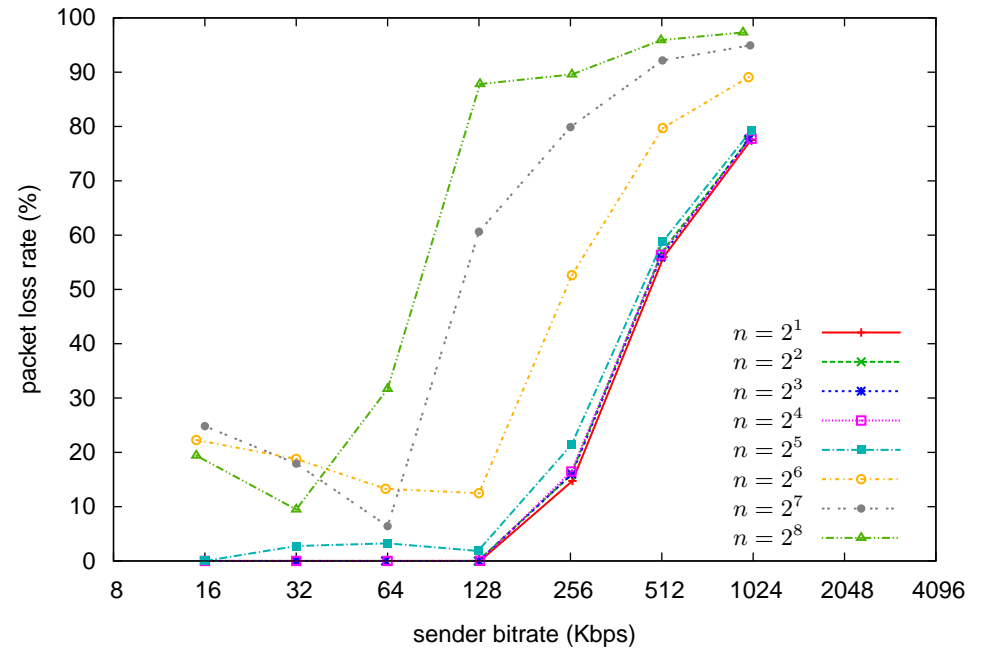


図 8 B/Mcast 中継機構におけるパケット損失率

この測定においては、図 9 のような環境下で測定を行った。

図 9 に示したノード A とノード B の 2 者間でレイテンシとスループットを測定した。またノード C は一定の間隔で mDNS パケットを送信している。スマート環境 S_1 内の PoolGW G_1 にはベイロード変換プラグインが実装されており、ノード C が送信する mDNS パケットは G_1 によってベイロード変換され、ノード A へと到達する。また、C が送信するパケットの送信間隔でビットレートを調整し、それぞれのビットレートに応じて、レイテンシおよびスループットを求めた。

レイテンシについては、1 キロバイトの ping(ICMP Echo Request) を 30 回送信し、RTT の平均、最小値および最大値を求めた。スループットについては、TCP のスループットを 60 秒間測定した。測定結果を図 10,11 に示す。これによれば、ベイロード変換を行うことで、スループットおよびレイテンシに影響が出ているのは明らかである。これはベイロード

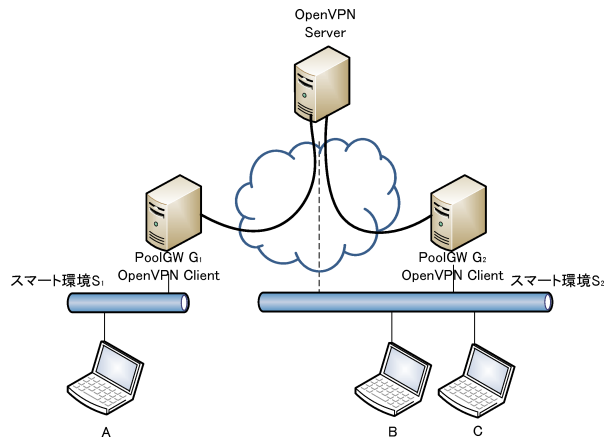


図 9 実験環境 2

変換を行うことで、ペイロード変換モジュールが PoolGW の CPU リソースを多く使用し、PoolGW の OpenVPN の動作を圧迫することが原因だと思われる。また、送信ビットレートを 32Kbps 以上とした場合、以後のスループットはほぼ一定の値を保っていた。これは AB 間でパケット落ちしているか、OpenVPN への CPU の割り当てが一定以下にはならないようになっているなどが考えられる。これに関しては今後詳細に測定を行っていく。

5.2 定性評価

5.2.1 モデルおよび要件との整合性

まず、本実装と PIM モデルとの整合性を考察する。本実装は、従来の PeerPool のすべての機能を継承している。また、2.6 節に掲げた 3 機能はいずれも、PoolGW のデーモンプログラムの実装変更のみで実現している。したがって、本実装は従来の PeerPool と同様に PIM の実装である。

次に、本実装が 2.2 節で述べた要件を満足するか否かを検証する。上と同様の理由から、本実装は要件 2-5 を満足する。従来の PeerPool は、2.5.3 節で述べた制約により要件 1 を満たさなかった。本実装では、B/Mcast 中継機構により B/Mcast 通信が可能になった。また、プライベートドメイン名の別名解決機構によりペイロードにプライベートドメイン名を含むプロトコルが利用可能になった。さらに、ペイロード中に IP アドレスを含むプロトコルについては、ペイロード変換プラグインを組み込むことにより対応できる。以上により、

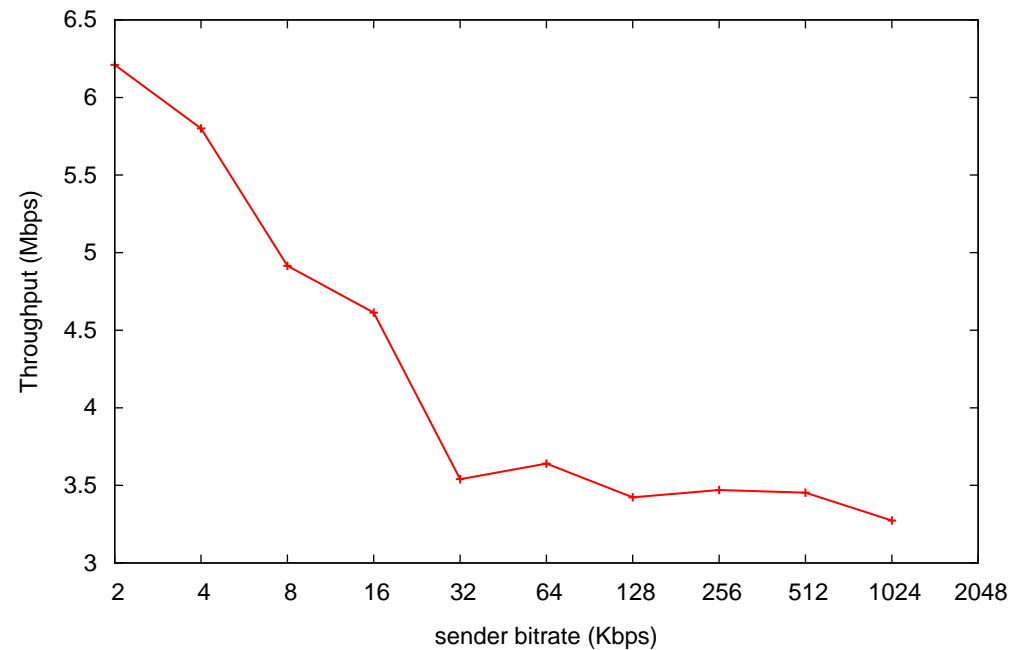


図 10 ペイロード変換によるスループットへの影響

要件が満たされた。

5.2.2 利用できるアプリケーションの比較

本節では、本提案、従来の PeerPool および 2 章で挙げた関連技術について、それぞれが分散スマート環境間での利用を可能にするアプリケーションを比較することにより、本研究の成果を定性的に評価する。スマート環境で利用されることが想定される代用的なアプリケーションである Web アプリケーション、UPnP、Bonjour を対象とした比較表 (表 3) を示す。

従来の PeerPool は、ペイロードにプライベートドメイン名および IP アドレスを含まない任意のユニキャスト通信を利用可能にする。したがって、マルチキャスト通信が必須である UPnP と Bonjour を利用できない。また、Web アプリケーションはデータ中にプライベートドメイン名およびプライベートアドレスを含む場合に問題となるため、条件付きで利

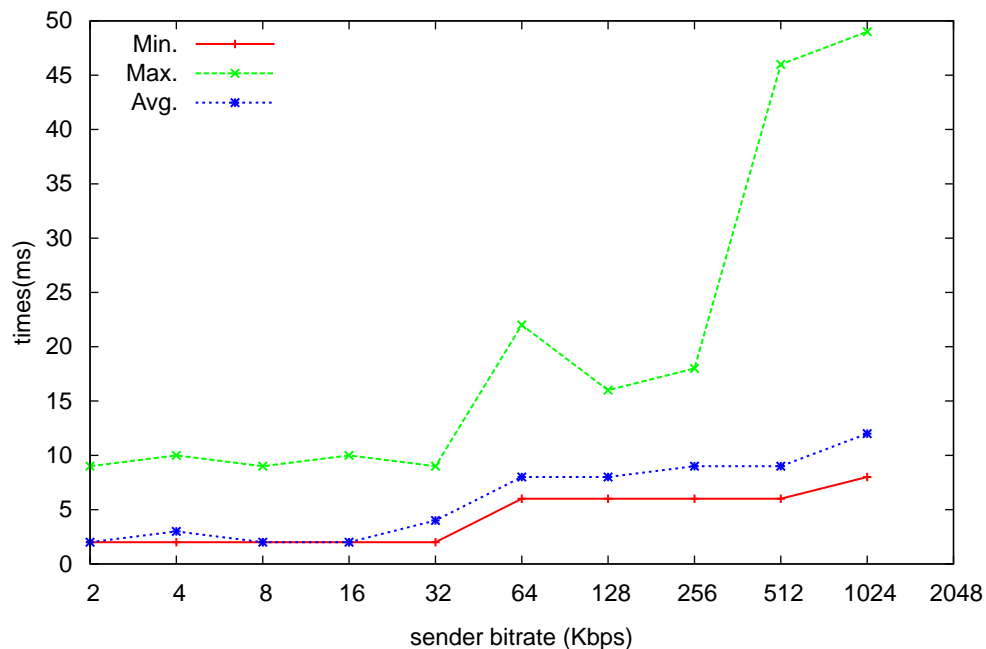


図 11 ペイロード変換によるレイテンシへの影響

用可能とした。

本提案では、マルチキャスト通信に対応することにより、UPnP と Bonjour が利用可能になった。また、プライベートドメイン名の別名解決により、データ中にドメイン名を含む場合の制約が解消された。さらに、データ中に IP アドレスを含む場合もプラグインによりプロトコルごとに個別に対応可能である。

RNSplicer は、Bonjour に特化した実装であるから原則としてそれ以外のプロトコルを利用できない。Shepherd は、従前の PeerPool と同等である。以上より、本研究によって分散スマート環境間で利用できるアプリケーションの幅が広がり、既存技術と比べても優位性が認められるといえる。

表 3 利用可能なアプリケーションの比較

√: 利用可 #: 条件付き利用可 ×: 利用不可

	Web	UPnP	Bonjour
本提案 + プラグイン	√	√	√
本提案	# [†]	# [†]	# [†]
PeerPool	# ^{†‡}	×	×
RNSplicer	# [§]	×	√
Shepherd	# ^{†‡}	×	×

[†] データ中に IP アドレスを含む場合に制約あり

[‡] データ中にドメイン名を含む場合に制約あり

[§] Bonjour で広告されている場合のみ利用可

6. まとめと今後の課題

本論文では、分散スマート環境間連携の実現を目標としてスマート環境接続モデル PIM を導入し、その実装および評価について述べた。本モデルによれば、オーバーレイネットワーク上の共有空間にノードの写像を適時生成することにより、分散スマート環境間で選択的なノードの共有を可能にする。提案したモデルについて、従来より我々が提案してきたネットワーク技術である PeerPool に 3 機能を追加することによる実装を示した。これらの機能により、既存のノードやネットワーク構成への変更を最小限に抑えつつ広範のアプリケーションに対応するスマート環境間接続技術を実現した。評価として、本実装を用いて 3 拠点のプライベートネットワーク間を接続し、性能を測定した。その結果、提案したプライベートドメイン名の別名解決機構が実用的な応答時間で動作することを確認した。また、提案したペイロード変換機構は、ペイロード変換に伴う通信への影響について述べ、一方、提案した B/Mcast 中継機構は、受信ノードの増加に伴うパケット損失率の増加が甚だしく、改善が望まれることを述べた。また、提案したシステムと関連技術について、それぞれが分散スマート環境間での利用を可能にするアプリケーションを比較し、本研究の成果が有用であることを論じた。

B/Mcast 中継機構のパケット損失の原因は、Jpcap を用いた実装により、PoolGW が流れてくるパケットを全てユーザランドにまで引き上げて処理を行うことによる過負荷が原因である。そこで、iptables¹¹⁾ において使用できる ULOG オプションを上手く用いることで、必要なパケットの処理のみユーザランドで行うといったことが可能となる。よって、こうした実装を行うことで本モデルの性能を引き上げることを今後の課題とする。

参 考 文 献

- 1) T. Okoshi, S. Wakayama, Y. Sugita, T. Iwamoto, J. Nakazawa, T. Nagata, D. Furusaka, M. Iwai, A. Kusumoto, N. Harashima, J. Yura, N. Nishio, Y. Tobe, Y. Ikeda, and H. Tokuda. Smart Space Laboratory Project: Toward the Next Generation Computing Environment. *IEEE Third Workshop on Networked Appliances (IWNA) 2001*, 2001.
- 2) 河口信夫, 稲垣康善. cogma: 動的ネットワーク環境における組み込み機器間の連携用ミドルウェア. 情報処理学会コンピュータシステム・シンポジウム, pages 1–8, 2001.
- 3) The NIST Smart Space Project. Online, 2008. <http://www.nist.gov/smartspace/>.
- 4) Mizukami, T. and Cho, K. Tunneling with service discovery: a remote access model and implementation. *Consumer Communications and Networking Conference, 2006 3rd IEEE*, 1:223–227, 2006.
- 5) 青柳禎矩, 高橋ひとみ, 斉藤匡人, 間博人, 徳田英幸. Shepherd: 利用者主導型仮想プライベート・サブネットワーク構築機構. 電子情報通信学会技術研究報告. *NS*, ネットワークシステム, 106(577):227–232, 2007.
- 6) 中野悦史, 西尾信彦. Peer Pool: DNS クエリによって構成されるオーバーレイネットワーク. *インターネットコンファレンス 2007 (IC2007)*, pages 3–12, 2007.
- 7) UPnP Forum. Online, 2009. <http://upnp.org/>.
- 8) Bonjour. Online, 2009. <http://developer.apple.com/networking/bonjour/>.
- 9) K. Fujii. Jpcap – a Java library for capturing and sending network packets. Online, 2007. <http://netresearch.ics.uci.edu/kfujii/jpcap/doc/>.
- 10) Microsoft Windows 2000 TCP/IP Implementation Details, Microsoft TechNet. Online, 2009. <http://technet.microsoft.com/en-au/library/bb726981.aspx>.
- 11) IPTABLES, 2009. <http://www.linux.or.jp/JM/html/iptables/man8/iptables.8.html>.