修士論文

実空間操作に基づくリンク層における
デバイスグルーピング機構の設計と実装

2006年2月3日

指導教員　青山　友紀　教授
森川　博之　助教授

東京大学大学院情報理工学系研究科
電子情報学専攻　46410

小森田 賢史

内容梗概

今後ユビキタスコンピューティング環境がわれわれの生活に浸透し，ネットワークに対応したデバイスが増加すると，それらをユーザが容易かつ安全に連携させ利用することが重要になる．本稿では， 既存サービス発見機構と連携した実空間上の直感的なデバイス選択，及びデバイスグルーピングによる通信保護に着目し，実空間操作に基づいてリンク層で安全な仮想通信網することで，デバイスを安全かつ容易に連携できる機構“ ViCon ”を提案し，ViCon の設計と実装について述べる．仮想通信網上でサービス発見とデバイス間通信を行うことにより，既存のサービス発見機構へ変更を加えることなく実空間上でのデバイス選択操作を反映させ，また安全な通信を実現する．動作確認と性能評価のため，実空間インタフェース機構と高性能デバイスと組込型デバイスとして PC と iPod にデバイスグルーピング機構を実装した．性能評価を行った結果，通信制御機構を用いた場合に PC では 22%の CPU 負荷増加，iPod では 24%の速度低下となった．

目 次

図 目 次

第1章

序論

1.1　本研究の背景と目的

現在では，ADSL や FTTH という安価で高速なインターネット環境や，1Gbps 程の高速な LAN 環境が整いつつある．このようなネットワークの普及と共に，ネットワーク上でサービスを行うことや，ネットワークを介してデバイスを利用，制御することが行われるようになってきた．

従来，PC 機器をはじめ家電の AV 機器は主に専用のケーブルを用い 1 対 1 で接続されていた．専用ケーブルでの接続は物理インフラの多様化を招き，相互接続性の問題があった．一方で LAN のパフォーマンスも向上してきたことから，LAN 対応のデバイスが増えている．デバイスを LAN に接続することでデバイス間の距離やコネクタ数など物理的な制約にとらわれず利用することができる．このような環境では LAN に接続されたデバイス同士を動的に組み合わせて利用することで多様なサービスを生み出すことができるようになると考えられる．

LAN 上のデバイスを相互に連携させるためには、次の 2 つの機構が重要になる．1 つ目は連携するデバイスを容易に選択するための機構である．デバイスを連携させるには，まずユーザが利用しようとするデバイスを選択しなくてはならない．例えば，“ 目の前のスピーカから音楽を出力したい ”,“ あのディスプレイにプレゼンテーションを表示したい ”というデバイスの指定が必要である．現在ではこのようなデバイスを指定する方法は PC のディスプレイに LAN 上のデバイスを列挙し，ユーザがそれを選択することで実現されている．この手法ではデバイス数が増加するにつれ列挙されるデバイスの数も増加し，最終的にはユーザが選択するのが困難になる．それらをを改善すべく，直感的にデバイスを選択することを目的として，実空間上で直接デバイスを選択・操作するための研究が行われている [1, 2, 3, 4]．しかしながら，これらの研究は実空間上の操作を実現しているものの，デバイスが連携するためのプロトコルはそれぞれの実空間操作に適した独自の形式を用いている．そのため，デバイス間の相互接続性を考慮した既存サービス発見機構との乖離が大きく，導入することが困難である．

2 つ目はデバイス間の通信を保護するための機構である．USB などの専用ケーブルを用いたデバイス間の通信は，インターネットを介し様々なサーバと接続する PC の通信と異なり，専用ケーブル内に閉じた安全なものであった．それに対して LAN 対応のデバイスを利用する場合ではデバイス間の通信におけるセキュリティが重要になってくる．今後 LAN に対応するデバイスが増加して行くと考えられるが，その結果デバイスが行う通信を他のデバイスから盗聴，妨害されることが問題となることが想定される．例えばキーボードが LAN 対応になった場合，あるユーザのキーボードの入力を他のユーザに盗聴されることなどが考えられる．これまでの通信を保護する技術は one-to-one 型の通信形態に利用することを目的としていた．例えば SSL/TLS[5, 6] を用いてユーザが許可したノードのみを接続可能にするためにはある任意のノードで認証情報を管理する必要がある．しかしながら，LAN に接続された複数のデバイスを連携させるといった用途を考えた場合，認証情報を集中管理しなければならない SSL/TLS などの通信保護技術は使用できない．

それに対して，筆者らはデバイス連携を容易かつ安全に行うことを目的とし，実空間操作によりデバイスをグルーピングしリンク層で安全な仮想通信網を構築する機構“ ViCon ”を提案する．ViCon では，既存のサービス発見機構とデバイス間の通信は，実空間操作に基づいて構築した仮想通信網上で行う．これにより，サービス発見の対象となるデバイスを仮想通信網上のデバイスに限定し，実空間での選択を既存サービス発見機構へ変更を加えることなく反映することが実現できる．また，デバイス間の通信を仮想通信網上で行うため，デバイス間の全ての通信を保護することが実現ができる．

本稿では ViCon の設計と実装について述べる．実装では PAVENET モジュール [7] を用いた実空間インタフェース機構と，高性能デバイスと組込型デバイスとして PC と iPod それぞれにデバ

イスグルーピング機構を実装した．PC では，PC 画面を他のディスプレイデバイスに出力するサービスを作成し，iPod では音楽を再生するサービスを作成した．PC と iPod において実空間上で選択したデバイス間でサービス発見を行い，サービスを利用できることを確認した．また，仮想通信網上の通信では CPU 処理能力にに余裕のある PC では 22%の CPU 負荷増加と 1%の速度低下となり，CPU 処理能力に余力がない iPod では 0%の負荷増加と 24%の速度低下となった．

1.2　本論文の構成

本論文は，以下の各章によって構成される．

第 1 章 序論
第 2 章 関連研究・技術
第 3 章 既存技術の問題
第 4 章 実空間操作に基づくリンク層におけるデバイスグルーピング
第 5 章 ViCon 実装
第 6 章 ViCon 実験・性能評価
第 7 章 結論

本稿では，まず第 2 章で既存のサービス発見機構と実空間インタフェースの研究，及び通信保護機構の技術を紹介し，第 3 章でその問題点を明らかにする．第 3 章で述べた問題に対し，第 4 章において実空間上でのデバイス選択により，デバイスをグルーピングし安全かつ容易にデバイス連携を実現する機構の設計を述べる．第 5 章では提案した手法に従い，より細かな仕様に触れながら実装について述べる．第 6 章では実装した ViCon の動作確認と性能評価を述べ，最後に第 7 章でまとめとする．

第2章

関連研究・技術

2.1　はじめに

ネットワークの普及と共にネットワーク上で提供されるサービスやネットワーク対応のデバイスは増加の一途をたどり，それに伴いより容易にサービスやデバイスを利用するためのネットワークプロトコルやユーザインタフェースが提案されている．また，ネットワークサービスの増加と共に，セキュリティ意識が高まり，安全にサービスやデバイスを利用することが求められている．

本章では，サービスをユーザが容易に利用するための関連研究，関連技術，及び安全にサービスやデバイスを利用するための認証，通信保護に関する研究について述べる．

2.2　ネットワークサービス

2.2.1　ネットワークサービスの遷移

グローバル規模のコミュニケーション手段として，メールや WWW といったサービスが利用されてきた．近年では LAN の普及とネットワーク対応機器の増加に伴い，LAN 上でデバイスを連携させて利用することが増えている．

グローバル規模のネットワークサービス

インターネットは手紙，電話，FAX といったメディアと比較し，それらを網羅するメディア的特性を持つ．例えば，多方向・双方向性について考える．直筆の手紙や電話，FAX は基本的には個人対個人の間に介在し，しかもその個人は，たいてい「どこそこのだれだれ」という具合に特定される．一方，テレビや新聞のような「マスコミ」は，不特定多数の人々に発信できる．けれども，電話や FAX なら発信者と受信者は相互に役割転換できるが，マスコミでは情報の発信者はひたすら情報を伝えつづけ，受け手は流されているものを一方的に見たり聴くだけである．インターネットならば，1 対 1 で双方向にやりとりをすることも，一人が特定多数の人々に発信することも，不特定多数の人々に向かって伝えることも可能である．

また，インターネットは伝達を従来技術よりも安価に行うことができる．マスコミの発信者はそのための設備や人材にかかる多額の経費を負担しなければならない．到底個人ではまかえず，組織的に動かざるをえない．書籍ような場合でも同様で，かたちを選ばなければテキストを作ることは個人でも可能だが，それを伝達するためには伝達する量に比例して費用がかかる．つまり，より多くの人々に伝えようとするとその分だけ費用と手間がかかる．その点，インターネットでは多くの人に見せても費用は変わらない．ひとりが見る情報量が少なくても多くても費用は変わらない．サーバーを読むため訪れる者が通信の費用を負担する．このように，インターネットは，手紙・電話・FAX のような個人性・双方向性とマスコミのような不特定性・多方向性をあわせもち，保存が容易であること映像や音声を組み合わせられること，しかもそれを安価で個人的に行うことができる．

高いメディア性と対費用効果をもつインターネットは，国境を越えたグローバル規模でサービスが展開されてきた．このグローバル規模で行われている代表的なサービスを挙げる．

電子メール（E-mail）　電子メールでは，パソコンの画面で文章（あるいは，映像や音声，ソフトウェア）を書いて，メーラ[1]で相手のアドレスを指定して，インターネットを介して送ることに

1　電子メールを読み書きするためのソフトウエアのこと

なる．なお電子メールの機能を利用して，特定の人々の間でのメール配信の方法としてメーリング・リストがある．後述のネットニュースや掲示板がメールによって行われると考えてよい．たとえば複数の人々の間で共同でなんらかの作業をしたり議論をしたりする場合に便利である．

チャット（Chat）　電子メールは手紙的な性格をもつもので，送りつけても相手がメールを受信して読まなければ伝わらない．その点,「チャット」は，パソコンの画面上で，同時に，相互に，しかも複数の人々による会話を可能にする．チャットをするには，チャットのプログラムを設けているホームページを使うか，または同じソフトを使って特定の相手と話すことができる．

ネットニュース（NetNews）　町中の掲示板のように，ネットワークの中に掲示板を設置し，不特定多数の人が見たり書いたりすることのできるようにしたものが，ネットニュースである．電子掲示板の草分けと言われる USE ネットには 5000 を超えるニューズグループやディスカッショングループがあり，文化や社会，法律，科学，建築，工学，趣味などさまざまなニュースがジャンル別に細分されて収められている．日本語で読める代表的なニュースグループでは fj や tnn，japan，kanto などがある．

ワールド・ワイド・ウェブ (World Wide Web)　インターネットといえば，長い間，文字や数字やソフトウエアなどをやりとりしたり，遠くにあるコンピュータを遠隔操作することが主だった．しかし，1992 年に HTML(Hyper Text Markup Language) というコンピュータ言語によって書かれた WWW(World Wide Web) の規格が発表され，文字だけでなく音声や動画や写真などを統一的な紙面[2]として表示し，誰にでも容易に見せることができるようになった．どのようなページがあるかは，サーチエンジンと呼ばれる検索のためのページを使って探す．近年では，Google[3]といった検索エンジンや多数のポータルサイト[4]が激しい競争を繰り広げている．

また，WWW をベースとした掲示板や電子商取引（Electric commerce）も普及が広がっている．

ソーシャルネットワーキングサイト (Social Networking Site)　ソーシャルネットワーキングサイトとは，参加者が互いに友人を紹介しあって，新たな友人関係を広げることを目的に開設されたコミュニティ型の Web サイトである．誰でも自由に参加できるサービスと,「既存の参加者からの招待がないと参加できない」というシステムになっているサービスがある．自分のプロフィールや写真を公開する機能や，新しくできた「友人」を登録するアドレス帳，友人に別の友人を紹介する機能，サイト内の友人のみ閲覧できる日記帳，友人間でのメッセージ交換に使う掲示板やカレンダーなどの機能が提供される．

電子商取引 (Electric Commerce)　電子商取引インターネットなどのネットワークを利用して，契約や決済などを行なう取引形態を指す．ネットワークの種類や取引の内容を限定しない，包括的な意味を持つ言葉である．従来から企業間の取引の一部は EDI(Electronic Data Interchange) などの技術を使って電子化されていたが，インターネットが一般消費者に普及するにつれて，消費者を直接対象にした電子商取引サービスが急激に成長している．

インターネットを通じた電子商取引は徐々に成熟期を迎えつつあり，決済・流通システムや決済データのセキュリティ保護システムなど，技術的な問題は解決しつつある．しかし，個人情報の管

2　HP: ホームページ
3　http://www.google.com/
4　インターネットの入り口となる巨大な Web サイト．Yahoo!,Excite,Infoseek 他 ISP など

理が不十分なことによるデータ漏洩や，ネット詐欺多発など，電子商取引ならではの問題点も顕在化しており，対策が急がれている．

LAN 内でのデバイス連携

エンドネットワークである LAN が広く普及し，グローバル規模ではなく閉じた LAN 内でサービスを行うことが増えている．グローバル規模のサービスでは，サービスを行うサーバを設置しユーザはそれらを介して，他のユーザとコミュニケーションを図るものが多数であった．LAN でのサービスは，グローバル規模のサービスと異なり他のユーザとのコミュニケーションを図るものではなく，LAN 内に存在するデバイス利用するためものが多い．

例えば，プリンタやスピーカ，ディスプレイなど，元来はシリアルケーブルやパラレルケーブル，USB(Universal Serial Bus) といった専用線で接続され利用されていた．しかしながら，専用線で PC とデバイスを 1 対 1 で接続する場合，デバイスを共有することができず，離れた場所に設置したデバイスを利用するためにはケーブルを引き延ばす必要があるなどのデメリットがある．そこで，デバイスを LAN へ接続し，ネットワーク上でデバイスを連携させ利用することが行われている．

デバイスをネットワークを介して連携して利用することは，従来の OA(Office Automation) 機器に止まらず，テレビやビデオといった家電へ対応させた情報家電が普及を始めている．1990 年代末頃から，従来の白物家電とは異なる，ディジタル通信・処理すなわち IT 技術を利用した情報家電と呼ばれる新しい機器が，一般家庭に導入されるようになった．情報家電は，携帯電話の普及に伴って発展を遂げているモバイルコンピューティングとともに，日本が世界をリードする可能性を持った，ユビキタスネットワーク社会を形成する重要な技術となる．AV 家電や PC 機器をネットワークに接続し即利用するためのプラグ・アンド・プレイ機能については，IP（Internet Protocol）をベースとする UPnP（Universal Plug & Play）[8] の標準化が進展している．ホームサーバと呼ばれる情報家電は，ホームネットワーク上のミドルウェアに位置づけられる UPnP を用いて，ホーム機器群を統一的に制御管理し，ルータ機能も備えて屋外との通信のゲートウェイとしても機能し，時に DVD 等の映像録画等のための大容量ファイルを備え，今後重要な役割を果たしていく．

LAN 上にデバイスを接続し，連携させて利用するサービスを以下に挙げる．

ネットワークプリンタ (Network Printer)　LAN などのネットワークに接続され，ネットワーク上の複数のユーザが利用できるように共有されたプリンタ.「共有プリンタ」とも呼ばれる．プリンタを直接パソコンに接続してそのパソコンをネットワークに接続する場合と,「プリンタサーバ」などの名前で呼ばれる専用の装置を介して接続する場合がある．また，プリンタ内部にプリンタサーバの機能を持たせておき，直接ネットワークに接続できるようになっているプリンタも存在する．ネットワークが普及する以前には接続切り替え機などを使って擬似的にプリンタの共有を実現していたが，この方法では印刷が完了するまで切り替えができないため，一つの印刷が終わってからでないと他のコンピュータから印刷命令ができなかった．

ネットワークオーディオ (Network Audio)　本来専用の AV ケーブルで接続されていたオーディオ機器を，Ethernet 上に接続し利用できるもの．現在では，Apple 社 [9] の AirMac Express が対応する．専用の再生ソフトである iTunes は同一サブネット内にある AirTunes 対応デバイスを自動的に検索し，接続可能なデバイスがあれば，音声出力先デバイスとして選択可能になる．たとえば AirMac Express を 2 個用い，それぞれリビングルームと寝室という名前を付け，対応する部屋に置いておく．すると，iTunes で“ リビングルーム ”を選んで音楽を再生すればリビングルームの

AirMac Express が,“ 寝室 ”を選べば寝室の AirMac Express が，それぞれ音楽を奏で始める．また，PC から汎用的に用いるシステムとして NICT[5]では，ネットワークにダイレクトに接続できるスピーカーが開発されている．これは PC の音声データを受信・再生でき，無線 LAN で音声をリモート再生するなど，多彩な使い方が可能である．

ネットワークディスプレイ (Network Display)　従来は，小さな LED 表示や電光掲示板などが主であったが，PC をセカンドディスプレイとして利用するネットワークサービス「MaxiVista」[10] や，直接ネットワークに対応したプラズマディスプレイ，プロジェクタなどが登場している．ネットワークに対応したディスプレイは，会議などディスプレイを共有する場で，RGB ケーブルを引き回すことなく，容易にプレゼンテーション資料などをできる．また，ディスプレイを分割して同時に利用するなど，物理ケーブル数にとらわれない利用が可能であり，柔軟性を向上させている．

ユビキタスコンピューティング

今後は，LAN 上でのデバイス連携に止まらず，さらに膨大な数のデバイスが身の回りにあふれ，それらを連携させ利用していくと予想されている．このような環境はユビキタス社会といわれ，身の回りに膨大な数のデバイスやサービスが偏在する環境へと移行すると，よりネットワークサービスを利用する機会が急増する．ユビキタスとは，生活や社会の至る所にコンピュータが存在し，コンピュータ同士が自律的に連携して動作することにより，人間の生活を強力にバックアップする情報環境である．1989 年に Xerox 社のパロアルト研究所が提唱した概念であるが，携帯電話などを中心とした小型情報端末の進化に代表されるコンピュータの小型化や，インターネットの爆発的な普及などの通信技術の発展・浸透に伴って，再び注目が集まっている．ユビキタスコンピューティングにおいては，コンピュータはその存在を意識させることなく，必要に応じてネットワークに蓄積された個人情報などを参照しながら，自動的に他のコンピュータと連携して処理を行なう．ユビキタスコンピューティングの研究から生まれた技術としては，VICS 情報と連動した経路探索・周辺情報探索を行なうカーナビゲーションシステムや，衣服と一体化することにより「身にまとう」ことができるウェアラブルコンピュータなどがある．

2.2.2　既存のサービス発見機構

LAN 上でデバイスを容易に連携させ利用するには，サービス発見機構が用いられる．サービス発見機構は，デバイスやサービスを利用するためのアドレッシングやサービス記述，サービス発見などに用いられるプロトコルを定めている．ここでは，後者について現在進められている規格と，構成するプロトコルについて述べる．

規格

規格では，既に標準化が進められているサービス発見機構について述べる．複数のプロトコルを組み合わせ，サービス発見機構として動作するよう全体のアーキテクチャが定められている．

5　独立行政法人・情報通信研究機構

UPnP（Universal Plug and Play）　UPnP[8] は，1998 年にサン・マイクロシステムズの Jini に対抗する意味合いでマイクロソフトによって提案された．ネットワークに接続される各種機器の検出と各種の設定を自動化し，主に一般家庭の利用者が利用しやすいネットワーク・プラットフォームを提供するための技術である．例えば，DVD で映画を観るために，DVD プレイヤの電源を入れると，スクリーンが天井から下りてプロジェクタのランプが点灯し，カーテンが閉まって照明が暗くなり，エアコンも ON になって映画の再生，投影が始まる，というような複数の機器の連携動作も可能となる．UPnP に関わる仕様の議論を行うためのオープンな業界組織として，UPnP フォーラムが翌年 1999 年 6 月に設立され，2004 年 8 月末現在約 700 の企業，団体が参加している．

UPnP の概念は，PC のアーキテクチャに採用されていたプラグ・アンド・プレイをネットワークのレベルにまで広げようとするものであり，機器のアドレッシングと検出，機器の機能の記述とその公開，制御とイベント処理等の基本仕様を定めている．これらの基本仕様は，インターネットで広く採用されている TCP/IP, HTTP, XML, SOAP 等に加え，UPnP フォーラムで公開されている SSDP, GENA 等の独自の技術を組合せている． このプロトコルスタックを図 2.1 に示す．

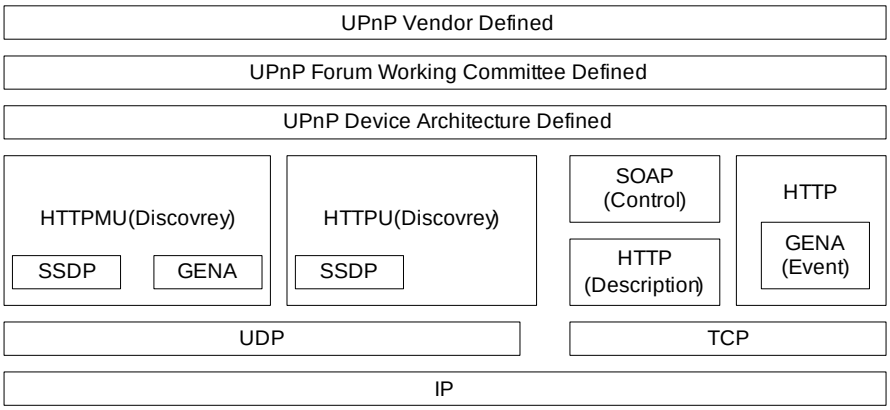

図 2.1: UPnP プロトコルスタック

DLNA（Digital Living Network Alliance）　2003 年 6 月に DHWG（Digital Home Working Group）として設立され，2004 年 6 月に名称が DLNA[11] に変更された．このアライアンスの目的は，音楽，写真，ビデオなどのディジタルコンテンツを，家電，PC，携帯端末等の機器間で容易に共有するための相互接続性のフレームワークを構築するため，公開された業界標準をベースとした設計ガイドラインを開発し，関係する業界や商品カテゴリにまたがるディジタル環境を融合することにある．

プロモータ企業として，富士通，HP，インテル，ケンウッド，レノボ，マイクロソフト，NEC パーソナルプロダクツ，ノキア，松下電器，フィリップス，サムソン，シャープ，ソニー，ST マイクロエレクトロニクス，トムソン，TI の 14 社（アルファベット順）が名を連ね，2004 年 8 月末現在 140 社余りがアライアンスに参加している．

2004 年 6 月に，家電共通の相互接続フレームワークに関する動向を踏まえ，IP ネットワークの家庭用プラットフォーム，商品，アプリケーションの詳細に関し,「ディジタルホーム設計ガイドライン V1.0」を発表した．基本的なパーソナルメディアとエンターテインメント目的のシナリオに

焦点を当てている．今後，新しいシナリオや商品に順次展開，見直しを図っていく．
設計ガイドライン V1.0 に採用される主要技術は，

- IP ネットワーク：
  TCP/IP プロトコルスウィート．将来は IPv4 と IPv6 の共存を想定．
- 物理ネットワーク：
  Ethernet（100Base-TX），無線 LAN（IEEE802.11b/a/g）
- データ転送：
  HTTP
- デバイスの検出・制御インタフェース：
  UPnP オプションとして，ブリッジング等により IEEE1394 対応機器とも接続

OSGi Alliance　OSGI Alliance[12] は 1999 年 3 月に設立された業界標準化団体で，当初は，Java 技術を利用し当時提案されていた Jini[13]，IEEE1394 上の HAVi[14](Home Audio Video interoperability) などの，PC 機器や AV 家電を接続するための様々なインタフェース間の相互変換（ゲートウェイ）を行う機能の実現を目指していた．しかし，現在は Java をベースにしながらも汎用的なソフトウェア部品化技術（OSGi プラットフォーム）の仕様を策定している．OSGi のメンバは 2004 年 10 月現在約 40 団体で，ホームオートメーション，ホームセキュリティ，情報家電制御に加え，自動車制御，いわゆるテレマティクスの領域でも利用されている．

2004 年 9 月には，国内の NTT，シャープ，東芝，日本 IBM，NEC，三菱電機の 6 社が「OSGi ユーザフォーラム Japan」を設立し，国内での普及を目指す動きが始まっている．以上の他に，白物家電を対象としたホームネットワーク規格を策定している ECHONET(Energy Conservation and HOmecare NETwork) コンソーシアム [15][6]，電話線を用いたホームネットワーク規格を策定する HomePNA[16][7]，電力線を用いたホームネットワーク規格を策定する HomePlug アライアンス [17] 等がある．

Jini　Jini[13] はパソコンや周辺機器，AV 機器，電話，家電製品など様々な機器をネットワークを通じて接続し，相互に機能を提供しあうための技術仕様．Sun Microsystems 社によって提唱され，IBM 社，Cisco Systems 社，Motorola 社，キヤノン，シャープ，ソニーなど多くの大手企業が支持している．Sun の Java 技術を基盤としているため，特定の OS やマイクロプロセッサなどに依存しない．また，Jini に対応した機器は，ネットワークにつなぐだけで複雑な操作や設定作業を伴うことなくすぐに機能する．Jini を利用するためには Java 実行環境である JVM(Java Virtual Machine) を搭載するか，ネットワーク上の他の機器が搭載している JVM を利用できることが必要になる．

サービス発見機構を構成するプロトコル

これらのプロトコルは，ネットワーク上に存在するデバイスを探し，利用するまでの準備を行う．これらのプロトコルは機器のアドレッシングと検出，機器の機能の記述とその公開，制御とイベント処理等の基本仕様を定めている．

6 ECHONET は日本で 1996 年より検討が開始され，ISO にも提案されている
7 Phoneline Networking Alliance: 1998 年に米国に提案され徐々に普及．日本では集合住宅向けの利用が始まっている

HTTP(Generic Event Notification Architecture)　Web サーバとクライアント (Web ブラウザなど) がデータを送受信するのに使われるプロトコル．HTML 文書や，文書に関連付けられている画像，音声，動画などのファイルを，表現形式などの情報を含めてやり取りできる．IETF によって，HTTP/1.0 は RFC 1945 として，HTTP/1.1 は RFC 2616 として規格化されている．

XML(eXtensible Markup Language)　XML[18] は文書やデータの意味や構造を記述するためのマークアップ言語の一つである．マークアップ言語とは,「タグ」と呼ばれる特定の文字列で地の文に構造を埋め込んでいく言語のことで，XML はユーザが独自のタグを指定できることから，マークアップ言語を作成するためのメタ言語とも言われる．もともと，同じく独自のタグを指定可能な SGML[19] のサブセットとして考案され，任意のデータを HTML と同様の感覚で送受信できることを目標に作成されたものである．XML はその性質上，他のマークアップ言語の骨組みとして使用されることが多い．

GENA(Generic Event Notification Architecture)　GENA[20] は， HTTP over TCP/IP およびマルチキャスト UDP を使用して通知を送受信する．

SOAP(Simple Object Access Protocol)　SOAP[21] は XML と HTTP などをベースとした，他のコンピュータにあるデータやサービスを呼び出すためのプロトコル．Microsoft 社や UserLand Software 社，Developmentor 社が中心となって開発された．SOAP による通信では，XML 文書にエンベロープと呼ばれる付帯情報が付いたメッセージを，HTTP などのプロトコルで交換する．サービスを利用するクライアントと，サービスを提供するサーバの双方が SOAP の生成・解釈エンジンを持つことで，異なる環境間でのオブジェクト呼び出しを可能にしている．

SSDP(Simple Service Discovery Protocol)　SSDP[22] とは HTTP ヘッダを拡張したシンプルなマルチキャスト・ディスカバリ・プロトコル．ディスカバリ・パケットには，XML 形式で記述されたデバイス・ディスクリプション・ドキュメントへのリンクが含まれています．デバイス・ディスクリプション・ドキュメントには，デバイス・タイプ，メーカ名，モデル名などのほかにも，UPnP Forum が定義したサービス・タイプを参照するための URL が記述されており，この URL を利用することで，サービス機能を XML 形式で詳しく記述したサービス・ディスクリプション・ドキュメントを取得することができる．

2.2.3　ユーザインタフェース機構

デバイスを連絵系させ利用するには，LAN 上で検出されたデバイスからユーザが目的とするデバイスを容易に選択し利用できることが，今後デバイスやサービスが増加するにしたがって重要になる．このユーザインタフェース機構として，ユーザに最適なデバイスを自動的に選択する機構とユーザが容易にデバイスを選択・操作できる機構について述べる．

自動デバイス選択機構

RFID(Radio Frequency Identification) や GPS(Global Positioning System) 情報，機器の種類等などを基に，ネットワーク上から自動的に適切な機器を検索する機構が提案されている．

STONE(Service Synthesizer on the Net)　STONE[23] は，デバイスが偏在する環境において，デバイスを利用する場合の透過性をネーミングの観点から捉え，必要なデバイスを自動的に連携させ目的のサービスを達成する機構である．このネーミングシステムは，ネットワーク上に遍在する様々な機能の透過的発見と接続によりサービスの透過的合成を実現するためにデザインされており，機能のインターフェースに着目した名前空間の集約管理と解決機構を備えている．STONE 上ではデバイスを「どのようなデータを受け付けるか」と「どのようにアウトプットするか」という 2 種類の機能で分類・認識する．

例えば，カメラはモノを撮影して MPEG-2 に変換するとする．そして MPEG しか受け付けないディスプレイがあったとする．従来の場合，カメラとディスプレイの間に PC が入って仲立ちし，変換や転送といった作業が必要となる．しかしながら，STONE というミドルウェアは，自動的にネットワーク中から画像を変換してくれる機能を，ネットワーク上で探し出し自動的にその間にいれる．仮に変換機能を持つものがなくても，2 つの機能を組み合わせることで解決を図る．つまり，ユーザーが，途中の作業や情報を調べる必要がなくなるのである．ユーザは結果のみを示すだけでよい．

AMIDEN アーキテクチャ(Architecture AMIDEN)　AMIDEN アーキテクチャ[24] とは，家電製品を機能単位で分け，それをネットワークで接続し，サービスを提供する機構である．家電製品がネットワーク化され機能が増加すると，操作スイッチなどが増加し，利便性の面で問題が生じる．AMIDEN アーキテクチャは，家電をコンセントに接続するだけで自動的にネットワークに接続し，ユーザが要求するサービスを複雑な操作を行うことなく提供することを目的としている．また，ネットワークを介して自動的に機器の連携を行うことも実現する．この AMIDEN アーキテクチャを備えた情報家電が実現すると，コンセントに電源を差し込むだけで自動的に機器の協調動作が可能となるため，例えばテレビをつけると自動的に照明が切り替わりテレビと協調して演出効果を出すようなシステムが開発可能となる．この際に複雑な設定を行うことなく協調制御を実現できる点が AMIDEN アーキテクチャの優れた点である．

EZ DEV　環境に多くの機器が存在する場合，ユーザにとってネットワーク上で発見したサービスが実世界のどのデバイスに該当するかの認識が困難になる可能性がある．そこで，EZ DEV[25] は各デバイスに LED を付け，ユーザへの通知を行い，ユーザの実世界におけるデバイスの発見・認知を支援する．ユーザがある環境に入った際，EZ DEV はユーザの周囲にあり利用可能なデバイスの LED を点滅させる．また，ユーザは携帯端末を操作し，サービスに対応するデバイスの LED を点滅させられる．さらに EZ DEV は，あるユーザがデバイスを使用して他のユーザをブロックしている際も LED を点滅させる．

EZ DEV はユーザの携帯端末に GUI を提供し，ユーザの周囲にあり利用可能なサービスをアイコンとして表示する．GUI 上の表示アイコンは各サービスごとに異なる．ユーザはアイコンにファイルをドラッグアンドドロップすると該当サービスを実行できる．

Follow-me application　Follow-me application[26] は，ユーザの位置を超音波を用いたセンサにより取得し，自動的にユーザに最も近いデバイスを選択する．これは，ユーザがどこにいてもサービスアプリケーションが，ユーザを追尾しサービスを提供することを行っている．

実空間ユーザインタフェース機構

ユーザが利用するデバイスを選択する際に，ディスプレイ上で選択するのではなく，直接実空間上でデバイスを選択，操作するための研究が行われている．多数の機器が存在する環境では自動決定には限界があり， ユーザが望んでいる機器を直接指示できる手段も必要と思われる．

Touch-And-Connect　Touch-and-Connect[1] は無線機器間の接続を，接続したい両機器のボタンを押すだけで直接的に指示できる手法である．無線機器間の接続の際には，一般にアドレスや名前の指定などの煩雑な設定が必要となる．この手法では，状態を表示可能なボタンを使用し，複数の人間が独立に操作を行う状況においても誤接続を防止する．ブロードキャスト通信を用いた本手法のプロトコルは，管理サーバを必要とせず，動的な端末の入退出にも対応するため，必要に応じて一時的に構築されるアドホックネットワークでも利用可能である．また，グループの導入によりセキュリティと使いやすさの向上を行っている．

また，この機構ではボタンの状態表示を工夫することにより，各機器に対してボタン１つのみで本手法を実現している．ボタンに，接続元指定，接続元指定成功，接続先指定，接続先指定成功，無効，の 5 つの状態を設ける．ユーザは，まず，接続元指定状態のボタンを押して接続元機器を指示し，ボタンが接続元指定成功状態になったことを確認する．その後，同様に接続先指定状態のボタンを押して接続先機器を指示する．各デバイスに実装された様子を図 2.2 に示す．

tranSticks　tranSticks[3] は，実装面から見ると，ユーザと計算機の双方が，組になった他のメディアを容易に識別できるように設定されたメディアである．ユーザのためには同じ色，形，印などを付けておく．計算機のためには，計算機が読み取れる形の ID を持たせておく．tranStick が差される機器は，tranStick が差されて ID が読み取れる状態で，ユーザのための識別部分がユーザに見えるようになっていたほうが望ましい．図 2.3 に tranStick を示す．ユーザには，組になったメディアどうしが，あたかも空間を越えて繋がっているかのようなインタフェースを提供する．この機能として

- 繋がったメディアが差されている機器と通信できる
- 繋がったメディアとデータを共有している

という二種類の基本的な機能が提供する．

この機能を実現するために，tranStick を検出したデバイスはディレクトリサーバを経由して，同じ ID を持つデバイスを探す．デバイスが見つかると ID のマッチングを行い，ローカル Proxy を経由して通信を行うことが可能となる．

tranStick と同系統の研究として，カメラでデバイスを写すことにより ID を読み取り接続を行う Gaze-Link[27] や，デバイス同士を近づけることで接続を行う FEEL[28] がある．

u-photo　u-Photo[29] は写真撮影によって，写っている情報家電やセンサの情報を取得する環境情報スナップショットである．撮影したデジタル写真は，情報家電やセンサ上のアプリケーションを起動する視覚的なコンソールとなる．ホームネットワーク環境おいては数多くの情報家電やセンサが配置されると予想されるが，それら実際の機器と，それらのネットワーク上の情報をどのように結びつけて取得するかが課題である．u-Photo では，情報家電やセンシングエリアを写真に取ることで，それらのネットワーク上のアプリケーション情報をデジタル写真上に GUI[8] として付加し，

8 Graphical User Interface

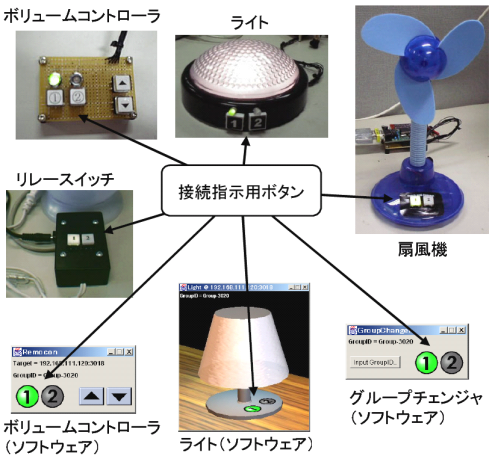

図 2.2: ToucnAndConnect

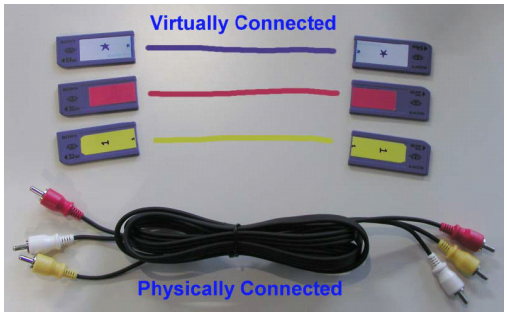

図 2.3: tranSticks

そこからすぐに情報取得や，アプリケーション起動が実現できる．加えて，ビデオの再生中といった動作中のアプリケーションの状態も写真の視覚情報とともに取得し，u-Photo を通じて他の情報家電への同じ作業の再現といった複数機器間の協調動作も実現できる．図 2.4 に画面中の写真に写っているプリンタへファイルをドラッグ・アンド・ドロップすることで印刷する様子を示す．

InfoPoint　InfoPoint[2] は，ディスプレイ上でアイコンをドラッグ・アンド・ドロップして利用するように，実空間上でデバイスからデバイスへドラッグ・アンド・ドロップし，デバイスを利用する機構である．各デバイスへ 2 次元バーコードを貼り付け，読み取り用 CCD カメラを搭載したドラッグ・アンド・ドロップを行う専用のコントロールデバイスを用いて操作を行う．操作は，入力元選択，移動，出力先選択，動作の選択，動作の実行に分かれ行われる．

各デバイスと専用のコントロールデバイスは，ネットワークに接続され，InfoPoint Manager を介して共通のデータベースへアクセスする．コントローラデバイスによる操作は InfoPoint Manager でデータベースへの情報蓄積され，選択されたデバイスへ通知される．図 2.5 に，ドラッグ・アンド・ドロップでプロジェクタに表示されているスライドを PC に転送している様子を示す．

同系統の研究として，ディスプレイから他のデバイスのディスプレイ上へ，ドラッグアンドドロップを行い，ファイルを転送することが可能な Pick-And-Drop[4] がある．

MultiNavi　ユーザの人影をカメラで上面より撮影し，画像処理を行うことによって，ユーザが腕で指し示している方向を入力する手法 [30] が提案されている．これを応用すれば，特別なデバイスを持たなくても，腕で指し示すことにより遠く離れた機器を指定可能なシステムが実現できる．しかし，腕の上下方向（ピッチ角）の角度入力が行えないため，多数の機器が存在する場合には詳細な指定が困難である．

Airreal　AirReal[31] は，レーザポインタを内蔵したリモコンで機器を指し示すことにより，様々な指示を行うことができる．特に，複数の機器を指し示すことによって，それらの機器間を接続し連携させることができる．機器に接触する必要がないため，遠く離れた機器を指定するために AirReal は有用である．しかし，事前に管理 PC に機器の位置を入力しておく必要がある．

図 2.4: u-Photo

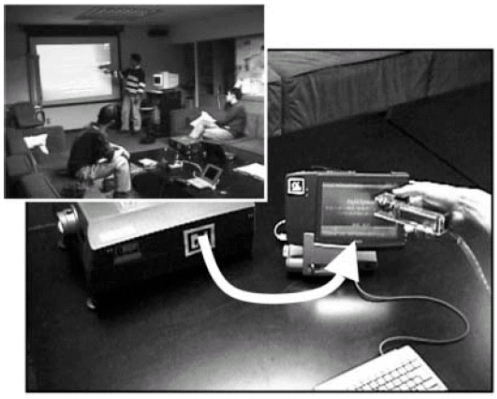

図 2.5: InfoPoint

2.3　通信保護機構

共通物理インフラ使う LAN 上でデバイスを連携しサービスを安全に利用するためには，認証と通信の保護が必要となる．

2.3.1　暗号化による通信の保護

情報セキュリティ

インターネットの利用は，当初研究者同士の間の情報の交換に用いられたが，その利用は企業内での業務効率の改善やコミュニケーションの手段として，新しい社会基盤となっている．特に電子商取引では，全ての情報がデジタル化され，実世界と異なる本人性の確認，データの改ざん検知が必要となり，情報セキュリティが重要になってくる．

従来，セキュリティ技術というと，軍事関係者や外交関係者などの一部の特殊な任務を持った関係者がその存在さえも否定した状況で，情報の秘匿を目的に利用していた．しかし，1976 年米国でウィットフィールド・ディフィー（Whitfield Diffie）とマーティン・ヘルマン（Martin Hellman），および 1978 年，ラルフ・マークル（Ralph Merkel）の 3 人によって公開鍵暗号のアイディアが発表された．アルゴリズムを公開する，鍵の一部を公開するというもので，これにより暗号化の技術が一般市民の手に入ってことと，デジタル社会の基盤になる基本技術となったことを意味する．

情報セキュリティとは,「偶発，故意にかかわらず，不正あるいは好ましくない破壊，改ざん，漏洩を，あるいは情報及び情報資産の使用を防止し，そこから復旧することである」と定義できる [32]．このセキュリティの確保には，

- 機密性 (Confidentiality):情報を不適切な人には絶対見せないようにすること．
- 完全性 (Integrity):情報が常に完全な状態で保たれ，不正によって改ざんされたり破壊されたりしないこと．
- 可用性 (Availability):ファイルやネットワーク上で保存されている情報や情報システムをいつでも利用できること．

が挙げられている．このための技術要素として，通信を秘匿する暗号化技術，本人性の確認や資源へどの程度のアクセスを許可するかを管理するアクセス管理技術 (本人認証，アクセス制御) が挙げられる．また，第三者以外の不正に対して，証拠性，原本性の確保も挙げられている．

情報セキュリティ技術のフレームワークは，大きく，基盤技術，応用技術，応用システム，コンプライアンスの 4 つから構成される．基盤技術として，暗号化技術，暗号化プロトコル，ハードウェアへの実装技術，モバイルセキュリティが挙げられる．応用技術として，バイオメトリクス技術や著作権保護，IC カードセキュリティ，コンピュータウィルス，ファイアウォール・VPN 技術がある．さらにこれらの技術を利用して公共システムを構築する応用システムや，セキュリティの運用や評価，また法体系を整えるコンプライアンスが必要とされる．

今日における具体的な脅威としては，インターネットを介して感染するウィルス，ワーム．サーバへの不正アクセス，無線 LAN を経由した LAN への不正アクセス，フィッシング詐欺など挙げられる．

暗号化プロトコル

情報セキュリティを確保するためには，数学を基礎とする暗号化技術をはじめ，社会システムに至るまで大きなフレームワークが必要である．ここでは，サービスを安全に利用するとは，サービスが行う通信を保護することして，通信のセキュリティにターゲットを絞り，通信内容を漏洩しないこと，通信内容を改ざんされないことの機密性と完全性を確保するための暗号化プロトコルについて述べる．

SSL(Secure Socket Layer)　SSL[5] は Netscape Communications 社が開発した，インターネット上で情報を暗号化して送受信するプロトコル．現在インターネットで広く使われている WWW や FTP などのデータを暗号化し，プライバシーに関わる情報やクレジットカード番号，企業秘密などを安全に送受信することができる．

SSL は公開鍵暗号や秘密鍵暗号，デジタル証明書，ハッシュ関数などのセキュリティ技術を組み合わせ，データの盗聴や改ざん，なりすましを防ぐことができる．OSI 参照モデルではセッション層 (第 5 層) とトランスポート層 (第 4 層) の境界で動作し，HTTP や FTP などの上位のプロトコルを利用するアプリケーションソフトからは，特に意識することなく透過的に利用することができる．SSL 3.0 をもとに若干の改良が加えられた TLS 1.0[6] が RFC 2246 として IETF で標準化されている．

なお，SSL による暗号通信で送受信されるデータの暗号化・復号化を高速に行なう専用ハードウェアもある．

TLS(Secure Socket Layer)　TLS[6] は，インターネット上で情報を暗号化して送受信するプロトコルの一つである．現在インターネットで広く使われている WWW や FTP などのデータを暗号化し，プライバシーに関わる情報やクレジットカード番号，企業秘密などを安全に送受信することができる．TLS は公開鍵暗号や秘密鍵暗号，デジタル証明書，ハッシュ関数などのセキュリティ技術を組み合わせ，データの盗聴や改ざん，なりすましを防ぐことができる．OSI 参照モデルではトランスポート層 (第 4 層) にあたり，HTTP や FTP などの上位のプロトコルを利用するアプリケーションソフトからは，特に意識することなく透過的に利用することができる．TLS 1.0 は Netscape Communications 社が開発した SSL 3.0 をもとに若干の改良が加えられたものである．

IPsec(Security Architecture for Internet Protocol)　IPSec[33] はインターネットで暗号通信を行なうための規格である．IP のパケットを暗号化して送受信するため，TCP や UDP など上位のプロトコルを利用するアプリケーションソフトは IPsec が使われていることを意識する必要はない．現在インターネットで使われている IPv4 ではオプションとして使用することができるが，次世代の IPv6 では標準で実装される．

WEP(Wired Equivalent Privacy)　無線通信における暗号化技術．無線通信は傍受が極めて容易であるため，送信されるパケットを暗号化して傍受者に内容を知られないようにすることで，有線通信と同様の安全性を持たせようとしている．

RC4 アルゴリズムをベースにした秘密鍵暗号方式で，IEEE によって標準化されており，IEEE 802.11b のセキュリティシステムとして採用されている．

秘密鍵に 40bit のデータを使う旧来の方式と，128bit のデータを使う新方式とが存在するが，WEP そのものに様々な脆弱性が発見・報告されており，暗号化技術としては既に低い信頼性しか持ち合わせていないと言われている．

PGP(Pretty Good Privacy)　PGP[34] は電子メールを暗号化して，安全に送受信できるようにするしくみの 1 つである．PGP はさまざまなプラットフォームに対する移植が早期から行われたこと，メールを送受信するメーラーソフトウェアがこれに対応するだけで簡単に使えることなどから，一部のユーザーの間で普及している．

PGP では，電子署名としては RSA[35] を用い，その電子署名とメール本文を IDEA（International Data Encryption Algorithm）と呼ばれる共有鍵暗号方式で暗号化する．そして IDEA の鍵を RSA によって暗号化する．メール本文の暗号化に IDEA を用いるのは，比較的処理の軽い IDEA でメール本文の暗号化，復号化を行うことで，処理速度の向上を図るためである．RSA による公開鍵は，送信相手にあらかじめ渡しておく必要がある．

PKI(Public Key Infrastructure)　公開鍵暗号を用いた技術・製品全般を指す言葉．RSA や楕円曲線暗号などの公開鍵暗号技術，SSL を組みこんだ Web サーバ/ブラウザ，S/MIME・PGP などを使った暗号化電子メール，デジタル証明書を発行する認証局 (CA) 構築サーバなどが含まれる．

PKI を用いるには社会的に信頼される大規模なインフラを構築する必要があり，認証局から認証される各サーバへの課金も生じている．

2.3.2　仮想通信網による通信保護機構

通信全体の保護

暗号化プロトコルは，各サービスに付随して用いられることが多い．SSL と HTTP を組み合わせた HTTPS や FTPS，POPS，IMAPS など，各サービスにおいてそれぞれ規格化された暗号化プロトコルを用いている．このような利用方法は，通信の漏洩と改ざんを防止するという通信保護の目的を果たしているが，実装されているのは上記で述べたような一部のサービスだけである．そのため，ファイル共有プロトコルなど既存のサービスは，通信保護が行われていないものも多々存在する．

通信の保護は，重要と考えられるサービスのみに提供すれば良いという考えた方も存在するが，既存のサービスに暗号化プロトコルを作業は手間であり，サーバソフトやクライアントソフトの

アップデートなど利用者にとっても負荷が掛かる．また，ウィルスやワームを遮断するために，サービスごとではなく通信そのものを遮断したい場合 (検疫ネットワーク) などがあり得る．そこで，サービスアプリケーションとは独立して通信を隔離，保護することが行われている．

サービスアプリから独立した通信保護とは，OSI モデルでの第 2 層，第 3 層，第 4 層などの間に，通信保護を行う層を入れるものである．上位のサービスアプリケーションからは，従来の通信プロトコルと同様に扱えるため，各サービスに暗号化プロトコルを追加することなく，通信の保護が実現できる．SSL に関して，本来の通信プロトコルの下位プロトコルとして動作する点では同様であるが，これらは各サービスで実装されている．これに対し，サービスから独立した通信保護機構は，各サービスが動作するプラットフォーム (OS) のプロトコルスタックや独立したアプリケーションとして実装され，各サービスアプリケーションからは通信インタフェース (NIC) や Proxy として認識される．IPSec は拠点間を結ぶ VPN を構築するための技術としても用いられることが多い．

仮想通信網プロトコル

VLAN(Virtual LAN) 検疫ネットワーク　LAN において，物理的な接続形態とは独立に，端末の仮想的なグループを設定するこである．LAN スイッチと呼ばれる機器の機能を利用して，端末の持つ MAC アドレスや IP アドレス，利用するプロトコルなどに応じてグループ化する．端末を物理的な位置を気にすることなくネットワーク構成を変更することができ，また，端末を移動しても設定を変更する必要がないというメリットがある．

この仕組みを利用して，検疫ネットワークが構築されている．検疫ネットワークとは，社内 LAN に接続しようとしたコンピュータを，いったん，LAN とは隔離されて存在する検査専用のネットワークに接続し，コンプライアンス検査を行い，問題がないことを確認してから社内のネットワークへの再接続を許可する仕組みのことである．問題のあるコンピュータは，対策を施さない限り社内のネットワークに接続できないので，前述のような出先で感染したノート PC などから社内 LAN を守る有効な対策としてにわかに注目を浴びている．

VPN (Virtual Private Network)　VPN とは，通信インフラとしてインターネットを利用しながら，ポイント to ポイントの専用線接続を可能にする技術．またはこの技術を利用して構築されたネットワークである．従来，遠隔地にある企業のコンピュータ同士を接続する場合には，専用線を利用したり，公衆回線を利用してダイアルアップ接続するのが一般的であった．しかし専用線では接続地点間の距離と通信速度に応じて通信料がかかり，公衆回線では接続距離と接続時間に応じて通信料がかかる．利用条件によって料金は異なるが，特に遠隔地を結ぶ場合，この際の通信料金はかなりの額になる．

これに対し，グローバルネットワークであるインターネットを利用すれば，接続距離とは無関係に遠隔地のコンピュータ同士を接続し，データ交換を行うことができる．ただしインターネットで利用する TCP/IP プロトコルは，データの暗号化や認証などは前提になっていないため，経路途中での情報漏洩や改ざんなどの危険性があり，機密性の高い情報を交換するのは不可能だった．

VPN は，送出側でデータを暗号化し，受信側でこれを解読することで，経路となるインターネットでは暗号化されたデータを送受信し，暗号化／解読を双方のゲートウェイで透過的に行うことで，あたかも専用線による LAN 接続がなされているような構成を可能にする．

この VPN を実現するために，多くのプロトコルが規定されている．グローバル規模においてリンク層での VPN を実現できる MPLS(Multi-Protocol Label Switching) や，Ethernet over IP によりリンク層の仮想ネットワークを構築できる L2VPN，PPP 上で L2VPN を実現する L2TP(Layer 2

Tunneling Protocol)，PPTP(Point-to-Point Tunneling Protocol)．また，IP 層の仮想ネットワークを構築する IPSec を用いた IP-VPN，クライアント側に特殊なソフトを必要としない VPN-SSL などがある．

Secure LAN　Secure LAN[36] は，情報の漏洩を防止するために，LAN 内における通信を全て暗号化する．このために，NIC(Network Interface Card) に暗号化チップを備える．なお，暗号鍵は LAN 内の全端末で同一であり，正規の端末が行う不正行為について防止することは出来ない．なお，外部へ接続される LAN にはゲートウェイが設定され，LAN 外への通信はゲートウェイで復号化され，また LAN 内への通信はゲートウェイで暗号化される．

また，通常の LAN において一部のサーバとクライアントのみにこの暗号化 NIC 機能を備え，鍵を持ったユーザのみサーバと通信できる製品もある．

MyNetSpace(MNS)　MNS[37] は，サービス主導の柔軟な仮想ネットワークを構築する技術である．MNS はユーザが柔軟に端末のグループを作り出し，それに基づいてサービス間における通信制御を行う．MNS は各々のユーザが定義する閉域グループで，MNS に参加しない端末との通信を許可しない．また参加認証機能をもち，ユーザが所有する端末や特定の部屋にある端末といった条件を満たさなければ端末は参加することができない．一方で，端末は条件を満たせばこれら複数の MNS に同時に参加できる．サービスと各 MNS を結びつけることによって，サービスは閉域グループ内で通信を行うことができる．

2.4　おわりに

本章では，サービスを容易かつ安全に利用するための既存研究・規格について，ネットワークサービスと通信保護機構の 2 つの観点からそれぞれ述べた．

ネットワークサービスでは，従来グローバル規模のコミュニケーションを目的としたサービスから，近年では LAN 内でのデバイスを連携させ利用するサービスが増加していることについて述べ，それらを利用するための UPnP や DLNA といった通信規格を紹介した．また，これらは相互接続性など通信とは別に，ユーザがより容易にサービスを利用できるためのユーザインタフェースについて取り上げ，特にデバイスを自動的に選択する機構や直感的にデバイスを選択できる実空間インタフェース機構について述べた．

通信の保護については，暗号化プロトコルは規定されているものの，各サービスごとの実装では普及に不十分であるとし，サービスとは独立した仮想通信網による通信の保護手法を述べた．

今後は LAN 上にデバイスがさらに増加すると予想され，よりユーザが容易かつ安全にデバイスを連携させ利用できる仕組みを整える必要があると考えられる．

第3章

既存技術の問題

3.1　はじめに

第 2 章では，LAN 上に増加するデバイスと，ユーザが容易かつ安全にデバイスを連携させを利用するための研究・技術について述べた．本章では，それらが実現できているかというの考察と至らない問題点を挙げる．

3.2　サービス発見機構とユーザインタフェース機構の連携

3.2.1　サービス利用における現状

デバイス間の通信に関しては既に規格団体が設立され，Windows OS に標準搭載されたり，情報家電として商品に組み込まれるなど，ユーザへの普及を目指して規格化が進められており，既に実用段階へ到達している．

一方で，ユーザインタフェースに関しては未だ標準化などは行われていない．研究としては，自動的にデバイスを選択・利用できる手法や，デバイスからデバイスへオブジェクトを移動する手法などがいくつも行われているのは第 2 章で述べたとおりである．しかしながら，実際には研究段階では実空間インタフェースによる操作が実現されているものの，実際の製品では各ベンダごとにユーザインタフェースを独自に作成しており，実空間インタフェースはまだ組み込まれていない．各ベンダはテレビなどの大画面で GUI[1] を駆使して，ユーザが分かりやすく操作できるようユーザインタフェースの開発に努めている．

デバイス間の通信規格が標準化へ急速に進んでいる一方で，ユーザインターフェースに関する技術や仕様が定まらず，各ベンダの自由に任せているには大きく以下の 2 つが理由として考えられる．

- ユーザインタフェースはデバイス間の通信を行う際に必須ではないこと
- 研究段階で通信規格との連携が図られていないこと

まず，大きな原因として，ユーザインタフェースはユーザが容易にサービスを利用する際に重要な技術ではあるが，デバイス同士の相互互換性など実際に通信を行いサービスを提供するときに必須ではないことが挙げられる．通信規格は，いくつものベンダが協力し統一しなければ，サービスを提供することもままならず，情報家電やネットワーク対応デバイスを開発する上で必須である．これに対して，ユーザインタフェースはベンダごとに操作方法が異なる場合でも，ユーザがそれぞれの操作方法を覚えることでデバイスの利用は可能である．

もう一つの理由は，ユーザインタフェースの研究が操作性を追求する余り，通信規格との連携を行わず，インタフェース機構にとって都合の良い独自のデータベース形式や通信方式を採用している点である．デバイス間の情報のやりとりや指示はベンダ間の相互接続性が必要とされ，そのために標準化が進められている．しかしながら，研究ではドラッグ・アンド・ドロップを実空間で実現するためのデータベース構造など，標準化された通信規格とは異なる手法により実現している．このため，既に標準化が行われているデバイス間の通信規格とユーザインタフェース機構との乖離が大きく，ユーザインタフェース機構を導入することが困難である．

1 Graphical User Interface

3.2.2　サービス発見機構に対するユーザインタフェース機構の位置づけ

ユーザがデバイスを連携させ利用するにあたり，操作性に大きな影響を持つユーザインタフェース機構は，今後デバイスが増加し操作性が悪くなるに従い重要になる機構である．本来であれば，両方の必要な機能を持ち寄り，規格を定めることが望ましいが，現状でデバイス間の通信規格を定めているサービス発見機構の標準化が先行している．そのため，相互接続性を実現したサービス発見機構をベースとして，ユーザインタフェース機構を連携させ適応させることが妥当であると思われる．

サービス発見機構とユーザインタフェース機構の研究で定められていることをそれぞれ挙げる．

サービス発見機構については，

- 物理ネットワーク
  Ex. Ethernet, Bluetooth, IEEE802.11
- 通信を行うプロトコル
  Ex. TCP/IP, HTTP, XML
- サービス記述
  Ex. SOAP, SSDP
- 発見，制御，イベント処理
  Ex. SSDP, GENA

など，通信に関わる部分と，サービス記述，及びサービスを制御するためのプロトコルが定められている．

ユーザインタフェース機構の研究については，

- インタフェース
  Ex, GUI on Screen，PDA, 指示デバイス
- 制御手法
  Ex, プッシュボタンの組み合わせ，指示デバイスによるドラッグ・アンド・ドロップ，レーザポインタでの指定，デバイスの挿入
- ユーザ操作を支援する付加情報
  Ex, ユーザの位置情報，デバイスの位置情報
- サービス記述
  Ex, ファイルコピー，プリンタ
- 制御処理
  Ex, TCP 接続，ファイルコピー

など，主にインタフェースとその操作方法，それらに付随する制御処理である．サービス記述やそれに基づいた制御処理などは，各研究ごとに独自に定められている．

サービス発見機構とユーザインタフェース機構を比較すると，サービスの記述と制御処理に関して重複してることが分かる．通信規格はプロトコルが定まっているのに対し，ユーザインタフェースは，制御手法にとって都合良い独自のサービス記述や制御処理を行っている．このため，サービスプロトコルをベースとしてユーザインタフェースを適用させるんは，サービス記述と制御処理を通信規格のものを用いるユーザインタフェースが求められる．

3.2.3　ユーザインタフェース機構に求められる機能

既に標準化が進んでいるサービスプロトコルに対し，ユーザインタフェースは独自のサービス記述や制御処理を行うことなく，連携して操作を行うことが望ましい．ユーザインタフェースの機能とは，ユーザが容易にサービスを選択でき，その動作を制御することが目的である．この操作に特化した機構は，通信規格を大きく拡張する必要が生じ，標準化との乖離が大きくなる．

ユーザインタフェースの改善が必要とされるのは，ユビキタス社会へ進展するに伴い，LAN 上に膨大な数のデバイスやサービスが存在することが原因である．デバイスやサービスの列挙を行い，そのリスト中からユーザが目的とするデバイスを選択するとは困難である．特に目に見えないサービスを除き，ユーザは実空間上のデバイスを意識している場合が多く，これをネットワーク上の名前と対応づけを行う作業はユーザの負荷となり得る．サービス記述の情報量を増やすことも可能ではあるが，例えば“ あのプリンタ ”，”あのスピーカ ”といったユーザからの相対的な位置情報に基づくデバイス選択が困難なことに変わりない．つまり，ユーザインタフェースとして現在求められているのは，実空間上で直感的にデバイスを選択することであると考える．

また，選択されたデバイスやサービスが，何を行うかどうのような制御を必要とするかは，デバイスの提供するサービスや相互接続性に大きく影響する部分であり，サービス発見機構に依存するところが大きい．そのため，ユーザインタフェースはサービス発見機構で交渉が行われた後に，ユーザに操作を反映させるために用いられる．このサービスの動作に関する操作は，実空間上での特殊な操作ではなく汎用的な操作が必要とされる．

サービス発見機構をベースとしてユーザインターフェスに必要とされる機能は

- 実空間上でのデバイス選択操作
- サービスの動作に関する選択操作

の 2 つに分けて考えることが出来る．特に，膨大なデバイスが存在する環境では実空間でのデバイス選択が重要な課題であると考え，まず実空間操作でデバイスを選択・利用できる機構を検討する．

3.3　仮想通信網による通信保護

3.3.1　ユーザ主導の動的な仮想通信網

デバイスやサービスを安全に利用するための通信保護は，UPnP などのサービス発見機構にも備わり，また SSL/TLS を用いて各サービスアプリケーションにも実装が行われている．しかしながら，第 2 章で述べたように既存サービスアプリケーション全てに実装することは，実装を行うベンダもアップデート作業を行うユーザにとっても付加となり得るため，サービスアプリケーションとは独立した通信保護機構を実現することが望ましい．

サービスアプリケーションと独立した通信保護機構のための仮想通信網を構築するための技術として第 2 章では，VLAN をはじめ VPN のための技術を紹介した．しかしながら，これらの技術は各ネットワーク機器やサーバに仮想通信網を構築するための設定が必要であり，これらの作業は管理者が行う必要がある．例えば，タグベース VLAN では VLAN ID を衝突しないよう割り当て，各デバイスに設定する必要がある．ポートベースでも同様にスイッチに VLAN ID を割り当てる作業が必要となる．VPN もサーバの設置やルーティングの設定，認証の設定を行う必要があり，さらに膨大な数のデバイスがサーバを介して通信を行うとサーバの負荷が高くなってしまう．

サービスごとに仮想通信網を構築し，通信を保護しようとするには，ユーザが動的に多数の仮想通信網を構築できる必要がある．従来の仮想通信網構築技術では，これを満たさない．そのため，ユーザにより動的にデバイス間で多数の仮想通信網を構築する技術が必要となる．

3.3.2　サービスの認証

LAN 上に存在するデバイスは，プリンタやスピーカなど共用のデバイスもあるが，MP3 プレイヤやディスプレイ，キーボードなど個人所有のデバイスも多々あることが予想される．この個人のデバイスを，他人が勝手に用いることがないようデバイスの利用の際には認証が必要となる．また，個人的なデバイスに限らずデバイスの詐称や盗聴・妨害を防ぐため，連携を指定されたデバイスのみを接続可能なようにする必要がある．

従来，通信保護における認証はサーバに認証情報を保持し，接続する際にパスワードなどの入力により認証を行っていた．しかしながら，LAN 内における複数のデバイスの連携では各デバイスは自律的に動作し，複数のデバイスと連携する．つまり，1 つのサーバに認証情報を保持しておくといった認証情報を集中管理することができない．

また認証作業は，具体的に事前に登録したユーザのパスワード等を利用の際に入力し確認する作業であり，現在では通信保護と同様に各サービスアプリケーションごとに実装されている．特に認証は，現段階で個人のデバイスが多くないことから，実装されていないことが多い．また，膨大な数のデバイスへパスワード等の設定や，利用の際の入力はユーザにとって使い勝手を悪くする一因にもなり得る．

以上をまとめると，LAN 上のデバイスを安全に連携させるためにの認証には

- ユーザへのデバイス自体の利用を認証するもの
- 連携するデバイスが本当にユーザの意図したデバイスであるか確認するもの

の 2 つがあると考えられる．1 つは，利用しようとするユーザが，そのデバイスを使って良いかどうかの認証を行う．2 つ目は，ユーザ本人は関係なく連携するデバイスが正しい物かどうかその正当性を認証するものである．例えば，連携を要求した先のデバイスが，本当に意図したデバイスであるか，第 3 者によって詐称されたデバイスではないかを確認しなければならない．SSL/TLS でも 2 つめの正しいデバイスを確認するために PKI といったインフラを利用する．しかしながら，膨大な数のデバイスがある環境では PKI の適用は困難である．

3.4　おわりに

本章では，既に標準化が進んでいるサービス発見機構と，研究は行われているものの実用化が遅れているユーザインタフェース機構についてとりあげ，必須である通信規格を持つサービス発見機構とユーザインタフェース機構の連携が必要であると述べた．特に，実空間で直感的にデバイスを選択できるユーザインタフェース機構は，今後デバイスが膨大に増加する中で重要な機能だと考え，この機構をサービス発見機構と連携することを検討する．

通信保護機構について，仮想通信網による通信保護には，ユーザ主導の動的で多数の仮想通信網を構築する手法が必要であると述べた．また，連携するデバイスの正当性を確認する必要性や，個人所有のデバイスが増加することが予想されデバイスを利用する際に認証が必要であることを述べた．

第4章

実空間操作に基づくリンク層におけるデバイスグルーピング

4.1　はじめに

第 3 章で述べたように，安全かつ容易なデバイス連携を行うにはには 2 つの問題点がある．

1 つ目は連携するデバイスを容易に選択するための機構である．これについてはデバイスが増加した場合も有効な実空間操作による直感的なデバイス選択・操作を行うための研究が行われている．しかしながら，これらの研究は実空間上の操作を実現しているものの，デバイスが連携するためのプロトコルはそれぞれの実空間操作に適した独自の形式を用いている．そのため，デバイス間の相互接続性を考慮した既存サービス発見機構との乖離が大きく，導入することが困難である．

2 つ目はデバイス間の通信を保護するための機構である．デバイス間の通信は，インターネットを介し様々なサーバと接続する PC の通信と異なり，専用ケーブル内に閉じた安全なものであった．その後，利便性ゆえに共通物理インフラの LAN 上で通信を行うようになった．今後も，LAN に対応するデバイスが増加し，その結果デバイスが行う通信を他のデバイスから盗聴，妨害されることが問題となることが想定される．例えば，キーボードの入力を盗聴されることは問題である．そのため，デバイス連携の通信を閉じたものとし，保護することが重要である．通常，通信を保護を行うためには，SSL/TLS によって通信を暗号化することが行われている．これらは各プロトコルごとに暗号化の実装を必要とし，またコネクションごとに処理が行われる．そのため，複数のプロトコルを用い，複数の相手と同時に通信するデバイス間の通信においては適さない．

本章では，これらの問題を解決するための実空間操作に基づくリンク層でのデバイスグルーピング機構について述べる．このデバイスグルーピング機構は，既存サービス発見機構と実空間操作の連携，及びデバイス間の包括的な通信の保護を実現するものである．

4.2　ViCon 設計

4.2.1　ViCon 動作概要

実空間操作と既存サービス発見機構の連携，及びデバイスグルーピングよる通信保護に着目し，実空間操作に基づいてリンク層でデバイスグルーピングを行い安全な仮想通信網を構築することで，実空間で選択したサービスを安全に利用できる機構“ ViCon ”を提案する．この機構は，実空間操作で指定されたデバイスの間で安全な仮想通信網を構築し，この通信網上でサービス発見と発見後の通信を行う．

この仮想通信網は，実空間操作を既存サービス発見機構へ反映させることと，通信を保護することの 2 つ働きを併せ持つ．

1 つ目は，既存のサービス発見を仮想通信網上で行うことで，サービス発見機構へ変更を加えることなく，サービス発見の対象となるデバイスが，実空間上で選択されたデバイスへ限定される．仮想通信網は，サービス記述やデバイス間の連携プロトコルについて定めておらず，既存サービス発見機構が通信できるデバイスの範囲を限定する．このデバイスを限定する範囲は，実空間操作によって操作されている．これは，第 3 章で述べたように，デバイス選択を既存サービス発見機構へ伝えるだけの必要最小限の機能に止まっている．そのため，UPnP や DLNA など既に標準化が進められている既存サービス発見機構と連携することが可能である．

2 つ目は，サービス発見後，サービスの通信が仮想通信網上で行われるため，通信の保護が行われることである．連携するデバイス間で構築された仮想通信網は暗号化によって保護されており，その上で行われる通信は全て保護される．仮想通信網は，デバイス間に閉じており他のデバイスからの盗聴や妨害を防止することが出来る．この通信保護は，SSL/TLS などと異なり複数のプロト

コルや複数のデバイス間でも，容易に適用することができる．また連携するデバイス間の認証については，実空間からの渡される情報を基にして行う．

なお，従来のサービス発見機構を仮想通信網上で動作させることを考慮し，仮想通信網はリンク層で構築する．

動作概要を図 4.1 に示す．(1) まずポインタのような指示デバイスを用いて実空間上でデバイスを選択する．次いで，(2) 選択されたデバイス間で暗号化された安全な仮想通信網を確立し，その通信網上で (3) サービス発見と (4) サービスのための通信を行う．なお，実空間操作を使用しない場合は，ステップ 1，2 を省略し従来のサービス発見を行っても構わない．

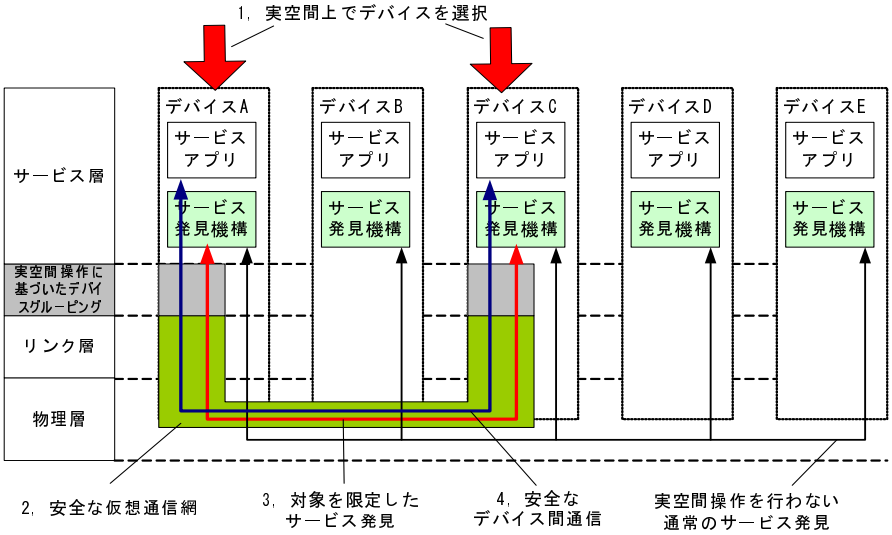

図 4.1: 実空間操作に基づくデバイスグルーピング

4.2.2　実空間での指示デバイス

実空間で操作を行うには指示デバイスが必要である．この指示デバイスには，現在提案されている手法では通信の方向によって大きく分類できる．以下のようなデバイスが挙げられる．

- 送信専用指示デバイス
  テレビなどのリモコンのように赤外線等で特定のデバイスへデータを送信するもの．指向性を高めたレーザポインタも利用されている．また，デバイスのボタンを押すなど操作されるデバイス側だけ装置が備わっているものもある．

- 送受信可能指示デバイス
  デバイスに取り付けられた 2 次元バーコードを読み取ることや，RFID を読み取る，また電波を利用して双方向の送受信が可能なデバイス．例えば無線 LAN を搭載した PDA をコントローラとして用いる．また USB フラッシュメモリにデータを保持し，デバイスへ USB メモリを差して受け渡す手法もある．またデバイスから受け取った情報を，異なる通信経路で他のデバイスへ送信する手法もある．

ViCon では，実空間情報を基にデバイス間で仮想通信網を構築する．この仮想通信網構築に際して必要な情報を，実空間でデバイスに渡す必要がある．．構築手法の詳細に関しては 4.2.3 で述べるが，それに先立ち望ましい実空間の指示デバイスの要求条件を挙げておく．

既に利用されてる家電などの既存デバイスでの実空間インタフェースは，殆どが送信専用デバイスである．携帯電話などには赤外線などにより双方向でデータを送受信することやカメラにより 2 次元バーコードを読み取る機能が備わっているが，安価な赤外線リモコンが大半を占める．

送信専用の指示デバイスは，デバイスの状態や型番を読み取って動作を変更するなどの細かな操作ができない．しかしながら，レーザポインタなどの高い指向性を持って遠くのデバイスへ情報を渡すことや至近距離のデバイスにのみ情報を渡すなど，実空間でデバイスを指定する際に容易である．また，単純で安価なデバイスという面でメリットがある．

RFID や 2 次元バーコードを用いると，各デバイスの情報を読み取り適した操作を行えるメリットがある．しかしながら，各デバイスに固有の値をグローバル規模で規定しなければならない．また，動く指示デバイス側へ情報を送る場合，指向性を持たせられないため，周囲にある他のデバイスが容易に通威信内容を読み取れるなど，通信の秘匿性という面で劣る．さらに，読み取り後，システムへ何かしらの操作情報を送らねばならない．このために送信する機能を持たねばならないが，通信経路を 2 つ以上持つと，システムが高価になるだけでなく複雑になる．

以上より，送信専用デバイスで実現できるほうが都合がよい．

4.2.3　実空間操作に基づく仮想通信網構築

構築手法の要求条件

実空間操作に基づいてデバイスをグルーピングし，仮想通信網を構築する必要がある．仮想通信網に必要な機能は，通信先のデバイスを限定することと暗号化することで仮想通信網上の通信を保護することである．

さらに，仮想通信網の望ましい要求条件として以下の 3 つが挙げられる．1 つ目は，グルーピングが容易な操作で実現できることである．デバイス連携を容易に行うことが目的である以上，操作は容易であることが望ましい．2 つ目は，多数のグルーピングを動的にユーザ主導で行えることである．これは，グルーピングはデバイスの連携数だけ行われ，グルーピングの指示はデバイスを利用するユーザが行うためである．3 つ目は，リンク層で仮想通信網を構築することである．これにより UPnP など IP 層で動く既存機構に変更を加えずにすむ．

共通鍵を用いた通信制御

4.2.3 の要求を踏まえ，共通鍵が暗号鍵と通信識別子としての性質を併せ持つことに着目し，共通鍵を用いた通信制御による仮想通信網構築を行う．この手法では，まず実空間で指示ポインタを用いて連携させるデバイスに共通鍵を渡す．次いで各デバイスはその共通鍵を用いて通信の暗号化とその復号化の可否による通信先のデバイス識別を行う．

連携するデバイスが共通の通信識別子を持ち相手を識別する手法は，各デバイスの MAC アドレスや IP アドレスといったデバイスを一意に示す固有の通信機別子を用いる手法と比較し，実空間操作が容易である．たとえば，デバイス A，B をグルーピングする場合を挙げる．固有の通信識別子を用いる手法では，指示デバイスが A から通信識別子 IDa を受け取り，B へ渡し，同様に B から通信識別子 IDb を受け取り A へ渡す必要がある．この作業は，グルーピングを行うデバイス数

が N 台に増えると操作の回数は $N(N-1)$ に増加する．一方，共通の通信識別子の場合，指示デバイスは通信識別子 IDc を A と B へ渡すだけでよい．これはデバイス数が N 台に増えた場合にも，操作数は N 回で済む．

ユーザが動的に多数のグルーピングを行うことを考慮すると，衝突を確率的に回避するために共通の通信識別子は大きな識別子空間を持つ必要がある．共通鍵は確率的に衝突を起こさない大きな空間を持ち，連携するデバイス間の秘密情報である．このため，共通鍵を共通の通信識別子として利用できる．

また，この共通鍵により実空間で指定されたデバイスのみが連携でき，他のデバイスが詐称して通信に介入することができなくなる．これは，認証情報を指示デバイス内に保持するために，自律的に動作する膨大なデバイスであっても適用できる．

具体的な処理を図 4.2 に示す．フレームのペイロードとその MAC(Message Authentication Code) を識別子を鍵として暗号化し，マルチキャストする．L2 マルチキャストアドレスは，共通鍵のハッシュを用いて，同一の共通鍵で同じマルチキャストアドレスとなるよう作成する．なお，マルチキャストアドレスはトラフィックの効率化を狙ったものであり，衝突を起こしても構わない．受信側は鍵を用いて復号化し，MAC を確認し，一致すれば同じ識別子として受信する．異なる場合は，ペイロードを正しく復号できていないため，フレームを破棄する．

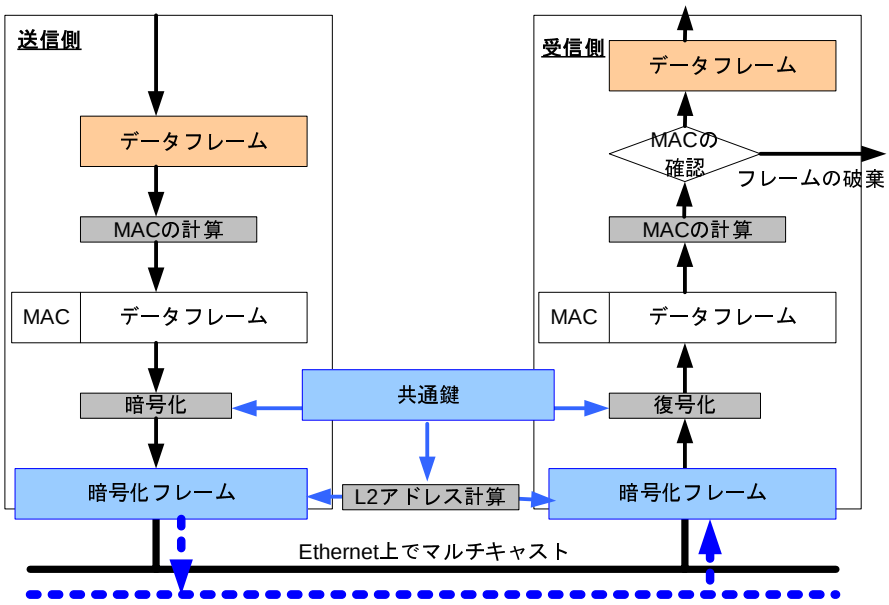

図 4.2: 共通鍵を用いた仮想通信網構築機構

4.2.4　既存サービス発見機構との連携

仮想通信網へのインタフェースは，既存サービス発見機構にとって通常の L2 インタフェースとして扱われるように作成する．特殊な処理を行う L2 インタフェースを作成しアプリケーションから通常の L2 インタフェースのように扱える技術は，直接カーネルで制御操作を行うことや，仮想

NIC(Network Interface Card) などとして実現可能であり，既に実用化されている技術である．1 つのデバイスが，複数の仮想通信網に接続する場合は，通信網の数だけ L2 インタフェースを作成する．

通信制御を行っている L2 インタフェースは，実空間で選択されたデバイスのみに繋がり，通信は暗号化されている．既存サービス発見機構はそれらを意識せず，従来と同じ動作を行える．なお，L2 インタフェースであるため，IP アドレスなどレイヤ 3 以上の設定は，行われていないが閉じた，仮想通信網はデバイス間で閉じているため，IP が必要であれば適当なアドレスを割り当てることで解決できる．

実空間操作から既存サービス発見機構が動作するまでの概要を述べる．まず 4.2.3 で述べた手法により，指示デバイスから連携させるデバイスに共通鍵を渡す．これをトリガとしてデバイスはインタフェースに共通鍵を渡し通信制御を開始する．次いで，既存サービス発見機構にインタフェースが有効になった旨の通知を行い，既存サービス発見機構でサービス発見を行う．なお，この共通鍵を受け取りインタフェースへ割り当て，サービス発見機構を呼び出すための“ サービス連携管理マネージャ ”というアプリケーションが，既存サービス発見機構とは別に動作しておく必要がある．ただし，このアプリケーションは共通鍵の受け渡しや，既存サービス発見機構を呼び出すだけの仲介を行うためのものであり，既存サービス発見機構へ変更を加えるものではない．

4.3　おわりに

本章では，デバイス連携を容易かつ安全に行うために，実空間操作に基づいたリンク層でのデバイスグルーピング機構により，実空間操作を既存サービス発見機構へ変更を加えることなく反映させ，またデバイス間の通信を包括的に保護する手法を提案した．

容易な実空間操作によりリンク層で仮想通信網を構築するための手法として，共通鍵が通信識別子と暗号鍵としての性質を持つことに着目し，共通鍵を用いた通信制御方法を提案した．この手法では，LAN での多数の動的なユーザ主導の仮想通信網構築が実現できる．

また，この仮想通信網と既存サービス発見機構がどのように連携して動作するのかを述べた．

今後はこれらの機構を実装し，機構の動作確認と仮想通信網の性能評価を行う．

第5章

ViCon実装

5.1　はじめに

第 4 章で，実空間操作に基づくリンク層でのデバイスグルーピング機構について述べた．指示デバイスを用いた実空間操作でデバイスに共通鍵を渡し，この共通鍵によって限定されたデバイス間の安全な仮想通信網を構築する．その通信網上でサービス発見とデバイス間通信を行うことで，安全かつ容易なデバイス連携を実現する．

本章では，提案した手法の動作確認及び性能評価を行うために実装を行う．デバイス連携の際に，PC など多機能で高性能なデバイスを用いる場合と，組み込みデバイスなど単機能で低性能のデバイスを用いる場合を想定した．それぞれを PC と iPod[38] に実装を行った．

5.2　実装概要

ViCon は共通鍵を渡す指示ポインタを用いた実空間インタフェース機構 (5.3)，共通鍵による通信デバイスの識別とデバイス間の安全な仮想通信網構築機構 (5.4)，通信チャンネル上でのサービス発見機構 (5.5) の 3 つで構成される．高性能な多機能デバイスと低性能の単機能デバイスの 2 つを実装・測定対象とした．

実空間インタフェースで，共通鍵を生成し保持する指示モジュールと，指示モジュールから共通鍵を受け取り連携するデバイス本体へ共通鍵を渡す受信モジュールで構成される．これらを PAVENET モジュール [7] を用いて作成した．

仮想通信網構築機構は，L2 の NIC で共通鍵を基に Ethernet フレームを暗号化や復号化，フィルタリングを行う．実装は，Windows と iPod Linux[39] に行った．高性能デバイスでは Windows を用いて同時に複数の仮想通信網が構築できるよう仮想 NIC を作成し，それぞれでフレーム処理を行う．低性能デバイスとしての iPod では，物理的な Ehternet インタフェースは無いものの，Ethrnet over Firewire によって，Ethernet での通信を行うことができる．この仮想的な Ethernet インタフェースでフレーム処理を行う．

サービス発見連携機構は，共通鍵を各インタフェースへ渡すことや共通鍵の受け取りをトリガとして，既存サービス発見機構を呼び出すことを行っている．今実装においては，サービスの情報要求と告知を行う簡単な独自プロトコルを用いた．ただし，これは既存サービス発見機構の代替物である．

また動作確認のため，実際に動作するサービスアプリケーションを作成する．高性能デバイスの PC 用と，低性能デバイスの iPod でそれぞれ以下のサービスアプリケーションを作成した．

ネットワークディスプレイサービス　多機能デバイスとして，PC を用いたネットワークディスプレイサービスを作成した．このサービスは，PC 画面をネットワーク上に出力するクライアントデバイスと，それを受信しディスプレイに出力するネットワークディスプレイデバイスで構成される．実空間上で指示ポインタを用いて，それらを選択するとクライアントの PC 画面をネットワークディスプレイに表示する．以下にその実装機器の詳細を記す．

- PC 画面出力クライアントデバイス 1 台
  Note PC(Endevoer NT7000Pro, CPU: Pentium M 1.7GHz, Memory: 1GB, OS: Windows XP Pro SP2)
- ネットワークディスプレイデバイス 2 台
  Desktop PC(CPU: Pentium M 2.0GHz, Memory: 1GB, OS: Windows XP Pro SP2)

ネットワークスピーカサービス　単機能デバイスとしてはを用いて音楽を再生するサービスを作成した．このサービスは，音楽を蓄えネットワーク上に送信するネットワーク MP3 プレイヤと，それを受信しスピーカから出力するネットワークスピーカで構成される．実空間上で，プレイヤとスピーカを選択すると，プレイヤで再生される音楽がスピーカから出力される．以下にその実装機器の詳細を記す．組込型デバイスであるとし，低性能なデバイスを用いている．

- ネットワーク MP3 プレイヤ 1 台
  Desktop PC(ThinkCentre M51, CPU: Pentium M 3.2GHz , Memory: 512MB, OS: Debian Linux 2.6.8)
- ネットワークスピーカ 2 台 iPod
  (Apple M9245J/A - PP5002, Memory: 32MB, OS: iPod Linux 2.4.24)

5.3　実空間インタフェース機構

5.3.1　概要

実空間インタフェースは，共通鍵を送信する指示部と，指示部から共通鍵を受け取り PC や iPod に転送する受信部の 2 つで構成される．指示部として，無作為に生成した 128bits の共通鍵を保持し送信できる機能を PAVENET モジュールに実装した．PAVENET モジュールはプッシュボタンを備えており，このボタンを受信部の近くで押すことで共通鍵を受信部へ渡すことが出来る．また，受信部として無線で共通鍵を受け取りシリアルに出力するものを PAVENET モジュールに実装した．受信部は常に受信状態で待機しており，受信したものを全てシリアル出力へ転送する．この受信部と PC は RS-232C ケーブルで接続し，iPod はリモコン端子のシリアル出力部分に受信部からのケーブルを直接半田付けにより接着した．共通鍵を受け取った PC や iPod は，5.4 で述べる通信制御機構へ共通鍵を渡す．

5.3.2　PAVENET

「PAVENET モジュール」[7] とは，猿渡らによって開発されている小型で低消費電力で使用しやすいシングル CPU の無線センサノードのハードウェアである．このハードウェアと組み合わせ，ハードウェア上で動作するソフトウェア「PAVENET OS」も開発されている．PAVENET は ViCon のプロトタイプ実装としての機能を備えており，これらを利用及び開発する環境が当青山森川研究室で備えられている．また，既に他の研究 [40, 41, 42, 43] でも利用されていることから採用した．

PAVENET モジュールは，315MHz 帯の微弱無線で動作する無線モジュールを搭載しており，最大で十数メートルほどの通信範囲を持つ．この出力は調整可能であり，受信側の調整と会わせて 5cm 程度のみに通信範囲を限定することができる．ViCon では特定のデバイスを指し示しそのデバイスのみに共通鍵を送信する必要がある．このため，PAVENET モジュールの通信距離を 5cm ほどに絞り，対象のデバイス近くで操作することでデバイスの選択を実現している．

5.3.3　指示モジュール

指示モジュールの機能は，デバイスをグルーピングするための共通鍵を生成，保持することと，その共通鍵を連携するデバイスへ送信することである．PAVENET 自体のユーザインタフェースは，電源スイッチとプッシュボタンのみである．そのため，複数の共通鍵を生成し保持した場合でもどの共通鍵を送信するかの操作が難しい．プロトタイプ実装ということもあり，それぞれ異なる共通鍵を持った指示モジュールを 3 つ作成した．

デバイスに対する操作として，仮想通信網への参加，脱退の 2 パターンがある．仮想通信網へ参加させる場合，その仮想通信網用の共通鍵を渡す．脱退させる場合は，仮想通信網の共通鍵とは異なる無効な共通鍵を渡す．このように，参加・脱退は渡す共通鍵が異なるだけである．

指示モジュールから受信モジュールへ渡すフレームを図 5.1 に示す．ヘッダには混線を回避するために ViCon プロトコルであるとして，“ ViCon: ”の文字列が入っている．続いて，渡す共通鍵が入る．この共通鍵は新たな仮想通信網を構築するたびにランダムに生成されが，今実装では各 PAVENET モジュールごとに固定している．プッシュボタンを押すとこのフレームを範囲 5cm 程へ送信する．

図 5.1: 共通鍵を渡す指示フレーム

5.3.4　受信モジュール

受信モジュールの機能は，指示モジュールからフレームを受け取り，共通鍵を v デバイスへ転送することである．PAVENET はシリアル出力を備えており，シリアル経由でデバイスへ共通鍵を転送する．

受信モジュールは，無線経由でフレームを受け取ると，まずヘッダに“ ViCon: ”があるか確認する．確認ができたら続く共通鍵を含めたフレーム全体をシリアルへ出力する．

PC へ受信モジュールを取り付ける場合は，PAVENET のデバッグボードを経由して RC-232C で PAVENET モジュールと PC を接続した．通信速度は 9600bps である．

iPod へも受信モジュールを取り付けるが，iPod は RC-232C 等の標準的なシリアルポートを備えていない．しかしながら，iPod のリモコン部分はシリアル端子となっており，RX 端子を備えている．そこで，PAVENET のシリアル出力端子を，iPod リモコンの RX 端子に半田付けした．配線図を図 5.2 に示す．本来シリアル端子の電圧は 12V であり，電圧変換が必要である．今回，iPod のシリアルコントローラと PAVENET が共に PIC を用いており，3.3V で両方とも動作するため直結が可能であった．通信速度は 9600bps である．

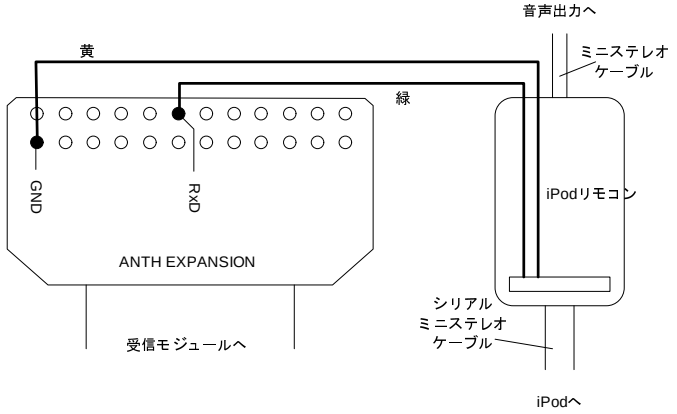

図 5.2: 受信モジュールを iPod に接続するための配線図

5.4　共通鍵を用いたリンク層での仮想通信網構築機構

5.4.1　概要

与えられた共通鍵を基に，NIC(Network Interface Card) で通信制御を行う．送信側は，Ethernet フレームのペイロードとその MAC を共通鍵を用いて暗号化し，今実装ではブロードキャストで送信する．受信側は，受け取ったフレームを共通鍵によって復号化し，MAC により整合性を確認する．確認できた場合は，同じ共通鍵を用いた NIC からのフレームであるとし受信する．整合性を確認できなかった場合は，共通鍵が異なるとしてフレームを破棄する．なお，MAC のアルゴリズムとして 16bits CRC，共通鍵を用いたストリーミング暗号には RC4 を用いた．

Windows でこの通信制御を行うために NDIS(Network Driver Interface Specification) 中間ドライバ [44] を作成した．NIDS 中間ドライバは，物理 NIC を扱う下位ドライバと IP 等を扱う上位ドライバの中間に位置し，L2 フレームのフィルタリングや改変，仮想 NIC の作成を行うことができる．Windows は高性能デバイスとして扱うため，1 つのデバイスで複数の通信やサービスを行えるように作成した．ドライバは，1 つの物理 NIC に対して複数の仮想 NIC を作成し，各仮想 NIC に対して別々の共通鍵を対応づける．送信時は，仮想 NIC の上位ドライバから渡されたフレームを対応づけられた共通鍵を使って処理を行い，下位ドライバへ渡す．受信時は，下位ドライバから受け取ったフレームに対して，各仮想 NIC の共通鍵で順に処理を行い，該当する仮想 NIC が見つかれば，その仮想 NIC の上位ドライバへフレームを渡す．

iPod に Ethernet の NIC は無いが，Ethernet over Firewire を用いて通信を行う．iPod Linux でアクセス制御を行うために，iPod Linux Kernel の ethernet プロトコルスタックへコードを追加した．iPod は低性能デバイスとして，1 つの NIC のみを持つよう作成した．カーネルの eth1394 で，送信時は暗号化処理，受信時は復号化と MAC 整合性の確認，及びフィルタリング処理を行う．また，測定のため Debian LInux 2.6.8 に ipod と Ethernet 通信するためのモジュール ipodeth1394 を追加し，同様の処理を行うコードを作成した．

以下で Windows と iPod のそれぞれの実装について詳しく述べる．

5.4.2　Windows への実装

Windows で仮想的な L2NIC を作成し，流れるフレームに特殊な処理を施しながらもアプリケーションから通常の NIC として扱えるようにするためには，通常のアプリケーションとは異なるデバイスドライバを作成する必要がある．

統合開発環境として Visual Studio 2003[45] を，デバイスドライバの開発には Windows DDK(Driver Development Kit) XP SP1 build 3663[46] を用いた．

以下では，Windows プロトコルスタックについて解説し，実装に必要な部分を述べる．次いで実装を行う NDIS 中間ドライバの処理概要と，フレームに対する処理を述べる．

Windows プロトコルスタック

Windows では，物理 NIC を扱うドライバから TCP セッションを管理するドライバ，ユーザが容易にソケットを扱えるようにする winsock ライブラリまで，階層化された構造を持つ．この階層図を図 5.3 に示す．まずカーネルモードの部分では下位層から順にハードウェアを操作する Miniport デバイスドライバ，有線 Ethernet や無線 LAN などのメディアの違いを吸収し汎用的な操作を提供する NDIS ドライバ，TCP/IP を提供する TCP/IP ドライバなどのプロトコルドライバ，トランスポートプロトコルを扱う TDI(Transport Driver Interface) ドライバと積まれている．ユーザモードの部分では，Winsock のプロトコルスタックを透過的に拡張することができるサービスプロバイダインタフェース (SPI) と，アプリケーション開発者から下位層を隠蔽するアプリケーションプログラミングインタフェース (API) が規定されている．

ViCon では，送受信されるフレームに対して暗号化や復号化，フィルタリングしてフレームを破棄するなどの操作を行う．また，1 つの物理に NIC に対して複数の仮想 NIC を作成し，複数の仮想通信網に参加することができるようにする．

以上の条件を実現できるプロトコルスタックは，NDIS ドライバである．

NDIS 中間ドライバ

NDIS 中間ドライバは，フレーム受信時には下位の Miniport ドライバからフレームを受け取り，上位のプロトコルドライバへフレームを渡す．またフレーム送信時には逆に上位プロトコルドライバからフレームを受け取り，下位の Miniport ドライバへフレームを渡す．この際に仮想的 NIC を複数あるよう扱うことができる．図 5.4 に，仮想 NIC が複数ある場合のフレームのフローを示す．1 つの物理に NIC に対応する Miniport ドライバからフレームを受け取ると，フレームをチェックし適切な仮想 NIC があればその仮想 NIC の上位プロトコルドライバへ渡す．逆に送信時，上位プロトコルドライバからフレームを受け取ると，仮想 NIC 独自の処理を施し下位の Miniport ドライバへ渡す．このように，本来の Miniport ドライバとプロトコルドライバの間に入り，双方に対して仮想的なドライバとして振る舞い，仮想的に NIC を増やすことができる．また，NDIS で仮想的に作成された NIC は，OS 上でも通常の L2NIC として扱うことができる．

フレーム処理

NDIS 中間ドライバでのフレーム処理について述べる．下位の Miniport ドライバからフレームを受け取る IM プロトコルドライバと，上位プロトコルからフレームを受け取る IM Miniport ドライバに分けて作る．

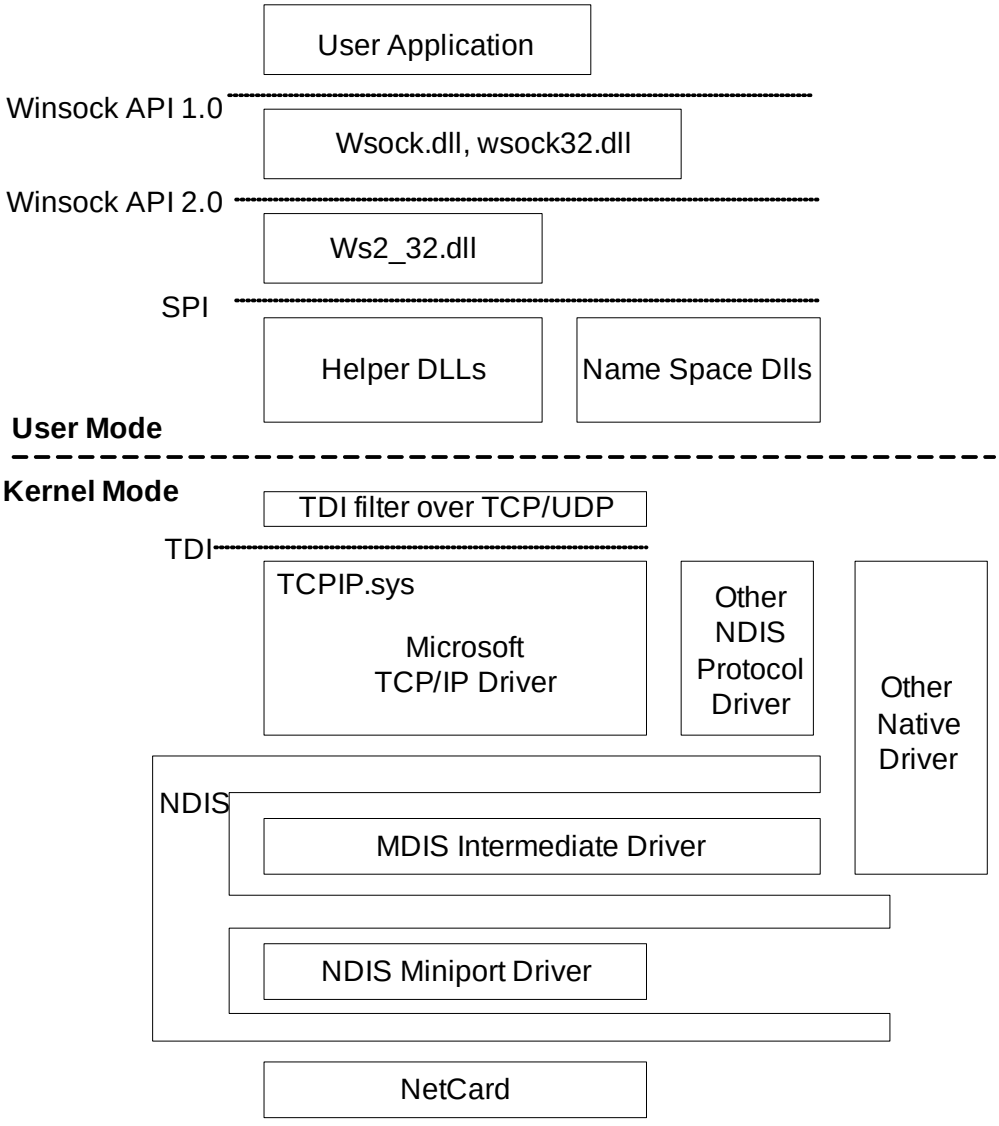

図 5.3: Windows プロトコルスタック

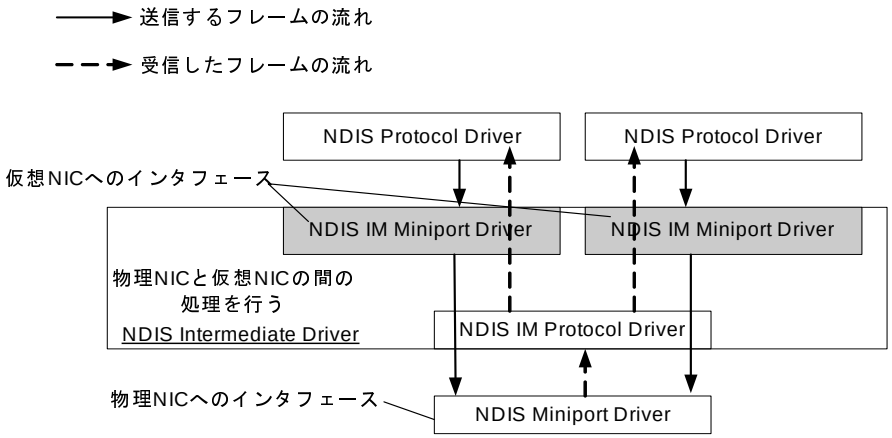

図 5.4: NDIS 中間ドライバでの仮想 NIC 処理

送信処理　送信する場合，上位プロトコルドライバから Ethernet フレームを受け取る．受け取ったフレームの Ethernet ヘッダ (14Bytes) 部分を除くペイロードの MAC(Message Authentication Code) を計算する．ここでは CRC 16bits を用いている．次にフレームを拡張子，Ethernet ヘッダとペイロード部分の間に 6bytes のスペースを設ける．新たに設けた 6bytes の先頭 4bytes には，フレームのシーケンス番号を入れる．シーケンス番号はフレームを送信するたびに 1 ずつ加算される．このシーケンスは，暗号化されたフレームを再度送りつけるリプライ攻撃を防止するために設けた．残りの 2bytes にはペイロード部分の CRC 値を入れる．値を入れた後，拡張された 6bytes 部分を含むペイロード全体を，仮想 NIC に対応づけられた共通鍵を用いて暗号化する．暗号化アルゴリズムとして RC4 を用いた．処理を行ったフレームのフォーマットを図 5.5 に示す．なお，フレームサイズが 60Bytes 未満の場合，NIC で自動的に 60Bytes まで保管される場合がある．この場合，復号化の際に CRC が一致しなくなる．そのため，60Bytes 未満の場合は，乱数で 60bytes まで補完し CRC の計算と暗号化を行う．

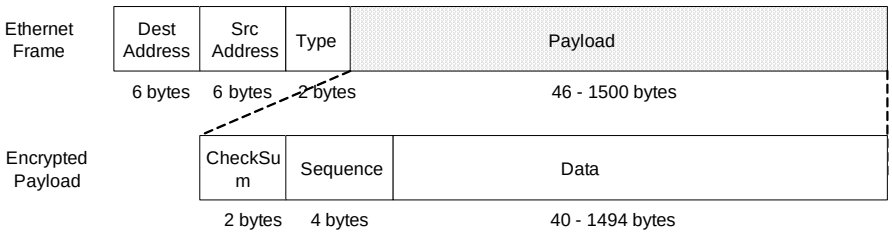

図 5.5: 暗号化された Ethernet フレームフォーマット

受信処理　フレームを受信する場合は，まず Miniport ドライバから暗号化済みのフレームを受け取る．受け取ったフレームに対してそれぞれの仮想 NIC に対応づけられた共通鍵で，拡張 6bytes

部分を含むペイロードの復号化を試みる．復号化後，Eternet ヘッダ，拡張 6bytes 部分を除くペイロードの CRC 16bits を計算し，拡張 6bytes 部分に保持されている CRC 値と比較する．CRC 値が一致すれば，同じ共通鍵を用いて暗号化されたフレームであると判断でき，フレームの処理を続行する．CRC が一致しなかった場合は，共通鍵が異なるとし同じ仮想通信網上のデバイスからのフレームでないと判断できる．この場合，フレームの処理を中断し，フレームを破棄する．処理を続行する場合はシーケンス番号を確認する．以前同じ送信元から送られてきたフレームのシーケンス番号と比較し，それよりも大きな値であったら受け取る．それ以下の場合，処理を中断しフレームを破棄する．さらに処理を続行する場合は，拡張された 6bytes 部分を取り除き，元の Ethernet ヘッダとペイロード部分だけに戻して，復号化に用いた共通鍵を持つ仮想 NIC の上位プロトコルドライバへフレームを渡す．

共通鍵管理

各仮想 NIC は，共通鍵とシーケンス番号に関する情報を保持する．また，各仮想 NIC は実空間インタフェースで受信した共通鍵を，受け取らねばならない．このために，共通鍵を受信モジュールと NDIS 中間ドライバの間で橋渡しを行うユーザモードで動作するサービス連携管理マネージャアプリケーションを作成した．サービス連携管理マネージャに関する詳細は 5.5 で述べる．

5.4.3　iPod Linux への実装

iPod Linux は単機能で低性能デバイスと見なすため，1 つの物理 NIC に対して流れるフレームに特殊な処理を施す．仮想 NIC を作成することはしない．このために，iPod Linxu のプロトコルスタックへコードを追加し，処理を行う．

開発環境としては Debian Linux 2.6.8 で iPod Linux 用にクロスコンパイル環境を整えた．また，測定用に Debian Linux 自体と iPod が Ethernet によるパケット通信ができるようにするために，Debian Linux 2.6.8 用に専用モジュールを開発した．

以下では，iPod Linux について解説し，実装に必要な部分を述べる．次いで実装を行うカーネル部分と，フレームに対する処理を述べる．

iPod Linux

iPod Linux は，Apple が市販している iPod デバイス上で動く Linux ディストリビューションである．実際には uLinux に iPod 用のパッチを当てて作成される．iPod 上でマルチタスクをサポートし，基本的な OS の機能をサポートしており，組み込み用 OS として扱うには十分な機能を持っている．iPod ではデバイスに物理的な Ethernet インタフェースが備わっていないが，Firewire インタフェースがあり，iPod Linux では Ethernet over Firewire をサポートする．このため，Ethernet over Firewire 上で Ethernet フレームを送受信することができる．しかしながら，送信可能なフレームサイズは最大で 170 であり，NIC 専用のハードウェアチップがないため処理は全て CPU で行われる．そのため，パフォーマンスは悪い．

カーネルプロトコルスタック

iPod Linux では，Ethernet over Firewire であるため，Ethernet フレームを ieee1394 パケットに変換して送信を行う．また，受信時には ieee1394 パケットを Ethernet フレームに変換する．そこで，今実装ではこの変換部分へフレーム処理コードを追加した．ieee1394 パケットと Ethernet フレームの変換部分は，Debian Linux でも ipodeth1394.ko モジュールとして同様のコードを組み込んである．そのため，同じコードをそのまま通常の Linux へ組み込むことができ，都合がよい．

フレーム処理

iPod Linux でも Windows と同様に，フレームに対して暗号化と復号化，フィルタリング処理を行う．処理を行うコードを変換を行う関数の途中に挿入する．

送信処理　送信する場合，受け取ったフレームの Ethernet ヘッダ (14Bytes) 部分を除くペイロードの MAC(Message Authentication Code) を計算する．ここでは CRC 16bits を用いている．次にフレームを拡張子，Ethernet ヘッダとペイロード部分の間に 6bytes のスペースを設ける．新たに設けた 6bytes の先頭 4bytes には，フレームのシーケンス番号を入れる．シーケンス番号はフレームを送信するたびに 1 ずつ加算される．このシーケンスは，暗号化されたフレームを再度送りつけるリプライ攻撃を防止するために設けた．残りの 2bytes にはペイロード部分の CRC 値を入れる．値を入れた後，拡張された 6bytes 部分を含むペイロード全体を，共通鍵を用いて暗号化する．処理を行ったフレームのフォーマットは Windows と同じく図 5.5 のようになる．

受信処理　フレームを受信する場合受け取ったフレームに対して共通鍵で，拡張 6bytes 部分を含むペイロードの復号化を試みる．復号化後，Eternet ヘッダ，拡張 6bytes 部分を除くペイロードの CRC 16bits を計算し，拡張 6bytes 部分に保持されている CRC 値と比較する．CRC 値が一致すれば，同じ共通鍵を用いて暗号化されたフレームであると判断でき，フレームの処理を続行する．CRC が一致しなかった場合は，共通鍵が異なるとし同じ仮想通信網上のデバイスからのフレームでないと判断できる．この場合，フレームの処理を中断し，フレームを破棄する．処理を続行する場合はシーケンス番号を確認する．以前同じ送信元から送られてきたフレームのシーケンス番号と比較し，それよりも大きな値であったら受け取る．それ以下の場合，処理を中断しフレームを破棄する．さらに処理を続行する場合は，拡張された 6bytes 部分を取り除き，元の Ethernet ヘッダとペイロード部分だけに戻す．

5.5　サービス発見連携機構

5.5.1　概要

サービス発見機構は，実空間インタフェースでの共通鍵の受信をトリガとしてサービス発見を行い，サービスが見つかると対応するアプリを動作させる．この際，サービス発見機構やアプリは，通信制御機構が動作している Ethernet の NIC を用いて通信を行う．通信制御機構はレイヤ 2 で動作しているため，サービス発見や通信に用いるレイヤ 3 以上のプロトコル (UPnP や TCP，UDP) を従来のまま使用できる．

今実装では，Windows，iPod Linux においてもカーネル部分で処理を行っているため，サービス発見機構やアプリからは通常の NIC として扱える．Windows で複数の仮想 NIC を作成した場合

は，NIC ごとに異なる仮想通信網を確立しサービスを利用できる．なお，今実装においては，サービス発見機構として独自の簡易サービス発見プロトコルを使用している．これは，定期的にサービス発見パケットを NIC 上にブロードキャストし，返答があるとその情報をアプリに通知する．アプリは通知された情報を基にサービスを行う．

5.5.2　サービス連携管理マネージャ

実空間インタフェースは共通鍵の送受信を行い，デバイスグルーピング機構は NIC での通信制御を行う．この際，実空間インタフェースから受信した共通鍵をデバイスグルーピング機構へ渡すアプリケーションが必要になる．また，共通鍵が NIC に渡されたことは仮想通信網が構築されたことを意味する．そのため，共通鍵が NIC に渡されたことをトリガとして，既存サービス発見機構を仮想通信網上で動作させる必要がある．これらを行う統括的なアプリケーションとして「サービス連携管理マネージャ」を作成した．Windows では，共通鍵の受け渡しとサービス発見機構の動作という基本機能に加え，以下の機能を持つ．

- 各仮想 NIC へ暗号化アルゴリズムを設定する機能
- 仮想 NIC 上のサービスをアイコン化する機能
- 監視するシリアルポートに関して設定を行う機能
- デバッグ機能

サービス連携管理マネージャは，常駐プロセスとして起動しシリアルポートを監視し 5.3 で述べたフレームを受け取る．ヘッダにある“ ViCon: ”の文字列を確認した後，続く共通鍵を取り出す．確認できなかった場合は，データを破棄する．取り出した共通鍵をすぐに通信制御機構へ反映させるために，カーネルモードのデバイスと通信を行い，稼働中の仮想 NIC のメモリ内容を直接書き換える．また，カーネルメモリ中に保持されている共通鍵は，PC の再起動時に初期化されてしまうため，Windows のレジストリにも書き込んでおく．デバイスドライバは，NIC が有効になった際にレジストリから該当する共通鍵を読み込み各仮想 NIC のメモリに書き込む．

仮想 NIC の設定を行うことや仮想 NIC の状況を取得するために，入出力要求パケット (IRP) によってドライバと通信する．このための機構を，NDIS 中間ドライバとサービス連携管理マネージャの双方に実装する．

なお，iPod Linux のサービス連携管理マネージャは，iPod での GUI が貧弱なこともあり基本機能のみを備えたものを作成した．

5.5.3　サービスアプリケーション

実空間操作により実際にサービスがすることを確認するために，サービスアプリケーションを作成した．Windows と iPod Linux にそれぞれサービスアプリケーションを作成してた．以下で，それぞれについて詳細を述べる．

ネットワークディスプレイサービス (Windows)

Windows を搭載した PC は多機能で高性能デバイスとして扱う．PC では，PC の画面をネットワーク上で転送し，他のディスプレイに表示するネットワークディスプレイサービスを作成した．このサービスは，PC 画面をネットワーク上に出力するクライアントデバイスと，それを受信しディスプレイに出力するネットワークディスプレイデバイスで構成される．実空間上で指示ポインタを用いて，それらを選択するとクライアントの PC 画面をネットワークディスプレイに表示する．

それぞれについて以下で説明する．

PC 画面出力クライアントデバイス

ハードウェア構成：Endevoer NT7000Pro, CPU: Pentium M 1.7GHz, Memory: 1GB
ソフトウェア構成：OS: Windows XP Pro SP2

クライアントデバイスでは，PC のデスクトップ画面を 200ms 間隔で定期的にキャプチャし，UDP パケットにデータを入れブロードキャストパケットして投げる．デスクトップ画面をキャプチャするには，Windows の GDI 関数を用いる．デスクトップのデバイスコンテキストを取得し，ビットマップオブジェクトを対応づけた後，ビットマップのバイト列をメモリにコピーする．この際，効率化を図るために TightVNC で用いられているフィルタドライバを利用した．このフィルタドライバは，デスクトップ内で変化があった部分を検出することができ，これにより差分の画像のみを送ることが可能になる．

画面をキャプチャしたバイト列は，そのままでは UDP パケットに収まらないため，複数のパケットに分割して送信する．この際，どの部分に該当する画像であるかを示すため，UDP パケットに含まれる画像の座標と幅と高さをヘッダ部分に保持する．

ネットワークディスプレイデバイス

ハードウェア構成：Be Silent M7000, CPU: Pentium M 2.0GHz, Memory: 1GB
ソフトウェア構成：OS: Windows XP Pro SP2

ネットワークディスプレイデバイスは，クライアントデバイスが送信する UDP パケットを受け取り，デスクトップへ再現する．デスクトップへ画像を描画する際には Windows の GDI 関数を用いる．デスクトップのデバイスコンテキストを取得し，ビットマップオブジェクトを対応づけた後，受信した UDP に保持されている画像をビットマップ上へコピーする．UDP には，画像のデータとその座標と幅と高さが含まれており，それらの情報を元に正しい位置へ描画する．

ネットワークスピーカサービス (iPod Linux)

単機能デバイスとしてはを用いて音楽を再生するサービスを作成した．このサービスは，音楽を蓄えネットワーク上に送信するネットワーク MP3 プレイヤと，それを受信しスピーカから出力するネットワークスピーカで構成される．実空間上で，プレイヤとスピーカを選択すると，プレイヤで再生される音楽がスピーカから出力される．以下にその実装機器の詳細を記す．組込型デバイスであるとし，低性能なデバイスを用いている．それぞれのデバイスについて，以下で詳細を述べる．

ネットワーク MP3 プレイヤ

ハードウェア構成：ThinkCentre M51, CPU: Pentium M 3.2GHz , Memory: 512MB
ソフトウェア構成：OS: Debian Linux 2.6.8

ネットワーク MP3 プレイヤは，ストレージに保存してある MP3 ファイルをデコードし，無圧縮の PCM 形式にしてネットワーク上へブロードキャストする．この際，UDP パケットに収まるようにするため，1 パケットに保持する大きさは 1kbytes ほどに制限する．iPod と PC 間のスループットが PCM44.1kHz，128bps，ステレオ音声を流すには，不十分であったため，PCM22kHz，128bps，モノラル音声にデコードしている．

ネットワークスピーカ

ハードウェア構成：Apple M9245J/A - PP5002, Memory: 32MB

ソフトウェア構成：OS: iPod Linux 2.4.24

ネットワークスピーカは，ネットワーク MP3 プレイヤが送信する UDP パケットを受信し，音声を取り出して音声出力デバイスへ出力する．iPod の音声出力は 44.1kHz，ステレオであるため受信した音声を元の 44.1kHz，ステレオ音声に戻した後，音声出力デバイスへ出力する．

5.6　おわりに

本章では, 実空間の基づくリンク層でのデバイスグルーピング機構の実装について述べた. PAVENET を用いた実空間インタフェースと，Windows 及び iPod Linux におけるリンク層でのデバイスグルーピング機構，サービス発見機構の連携と動作確認用にサービスアプリケーションを作成した．

次章では，実装した機構の動作確認とリンク層でのデバイスグルーピング機構によって構築される仮想通信網上での性能評価を行う．

第6章

ViCon実験・性能評価

6.1　はじめに

第 5 章では，提案した実空間に基づいたリンク層でのデバイスグルーピング機構について詳細に触れながら，実装について述べた．この機構は，指示デバイスを用いて共通鍵を実空間で直接渡す実空間インタフェース機構，共通鍵を用いて動的なユーザ主導の仮想通信網を構築するリンク層での仮想通信網構築機構，仮想通信網上で動作するサービス発見機構の 3 つで構成される．

本章では，上記の 3 つの機構についてそれぞれ動作を確認する．また，仮想通信網での通信性能を測定し評価を行う．測定では TCP と UDP の速度と，通信時の CPU 負荷を測定した．

6.2　ViCon 動作確認

6.2.1　実空間インタフェース機構

実空間インタフェースは共通鍵を保持し送信する指示モジュールと，共通鍵を受信しデバイスへシリアル経由で転送する受信モジュールから構成される．

PAVENET モジュールで実装した指示モジュールを図 6.1 に示す．PAVENET モジュールのプッシュボタンを押すと，LED が点灯し共通鍵を送信する．

PAVENET モジュールで実装した受信モジュールを図 6.2 に示す．PC と受信モジュールは図 6.2 に示すとおり，RS-232C ケーブルで接続されている．なお，図中の受信モジュールはパッケージ化されておらず基盤がむき出しになっているが，機能はパッケージ済みの PAVENET モジュールと変わらない．実際に指示モジュールから送信された共通鍵を受信モジュールで受け取り，PC へ転送している様子を 6.3 に示す．TeraTerm[47] でシリアルポートに接続し，シリアル経由で送信されてる文字列をダンプしている．図に示すとおり ViCon と続く共通鍵を受け取っていることが分かる．

iPod に受信モジュールを接続した様子を図 6.4 に示す．iPod はシリアル端子がリモコン用に備えられており，受信モジュールから半田付けで直接端子に繋げている．iPod でも PC と同様に共通鍵を受信できることを確認した．

図 6.1: PAVENET モジュールに実装した指示モジュール

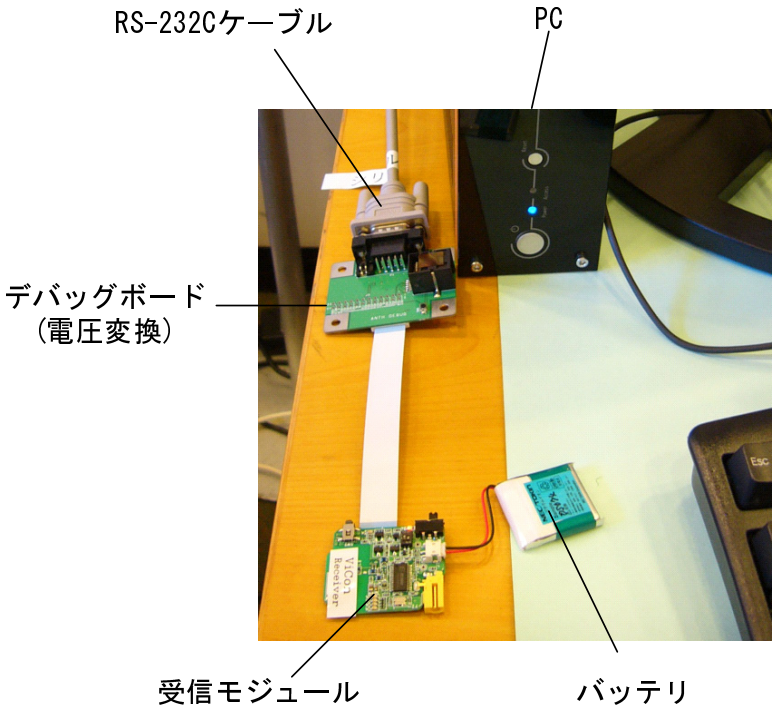

図 6.2: RS-232C ケーブルで接続した受信モジュール

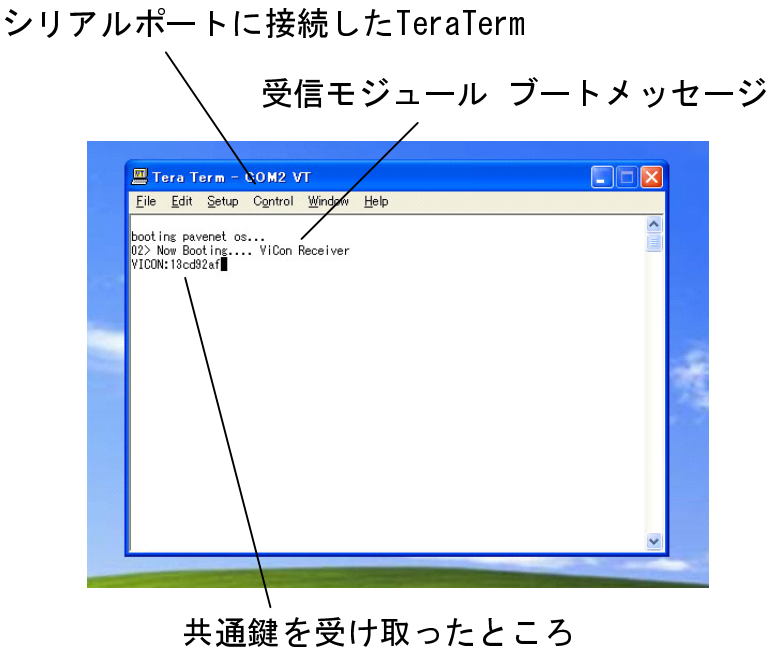

図 6.3: 共通鍵の受信を確認

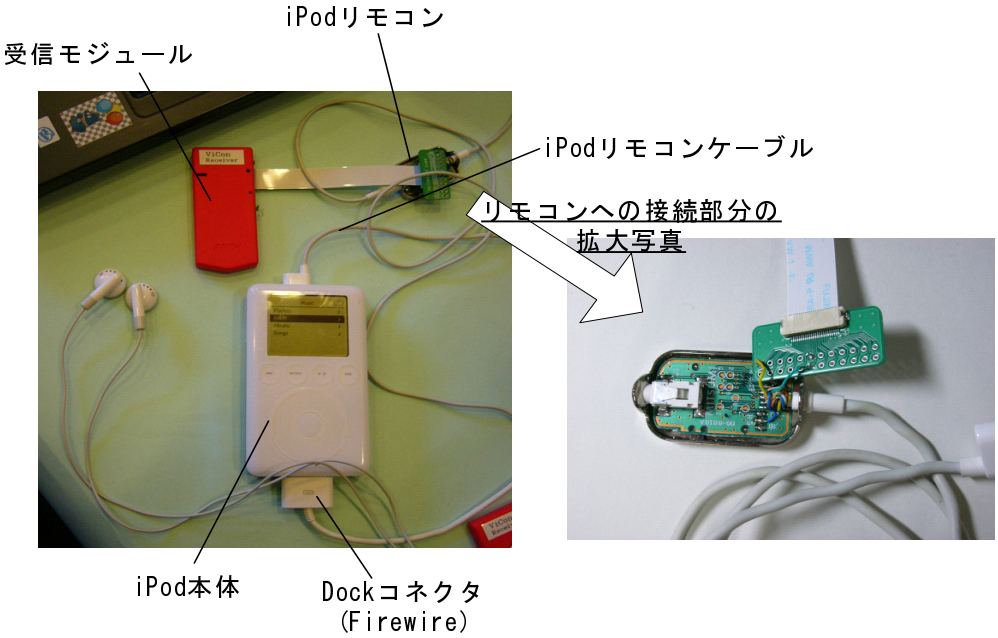

図 6.4: iPod に接続した受信モジュール

6.2.2　共通鍵を用いたリンク層での仮想通信網構築機構

Windows に作成したドライバをインストールし，仮想 NIC を作成した．図 6.5 ではインストールしたデバイスドライバにより物理 NIC1 つに対して 2 つの仮想 NIC を作成している様子を，図 6.6 では仮想 NIC に対して設定を行っているダイアログを示している．設定ダイアログでは各仮想 NIC に暗号化アルゴリズム，共通鍵を設定している．また，測定の際に復号化を行う順が分かるよう，各インタフェースに処理を行う順番が 0,1，... と記してある．

仮想通信網は同じ共通鍵を与えられた仮想 NIC 間で構築される．Ethernet ケーブルで接続された 2 台の PC に仮想 NIC 用のドライバをインストールし，仮想通信網上での通信を確認した．確認は適当な共通鍵 “0x11223344”を割り当て，Ping を用いて行った．また，片方の仮想 NIC の共通鍵を異なる鍵に変更することで，通信が行われないことも確認した．なお，IP アドレスは適当なプライベートアドレスを割り振った．

通信が暗号化されている様子を Ethereal[48] を用いて確認した．確認は 2 台の PC 間に測定用 PC を接続し，2 台の間で送受信されるフレームをキャプチャすることにより行った．その様子を図 6.7 に示す．通常の通信の場合 (a) と比較して，仮想通信網上で行われている通信 (b) はパケットの内容が無意味なバイナリ列に変化し，通信内容が読み取れなくなっている．

iPod Linux でも同様に，Firewire ケーブルで iPod と測定用 PC を接続し通信の確認を行った．また，iPod 同士を接続し，iPod 同士でも通信が行われることを確認した．

6.2.3　サービス発見連携機構

実空間インタフェースと仮想通信網構築機構を用い，実際にサービス発見機構とサービスアプリケーションを動作させる．

図 6.5: 仮想通信網に接続する仮想インタフェース

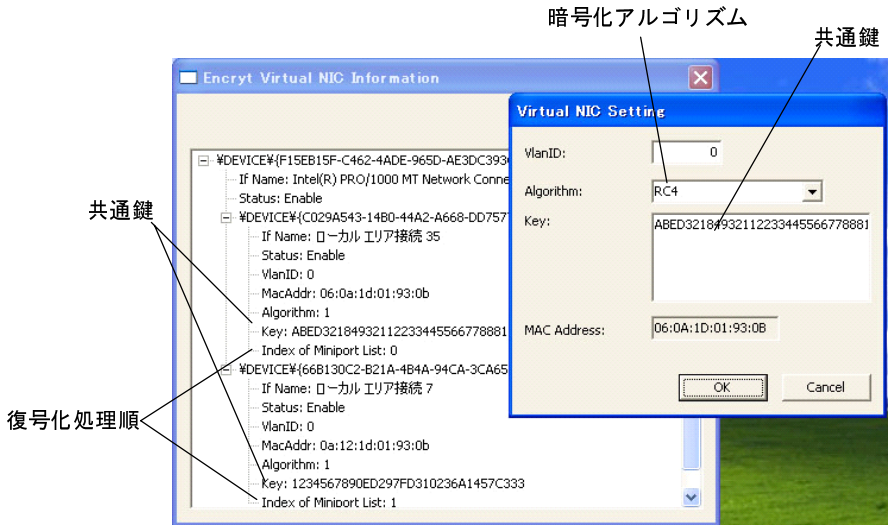

図 6.6: 仮想インタフェースの管理画面

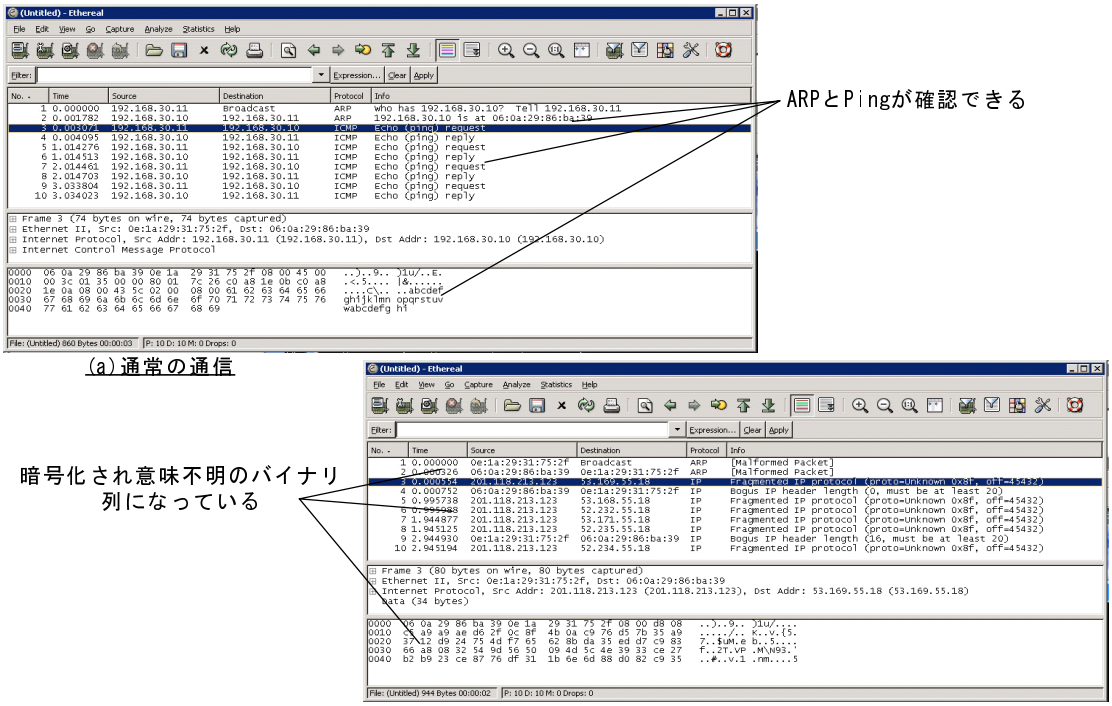

図 6.7: Ethereal を用いたパケットキャプチャ画面

ネットワークディスプレイサービス　ネットワークディスプレイを実装した様子を図 6.8 に示す．指示機能を備えた PAVENET モジュールを用いて，利用する PC の受信部を選択すると，選択された PC 間で PC 画面を転送している様子を示している．

ネットワークスピーカサービス　図 6.9 でネットワークスピーカを iPod に実装した様子を示す．ネットワーク MP3 プレイヤとネットワークディスプレイを PAVENET モジュールを用いて選択すると，音楽が再生される様子を示している．

これらにより，PC を用いたネットワークディスプレイサービス，iPod でのネットワークスピーカサービスともに実空間で選択することにより，利用できることを確認した．

6.3　ViCon 性能評価

6.3.1　測定実験

PC 及び iPod で，共通鍵を用いたデバイスグルーピング機構の TCP と UDP の速度，及び通信時の CPU 負荷を計測した．

PC での測定

PC での測定では，グルーピング機構を備えた PC2 台をクロスケーブルで直結し，Netperf[49] を用いて TCP の速度を測定した．測定は，仮想通信網を使わない通常の通信の場合と，NIC で通

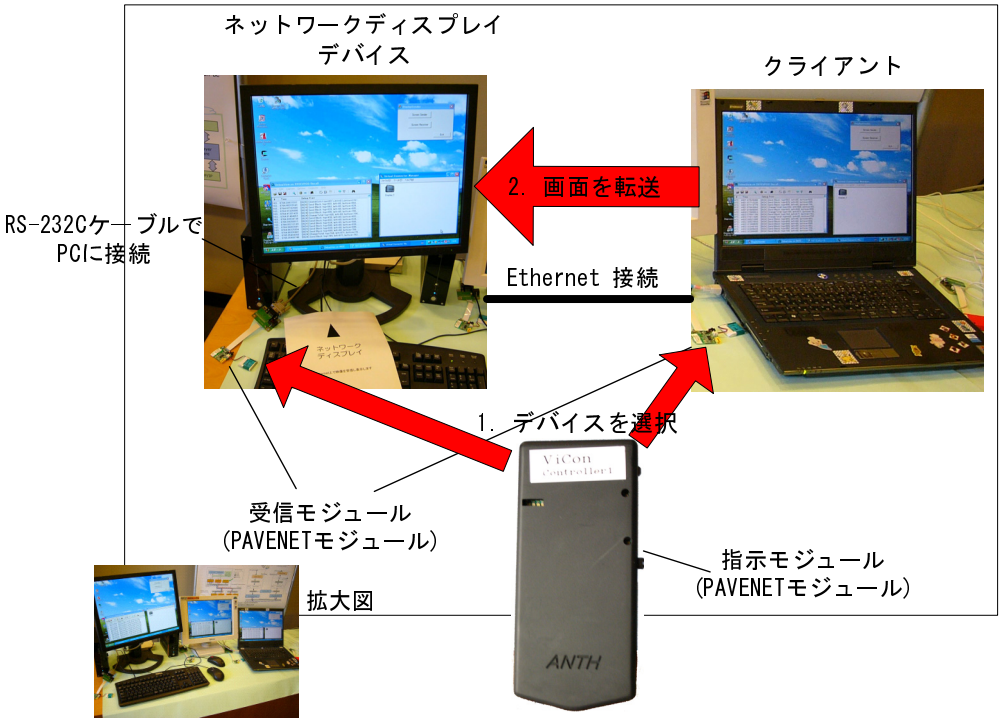

図 6.8: ネットワークディスプレイサービスが動作する様子

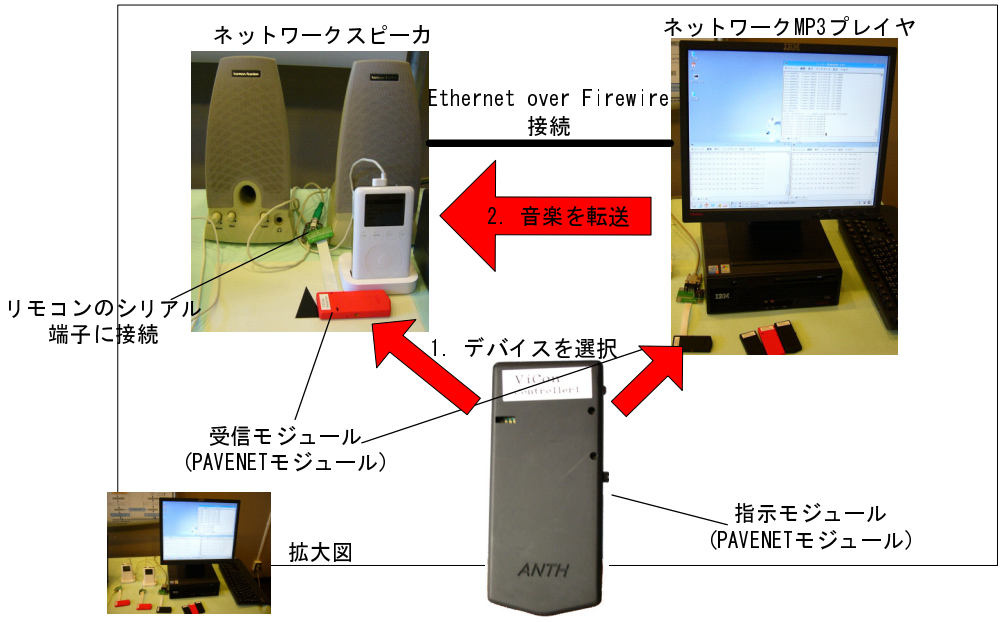

図 6.9: ネットワークスピーカサービスが動作する様子

信制御を行い仮想通信網上で通信する場合の 2 通りを行った．フレーム長による負荷の変化を見るため，MTU を 100〜1470Bytes まで変化させながら，それぞれ 10 秒間，5 回ずつ測定しその平均を求めた．

また，複数の仮想 NIC がある場合は，図 6.10 に示すように，順に復号化処理を行う．そのため処理が後に行われる NIC ほど，事前に失敗する復号化処理を繰り返すため処理負荷が高くなると思われる．例えば，2 番目のインタフェースで通過するフレームでも，1 番目のインタフェースの復号化処理が失敗した後に行われるため，1 フレームに対して復号化を 2 回行うことになる．この負荷を測定するために，フレームを受け取った際に処理を行う順に NIC をインタフェース 1，2 と名付け，それぞれのインタフェースで通信を行い，受信側で CPU 負荷の測定を行った．

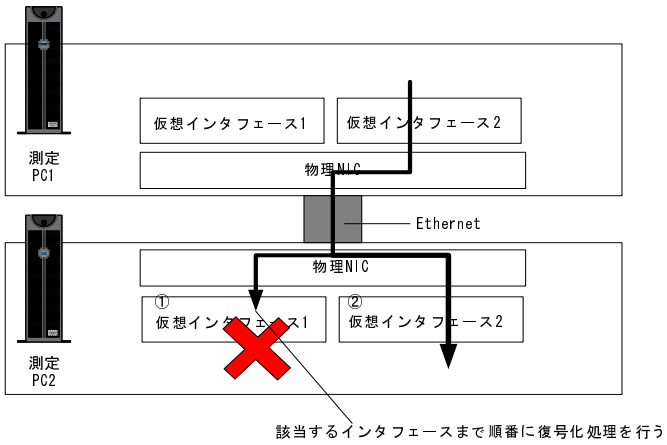

図 6.10: 複数インタフェースがある場合の処理手順

iPod での測定

iPod の測定は，iPod と測定用 PC を Firewire で直結し，1MBytes を転送することで TCP の速度を測定した．PC の場合と同様に，仮想通信網を使わない通常の通信の場合と，NIC で通信制御を行い仮想通信網上で通信を行う場合の 2 通りの測定を行った．iPod では使用できる MTU が最大 170Bytes であるため，MTU は 100〜170Bytes まで変化させ，それぞれ 5 回ずつ転送に要する時間を測定し速度を求めた．

6.3.2　評価

PC での速度と CPU 負荷

PC での TCP 速度の測定結果を図 6.11, 図 6.12 に示す．仮想通信網を構築するための通信制御を置こなうと，行わなかった場合と比較して約 0.7Mbps の速度低下となった．制御のための CPU 負荷は，TCP の送信側では約 22%，受信側では約 30%の負荷増大となった．MTU が小さいとき場合に処理負荷が高くなるのは，フレーム毎に鍵のスケジューリングやフレームの拡張を行っているためだと考えられる．

複数インタフェースがある場合でも，インタフェース 1 と 2 の違いによる TCP 速度，CPU 負荷の違いはなかった．このためこの負荷増加は，暗号化の計算によるものではなく，フレームを拡張するために新たにメモリを確保するなど実装面での要素が大きいと考えられる．また，送信側で CPU 負荷と比較し，受信側の CPU 負荷は MTU の値に依らず一定の値を保っている．これは，送信側の負荷が増大したことによりパケットの送信量が低下し，受信側で行う処理量が減ったと考えられる．

PC での UDP 速度の測定結果を図 6.13, 図 6.14 に示す．制御を行った場合では約 0.8Mbps の速度低下となった．また送信側で約 20%の負荷増加，受信側で約 28%の負荷増加となっている．TCP の場合と同様に，インタフェース 1，2 による違いは無かった．

iPod での速度と CPU 負荷

iPod の TCP の測定結果を図 6.15 に示す．通信制御を行わなかった場合と比較して，通信制御を行い仮想通信網上で通信を行った場合では，約 24%速度が低下している．CPU はどちらにおいても 100%であった．PC の場合と異なり，速度が大きく低下したのは iPod が低性能デバイスであり，元々の CPU 負荷 100%であったため，制御処理が大きく影響したと思われる．

iPod の UDP の測定結果を, 図 6.16 に示す．通信制御を行わなかった場合と比較して，通信制御を行い仮想通信網上で通信を行った場合では，約 25%速度が低下している．CPU はどちらにおいても 100%であった．

以上のことから，PC ではソフトウェアで制御を行うことが十分可能であると思われる．iPod など組込型デバイスでは，通信制御によって全体のパフォーマンスが 7 割に低下することを考慮すると，速度を必要とするデバイスではソフトウェア処理では不十分であろう．このような場合は，ハードウェアで処理を行う暗号化チップなどを利用することが考えられる．

6.4　おわりに

本章では，実装した実空間に基づくリンク層でのデバイスグルーピング機構の動作確認と性能評価を行った．この機構は，実空間インターフェス機構と，共通鍵を用いたリンク層での仮想通信網構築機構，サービス発見機構から構成される．それぞれを確認し，最後にサービスアプリケーションが実空間操作により実行されることを確認した．

測定では，共通鍵を用いた仮想通信網上で TCP と UDP の速度，及び通信時の CPU 負荷を測定した．PC などの高性能デバイスでは，CPU 処理負荷が増加したものの，通信速度は変わらなかった．一方で iPod では元々CPU を 100%利用していたため，仮想通信網上での通信は速度が低下した．

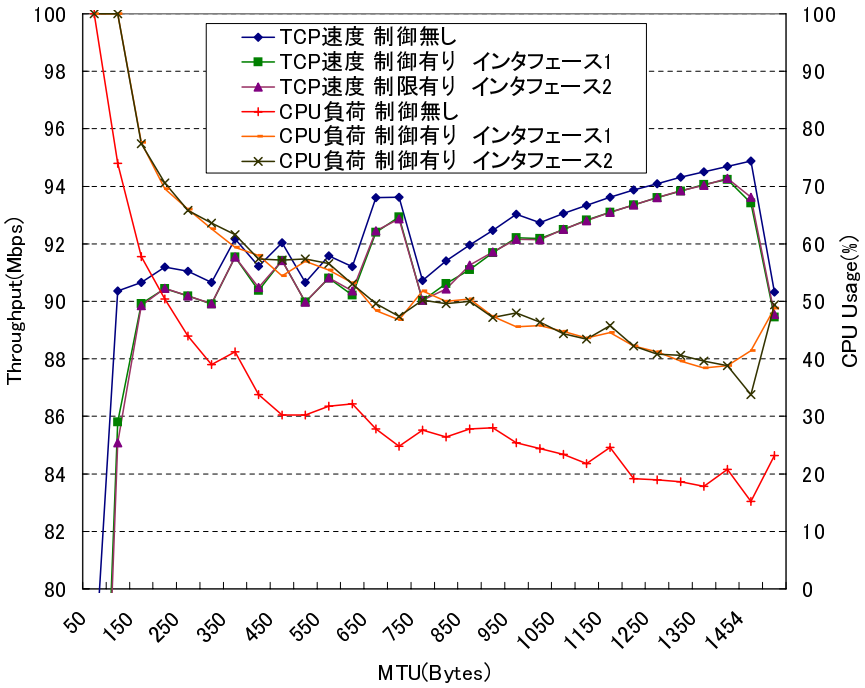

図 6.11: PC での TCP 速度 送信側

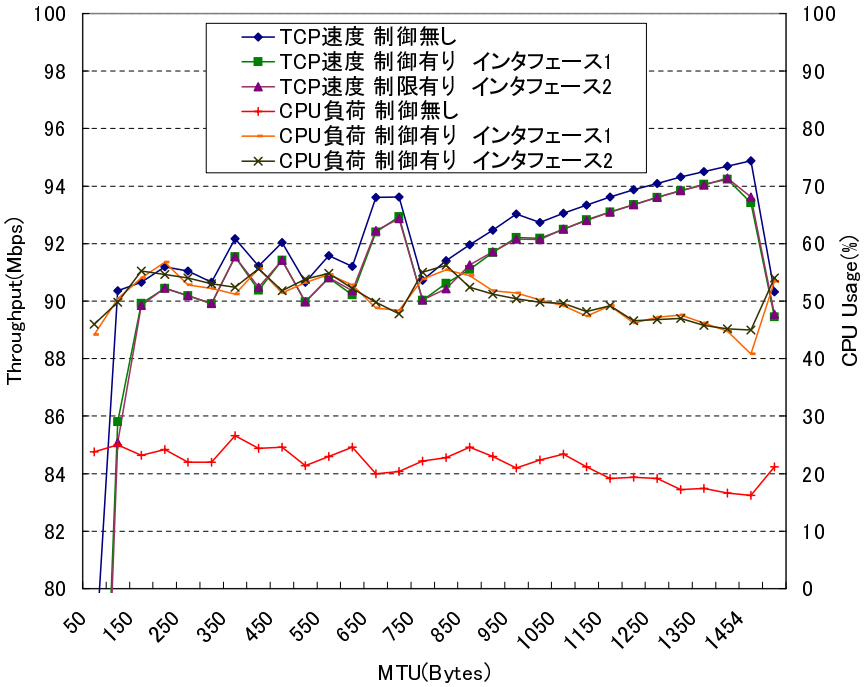

図 6.12: PC での TCP 速度 受信側

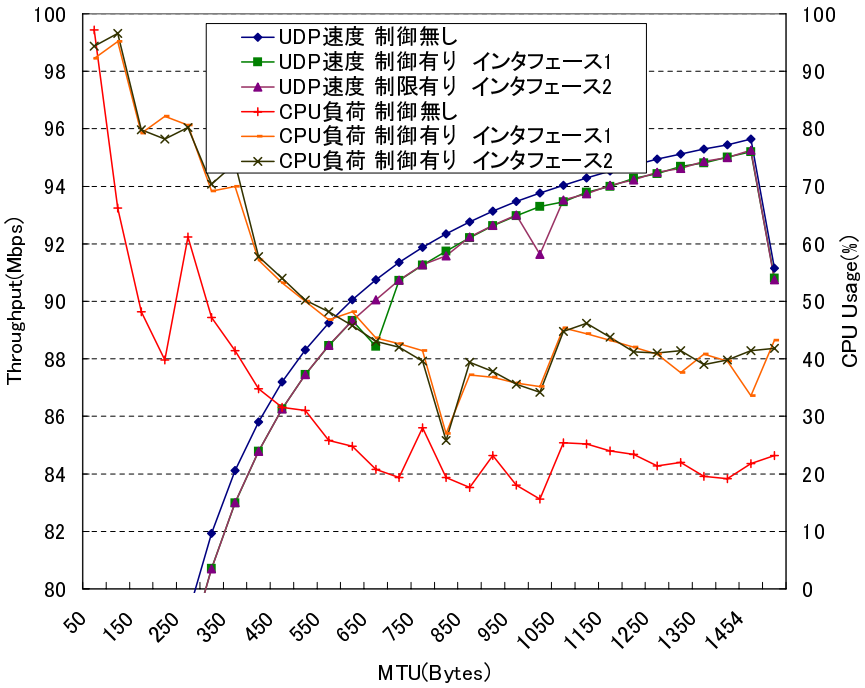

図 6.13: PC での UDP 速度 送信側

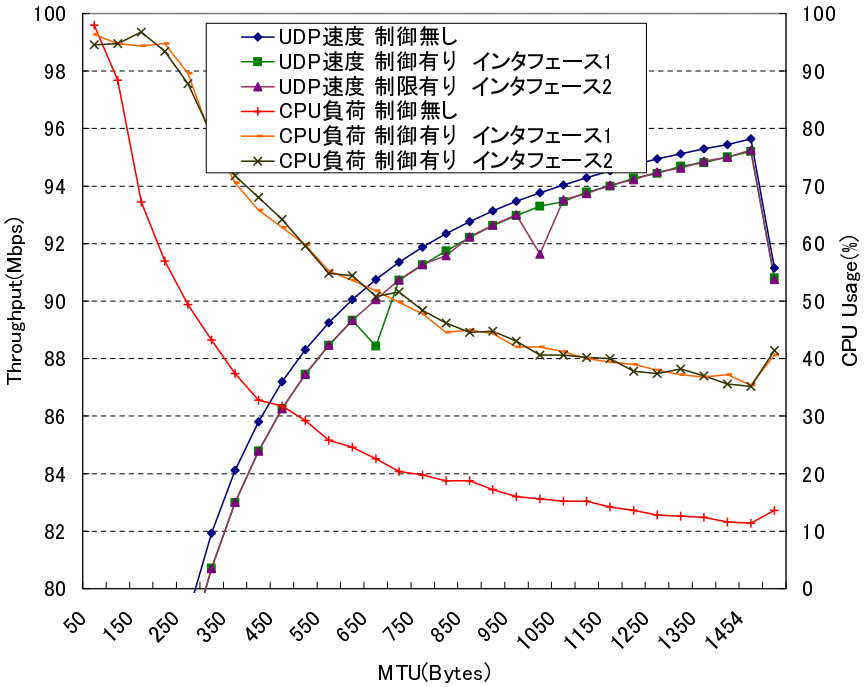

図 6.14: PC での UDP 速度 受信側

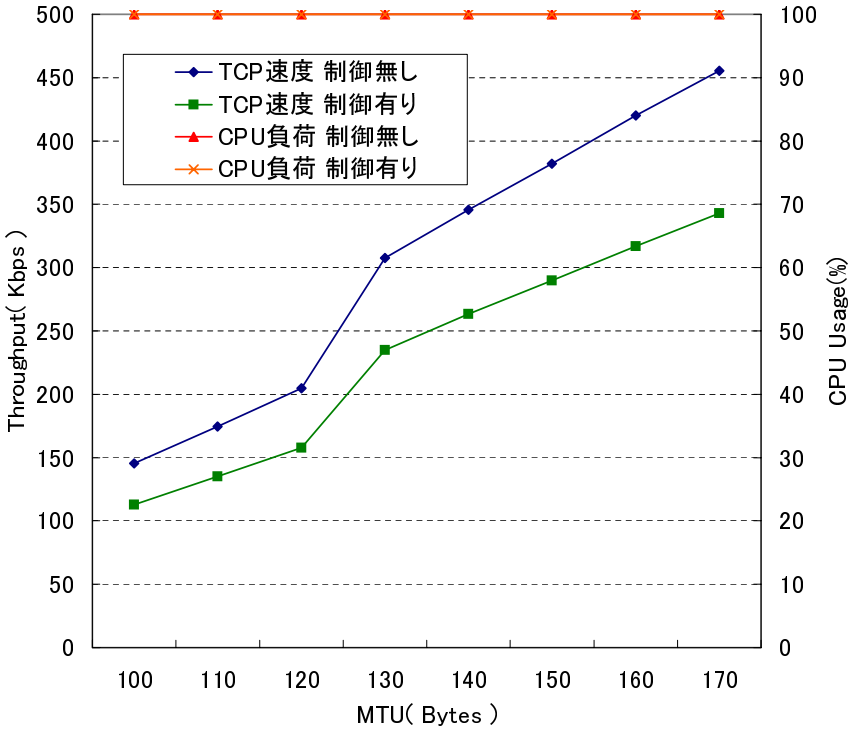

図 6.15: iPod での TCP 速度

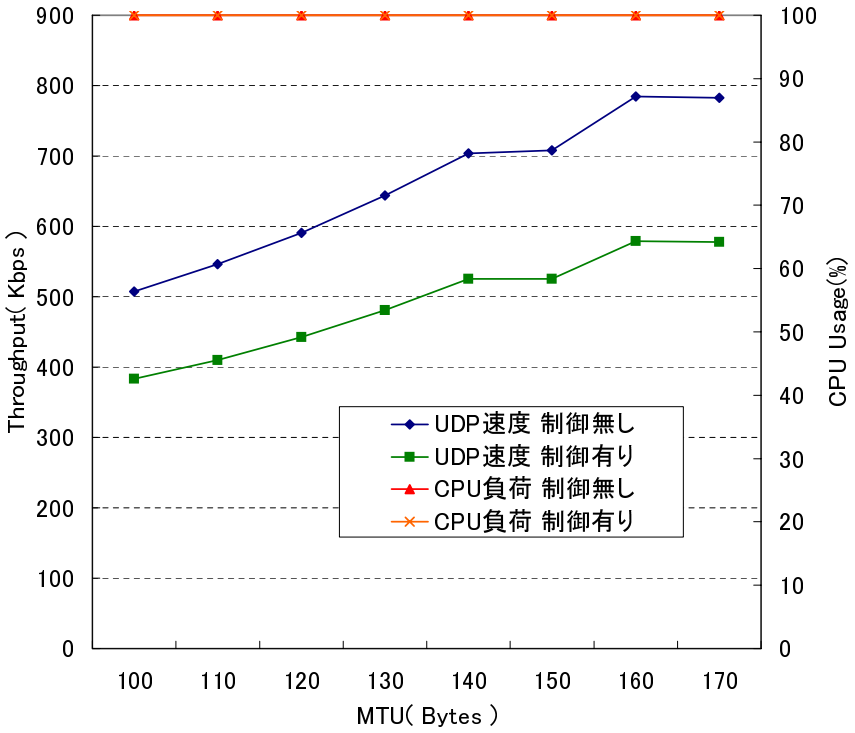

図 6.16: iPod での UDP 速度

第7章

結論

7.1　本研究の主たる成果

本研究では，利便性故に LAN 上で通信を行うデバイスが増加し，それにしたがってデバイスを選択するユーザインタフェースとデバイス間の通信を安全に行う通信保護が重要になると述べた．

第 3 章では，問題を解決する既存技術についてそれが現実的には導入が困難な状況にあることを述べ，既に標準化が進んでいる既存サービス発見機構に対するユーザインタフェースの位置づけと要求される機能を述べた．また，従来の暗号化プロトコルは，認証情報の管理の問題とコネクション単位の通信保護は実装する手間を考慮し，そのまま適用することが困難であると述べた．

第 4 章では，上記の問題に対して実空間情報に基づいてリンク層でデバイスグルーピングを行う手法を提案した．デバイスグルーピングして仮想通信網を構築し，この通信網上でサービス発見とその後のデバイス間の通信を行う．これにより，既存サービス発見機構に変更を加えることなく，サービス発見の対象となるデバイスが実空間で選択されたデバイスに限定される．加えて，デバイス間の通信も保護される．

第 5 章では，上記で設計した機構の実装を行った．第 6 章では，実装した機構の動作確認を行った．また，仮想通信網上での通信速度の評価を行った．

今後，ユビキタス社会と言われあらゆるものがネットワークに接続する環境では，様々な機能を持ったデバイスがネットワーク上に登場するであろう．ネットワークにデバイスが対応することにより，物理的な制約を受けずあらゆるデバイスと連携してより便利なサービスを提供することが可能になると思われる．また，現在では 1 つにまとまっている機能をネットワーク上で複数に分解，また連携させて利用することも考えられる．その時その場に適したデバイスを自由に組み上げ 1 つの機能を実現することも，期待されているアプリケーションだ．

しかしながら，そのような環境では膨大な数のデバイスの中からユーザが利用するデバイスを適切に選択し利用して行かなくてはならない．また，物理的境界が無くなった環境では，目に見えない攻撃に対して安全にデバイスを利用していく必要もあろう．

このような問題に対して，実空間指向のインターフェスや通信保護機構の導入は必要であろう．その形態として本研究の成果が用いられ，より豊かな生活を創造する一助となることを願っている．

7.2　今後の課題と展望

本研究における今後の展望と，それに向けて検討すべき課題について以下に述べる．

既存サービス発見機構の動作確認

現在の実装では，サービス情報を要求・広告するだけの簡易サービス発見機構を用いた．実際には，UPnP といった既存サービス発見機構をターゲットとして設計してあるため，今後 UPnP を実際にどうさせ，その動作確認を行うべきであろう．

指示モジュールの高性能化

現在の実装では，実空間上で 1 つの共通鍵をデバイスに渡すだけに止まっている．1 つの指示モジュールに 1 つの共通鍵しか保持できないため，デバイスを仮想通信網に参加・脱退させるには，異なる共通鍵を持つ複数の指示モジュールを用いる必要があった．

今後は，1 つの指示モジュールで動的に共通鍵を生成し保持する．そしてデバイスへ共通鍵を送信する場合に，どの仮想通信網に接続するか共通鍵を選択できるようにすべきであろう．

単一デバイスに複数の仮想 NIC がある場合の操作

現在の実装では，1 つの受信モジュールに 1 つの共通鍵しか与えられない．Windows では測定のために複数の仮想 NIC を作成し，それぞれに異なる共通鍵を対応づけることが可能である．しかしながら，実空間インタフェースでは，どの仮想 NIC に共通鍵を割り当てるかを指示できないため，現在の実装では必ず 1 番目の仮想 NIC に割り当てられる．

これに対して，何度か鍵を送信するとそれに応じた仮想NIC に共通鍵を対応づけることや，ディスプレイを持つデバイスであれば画面に選択肢を出すなど，共通鍵を渡す以外の操作が必要となる．

受信モジュールを内包する Ethernet アダプタ

現在では，各 OS のカーネルに共通鍵を用いた仮想通信網構築機構を実装している．この仮想通信網構築機構が行うことは，共通鍵に基づいたパケットの暗号化とフィルタリングであるため，Ethernet ケーブルの途中に介入し，直接制御を行っても良い．

そのため，受信モジュールと Ethernet アダプタを備え仮想通信網構築を行うデバイスを，LAN ケーブルの途中に設置し，制御を行うことが考えられる．この手法では，各 OS やデバイスに変更を加えることなく，実空間上の操作を反映，通信の保護が実現できるため，既存の機構へ組み込みやすい．

認証付き実空間インタフェース

現在の実空間インタフェースでは，実空間上でデバイスを選択できると，どのデバイスでも利用可能になる．しかしながら，実際には個人所有のデバイスや共有デバイス，特定の人のみ利用できるデバイスなど，デバイスへのアクセス制御を行う必要がある．

このために，各デバイスに認証のパスワードを保持し，指示モジュールと受信モジュール間でそのパスワードに基づく認証を行う．パスワードを保持する指示モジュールであれば認証に成功し処理を続行でき，失敗すれば利用できないとして共通鍵の受信を拒否するといった認証機構を実空間インタフェースに備えるべきであろう．

謝辞

修士論文研究を進める過程で，知識はもちろんのこと，研究に対する心構えなども御教授くださり，また打ち合わせのみならず常日頃から有益な御指導，御批評，御鞭撻いただいた青山友紀教授，森川博之助教授に深く感謝致します．渡邉廣次助手、川北敦子秘書、宮島史子秘書にはすばらしい研究環境を提供していただき感謝しております。

特に，本研究の全ての面において多大な御指導と御協力をいただきました猿渡俊介氏，金子晋丈氏，打ち合わせや日頃の議論などにおいて数々の貴重な御助言をくださった川原圭博氏，今泉英明氏，三村和氏，に深く感謝します．そして，本研究の上で日頃から励まし，議論に参加してくださった，岡敏生氏，松本延孝氏，川西直氏，鈴木誠氏，小澤政博氏に厚く御礼申し上げます

また、青山・森川研究室の諸先輩方には常日頃から励ましていただき、充実した研究生活を送ることができました。この場を借りて御礼を申し上げます。最後に研究生活ともにした同輩の丸山達也君，チャンフェイさん，オクゼウク君，周欣さん，平井肇君，堀江信吾君，木田信雄君，水野浩太郎君，Jean Olivier Caron 君と研究室での苦楽をともに過ごせたことに心より感謝いたします。

2006 年 2 月 3 日

参考文献

[1] Y. Iwasaki, N. Kawaguchi, and Y. Inagaki. “touch-and-connect: A connection request framework for ad-hoc networks and the pervasive computing environment”. *In Proceedings of 1st IEEE International Conference on Pervasive Computing and Communications*, pp. 20–29, March 2003.

[2] N. Kohtake, J. Rekimoto, and Y. Anzai. “infopoint: A device that provides a uniform user interface to allow appliances to work together over a network”. *Personal and Ubiquitous Computing*, Vol. 5, No. 4, pp. 264–274, 2001.

[3] Y. Ayatsuka and J. Rekimoto. “transticks: Physically manipulatable virtual connections”. *Proceedings of Computer-Human Interaction(CHI2005)*, pp. 251–260, 2005.

[4] J. Rekimoto. “pick-and-drop: A direct manip-ulation technique for multiple computer environments”. *Proceedings of User Interface Software and Technology(UIST'97)*, pp. 31–39, 1997.

[5] A. Frier, P. Karlton, and P. Kocher. *“The SSL 3.0 Protocol”*. Netscape Communications Corp., November 1996.

[6] T. Dierks and C.A llen. “the tls protocol version 1.0”, January 1999. http://www.ietf.org/rfc/rfc2246.txt.

[7] 猿渡俊介, 森川博之, 青山友紀. “シングル cpu で実現される無線センサノードの実装”. 電子情報通信学会ソサイエティ大会.

[8] *“UPnP Forum”*. http://www.upnp.org/.

[9] *“Apple”*. http://www.apple.com/.

[10] *“MaxiVista Multi Monitor Software”*. http://www.maxivista.com/.

[11] *“Digital Living Network Alliance”*. http://www.dlna.org/home/.

[12] *“OSGI Alliance”*. http://www.osgi.org/.

[13] *“Jini Network Technology”*. http://www.sun.com/software/jini/.

[14] *“HAVi”*. http://www.havi.org/.

[15] *“ECHONET CONSORTIUM”*. http://www.echonet.gr.jp/.

[16] *“Home PNA”*. http://www.homepna.org/.

[17] *"HomePlug"*. http://www.homeplug.org/.

[18] T. Bray, J. Paoli, C. Sperberg-McQueen, and E. Maler. "extensible markup language (xml) 1.0 (second edition)", October 2000. http://www.w3.org/TR/REC-xml.

[19] "information processing - text and office systems - standard generalized markup language (sgml)". International Organization for Standardization (ISO), October 1986. ISO 8879.

[20] J. Cohen, S. Aggarwal, and Y. Goland. "general event notification architecture base: Client to arbiter ", September 2000. http://www.upnp.org/download/draft-cohen-gena-client-01.txt.

[21] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Winer. "simple object access protocol (soap) 1.1", May 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508.

[22] Y. Goland, T. Cai, P. Leach, Y. Gu, and S. Albright. "simple service discovery protocol/1.0 operating without an arbiter", October 1999. http://www.upnp.org/download/draft_cai_ssdp_v1_03.txt.

[23] 南正輝, 森川博之, 青山友紀. "ユビキタス環境におけるサービス合成支援のためのインタフェース指向ネームサービス"". 電子情報通信学会論文誌, Vol. J86-B, No. 5, pp. 777–789, May 2003.

[24] M.Minoh and T. Kamae." networked appliance and their peer to peer architecture amiden ". *IEEE Communications Magazine*, Vol. 39, No. 10, pp. 80–84, 2001.

[25] 大澤亮, 村上朝一, 中西健一, 高汐一紀, 徳田英幸. "ez dev:位置情報を利用しユーザのデバイス利用を支援するアプリケーション". 情報処理学会 全国大会.

[26] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. "the anatomy of context-aware application". *MOBICOM'99*, pp. 59–68, 1999.

[27] 綾塚祐二, 松下伸行, 暦本純一. "実世界指向ユーザインタフェースにおける「見ているものに接続する」というメタファ". 情報処理学会論文誌, Vol. 42, No. 6, pp. 1330–1337, June 2001.

[28] J. Rekimoto, Y. Ayatsuka, M. Kohno, and H. Oba. "proximal interactions: A direct manipulation technique for wireless networking". *n Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)*, pp. 511–518, September 2003.

[29] G. Suzuki, S. Aoki, T. Iwamoto, D. Maruyama, N. Kohtake T. Koda, K. Takashio, and H. Tokuda. "u-photo: Interacting with pervasive services using digital still images". *The 3rd International Conference on Pervasive Computing (Pervasive 2005)*, May 2005.

[30] 木原民雄. "実写映像の多人数操作による情報ナビゲーションシステム". マルチメディア，分散，協調とモバイル（DICOMO 2002）シンポジウム論文集, pp. 9–12, 2002.

[31] T. Hoshino, Y. Horii, Y. Maruyama, A. Katayama, Y. Shi-bata, and T. Yoshimaru. "airreal: Object-oriented user interface for home network system". *Workshop on InteractiveSystems and Software(WISS 2001)*, pp. 113–118, 2001.

[32] 瀬戸洋一. “ユビキタス時代の情報セキュリティ技術”. 日本工業出版.

[33] S. Kent and R. Atkinson. “security architecture for the internet protocol”, November 1998. http://www.ietf.org/rfc/rfc2401.txt.

[34] M. Elkins. “mime security with pretty good privacy (pgp)”, October 1996. http://www.faqs.org/rfcs/rfc2015.html.

[35] J. Jonsson and B. Kaliski. “public-key cryptography standards (pkcs) #1: Rsa cryptography specifications version 2.1”, February 2003. http://www.ietf.org/rfc/rfc3447.txt.

[36] R. Khoussainov and A. Patel. “lan security: problems and solutions for ethernet networks”. *Computer Standards & Interfaces*, Vol. 22, pp. 191–2002, August 2000.

[37] 三村和, 飛岡良明, 森川博之, 青山友紀. “サービス指向グルーピング機構を用いたユーザ主導ネットワークの構築”. 第 13 回マルチメディア通信と分散処理（DPS）ワークショップ, pp. 290–295, November 2005.

[38] *“Apple - iPod family”*. http://www.apple.com/ipod/.

[39] *“wikiPodLinux”*. http://ipodlinux.org/Main_Page/.

[40] 猿渡俊介, 森川博之, 青山友紀. “ユーザによる制御が可能なセンサ-アクチュエータネットワーク技術の設計”. 電子情報通信学会技術研究報告, SN2006-1, 2006.

[41] 堀江信吾, 猿渡俊介, 倉田成人, 森川博之, 青山友紀. “無線センサネットワークを用いた地震モニタリングにおける同期性能の評価”. 電子情報通信学会技術報告, 第 2 回センサネットワーク研究会, June 2005.

[42] 倉沢央, 川原圭博, 森川博之, 青山友紀. “単一の無線加速度センサを用いたユーザコンテキストの推定”. 電子情報通信学会ソサイエティ大会, B-19-23, September 2005.

[43] M. Minami, T. Morito, H. Morikawa, and T. Aoyama. “solar biscuit: A battery-less wireless sensor network system for environmental monitoring applications”. *In Proceedings of the 2nd International Workshop on Networked Sensing Systems*, June 2005.

[44] *“Network Driver Interface Specification”*. http://www.microsoft.com/whdc/device/network/ndis/default.mspx.

[45] *“Visual Studio”*. http://www.microsoft.com/japan/msdn/vstudio/.

[46] *“Windows Driver Development Kit”*. http://www.microsoft.com/whdc/devtools/ddk/default.mspx.

[47] *“TeraTermPro”*. http://hp.vector.co.jp/authors/VA002416/.

[48] *“Ethereal: A Network Protocol Analyzer”*. http://www.ethereal.com/.

[49] *“Netperf Homepage”*. http://www.netperf.org/netperf/NetperfPage.html.

発表文献

[1] 小森田 賢史, 金子 晋丈, 森川 博之, 青山 友紀, “CommoNet: 基地局間連携による自律分散的マイクロモビリティサポート”, 電子情報通信学会総合大会, B-6-65, March 2004.

[2] 小森田 賢史, 金子 晋丈, 森川 博之, 青山 友紀, “自律分散的マイクロモビリティサポートのための基地局間マルチホップ網”, 電子情報通信学会技術研究報告, MoMuC2004-26, May 2004.

[3] 小森田 賢史, 森川 博之, 青山 友紀, “至近距離通信デバイスを用いた視覚的操作による通信指示機構”, 電子情報通信学会ソサイエティ大会, B-6-80, September 2005.

[4] 小森田 賢史, 森川 博之, 青山 友紀, “視覚的操作による通信制御機構の設計と実装”, 電子情報通信学会総合大会, B-6-10, March 2006.

[5] 小森田 賢史, 森川 博之, 青山 友紀: “実空間操作に基づくリンク層におけるデバイスグルーピング機構”, 電子情報通信学会技術研究報告, IN2006-03, March 2006.

[6] 金子 晋丈, 小森田 賢史, 森川 博之, 青山 友紀, “セッション層モビリティサポートにおける高信頼データ転送の性能評価,” 電子情報通信学会ソサイエティ大会, B-6-40, September 2003.

[7] 中島 亮, 小森田 賢史, 森川 博之, 青山 友紀, “CommoNet における基地局探索及び通信保護機構,” 電子情報通信学会総合大会, B-6-36, March 2005.

[8] 後郷 和孝, 神谷 弘樹, 渋井 理恵, 金子 晋丈, 玉 載旭, 小森田 賢史, 藤巻 聡美, 寺岡 文男, “リンク層情報を利用したネットワーク層主導高速ハンドオーバ機構の設計と実装,” 電子情報通信学会技術研究報告, MoMuC2005-3, May 2005.

[9] 寺岡 文男, 後郷 和孝, 神谷 弘樹, 渋井 理恵, 金子 晋丈, 小森田 賢史, 玉 載旭, “移動通信装置およびプログラム,” 特願 2005-151690, May 2005.