平成20年度　博士論文

邦文題目

安全性と移動性を両立する柔軟なグループ通信アーキテクチャに関する研究

英文題目

A Study on Flexible Group Communication Architecture Compatible with both Security and Mobility

電気電子・情報・材料工学専攻

(学籍番号: 063441506)

鈴木　秀和

提出日: 平成20年11月7日

名城大学大学院理工学研究科

内容要旨

ユビキタスネットワークを実現するためには，暗号化通信，移動通信，エンドツーエンド通信を同時に，かつ容易に実現できるアーキテクチャが要求される．先行研究の多くは，IETF（Internet Engineering Task Force）により標準化された IPsec や Mobile IP を導入することを想定している．暗号化通信を実現する IPsec は強靭なセキュリティを提供できるが，NAT（Network Address Translator）やファイアウォールとの相性が悪く，スループットが低下するという課題がある．また，ノードの位置の変化に応じて暗号化通信に必要なセキュリティポリシを変更する必要があり，多大な管理負荷が発生する．これは IPsec の複雑な仕様に起因しており，専門的知識を持たない一般ユーザが使用することは困難である．移動通信を実現する Mobile IP はノードの位置を管理する特殊なサーバをインフラに整備する必要があり，通信を中継しなければならないため，エンドツーエンド通信の考え方と矛盾する．また，カプセル化転送処理を行うためスループットの低下が課題となっている．IPv4 ネットワークにおいてエンドツーエンド通信を実現するためには NAT 越え問題を解決する必要がある．既存技術はアプリケーションに依存するため，ユビキタスネットワーク環境では適していない．そのため，先行研究の多くはエンドツーエンド通信を実現できる IPv6 ネットワークを基盤とし，その上で暗号化通信と移動通信を実現するアプローチを採用している．しかし，IPv6 ネットワークは当初の想定とは異なり，ほとんど普及していない．また，IPv4/IPv6 環境が当分の間混在することが想定されているため，IPv4 ネットワークにおいて暗号化通信，移動通信，エンドツーエンド通信を実現することは意義がある．

本研究では暗号化通信，移動通信，エンドツーエンド通信を同時に実現できるネットワークの概念として，FPN（Flexible Private Network）を提唱する．FPN では暗号化通信を実現するためにセキュア通信グループを構築する．FPN 環境下におけるノードは複数のセキュア通信グループに帰属することが可能で，同一グループのメンバ間の通信は暗号化される．異なるグループのメンバとの通信は破棄したり平文のまま通信を行ったりすることが可能である．さらに，ユビキタスネットワークを実現するために必要な機能として，FPN は「位置透過性」,「移動透過性」,「アドレス空間透過性」を有する．これにより，暗号化通信，移動通信，エンドツーエンド通信に必要な情報をユーザが設定するのではなく，システムが自律的にネットワーク構成の変化を学習し，設定情報を動的に生成することができる．

本論文は FPN を実現するために必要な個々の要素技術について提案し，それらの成果を新たなグループ通信アーキテクチャとして取りまとめたものである．本論文は以下の構成からなる．

1 章 では本研究の背景および目的を示す．本研究の主題である FPN を明確に定義し，それを実現するためのグループ通信アーキテクチャとして GSCIP（Grouping for Secure Communication for IP；ジースキップと呼ぶ）を提案する．GSCIP は 2 章 以降で提案する 3 つのプロトコルとオリジナルの暗号化通信方式から構成される．GSCIP ではセキュア通信グループと暗号鍵を 1 対 1 に対応づけることにより，セキュア通信グループの定義を行う．これにより，ノードの位置が変化しても IP アドレスに依存することなくセキュア通信グループの関係を維持することができる．GSCIP アーキテクチャを導入したシステムにおいて，通信相手ノードを認証する方法から暗号化通信を

行うまでの一連の流れを示す．また本論文の構成を示し，GSCIP を構成する種々のプロトコルや 3 つの透過性との関係性を示す．

2 章 では位置透過性を実現する動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol）を提案する．DPRP は GSCIP アーキテクチャの主要プロトコルとして位置づけられる．ノードは通信開始時に相手ノードが同一セキュア通信グループのメンバか確認し，暗号化通信に必要な動作処理情報を動的に生成する．ノードの位置が変化しても通信開始時に動的かつ高速に動作処理情報を再生成する仕組みを示す．また，FPN を GSCIP/DPRP と IPsec/IKE（Internet Key Exchange）により実現するために必要なコストを比較する．導入時に発生する初期管理コスト，ユーザが移動してネットワーク構成が変化した場合に発生するコスト，およびセキュア通信グループのメンバを追加した場合に発生するコストを詳細に算出し，提案アーキテクチャがユーザの管理負荷を大幅に軽減できることを示す．

3 章 では NAT やファイアウォールを通過でき，かつ高スループットを実現できる暗号化通信方式 PCCOM（Practical Cipher Communication）を提案する．PCCOM は DPRP により生成された動作処理情報に基づき，通信パケットの暗号化を行う．ホームネットワークとインターネットの間に設置される NAT やファイアウォールを通過できるため，IPv4 ネットワークにおけるノード間の通信をエンドエンドで暗号化することを可能とした．パケットフォーマットを変えないまま，本人性確認とパケットの完全性保証を実現し，かつ IPsec に対して高スループットを維持できること示す．また，IPsec と PCCOM が有効な適用環境について議論し，両者のすみ分けが可能であることを示す．

4 章 では移動透過性を実現する Mobile PPC（Mobile Peer-to-Peer Communication protocol）を提案する．Mobile PPC はエンドツーエンド通信を基盤とし，ノードの移動前後の IP アドレスの対応関係を通信ノード間で共有する．その後はノードの IP 層においてアドレス変換処理を実施することにより，IP アドレスの変化を隠蔽して通信を継続することが可能となる．既存技術の Mobile IP に対して，低遅延・高スループットを実現でき，かつ段階的な普及が可能であることを示す．また，IPv4 ネットワークにおいて移動通信を実現する場合に必要な機能や，実運用する際の障害となる問題点を整理し，その解決方法について議論する．

5 章 ではアドレス空間透過性を実現する外部動的マッピング方式と NAT-f（NAT-free protocol）を提案する．NAT 外部のノードは NAT 配下のプライベートネットワークに存在する内部ノードを仮想的に識別し，通信開始時に NAT と協調することによりに内部ノードとの通信に必要なマッピング情報を動的に生成する．外部ノードは IP 層において，内部ノード宛の通信パケットの宛先を仮想 IP アドレスから NAT にマッピングされたアドレスとポート番号にアドレス変換することにより，NAT 越え通信を可能とした．提案方式はノードと NAT 間だけでなく，ホームネットワーク間の異なる内部ノード同士の通信にも適用可能である．既存技術に対して，アプリケーションに依存せず，特殊なサーバも必要なく，高い汎用性を有することを示す．また，プロトタイプシステムを構築して性能評価を行い，エンドツーエンド通信のスループットを維持できることを示す．

6 章 では GSCIP アーキテクチャの応用研究の一例として，NAT-f と Mobile PPC を融合した新たな移動通信の実現手法について論じる．提案手法は NAT-f を応用することにより，従来の移動透

過性研究の考え方とは逆に通信相手ノードがプライベートネットワークに存在する場合の移動透過性を実現する．NAT-f と Mobile PPC はアドレス変換処理に基づくアーキテクチャであり，かつ異なる処理タイミングで動作するため，容易に機能を統合することが可能である．プロトタイプシステムの評価及びセキュリティや対応可能な通信ケースに関する考察を行う．また，提案アーキテクチャがホストモビリティに限らず，ネットワーク単位の移動性を実現するネットワークモビリティや，既存の移動透過性技術である Mobile IP など様々なシステムに応用可能であることを示す．

最後に 7 章 で本研究を総括し，今後の課題を示す．

目 次

図 目 次

表目次

第1章　序論

1.1　研究の背景

我が国は 2004 年に u-Japan 構想を発表し，世界最先端レベルの ICT 国家となることを目標としている [1]．なかでも，ICT 分野における将来の期待は，ユビキタスネットワーク技術に集まっている．ユビキタスネットワークとは，「いつでも，どこでも，何でも，誰でもアクセスが可能なネットワーク環境」として定義されている [2]．ユビキタスネットワークの実現により，コンピュータのみならず，デジタルテレビや冷蔵庫などの情報家電機器や，IC タグやセンサが付与されたあらゆるモノが相互に接続して，情報を共有することが可能になる．このようなネットワークは，従来のクライアントサーバモデルに基づくネットワークアーキテクチャだけではなく，エンドツーエンドあるいはピアツーピアモデルなどのネットワークアーキテクチャが重要になる．ネットワークに接続するノードが小型化し，携帯性が高まることにより，子供から老人までのあらゆるユーザが利用することが想定されている．従って，様々なサービス展開が期待できユーザの利便性が高まる一方，セキュリティやプライバシの保証が大きな課題となる．また，携帯性の向上により，ユーザは家庭，外出先，職場などを移動しながらでもネットワークに接続できるため，モバイル通信が一層普及することが期待される．

上記のようなユビキタスネットワークを実現するためには，3 種類の通信，すなわち暗号化通信，移動通信，エンドツーエンド通信を同時に行うことが重要である．暗号化通信技術は，アプリケーションレベルで実現するものとネットワークレベルで実現するものに分類できる．

アプリケーションレベルで実現する技術は，ユーザが使用するアプリケーションごとに暗号化機能を実装する方式である．代表的な技術として，HTTP 通信には SSL（Secure Socket Layer）/TLS（Transport Layer Security） [3]，リモートログインには SSH（Secure Shell） [4–8]，メールには S/MIME（Secure/Multipurpose Internet Mail Extensions） [9, 10] や PGP（Pretty Good Privacy） [11] などがある．これらはユーザが設定なしに利用できるものが多く，通信が暗号化されているかどうかをアプリケーションに表示できるため，ユーザが容易に確認できる．また，暗号化範囲がトランスポートヘッダ以降のデータ部となるため，ファイアウォールによる制御が容易であるなどの利点がある．しかし，アプリケーション個々に実装が必要であるため，暗号化機能を実装しないアプリケーションの通信を保護することはできない．また，暗号化範囲に IP ヘッダとトランスポートヘッダが含まれないため，送信元および宛先情報の秘匿や正当性を保証することができないなどの課題がある．ユビキタスネットワークでは様々なアプリケーションが利用されることを鑑みると，アプリケーションに依存しないネットワークレベルの暗号化技術が望ましい．

ネットワークレベルの代表技術として，IPsec [12] がある．IPsec は IP 層に実装されており，全

ての IP 通信を暗号化することができ，高いセキュリティ強度を有している．また，様々な利用形態に対応できるよう高い汎用性を持つなどの利点がある．しかし，IPsec ではセキュリティポリシとセキュリティアソシエーション情報をノードに設定する必要があり，これらの設定はかなり複雑なものとなっている．例えば，送信元および宛先の IP アドレスや上位プロトコルの番号など，ネットワーク環境や通信相手ごとに異なる多くの情報を指定しなければならないため，専門知識のないユーザが利用することは難しい．そのため，多くの OS やネットワーク機器に実装はされているものの，現状では拠点間を結ぶ VPN（Virtual Private Network）[13] の構築に利用されている程度で，広く普及していない．

インターネットで利用されている通信プロトコルである TCP/IP は，IP アドレスとポート番号およびプロトコル番号の情報によりノード間の通信を識別している．IP アドレスはノード識別子としての役割だけでなくネットワーク上における位置の情報も含んでいるため，ノードがネットワークを移動すると異なる IP アドレスが割り振られる．そのため，通信中に移動して IP アドレスが変化すると，TCP/IP では別の通信として見なされて通信を継続することができない．移動通信を実現するためには上記課題を解決する技術が必須であり，これまで多くの方式が研究されている [14].

移動通信を実現する代表的な技術として，Mobile IP [15, 16] がある．Mobile IP では移動ノード MN（Mobile Node）の識別を行う Identifier とノードの位置を示す Locator を分離し，それらの対応を管理する HA（Home Agent）をネットワークに設置する．MN は Identifier としてユニークなホームアドレス HoA（Home Address）を保持し，移動先ネットワークで割り当てられる気付けアドレス CoA（Care-of Address）を Locator として用いる．通信相手ノード CN（Correspondent Node）は MN の位置にかかわらず宛先を MN の HoA として通信を行う．MN が通信中に別のネットワークに移動すると，HA に対して自身の HoA と取得した CoA のマッピングを登録する．以後，CN からの通信は HA が代理受信し，MN へトンネル転送される．このように，Mobile IP では Identifier と Locator の分離と HA による通信の中継処理により，MN が移動しても通信を継続できる．

Mobile IP は IETF（Internet Engineering Task Force）での十分な検討を経て確立された技術であり，3GPP（Third Generation Partnership Project）[17] ではキャリア主導でノードのハンドリングを行う PMIP（Proxy Mobile IP）[18, 19] が，また WiMAX（Worldwide Interoperability for Microwave Access）[20] では PMIP に加えてクライアント側で移動通知処理を行う CMIP（Client Mobile IP）も採用されている．しかし，HA という特殊な装置が必要である他，通信経路に冗長が発生したり，トンネリング転送時に余分なヘッダが必要になるなどの問題点があり，ユビキタスネットワークで求められるエンドツーエンド通信の特徴や利点と矛盾する点がある．

IPv4 ではグローバル IP アドレス枯渇問題の対策として NAT（Network Address Translator）[21] が導入され，企業ネットワークやホームネットワークにはプライベートネットワークを構築する形態となった．NAT 配下に存在するノードには，プライベートネットワーク内でのみ有効なプライベート IP アドレスが割り当てられ，それらのノードが NAT 外部のノードと通信を行う場合は NAT においてプライベート IP アドレスからグローバル IP アドレスに変換する必要がある．この変換処理に必要なマッピングテーブルは，プライベートネットワークからグローバルネットワー

クに向けてパケットが送信される際に生成される．従って，グローバルネットワーク側のノードからプライベートネットワークのノードに対して通信を開始することができない．この結果，双方向接続性が失われエンドツーエンド通信を実現することが困難となった．

上記問題は NAT 越え問題として広く知られており，これを解決する様々な技術が研究されている [22–26]．NAT 越え技術の多くは個々のアプリケーションに機能を実装し，予めインターネット上のサーバと連携することにより NAT に対してマッピングテーブルを生成しておく．グローバルネットワーク上のノードは，生成されたマッピングテーブルの情報を通信相手から直接通知してもらう，またはインターネット上のサーバからマッピング情報を取得することにより，NAT 配下のノードに対して通信を開始することができる．しかし，アプリケーションに依存した解決手法であるため，ユーザが利用したいアプリケーションが対応していない場合は，自由な双方向通信を実現できないなどの課題がある．

上記のような NAT 越え問題は，NAT の利用が必要不可欠である IPv4 ネットワークで生じる問題である．また，移動通信を実現するためにはグローバルユニークな Identifier を全ノードに割り当てる必要があるが，グローバル IP アドレスの数が限定されている IPv4 ネットワークでは実現の可能性は極めて低い．ユビキタスネットワークではあらゆるノードがネットワークに接続するため，今以上にグローバル IP アドレスが必要となる．そこで IPv4 の基本的な理念を踏襲しつつ，広大なアドレス空間に基づいた新しいネットワークプロトコルとして研究開発された IPv6 [27] が注目されている．

これまでのユビキタスネットワークの実現を目的とした先行研究は，インターネットに接続する全てのノードにグローバルアドレスを提供でき，エンドツーエンド通信が可能な IPv6 を前提としたものがほとんどである．IPv6 では，暗号化通信方式としてネットワークレベルのセキュリティ技術である IPsec が標準化されている．また，移動通信を実現する方式としては Mobile IPv6 [16] が標準化されており，ユビキタスネットワークで求められる 3 つの通信を実現できる技術的基盤が整っている．しかし，IPv6 サービスへの取り組みは当初の想定とは異なり，ほとんど進んでいないのが現状である [28]．先行研究は既に方式としては確立されているものの，実ネットワーク環境においてサービス展開および運用するまでに至っていない．また，仮に IPv6 が中心の世の中に移行したとしても，IPv4 との互換性がないため，IPv4/IPv6 の共存・混在環境が相当継続することが想定されている．そのため，IPv6 は勿論，IPv4 であってもユビキタスネット社会の恩恵を享受できるようにすることは，ユビキタスネットワークのキーワード「誰でも，何でも」を実現化するために，ひいては u-Japan 戦略を確実に推進する上で重要であると考えられる．

1.2 研究の目的と実現アプローチ

上記のような背景から，本研究では IPv4 を中心として暗号化通信，移動通信，エンドツーエンド通信の異なる 3 つの通信を同時に実現できるユビキタスネットワークの構築を目的とする．IPv4 において暗号化通信および移動通信を実現するために，既に存在する IPsec と Mobile IPv4 [15] の技術を利用することは可能である．しかし，これらの技術は通信性能の低下が大きく，NAT やファ

イアウォールとの相性が悪いという課題がある．また，専用装置によるインフラ整備が必要であったり，エンドツーエンド通信の利点を阻害するなどの問題がある．さらに複雑な設定を手動で行わなければならず，ユーザ間の通信にはほとんど普及していない．エンドツーエンド通信を実現するためには，NAT 越え技術が必須であるが，既存の研究はその多くがアプリケーションに依存した解決策であり，ユビキタスネット社会で想定される多種多様なアプリケーションに対応することは難しい．

これまでに，IPv4 ネットワークを基盤として暗号化通信，移動通信，NAT 越え技術のうち，2 種類の技術を組み合わせた研究は行われているが，すべての技術を同時に実現する試みはなされていない．そこで，本論文におけるユビキタスネットワークの要求仕様を以下のように設定し，3 つの通信を同時に実現するための新たな通信アーキテクチャを設計する指針とする．

要件 1: 高セキュリティと低管理負荷の両立

一般にセキュリティの向上を図ることによって，ネットワークシステムの運用や管理が難しくなる傾向がある．必要十分なセキュリティ強度を確保しつつ，ユーザが行うべき設定作業を削減し，管理負荷を低減する．

要件 2: ノードの位置に依存しない柔軟性

通信開始側および通信相手側のノードは，IPv4/IPv6 グローバルネットワークや NAT 配下に構築されている IPv4 プライベートネットワークに存在することが想定される．また，ノードは移動することも可能であるため，ノードの位置に依存しない柔軟性を実現できるシステム設計を行う．

要件 3: アプリケーションに依存しない汎用性

ユビキタスネットワークではあらゆるノードがインターネットに接続し，多種多様なアプリケーションが利用されることが想定される．暗号化処理や認証処理をアプリケーションごとに実装するのではなく，ネットワークレベルで実装することにより，アプリケーションに依存しないセキュリティ機能を提供する．

要件 4: 特殊なサーバの導入回避

エンドツーエンド通信を実現するために，できる限りノード間で所要の処理を完結させ，特殊なサーバの導入を回避する．サーバが必要となる場合は，既存ネットワークで運用されている装置を積極的に活用する．

要件 5: 低遅延・高スループット

パケットのカプセル化処理や特殊なサーバによる中継処理を行わないことにより，通信開始時に発生する通信遅延を抑えつつ，高スループットを実現する．

本研究の実現アプローチとして，まず上記要件を満たすセキュア通信ネットワークを設計する．ここで，セキュア通信ネットワークとは不正侵入，データの盗聴や漏洩，改竄などの様々なセキュリティ脅威から保護されたネットワークである．セキュア通信ネットワークを構築する方法とし

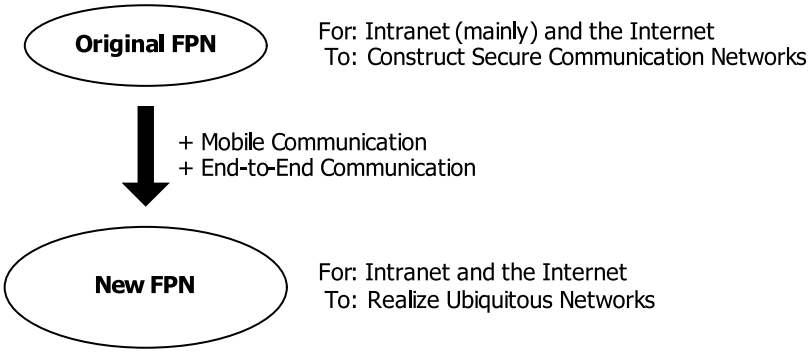

図 1.1　フレキシブルプライベートネットワークの拡張

て，VPN がある．VPN は通信キャリアが保有するバックボーンネットワークやインターネットに仮想的な通信路を確立し，企業内ネットワークの各拠点間やノードとホームネットワーク間を接続するシステムである．VPN は主にカプセル化と暗号化の機能から構成され，拠点間を接続する VPN を実現する IPsec を用いた方法では，ドメイン単位のセキュア通信ネットワークを構成できる．しかし，企業では部門単位の業務グループと部門横断の個人単位の業務グループが混在することがあるため，複数の業務グループとユーザを対応づけできることが望まれるが，VPN ではこのようなきめ細かいセキュア通信ネットワークを構築することは困難である．

そこで，企業ネットワークにおいて容易に多重帰属可能なセキュア通信グループを構築するシステムとしてフレキシブルプライベートネットワーク FPN（Flexible Private Network）が提案されている [29]．FPN とは単一の暗号鍵により定義されるセキュア通信グループを意味し，IP サブネットワークから独立して構築することができる特徴がある．このようなセキュア通信グループの構築手法は，文献 [30] において提案されている．この構築手法によると，ノードの位置，すなわち IP アドレスに依存することなくセキュア通信グループの関係を維持することができるため，ネットワーク構成の変化時に発生する管理負荷を抑制することができる．文献 [29] では企業ネットワークにおいて FPN を実現する方法を提案しているが，本研究では図 1.1 に示すように FPN の考え方を基本し，かつ FPN の適用範囲をホームネットワークおよびインターネットまで拡張する．さらに，ユビキタスネットワークを実現するために必要な要素技術，すなわち移動通信とエンドツーエンド通信を実現する技術を追加し，新たなネットワークの概念として明確に定義する．

1.3　フレキシブルプライベートネットワークの定義

本研究における FPN とは，安全性と柔軟性を両立させたネットワークの概念であり，ユビキタスネットワークのあるべき姿を示したものと定義する．図 1.2 に FPN の概念を示す．FPN では個人単位とドメイン単位の要素が混在する環境に対してセキュア通信グループの定義ができる．同一セキュア通信グループに属するノード間の通信はその安全性が保証され，異なるセキュア通信グループに属するノードからのアクセスを拒否することができる．ノードおよびドメインは複数のセキュア通信グループに多重帰属することが可能で，個人単位やドメイン単位というグループ

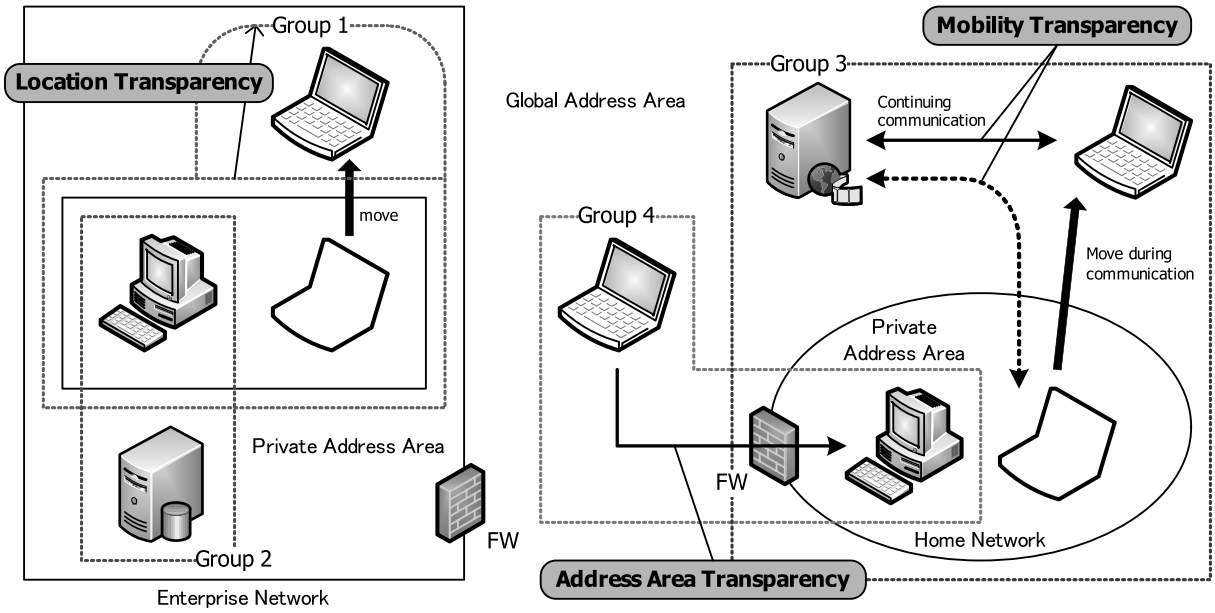

図 1.2　FPN の概念

の違いを意識する必要はない．またセキュリティドメインが階層的に構築されていたり，セキュリティドメイン内に異なるセキュア通信グループに属するノードが存在するような環境（多段構成ネットワーク）であってもかまわない．

FPN はこのようなネットワーク環境を前提とし，更に以下に示す位置透過性，移動透過性，アドレス空間透過性を実現したものである．

1. **位置透過性（Location Transparency）**
   ノードやドメインは移動可能であり，かつノードが特定のドメインの内外を往復するなどしてネットワーク構成が変わっても，予め定義されているセキュア通信グループの関係は維持される．このとき設定情報をネットワーク管理者が更新する必要はなく，システムが自動的にネットワーク構成の変化を学習する．この機能を位置透過性と呼ぶ．位置透過性は，ノードが通信していない状態（オフライン）での移動を想定したもので，人事異動に伴う引越しや出張先から通常の業務を行えるようにするための機能である．

2. **移動透過性（Mobility Transparency）**
   ノードが通信中（オンライン）の状態において移動することもありうる．通信中に移動すると，ノードの IP アドレスが変化するため，そのままでは通信が継続できない．これは TCP コネクションや UDP ストリームを管理する情報に通信ペアの IP アドレスが含まれているためである．上位アプリケーションに対しては IP アドレスが変化したことを隠蔽して通信を継続できるようにすることが望ましい．この機能を移動透過性と呼ぶ．

3. **アドレス空間透過性（Address Area Transparency）**
   IPv4 の通信環境においては，プライベートアドレス空間とグローバルアドレス空間が存在し，現状では両者の間で自由な通信ができない．これは NAT によりプライベートアドレス空

間がグローバルアドレス空間から隠蔽されるためである．従って，グローバルアドレス空間側からプライベートアドレス空間側のノードに対して通信を開始することができない．NAT とノードが連携してアドレス空間の違いを意識することなく通信できることが望ましい．この機能をアドレス空間透過性と呼ぶ.

本研究の目的は上記のように拡張定義した FPN を実現することである．また，FPN の概念を基本とし，異なる 3 つの透過性を設定することにより，独立した個々の研究テーマの方向性を統一することが可能となる.

FPN の適用範囲としては，企業ネットワークにおけるイントラネット内部と，ホームネットワークを含むインターネット上の 2 種類を想定し，様々なシステム構成に応じて管理負荷の増加を抑えながらセキュリティの向上を図ることができる.

1.3.1 企業ネットワークにおけるイントラネット内部における FPN

イントラネットでは多段構成ネットワークになることが多く，組織変更，人事異動や出張による場所の移動等が頻繁に行われるため，FPN の概念の適用は有効である．なお，企業ネットワークとインターネットとの間には強固なファイアウォールが設置され，セキュリティポリシにより自由な通信が禁止されているため，両者をまたがる FPN の構築は想定しない.

ただし，今日のビジネスシーンでは図 1.3 のようにインターネットを利用して本社と支社のイントラネットを接続したり，ユーザが外出先からイントラネットに接続するリモートアクセスを行うことが十分考えられる．このような場合，イントラネット間またはユーザとイントラネットとの間に VPN を構築するのが一般的である．VPN を構築することにより，本社イントラネットと物理的位置が離れている支社イントラネットや，リモートユーザがあたかも本社イントラネット内部に存在しているかのように振る舞うことができる．従って，VPN により接続されたイントラネットやリモートユーザに対して，図 1.3 における Group 1 や Group 4 のように FPN を拡張し

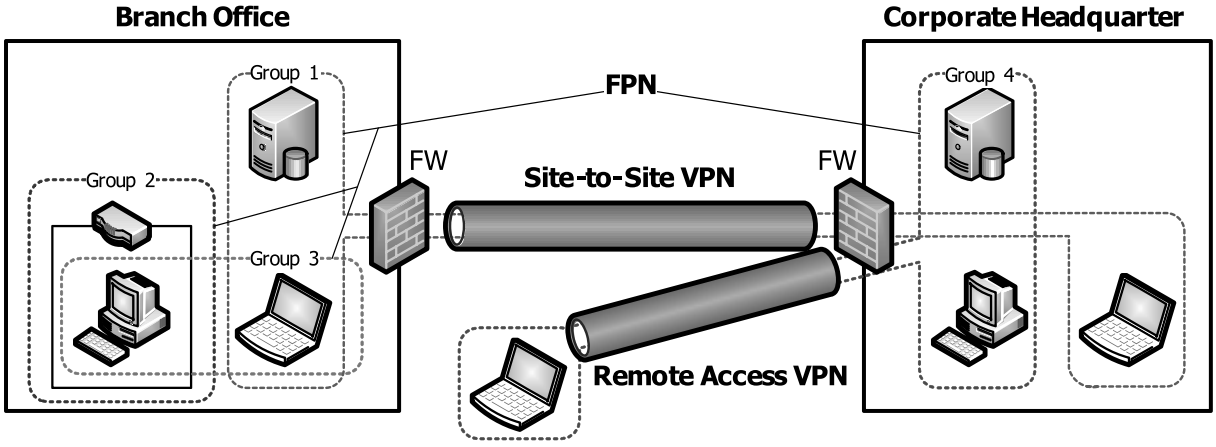

図 1.3　イントラネットにおける FPN

て構築することは十分に考えられる．このような手法は本論文の主たる内容ではないが，別研究として検討している [31].

企業ネットワークにおける FPN では，グローバルアドレス空間とプライベートアドレス空間を意識する必要がなく，アドレス空間透過性の重要度はさほど高くはない．しかし，イントラネット内の部門ネットワークに NAT を設置するような多段 NAT 構成も考えら，この様な場合に NAT 外部から NAT 内部へのサブネットワークへ通信を開始する場合は，アドレス空間透過性が求められる．

1.3.2 ホームネットワークを含むインターネット上における FPN

ホームネットワークは従来のパーソナルコンピュータに加えて，AV 機器や白物家電の他，照明機器などの住宅設備やセキュリティシステムのセンサ等の様々なデバイスが相互に接続され，協調連携するようなスタイルが見込まれている [32]．そのため，ホームネットワークには企業のような強固なファイアウォールは必要なく，ユーザは外出先からホームネットワーク内のノードのコンテンツを取得したり，ノードを制御するなどの高度なサービスを安全に利用できることが要求される．

そこで，ホームネットワークとインターネットをまたがる FPN の構築を想定する．プライベートアドレス空間のホームネットワークへ通信を開始する場合は NAT 越え問題を解決する必要があり，アドレス空間透過性は重要な機能として位置づけられる．

1.4 グループ通信アーキテクチャ GSCIP

FPN の概念を実現するには様々な方式がありうる．以下に提案する GSCIP（Grouping for Secure Communication for IP；ジースキップと呼ぶ） [33] は FPN を実現するための柔軟なグループ通信アーキテクチャであり，それを構成する種々の通信プロトコルは統一性が保たれている．これらのプロトコルには以下に述べる共通した条件がある．

図 1.4 に GSCIP の基本となるセキュア通信グループの定義方法を示す．GSCIP におけるセキュア通信グループの構成要素を GE と呼ぶ．サブネットを構成するルータタイプの GEN（GE for Network），各ノードにインストールされるソフトウェアタイプの GES（GE for Software），重要なサーバの直前に設置して GES と同じ役割を果たすブリッジタイプの GEA（GE for Adapter）がある．GEN の配下に存在する一般ノード（以下 Term と略記する）は，GEN により一括して保護される．GSCIP では同一の暗号鍵を所持する GE の集合を同一セキュア通信グループとして定義する．この暗号鍵をグループ鍵 GK（Group Key）と呼ぶ．同一のセキュア通信グループの GE 間の通信は GK を用いて暗号化される．

GE には同一セキュア通信グループに所属しないノードとの通信を一切禁止する閉域モード CL（Closed Mode）と，異なるセキュア通信グループに所属するノードとは平文での通信が可能な開放モード OP（Open Mode）という 2 つの動作モード OM（Operation Mode）がある．一般に GEN

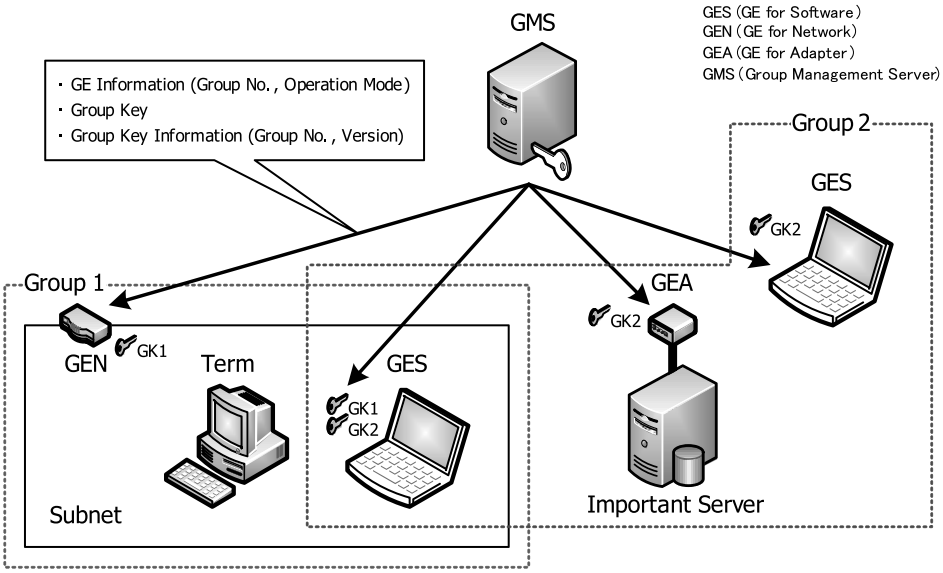

図 1.4　通信グループの定義方法

や重要サーバの直前に設置される GEA は閉域モード，クライアントとして利用される GES は開放モードが定義される．

GE に必要な情報は管理装置 GMS（Group Management Server）で定義される．この情報を GE 情報と呼び，グループ番号と動作モードから構成される．セキュア通信グループは IP アドレスに依存することなく論理的に定義し，個人単位/ドメイン単位が混在したり，1 ユーザに対して重複した複数のセキュア通信グループを定義できる．またサブネット内に存在する個々のノードに対して，そのサブネットとは別のセキュア通信グループを定義することもできる．

GMS ではセキュア通信グループの定義の他に，グループ鍵 GK の生成，更新処理などを行う．グループ鍵 GK は定義されたセキュア通信グループに対応して生成され，定期的に更新される．このときグループ鍵 GK には鍵を識別する情報が付与される．この付与される情報をグループ鍵情報と呼び，グループ番号とバージョン番号から構成される．GE 情報とグループ鍵情報に含まれているグループ番号により，セキュア通信グループとグループ鍵 GK を 1 対 1 に対応づけることができる．

GSCIP において位置透過性を実現するには以下のような機能要素が必要である．すなわち，(1) GMS から GE への定義情報の配送，(2) GE 間の認証と動作処理情報の生成，(3) 動作処理情報に基づく通信パケットの処理である．これらの機能はそれぞれ独立して定義されており，2 章で述べる動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol） [34] は (2) の機能を満たすためのプロトコルであり，3 章で述べる PCCOM（Practical Cipher Communication） [35] は (3) の暗号化機能を満たすシステムである．

(1) **GMS から GE への定義情報の配送**

GE は電源投入時などの初期状態において，GMS から GE ごとに定義されている情報を取得する．この情報には GE 情報，グループ鍵 GK とグループ鍵情報，およびシステム全体で共通に

用いる共通鍵 CK（Common Key）が含まれる．GMS と GE の間は公開鍵を用いた確実な認証と暗号化が実行される．これにより各 GE は必要な情報を予め保持することができる．グループ鍵 GK および共通鍵 CK は GMS から定期的に配送され更新される．

なお，GMS と GE 間の認証および暗号化通信の実現方法として SPAIC（Secure Protocol for Authentication with IC card） [36–38] と呼ぶプロトコルを別途提案している[1]．

(2) **GE 間の認証と動作処理情報の生成**

ノード間の通信開始に先立ち，通信経路上に存在する GE は DPRP により相互に情報交換を行い，通信相手の認証や，通信パケットの処理に必要となる動作処理情報を生成する．GE に定義されたグループ番号や動作モードの組み合わせにより，通信パケットに対する処理内容が決まる．DPRP は通信開始に先立ち実施されるので，ネットワークの物理構成が変化しても GE にはネットワーク構成に応じた動作処理情報が自動生成され，位置透過性が実現される．

(3) **動作処理情報に基づく通信パケットの処理**

TCP/UDP パケットは (2) で決定した動作処理情報に基づいて処理される．処理内容が “Encrypt”，“Decrypt” の場合，グループ鍵 GK で暗号化/復号される．“Transparent” の場合，パケットは透過中継される．“Discard” の場合，パケットは破棄される．

本システムでは管理者が GMS にて GE 情報の変更を行うことにより，セキュア通信グループのメンバ構成を管理する．すなわち，ユーザが自発的にセキュア通信グループへ参加したり，離脱することはできない．企業ネットワークの場合，セキュア通信グループの変更処理は組織変更や人事異動が発生した際に行われ，これらは一般に 4 月 1 日付けなどのように日単位であるため，鍵の定期更新と同期させて実行することが可能である．鍵の更新間隔は管理者が定めることができるが，一般に 24 時間間隔で夜間に実施する等と決めておく．これにより GE は電源投入時に確実に最新の鍵を取得することができる．

また GE が保持する鍵の更新は通信中に行うことも可能である．この場合，全ての GE の鍵更新が完了するまでに，新旧の鍵が混在する時間帯が発生してしまうが，鍵のバージョン番号により誤った鍵で通信しないように考慮されている．通信中に鍵が更新された場合，GE は動作処理情報を初期化する．これにより次の通信開始時には必ず GE 間で 2 章にて詳述する DPRP ネゴシエーションが実行され，グループ鍵情報の交換を行う．古いグループ鍵を持つ GE はネゴシエーションを中止して，GMS に対して新しいグループ鍵を要求する．この方法によれば通信中においてもグループ鍵の更新が可能で，かつ通信中の GE に一時的な遅延が発生するだけで済むため，セキュア通信グループのメンバ数に十分スケールできる．出張等によりユーザの場所が変化した場合は，ユーザの所属自体が変更されるわけではないため，グループ鍵の更新は必要ない．

GMS は通常，イントラネット内に 1 台設置される．ただし通信グループの規模が大きくなる場合は，GMS の処理負荷が増大するため，GMS を分散して設置することが望ましい．この場合は，GMS を DNS のようにツリー構造で管理し，グループ番号を階層化するなどにより，管理情報の一貫性を確保する必要がある．ホームネットワークを含むインターネットにおいて FPN を構築す

[1]SPAIC については付録 D.2 にて概要を示す．

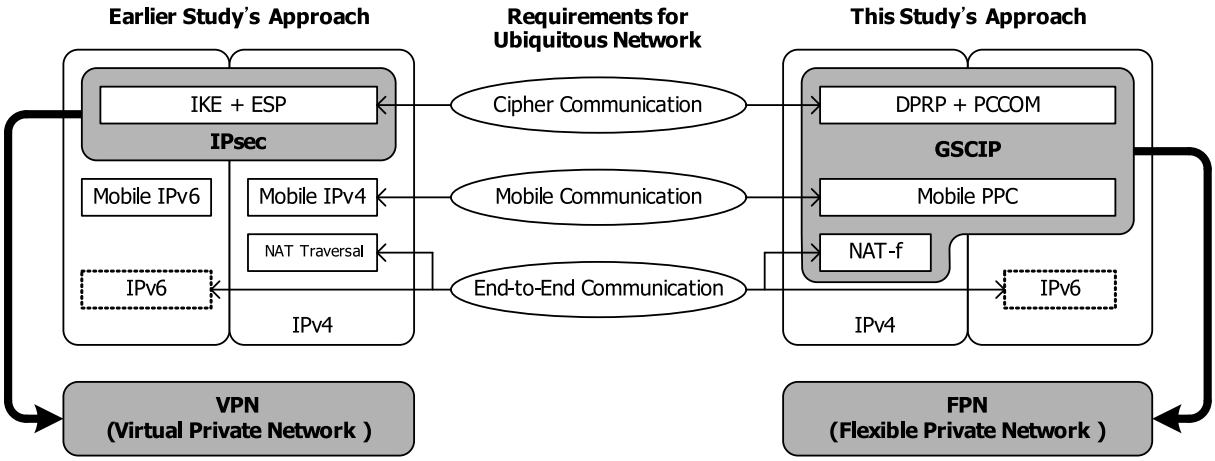

図 1.5　先行研究と本研究におけるアーキテクチャの比較

る場合は，GMS をインターネット上に設置する必要がある．GMS はサービスプロバイダにより管理され，ユーザはセキュア通信グループに参加・離脱または新たに定義したい場合は，GMS の管理者にこれらの依頼を行う方法が考えられる．

GMS のシステムは GE 情報やグループ鍵情報を格納するデータベース，GE に情報を配送するデーモンプロセス，および管理者からの操作を受ける Web アプリケーションから構成される．管理者は管理用端末から SSL により GMS にアクセスし，所定の操作を行うことができる．GMS システムについては文献 [39] にて別途検討を行っており，付録 D.1 を参照されたい．

図 1.5 に先行研究と本研究におけるアーキテクチャの違いを示す．先行研究は IPv6 を基盤とし，暗号化通信の実現には IPsec を，移動通信の実現には Mobile IPv6 を導入するアプローチであることは既に述べた．IPv4 においても実現することはできるが，Mobile IP は IPv4 と IPv6 の互換性がないため別々のシステムを構成する必要がある．また，IPv4 ネットワークにおいてエンドツーエンド通信を実現するために，NAT 越え技術を適用する方法が考えられるが，従来の NAT 越え技術は IPsec や Mobile IP との連携を考慮せずに検討されてきた．IPsec や Mobile IP は NAT との相性が悪く，NAT 越えを行うために別途カプセル化処理などを行う必要がある [40–42]．従って，これらの技術と親和性を保ちながら連携することが可能か，文献 [43, 44] のように別途議論する必要がある．

これに対して，本研究は暗号化通信の実現には DPRP と PCCOM を，移動通信の実現には Mobile PPC（Mobile Peer-to-Peer Communication protocol） [45] を提案する．これらのプロトコルは IPv4/IPv6 のいずれにおいても同様のアーキテクチャを採用しており，両者の互換性を有する設計となっている．エンドツーエンド通信の実現には，IPv4 においては新たに提案する NAT-f（NAT-free protocol）[46] により NAT 越え問題を解決する．なお，IPv6 においては NAT-f を利用せず，IPv6 が提供する双方向接続性によりエンドツーエンド通信を実現する．本研究で提案する GSCIP は統一した概念のもとに上記プロトコルを設計しているため，容易に連携・統合することが可能である．

1.5　本論文の構成

本論文は GSCIP を構成する 3 つのプロトコルと暗号化通信方式に関する提案を基本とし，それらの関係性や統合に関する応用研究から構成される．図 1.6 に本論文の構成と，各章の関係性を示す．

2 章及び 3 章では，暗号化通信に関連して，位置透過性を実現する動的処理解プロトコル DPRP と暗号通信方式 PCCOM について論じる．既存技術である IPsec/IKE アーキテクチャに対して，提案アーキテクチャが必要十分な安全性と高い柔軟性を低い管理負荷で提供できることを示す．また，プロトタイプシステムを構築し，提案アーキテクチャは低遅延・高スループットを実現できることを実証する．

4 章では，移動通信に関連して，移動透過性を実現するプロトコル Mobile PPC について論じる．既存の移動透過性プロトコルの課題を解決し，本論文におけるユビキタスネットワークの要求仕様を満たせることを示す．また，既存の IPv4 ネットワークにおいて移動透過性を実現するための技術的問題点を明らかにし，その解決策について議論する．

5 章では，エンドツーエンド通信に関連して，アドレス空間透過性を実現する外部動的マッピング方式と，それを実現する NAT 越えプロトコル NAT-f について論じる．既存の NAT 越え技術に対して，通信性能を損なうことなく高い汎用性を実現できることを示す．

6 章では 2 つの独立したプロトコル，すなわち Mobile PPC と NAT-f を統合し，従来技術では実現できなかった新たな移動パターンを実現できることを示す．プロトタイプシステムを評価することにより，プロトコルの統合に関わる性能低下がないことを実証する．また，提案アーキテクチャがネットワークモビリティや Mobile IP など様々なシステムに対して応用可能であることを示す．

7 章では，本論文を総括し，本研究の成果と今後の課題を示す．

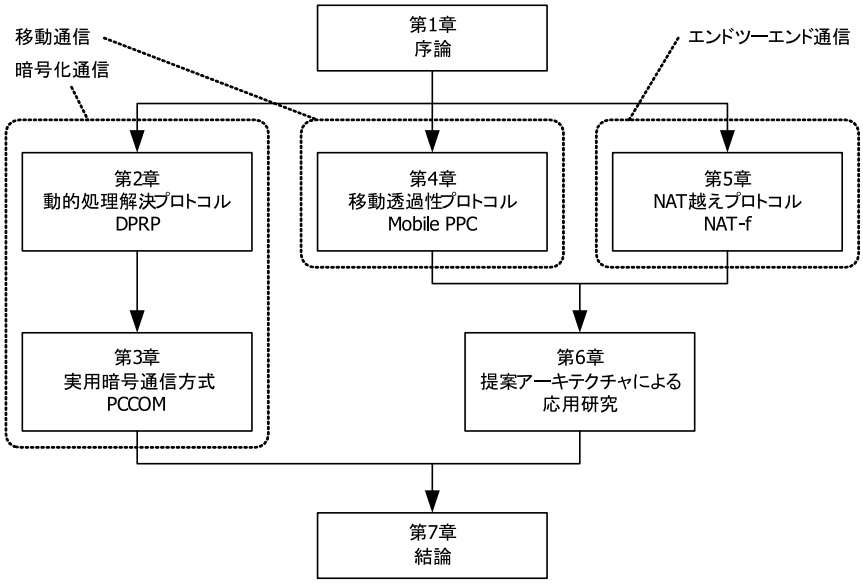

図 1.6　本論文の構成

第2章　動的処理解決プロトコルDPRP

2.1　研究の背景と目的

企業ネットワークでは，不正侵入，データの盗聴や漏洩・改竄などに対する様々なセキュリティ対策が重要な課題となっている．外部からの侵入防止に対しては，通信の暗号化やディジタル署名など，セキュリティ強度の高い技術を駆使したり，ファイアウォールやIDS（Intrusion Detection System）などと併用するなど，様々な工夫がなされている．しかし企業ネットワークのセキュリティの脅威は組織内部にも存在し，社員や内部関係者の不正による犯罪が多く報告されている [47]. 企業ネットワーク内部のセキュリティ対策としては，ユーザ名とパスワードによる簡単な相手認証，アクセス制御程度しか行われていないのが現状であり，有効な対策が今後必要になると考えられる．

このような状況に対応するため，セキュア通信グループの構築は有効な方法である．これはネットワークのインフラ環境をそのまま利用しながら，同一グループのメンバ間の通信の安全を確保する方法であり，以下のように様々な研究が行われている．セキュア通信グループの構築は個人単位に実現する方法 [48–52]，ドメイン単位に実現する方法 [53–56]，および両者を混在させた方法 [30, 57, 58] に分類できる．

個人単位に実現する方法はエンドノードにセキュリティ機能を実装する方法で，代表技術としてIPsec [12] トランスポートモードがある．この方法ではきめ細かいセキュア通信グループの定義が可能であるが，全てのノードに機能を実装する必要があり，規模が大きくなると管理負荷が大きくなる．

ドメイン単位に実現する方法はセキュリティゲートウェイ（以下SGW）間に安全な通信経路を構築することにより，各SGW配下のサブネットをセキュア通信グループの単位として定義する方法で，代表技術としてVPN（Virtual Private Network） [13] で一般的に使用されているIPsecトンネルモードがある．この方法ではSGWだけにセキュリティ機能を実装すればよいが，個人単位の場合のようなきめ細かいセキュア通信グループを定義することが難しい．

両者の利点を共に生かすためには，個人単位のセキュア通信グループとドメイン単位のセキュア通信グループを混在できる方式が望ましい．これは例えば特定のドメインの中に，別のセキュア通信グループに重複帰属する個人が存在するような場合にも対応できる方式である．企業では部門単位の業務グループと部門横断の個人単位の業務グループが混在することがあり，混在型はセキュア通信グループをこのような業務グループと対応づけて定義するのに適している．また特定の個人がセキュリティドメインの内部と外部の間を移動することによりネットワーク構成が変化するような場合に対しても柔軟に対応できることが望まれる．

IPsec はトランスポートモードおよびトンネルモードの互換性が無く，上記のような混在環境への適用には向いていない．IPsec では通信経路上に同一モードの IPsec 機能を持つ装置が対で存在することが前提となっており，混在環境を実現するにはエンドノードにトランスポートモードとトンネルモードの両方を設定しなければならないなど管理負荷が大きくなるという課題がある．

文献 [57, 58] は SOCKS [59] や SSL [3] を拡張して階層的に構築されたセキュリティドメインにも対応可能とした VPN 構築手法である．セキュリティドメインの最も外側の SGW から内側に向かって 1 ホップずつ SGW を認証していくことにより，混在環境に近いシステムを実現している．しかし SGW は次ホップの SGW を特定するために必要な経路情報を管理しなければならず，管理負荷の軽減にはつながっていない．

なお，セキュア通信グループを構築する手法としてマルチキャストグループを通信グループとして構成する方法があるが [60–63]，これらはグループメンバに一括して安全に情報を配送することが目的であり，本論文で扱う業務に対応した双方向の通信とは用途が異なる．

動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol）[34] は GSCIP（Grouping for Secure Communication for IP） [33] の一機能を構成するものであり，フレキシブルプライベートネットワーク FPN（Flexible Private Network）で実現すべき透過性のうち，位置透過性を実現するものである．DPRP はエンドノード間の通信に先立って通信経路上に存在する複数の GSCIP 構成装置 GE（GSCIP Element）が相互に情報交換し，通信パケットの処理に必要となる動作処理情報テーブル PIT（Process Information Table）を各 GE に自動生成する．ネットワークの物理的構成に変化があっても，GE の保持する動作処理情報が DPRP により動的に再生成されるため，管理者の管理負荷を大幅に軽減できる．

渡邊らは文献 [64] において DPRP の原案を提案している．ただし，この時点では FPN や GSCIP の概念が定義されておらず，DPRP の位置づけが不明確であった．また，通信経路上の中間装置では決定された動作処理情報を無条件に登録していたため，動作処理情報テーブルが偽造される恐れがあった．

そこで本章では DPRP を FPN における位置透過性を実現し，GSCIP を構成するプロトコル群の一部として明確に位置づける．これに伴い，通信経路上の GE の情報交換に認証機能を追加し，厳密にシーケンスを定義した．また，このようにして確立した DPRP 仕様を FreeBSD に実装した．GE が送受信する通信パケットを IP 層から抜き出して処理を行い，差し戻すことで既存の処理に影響を与えない方式を実現した．この方式は今後の GSCIP の展開に応用が利く方式であり，シンプルな構造で必要な機能を実現できる．性能評価の結果，DPRP は TCP/UDP 通信にほとんど影響を与えることなく，動作処理情報を生成できることを確認した．また GSCIP/DPRP，IPsec/IKE（Internet Key Exchange） [12, 65] の導入時やネットワーク構成変化時に発生するコストを比較し，DPRP では大幅に管理負荷を軽減できることを示した．

以降，2.2 節で IPsec/IKE について，2.3 節で DPRP の動作概要について述べる．2.4 節で実装方式について述べ，2.5 節で性能評価実験の結果と，管理負荷の評価について述べる．最後に 2.6 節でまとめる．

2.2 既存技術

ネットワークセキュリティの既存技術として IPsec を取り上げる．図 2.1 に IPsec のシステム構成を示す．IPsec は IP 層で動作するカーネルモジュールとユーザランドで動作する IKE から構成され，IPsec 通信に必要なデータとして SPD（Security Policy Database）と SAD（Security Association Database）がある．IPsec を実装したノードはセキュリティポリシに従って IP パケットの処理を行う．セキュリティポリシの処理内容は以下の 3 種類がある．

- “discard”：パケットを破棄
- “bypass IPsec”：IPsec を適用せずに通常の処理を実行
- “apply IPsec”：IPsec を適用

IPsec が適用される場合，図 2.2 に示すカプセル化モードが使用される．トランスポートモードは，IP ペイロード部の暗号化や認証によりセキュリティを確保し，主に ST1 と ST2 のようなエンドノード間で利用される．トンネルモードは，IP パケット全体に対してセキュリティ機能を適用し，主に SGW 間で VPN を構築する場合に利用される．図 2.2 における IPsec 機能を実装しない Term1 と Term2 間の通信は，SGW1 と SGW2 間で暗号化される．

表 2.1 に ST1 と SGW1 のセキュリティポリシが格納された SPD の例を示す．セキュリティポリシではどのトラヒックに対して IPsec を適用するか否か指定し，IPsec を適用する場合はさらに適用する IPsec セキュリティプロトコルやカプセル化モード（トランスポートモードまたはトンネルモード），暗号化アルゴリズムや認証アルゴリズムなどのパラメータを選定する．IPsec セキュリティプロトコルには，ESP（Encapsulating Security Payload）[66] と AH（Authentication Header）[67] があり，目的に応じていずれかを選択して利用する．AH は送信元の認証，データの完全性確保，リプレイアタックの防御などの機能を提供する．ESP は AH の機能に加えて，データの暗号化機能を提供し，トラヒック情報やデータの機密性を確保する．

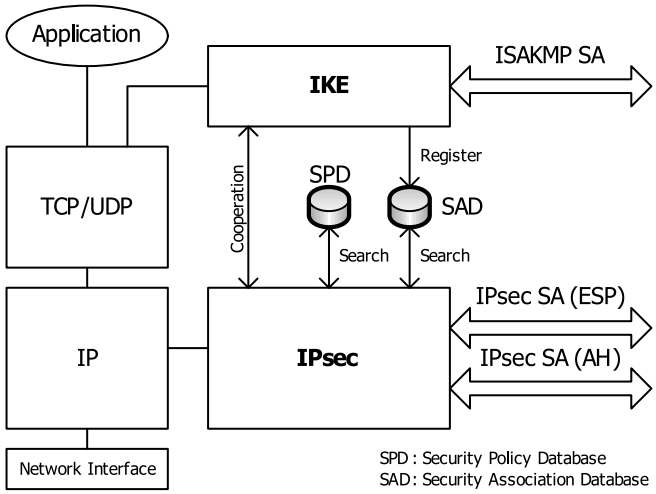

図 2.1　IPsec システム構造

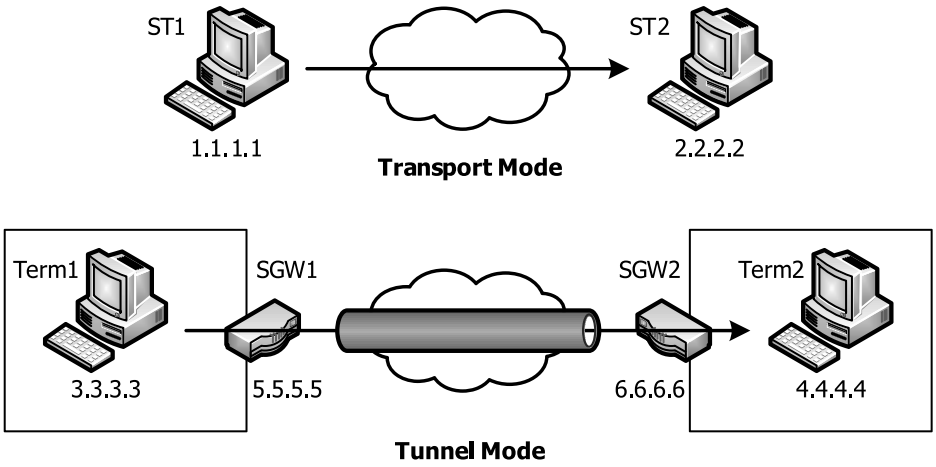

図 2.2　IPsec におけるカプセル化モード

表 2.1　IPsec における SPD の例

ST1

From	To	Protocol	Port	Security Policy
1.1.1.1	2.2.2.2	TCP	1000	Apply IPsec Transport, ESP with 3DES
Any	Any	Any	Any	Bypass IPsec

SGW1

From	To	Protocol	Port	Security Policy
3.3.3.0/24	4.4.4.0/24	Any	Any	Apply IPsec Tunnel, ESP from 5.5.5.5 to 6.6.6.6 with AES
Any	Any	Any	Any	Bypass IPsec

ノードが IP パケットを送信する際，IP 層から IPsec 処理部に処理が移り，IP パケットの送信元/宛先 IP アドレス，上位層プロトコル，ポート番号をキーとして送信用 SPD を検索する．該当するセキュリティポリシの処理内容に従って IP パケットを処理する．IPsec を適用する場合は，次に送信用 SAD から該当するセキュリティアソシエーション（以後 SA）を検索する．該当する SA が存在しない場合は，IKE により新しい IPsec SA を生成する．IKE は IPsec セキュリティプロトコルの SA と鍵の管理を行うためのフレームワークである ISAKMP（Internet Security Association and Key Management Protocol）[68] 上で鍵交換プロトコル Oakley [69] を動作させたプロトコルである．IKE では最初に通信相手 IPsec ノードと ISAKMP SA を確立し，安全な通信路を構築する．その後，ISAKMP SA を使用して表 2.2 に示すような IPsec SA を確立し，SAD に登録する．これにより，再度 IP パケットを送信する際は該当する SA が存在するため，SA パラメータとして保存されている暗号化アルゴリズムや暗号鍵，初期ベクトル IV（Initialization Vector）などを使用して ESP 処理または AH 処理を行う．IPsec 処理が適用されたら，IP パケットを通信相手に送信する．

表 2.2　IPsec における SAD の例

Source	Destination	IPsec Protocol	SPI	SA Parameters
1.1.1.1	2.2.2.2	ESP	1000	Encapsulation Mode: Transport Encryption Algorithm: Blowfish-CBC, Encryption Key: 1048ea648 246a5f7 IV: a7326d1278f12ba2 Authentication Algorithm: HMAC-SHA-1-96 Authentication Key: 826170af1 288e8da Sequence Number: 832

IP パケット受信時の処理は上記と逆の手順となる．すなわち，受信用 SAD から該当する IPsec SA を参照して AH 処理または ESP 処理を行う．その後，受信用 SPD から該当するセキュリティポリシを確認し，適用されていた IPsec 処理の内容が一致しているか確認する．一致していればデータを上位層へ渡し，一致していなければパケットを破棄する．

セキュリティポリシは通信開始時に必須の情報であるため，一般にユーザが初期設定として送信用と受信用の SPD に設定しておかなければならない．また，IKE により ISAKMP SA および IPsec SA を構築する際に必要となる事前共有秘密鍵や，通信相手の IP アドレス，暗号化アルゴリズム，認証方式などを設定しておく必要がある．これらの設定は高度な専門知識が要求されることや，設定項目が多岐にわたり複雑であることから，導入の敷居が高い．また，設定情報に IP アドレスを含んでいるため，ノードの移動に伴い IP アドレスが変化すると設定を変更する必要があり，管理負荷が高いという課題がある．

2.3　提案方式

本節で提案する動的処理解決プロトコル DPRP はノード間の通信開始に先立ち，通信経路上のすべての GE 間で設定されている情報を相互に交換して，通信パケットの処理内容を決定する．各 GE は FPN におけるセキュア通信グループに帰属しており，DPRP により交換したグループ番号を確認することにより通信可否を判断する．ここで，通信開始ノードおよび通信相手ノードに最も近い GE をそれぞれ始点 GE，終点 GE と呼び，両エンド GE 間に存在する GE を中間 GE と呼ぶ．決定した動作処理情報は各 GE に生成される動作処理情報テーブル PIT に保存される．PIT は送信元/宛先 IP アドレスとポート番号，プロトコル番号，処理内容，グループ鍵情報などの情報から構成されている．このうち動作処理情報は，通信パケットの処理内容およびグループ鍵情報のことを示す．

提案システムは IP アドレスに依存しないグループ鍵によるセキュア通信グループの構築方式と，DPRP によるオンデマンドなノード間の情報交換および PIT の生成により，複雑な IPsec アーキテクチャと高い管理負荷の問題を解決する．

図 2.3 にネットワーク構成例と GE 定義情報を示す．図 2.3 は GES1 が GEN により構成された

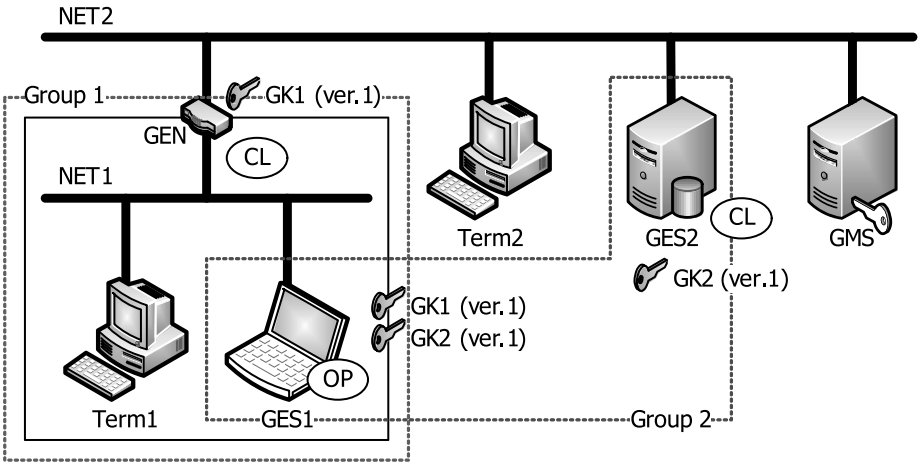

図 2.3　ネットワーク構成図と GE 定義情報

表 2.3　ノード間の通信可否と各 GE が保持する動作処理情報

通信ペア		通信可否	動作処理情報 GES1	GEN	GES2
GES1	GES2	○	E2	T	E2
GES1	Term1	○	T	—	—
GES1	Term2	×	D	D	—
GES2	Term1	×	—	D	D
GES2	Term2	×	—	—	D
Term1	Term2	×	—	D	—

E*x*: Encrypt/Decrypt by GK*x*　T: Transparent
D: Discard　—: No Record

部門サブネットワーク NET1（Group1）の内部に存在し，かつ GES2 へのアクセスが許可されているセキュア通信グループ（Group2）に所属している状況を想定している．GES1 は NET1 の外部 NET2 へ移動した場合，部門内の一般ノード Term1 との通信が可能なように GK1 も予め保持している．GES2 は他のグループからの通信を拒否するために閉域モード，GES1 は同一部門の一般ノードとも通信するため開放モード，GEN は部門内の一般ノードを保護するために閉域モードがそれぞれ定義されている．各 GE が所属するグループ番号とそれに対応するグループ鍵 GK は，既に GMS から配送されているものとする．

ここで，図 2.3 の状態においてノード間に生成されるべき動作処理情報を表 2.3 に示す．GES1 と GES2 間の通信に着目すると，始点 GE となる GES1 と終点 GE となる GES2 は通信パケットを GK2 で暗号化/復号し，中間 GE である GEN は通信パケットを透過中継[1]する．DPRP はこのような動作処理情報を自動的に生成する役割を持つ．

[1]該当パケットに対して暗号化処理を行わないことを意味し，“Transparent” と表記する．IPsec における “bypass IPsec” に該当する．

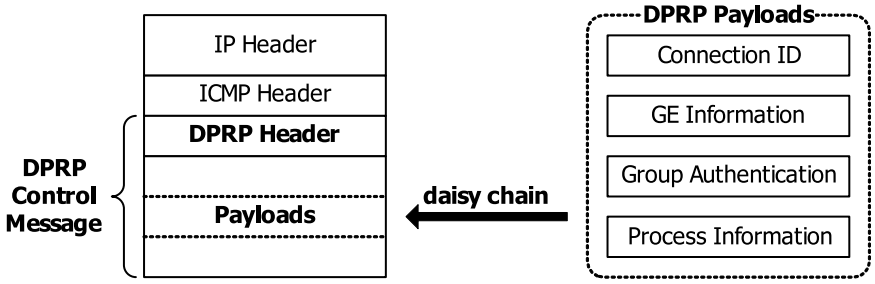

図 2.4　DPRP 制御メッセージフォーマット

2.3.1　プロトコル定義

ここでは DPRP プロトコルの仕組みを述べる．なお，本論文で用いる記号については付録 A で定義したので，適宜参照されたい．DPRP は ICMP Echo をベースとした制御メッセージとして定義されており，始点 GE と終点 GE 間でネゴシエーションを行われる．図 2.4 に DPRP 制御メッセージを示す．ICMP ヘッダの下に DPRP ヘッダを定義し，以下に示す 4 種類の制御メッセージを識別する．

1. **DDE（Detect Destination End-GE）**
   終点 GE を決定するための制御メッセージ．DPRP ネゴシエーションを開始した GE から実際の通信相手ノードに向けて送信される．

2. **RGI（Report GE Information）**
   始点 GE を決定するため，また通信経路上の GE に定義されている情報を始点 GE に通知するための制御メッセージ．終点 GE から実際の通信開始ノードに向けて送信される．

3. **MPIT（Make Process Information Table）**
   通信経路上の各 GE に決定した動作処理情報を通知するための制御メッセージ．始点 GE から終点 GE に向けて送信される．

4. **CDN（Complete DPRP Negotiation）**
   DPRP ネゴシエーションの完了を通知するための制御メッセージ．ネゴシエーション処理が正常に完了した場合は，終点 GE から始点 GE に向けて送信される．正常に完了しなかった場合は，エラーが発生した GE から始点 GE に向けて送信される．

DPRP ヘッダには DPRP 制御メッセージであることを示す識別子（$DPRP_{ID}$），ネゴシエーション識別子（NID）や CDN により通知される確認情報（STS）などが記載される．以後，DPRP ヘッダは以下のように記号で定義する．

$$HDR = DPRP_{ID}, NID, STS \tag{2.1}$$

制御メッセージで交換されるデータは以下の 4 種類のペイロードに記載され，各ペイロードは数珠つなぎの構造を採用している．

通信識別子ペイロード

TCP/UDP パケットの送信元/宛先 IP アドレス（IP_{Src}，IP_{Dst}），送信元/宛先ポート番号（$port_{Src}$，$port_{Dst}$），およびプロトコル番号（$proto$）の 5 つの情報から構成される．この情報を通信識別子 CID（Connection ID）と呼び，以下のように表記する．

$$CID = (IP_{Src}, IP_{Dst}, port_{Src}, port_{Dst}, proto) \tag{2.2}$$

通信識別子ペイロードは DDE と RGI に記載される．

GE 情報ペイロード

GE にログイン中のユーザ ID（UID），GE の動作モード（OM），方向情報（Dir），動作処理情報の認証に用いる乱数値（aID），および GE が保持するグループ鍵の情報（GKI）から構成される．ある GE の GE 情報 N_{GE} を以下のように表記する．

$$N_{GE} = (UID_{GE}, OM_{GE}, Dir_{GE}, aID_{GE}, \{GKI\}_{GE}) \tag{2.3}$$

ここで，方向情報とは DDE が GEN の配下から出る方向（outbound）なのか，配下へ入る方向（inbound）なのかを示す情報である．また，$\{GKI\}_{GE}$ は GE が保持する全てのグループ鍵情報を意味する．GE 情報ペイロードは RGI に記載される．

グループ認証ペイロード

決定したグループ鍵で暗号化されたネゴシエーション識別子（NID）から構成される．グループ認証ペイロードは MPIT に記載される．

動作処理情報ペイロード

該当する GE のユーザ ID（UID），認証情報（aID），決定した処理内容（$Proc$），および決定したグループ鍵の情報（$DGKI$）から構成される．ある GE に関する動作処理情報 P_{GE} を以下のように表記する．

$$P_{GE} = (UID_{GE}, aID_{GE}, Proc_{GE}, DGKI) \tag{2.4}$$

動作処理情報ペイロードは MPIT に記載される．

2.3.2 動作概要

図 2.5 に GES1 が GES2 に通信を開始する際のシーケンスを示す．提案方式における通信は以下に示すステップに従って行われる．

Step 1: GES1 が GES2 と通信を開始する際，アプリケーションがソケットを通じてデータを送信する処理を行う．これにより，TCP/UDP パケット

$$IP_{GES1} : port_{GES1} \rightarrow IP_{GES2} : port_{GES2} \quad [proto] \tag{2.5}$$

が IP 層へ渡される．

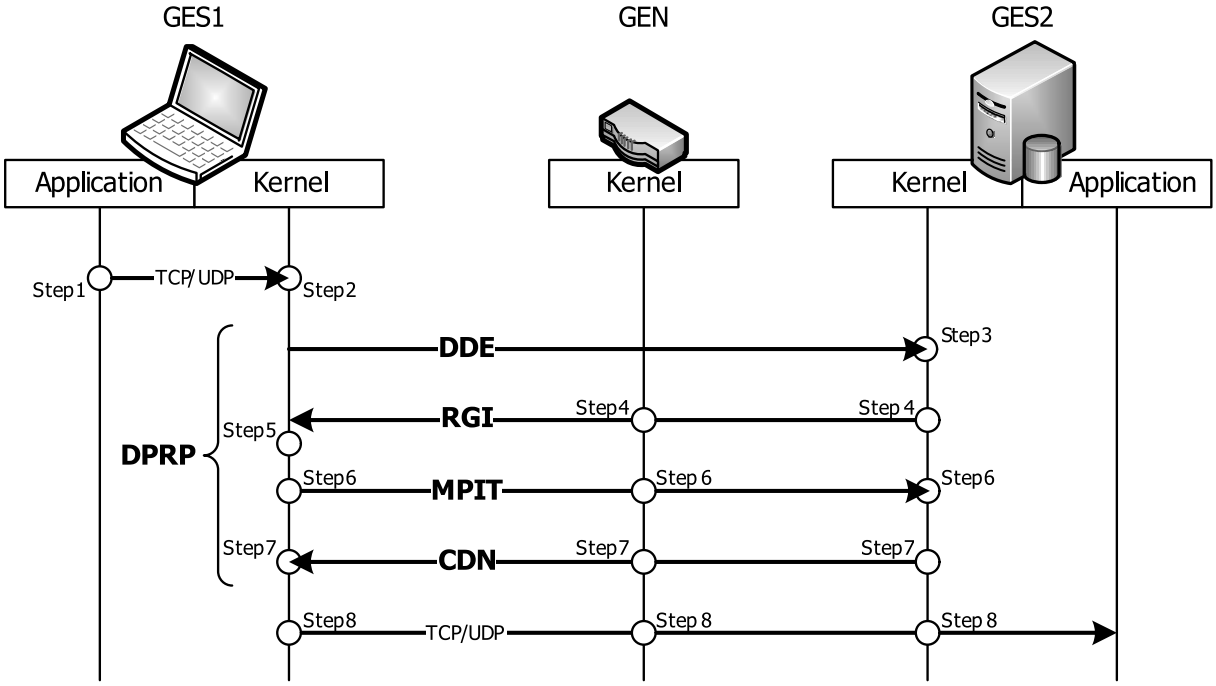

図 2.5　DPRP ネゴシエーションと処理内容

Step 2: IP 層において TCP/UDP パケットの通信識別子をキーとして PIT を検索する．送信するパケットに該当する動作処理情報がある場合は，Step 8 のパケット処理へ移る．初めて通信を行う場合は該当する動作処理情報が存在しないため，DPRP ネゴシエーションを開始する．GES1 は乱数値を生成し，これを *NID* として DPRP ヘッダに設定して DDE を生成する．

DDE:
GES1 → GES2:　$HDR, E_{CK}(CID)$

DDE に記載される通信識別子ペイロードには式 (2.5) の情報が設定され，共通鍵 CK で暗号化される．GES1 は上記 DDE をトリガパケットの宛先ノードである GES2 へ送信し，処理中だった TCP/UDP パケットを一時的に待避しておく[2]．

Step 3: DDE を受信した GES2 は終点 GE となるため，RGI を生成する．RGI における DPRP ヘッダと通信識別子ペイロードは，DDE のものをそのまま利用する．

RGI:
GES2 → GES1:　$HDR, E_{CK}(CID), E_{CK}(N_{GES2})$　(between GES2 and GEN)
$HDR, E_{CK}(CID), E_{CK}(N_{GES2}), E_{CK}(N_{GEN})$
(between GEN and GES1)

[2]待避されたパケット，すなわち DPRP ネゴシエーションを開始するきっかけとなったパケットを “トリガパケット” と呼ぶ．

Step 4: GES2 は，RGI に自身の GE 情報 N_{GES2} を共通鍵 CK で暗号化して，通信識別子ペイロードにつなげる．ここで RGI に追記した N_{GES2} に含まれる aID_{GES2} や DPRP ヘッダに記載されている NID を一時的に記録しておくために，PIT を作成する．これらの情報は，RGI 以降の DPRP 制御メッセージを認証するために利用される．GES2 は RGI をトリガパケットの送信元ノードである GES1 に送信する．

中間 GE の GEN は RGI を受信したら，暗号化した自身の GE 情報 $E_{CK}(N_{GEN})$ を RGI に追加して転送する．

Step 5: RGI を受信した GES1 は始点 GE となり，受信した通信経路上の全 GE の GE 情報と自身の GE 情報 N_{GES1} から動作処理情報を決定する．動作処理情報の決定方法は 2.3.3 項にて後述する．

動作処理情報を決定後，GES1 は自らの動作処理情報 P_{GES1} を PIT に仮登録する．その後，決定したグループ鍵（GES1 と GES2 間の場合，GK2）で暗号化した NID をグループ認証ペイロードに設定し，MPIT に記載する．さらに，残りの動作処理情報を共通鍵 CK で暗号化して MPIT に追加したら，終点 GE である GES2 へ送信する．

MPIT:
GES1 → GES2: $HDR, E_{GK2}(NID), E_{CK}(P_{GEN}), E_{CK}(P_{GES2})$

Step 6: MPIT を受信した各 GE は，該当する動作処理情報 P_{GE} を復号して取得する．取得した NID，UID_{GE}，aID_{GE} と PIT に登録しておいた情報を比較して認証を行う．ここで，処理内容が “Encrypt” または “Decrypt” の場合，さらに決定したグループ鍵 DGK によりグループ認証ペイロードの復号し，NID を比較して認証する．認証処理が正常であれば，動作処理情報を PIT に仮登録する．

終点 GE の GES2 は PIT の登録を完了すると，DPRP ネゴシエーションが完了したことを通知する CDN を生成する．

CDN:
GES2 → GES1: HDR

MPIT による PIT の仮登録が正常に完了した場合，DPRP ヘッダの確認情報 STS には OK が設定される．MPIT 受信時における認証処理に失敗した場合は NG が設定され，速やかに CDN を始点 GE の GES1 へ送信して Step 7 の処理を行う．

Step 7: CDN は始点 GE の GES1 に向けて送信される．CDN を受信した各 GE は，確認情報 STS が OK なら仮登録していた PIT を有効な状態に確定する．STS が NG なら，仮登録していた PIT を破棄する．始点 GE の GES1 が PIT を確定すると DPRP ネゴシエーションを完了し，待避していたトリガパケットを復帰させて TCP/UDP 通信を開始する．

表 2.4　GES1-GES2 間に生成される PIT の一例

GES1

IP_{Src}	IP_{Dst}	$port_{Src}$	$port_{Dst}$	$proto$	Process Information
1.1.1.1	2.2.2.2	49230	21	TCP	Encrypt with GK2 (ver.1)
2.2.2.1	1.1.1.1	21	49230	TCP	Decrypt with GK2 (ver.1)

GEN

IP_{Src}	IP_{Dst}	$port_{Src}$	$port_{Dst}$	$proto$	Process Information
1.1.1.1	2.2.2.2	49230	21	TCP	Transparent
2.2.2.1	1.1.1.1	21	49230	TCP	Transparent

GES2

IP_{Src}	IP_{Dst}	$port_{Src}$	$port_{Dst}$	$proto$	Process Information
1.1.1.1	2.2.2.2	49230	21	TCP	Decrypt with GK2 (ver.1)
2.2.2.1	1.1.1.1	21	49230	TCP	Encrypt with GK2 (ver.1)

Step 8: 以後，各 GE は PIT の動作処理情報に従って，TCP/UDP パケットに対して暗号化/復号，透過中継，破棄のいずれかの処理を実行する．表 2.4 に GES1-GES2 間に生成される PIT を示す．これは IP_{GES1}，IP_{GES2} をそれぞれ 1.1.1.1，2.2.2.2 として，GES1 が GES2 へ FTP 接続した場合（$port_{GES1}$：49230，$port_{GES2}$：21 とする）に生成される PIT の一例である．この場合，GES1 から GES2 への送信パケットは GES1 においてグループ鍵 GK2 により暗号化される．GEN では暗号化されたパケットを透過中継し，GES2 においてグループ鍵 GK2 により復号されて，上位アプリケーションへデータが渡される．

以上の処理により，通信経路上に暗号化通信に必要な動作処理情報を動的に生成することができる．なお，ノードが移動して IP アドレスが変化した場合，該当する PIT が存在しないため，新たに DPRP により対応した動作処理情報が再生成される．移動前に生成された古い動作処理情報は参照されることがないため，一定時間が経過すると自動的に削除される．

2.3.3　動作処理情報の決定プロセス

図 2.6 に動作処理情報の決定処理フローを示す．RGI により収集された GE 情報 N_{GE} は，RGI 転送中に設定されたネゴシエーションの方向情報により，始点 GE 側と終点 GE 側の情報に分割される．方向情報は図 2.7 に示すように，DDE が GEN が構成するネットワークから出る方向（outbound）か，ネットワークに入る方向（inbound）なのかを表す．実際の方向情報は，RGI が各 GE を通過する際に追加される GE 情報 N_{GE} に記載される．すなわち，RGI が配下ネットワークから出る場合は，DDE が配下ネットワークに入ることと等価であり，RGI の方向に基づいて DDE の方向情報を判断する．

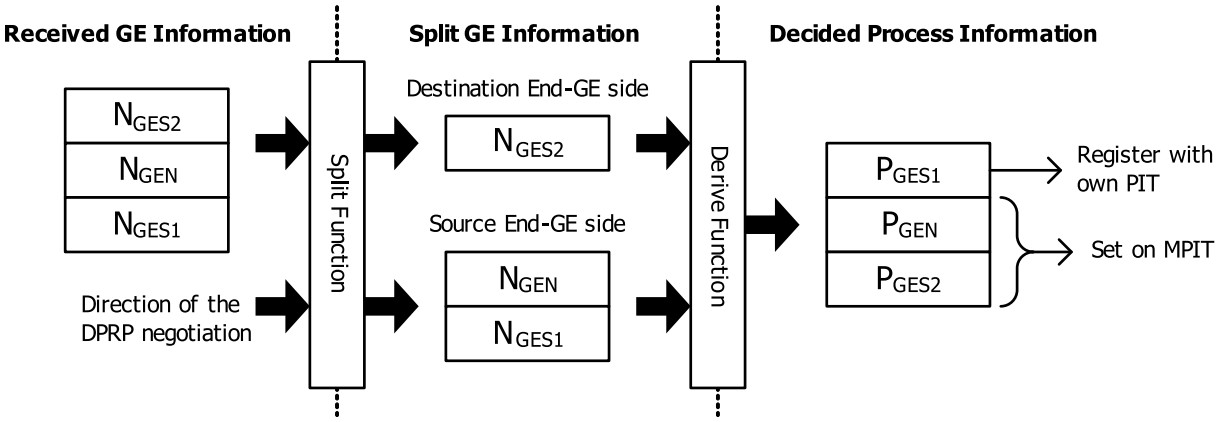

図 2.6　動作処理情報の決定処理フロー

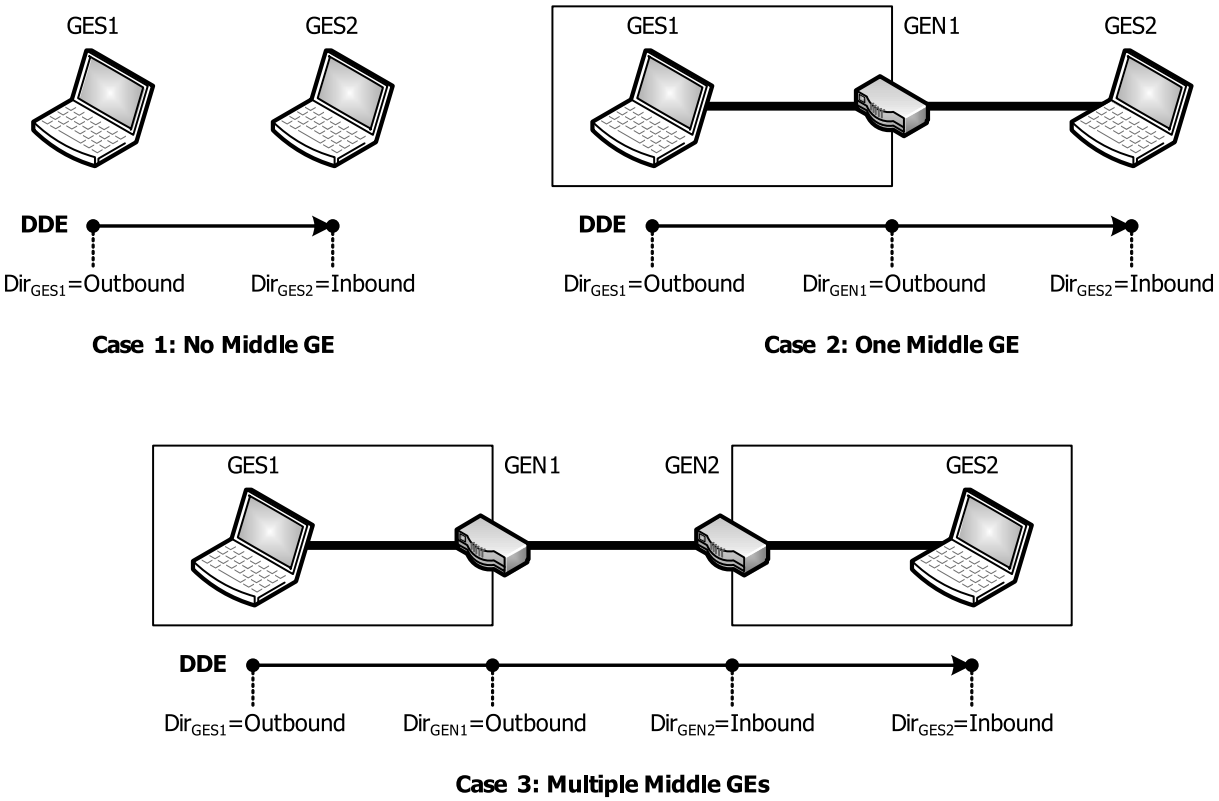

図 2.7　ネゴシエーションの方向情報

Case 1 のように同一ネットワーク内の GES 同士，あるいは異なるネットワークに存在する GE 間に中間 GE が存在しない場合は，DDE 送信側 GE には “outbound”，DDE 受信側 GE には “inbound” が設定される．

Case 2 は 1 台の中間 GE が存在する場合である．GEN1 では DDE が配下ネットワークから出る方向のため “outbound” が設定される．エンド GE は Case 1 と同様の仕組みで方向情報が設定される．

Case 3 は異なる GEN の配下に位置する GE 同士が通信する場合である．考え方は Case 2 と同じで，GEN1 では “outbound” が，GEN2 では “inbound” が設定される．この例では中間 GE が 2 台であるが，さらに多くの中間 GE が存在しても同様である．

始点 GE は収集した方向情報の境界部分，すなわち “inbound’’ と “outbound” の境界で GE 情報を分割する．“outbound’’ 側と “inbound” 側に分割された情報を，それぞれ始点 GE 側の情報と終点 GE 側の情報として区別する．GE 情報の分割後，動作処理情報の決定プロセスを実行する．なお，通信経路上に GE が 1 台のみの場合は分割処理を行わず，ただちに動作処理情報の決定プロセスを行う．このような状況は，通信開始側ノードまたは通信相手ノードのどちらかが一般ノード，かつ中間 GE が存在しない場合，または通信ペアが共に一般ノードで通信経路上に 1 台の GEN が存在する場合に発生する．

ここからは，動作処理情報の決定プロセスについて詳述する．始点 GE は取得した GE 情報 N_{GE} の数，および N_{GE} に含まれるグループ鍵情報 $\{GKI\}_{GE}$ と動作モード OM_{GE} の組み合わせにより，処理内容 $Proc_{GE}$ を決定する．

GE 情報が 1 つの場合

通信経路上に 1 台の GE だけ存在する場合は，自身の N_{GE} に含まれる OM_{GE} を確認する．OM_{GE} が開放モード（OP）の場合，$Proc_{GE}$ には透過中継を示す “Transparent” が設定される．OM_{GE} が閉域モード（CL）の場合は，破棄を示す “Discard” が設定される．

$$Proc_{GE} = \begin{cases} “Transparent” & if\ \ OM_{GE} = OP \\ “Discard” & if\ \ OM_{GE} = CL \end{cases} \tag{2.6}$$

GE 情報が 2 つ以上の場合

始点 GE 側の情報と終点 GE 側の $\{GKI\}_{GE}$ と OM_{GE} を比較して $Proc_{GE}$ を決定する．図 2.8 に GE 情報の比較順序を示す．収集した GE 情報が n 個とすると，GE 情報は始点 GE 側から順に $N_1, \cdots, N_n$ とすることができる．さらに，分割された始点 GE 側の情報を $N_1, \cdots, N_e$，終点 GE 側の情報を $N_{e+1}, \cdots, N_n$ とする．

まず，N_1 を基準として最も離れた終点 GE 側の GE 情報 N_n の両者を比較する．ここで，$\{GKI\}_1$ と $\{GKI\}_n$ の中に一致するグループ鍵情報が存在したら，エンド GE 間で暗号化し，中間 GE は透

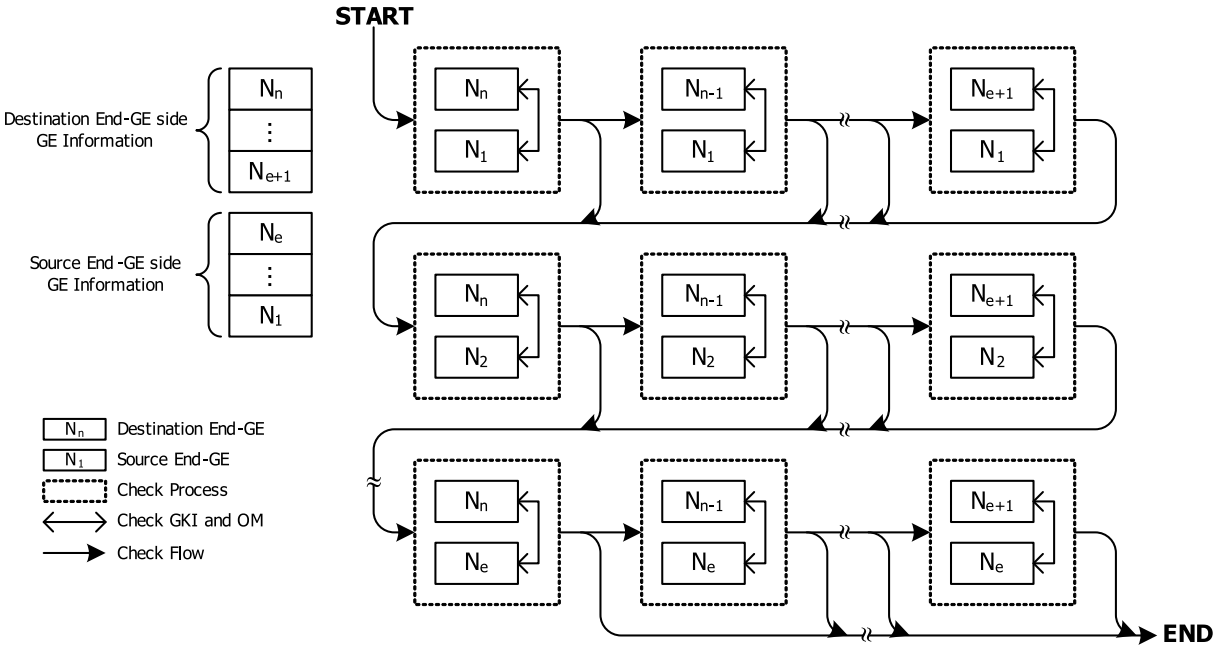

図 2.8　GE 情報の比較順序

過中継する処理内容を決定する．一致するグループ鍵情報が存在しない場合は，OM_1 と OM_n を確認する．OM_1 と OM_n が共に CL だった場合，通信できないため通信パケットを破棄する処理内容を決定する．OM_n が OP だった場合，始点 GE から見て 1 つ手前の GE の情報，すなわち N_{n-1} と再度グループ鍵情報の比較を行う．OM_1 が OP，OM_n が CL だった場合は，始点 GE から 1 つ先に進んだ情報，すなわち N_2 と再度グループ鍵情報の比較を行う．

比較を順に繰り返し，$N_p \in \{N_1, \cdots, N_e\}$ と $N_q \in \{N_{e+1}, \cdots, N_n\}$ の比較により暗号化する処理内容が決定した場合，各 GE の処理内容 $Proc_1, \cdots, Proc_n$ は以下のようになる．

$$Proc_i = \begin{cases} \text{“}Encrypt\text{”} & if \ \ i = p \\ \text{“}Transparent\text{”} & if \ \ i \neq p, q \quad (i = 1, \cdots, n) \\ \text{“}Decrypt\text{”} & if \ \ i = q \end{cases} \tag{2.7}$$

上記のように比較対象を順にシフトしていき，N_e と N_{e+1} の比較においても一致するグループ鍵が無い場合は，動作モードにより処理内容が決定する．

$$Proc_i = \begin{cases} \text{“}Transparent\text{”} & if \ \ \forall OM_{GE} \in \{OM_1, \cdots, OM_n\}, OM_{GE} = OP \\ \text{“}Discard\text{”} & if \ \ \exists OM_{GE} \in \{OM_1, \cdots, OM_n\}, OM_{GE} = CL \end{cases} \quad (i = 1, \cdots, n) \tag{2.8}$$

2.3.4 安全性

DPRP ヘッダに記載されるネゴシエーション識別子（*NID*）はセッションごとに異なる乱数値で，PIT の生成過程から削除されるまで PIT に登録される．DPRP 制御メッセージを受信した際，ICMP ヘッダに記載されているシーケンス番号（$ICMP_{Seq}$）と *NID* をチェックし，リプレイ攻撃に対処する．

DPRP 制御メッセージを CK で暗号化する際の暗号アルゴリズムは AES（Advanced Encryption Standard）[70] の CBC（Cipher Block Chaining）モードを採用した．AES は 2001 年に NIST（National Institute of Standards and Technology）により次世代標準暗号として制定されたブロック暗号のアルゴリズムであり，高い暗号強度と優れた処理速度を有している．図 2.9 に CBC モードにおける暗号化の仕組みを示す．平文 M を n 個のブロック $m_1,\cdots,m_n$ に区切ると，各ブロックの平文 m_i と前のブロックの暗号文ブロック c_{i-1} との XOR をとり，暗号鍵 K により暗号化したデータを暗号文ブロック c_i とする．出力された暗号文ブロック $c_1,\cdots,c_n$ を結合することにより，暗号文 C を得る．

$$C = \{c_1, c_2, \cdots, c_n\} \qquad c_i = \begin{cases} E_K(IV \oplus m_i) & (i = 1) \\ E_K(c_{i-1} \oplus m_i) & (i \neq 1) \end{cases} \tag{2.9}$$

CBC モードでは，最初のブロックの XOR にブロックサイズと同じサイズの初期ベクトル IV が必要となる．AES におけるブロックサイズは 128 bit の固定長となる．

そこで，IV には ICMP のシーケンス番号やネゴシエーション識別子等の情報を含むデータを MD5 [71] により算出したハッシュ値を用いる．表 2.5 に DPRP 制御メッセージにおける各種デー

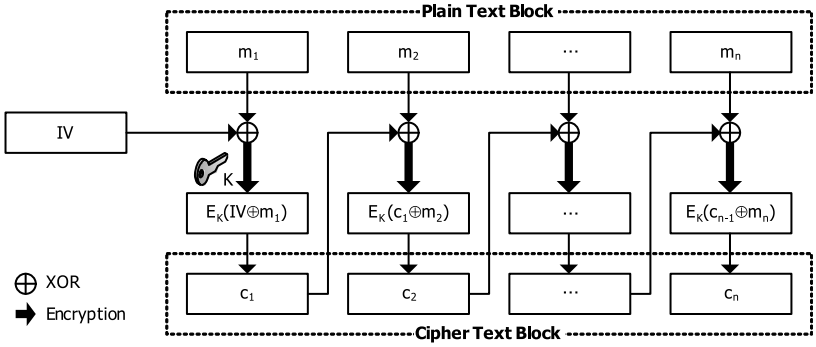

図 2.9　CBC モードにおける暗号化

表 2.5　DPRP 制御メッセージの暗号化に必要な初期ベクトルの生成法

初期ベクトル	暗号化対象	生成法
$IV1$	$E_{CK}(CID)$	$h(CKI \parallel NID \parallel ICMP_{ID} \parallel ICMP_{Seq})$
$IV2$	$E_{CK}(N_{GE})$	$h(CKI \parallel NID \parallel ICMP_{ID} \parallel ICMP_{Seq} \parallel IP_{Dst})$
$IV3$	$E_{DGK}(NID)$	$h(DGKI \parallel NID \parallel ICMP_{ID} \parallel ICMP_{Seq})$
$IV4_{GE}$	$E_{CK}(P_{GE})$	$h(CKI \parallel NID \parallel ICMP_{ID} \parallel ICMP_{Seq} \parallel N_{GE})$

タの暗号化に必要な IV の生成法を示す．各 DPRP 制御メッセージを受信した GE は，受信したメッセージの情報と共通鍵 CK の鍵情報を用いて IV を算出してから復号処理を行う．従って，第三者により DPRP 制御メッセージが改竄されていても復号時に異常を検出することが可能である．

また MPIT 受信時は，IV を求めるために PIT に一時的に登録していた *aID* や *NID* の情報，即ち RGI で通知した情報 N_{GE} を用いる．従って第三者が不正な MPIT により，意図的に GE 間の動作処理情報を生成したり，セッションをハイジャックすることは極めて困難である．更に DPRP ネゴシエーションの過程において不正が検出された場合，CDN により NG を報告し，仮登録中であった PIT をクリアすることができる．以上の処理により，PIT は安全に GE 内に生成することができる．

2.4 実装

DPRP は IP 層に実装される．GSCIP を実現するモジュール群のことを GPACK（GSCIP Package）と呼び，DPRP はその一部を構成する．OS には IP 層の情報が豊富な FreeBSD を選択した．

図 2.10 に GPACK の実装概要を示す．GPACK は IP 層の入出力関数 `ip_input()`, `ip_output()` から呼び出され，DPRP 対応の処理などを行い，パケットを元の場所に差し戻す．この方式では既存の IP 層の処理は GPACK の影響を一切受けることがない．DPRP ではトリガとなった TCP/UDP パケットを一時待避するが，待避パケットをそのままカーネルに残しておき，一連の DPRP 処理が終了した時点でカーネル内から直接送信する．

DPRP により生成される PIT や，GMS から配送された共通鍵 CK およびグループ鍵 GK の保存領域はカーネルメモリ空間に作成し，不要になったら削除する．これらの処理は全てカーネル処理で閉じており，暗号鍵が処理過程で漏洩する可能性は極めて低い．

PIT はハッシュテーブルとして実装する．ハッシュの検索キーは通信識別子，すなわち送受信パケットの送信元/宛先 IP アドレスとポート番号，プロトコル番号である．PIT レコードにはカウンタ値が定義されており，カーネルタイマ処理により減少していく．PIT レコードが参照される度，カウンタ値は初期値に戻される．一定時間参照されていない PIT レコードはカウンタ値が 0 になりノード間の通信が行われていないと判断されて削除される．削除までの時間は ARP（Address Resolution Protocol） [72] キャッシュと同等の 5 分とした．

GPACK は IP 入出力関数より受け取った通信パケットの種類を判別してから，適切なモジュールを選択し実行する．図 2.11 に GPACK における TCP/UDP パケット処理を示す．送受信パケットが TCP/UDP の場合，PIT の検索を行う．該当する PIT レコードが存在した場合，PIT の内容に従ってパケットの処理を実行する．該当する PIT レコードが存在しない場合，DPRP モジュールに処理が渡され，DPRP モジュールは DDE を作成して `ip_output()` に渡し送信する．その後，ネゴシエーションのトリガとなった TCP/UDP パケットを待避する．

送受信パケットが ICMP の場合，通常の ICMP パケットか DPRP 制御メッセージかをチェックし，通常の ICMP パケットであれば GPACK で処理を行わずに IP 層へ戻す．DPRP 制御メッセージの場合，DPRP モジュールに渡され，PIT の生成，動作処理情報の決定，認証，DPRP 制御メッ

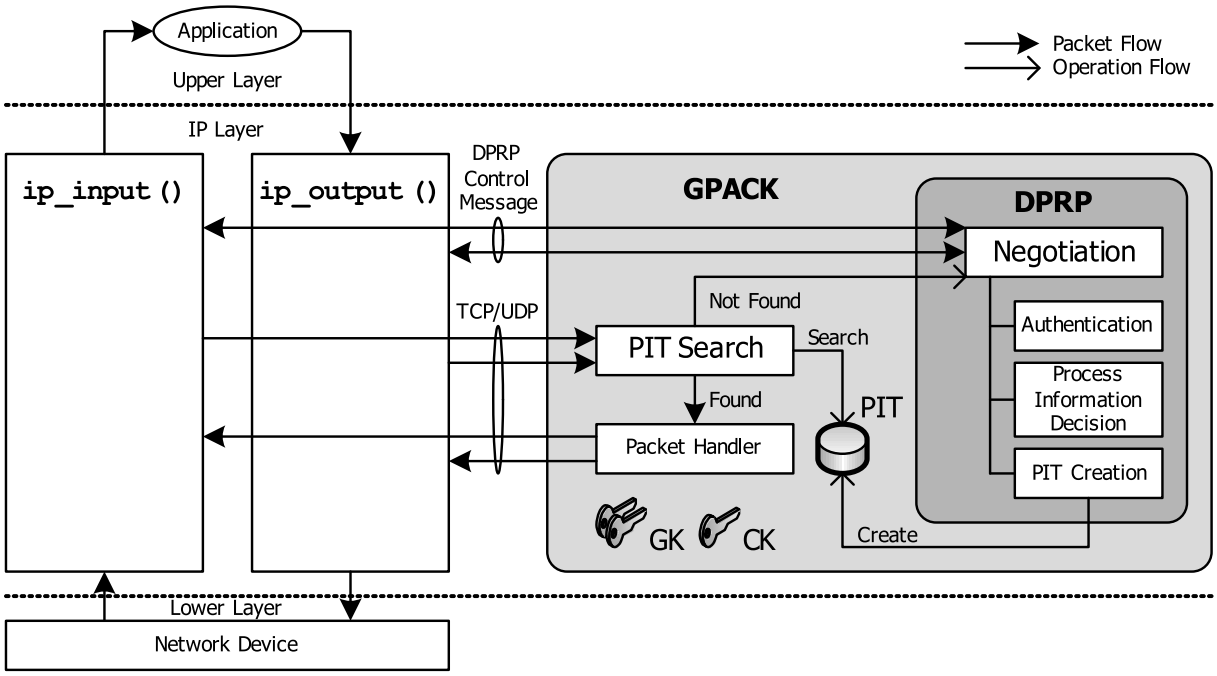

図 2.10　DPRP モジュールの実装

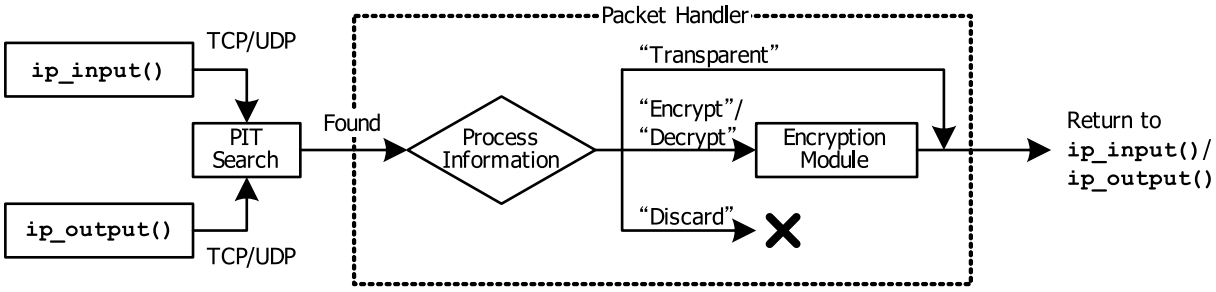

図 2.11　GPACK における TCP/UDP パケット処理

セージの生成などのプロセスを実行する．

DPRP 制御メッセージは生成後に共通鍵 CK により暗号化される．共通鍵 CK およびグループ鍵 GK の鍵長は 128 bit とし，暗号ライブラリには FreeBSD 5.3-RELEASE に実装されている OpenSSL（Version 0.9.7d） [73] を用いた．

2.5　評価

2.5.1　DPRP の性能

100BASE-TX の Ethernet において，GES1 が GES2 に FTP 接続を行う場合の DPRP の性能を測定した．性能測定に使用した各装置仕様は CPU が Pentium4 2.4 GHz，メモリが 512 MByte である．DPRP ネゴシエーションのオーバヘッド時間および DPRP モジュールの内部処理時間を測定

した．また，GSCIP では TCP/UDP パケットを送受信する際，必ず PIT 検索を行うため通信性能に影響が出る可能性がある．そのため PIT 検索のオーバヘッドを調査するため，GSCIP 実装時と未実装時の FTP スループットを暗号化しない状態で比較した．なお，各 GE は予めグループ番号，共通鍵 CK およびグループ鍵 GK を保持しているものとした．

オーバヘッドとモジュールの処理時間

オーバヘッドの測定には，ネットワークアナライザ Ethereal [74] を用いた．参考のために，同一条件下における IPsec/IKEv1 の処理時間も測定した．FreeBSD に実装されている KAME プロトコルスタック [75] および IKE デーモン `racoon` [76] を使用し，事前共有鍵方式のメインモードで行った．IKEv2 [77] は現時点では安定して動作するソフトウェアが存在しないため今回は測定を見送った．測定対象は DPRP では図 2.12 (a) に示す `[I]` DPRP ネゴシエーション時間（DDE～CDN 間）と，`[II]` TCP の最初の SYN パケットが GES1 から送信されるまでの時間（通信開始までの時間）である．一方，IKE では図 2.12 (b) に示す `[I]` IKE ネゴシエーション時間（$\mathrm{ISAKMP}_{\mathrm{FIRST}}$～$\mathrm{ISAKMP}_{\mathrm{LAST}}$ 間）と，`[II]` 通信開始までの時間である．図 2.12 (b) における ST1，ST2，SGW はそれぞれ GES1，GES2，GEN の位置に該当し，IPsec 機能を実装した装置である．

それぞれのオーバヘッド測定結果を表 2.6 に示す．DPRP のネゴシエーション時間は 1.01 msec，通信開始までの時間は 1.04 msec となった．それに対し，IKE のネゴシエーション時間は 1105.95 msec（約 1 sec），通信開始までの時間は 2994.03 msec（約 3 sec）となった．

内部処理時間の測定には RDTSC（Read Time-Stamp Counter）[78] を用いた．測定箇所は図 2.12 (a) に示す○印の部分である．GPACK モジュールの処理時間と DPRP 制御メッセージの暗号処理時間を表 2.7 に示す．GES1-GES2 間のネゴシエーションにおいて，GES1，GEN，GES2 の内部処理時間はそれぞれ 96.59 μsec，44.32 μsec，62.43 μsec となった．またこのうち，約 30 % が DPRP 制御メッセージの暗号化/復号，ならびに認証処理に要する時間であった．

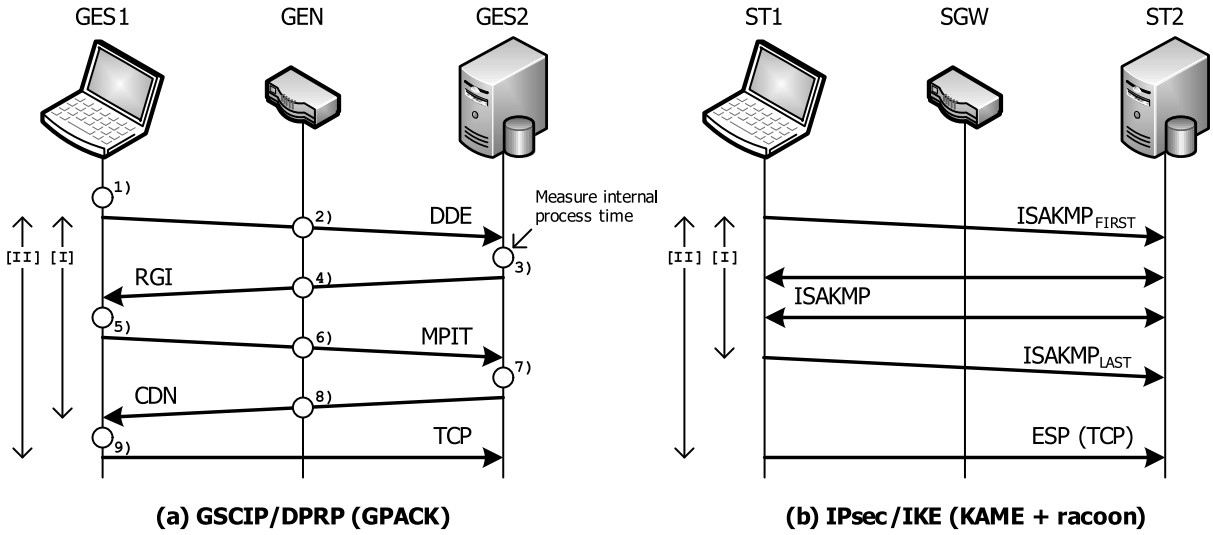

図 2.12　測定ポイント

この結果より，GPACK モジュールの処理時間は十分に短く，実用上問題ないことが分かった．本実験では中間 GE が 1 台だけであったが，エンド GE 間に n 台の GEN が存在し，全ての GE が上記と同じ処理時間と仮定すると，提案アーキテクチャにおける通信開始時のオーバヘッド *Delay* は以下の式で算出できる．

$$Delay = 2RTT + 44.32n + 159.02 \tag{2.10}$$

ここで，RTT はエンドノード間の RTT（Round Trip Time）値である．表 2.6 の結果は，実験環境における小さな RTT 値によるものである．実環境における RTT 値については，文献 [79, 80] が参考になる．一般に日本国内であれば，RTT 値は最大約 30 msec と見積もることができるため[3]，このような場合においても通信開始時のオーバヘッドは十分に小さい．

これらの測定結果より，DPRP は通信開始に先立つネゴシエーションであることを考えると，TCP 通信にはほとんど影響を与えることがないといえる．これに対し IKE では，DPRP の測定結果と比べて 3 桁以上遅い結果となっている．これは GSCIP と IPsec の通信開始時における認証の考え方の違いに起因している．

GSCIP では GE の起動時に GMS との間で公開鍵を用いた認証を行い，予めグループ鍵 GK を

表 2.6　オーバヘッドの測定結果

	GSCIP/DPRP	IPsec/IKE
[I]　ネゴシエーション時間	1.01	1105.95
[II]　通信開始までの時間	1.04	2994.03

単位：msec

表 2.7　GE における GPACK モジュールの内部処理時間

測定箇所	GES1	GEN	GES2
1)	43.06 (13.21)	—	—
2)	—	4.86 (0.00)	—
3)	—	—	46.26 (16.11)
4)	—	28.48 (15.29)	—
5)	42.36 (10.36)	—	—
6)	—	9.61 (3.57)	—
7)	—	—	16.17 (3.09)
8)	—	1.37 (0.00)	—
9)	11.17 (0.00)	—	—
合計	96.59 (23.57)	44.32 (18.86)	62.43 (19.20)

( ) 内は暗号化処理が占める時間　単位：μsec

[3]光が 1 km 進むためには約 5 msec が必要であり，光回線で結ばれた東京，沖縄間（約 1,500 km）においては約 7.5 msec の遅延が発生する．これにルータを通過する際に発生する機器遅延や通信経路による距離増加などを加味すると，RTT 値は約 30 msec となる．

表 2.8　FTP スループットの違い

	GSCIP 実装時	GSCIP 未実装時
スループット	82.15	82.31

単位：Mbit/s

取得しておく．これは認証機能の一部を前処理していることに相当する．そのため，GE 起動時は GMS との間の処理時間だけ遅延が発生するが，通信開始時は DPRP により共有秘密鍵である GK を用いたエンドノード間認証が行われるため，ノード間の通信開始時における遅延は十分に少ない．一方，IPsec は通信開始時にエンドノード間で事前共有秘密鍵や公開鍵，ディジタル署名などで認証し，かつ通信パケットを暗号化する共有鍵を DH 鍵交換 [81, 82] により別途生成している．このため，GSCIP と比べて通信開始時の認証に関わるオーバヘッドが大きい．

また通信開始までの時間については上記以上の大きな差が生じている．これは GSCIP/DPRP と IPsec/IKE の実装モデルの違いに起因している．DPRP は実装がシンプルなため全ての処理をカーネルで実行でき，カーネル内でのパケットの待避や復帰などの処理が可能である．そのため TCP の再送処理が発生することがなく，わずかな遅延で TCP 通信を開始することができる．一方，IKE は汎用的な利用を想定しているため，アプリケーションレベルで動作させており，カーネルに実装されている IPsec カーネルモジュールとリアルタイムに連携することが難しい．その結果，パケットを破棄して IKE ネゴシエーションを開始する．すなわち，実際に送信するパケットは TCP の再送処理に頼ることで通信を実現している．そのため TCP の再送タイムアウト RTO（Retransmission Time Out）の初期値である約 3 sec 後に IPsec ESP による暗号化通信が始まっている．

FTP のスループット値

FTP のスループット値は FreeBSD の FTP クライアントソフトに表示される値を採用した．測定方法は GES2 から 500 MByte のファイルをダウンロードした．GPACK 実装時と未実装時における FTP スループット値を表 2.8 に示す．GSCIP 実装時では 82.15 Mbit/s，GSCIP 未実装時では 82.31 Mbit/s となった．

FTP スループットでは，両者の差は 0.2 % 程度であった．即ち，PIT 検索のオーバヘッドは十分許容できる範囲である．DPRP は IP 層で動作するプロトコルであるため，UDP 通信の場合においても上記結果と同等の性能を得ることができる．

2.5.2　管理負荷

FPN におけるセキュア通信グループを GSCIP と IPsec で実現する場合に発生する管理負荷を評価し，提案する位置透過性の有効性を検証する．評価項目は初期管理負荷，ネットワークの構成変化時に発生する管理負荷，およびセキュア通信グループのメンバ構成変化時に発生する管理負

表 2.9　設定内容と項目数の比較

GSCIP/DPRP

	設定内容	項目数
グループ鍵	グループ番号，バージョン番号，鍵データ	3
GE 情報	動作モード（OP/CL），グループ番号	2

IPsec/IKE

	設定内容	項目数
共通秘密鍵	通信相手識別子，鍵データ	2
セキュリティポリシ	通信ペア識別子，処理内容， セキュリティプロトコル（ESP/AH）， カプセル化モード（Transport/Tunnel）， SGW ペア識別子，他	8 *1 14 *2 16 *3
IKE	通信相手識別子，交換モード（Main/Aggressive）， 暗号化アルゴリズム，ハッシュアルゴリズム，認証方式，他	12

*1 処理内容が “discard”/“bypass IPsec” の場合
*2 処理内容が “apply IPsec” で，カプセル化モードが Transport の場合
*3 処理内容が “apply IPsec” で，カプセル化モードが Tunnel の場合

荷とし，各管理負荷を算出する．ここでの構成変化とは引っ越し，人事異動や出張などオフラインでの移動による変化であり，通信中の移動は考えない．

GSCIP の場合と IPsec の場合における設定内容と，各設定 1 つあたりに必要な項目数の比較を表 2.9 に示す．GGSCIP ではグループ鍵と GE 情報の設定が必要で，各設定に必要な項目数はそれぞれ 3，5 である．一方，IPsec では事前にエンドノードで共有する秘密鍵，どの通信パケットに対してどのような処理を行うかを定めたセキュリティポリシ，および IKE の設定が必要である．セキュリティポリシは双方向定義する必要があり，処理内容やモードに応じて項目数が異なる．各設定に必要な項目数はそれぞれ 2，8〜16，12 である．IPsec における共有秘密鍵，セキュリティポリシ，IKE の各設定には通信相手識別子，通信ペア識別子，および自ノード識別子の項目が含まれており，管理者およびユーザはこれらの項目に IP アドレスまたは FQDN などのユーザ ID を設定する必要がある．

初期管理負荷

図 2.3，表 2.3 で表される通信環境を GSCIP および IPsec で実現するために，各装置に必要な初期管理負荷を表 2.10 に示す．ここで初期管理負荷とは表 2.9 で示した設定 1 つあたりに必要な項目数に，実際に設定する数を掛けた値である．GSCIP の場合，GES1 は 2 つのセキュア通信グループに所属するため，初期管理負荷の合計は 8 となる．同様に GEN，GES2 の初期管理負荷はそれぞれ 5 となる．一方，IPsec の場合，ST1 は 2 個の共有秘密鍵を保持し，ST2 に対するトランスポートモードのセキュリティポリシと IKE の設定が必要である．そのため初期管理負荷はそれ

表 2.10　初期管理負荷

GSCIP/DPRP

	GES1	GEN	GES2
グループ鍵	6	3	3
GE 情報	2	2	2
管理負荷の合計	8	5	5

IPsec/IKE

	ST1	SGW	ST2
共通秘密鍵	4	2	2
セキュリティポリシ	14 $^{\star 1}$	16 $^{\star 2}$	22 $^{\star 3}$
IKE	12	12	12
管理負荷の合計	30	30	36

$^{\star 1}$ “apply IPsec” (Transport)：14
$^{\star 2}$ “bypass IPsec”：8，“discard”：8
$^{\star 3}$ “apply IPsec” (Transport)：14，“discard”：8

表 2.11　ネットワーク構成変化時の動作処理情報の変化

通信ペア		通信	動作処理情報		
		可否	GES1	GEN	GES2
GES1	GES2	○	E2	T → —	E2
GES1	Term1	○	T → E1	— → E1	—
GES1	Term2	× → ○	D → T	D → —	—
GES2	Term1	×	—	D	D
GES2	Term2	×	—	—	D
Term1	Term2	×	—	D	—

E*x*: Encrypt/Decrypt by GK*x*　T: Transparent
D: Discard　—: No Record

ぞれ 4，14，12 となり，ST1 の初期管理負荷の合計は 30 となる．同様に SGW，ST2 の初期管理負荷は 30，36 となる．

GSCIP は GE が所属するグループ数の増加に伴い初期管理負荷も増加するが，その増分はわずかである．これに対して IPsec はトランスポートモードのセキュリティポリシを 1 つ設定する度に，初期管理負荷が両エンドノードおよび通信経路上に存在する SGW にそれぞれ 14，8 ずつ増加する．

ネットワーク構成変化時に発生する管理負荷

図 2.3 において GES1（IPsec では ST1）が NET1 から NET2 へ移動した際，ノード間で生成されるべき動作処理情報が表 2.3 に対してどのように変化するかを表 2.11 に示す．またこのような変化に対して発生する管理負荷を表 2.12 に示す．GSCIP ではノードが移動しても，その都度 DPRP により動作処理情報を新しく生成するため，ユーザや管理者が行う作業は一切発生しない．一方，IPsec で同様の構成を実現しようとすると，ST1 は移動により IP アドレスが変化するため，通信識別子を変更する必要がある．ST1 は ST2 に対するトランスポートモードのセキュリティポリシ

表 2.12　ネットワーク構成変化時の管理負荷（GES1 が NET1 から NET2 へ移動した場合）

GSCIP/DPRP

	GES1	GEN	GES2
グループ鍵	0	0	0
GE 情報	0	0	0
管理負荷の合計	0	0	0

IPsec/IKE

	ST1	SGW	ST2
共通秘密鍵	0	1（変更：1）	1（変更：1）
セキュリティポリシ	20（変更：4 $^{\star 1}$，追加：16 $^{\star 2}$）	16（追加：16 $^{\star 2}$）	4（変更：4 $^{\star 1}$）
IKE	1	0	0
管理負荷の合計	21	17	5

$^{\star 1}$ Transport mode の設定を変更　$^{\star 2}$ Tunnel mode の設定を追加

と IKE の設定を変更する必要があり，その管理負荷はそれぞれ 4，1 となる．さらに同一部門の Term1 と通信するために，SGW に対するトンネルモードのセキュリティポリシの設定を新たに追加する必要がある．その管理負荷は 16 となり，ST1 の管理負荷の合計は 21 となる．

図 2.3 のネットワーク環境はシンプルな構成であるため，移動後の ST1 と Term1 間の通信経路上に SGW が 1 台しか存在しないが，実際の環境を想定した場合，SGW の台数が 2 台以上存在することも十分考えられる．この場合，さらに設定追加に伴う管理負荷が増加する．

セキュア通信グループのメンバ構成変化時に発生する管理負荷

図 2.3 において Group 1 に所属し，開放モードに定義された GES3（IPsec では ST3）を新たに NET1 に配置する場合に発生する管理負荷を表 2.13 に示す．GSCIP では管理者が GMS において GES3 の GE 情報を追加定義する．GES3 は電源投入時に定義された GE 情報とグループ鍵を GMS から取得し，自動的に設定される（合計 5）．後は DPRP により動作処理情報を自律的に生成するため管理負荷はほとんど発生しない．一方，IPsec では ST3 に共有秘密鍵，セキュリティポリシ，および IKE の設定を行う必要がある（合計 30）．さらにメンバ構成の変化が発生するセキュア通信グループのメンバ全員（ST1，SGW）に共有秘密鍵，セキュリティポリシの設定を追加する必要があり，大きな管理負荷（ST1 の合計 16，SGW の合計 2）発生する．実際の環境では 1 つのセキュア通信グループに大勢のメンバがいることが想定されるため，さらに設定追加に伴う管理負荷が増加する．

これらのことから，GSCIP は初期導入時や，ノードの移動に伴う管理負荷が発生しないため，IPsec/IKE に対して大幅な運用管理負荷の軽減を実現できた．特にノード移動に伴う管理負荷に着目すると，提案システムと既存技術を比較するとその管理負荷の違いが顕著であり，位置透過性の有効性を実証できたといえる．

表 2.13　メンバ構成変化の管理負荷（GES3 追加の場合）

GSCIP/DPRP

	GES1	GEN	GES2	GES3
グループ鍵	0	0	0	3
GE 情報	0	0	0	2
管理負荷の合計	0	0	0	5

IPsec/IKE

	ST1	SGW	ST2	ST3
共通秘密鍵	2	2	0	4
セキュリティポリシ	14 $^{\star 1}$	0	0	14$^{\star 1}$
IKE	0	0	0	12
管理負荷の合計	16	2	0	30

$^{\star 1}$ Transport mode の設定を追加

2.6　結論

DPRP は FPN の前提となる個人単位とドメイン単位のセキュア通信グループが混在する環境において，ノード間の認証と暗号化通信に必要な動作処理情報を動的に生成し，位置透過性を実現することができる．

FreeBSD の IP 層を改造し，DPRP モジュールを組み込んだ．GE が送受信する通信パケットを IP 層から抜き出して処理を行い，差し戻すことで既存の処理に影響を与えない方式を実現した．DPRP の性能を測定した結果，高速かつ安全に通信相手を認証することが可能で，暗号化通信に必要な動作処理情報を動的に生成できることを確認した．

IPsec/IKE と性能を比較した結果，十分に短い時間でネゴシエーションを完了し，かつ TCP/UDP 通信に与える影響がほとんど無いことがわかった．また，ネットワークの物理構成の変更時における管理者やユーザの管理負荷について評価した結果，IPsec で FPN を構築した場合と比較して大幅な負荷軽減を実現でき，位置透過性の有効性を示した．

第3章　実用暗号通信方式PCCOM

3.1　研究の背景と目的

ネットワークにおけるセキュリティ上の脅威は年々深刻な問題となっており，セキュリティ技術の重要性が高まっている．その中でも，IPsec ESP [12, 66] のように IP 層でパケットの暗号化などを行うことによりネットワーク自体のセキュリティを確保するネットワークセキュリティ技術は，利用するアプリケーションを意識することなく安全を確保できることから，ネットワークの根本的なセキュリティ対策として期待されている．

しかし実際には，IPsec ESP は NAT/NAPT（以後 NAT と総称する）やファイアウォールを挟むような環境では使用することができず，普及が進んでいないのが現状である．このことから，ファイアウォールや NAT との共存が可能な暗号化通信は有効な技術と考えられる．しかし，セキュリティ強度と柔軟性・利便性といった実用度は相反する要素であり，ひとつの技術であらゆる要求に対応するのは困難である．従って今後のセキュリティ技術は，セキュリティ強度と実用度を想定する利用形態に応じて，それぞれに適した方式を検討することが重要になると考えられる．

IPsec ESP は，盗聴を防止する暗号化の他に，なりすましを防止する本人性確認（正当な相手であることの保証）や改竄を防止するパケットの完全性保証（パケットが改竄されていないことの保証）などの機能を提供している．また，ESP にはトランスポートモードとトンネルモードがあり，前者は End-to-End の IPsec 通信を適用する際に利用し，後者は主に Gateway-to-Gateway や Host-to-Gateway の IPsec 通信を適用する際に利用する．しかし現実の適用例を見ると，インターネット VPN（Virtual Private Network）の構築手段として Gateway-to-Gateway でトンネルモードを用いる例を除くとあまり普及していない．これは，パケットの暗号化や完全性保証がもたらす NAT やファイアウォールとの相性の悪さに起因している．

これらの課題を解決するために，UDP ヘッダで更に ESP パケットをカプセル化して NAT を通過させる方法（UDP Encapsulation of IPsec Packets） [40] が提案されているが，カプセルヘッダの部分は完全性保証の範囲に含めることはできず，ヘッダの追加によるオーバヘッドの増加やフラグメントの発生などの課題が発生する．また，ESP の暗号化を階層化し，TCP/UDP ヘッダの内容をルータやファイアウォールが参照できるようにする ML-IPsec（Multi-Layer IPsec） [83] が提案されているが，この方法では既存のシステムを変更する必要がある．

一方，文献 [30] ではパケットフォーマットを変えないまま特定の範囲を暗号化する方式が提案されている（以下，置換方式と呼ぶ）．置換方式は，ポート番号を平文のままとするため，ファイアウォールの通過が可能であり，パケットフォーマットを変えないためヘッダオーバヘッドやフラグメントが発生せず高スループットを実現できるという利点がある．しかし，置換方式では

TCP/UDP チェックサム [84–86] を暗号化範囲に含めているため，NAT によるチェックサムの書き換えに対応できず，NAT を通過することができない．また単に平文と暗号文を置き換えるだけのため，本人性確認とパケットの完全性保証を実現していない．

本章では置換方式の利点に着目し，置換方式を改良することによって，NAT とも共存でき，かつ本人性確認とパケットの完全性保証も確実に行える暗号通信方式 PCCOM（Practical Cipher Communication）[35] を提案する．PCCOM は本人性確認とパケット全体の完全性保証を，共通秘密鍵とパケットの内容から生成した疑似データと呼ぶ値を用いて，TCP/UDP チェックサムを新たに再計算することにより実現する．この方法によると NAT やファイアウォールと共存することが可能で，かつパケットフォーマットを変えないためヘッダオーバヘッドやフラグメントが発生せず高スループットを実現できる．なお，PCCOM は事前に送信側と受信側で共通秘密鍵を共有していること，パケットの処理内容を記述した動作処理情報テーブルを既に保持していることを前提としている．

PCCOM の有効性を確認するために試作システムを開発した．PCCOM がパケットフォーマットを変えずに処理する方式であることが，実装の容易さをもたらし，性能的にも有利であることについて述べる．評価の結果，高スループットを実現できることを確認した．また，PCCOM の安全性評価を行い，IPsec ESP とのすみわけについて考察した．

本章の構成は以下のとおりである．3.2 節で既存技術とその制約について説明した後，3.3 節で実用暗号通信 PCCOM を提案する．3.4 節では PCCOM の実装について説明し，3.5 節では実装したシステムを用いた PCCOM の性能評価を行い，PCCOM の安全性評価と，IPsec ESP とのすみわけについて述べる．最後に 3.6 節でまとめる．

3.2 既存技術

IPsec ESP のトランスポートモードとトンネルモードのパケットフォーマットを図 3.1 に示す．トランスポートモードでは，IP ヘッダとそのペイロードの間に ESP ヘッダを挿入し，元の IP パケットのペイロード部分を暗号化する．トンネルモードでは，セキュリティゲートウェイ（以後 SGW）のアドレスを含む新しい IP ヘッダでカプセル化し，カプセル内のデータ，すなわち元の IP パケットを暗号化する．ESP トレーラは，ブロック暗号のブロック長の整数倍に暗号化するデータの長さを揃えるために用いる．また，ESP ヘッダから ESP トレーラまでの完全性を保証する認証値 ICV（Integrity Check Value）を計算し，ESP 認証値（ESP Auth）としてパケットの末尾に付加する．

いずれのモードにおいても TCP/UDP のポート番号が暗号化範囲に含まれているため，そのパケットがどのような用途に用いられるかがファイアウォールで判別できない．その結果，ファイアウォールでは全ての IPsec トラヒックの通過を禁止してしまう場合が多い．また，TCP/UDP チェックサムフィールドが暗号化範囲・完全性保証の範囲に含まれているため，IP アドレスの変換を伴う NAT を通過すると偽造パケットと見なされ，IPsec 処理によってパケットが破棄される．これは TCP/IP が綺麗な階層構造になっておらず，TCP/UDP チェックサムでありながら IP アドレスも

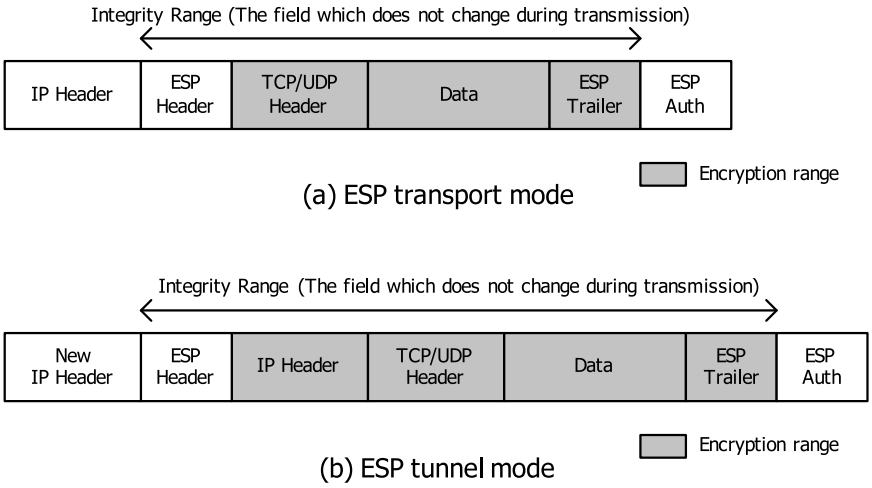

図 3.1　IPsec ESP のパケットフォーマット

図 3.2　置換方式のパケットフォーマット

チェックサムの演算範囲に含んでいることが根本的な理由である．トンネルモードにおいては，IP アドレスのみを変換する純粋の NAT を通過することは可能であるが，ポート番号の変換も伴う NAPT（IP マスカレード）は通過できない．

このような状況に対処するために，市販のルータにおいて，UDP ポート 500 番のエントリを持っているノードに対して ESP パケットを転送することで NAT を通過させているものがある（IPsec パススルーと呼ぶ）が，この方法ではひとつのノードだけしか ESP の通信は機能しない．一方 IETF では，UDP ヘッダで更に ESP をカプセル化して NAT を通過させる方法（UDP Encapsulation of IPsec Packets）が提案されているが，カプセル部分は完全性保証の範囲に含むことはできず，ヘッダの追加によるオーバヘッドの増加やフラグメントの発生などの課題が発生する．

以上のことから，IPsec をシステムに導入するには既存設備との相性やスループットの低下を考慮する必要がある．

図 3.2 に，PCCOM のベースとなっている置換方式のパケットフォーマットを示す．暗号化後のパケットフォーマットはオリジナルフォーマットから変化させず，平文と暗号文をそのまま置き換える．ファイアウォールがポート番号を識別できるように，また TCP/UDP チェックサムから暗号文の内容が推測されるのを防ぐために，暗号化範囲を TCP/UDP ヘッダのチェックサムフィールド以降の全ての部分としている．TCP/UDP ポート番号が平文であるため，ファイアウォールによるフィルタリングが有効になるうえ，パケット長が変わらないためスループットの低下が少ないという利点がある．この方式はイントラネット内では有効であるが，TCP/UDP チェックサムフィールドが暗号化範囲に入っているためチェックサムの書き換えを伴う NAT を通過できない．また，本人性確認とパケットの完全性保証を実現していないため，なりすましや改竄の恐れがある．

3.3 提案方式

PCCOM が提供する機能は，暗号化による機密性確保，本人性確認とパケットの完全性保証である．また，NAT やファイアウォールとの共存ができ，パケットフォーマットを変えないため高スループットを実現できるなどの特徴がある．なお，IP アドレスとポート番号は NAT で内容が変換されるため完全性保証の範囲に含めない．この部分の保証に関しては，パケットの処理内容を記述した動作処理テーブルの検索過程でその内容を保証する．

3.3.1 PCCOM の原理

PCCOM のパケットフォーマットを図 3.3 に示す．PCCOM では，共通秘密鍵とパケットの内容から生成した疑似データと呼ぶ値を用いて，TCP/UDP チェックサムに独自の計算を施すことにより，本人性確認とパケットの完全性保証を行う．以下にその原理を示す．

PCCOM では本人性確認と完全性保証を実現するために，まず CB（Checksum Base）と呼ぶチェックサムベース値を定義する．CB は図 3.4 に示すように，IP ヘッダと TCP/UDP ヘッダで転送中に値の変化しないフィールド[1]，および事前に秘密裏に共有している共通秘密鍵を含めた値から生成したハッシュ値である．CB の種には共通秘密鍵の他に，シーケンス番号のように初期値が乱数で決まりパケットごとに値が変化するフィールドを含んでおり，CB 値を第三者が推測するのは極めて困難である．この CB は，以下のように本人性確認とパケットの完全性保証を実現するための

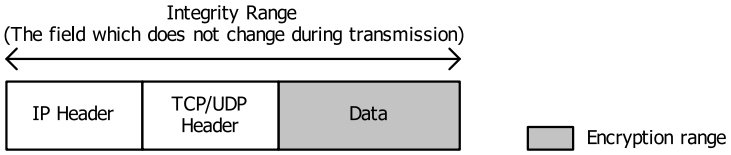

図 3.3　PCCOM のパケットフォーマット

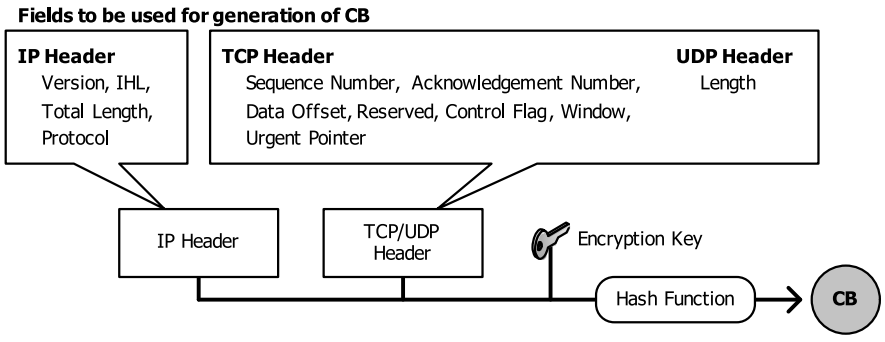

図 3.4　CB の生成方法

[1]IP ヘッダはバージョン，ヘッダ長，パケット長，プロトコルの 4 フィールド．TCP ヘッダはシーケンス番号，確認応答番号，データオフセット，予約，コードビット，ウィンドウサイズ，緊急ポインタの 7 フィールド．UDP ヘッダはパケット長の 1 フィールド．

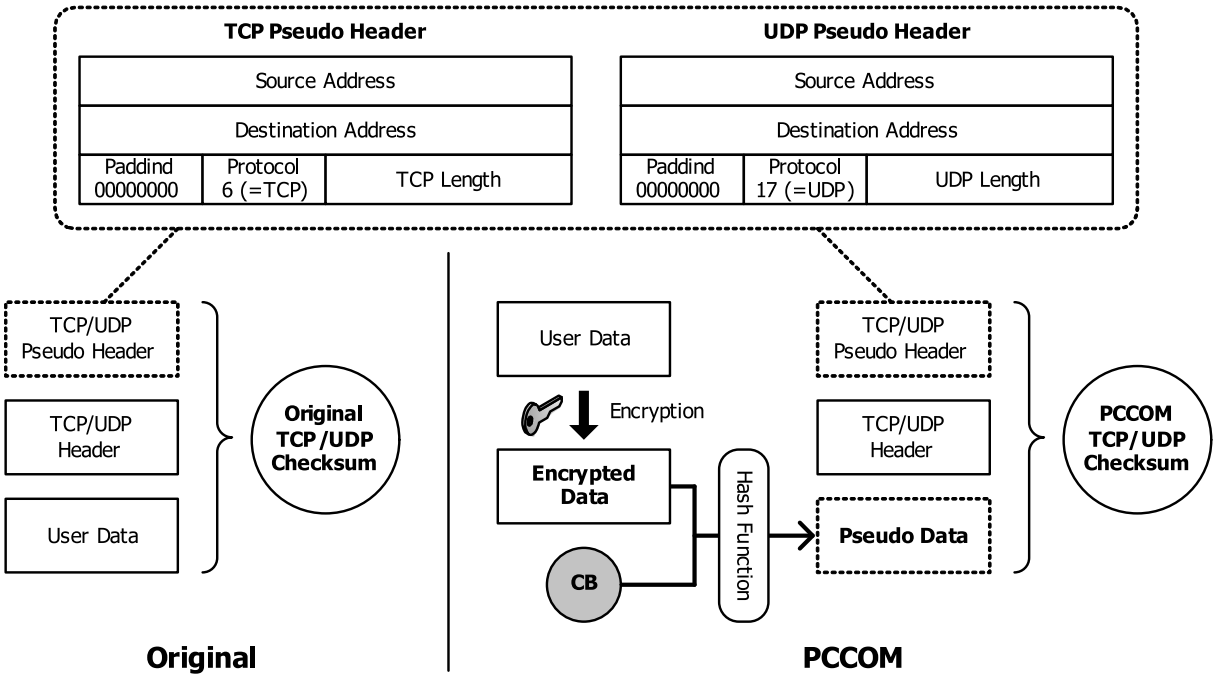

図 3.5　チェックサム計算範囲の違い

キーデータとなる．

一般通信と PCCOM の，TCP/UDP チェックサムの計算範囲の違いを図 3.5 に示す．図中の点線はチェックサム計算時に疑似的に作成する情報を指す．一般の通信では TCP/UDP チェックサムは，TCP/UDP ヘッダ，TCP/UDP 疑似ヘッダ，ユーザデータから計算される．ここで，TCP/UDP 疑似ヘッダには IP アドレスの値を含む．このため，NAT を経由して IP アドレスが変わると，TCP/UDP チェックサムも書き換えが必要となる．一方，PCCOM では TCP/UDP チェックサムは，TCP/UDP ヘッダ，TCP/UDP 疑似ヘッダ，疑似データから計算される．ここで，疑似データとは暗号化後のデータと CB を元に求めたハッシュ値である．

完全性保証のプロセスを以下に述べる．送信側ではデータの暗号化後，上記疑似データを用いて TCP/UDP チェックサムの再計算を行う．受信側ではデータの復号を行う前に，同様の方法で生成した疑似データを用いて TCP/UDP チェックサムを検証する．検証結果が正常であれば，復号を行いオリジナルチェックサムの再計算を行って上位層（TCP/UDP）に渡す．この方式により，暗号化データと CB 生成に用いたフィールドの完全性を保証することができると同時に，本人性確認も実現される．パケットの改竄者が改竄を隠蔽するために，パケットの一部を書き換えると同時に TCP/UDP チェックサムを再計算しようとしても，疑似データの内容が分からないので正しい計算を行うことはできない．なお，IP アドレスとポート番号は NAT にて変換されるので CB 生成の範囲には含めない．IP アドレスとポート番号の保証方法については次項で述べる．

上記の演算方式によると，通信経路上に NAT が介在して IP アドレス，ポート番号，チェックサムが書き換えられたとしても，完全性保証，本人性確認の考え方は維持される．なぜなら，NAT における TCP/UDP チェックサムの書き換えは，文献 [21] で規定されているように変換部分の差分を

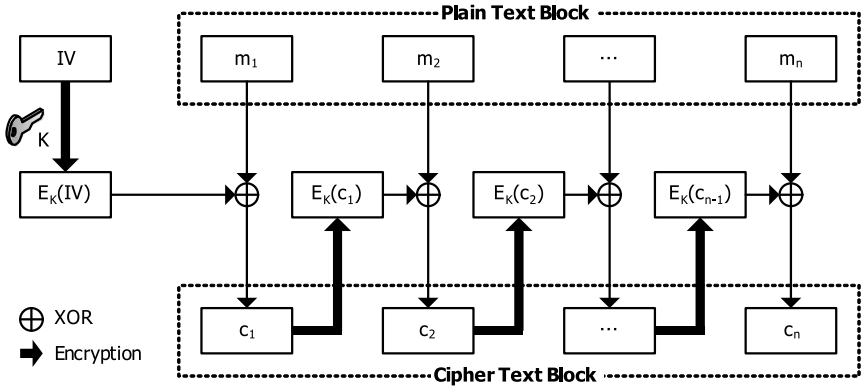

図 3.6　CFB モードにおける暗号化

計算するだけであり，受信側で行うチェックサムの検証には影響を与えないためである．PCCOM ではパケットの暗号化範囲はユーザデータ部分のみとしているが，本人性確認とパケット全体の完全性保証が施されているため，パケットの偽造による TCP セッションハイジャックなどの攻撃を防ぐことができる．また，PCCOM ではファイアウォールが TCP/UDP ヘッダの内容を用いたフィルタリングを行うことが可能であるため，実用面でのメリットが大きいと考えられる．

暗号アルゴリズムとしては AES [70] の CFB（Cipher FeedBack）モードを採用した．図 3.6 に CFB モードにおける暗号化の仕組みを示す．平文 M を n 個のブロック $m_1, \cdots, m_n$ に区切ると，各ブロックの平文 m_i と前のブロックの暗号文ブロック c_{i-1} を暗号鍵 K により暗号化した $E_K(c_{i-1})$ との XOR を暗号文ブロック c_i とする．出力された暗号文ブロック $c_1, \cdots, c_n$ を結合することにより，暗号文 C を得る．最初のブロックの暗号化には初期ベクトル IV（Initialization Vector）が必要となるが，PCCOM では CB 値を流用する．

$$C = \{c_1, c_2, \cdots, c_n\} \qquad c_i = \begin{cases} m_i \oplus E_K(IV) & (i = 1) \\ m_i \oplus E_K(c_{i-1}) & (i \neq 1) \end{cases} \tag{3.1}$$

CFB モードはブロック暗号化処理の 1 つで，ブロック長を任意に設定できるため，バイトまたはビット単位で暗号化することができる．ストリーム暗号と同様にパディングは不要で，常に平文サイズと暗号文サイズが一致する．従って，高スループットが実現でき，かつフラグメントの発生を懸念する必要がない．

3.3.2　IP アドレス・ポート番号の保証

PCCOM では，IP アドレスとポート番号は NAT を経由する際に値が変化するため CB 生成の範囲に含めていない．そのため，このままでは通信経路上で送信元アドレスの改竄や，ポート番号の改竄によるアプリケーションの誤作動などを招く可能性がある．これらを防ぐために，IP アドレスとポート番号の完全性は，パケットの処理内容を記述した動作処理情報テーブルの検索過程で保証する．

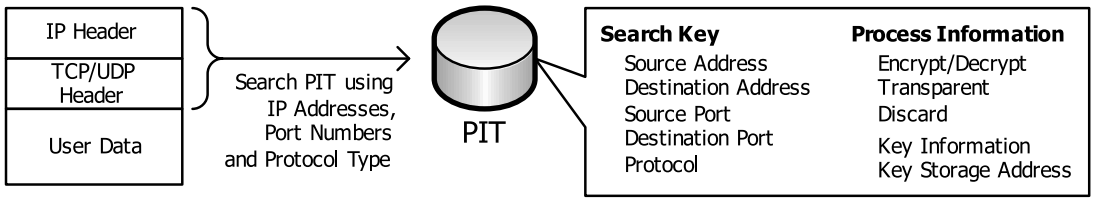

図 3.7　テーブル検索処理

動作処理情報テーブルとは IPsec における SAD（Security Association Database）や，2 章で述べた本研究における PIT（Process Information Table）に相当する．図 3.7 に動作処理情報テーブルの検索処理を示す．テーブル内には送信元と宛先の IP アドレスとポート番号，プロトコル番号とそれに対応するパケットの処理内容（暗号化/復号，透過中継，破棄），使用する共通秘密鍵に関する情報などが記述されている．送信側と受信側の両ノードは通信の開始前に設定情報の交換を行い，両ノードで通信パケットの処理に必要となる動作処理情報テーブルを生成してカーネルに保存する．

送信側ノードはパケット送信時に，受信側ノードはパケット受信時に，パケットの IP アドレス，ポート番号，プロトコル番号をキーに動作処理情報テーブルを検索し，その内容に従って暗号化/復号などの処理を実行する．従って受信側の動作処理情報テーブルを検索後，テーブルの内容から IP アドレス，ポート番号，プロトコル番号を再度確認し，テーブル内に該当パケットの情報が正しく存在したら，IP アドレスとポート番号は改竄されていなかったことが保証される．なお，一定時間以上参照されない動作処理情報テーブルのレコードは削除される．また，事前に設定した有効期限より長い間通信が継続された場合には，再度その内容を更新するための手続きが実行される．

この方式は事前に正しい内容のテーブルが生成されていることが前提となる．正しいテーブルの生成を保証する方式としては，IKE（Internet Key Exchange） [65] などの既存の技術や，本研究における DPRP（Dynamic Process Resolution Protocol） [34] を流用することが可能である．ここで，IKE は安全な通信路を確立する SA（Security Association）と共通秘密鍵を管理するプロトコルであり，例えば，受信側での SA の特定には IP アドレス，ポート番号，プロトコル番号を用い，PCCOM 特有の部分を PCCOM DOI（Domain of Interpretation）として定義することにより，動作処理情報テーブルの生成が実現できる．

3.4　実装

PCCOM の試作システムを開発し，動作検証を行った．本節では試作システムの実装方式，仕様・構成と動作概要について記述する．

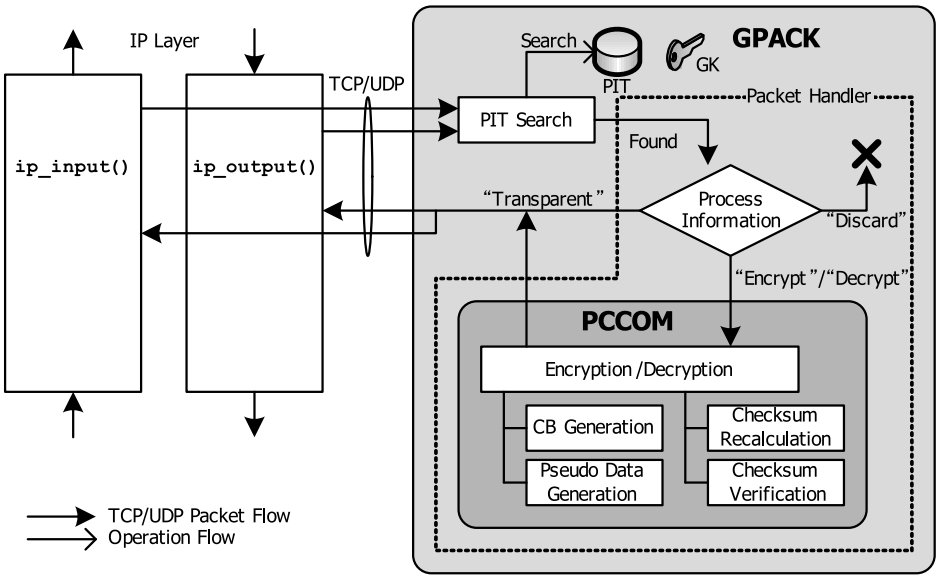

図 3.8　試作システムの実装方式

3.4.1　実装方式

試作システムは，FreeBSD 5.3-RELEASE のカーネル内に実装した．試作システムの実装方式を図 3.8 に示す．PCCOM は 2.4 節にて示した GSCIP のモジュール GPACK の一部として実装される．IP 層で行われる既存の処理に一切の変更を加えず，IP 層の入出力関数である `ip_input()`，`ip_output()` から GSCIP モジュールへ処理を渡し，動作処理情報テーブル PIT を検索する．該当する動作処理情報が存在し，処理内容が暗号化/復号であれば PCCOM モジュールが暗号化処理を行う．一連の処理を完了したら，`ip_input()`，`ip_output()` に差し戻す．

PCCOM はパケットフォーマットを変えずに処理する方式であるため，この様な方式を容易に実現できる上，高スループットを発揮できるという利点がある．一方，IPsec はヘッダの追加などパケットフォーマットに変更があるため，IP 層全体に渡って処理の変更が必要となる．

3.4.2　システムの仕様・構成と動作概要

試作システムの仕様を表 3.1 に示す．動作処理情報テーブルは 2.4 節において実装したものを利用した．暗号化アルゴリズムの AES およびハッシュ関数の MD5 には，暗号ライブラリである OpenSSL（openssl-0.9.7d） [73] を採用した．なお，鍵長は 128 bit とした．

PCCOM モジュールはメインモジュールとサブモジュールから構成される．メインモジュールでは暗号化/復号モジュールや各サブモジュールを呼び出す処理を行う．サブモジュールは，CB 生成モジュール，疑似データ生成モジュール，チェックサム再計算モジュール，チェックサム検証モジュールから構成される．PCCOM モジュールは通信パケットに対し，予め作成済みの動作処理情報テーブルの処理内容に基づき処理を実行する．動作処理情報テーブルには IP アドレス，ポート番号，プロトコル番号と，それに対応する処理内容，すなわち暗号化/復号，透過中継，破棄な

表 3.1　試作システムの仕様

項目	内容
テーブル検索方式	ハッシュ法
暗号アルゴリズム	AES（CFB モード）
鍵長	128 bit
ハッシュ関数	MD5

表 3.2　実験ノードの仕様

項目	内容
CPU	Pentium4 2.4 GHz
Memory	256 MByte
NIC	10BASE-T，100BASE-TX，1000BASE-TX
OS	FreeBSD 5.3-RELEASE

どが記されている．

試作システムを用いて，パケットフィルタリングタイプのファイアウォールおよび NAT を中継して通信できることを確認し，パケットの内容を書き換えた場合，不正パケットとして検出できることを確認した．

3.5　評価

3.5.1　試作システムの性能評価

試作システムを実装した 2 台のノード間の通信性能を測定した．参考のために IPsec ESP（KAME [75]）を実装した場合を測定し比較した．また，PCCOM 内部の処理時間をモジュール別に測定し，処理のボトルネックとなっている部分を明らかにした．実験に用いたノードの仕様を表 3.2 に示す．IPsec の設定は，試作システムの仕様と条件が同じになるように，ESP トランスポートモードで，暗号アルゴリズムは AES（鍵長は 128 bit），認証アルゴリズムは HMAC-MD5 [87] とし，リプレイ防御機能は OFF とした．

通信性能の測定

図 3.9 は IP パケット長とスループットの関係を，10BASE，100BASE，1000BASE の通信環境ごとに，暗号化をしない場合（以下，Normal と呼ぶ），PCCOM の場合，IPsec ESP の場合のそれぞれについて示したものである．スループットの測定にはネットワークベンチマークソフト Netperf [88] を用いて，10 回試行の平均値をとった．

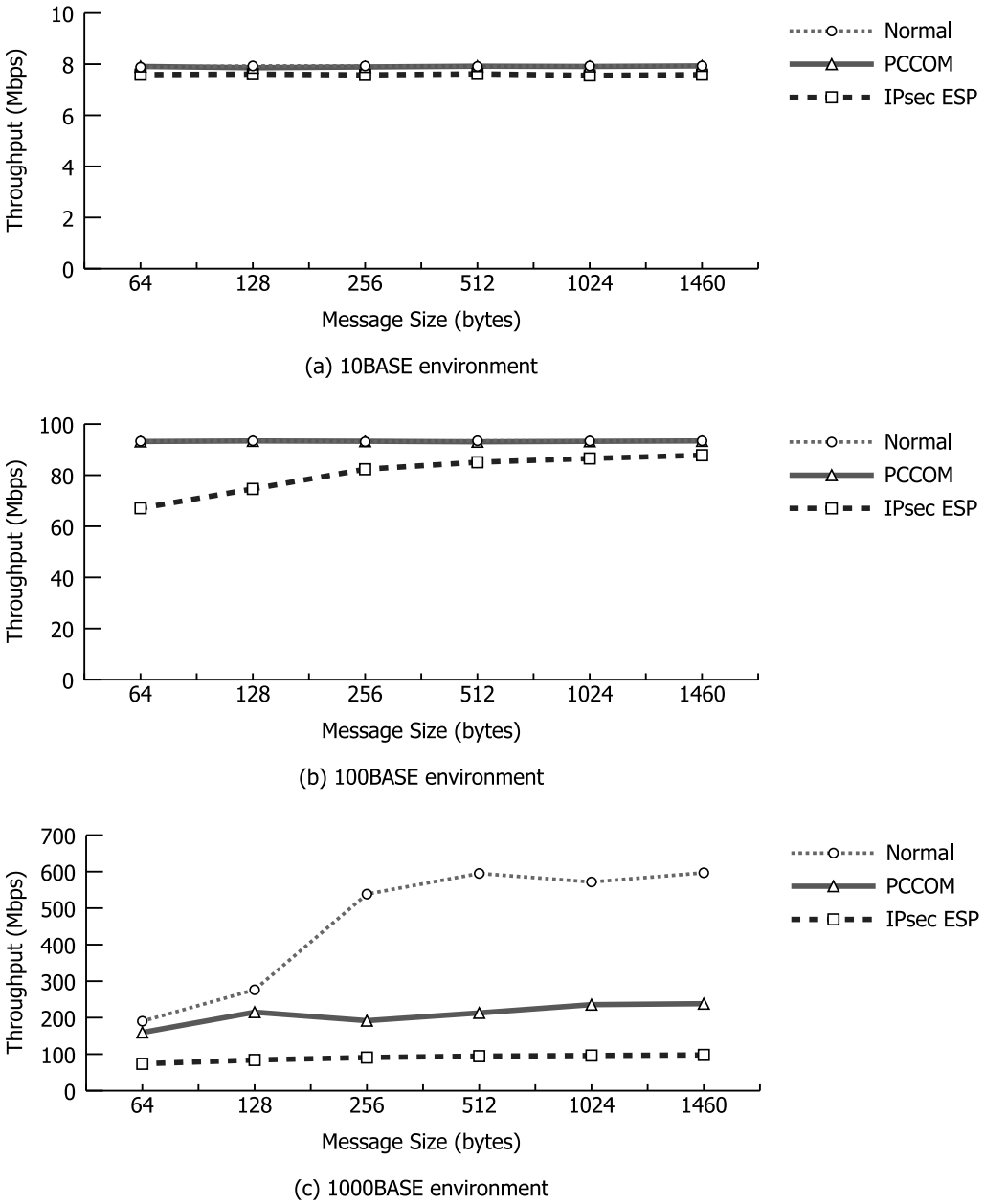

図 3.9　スループット測定結果

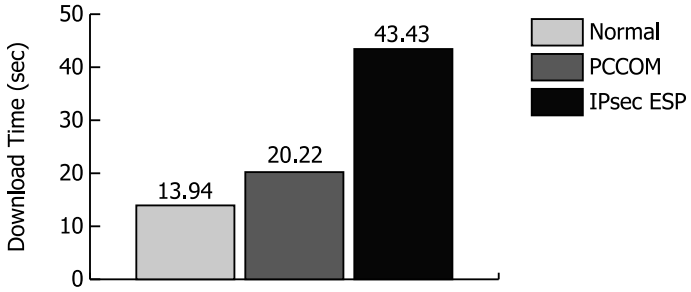

図 3.10　500 MByte のファイルの FTP ダウンロード時間

10BASE の環境では，ESP においては若干の性能低下が見られたものの，処理すべきパケット数が少ないため，PCCOM，ESP とも処理オーバヘッドはボトルネックとなっていない．100BASE の環境では，Normal と PCCOM は NIC の上限性能を発揮しており PCCOM に性能低下は見られなかった．それに対して ESP はメッセージサイズ 1460 byte のパケット（以下，長パケットと呼ぶ）では Normal から約 6 % 性能が低下しており，メッセージサイズ 64 byte のパケット（以下，短パケットと呼ぶ）では約 28.1 % 低下している．また 1000BASE の環境では，長パケットの場合 PCCOM は Normal から約 60.1 % 性能が低下しており，ESP では約 83.6 % 低下している．短パケットの場合 PCCOM は Normal から約 16.2 % 性能が低下しており，ESP では約 61.3 % 低下している．

パケットサイズが短くなるほどスループットが落ち込むのは，相対的に処理すべきパケット数が多くなるので，ソフトウェアによるオーバヘッドの占める割合が大きくなるためである．とりわけ ESP の短パケットでは，ヘッダの追加など暗号化以外の処理がボトルネックとなっており，その影響が顕著に現れているといえる．

次に，1000BASE の環境において，FTP で 500 MByte のファイルをダウンロードするのに要した時間を図 3.10 に示す．測定結果は 10 回試行の平均値である．PCCOM は Normal の約 145.1 % の時間であるのに対し，ESP は約 311.6 % の時間を要している．

PCCOM 内部の処理コスト

PCCOM における処理過程での処理コストを調べるために PCCOM の内部処理時間をモジュール別に測定した．内部処理時間は，RDTSC（Read Time-Stamp Counter） [78] を用いて処理前後の CPU クロックカウンタ値を求めて算出した．なお，PCCOM モジュールとは直接関係ないが，GPACK モジュールにおける PIT 検索に係わる処理時間についても測定した．

内部処理時間とそれぞれの比率を表 3.3 に示す．測定結果は FTP の通信中に流れた IP データグラム長 1460 byte のパケット 10 個の結果の平均値である．表 3.3 より，送信側，受信側ともに暗号化/復号処理が全体の 80 % 以上を占めていることが分かる．専用のハードウェア暗号エンジンを用いるなどで，処理時間の大幅な短縮が期待でき，より Normal に近い性能を発揮できると考えられる．また，動作処理情報テーブルの検索処理は約 0.27 μsec と全体の約 1 % 程度であり，検索処理のオーバヘッドは問題とならない．

表 3.3　内部処理時間とそれぞれの比率

	測定対象	処理時間（μsec）	比率（%）
送信側	GPACK（PIT 検索以外）	0.547	1.8
	GPACK（PIT 検索）	0.268	0.9
	暗号化	26.043	87.6
	CB 生成	0.868	2.9
	疑似データ生成	1.704	5.7
	チェックサム再計算（独自）	0.294	1.0
受信側	GPACK（PIT 検索以外）	0.545	1.7
	GPACK（PIT 検索）	0.269	0.8
	復号	25.547	80.6
	CB 生成	0.890	2.8
	疑似データ生成	2.863	9.0
	チェックサム検証（独自）	0.281	0.9
	チェックサム再計算（通常）	1.286	4.1

PCCOM の安全性

PCCOM が提供する機能はデータの機密性確保，本人性確認，パケットの完全性保証である．その上で考えられる脅威を以下に述べる．

PCCOM では IP ヘッダ，TCP/UDP ヘッダが平文であるためトラフィックの内容を解析される恐れがあるが，ファイアウォールの通過を可能とするにはヘッダ部分がファイアウォールに見えることが必須である．すなわち，ファイアウォールのパケットフィルタリングを可能にすることとトラフィック解析を不可とすることを同時に満足させることはできない．PCCOM は前者に重点を置くシステムを対象とする場合に有効である．

PCCOM では認証値がチェックサムフィールド長 16 bit であるため，2^{-16} の確率でパケットの偽造や完全性保証範囲のフィールドの改竄に成功する．パケットの偽造を利用した代表的な攻撃として TCP セッションハイジャックが考えられるが，ハイジャックを成功させるには，通信を中断させる RST パケット，再接続の SYN パケットとその SYN/ACK に対する ACK パケットの，3 ステップのパケットの偽造を成功させる必要があるため，実際にハイジャックに成功する確率は極めて低い．仮にハイジャックに成功したとしても，ユーザデータは暗号化範囲であるため意図したデータを送ることはできない．

また，ユーザデータは暗号化されているため意図した改竄は困難である．仮にユーザデータ部分が改竄された場合，受信側で行う復号の結果が予期せぬ値となり，多くのアプリケーションではエラー処理により破棄される．音声などのストリーミングのパケットは，改竄されて予期せぬ復号結果となった場合はノイズとして扱われる．

PCCOM では，IP アドレスとポート番号は NAT を経由する際に値が変化するため完全性保証の範囲に含めていない．これらの完全性は，パケットの処理内容を記述した動作処理情報テーブルの検索過程で保証する．従って，通信経路上で送信元アドレスの改竄や，ポート番号の改竄によ

るアプリケーションの誤作動などを招く行為を防ぐことができる．

動作処理情報テーブルはカーネル内に保存されるため，カーネルをハックされない限りその内容を改竄することは困難である．またテーブルを生成する際には，両ノード間の確実な認証を必要とするため，誤った情報登録の可能性は低いと考えられる．

IPsec ESP とのすみわけ

IPsec ESP と PCCOM を 7 項目において定性的に比較した結果を表 3.4 に示す．

IPsec ESP は，高い機密性と強力な認証機能を提供しているが，TCP/UDP ヘッダの暗号化や完全性保証が原因で NAT やファイアウォールと共存することができない．また，ヘッダの追加によるオーバヘッドやフラグメントが発生する．

PCCOM は，パケットフォーマットを変えないまま本人性確認とパケットの完全性保証を実現しており，NAT やファイアウォールと共存することができる．また，フラグメントが発生せず，高スループットを実現できるというメリットがある．暗号化範囲はポート番号によるフィルタリングを可能とするためユーザデータ部分のみとしているが，本人性確認・完全性保証の実現により TCP/UDP ヘッダが平文であることによる安全性低下を防止している．IP ヘッダ，TCP/UDP ヘッダは平文であるため，トラフィック解析をされる懸念があるが，ファイアウォールのパケットフィルタリングによって，管理者が許可した用途のパケットのみを通過させることができるという利点がある．

IPsec ESP は，強靭なセキュリティを必要とする部門への適用が適しており，通信経路上に NAT やファイアウォールが存在してはいけない．また，スループットの低下が問題とならないことを確認する必要がある．用例としては，イントラネット内部でも特に強靭なセキュリティを要する部門や，インターネット上で拠点間通信などの重要データの取引が行われるような環境に適している．

それに対し PCCOM は，NAT やファイアウォールとの共存が可能で，高スループットを実現できるなどの理由で，比較的広範囲への適用が可能と考えられる．用例としては，高スループットを要するアプリケーションの通信形態として多い P2P 通信や，パケットフィルタリングタイプの

表 3.4　IPsec ESP との比較

	IPsec ESP	PCCOM
機密性	◎	○
本人性確認	◎	○
完全性保証	◎	○
NAT	×	○
ファイアウォール	×	○
フラグメント	×	○
トラフィック解析	○	△

ファイアウォールを備えたホームネットワークへのアクセス，部門ごとにファイアウォールを設置している場合が多いイントラネット内の通信に有効と考えられる．

3.6 結論

NAT やファイアウォールと共存でき，オリジナルパケットのフォーマットを変えないまま，本人性確認とパケットの完全性保証を行うことができる暗号通信方式 PCCOM を提案した．PCCOM は本人性確認と IP アドレス・ポート番号を除くパケットの完全性保証を，共通秘密鍵とパケットの内容から生成した疑似データと呼ぶ値を用いて，TCP/UDP チェックサムを再計算することにより実現する．また，IP アドレスとポート番号については動作処理情報テーブルを検索する過程でその内容を保証する．

PCCOM の有効性を確認するために試作システムを実装し，動作検証を行った．性能測定の結果，高スループットが得られることを確認した．また，PCCOM の安全性について考察し，IPsec ESP とのすみわけが可能であることを示した．

第4章　移動透過性プロトコルMobile PPC

4.1　研究の背景と目的

ノート PC や PDA などのモバイル機器を持ち歩き，行く先々でインターネットに接続して利用するユーザが増加している．この様な状況下では，通信中にユーザが移動しても，通信を継続できることが要求される．TCP/IP では，IP アドレスがノード識別子としての役割だけでなく位置の情報も含んでいるため，ノードがネットワークを移動すると異なる IP アドレスが割り振られる．

トランスポート層では IP アドレスが通信識別子の一部として用いられており，IP アドレスが異なると別の通信と見なされ通信を継続することができない．この課題を解決するために，ノードが移動しても通信を継続できる機能を移動透過性と呼び，これまで多くの方式が研究されている [14]．移動透過性とはノードの位置に依存せず通信を開始できる移動ノード到達性と，ノードが移動しても通信相手との間に確立したコネクションを維持する通信継続性に分けられ，両者を実現する必要がある．

移動透過性の研究を大きく分類すると，特殊な中継サーバを用いるプロキシ方式と，それを必要としないエンドツーエンド方式がある．プロキシ方式は，移動ノード MN（Mobile Node）と通信相手ノード CN（Correspondent Node）の間にプロキシサーバが介在し，プロキシサーバが MN の IP アドレス変化を CN から隠蔽する．エンドツーエンド方式はエンドノード間で移動に伴う課題を解決し，上位ソフトウェアに対して IP アドレスの変化を隠蔽する．また，別の分類方法として，移動透過性を実現するレイヤの違いにより，ネットワーク層で実現する方式とトランスポート層で実現する方式がある．トランスポート層では通信識別子の制御がやりやすいという利点があるが，TCP または UDP のどちらに適用するかによりその方式が異なる．これに対し，ネットワーク層での実現方法は，TCP/UDP のいずれにも対応できる点で有効である．

Mobile IP [15,89–92] は，プロキシ方式をネットワーク層で実現する．プロキシサーバとして MN の位置を管理するホームエージェント HA（Home Agent）を導入し，CN 側から MN への通信パケットは HA が代理受信し，MN へトンネリング転送を行う．MN 側から CN への通信パケットは直接送信される．MN は移動しても変化しないホームアドレスを保持しており，CN は MN 宛の通信パケットの宛先をホームアドレスとしているため，MN が移動しても通信識別子が変化せず通信を継続できる．Mobile IP は IETF（Internet Engineering Task Force）での十分な検討を経て確立された技術であるが，HA という特殊な装置が必要である他，通信経路に冗長が発生したり，トンネリング転送時に余分なヘッダが必要になるなどの問題点がある．

MSOCKS [93] は，プロキシ方式をトランスポート層で実現する．プロキシサーバとして，SOCKS サーバ [59] を導入する．DNS サーバには，MN のホスト名に対して SOCKS サーバの IP アドレス

を登録する．CN は SOCKS サーバを通信相手ノードであると認識する．SOCKS サーバは MN と SOCKS サーバ間，SOCKS サーバと CN 間で確立された異なる TCP コネクションを結合しなおすことにより通信を継続する．MSOCKS は，ヘッダオーバヘッドは発生しないが，両方向の通信とも SOCKS サーバを経由するので冗長な経路が発生する．

TCP-R [94]，TCP Migrate [95]，MMSP [96] はエンドツーエンド方式をトンランスポート層において実現する．TCP-R，TCP Migrate は，MN の IP アドレスが変化したときに，TCP オプションフィールドを用いて MN から CN に変更情報を通知し，エンドノード間で TCP コネクションを張り直す．この方式では，TCP 機能の拡張が必要であり，またアプリケーションも TCP に限定される．MMSP は，UDP を拡張し，MN の IP アドレスが変化したときに，独自に定義したパケットで相手に通知する．IP アドレスの通知が完了するまでの間，新旧の IP アドレスを保持しておくことなどによりパケットロスを軽減する工夫をしている．この方式では，アプリケーションが MMSP に対応している必要があり，かつ UDP に限定される．

LIN6（Location Independent Networking for IPv6）[97]，MAT（Mobile IP with Address Translation）[98] はエンドツーエンド方式をネットワーク層において実現する．LIN6 では，IPv6 アドレス空間の内容をノード識別子と位置指示子という 2 種類の空間に分離させ，ノード識別子と IP アドレスの対応を保持する位置管理装置を設けることにより，IP アドレスの変化を上位ソフトウェアから隠蔽する．しかし，LIN6 ではアドレス空間のビット数が半分になることからアドレス利用効率が大きく低下する上，独自のアドレス体系をグローバルユニークに割り当てる必要がある．また，IPv4 ではアドレス空間が不足するため適用できない．

MAT は LIN6 の課題を解決するもので，アドレス空間を分割することはせず，ノード識別子と位置指示子に対応する IP アドレスを別途定義して両者を変換する．この方式では通常の IP アドレス体系を適用することができ，IPv4 でも同様の考えを適用できる．しかし，独自の位置管理装置が必要になる点は変わっていない．また，MAT 非対応のノードは通信開始時に MN がホームネットワーク上にいないと MN の位置指示子となる IP アドレスを知ることができず，通信を開始することができないという課題がある．

この他，Shim6 [99–101] や HIP（Host Identity Protocol）[102–109] が IETF で検討，提案されている．これらの技術はノードの識別を行う Identifier とノードの位置を示す Locator を分割する方式であり，現在研究段階の技術であり，技術の普及が課題となっている．

Mobile IPv6 [16] は，Mobile IP を IPv6 用に拡張したもので，MN が移動後に経路最適化と呼ぶ機能が標準で追加され，冗長な経路を通らない通信が可能となった．しかし，通信開始時には HA を経由しなければならないため，HA が必須となることに変わりない．また，経路最適化時にはヘッダのオーバヘッドが常時発生する．

今後のユビキタス社会を想定するとネットワークを最大限に活かせる P2P 通信の要求がますます増加すると考えられる．ここで P2P 通信とは VoIP（Voice over IP）のようなリアルタイム性を必要とするアプリケーションが想定されるべきである．例えば，音声の遅延に関するガイドラインが規定されている ITU-T 勧告 G.114 では，通話遅延が 150 msec 以下であれば十分な通話が可能であるとされている [110]．この遅延には IP 電話機器で発生する遅延と，ネットワークにおける

伝播遅延を考慮する必要がある[1]．プロキシ方式のように，一般通信において特殊な装置を経由する方式では伝播遅延は増加するため，P2P 通信の特徴である柔軟性やリアルタイム性が失われる懸念がある．また，エンドツーエンド方式でも特殊な位置管理装置を必要とする方式は，十分な普及に至るまでその機能が発揮できないうえ，サーバの二重化などの対策が必須であり管理負荷が大きい．

P2P 通信が個人間の通信が主体となることを踏まえると，エンドツーエンド方式でかつ，特殊な位置管理装置を必要せずに移動透過な通信を実現できることが望まれる．また，実装レイヤについては，TCP/UDP の区別なく利用可能なネットワーク層での実現方法が有利と考えられる．さらに，現状のネットワークは IPv4 が主体であることから，IPv4 での実装が可能であることが望ましい．

本章では，エンドノードの IP 層にアドレス変換処理機能を導入し，エンドツーエンド方式をネットワーク層で実現する Mobile PPC（Mobile Peer-to-Peer Communication protocol） [45] を提案する．本提案では移動ノード到達性と通信継続性を明確に分離する．移動ノード到達性には既存の Dynamic DNS（DDNS） [111] を適用し，通信継続性に対して Mobile PPC を適用する．

Mobile PPC では，MN の IP アドレスが変化した際，MN から CN に対して変化情報を直接報告し，両ノードの IP 層の中にアドレス変換テーブル CIT（Connection ID Table）を生成する．以後の通信パケットは上記 CIT に基づきアドレス変換する．この方式により，IP アドレスの変化は上位ソフトウェアから隠蔽することができ，通信継続性を容易に実現することができる．Mobile PPC は IPv4/IPv6 の両者に適用可能であり，かつ既存システムとの上位互換性を有していることから，段階的な普及が期待できる．

Mobile PPC を FreeBSD の IPv4 上に実装し，動作確認と性能測定を実施した結果，Mobile PPC はスループットの低下がほとんどない通信継続性を実現できることを確認した．

以下，4.2 節で従来技術の例として Mobile IP，LIN6，MAT について記述し，4.3 節で Mobile PPC の原理と詳細について記述する．4.4.1 節で Mobile PPC の実装，4.5 節で性能測定結果とセキュリティ及び既存システムとの互換性について考察する．最後に 4.6 節でまとめる．

4.2 既存技術

従来技術として，プロキシ方式の代表 Mobile IP，エンドツーエンド方式の代表 LIN6，MAT，Shim6 及び HIP をとりあげる．いずれもネットワーク層による実現方法であり，トランスポート層より上位のソフトウェアに一切影響を与えないという利点がある．

[1]送信側 IP 電話機器における遅延には，アナログ信号を音声コーデックによりデジタル化する際に発生する圧縮遅延（〜15 msec）と，圧縮した音声データを IP パケット化する際に発生するパケット化遅延（〜20 msec）がある．一方，受信側 IP 電話機器では，揺らぎ吸収遅延（〜40 msec），逆パケット化遅延（〜20 msec），伸張遅延（〜5 msec）が発生する．従って，IP 電話機器において発生する遅延は機器性能により変化するが，約 100 msec と見積もることができる．

4.2.1 Mobile IP

図 4.1 に Mobile IP の通信を示す．Mobile IP では MN の識別を行う Identifier とノードの位置を示す Locator を分離し，それらの対応を管理する HA をネットワークに設置する．MN は Identifier として移動によって変化しないユニークなホームアドレス HoA（Home Address）を保持し，移動先ネットワークで割り当てられる気付けアドレス CoA（Care-of Address）を Locator として用いる．HA は MN の HoA と CoA の対応付けを行い，HoA 宛のパケットを代理受信し，CoA 宛に転送する役割を持つ．

Mobile IP の動作は，HA への登録とデータ通信に分けることができる．MN は別のネットワークへ移動した場合，移動先ネットワークで新しく取得した CoA を HA へ登録する．HA は MN の HoA と CoA の対応付けを更新する．CN は MN へ通信パケットを送信する場合，宛先を MN の HoA とする．HA はこのパケットを代理受信し，CoA 宛の IP ヘッダでカプセル化して MN に転送する．MN から CN への通信パケットは CN 宛に直接送信される．このとき送信元アドレスは HoA とする．

Mobile IP は，このように HA という特殊な装置を導入し，CN が常にホームネットワークにいる MN と通信しているように見せかけることにより移動透過性を実現する．MN 宛のパケットは必ず HA を経由するため，通信経路が冗長な三角経路となり，HA と MN 間は IP トンネルとなる．また，MN から CN へパケットを送信する場合に，送信元アドレスとして使われる HoA は MN のインターネット上での位置を正しく表していないため，途中のルータが Ingress Filtering [112] を行っていると，送信元アドレスを偽っている不正パケットと見なして破棄する可能性がある．

Mobile IP は，クライアントサーバ環境においては，CN として従来の固定サーバをそのまま利用できる点で有効である．しかし，P2P 通信が主体となる今後のネットワーク環境においては，必ずしも最適な方式とは言えない．

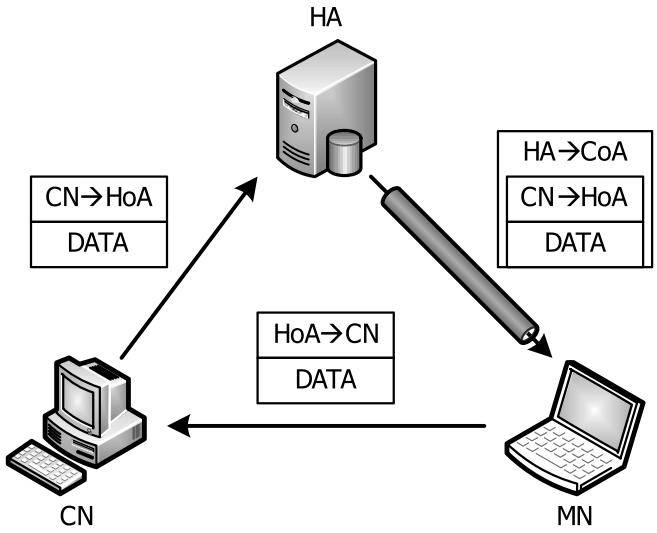

図 4.1　Mobile IP の通信

4.2.2 LIN6

LIN6 は，IP アドレスに含まれているノード識別子と位置指示子としての情報を明確に分離させ，IPv6 アドレス体系自体を見直す提案である．すなわち IPv6 アドレスの上位 64 bit を位置指示子，下位 64 bit をノード識別子として扱う．また，上位 64 bit に対し LIN6 プレフィックスと呼ばれる固定値を定義しておき，IP 層よりも上位層ではノード識別子と LIN6 プレフィックスを合わせた LIN6 汎用アドレス，下位層では位置指示子とノード識別子を合わせた LIN6 アドレスとなるように IP 層で変換を行う．上位層ではノードの位置や移動にかかわらず常に LIN6 汎用アドレスを用いる．

図 4.2 に LIN6 の通信方式を示す．LIN6 はエンドツーエンド方式であるため両ノードは対等の関係にあるが，説明のため移動する側のノードを MN，通信相手側のノードを CN と呼ぶ．MA（Mapping Agent）は MN のノード識別子と現在の位置情報との対応関係を常時保持している．CN が MN の LIN6 アドレスを知るためには，DNS からまず MA の IP アドレスを知り，MA から MN の LIN6 アドレスを取得する．MN が CN の LIN6 アドレスを知るときも同様の手順をとる．MN が CN と通信中に別のネットワークに移動した際には MA に位置指示子に変化を通知し，CN に対して MA から MN の LIN6 アドレスを再取得するように通知する．

LIN6 は，上記のように IP アドレスの役割を明確に分割したという点で評価できるが，IPv6 のアドレス構造を 2 分割するためアドレスの利用効率が大きく低下する．さらに，独自のアドレス体系を持つことになるため，ノード識別子のグローバルユニークな割り当てが必要となりその管理機構が必要になる．また，位置管理装置として MA のような特殊な装置が必要になる．IPv4 に対してはアドレス空間を分割する余裕がないため適用が困難である．

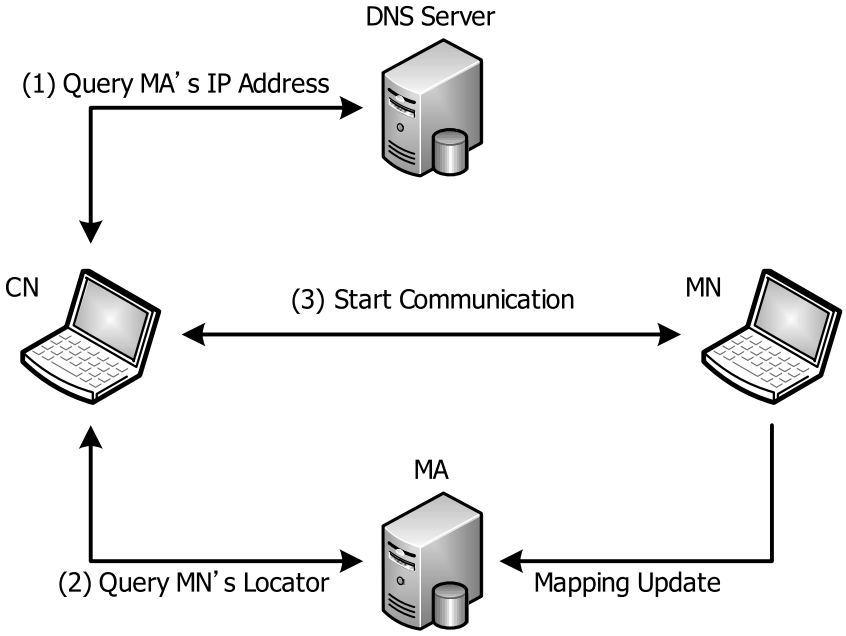

図 4.2　LIN6 の通信方式

4.2.3 MAT

MAT は，LIN6 と同様にノード識別子（ホームアドレス）と位置指示子（モバイルアドレス）を示す 2 つの IP アドレスを定義しているが，両者の対応関係（以下，マッピング情報）を保持する位置管理装置 IMS（IP Address Mapping Server）をネットワーク上に設置し，両者の間でアドレス変換を行う点が異なる．

図 4.3 に MAT の通信方式を示す．MAT も LIN6 と同様に DNS から IMS のアドレスを取得する．通信相手の IP アドレスを知る順序は，LIN6 における MA を IMS に置き換えたものと似ている．ただし，MN から CN へ通信を開始する場合には，新規に定義した IP ヘッダオプションを用いて，MN のホームアドレスを通知する．CN がパケットを返信する際には，通知されたホームアドレスを元に IMS から MN のホームアドレスとモバイルアドレスの対応を取得する．MN が CN と通信中に別のネットワークに移動した際には IMS にモバイルアドレスの更新を通知する一方，CN に対して IMS から MN のマッピング情報を再取得するように通知する．

MAT では，Identifier となるホームアドレスと Locator となるモバイルアドレスは共に通常の IP アドレス体系を使用することができ，原理的に IPv4 と IPv6 のどちらにも適応することが可能である．

このように，MAT では LIN6 の考えをもとにしているが，アドレス変換を行うことで LIN6 の課題をいくつか解決している．しかし，マッピング情報を保持する特殊な装置が必要である点は同様である．また，DNS に独自のレコードを追加するため，MAT 非対応のノードは，MN のモバイルアドレスを知ることができない．そのため MN がホームネットワーク上にいないと通信を開始できず，移動ノード到達性に制約がある．

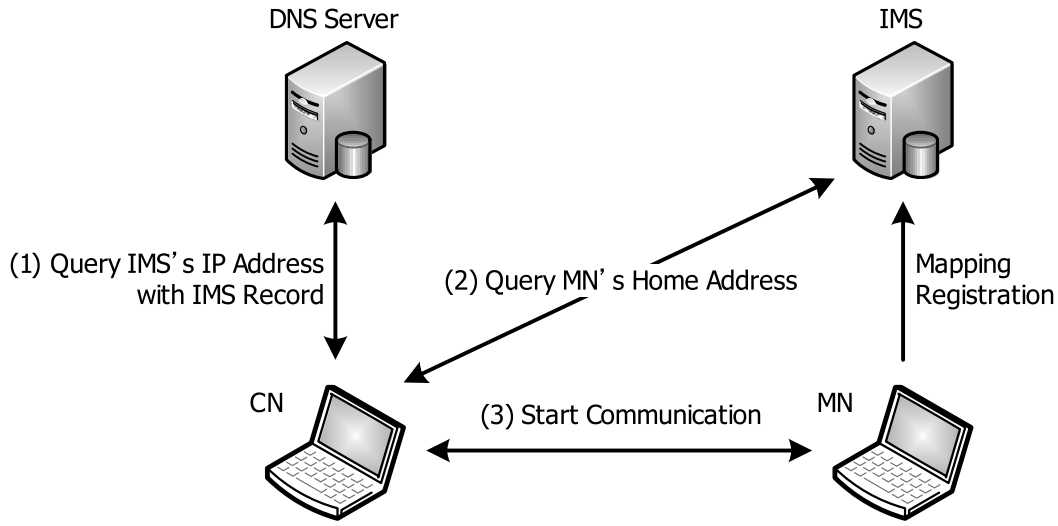

図 4.3　MAT の通信方式

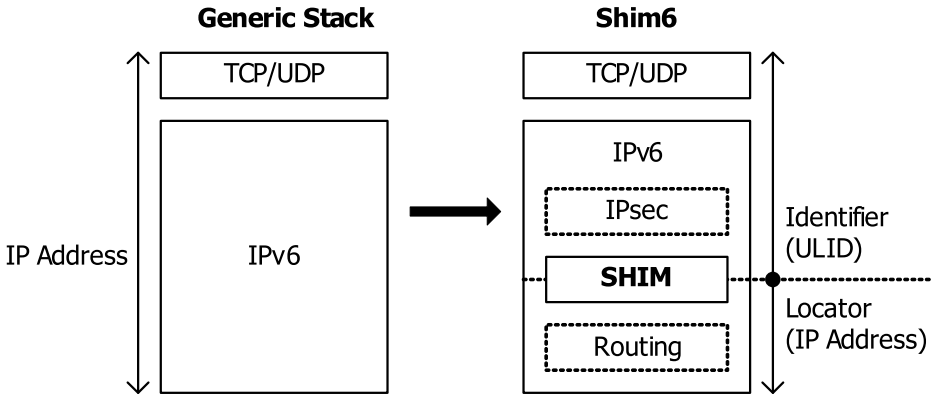

図 4.4　Shim6 プロトコルスタック

4.2.4　Shim6

Shim6 はルーチングスケーラビリティの問題を解決することを目的として仕様策定が行われており，IPv6 において PI（Provider Independent）アドレス[2]を導入せずにマルチホームを実現するためのプロトコルである．

図 4.4 に Shim6 のプロトコルスタックを示す．SHIM6 レイヤは IP 層のルーチング処理部の上位に実装され，ネットワーク層を上位に向けた自ノード宛処理部と下位に向けた中継処理部に分割する．下位は接続先プロバイダから割り当てられた IPv6 アドレスを Locator とし，そのうちいずれかをノード識別子 ULID（Upper Layer Identifier）に設定し Identifier として用いる．通信パケット送信時は SHIM6 レイヤにおいて送信元 IP アドレスを ULID から Locator アドレスに変換されて，ネットワークへ送出される．Shim6 は Mobile IP とは異なりネットワークインフラに依存せず移動透過性を実現できる．

しかし，多くの ISP で実施されているトラヒックエンジニアリング[3]が困難であったり，サーバへの負荷が過剰になるなどの課題があり，普及が容易でないと見込まれている．

4.2.5　HIP

HIP はマルチホーミングによるモビリティと IPsec [12] によるセキュリティを確保が可能なアーキテクチャである．図 4.5 に HIP のプロトコルスタックを示す．トランスポート層とネットワーク層の間に実装され，Identifier として HI（Host Identifier）と呼ぶ識別子を，Locator として IP アドレスを用いる．アプリケーションは HI によりノードを識別し，通信を行う際は HI を 128 bit にハッシュした HIT（Host Identity Tag）と呼ばれる値を用いる．HIT は IPv6 と同じアドレス形式をとり，宛先アドレスを指定するために利用される．パケット送信時は Host Identity 層において，HI から実際の IP アドレスに変換されネットワークへ送出される．

[2]ISP（Internet Service Provider）に割り振られている大きな連続した IP アドレス帯から独立した IP アドレス．
[3]トラヒック量のバランスをとり，ある特定の通信回線にだけ負荷がかからないようにする技術．

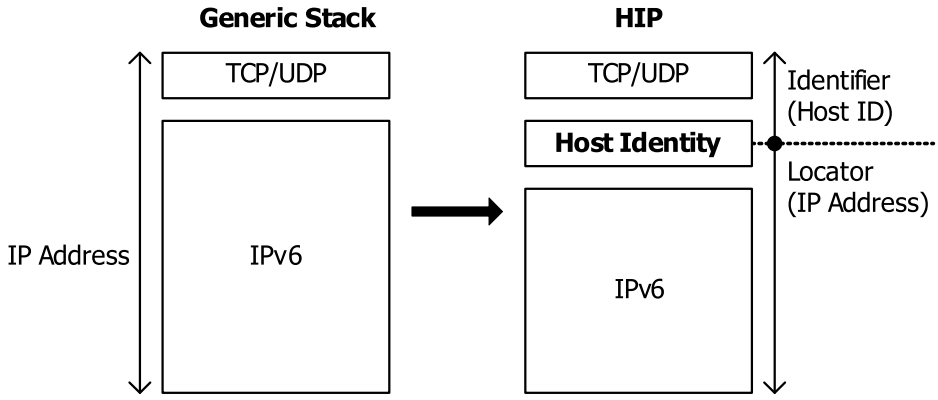

図 4.5　HIP プロトコルスタック

HIP における IPsec では ESP（Encapsulated Security Payload） [66] により暗号化される．HI は公開鍵となっており，通信開始時のセッション確立時に通信相手と HI を交換し，公開鍵を取得する．この公開鍵を用いて認証付き Diffie-Hellman 鍵交換を行い，通信パケットの暗号化処理のための共通鍵を共有する [104].

HIP は HI を特殊な DNS サーバで管理するため，新たな第 3 の装置を設置する必要がある．さらにグローバルに一意な HI の分配体系を維持するために，多大な管理負荷が発生することが想定される．IPsec を前提として設計されているため，スループットの低下が課題となる．また，公開鍵である HI の完全性を保証するシステムやセキュリティを考慮しなければならない.

4.3　提案方式

4.3.1　位置づけと概要

従来技術では通信継続性を実現する場合においても位置管理装置（HA，MA，IMS）に IP アドレスを照会するため，いずれも移動ノード到達性と通信継続性の機能を 1 種類の位置管理装置で実現しようと試みている．提案方式では両者の機能を明確に分離する．移動ノード到達性の実現には，ホスト名と IP アドレスの関係を動的に管理する DDNS を用いることができる．MN は初期立ち上げ時や移動時に新たな IP アドレスを取得すると必ず DDNS サーバにその情報を登録するため，移動ノード到達性を満たすことができる.

以後の説明では，通信が開始されるときには DDNS サーバから通信相手の IP アドレスを取得していることを前提とする．本章で提案する Mobile PPC は通信継続性を実現するための技術であり，LIN6，MAT と同様にエンドツーエンド方式と位置づけられる.

Mobile PPC の機能は，移動情報の通知処理と IP アドレスの変換処理に分けられる．通知処理は，MN が別のネットワークへ移動した場合，移動先ネットワークで新しく取得した IP アドレスを CN に通知する．通知処理により，MN と CN は移動前と移動後の IP アドレスの対応関係を記すテーブル CIT を更新する．CIT エントリは，通信が開始される際にコネクション単位で生成されるもので，MN が移動するたびにその内容が書き換えられる．IP アドレスの変換処理は，すべ

てのパケットに対して CIT を参照しながらアドレス変換を実行する．このような方式により，上位層に対しては移動による IP アドレスの変化が隠蔽され，上位層はアドレスの変化に気づくことなくコネクションを維持できる．

4.3.2 移動情報の通知処理

図 4.6 に Mobile PPC による通信手順を示す．MN と CN 間で新しく通信が開始されると，エンドノードは送受信パケットを元に上位層がコネクションを識別する際に用いられる通信識別子（両ノードの IP アドレスとポート番号，プロトコル番号の組）が記された CIT エントリを生成する[4]．

$$\text{MN:} \quad IP_{MN} : s \leftrightarrow IP_{CN} : d \quad [proto] \tag{4.1}$$

$$\text{CN:} \quad IP_{CN} : d \leftrightarrow IP_{MN} : s \quad [proto] \tag{4.2}$$

上記は両ノードに生成される CIT エントリの例であり，s と d は送信元/宛先ポート番号，*proto* は両ノード間で行う通信の種類に応じて TCP または UDP が記載される．CIT エントリは，移動前と移動後の通信識別子の情報から構成される．なお，通信開始時点では IP 層でのアドレス変換は実行されない．

MN が CN と通信中に別のネットワークへ移動すると，MN は移動先で DHCP [113] サーバなどから新しく IP アドレスを取得する．ここで MN の IP アドレスが IP_{MN} から $IP_{MN}{}^2$ に変化したとすると，移動前後の IP アドレスと移動前に確立していたコネクションに関する通信識別子の情報を含む CU（CIT Update）Request メッセージを生成し，CN へ送信する．CU Request は CN に対して

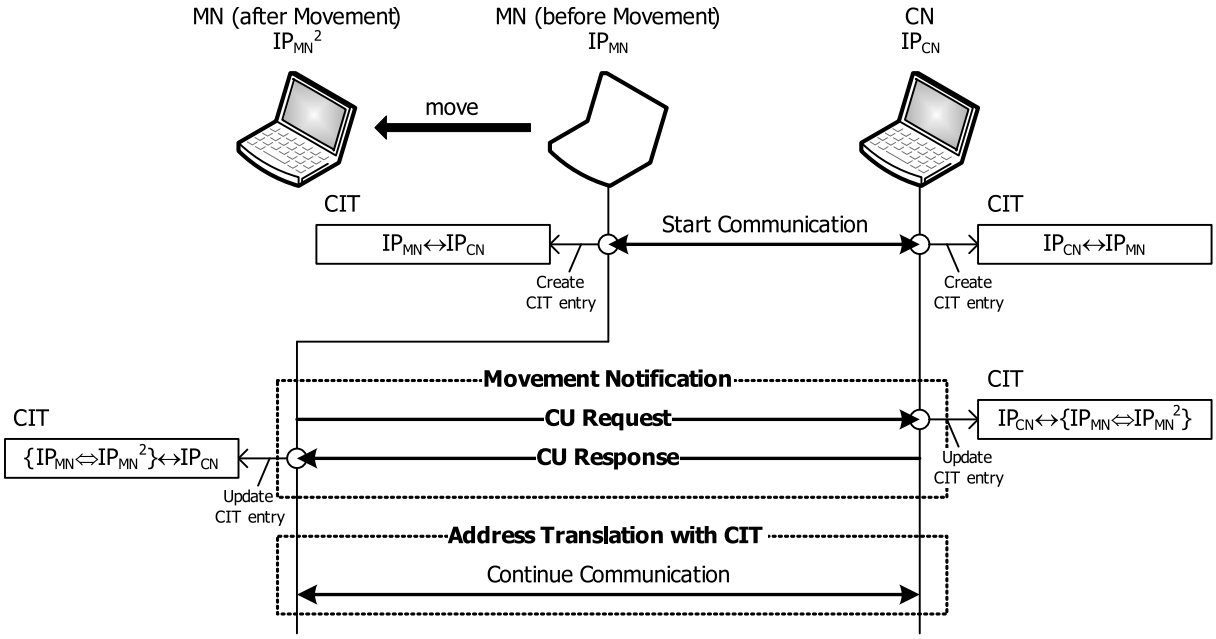

図 4.6　Mobile PPC の通信手順

[4]本章における図中の CIT では，ポート番号およびプロトコル番号は省略する．

移動を通知するとともに CIT エントリの更新を要求する．CN は通知された情報を元に式 (4.2) に示した自身の CIT エントリを更新し，CU Response メッセージを返信する．MN は，CU Response を受信後に式 (4.1) に示した自身の CIT エントリを更新する．両ノードで更新された CIT エントリを以下に示す．

$$\text{MN:} \quad \{IP_{MN} : s \overset{CIT}{\Longleftrightarrow} IP_{MN}{}^{2} : s\} \leftrightarrow IP_{CN} : d \quad [proto] \tag{4.3}$$

$$\text{CN:} \quad IP_{CN} : d \leftrightarrow \{IP_{MN} : s \overset{CIT}{\Longleftrightarrow} IP_{MN}{}^{2} : s\} \quad [proto] \tag{4.4}$$

エンドノードで更新された CIT エントリは，MN の移動前と移動後の IP アドレスの対応関係が登録され，以後の通信パケットに対する IP アドレス変換処理に用いられる．アドレス変換処理については 4.3.3 項にて述べる．

CU Request/Response は ICMP Echo をベースに定義されている．図 4.7 に CU メッセージフォーマットを示す．CU Request と CU Response は共通のフォーマットである．ヘッダ部には Mobile PPC 制御メッセージを識別するための ID，CU Request/Response の識別情報（Type，Code）などが記載される．データ部には移動前後の IP アドレス（Old IP Address/New IP Address），CN と確立していたコネクション数（Connection Count），および初期通信識別子（Initial Connection ID）がコネクション数の分だけ含まれる．初期通信識別とはコネクションを確立した際の IP アドレスとポート番号およびプロトコル番号の組であり，通信開始時に生成した CIT エントリの情報と一致する．

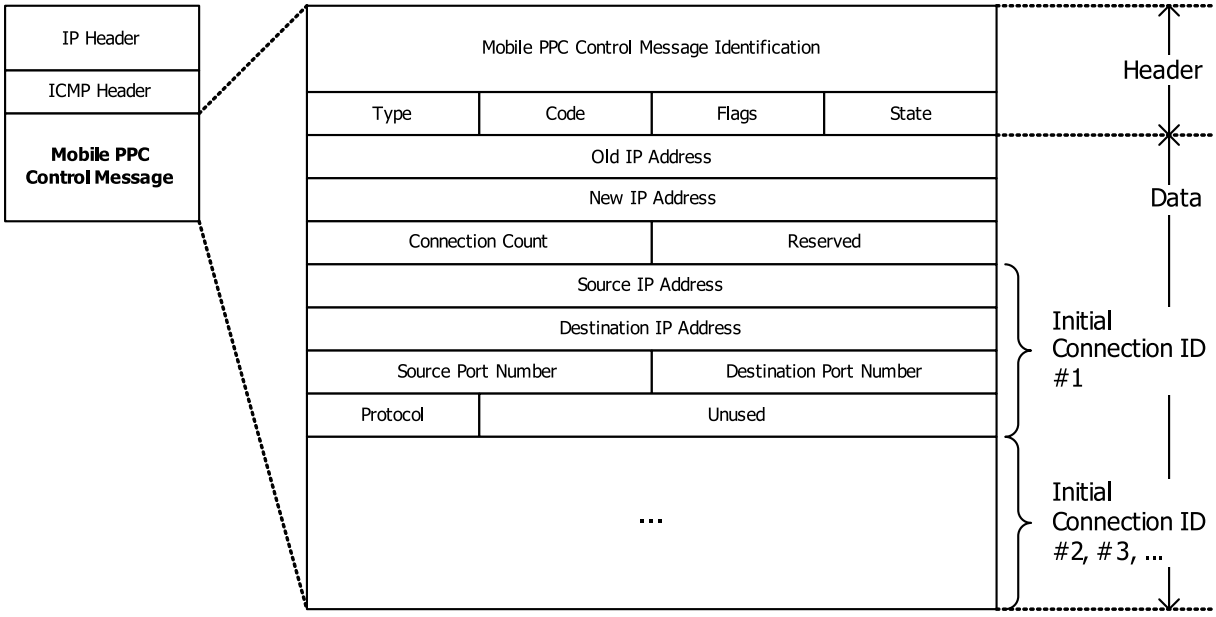

図 4.7　CU メッセージフォーマット

4.3.3 アドレス変換処理

図 4.8 に MN の IP アドレスが IP_{MN} から $IP_{MN}{}^2$ へ変化した場合のアドレス変換処理を示す．MN から送信されるパケットの送信元 IP アドレスは，IP 層で式 (4.3) の CIT エントリを参照し，MN 移動前の IP アドレス IP_{MN} から移動後の IP アドレス $IP_{MN}{}^2$ へ変換される．このパケットを受信した CN は MN と同様に式 (4.4) の CIT エントリを参照して，パケットの送信元 IP アドレスを移動後の IP アドレス $IP_{MN}{}^2$ から移動前の IP アドレス IP_{MN} へ変換を行い上位層へ渡す．CN から MN への逆方向のパケットについても，宛先 IP アドレスに対して上記と同様のアドレス変換を行う．

このように IP 層において正しくルーチングされるようにアドレス変換し，上位層に対してはその変化を隠蔽するため通信中に MN が移動してもコネクションを維持させることが可能となる．

上記は MN が 1 回移動した場合を示したが，MN は移動を複数回繰り返しても通信を継続することができる．また，移動後に通信を継続している際に，新たにコネクションを確立するようなケースにも対応している．図 4.9 に複数回移動した場合のアドレス変換の適用方法を示す．図 4.9 は MN と CN が通信中に MN が移動を繰り返し，IP アドレスが IP_{MN} から $IP_{MN}{}^2$，$IP_{MN}{}^3$ へと変化した場合を表している．アドレス変換処理は，コネクション確立時点における MN の IP アドレスがベースとなる．従って MN が 2 回移動した状態では，MN は上位層では通信 #1 の自ノード IP アドレスを IP_{MN}，通信 #2 の自ノード IP アドレスを $IP_{MN}{}^2$ として認識する．このように Mobile PPC では，コネクション毎に上位層で認識する自身の IP アドレスが異なる．移動を繰り返した場合には，通信開始時の IP アドレスと移動先で取得した IP アドレスとの間でアドレス変換を行う．

MN の IP アドレスが $IP_{MN}{}^3$ に変化した時点では，IP アドレス IP_{MN}，$IP_{MN}{}^2$ は実際に MN の NIC（Network Interface Card）に割り当てられた IP アドレスではなくなる．その結果，例えば別のノード MN2 が取得した IP アドレス IP_{MN2} が，IP_{MN} または $IP_{MN}{}^2$ と同じになる可能性がある．まれな

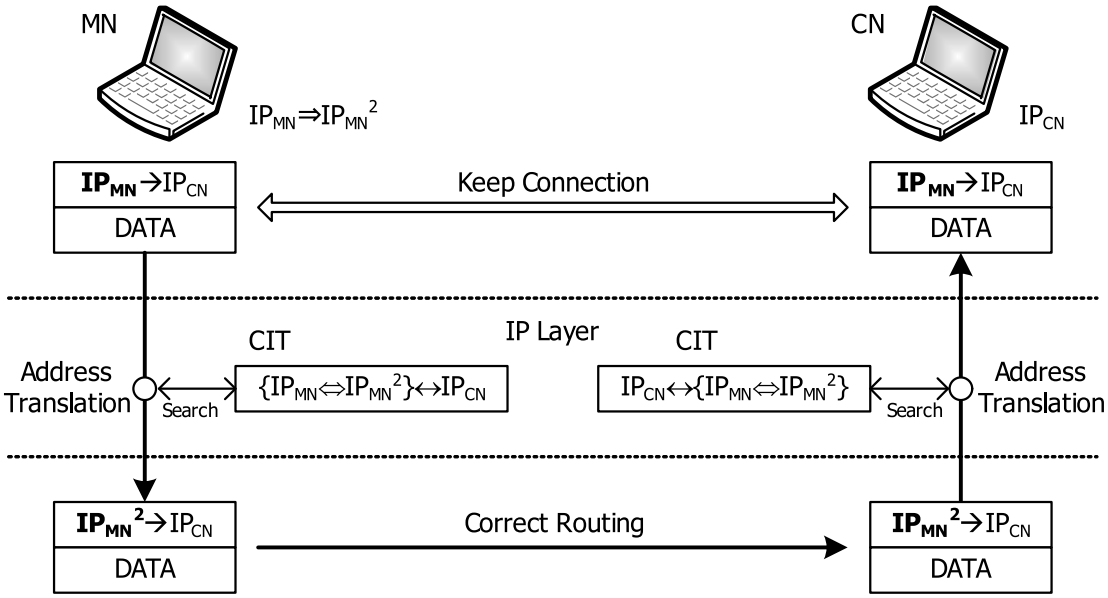

図 4.8　アドレス変換処理

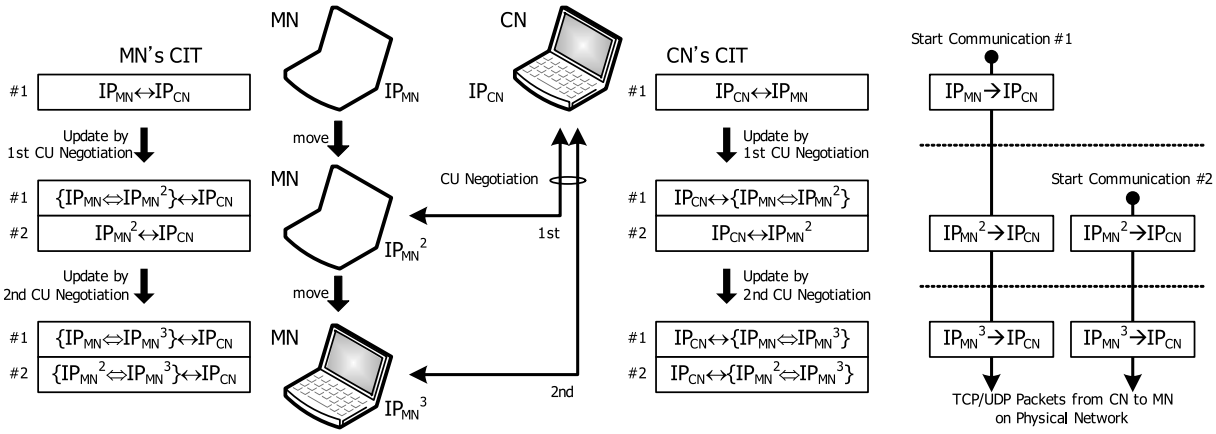

図 4.9　複数回の移動におけるアドレス変換の適用方法

ケースとして，MN1 がアドレス変換を適応している通信と，MN1 が MN2 と新しく開始する通信の上位層における通信識別子が一致する可能性が考えられる．この場合，Mobile PPC では新たな通信の通信識別子が通信中の通信識別子と一致することを検出すると，新たな通信に対してポート変換を適用する．この方法により通信識別子の一致を防止することができ，両通信を区別することができる．このような理由により，CIT エントリの変換情報として IP アドレスだけでなくポート番号も含まれている．

4.3.4　従来技術との比較

Mobile PPC と従来技術の比較を表 4.1 に整理する．従来技術の課題については 4.2 節で記述済みであるため，本節では Mobile PPC の利点を中心に記述する．

Mobile PPC は移動ノード到達性を実現するために第 3 の装置として DDNS サーバを利用するが，これは通信継続性を実現するために必要となる特有の装置ではなく，既存環境への適用が容易である．他の従来方式でも通信開始時に CN の IP アドレスを解決する際に DNS サーバは必須であり，その意味では提案方式は余分な装置を極力排除している．このため特有の装置による一点障害の課題がなく，二重化などの措置も不要で管理が容易である．

DDNS サーバは DNS サーバのデータベースを動的に更新できるように設定されたもので，DNS の延長技術である．例えば，市場シェアが最も高く，インターネットにおけるデファクトスタンダードの DNS サーバアプリケーション BIND（Berkeley Internet Name Domain）[114] では，設定ファイルを一部変更するだけで DDNS サーバとして動作させることができる．また，DDNS サービスを提供する事業者や ISP（Internet Service Provider）は数多く存在しており[5]，これら既存のサービスをそのまま利用することも可能である．

Mobile PPC はエンドツーエンド方式であるため，冗長な通信経路は発生せず，アドレス変換を行うだけなので，パケットサイズの冗長はなく高スループットが期待できる[6]．ただし，通信継続

[5]国内外のプロバイダが提供する DDNS サービスは付録 B.2.2 に示す．
[6]スループットに関する性能評価は 4.5.3 項で示す．

表 4.1　従来技術との比較

	Mobile IP	Mobile IPv6	LIN6	MAT	提案方式
特有の装置の存在	HA	HA	MA	IMS	なし
一点障害	×	△	△	△	○
通信経路の冗長	×	△	○	○	○
パケットサイズの冗長	△	△	○	○	○
CN への実装	○	○	○	△	△
移動ノード到達性	○	○	○	△	○
通信開始時のオーバヘッド	○	○	○	○	△
IPv4/IPv6 両者に対応	×	×	×	○*	○
アドレス制約	○	○	×	○	○

* IPv4 における実装は未完了

性を実現するために CN への実装が必要となる．同じエンドツーエンド方式である LIN6 においては，MA をプロキシとして拡張することにより，LIN6 を実装しない一般ノードに対する移動透過性を提供する方式 [115] が検討されている．Mobile PPC は一般ノードとの通信継続性は現時点ではサポートしていないが，一般ノードとの上位互換性があるため，移動しない限り通信は可能である．通信相手が一般ノードの場合において移動透過性を実現するためには，文献 [115] のようにプロキシサーバを利用する必要がある．

移動ノード到達性については既に動作実績のある DDNS サーバにより満たすことができる．ただし，MN への誤接続を回避するためにネームキャッシュの有効期限を短くする必要があり，従来技術と比較して通信開始時に行う名前解決処理に伴うオーバヘッドが若干大きくなると考えられる．DDNS サーバの利用に関する考察については 4.5.5 項にて述べる．

Mobile PPC は原理的に IPv4 と IPv6 の両者に適用可能である．多くの従来技術は検討対象や実装が IPv6 ベースであるが，IPv6 が普及していない現状では IPv4 における実装が可能であることは大きな利点である．また IP アドレスは従来のアドレス体系をそのまま利用できるため，アドレスの制約はない．さらにエンドノードのみに実装すればよいこと，一般ノードとの上位互換性があるため，段階的な普及が期待できる．以上の比較結果より，提案方式は今後のユビキタス社会に最も適した方式と考えられる．

4.4　実装

Mobile PPC を FreeBSD 5.2.1-RELEASE 上に実装し動作を検証した．本節では Mobile PPC のモジュール構成と CIT エントリのフォーマット詳細について記述する．

4.4.1 モジュール構成

Mobile PPC のモジュール構成を図 4.10 に示す．Mobile PPC は 2.4 節にて示した GSCIP（Grouping for Secure Communication for IP） [33] のモジュール GPACK の一部として実装される．パケット受信時には IP 入力関数である `ip_input()` から，パケット送信時には IP 出力関数である `ip_output()` から GPACK を経由して Mobile PPC モジュールにおいてアドレス変換処理を終えたら差し戻す形をとっている．これにより，既存の処理には一切変更を加えることなく機能実装を実現できる．

Mobile PPC を実現するモジュールは CIT 検索モジュール，アドレス変換モジュール，移動管理モジュールの 3 つがある．

CIT 検索モジュールは，TCP/UDP パケットを送受信する際に呼び出され，パケットの通信識別子をキーとして該当する CIT エントリが存在するか確認する．該当する CIT エントリが存在し，通信相手ノードにより更新されていたら，TCP/UDP パケットはアドレス変換モジュールによりアドレス変換処理が行われる．アドレス変換モジュールは TCP/UDP パケットの IP アドレスおよびポート番号を変換する．この変換に伴うチェックサムの更新は，NAT（Network Address Translator）[21] と同様に差分計算を行うことにより再計算する．該当する CIT エントリが存在しても，移動通知処理により更新されていなければ，TCP/UDP パケットはそのまま `ip_input()` または `ip_output()` に戻される．CIT 検索の結果，該当するエントリが存在しなかった場合，TCP/UDP パケットの通信識別子を用いて CIT エントリを新たに生成，登録する．この場合，処理中の TCP/UDP パケットはそのまま `ip_input()` または `ip_output()` へ戻される．

移動管理モジュールは，IP アドレス変更時における CU および CU REPLY による移動情報通知

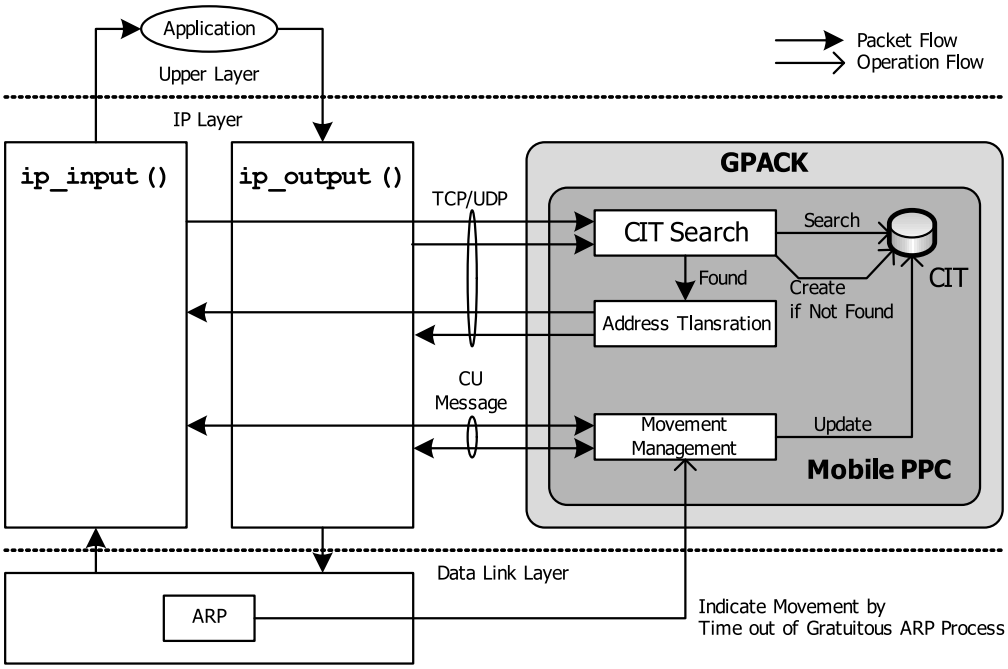

図 4.10　モジュール構成

処理を行う．MN が移動した場合，移動先ネットワークに存在する DHCP サーバにより新しい IP アドレスを取得する．DHCP サーバから IP アドレスの使用許可 DHCP ACK を受信した時点では，取得した IP アドレスがまだ確定しておらず，その後に必ず Gratuitous ARP [72] を用いた IP アドレス重複確認が行われる．そのため，移動管理モジュールは，上記 ARP 処理のタイムアウトと同時に FreeBSD の ARP 処理関数より Mobile PPC モジュールの移動管理モジュールが呼び出され，移動情報通知処理を行う

4.4.2 CIT

表 4.2 に CIT のフォーマットを示す．CIT は通信開始時の初期通信識別子（Initial Connection ID），移動後の新しい通信識別子（New Connection ID），およびエントリ状態情報（State）からなり，2048 レコードから構成されるハッシュテーブルとして実装されている．．初期通信識別子は，4.3.2 項で述べたように通信が開始された際に登録される．新通信識別子は，CU Request および CU Response による通知処理によって登録されるフィールドであり，新規に生成・登録された際は値が設定されていない．エントリ状態情報は現在の CIT エントリの状態を示しており，以下の 4 つが定義されている．

1. READY：CIT エントリ生成時の状態
2. CU WAITING：CU Request を送信し，通信相手ノードからの CU Response 待ち状態
3. ACTIVE：CIT エントリが移動通知処理により更新された状態
4. DISCARD：有効期限を超過し，CIT エントリを削除しなければいけない状態

ノードはエントリ状態情報を確認することにより，アドレス変換が必要かどうかを判断する．エントリ状態情報が READY の場合，CIT に基づくアドレス変換処理は行われず，ACTIVE の場合にアドレス変換処理が行われる．通常はポート変換を行わないが，4.3.3 項で示したように MN が移動前に使用していた IP アドレスが他のノードに割り当てられた場合，上位層で認識する初期通信識別子が常にユニークになるように，必要な場合に限りポート変換を適用する．なお，CIT エントリは 1 つのコネクションに対して送信用と受信用の 2 つのエントリが生成される．

上記の他に，CIT エントリの管理用情報としてエントリ生成時刻，最終参照時刻や有効期限などのフィールドがある．Mobile PPC モジュールにはカーネルタイマにより CIT を削除する機能を

表 4.2　CIT フォーマット

Initial Connection ID					New Connection ID					State
saddr	daddr	sport	dport	proto	saddr	daddr	sport	dport	proto	
IP_{MN}	IP_{CN}	$s1$	$d1$	TCP	$IP_{MN}{}^2$	IP_{CN}	$s1$	$d1$	TCP	ACTIVE
$IP_{MN}{}^2$	IP_{CN}	$s2$	$d2$	TCP	—	—	—	—	—	READY

⋆ saddr: Source IP Address　daddr: Destination IP Address　sport: Source Port Number
dport: Destination IP Address　proto: Protocol

実装している．CIT エントリに該当する通信が一定時間行われていないことを管理用情報から判断すると，該当する CIT エントリの状態情報が DISCARD に遷移し，タイマ処理により削除される．また，処理中のパケットが TCP の場合，TCP フラグに応じて CIT エントリを削除する場合もある．例えば，FIN ACK および RST を受信した場合は TCP コネクションをクローズするため，該当する CIT エントリの状態情報をすぐさま DISCARD に遷移させる．このような処理により，不要な CIT エントリを保持しない仕組みを実現している．

4.5 評価

Mobile PPC を試作し，両エンドノードが移動を繰り返しても通信を継続できることを確認した．本節では，試作システムの性能測定結果，および同一条件下における Mobile IP とのスループット比較を行った．

4.5.1 パケット処理時間

Pentium M 1.8 GHz の CPU を搭載し，100BASE-TX で接続された PC 上で Mobile PPC モジュールのパケット処理時間測定した．ここで Mobile PPC モジュールの処理時間とは，`ip_input()` および `ip_output()` から GPACK モジュールが呼び出され，Mobile PPC による処理が行われた後，`ip_input()` または `ip_output()` に差し戻すまでの時間である．これは Mobile PPC を実装することにより，全ての TCP/UDP パケットに対して CIT 検索が行われるため，これらにかかるオーバヘッドを調査するものである．

処理時間の測定には RDTSC（Read Time Stamp Counter） [78] を用いた．RDTSC は CPU のカウンタから周波数クロックを取得する命令で，モジュール処理に費やした時間を正確に算出することができる[7]．

表 4.3 に Mobile PPC モジュールのパケット処理時間の測定結果を示す．測定結果は FTP 通信中に流れた 1,500 byte の通信パケット 1,000 個の処理時間の平均である．測定結果は，アドレス変換を行わない場合は 0.31 μsec [8]，アドレス変換を行う場合は 0.54 μsec であった．1 パケットにか

表 4.3 Mobile PPC モジュールのパケット処理時間

	アドレス変換処理の有無	
	変換なし	変換あり
IP 層全体の処理時間	21.03	21.26
Mobile PPC モジュールの処理時間	0.31	0.54
(Mobile PPC モジュールの比率)	(1.47 %)	(2.53 %)

処理時間の単位：μsec

[7]測定に用いた機器の CPU 周波数が 1.8 GHz のため，分解能は $1/1.8\ \text{GHz} = 0.56$ nsec となる．

[8]文献 [35] において類似の処理が RDTSC により性能測定されており，約 0.27 μsec という測定結果が示されている．本章での測定値も妥当な結果であると考えられる．

かる IP 層全体の処理時間は約 21 μsec であり，Mobile PPC モジュール処理時間の占める比率はアドレス変換を行わない場合は 1.47 %，アドレス変換を行う場合は 2.53 % であった．このことから Mobile PPC を実装したことによるオーバヘッドの増加は十分小さいと言える．

4.5.2 通信断絶時間の測定

Mobile PPC の移動透過性にかかわる処理時間，すなわち通信断絶時間を図 4.11 に示す測定環境で測定した．2 つのルータ R1，R2 によりサブネットが異なる 3 つのネットワークを用意し，MN の移動先となるネットワークには DHCP サーバを設置した．MN と CN に Mobile PPC を実装し，DHCP サーバおよびクライアントには，ISC DHCP v2 パッケージ [116] を使用し，パラメータは DHCP プロトコルのデフォルト値を使用した．有線 LAN は 100BASE-TX で構成し，MN は IEEE802.11b により無線アクセスポイントに接続した．MN から CN へ連続的に FTP を用いたデータ転送を実行させておき，MN を別のネットワークに移動させ，MN 側で直接コマンド `dhclient` を実行することにより，DHCP サーバから新しく IP アドレスを取得させた．

図 4.12 に MN が異なるネットワークへ移動した際に発生するシーケンスを示す．MN が通信を再開するまでに発生する通信断絶時間は，(I) L2 ハンドオーバ時間，(II) DHCP サーバからの IP アドレス取得時間と (III) エンドノード間で行われる移動情報通知処理時間の合計として算出できる．DHCP による IP アドレス取得時間については本提案方式の主題ではないが，参考のために測定を行った．

表 4.4 に IP アドレス取得時間の測定結果を示す．この時間には MN と DHCP サーバ間の 2 往復の DHCP シーケンスと IP アドレス取得後に行われる Gratuitous ARP によるアドレス重複確認の処理が含まれる．表 4.4 に示すように約 2〜5 sec（平均 3.34 sec）の時間を要し，通信断絶時間のほとんどの割合を占める．

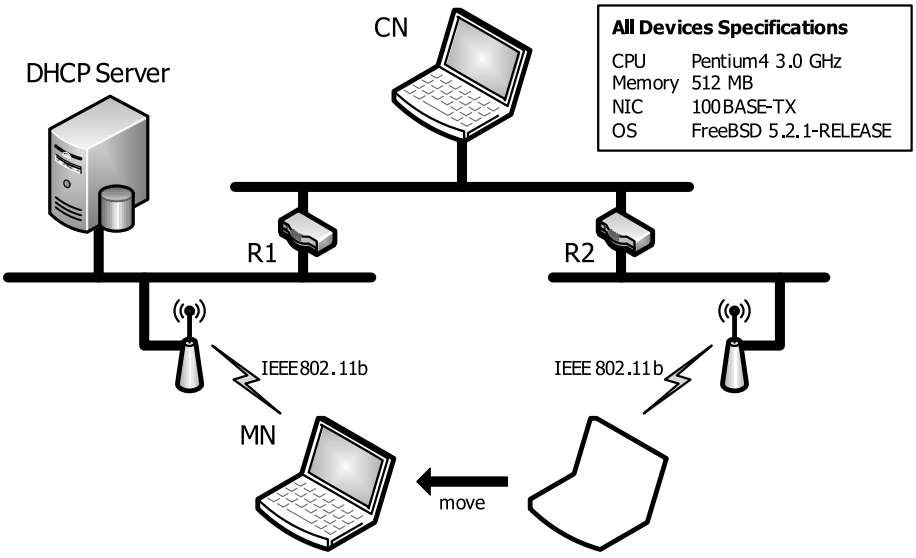

図 4.11 通信断絶時間の測定環境と機器仕様

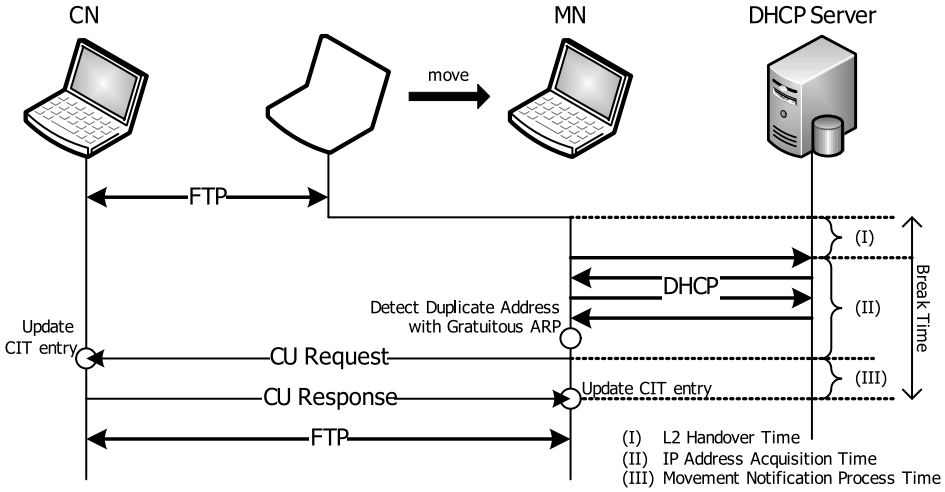

図 4.12　MN 移動時のシーケンス

表 4.4　DHCP サーバからの IP アドレス取得時間

	最大	最小	平均
アドレス取得時間	4.85	2.99	3.34

単位：sec

表 4.5　移動情報の通知処理時間

コネクション数	1	2	3	4	5
CU Request/Response 到達時間	288	258	253	267	326
MN/CN の CIT 更新時間	38	40	44	45	47
移動情報通知処理時間	326	298	293	312	373

単位：μsec

次に表 4.5 に移動情報通知処理時間の測定結果を示す．移動情報通知処理時間には，CU Request および CU Response の伝達時間，および MN と CN における CIT 更新時間が含まれる．エンドノード間で確立しているコネクション数を 1 から 5 に増やした場合，MN と CN の CIT 更新時間は 38〜47 μsec となった．また，CU Request および CU Response の到達時間は 253〜326 μsec となった．パケット到達時間に多少ゆらぎがあるのは，測定環境に無線 LAN があり，周囲の環境による影響を受けたためだと考えられる．このことから，移動情報通知処理時間はパケットの伝達時間が大半をしめており，エンドノード間のコネクション数による影響はほとんどないと言える．

図 4.13 に図 4.12 と同様の条件において MN が移動した時の TCP シーケンス番号の変化を示す．無線 LAN は周囲の条件に依存するため，評価システム内はすべて 100BASE-TX の有線 LAN とした．ネットワークの移動は MN の LAN ケーブルを移動先ネットワークにつなぎ直し，その後直ち

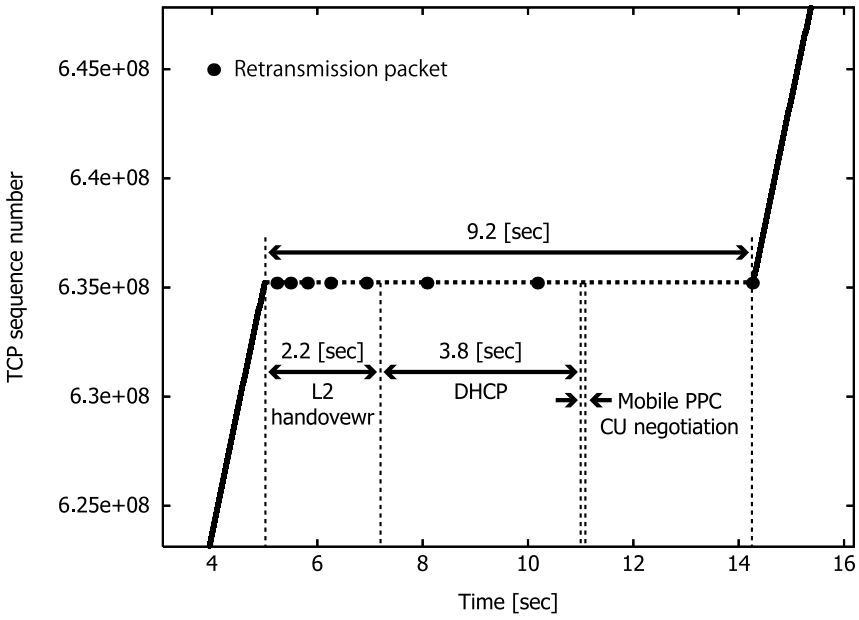

図 4.13　MN 移動時における TCP シーケンス番号の変化

に MN から IP アドレスを取得するコマンドを実行することによりエミュレートした．上記時間は MN が物理的にネットワークから切り離された状態であり，L2 ハンドオーバの時間と見なすことができる．

測定の結果，L2 ハンドオーバに約 2.2 sec，DHCP による IP アドレス取得に 3.8 sec を要した．表 4.5 からわかるように Mobile PPC による移動情報通知処理時間は合計 300 μsec 程度であり，ほとんど無視できる．このように通信を継続するために必要な処理は約 6 sec [9] で完了したが，実際の通信が再開されるまでには約 9.2 sec を要した．これは TCP 再送制御が機能したためであり，図 4.13 からわかるように，8 回目の再送パケットにより通信が再開されたためである．

文献 [98] では，IPv4 ネットワークにおいて実験的にハンドオーバ時間を測定しており，DHCP による IP アドレスの取得には約 8.6 sec，IMS へのマッピング情報の伝達時間に約 0.1 sec 要したことが示されている．実際には上記合計時間に L2 ハンドオーバ時間が含まれ，さらに TCP 再送制御により通信断絶時間が長くなることが推測される．

実際には無線 LAN の L2 ハンドオーバは 50〜400 msec で完了するため [117]，通信断絶時間を短縮するには IP アドレス取得時間を短縮することが重要であることがわかる．本件の解決策としては，DHCP クライアントアプリケーションの最適化によるシームレスハンドオーバ [118] や，アドレス重複確認の高速化 [119] などが有効と考えられる．

なお，IPv6 ではルータから広告される RA（Router Advertisement）によりネットワークの移動を検知し，IP アドレスを自動生成する機能がある．RA の送信間隔はネットワークへの負荷を考慮して，一般的に 1.5 sec 以上とされている [120, 121]．また，IP アドレスの重複確認 DAD（Duplicate Address Detection） [122] には約 1 sec 程度の時間が必要となるが，IPv4 と比較すると高速にハンドオーバ処理を完了することができる．

[9] 2.2 sec+3.8 sec+300 μsec $\fallingdotseq$ 6 sec

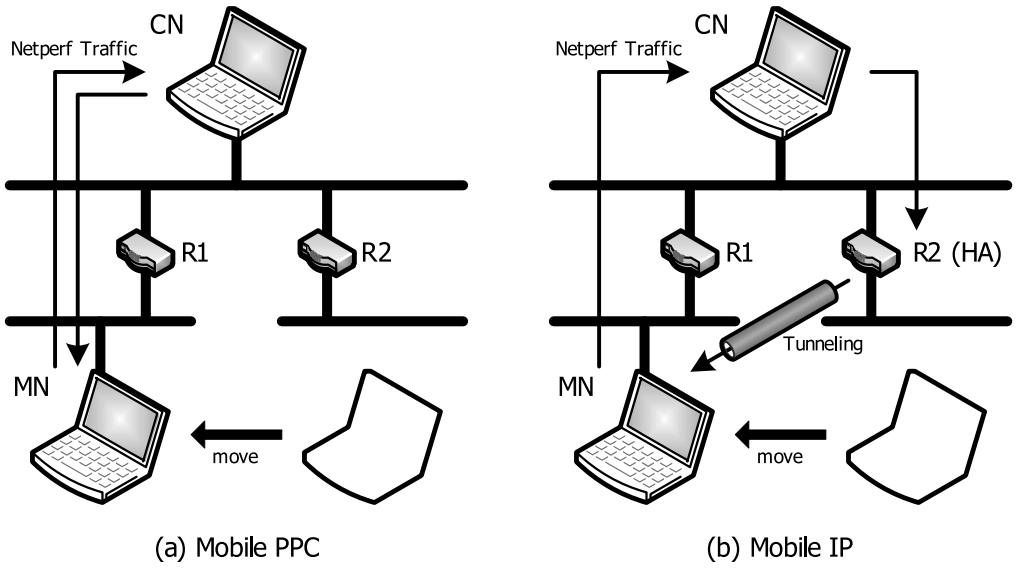

図 4.14　スループット評価システムの構成

4.5.3　Mobile IP とのスループット比較

Mobile PPC と Mobile IP のスループット比較を行うために図 4.14 のような評価システムを構築した．Mobile PPC 環境（図 4.14 (a)）では MN，CN に Mobile PPC を実装し，Mobile IP 環境（図 4.14 (b)）では，MN に Mobile IPv4 を実装し，R2 に HA の機能を実装した．Mobile IPv4 には PSU Mobile-IP パッケージ [123] を使用し，動作モードは Mobile PPC との比較をしやすくするため FA が不要な co-located care-of address モードとした．使用した装置は図 4.11 に示したものと同一で，ネットワークの移動は LAN ケーブルをつなぎ直すことでエミュレートした．図 4.14 中の矢印は，MN と CN 間のトラヒックの方向を示しており，Mobile IP では HA として動作する R2 から MN の間はトンネリング転送される．

上記環境下で，以下のケースにおける MN-CN 間のスループットを測定した．

Case 1： Mobile PPC を実装しない状態
Case 2： MN と CN が Mobile PPC を実装しているがアドレス変換をする前（移動前）
Case 3： Mobile PPC を実装し，かつアドレス変換をしている時（移動後）
Case 4： MN が Mobile IP を実装し HA 配下にいる時（移動前）
Case 5： Mobile IP を実装して IP トンネリング通信をしている時（移動後）

表 4.6 に測定結果を示す．スループットの測定には, ネットワークベンチマークソフト Netperf [88] を使用し，20 回の平均値とした．表 4.6 よりわかるように Mobile PPC では何も実装していない状態（Case 1）に比べ，移動前（Case 2）と移動後（Case 3）ともにスループットの低下はほとんど見られなかった．

Mobile IP は移動前（Case 4）ではスループットの低下はほんどないが，移動後（Case 5）では，IP カプセル化によるオーバヘッドと通信経路の冗長により，スループットが 8.6 % ほど低下した．図 4.14 (b) の測定構成では，HA と MN が 1 ホップ分離れている構成であるが，一般的なネット

表 4.6　スループットの比較

状態		スループット	低下率 (%)
General	Case 1	93.237	—
Mobile PPC	Case 2	93.236	0.001
	Case 3	93.193	0.047
Mobile IP	Case 4	93.231	0.006
	Case 5	85.202	8.618

スループット単位：Mbit/s

ワーク構成では HA を経由することによりラウンドトリップ遅延がさらに増加する可能性があり，スループットがさらに低下することが予想される．

4.5.4　セキュリティの考察

エンドツーエンド方式で移動透過性を実現する場合，通信継続の際には悪意あるユーザによるなりすましを防止するため，ノード間における確実な認証が必要である．グローバルな環境では通信相手は不特定多数となるため，事前に認証に必要な共有鍵や証明書を共有することは難しい．Mobile IPv6 では MN と HA が共有鍵を事前に保持していることを前提とし，CN が HA と MN に宛てて送信した両方のデータを MN が正しく受信することによって共有鍵を生成する仕組み（Return Routability）が提案されている [16]．このような方式は特定の装置を使用しない Mobile PPC には適していない．

そこで Mobile PPC では認証機能を実現するために，通信開始時に Diffie-Hellman 鍵交換 [81,82] を利用した共有鍵の生成，および移動時の成りすましを防止するための認証機構 [124] を適用する．Mobile PPC における認証機構は付録 E.1 に詳細を示すが，その概要は以下の通りである．通信開始に先立ち，MN と CN の間で 2 往復の鍵共有ネゴシエーションを行う．DH 鍵交換は動作が重いため，1 往復目にクッキー交換を行い，認証処理に関わる DoS 攻撃を防止する．2 往復目に DH 鍵交換を行い，共通鍵である認証鍵を共有する．この認証鍵を使用して，移動情報の通知時に交換する CU Request および CU Response に署名を付加することにより，悪意のある第三者によるなりすましを防止する．

この認証機構は DH 演算を行うため，その処理負荷が通信開始時に発生するオーバヘッドに影響を与える可能性が考えられるが，実装を工夫することにより解決することができる．認証鍵共有に関わる処理時間については，6.5.2 項にて示す．

Mobile PPC で暗号化通信を行う場合，IPsec を適用することが可能である．IPsec では TCP/UDP チェックサムフィールドが暗号化，完全性保証の範囲に含まれているため，IP アドレスが変換されると偽造パケットと見なされ破棄されてしまう問題がある．しかし，Mobile PPC のアドレス変換処理は IP 層の上位部分で実行されており，IPsec 処理が Mobile PPC 処理に影響を受けることはない．ただし，IPsec 適用時には通信継続時に新たな IP アドレスに対する SA（Security Association）

を再生成する必要があり，IKE（Internet Key Exchange） [65] が実行されるため通信断絶時間が増加すると考えられる．2.5.1 項で示した性能評価の結果によると，IKE のネゴシエーションには約 1 sec を要しており，Mobile PPC による移動情報通知処理の前に実行されることになる．

なお，Mobile PPC はもともと GSCIP アーキテクチャの枠組みの中で移動透過性を実現する手段として提案したものである．1.4 節で示したように GSCIP においては，あらかじめ同一のセキュア通信グループに対して同一のグループ暗号鍵を割り当てる．このような環境下において通信相手のノードが同一のセキュア通信グループのメンバであった場合，通信開始時に DH 鍵交換を行う必要はなく，移動時にグループ暗号鍵を用いた認証を容易に実現することができる．また GSCIP では暗号化通信に 3 章で提案した PCCOM（Practical Cipher Communication） [35] を適用することを想定している．PCCOM ではパケット長を変えないまま認証を含む暗号化通信が可能であり，高スループットを維持することが可能である．また PCCOM の暗号化処理に必要な動作処理情報は，2 章で提案した動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol） [34] によって通信開始時に高速に生成される．

4.5.5 既存環境への対応

Mobile PPC は上位層に対してアドレスの変化を隠蔽するため，上位層で IP アドレスを直接意識する SIP（Session Initiation Protocol） [125] や FTP のようなプロトコルでも問題なく動作する．また，Mobile PPC は IP 層にモジュールを実装しているが，上位層とのインタフェースには一切影響を与えておらず，SCTP（Stream Control Transmission Protocol） [126] や DCCP（Datagram Congestion Control Protocol） [127] など IP 層の上位で新規に定義されたプロトコルに対しても適用することができる．

Mobile PPC 適用システムにおいて，MN が NAT 配下に移動することも想定される．Mobile IP では MN と HA の間に UDP トンネリングを形成することにより，NAT 越えに対応した移動透過性を実現している [42]．Mobile PPC も同様の原理で対応することができる．Mobile PPC では MN と CN の間にトンネルを形成する．送信側はアドレス変換処理後のパケットを UDP でカプセル化して通信相手へ送信することにより，NAT 変換処理の影響を受けることなく，Mobile PPC のアドレス変換処理を実行することができる．ただし，この手法はカプセル化処理を行うためスループットが低下するという課題がある．

そこで筆者らは文献 [128] において，カプセル化処理ではなく NAT 越え技術の一つである Hole Punching [22, 23] を用いた手法を提案している．提案手法はプライベートネットワークに存在する MN が CN に対して Hole Punching を行うことにより，通信経路上の NAT にマッピングテーブルを生成する．MN は移動後に送信する CU Request に，移動前の IP アドレスおよびポート番号として NAT でマッピングされた外部アドレス[10]を通知して CIT エントリを更新する．この手法によるとカプセル化処理は不要で，アドレス変換処理だけで正しく通信識別子の関係を維持することができる．上記手法については付録 E.2 にて詳細を示す．

[10] NAT のグローバル IP アドレスと動的に割り当てられたポート番号.

本提案では移動ノード到達性を実現するために既存の DDNS サーバを利用しているが，MN への誤接続を回避するためにネームキャッシュの有効期限を短く設定する必要がある．このため，MN の IP アドレス取得時に発生するオーバヘッドの多くはネームキャッシュがない場合の問い合わせ時間となる．文献 [98] によると，DNS への問い合わせ時間の平均はキャッシュがある場合で約 15 msec，キャッシュがない場合で約 350 msec である．この通信開始時のオーバヘッドが実用上の問題となるか今後検討を行う必要がある．また，DNS へ登録されたリソースレコードは明示的に削除しない限り残り続けることになるため，リソースレコードに登録した IP アドレスが別のノードに再利用された場合にも誤接続の可能性がある．これはノードがオフラインになった場合などに発生すると考えられる．この解決策として，前述したネームキャッシュの有効期限に関わるリソースレコードの TTL（Time To Live）を小さくする方法や，IP アドレスを割り当てる DHCP サーバと DDNS サーバが協調してネットワークから離脱したノードのレコードを削除するなどの方法が考えられる．

一般に無線環境でネットワークの移動を行った場合，無線レイヤと IP 層が独立してハンドオーバを実行するためパケットロスが避けられない．また，Mobile PPC のように両エンドノードが共に移動可能な方式の場合，両エンドノードが全く同時に移動した場合に移動情報通知メッセージが通信相手に到達せず，MN は互いに移動したことを知ることができないという可能性が考えられる．これは一般に “Double Jump Problem” [129] として知られている．これらの課題はエンドツーエンド方式共通の課題である．この解決策としてグローバルネットワーク上に Rendezvous サーバを設置し，このサーバを中継することにより解決する方法がある [130–133]．しかし，この手法では第 3 の装置を導入する必要があり，Mobile PPC の利点を生かすことができない．従って，MN が一時的に新旧 2 つの IP アドレスを保持させたり，無線レイヤと Mobile PPC が連携するなどの工夫が必要になると考えられる．

本研究ではノードに無線 LAN カードを 2 枚搭載し，移動前の IP アドレス宛のパケットも一定の時間内であれば受信できるようにする試みを行っており，パケットロスレスハンドオーバを実現する方式を文献 [134] で提案している．提案方式によると，通信に与える影響は十分に小さく，パケットロスがほとんど発生せずにハンドオーバできる．ハンドオーバに関する提案方式については，付録 E.3 を参照されたい．

エンドツーエンド方式を採用する移動透過性技術に共通する課題として，通信相手ノードが移動透過性を実現する機能を実装していなければ通信を継続できないことが挙げられる．WWW サーバなどインターネット上の一般ノードが通信相手であっても移動透過性を実現できる仕組みがあることが望ましい．本研究では Mobile PPC 機能を実装したプロキシサーバを導入することを検討している [135, 136]．移動後のアドレス変換処理をプロキシサーバに代行させることにより通信相手が一般ノードであっても移動透過性を保証することができる[11]．なお，プロキシサーバはあくまでもオプションとしての位置づけであり，通信相手が Mobile PPC に対応している場合はエンドツーエンドで通信を行う．

[11] プロキシサーバを利用した Mobile PPC については，付録 E.8 に示す

4.6 結論

本章では，エンドノード間で移動透過性を実現する Mobile PPC について提案した．Mobile PPC は，エンドノードの IP 層にアドレス変換処理機能を導入する．MN の IP アドレスが変化した時，MN から CN に IP アドレスの変化情報を通知し，アドレス変換テーブル CIT を更新する．移動後の通信パケットは上記 CIT に基づいて IP 層でアドレス変換される．この方式により，IP アドレスの変化は上位ソフトウェアから隠蔽される．IP アドレスの変換処理は IP 層で行われるため，上位ソフトウェアを変更する必要がなく，IP アドレスを単に変換する方式であるためパケット長が変化することがない．特殊な位置管理サーバが不要であり，既存システムとの上位互換性を有することから，従来技術の方式に比べ段階的な普及が期待できる．

IPv4 において Mobile PPC を FreeBSD 上に実装し，動作の確認と性能測定を行った．その結果，IP アドレスの変換処理による性能の低下がほとんどないことがわかった．さらに，Mobile IP とスループットの比較を行った結果，Mobile PPC の処理が通信に与える影響は Mobile IP に比べて小さいことを示した．ノードが移動した際に発生する Mobile PPC 自体のオーバヘッドは少ないものの，IP アドレス取得によるオーバヘッドを減らす工夫が別途必要であることがわかった．

第5章　NAT越えプロトコルNAT-f

5.1　研究の背景と目的

IPv4におけるグローバルアドレスの枯渇問題を解決するため，企業や家庭等のネットワークに対してプライベートアドレスを導入し，インターネットとの接点にアドレス変換装置NAT（Network Address Translator） [21] を設置する形態が一般となっている．従来のインターネットの利用形態はWWWの閲覧やメールの利用など，サーバ/クライアントモデルに基づいたシステムであり，一般にグローバルアドレス空間に設置されたサーバに対して，プライベートアドレス空間に存在するノード側から通信を開始していた．そのため，いわゆるNAT越え問題が表面化するようなことはなかった．

NAT越え問題とはNATを介すると，グローバルアドレス空間側からプライベートアドレス空間側へ通信を開始できないという問題である．しかし，近年では計算機の高性能，小型化や高速ネットワークインフラの普及に伴い，IP電話やマルチメディア通信など個人間のエンドツーエンド通信が増加してきた．このような利用形態では，グローバルアドレス空間側からプライベートアドレス空間へ向けて通信を開始することが十分に想定されるため，NAT越え技術の必要性が高まってきた．

NATのマッピング処理は，原理的にプライベートアドレス空間からグローバルアドレス空間へのアクセス時にのみ実行される．また，そもそもグローバルアドレス空間側からプライベートアドレス空間内のIPアドレスは見えないため，プライベートアドレス空間内のノードを指定することができない．この制約を緩和するために，NATのマッピングを予め静的に設定しておくポートフォワーディング機能があるが，ポート番号1つに対して1台のノードしか設定できない上，動的に変更できないため汎用性に欠ける．

一般に企業ネットワークでは堅固なファイアウォールが設置されており，外部から組織内のサーバにアクセスするような通信を遮断していたため，NATの制約が表面化することはなかった．しかしホームネットワークでは企業のような堅固なファイアウォールは必要ではなく，外出先からホームネットワーク内のノードに自由にアクセスしたいという要求が十分に考えられる．NATが不要となるIPv6技術は現在のところ，ホームネットワークへの導入はほとんど進んでおらず，導入が始まったとしてもIPv4/IPv6の混在環境が当分続くことが想定される．今後の利用形態の多様化を考慮すれば，NATの制約を除去することは有益である．

ここで本章におけるNATとはポート番号の変換も行うNAPT（Network Address Port Translator）を含むものとする．また，NAT配下のノードを内部ノード，NAT外部のノードを外部ノードと呼称する．

IPv4 ネットワークにおいてエンドツーエンド通信を実現するために必須となる NAT 越え技術はこれまで様々な検討がなされているが，大別するとアプリケーションレベルの解決手法とネットワークレベルの解決手法に分類できる．

アプリケーションレベルの解決手法とは，ホームネットワークを形成する NAT をそのまま利用することを想定したものが多い．エンドノードが使用するアプリケーションに専用の機能を実装し，両ノードが共にアクセス可能な位置に専用のサーバを設置する．内部ノード側のアクションにより，NAT ではアドレス/ポート変換のマッピングが行われ，専用サーバへマッピングアドレス[1]が通知される．外部ノードは内部ノードと通信する場合，専用サーバからマッピングアドレスを取得し，ここに対してパケットを送信することにより，NAT 越え通信を実現する．この方式はアプリケーションにその仕組みを実装する必要があり，内部ノードがマッピングアドレスを専用サーバに通知しなければ，外部ノードは内部ノードに対して通信を開始することができない．

一方，ネットワークレベルの解決手法とは，NAT に独自の機能を実装することによりアプリケーションに依存しない汎用性を提供できる．NAT の他に，エンドノードや専用のサーバにも機能を実装する必要がある．この手法は NAT のマッピング機能を独自の処理に置き換え，内部ノードへ転送することにより，NAT 越え通信を実現する．またアプリケーションレベルの解決手法のように，内部ノード側は予めアクションを起こす必要はなく，外部ノードは内部ノードへ自由に通信を開始することができる．しかし，通信遅延の増加やスループットが低下したり，解決手法に特化した専用機器が必要になるなどの課題がある．

また，両解決手法の共通の課題として，専用のサーバが必須となる．専用のサーバが正常に動作していないと，外部ノードは内部ノードに対して通信を開始することができない．そのため，専用サーバの冗長化などによるシステムの安定性向上や耐障害性が求められる．

本章ではアプリケーションに依存しないネットワークレベルの解決手法に着目し，かつ専用サーバを利用しない手法として外部動的マッピング方式を提案する [46]．また，この方式を実現するためのプロトコルとして，NAT-f（NAT-free protocol）を定義する．提案方式は外部ノードが通信開始に先立ち，NAT と NAT-f によるネゴシエーションを行うことにより，NAT にマッピング処理を実行させる．外部ノードは NAT から直接マッピングアドレスを取得するため，専用のサーバは必要ない．外部ノードは IP 層において，送信パケットの宛先をマッピングアドレスと一致するようにアドレス/ポート変換を行う．NAT はエンドノード間の通信に対して，通常のアドレス変換処理のみを行うため，既存のネットワークレベルの解決手法の課題であった通信遅延の増加や，スループットの低下を解決できる．

NAT-f を FreeBSD に実装し，動作検証および性能測定を行った．エンドノード間の初期遅延およびスループットを評価した結果，実用上問題ない性能を有することを確認した．

以降，5.2 節で既存の NAT 越え技術を分類し，その概要と課題について整理する．5.3 節で外部動的マッピング方式を提案する．5.4 節では NAT-f の実装について述べ，5.5 節で提案方式の動作検証と性能評価の結果を示す．最後に 5.6 節でまとめる．

[1]NAT 外側のグローバル IP アドレスとマッピング時に動的に割り当てられたポート番号の組.

5.2 既存技術

本節では既存の NAT 越え技術を実現方式，および実装の観点から分類し，それらの特徴を整理する．

以後，外部ノードを EN（External Node），内部ノードを IN（Internal Node），両ノードが共にアクセス可能な専用サーバを RS（Rendezvous Server）と略する．

1. **Hole punching 方式**

   Hole punching は EN が RS から IN に対応するマッピングアドレスを取得し，そこに向けて通信することにより NAT 越え通信を実現する技術であり，文献 [22,23] において詳細に議論されている．IN は定期的に RS と通信を行い，NAT では IN に対するマッピングアドレスが割り当てられる．RS は IN から送信されたパケットの送信元 IP アドレスおよびポート番号から，マッピングアドレスを取得することができる．EN は RS より IN のマッピングアドレスを取得し，マッピングアドレス宛へ通信することにより，IN への通信を実現している．

   この方式は最も普及している Cone NAT [2]に対応できることから，既に実用化されている．しかし，UDP 通信アプリケーションに限定されたり，Symmetric NAT [3]に対応できないなどの課題がある．

   Hole punching を利用した代表的な技術として，STUN（Simple Traversal of UDP Through Network Address Translators） [137] がある．このほか，Hole punching を利用した IPv6 over UDP/IPv4 技術として Teredo [138] がある．近年は STUN を拡張することにより，TCP や Symmetric NAT に対応できる手法 [139–141] が検討されている．

2. **サーバ中継方式**

   サーバ中継方式は EN と IN 間の通信を RS が仲介することで NAT 越え通信を実現する．この方式は Cone NAT と Symmetric NAT の両方に対応することができる．しかし，全ての通信が RS を経由するため，RS にネットワーク負荷や処理負荷が集中したり，RS の設置や二重化などにコストがかかるという課題がある．また経路が冗長になることなどから，今後さらに普及する P2P 通信の特徴である柔軟性やリアルタイム性が失われる懸念がある．

   IETF（Internet Engineering Task Force）では STUN の追加機構としてサーバ中継方式が定義されており，TURN（Traversal Using Relay NAT） [142] と呼ばれている．

3. **内部動的マッピング方式**

   この方式は NAT に機能を実装し，IN からの指示により動的にマッピングを行う方式である．IN は NAT から設定されたマッピングアドレスを取得して利用することができる．EN がマッピングアドレスを取得するために，通常 IN はアプリケーションサーバとして用意された RS へマッピングアドレスを通知する必要がある．

   内部動的マッピング方式として UPnP（Universal Plug and Play） [143] や NAT Port Mapping Protocol [144] がある．

[2]宛先が変化しても割り当てられるポート番号が変化しない型式.
[3]宛先が変化すると割り当てられるポート番号も必ず変化する型式.

4. **SIP 拡張方式**

   近年, 個人間のリアルタイムコミュニケーションに必要となる SIP（Session Initiation Protocol）[125] が注目されている．SIP 拡張方式は SIP メッセージのフォーマットを拡張することにより，マッピングアドレスを通信相手に通知することにより，NAT 越え通信を実現する．SIP はインターネットアプリケーションや P2P 通信との親和性も高いことから，有望な手法といえる．しかし，SIP ベースのアプリケーションに限定されることや，仕組みが複雑であるなどの課題がある．

   SIP 拡張方式として NUTSS（NATs, URIs, Tunnels, SIP, and STUNT）[145] や，マッピングアドレス取得に STUN や TURN を用いるフレームワークとして ICE（Interactive Connectivity Establishment） [146] がある．

5. **トンネリング方式**

   トンネリング方式は EN または RS と NAT 間において IN 宛のパケットをカプセル化し，NAT でデカプセルして内部へ転送する．AVES（Address Virtualization Enabling Service）[147] は AVES 対応 DNS サーバと waypoint と呼ぶ RS を導入する．EN は AVES 対応 DNS サーバに IN の名前解決を行った後，IN 宛のパケットを waypoint へ送信する．waypoint は受信したパケットの宛先を IN へとアドレス変換した後，NAT との間に IP-in-IP トンネルを形成して転送する．NAT はデカプセル化して IN へ転送することにより NAT 越え通信を実現する．通信を行うエンドノードには一切の実装を必要としないが，中継転送やカプセル化による通信遅延の増加やスループットの低下が発生するため，リアルタイム性が失われるなどの課題がある．

   このほか，EN と NAT 間でトンネルを形成して NAT 越えを実現する技術として，NATS（NAT with Sub-Address） [148] がある．

6. **IP ルーチング拡張方式**

   この方式は IP のルーチング方式を拡張することにより，NAT 配下にパケットを転送する．IP パケットに新たなヘッダを追加し，複数の宛先 IP アドレスを扱えるように拡張する．EN は宛先として NAT のグローバル IP アドレスを，追加したヘッダに IN のプライベート IP アドレスを記載する．NAT は EN からのパケットを受信すると，NAT 処理を行わず，記載されているプライベート IP アドレスへ転送することにより，NAT 越え通信を実現する．しかし，EN，NAT，IN の全てがプロトコルスタックを拡張する必要がある．またパケットフォーマットを変化してしまうため，プロトコルタイプをチェックするような装置では本来の制御が行われないなど，他のシステムに影響を及ぼす可能性がある．

   IP ルーチング拡張方式として，4+4 [149] や IPNL（for IP Next Layer） [150] などがある．

表 5.1 に既存技術の実装箇所と必要な装置についてまとめる．アプリケーションレベルの解決手法は，NAT のマッピング処理の仕組みをそのまま利用するため，既存の NAT を変更する必要がない．その代わり，エンドノードのアプリケーションに機能を実装する必要がある．UPnP はさらに NAT に機能を実装する必要があるが，既に多くの NAT に実装されているため，導入は容易で

表 5.1　NAT 越えの既存技術とその実装箇所

	実装方法	実現方式	実装箇所			RS
			EN	IN	NAT	
STUN	APP	Hole punching	✓	✓	—	STUN サーバ
TURN	APP	サーバ中継	✓	✓	—	TURN サーバ
UPnP	APP	内部動的マッピング	✓	✓	✓	アプリケーションサーバ *1
ICE	APP	SIP 拡張	✓	✓	—	STUN/TURN サーバ
AVES	NET	トンネリング	—	—	✓	DNS サーバ *2，waypoint
4+4	NET	IP ルーチング拡張	✓	✓	✓	なし

APP：アプリケーションレベルの解決手法　NET：ネットワークレベルの解決手法
*1 EN がマッピングアドレスを取得するために必要（NAT 越えのために直接必要ではない）
*2 AVES 対応の特殊な DNS サーバ

表 5.2　NAT 越え技術の要求条件と既存技術の満足度

	STUN	TURN	UPnP	ICE	AVES	4+4
汎用性	×	×	×	×	○	○
実現の容易さ	○	○	○	○	△	×
TCP 通信への対応	×	○	○	○	○	○
Symmetric NAT への対応	×	○	×	○	—*	—*
低遅延	○	×	○	△	×	○
高スループット	○	△	○	△	×	○

* NAT は独自処理を行うため，マッピング機能は不要である．よって評価対象外とする．

ある．一方，ネットワークレベルの解決手法は NAT と EN または RS に機能を実装する代わりに，エンドノードは通常のアプリケーションを利用できる．また，アプリケーションレベルの解決手法，ネットワークレベルの解決手法に問わず，RS には高い耐障害性が要求されるといった課題がある．4+4 は RS が必要ないが，全てのカーネルを改造しなければならない．

表 5.2 に本研究における NAT 越え技術の要求条件と，既存技術の満足度を示す．表中の “○”，“△”，“×” は各要求条件をみたしているかどうかを示す．“△” は場合によって満足しない，または一部満足していないことを表す．

汎用性は，“○” がアプリケーションに依存しないことを，“×” がアプリケーションに依存することを示す．これはアプリケーションレベルの解決手法とネットワークレベルの解決手法の違いに対応する．ネットワークレベルの解決手法は汎用性がある一方，NAT の変更やカーネルへの機能実装が必要であるため，実現の容易さはアプリケーションレベルの解決手法より劣る．すなわち，両者の要求条件はトレードオフの関係にある．4+4 は IP ヘッダの構成が独自のため，EN，NAT だけでなく，IN に対してもプロトコルスタックに大きな変更を加える必要がある．そのため，他のネットワークレベルの技術に影響を及ぼすことが懸念される．例えば，IPsec を利用した場合，4+4

は NAT においてヘッダの内容を変更するため，パケットが偽造されたものと見なされてしまうなどの問題が生じる．これらの理由により，AVES 及び 4+4 の実現の容易さは，“△” と “×” とした．

UDP だけでなく TCP にも対応することが望まれるが，STUN は TCP 通信に対応できない．さらに Cone NAT だけでなく Symmetric NAT にも対応することが望まれているが，STUN と UPnP は原理上，Symmetric NAT に対応することはできない．

TURN や AVES は全ての通信パケットが RS を中継するため，遅延が増加する．ICE は NAT の種類に応じて STUN と TURN を切り替えるため，通信パケットが RS を中継する場合がある．このため，TURN と AVES の遅延は “×”，ICE の遅延は “△” とした．

また，RS と EN/NAT の距離が離れると，TCP の場合ラウンドトリップタイムが大きくなるため，スループットが低下することが想定される．AVES はさらにカプセル化処理を行うため，TURN よりスループットが低下する．文献 [151] によると，カプセル化を行った場合，スループットが 30 % 低下することが報告されている．ICE は上記でも述べた通り，TURN の仕組みを利用した場合はスループットが低下する場合がある．従って，TURN と ICE のスループットは “△”，AVES のスループットは “×” とした．また，STUN，UPnP はエンドツーエンドによる通信を行い，通信パケットに対しては NAT 処理しか行われないため，低遅延，高スループットを実現できる．4+4 は独自のルーチング処理を行うが，このための処理時間は NAT のアドレス変換処理の 40 % 以下である [149]．従って，STUN，UPnP，4+4 の遅延とスループットは “○” とした．

以上のことから，ネットワークレベルの解決手法の課題は，実現の容易さとエンドツーエンドの通信性能にあるといえる．

5.3 提案方式

提案方式は汎用的な解決を実現できるネットワークレベルの解決手法に分類できる．また NAT 越えを行うために新たな RS を導入せず，EN と NAT だけで NAT 越え問題の解決を実現する．本提案方式は EN が IN へ通信を開始する際，EN 側から NAT に対してマッピングを行うよう指示する．これを外部動的マッピング方式と呼ぶ．また，この処理を実現するプロトコルとして NAT-f を定義する．上記処理は Mapping Request と Mapping Response メッセージによる 1 往復のネゴシエーションにより構成される．その後，EN は送信パケットの宛先がマッピングアドレスと一致するようにアドレス変換を行うことにより，NAT 越え通信を実現する．RS による中継転送やカプセル化処理を行わないため，既存のネットワークレベルの解決手法のような通信遅延の増加やスループットの低下は発生しない．

5.3.1 初期登録情報

図 5.1 にシステム構成と初期設定情報を示す．EN と NAT は NAT-f 機能を実装し[4]，IN への機能実装は行わない．Dynamic DNS（以下 DDNS）サーバ [111] は IN の名前解決のために利用する．

[4]NAT-f の機能を実装した NAT を NAT-f ルータと呼ぶ．

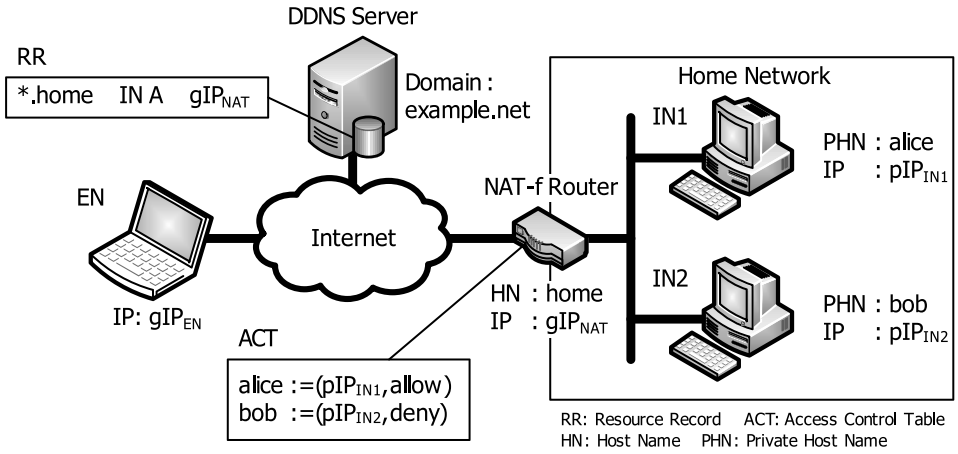

図 5.1　提案方式のシステム構成と初期設定情報

表 5.3　IN が複数存在する場合の DNS 登録パターン

	登録レコード	登録内容			
方法 1	Wildcard A	∗.*home*	IN	A	gIP_{NAT}
方法 2	A & CNAME	*home*	IN	A	gIP_{NAT}
		alice.home	IN	CNAME	*home*
		bob.home	IN	CNAME	*home*
方法 3	A	*alice*	IN	A	gIP_{NAT}
		bob	IN	A	gIP_{NAT}

NAT 越えのための機能は一切不要であり，既に運用されているものをそのまま利用できる．

事前準備として，ホームネットワークのユーザは DDNS サービスプロバイダに登録し，ホスト名を取得する．以後，*example.net* を管理するプロバイダからホスト名 *home* を取得したものと仮定する．NAT-f ルータのグローバル IP アドレス gIP_{NAT} は取得したホスト名とともに DDNS サーバによって管理されるものとする．

複数の IN を外部へ公開する場合，DDNS サーバには表 5.3 に示す 3 つの方法のいずれかにより登録を行う．DDNS サーバがワイルドカード機能を利用できる場合，ホスト名をワイルドカード A レコードとして登録する．ワイルドカードとはアスタリスクラベル “∗”で始まるドメイン名に対して，任意の文字列をドメインの先頭に指定しても，1 つのリソースレコードにより IP アドレスを取得できる機能である [152]．ワイルドカードが利用できない場合は，CNAME レコードを用いる．取得したホスト名は A レコードとして，IN の名前を CNAME レコードとして複数登録する．これら 2 つの登録方法では，NAT-f ルータのホスト名がインターネット側からホームネットワークを特定するために利用される．DDNS サーバがワイルドカード，および CNAME 登録に対応していない場合は，複数の A レコードを登録することになる．本章ではワイルドカードに対応した DDNS サーバとして以後説明する．

また IN の名前，プライベート IP アドレス，および外部からのアクセス許可情報を

$$alice := (pIP_{IN1}, allow) \qquad bob = (pIP_{IN2}, deny) \tag{5.1}$$

として NAT-f ルータのアクセス制御テーブル ACT（Access Control Table）に登録する．IN の名前はユーザが自由に決めることが可能で，インターネット上でユニークである必要はなく，ホームネットワーク内で IN を識別できればよい[5]．一般のホスト名と区別するため，これをプライベートホスト名と呼ぶ．アクセス許可情報には許可（*allow*）または拒否（*deny*）が設定され，該当する IN への NAT 越え通信可否を制御する．例えば式 (5.1) は，IN1 のプライベートホスト名が *alice*，IP アドレスが pIP_{IN1} であり，外部からのアクセスを許可することを表している．

5.3.2 動作概要

図 5.2 に EN から IN1 へ通信を開始する場合における提案方式の通信シーケンスを示す．提案方式における通信は以下に示す 3 フェーズから構成される．

1. **DNS 名前解決処理**

   Step 1: EN は IN1 へ通信を開始する際，NAT-f ルータの FQDN の先頭に IN のプライベートホスト名を付加した “*alice.home.example.net*” を用いて DDNS サーバに名前解決の依頼を行う．DDNS サーバはワイルドカード機能により，NAT-f ルータの IP アドレス gIP_{NAT} を DNS A Reply により応答する．

   Step 2: EN はカーネルにおいて DNS A Reply をフッキングして，取得した NAT-f ルータの IP アドレスを仮想 IP アドレス vIP_{IN1} に書き換える．このとき仮想 IP アドレスは，

   $$vIP_{IN1} := (alice, gIP_{NAT}) \tag{5.2}$$

   のように IN1 のプライベートホスト名と NAT-f ルータの IP アドレスと関連付けられ，名前関連テーブル NRT（Name Relation Table）にキャッシュされる．仮想 IP アドレスとは IN を一意に特定するために割り当てる IP アドレスであり，EN の IP 層より上位でのみ有効な値である．仮想 IP アドレスへの書き換えの必要性については 5.3.3 項で述べる．ここで生成した NRT は，後に実行する NAT-f ネゴシエーションで通知すべき情報を特定するために用いられる．アプリケーションへは仮想 IP アドレスを IN の IP アドレスとして報告する．

2. **NAT-f ネゴシエーション処理**

   Step 3: アプリケーションはソケットを通じて *alice* に向けてデータを送信する処理を行う．これにより，宛先が仮想 IP アドレスである TCP/UDP パケット

   $$gIP_{EN} : s \rightarrow vIP_{IN1} : d \quad [proto] \tag{5.3}$$

[5]ただし，DDNS サーバがワイルドカードまたは CNAME に対応している場合に限る．複数の A レコードを登録する場合は，インターネット上でユニークでなければならない．

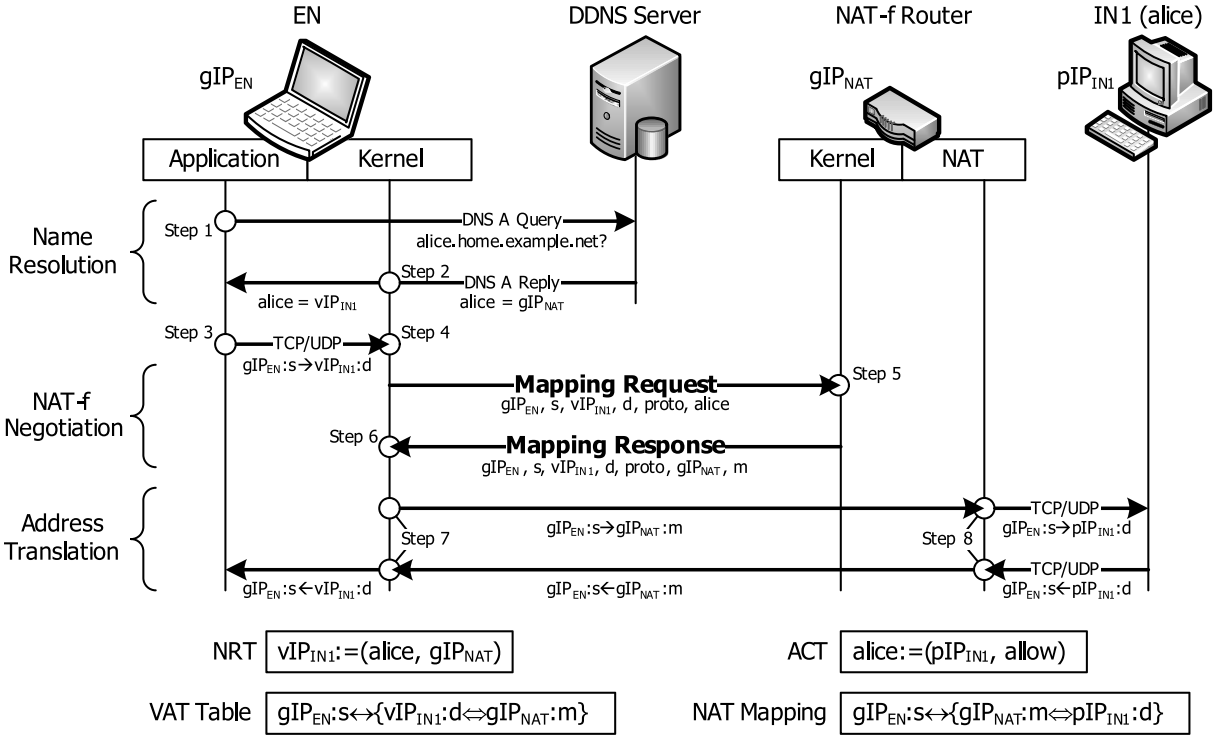

図 5.2　提案方式における NAT 越え通信シーケンス

が IP 層へ渡される．

Step 4: 宛先 IP アドレスが仮想 IP アドレスとなっている TCP/UDP パケットが IP 層に渡されると，送信元/宛先 IP アドレスとポート番号，およびプロトコル番号（すなわち通信識別子）をキーとして，仮想アドレス変換（VAT; Virtual Address Translation）テーブルを参照する．VAT テーブルとは仮想 IP アドレスと NAT-f ルータで割り当てられたマッピングアドレスとの相互変換関係が記されたテーブルで，NAT-f ネゴシエーション完了時に生成される．該当する情報が存在すれば，Step 7 の動作を行う．
該当する情報が存在しなければ，宛先 IP アドレス vIP_{IN1} をキーとして NRT を参照して仮想 IP アドレスに関連付けられた情報，すなわち式 (5.2) に示したエントリから NAT-f ルータの IP アドレス gIP_{NAT} と IN1 のプライベートホスト名 *alice* を取得する．そして TCP/UDP パケットを一時的に待避させてから，NAT-f ネゴシエーションを開始する．EN はネゴシエーションのトリガとなった式 (5.3) の TCP/UDP パケットの通信識別子，および IN1 のプライベートホスト名 *alice* を Mapping Request メッセージに載せて NAT-f ルータに通知する．

Step 5: NAT-f ルータは Mapping Request に記載されているプライベートホスト名を取得して，ACT をチェックする．一致するプライベートホスト名が存在し，かつアクセスが許可されていれば，受信した情報と該当する IN のプライベート IP アドレスからマッピング情報を生成する．ここでは，式 (5.1) より *alice* のプライベート IP アドレスが pIP_{IN1}

であることがわかるため，以下のようなマッピング情報が生成される．

$$\text{NAT-f Router:} \qquad gIP_{EN} : s \leftrightarrow \{gIP_{NAT} : m \overset{NAT}{\Longleftrightarrow} pIP_{IN1} : d\} \quad [proto] \tag{5.4}$$

これは IP アドレス・ポート番号が $gIP_{EN} : s$ および $pIP_{IN1} : d$，すなわち EN と IN1（*alice*）間の通信に対応するマッピングアドレスが $gIP_{NAT} : m$ であることを意味する．NAT-f ルータは先ほど EN から受信した情報とマッピングアドレスを Mapping Response メッセージに載せて，EN へ応答する．

Step 6: EN は NAT-f ルータから Mapping Response を受信すると，取得した情報から仮想 IP アドレスとマッピングアドレスの相互変換関係を示すエントリ

$$\text{EN:} \qquad gIP_{EN} : s \leftrightarrow \{vIP_{IN1} : d \overset{VAT}{\Longleftrightarrow} gIP_{NAT} : m\} \quad [proto] \tag{5.5}$$

を生成し，VAT テーブルに格納する．その後，一時的に待避していた TCP/UDP パケットを復帰させて NAT-f ネゴシエーションを完了する．

3. **通信中の仮想アドレス変換処理**

Step 7: 復帰した TCP/UDP パケットは式 (5.5) に示す VAT テーブルのエントリに基づいて，宛先 IP アドレスとポート番号が $vIP_{IN1} : d$ から $gIP_{NAT} : m$ に変換された後，NAT-f ルータへ送信される．

Step 8: NAT-f ルータは既に Step 5 において式 (5.4) に示すマッピングがなされているため，通常の NAT 処理により宛先 IP アドレスとポート番号を $gIP_{NAT} : m$ から $pIP_{IN1} : d$ に変換して，該当する IN1（*alice*）へパケットを転送する．

以上の処理により，EN から IN1 へのパケット転送が完了する．IN1 から EN への応答パケットに対しては，上記と逆の変換を行う．EN では，マッピング情報及び VAT テーブルに基づいて，送信元 IP アドレスとポート番号を変換してからアプリケーションへと渡す．以後，EN と IN1 間の通信は Step 7，Step 8 を繰り返す．このようにして EN から IN1 への通信開始を可能とし，NAT 越え通信を実現することができる．

5.3.3 同時通信

図 5.1 において EN が IN1（*alice*）と通信中に同一ホームネットワークの IN2（*bob*）へ通信を開始する場合を考える．EN は DNS 応答パケットに記載された NAT-f ルータの IP アドレスを別の仮想 IP アドレス vIP_{IN2} に書き換えて，アプリケーションに報告する．このとき NRT にキャッシュされる情報は，

$$vIP_{IN2} := (bob, gIP_{NAT}) \tag{5.6}$$

となる．

アプリケーションは宛先 IP アドレスを *alice* 宛なら vIP_{IN1}，*bob* 宛なら vIP_{IN2} として設定するため，IP 層では NAT 配下のどの IN に対して NAT-f ネゴシエーション，および仮想アドレス変換を行えばよいのかを区別することがきる．

例えば，EN が IN2 に対して送信しようとする TCP/UDP パケットを

$$gIP_{EN} : s2 \rightarrow vIP_{IN2} : d \quad [proto] \tag{5.7}$$

のように IN1 と同じ宛先ポート番号であると仮定する．これはホームネットワーク内に複数に同一サービスを提供するサーバを設置する場合が考えられる．EN は Step 4 の手順により，NAT-f ルータに対して Mapping Request を送信する．NAT-f ルータは Step 5 で示したとおり，EN から通知された IN2 のプライベートホスト名と ACT から，*bob* に対応したマッピング情報

$$\text{NAT-f Router:} \qquad gIP_{EN} : s2 \leftrightarrow \{gIP_{NAT} : m2 \overset{NAT}{\Longleftrightarrow} pIP_{IN2} : d\} \quad [proto] \tag{5.8}$$

を生成する．

EN は Mapping Response により通知されたマッピングアドレス $gIP_{NAT} : m2$ から，エントリ

$$\text{EN:} \qquad gIP_{EN} : s2 \leftrightarrow \{vIP_{IN2} : d \overset{VAT}{\Longleftrightarrow} gIP_{NAT} : m2\} \quad [proto] \tag{5.9}$$

を生成し，VAT テーブルに格納する．この結果，*alice* 宛の通信は NAT-f ルータのポート番号 *m* に対して，また *bob* 宛の通信はポート番号 *m2* に対して送信することになる．

このように仮想 IP アドレスを使用することにより，EN は NAT-f ルータ配下の複数の IN と同時に通信を行うことが可能になる．

5.3.4 異なるホームネットワーク内の IN 同士の通信

NAT 越え技術は異なるプライベートアドレス空間に存在するノード間の通信にも適用できることが望ましい．本提案方式においては，EN に実装していた DNS 応答書き換え処理，ネゴシエーション処理および仮想アドレス変換処理を，それぞれの IN を配下に持つ NAT-f ルータに追加実装することにより，このようなシステムを実現することが可能である．本研究では異なるプライベートアドレス空間のノード間の通信を実現するシステムを CIPA（Communication between terminals in Independent Private Address areas；サイパ） [153] と呼ぶ．

図 5.3 にプライベートネットワーク間の通信シーケンスを示す．IN1（*alice*）から IN3（*carol*）に対して通信を開始する場合について述べる．図 5.3 と図 5.2 の Step は対応しており，同一の処理を行う．NAT-f Router 1 は EN で行っていたカーネル部の処理を実行する前に，以下の NAT 処理（Step 3.5）が行われる．

Step 3.5: IN1 から受信した TCP/UDP パケット

$$pIP_{IN1} : s \rightarrow vIP_{IN3} : d \quad [proto] \tag{5.10}$$

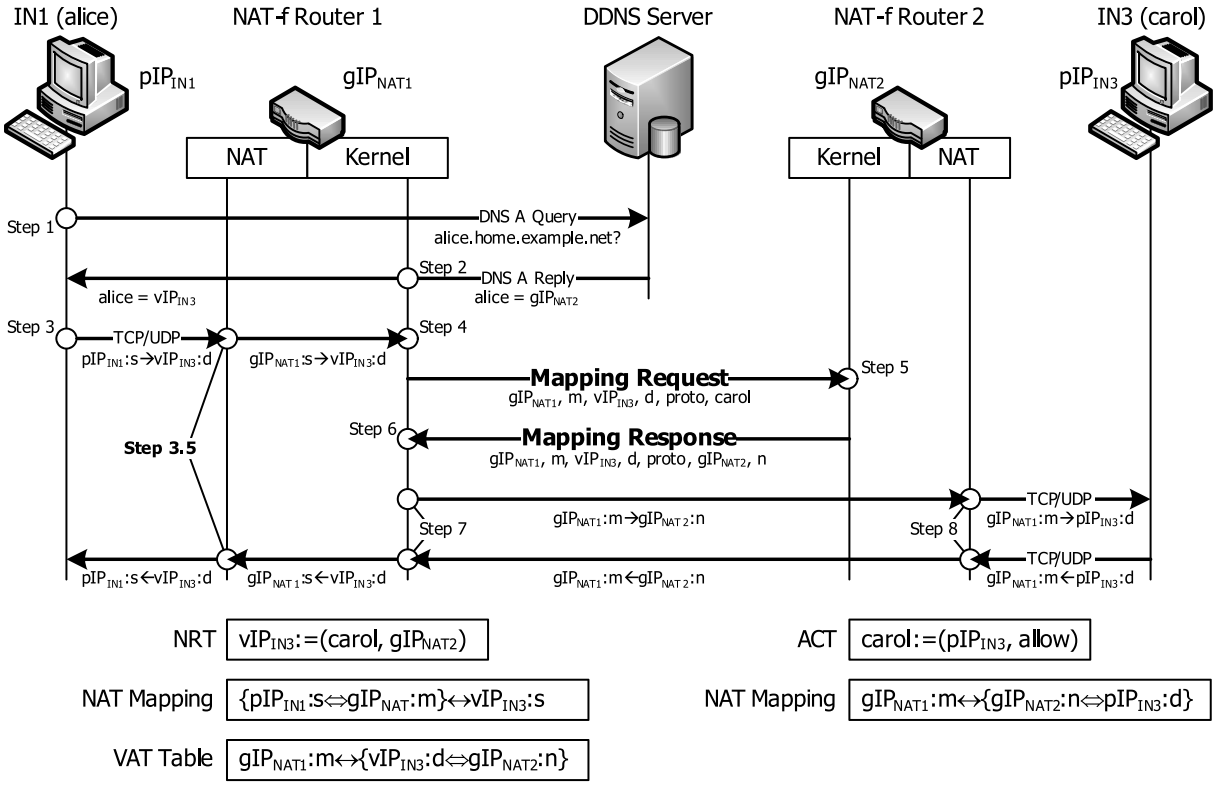

図 5.3　プライベートネットワーク間の通信シーケンス

は通常の NAT 処理に従って，送信元 IP アドレスとポート番号が $pIP_{IN1} : s$ から $gIP_{NAT1} : m$ に変換された後，IP 層に渡される．このとき，生成されるマッピング情報は

$$\text{NAT-f Router 1:} \qquad \{pIP_{IN1} : s \overset{NAT}{\Longleftrightarrow} gIP_{NAT1} : m\} \leftrightarrow vIP_{IN3} : d \quad [proto] \tag{5.11}$$

のように示される．

以後，5.3.2 項の手順により，NAT-f Router 2 には NAT マッピング情報

$$\text{NAT-f Router 2:} \qquad gIP_{NAT1} : m \leftrightarrow \{gIP_{NAT2} : n \overset{NAT}{\Longleftrightarrow} pIP_{IN3} : d\} \quad [proto] \tag{5.12}$$

が，NAT-f Router 1 には VAT エントリ

$$\text{NAT-f Router 1:} \qquad gIP_{NAT1} : m \leftrightarrow \{vIP_{IN3} : d \overset{VAT}{\Longleftrightarrow} gIP_{NAT2} : n\} \quad [proto] \tag{5.13}$$

が生成される．

従って *alice* から *carol* への通信は，NAT-f Router 1 において NAT，VAT の順序でアドレス変換が行われる．さらに NAT-f Router 2 において，NAT によるアドレス変換が行われ，*carol* への通信開始を実現できる．

このように，NAT-f ネゴシエーションは 2 台の NAT-f ルータ間で行うので，実際に通信するエンドノードには特別な機能を実装する必要はない．また，IN1（*alice*）と IN3（*carol*）はそれぞれ異なるホームネットワークに存在するため，同一のプライベート IP アドレスとなる場合が十分に想定されるが，pIP_{IN1} と pIP_{IN3} が同一であっても上記変換処理により通信可能である．

表 5.4　NAT 越え要求条件に対する提案方式の満足度

	提案方式の満足度
汎用性	○
実現の容易さ	△
TCP 通信への対応	○
Symmetric NAT への対応	○
低遅延	○
高スループット	○

5.3.5　既存技術との比較

表 5.4 に NAT 越え技術の要求条件に対する，提案方式の満足度を示す．提案方式はネットワークレベルの解決手法であるため，アプリケーションから独立しており，汎用性を有する．EN と NAT のカーネルに変更が必要となるが，4+4 のような大幅なプロトコルスタックの改良は不要で，また通信パケットのフォーマットも変更しないため，他のネットワークレベルの技術との互換性がある．本提案方式の実装では，ユーザにセットアップスクリプトを提供することにより，カーネルへの機能実装でありながら通常のアプリケーションと同じ感覚でインストールすることができる．

ただし，ホームネットワークに設置されている既存の NAT ルータを NAT-f 対応ルータに変更する必要があるため，STUN や TURN のように既存の NAT ルータをそのまま利用できる方式と比べると，提案方式の実現の容易さは劣るといえる．しかし，近年では NGN（Next Generation Network）により多様化していくサービスに対応したり，ユビキタスネットワーク実現のために，ホームゲートウェイに様々な機能を実装することが検討されている [32, 154–156]．従って，ユーザは新しいサービスあるいは機能を利用するに当たり，ホームゲートウェイを新たに置き換えることはそれほど大きな問題ではないと考えられる．以上のことから，提案方式の実現の容易さは “△” とした．

NAT-f ネゴシエーションではトリガとなったパケットのプロトコルタイプに応じた NAT マッピングが行われるため，TCP/UDP の双方に対応することができる．NAT-f ネゴシエーションにより生成される VAT テーブルのエントリには EN の IP アドレスと共にポート番号も含んでいるため，NAT では EN と IN 間で確立するコネクションごとにマッピング処理が行われる．そのため，Cone NAT と Symmetric NAT の双方に対応することができる．

提案方式では NAT-f ネゴシエーションおよびその後の通信は全てエンドツーエンドで実現できるため，TURN や AVES のような冗長経路による通信遅延は発生しない．また，通信パケットに対してカプセル化処理は行わず，仮想アドレス変換と NAT 処理を実行する．この方法により，提案方式を実装しなかった場合と比べて同等のスループットを得ることができる．なお，提案方式は通信開始時に EN と NAT 間で，NAT-f による事前ネゴシエーションを行うが，事前ネゴシエーション自体は 4+4 を除く他の既存技術でも必須の処理である．特に NAT-f はカーネルで処理するため，極めて短時間で終了し，通信開始時の他の処理に与える影響はほとんど無い．NAT-f にかか

る時間とスループットの実測値に関しては，5.5.2 項で示す．

アプリケーションレベルの解決手法の既存方式は必ず IN からのアクションによりマッピングを行っているが，提案技術では EN からのアクションによりマッピング処理を行うことができる．これは今後一層普及する家庭内の情報通信機器に対して，NAT 越え通信を実現するための特殊な機能を保持しなくてもよいことを意味する．

さらに，RS のような特殊な装置も不要になるため，耐障害性にも優れるなどの特徴がある．RS の機能を EN や NAT に分散して実装することにより，RS を不要にすることは技術的には可能であるが，運用面においていくつかの課題が生じる．AVES は AVES 専用 DNS サーバを運用する必要がある．DNS サーバ機能を NAT に実装することは，ホームネットワークで DNS を運用・管理することになり一般のユーザには難しい．STUN，TURN，ICE は STUN サーバおよび TURN サーバが必要となる．これらのサーバには複数の IN のマッピング情報が一元管理されているため，EN はこのサーバに問い合わせを行えば，該当する IN に対するマッピングアドレスを取得することができる．しかし，RS 機能を NAT に実装した場合，従来一カ所で管理されていた情報を個々の NAT に分散管理することになる．そのため，EN は IN のマッピングアドレスを取得する際，その IN の情報が登録されている NAT の所在を特定する仕組みが別途必要になり，非効率なシステムになってしまう．以上のことより，既存技術の RS は方式として必要な装置といえる．

提案方式では一般の DNS サーバを DDNS サーバに置き換えた構成になる．DNS サーバは全ての方式で必須の装置であり，STUN，TURN，UPnP，ICE は各 RS の名前解決のために DNS を利用する．AVES は RS が専用 DNS サーバの役割を果たし，IN の名前解決のために利用する．4+4 は DNS サーバよりルーチングに必要な IP アドレスを取得する．提案方式が DDNS を利用する必然性は，一般にホームネットワークには変動グローバルアドレスが割り当てられるため，ホームネットワークのユーザが定期的に DDNS サーバに登録した情報を更新する必要があるためである．ホームネットワークに割り当てられるグローバルアドレスが固定である場合は，一般の DNS サーバでも構わない．

5.4 実装

プロトタイプシステムとして，NAT-f モジュールを FreeBSD 5.3-RELEASE の IP 層に実装した．図 5.4 に EN の実装概要を示す．NAT-f モジュールは 2.4 節にて示した GSCIP（Grouping for Secure Communication for IP） [33] のモジュール GPACK の一部として実装される．パケット受信時には IP 入力関数である `ip_input()` から，パケット送信時には IP 出力関数である `ip_output()` から GPACK を経由して，NAT-f モジュールにおいて NAT-f 対応の処理を終えたら TCP/UDP パケットを差し戻す形をとっている．NAT-f ネゴシエーションを実行する際，最初の TCP/UDP パケットを一時待避するが，パケットデータが格納されているメモリ領域は解放せず，カーネル内に留めておく．ネゴシエーションが完了した時点でこのパケットを `ip_output()` へ渡すことにより，一時中断していた通信を即座に開始することができる．これによりネゴシエーションに伴う遅延を最小限に抑えることが可能になる．

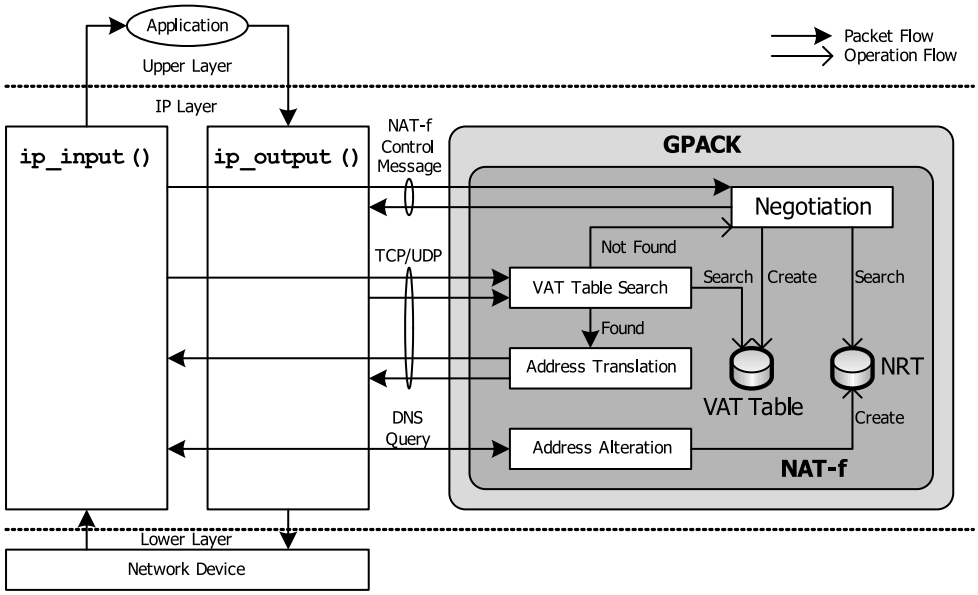

図 5.4　EN の実装概要

NRT，VAT テーブルはカーネルメモリ空間にハッシュテーブルとして実装する．NRT にキャッシュされるデータの生存時間は DNS 応答パケットに記載されている TTL 値を設定する．VAT エントリは無通信状態が一定時間以上続いた場合にカーネルタイマにより削除される．また TCP コネクションが切断された場合にも削除される．VAT エントリのタイムアウト値は NAT-f ネゴシエーションの応答に記載されている TTL 値を設定する．

仮想 IP アドレスは IN のプライベートホスト名に対応して割り当てられる．仮想 IP アドレスを “*A.B.C.D*” と表記した場合，各バイトには以下に示す値が設定される．*A* にはホームネットワークのネットワークアドレス上位 1 バイト目[6]と異なる値が設定され，NAT-f モジュールはこれにより仮想 IP アドレスであるか否かを識別する．プロトタイプシステムでは，実験的目的のために予約されているクラス E にあたる 240 を設定した．*B* は初期値として 0 が設定される．*C* は IN のドメイン名のハッシュ値，*D* は IN のプライベートホスト名のハッシュ値が設定される．ハッシュ関数の出力値の範囲は 1〜254 とした．

このように仮想 IP アドレスを割り当てることにより，異なるプライベートネットワークに同じプライベートホスト名のノードが存在しても，割り当てられる仮想 IP アドレスが異なるため両ノードを識別することができる．ハッシュが衝突した場合は，*B* を異なる値に設定することにより，IN のプライベートホスト名と仮想 IP アドレスは一意に対応する．

図 5.5 に NAT-f ルータの実装概要を示す．NAT-f ルータには今回実装した NAT-f モジュールに加えて，NAT デーモ `natd` [7]を動作させる．NAT-f ルータが受信したパケットは divert ソケットを通じて，`natd` でアドレス変換処理が行われる．図 5.6 に NAT マッピング手法を示す．NAT-f ルータは NAT-f ネゴシエーションパケットを受信すると，NAT-f モジュールにおいて受信した情報と

[6]通常，プライベートアドレスを示す 192（クラス C）が割り当てられている．
[7]FreeBSD で標準的に利用されるユーザランドで動作する NAT アプリケーション．

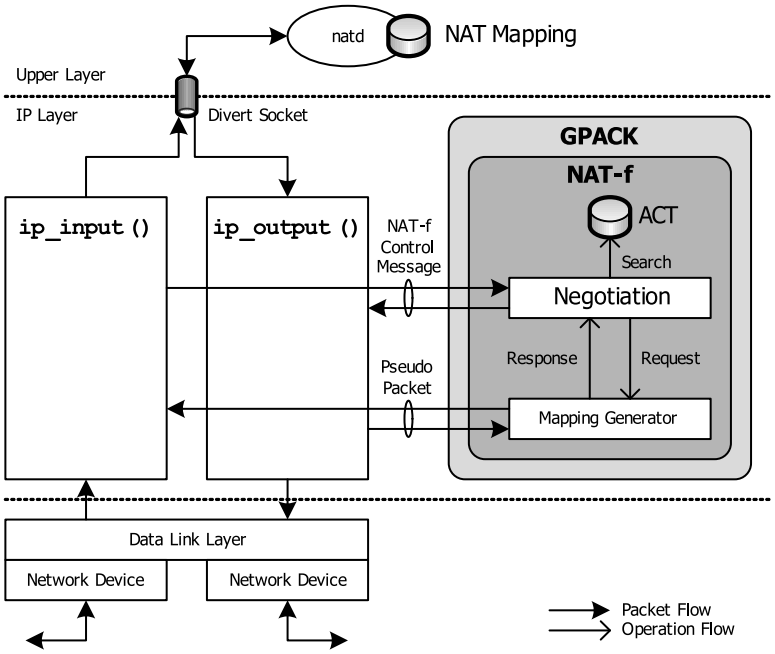

図 5.5　NAT-f ルータの実装概要

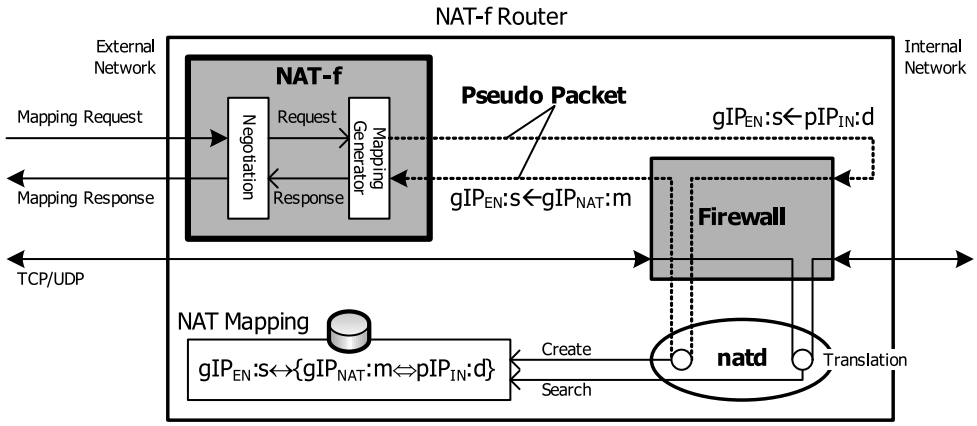

図 5.6　疑似パケットによる NAT マッピング手法

ACT の内容から

$$pIP_{IN} : d \rightarrow gIP_{EN} : s \quad [proto] \tag{5.14}$$

のようなパケットデータを作成する．これを疑似パケット [157, 158] と呼ぶ．疑似パケットは IN から EN へパケットが送信されたように見せかけたものであり，このパケットを `ip_input()` へ渡すと，`natd` は IN から EN へ送信されるパケットを受信したと判断して，マッピング処理を行う．アドレス変換後の疑似パケットは送信処理の際に NAT-f モジュールへ渡され，Mapping Response に記載する情報として使用される．疑似パケットは実際のネットワークには送信されず，Mapping Response が生成された後，破棄される．

ここで疑似パケットか否かを判断する方法を述べる前に，図 5.7 に Mapping メッセージと疑似

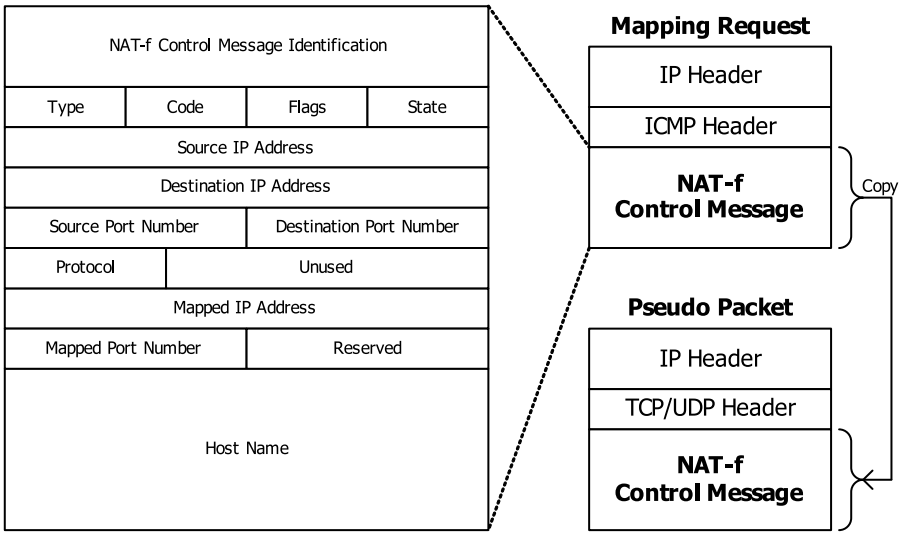

図 5.7　Mapping メッセージと疑似パケットのフォーマット

パケットのフォーマットを示す．Mapping Request/Response は ICMP Echo をベースに定義されている．ヘッダ部には NAT-f 制御メッセージを識別するための ID，Mapping Request/Response の識別情報（Type，Code）などが記載される．データ部には NAT-f ネゴシエーションのトリガとなった TCP/UDP パケットの通信識別子（送信元/宛先 IP アドレスとポート番号，およびプロトコル番号），NAT-f ルータで生成されるマッピングアドレス（Mapped IP Address と Mapped Port Number の組）とホスト名が記載される．

一方，疑似パケットは NAT マッピングを生成するためのパケットデータであるため，トランスポートヘッダは TCP または UDP となり，ポート番号はトリガパケットの情報が設定される．そのため，ポート番号を参照しても疑似パケットか否かを判断することはできない．そこで，TCP/UDP ペイロード部に Mapping Request のヘッダ部以降のデータをそのまま記載する．NAT-f ルータは疑似パケットを生成後，全ての送信 TCP/UDP パケットのペイロード部をチェックして NAT-f 制御メッセージヘッダがあるか確認する．これにより，疑似パケットを判断することができる．なお，Mapping Response は，疑似パケットの送信元 IP アドレス/ポート番号と TCP/UDP ペイロード部のデータ，すなわち Mapping Request の情報から生成される．

この疑似パケットによるマッピング生成手法によると，NAT 処理には一切の変更を必要としないため，Packet Filter [159] など他の NAT アプリケーションでもそのまま利用することができる．さらに NAT に NAT-f モジュールを実装するだけで，NAT-f ルータを実現することができるため，Linux などの異なるプラットフォームにおいても NAT-f ルータを容易に実現することが可能である．プロトタイプシステムにおける NAT マッピングのタイムアウト値は，UDP が 60 sec，コネクション確立後の TCP が 86,400 sec（24 時間）であり，これらの値を NAT-f ネゴシエーションの応答に記載する TTL 値とする．

プロトタイプシステムでは ACT はユーザが手動で設定する必要があるが，DHCP によるアドレス割り当て時や，NAT-f ルータが NBNS（NetBIOS Name Server） [160, 161] プロトコルを利用し

て配下ネットワークに存在する IN の NetBIOS 名を収集することにより自動生成することも可能である．

NAT-f 制御メッセージ Mapping Request および Mapping Response は前述した通り，ICMP Echo をベースとして定義されている．従って NAT-f ネゴシエーションの宛先装置が NAT-f に対応していない場合，通常の ICMP Echo Reply が返信され，NAT が NAT-f に対応しているか否かを確認することができる．非対応の場合，EN は仮想 IP アドレスを DNS 名前解決時に取得した本来の IP アドレスに変換する事を示すエントリ

$$\text{EN:} \qquad gIP_{EN} : s \leftrightarrow \{vIP_{IN} : d \overset{VAT}{\Longleftrightarrow} gIP_{NAT} : d\} \quad [proto] \tag{5.15}$$

を生成し，VAT テーブルに格納する．以後，EN は式 (5.15) に基づいたアドレス変換処理を行い，NAT ルータとの通信を開始する．これは，EN および NAT ルータが NAT-f を実装していない場合と同じ通信となる．

5.5 評価

5.5.1 動作検証

EN から IN へ FTP 接続を行った結果，ポート番号が変化してもファイル転送が行えることを確認した．また複数の IN に対して，同時に HTTP 通信ができることを確認した．その結果，EN と IN の間で自由な双方向通信が可能であることを実証できた．

5.5.2 性能評価

図 5.1 のシステム構成において，EN と IN が通信を行う場合の性能測定を行った．性能測定に使用した各装置の仕様は CPU が Pentium4 3.0 GHz，メモリが 512 MByte である．またネットワーク環境は 100BASE-TX の Ethernet であり，EN，NAT-f ルータ，DDNS サーバをスイッチで接続した．

提案方式のオーバヘッドを明らかにするために，実際の通信が開始されるまでの時間にはネットワークアナライザ Ethereal [74] を，また実装した NAT-f モジュールの内部処理時間には RDTSC（Read Time-Stamp Counter）[78] を用いて測定した．RDTSC は CPU のカウンタから周波数クロックを取得する命令で，モジュール処理に費やした時間を正確に算出することができる．

次に，EN における仮想アドレス変換処理が通信性能に与える影響を明らかにするために，Netperf [88] を用いて EN から IN への TCP/UDP スループットを測定した．比較のために提案方式を実装しない環境として，IN から EN へのスループットについても測定した．測定時間は 10 秒間とし，測定結果はいずれも 10 回試行の平均値である．

1. **通信開始時のオーバヘッド**

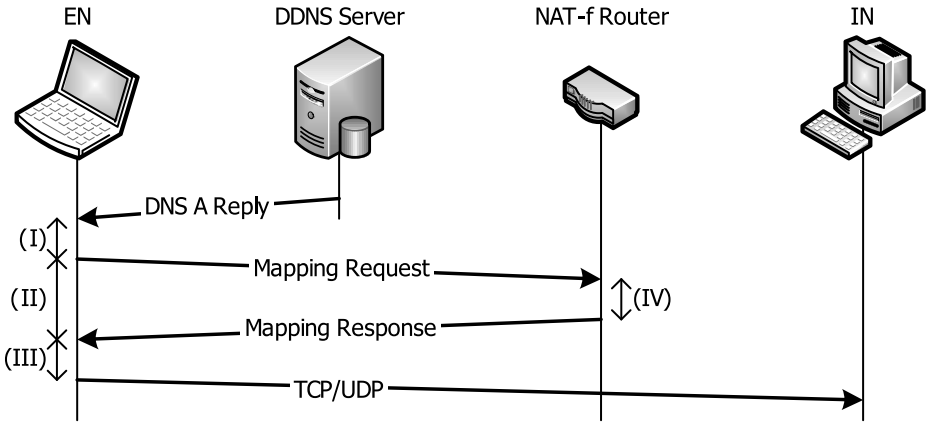

図 5.8　オーバヘッドの測定箇所

表 5.5　通信開始時におけるオーバヘッドの測定結果

測定箇所	処理内容	処理時間（μsec）
(I)	DNS A Reply のアドレス書き換え アプリケーションによる Socket 処理 VAT テーブルおよび NRT 検索 Mapping Request 生成および送信処理	238
(II)	EN と NAT-f Router 間の RTT*	388
	NAT-f Router における処理 (IV)	
(III)	VAT テーブル生成 仮想アドレス変換処理	27
(IV)	ACT 検索 疑似パケット生成 NAT マッピング生成処理 Mapping Response 生成および送信処理	114

* Round Trip Time

通信開始時のオーバヘッドに関して，図 5.8 に測定箇所を，表 5.5 にその測定結果を示す．EN が DNS 応答パケットを受信してから Mapping Request メッセージを送信するまでの時間は 238 μsec であった．このうち，DNS 応答書き換え処理（Step 2）が 7.13 μsec，NAT-f ネゴシエーション開始処理（Step 4）が 3.38 μsec を占めていた．また EN が Mapping Request を送信してから Mapping Response を受信するまでの時間は 388 μsec であり，NAT-f ルータのネゴシエーション処理時間（Step 5）は 114 μsec であった．EN はマッピング応答パケット受信後，27 μsec 後（Step 6+Step 7）に一時中断していた通信を開始した．すなわち，通信開始時に発生するオーバヘッドは約 650 μsec [8] となり，提案方式は実用上，通信開始に影響を与えないことがわかる．これは NAT-f ネゴシエーションのトリガとなった TCP/UDP パケットをカーネル内で待避し，復帰処理を行った結果である．このため，通信開始時に TCP

[8] $238 + 388 + 27 \fallingdotseq 650 \mu$sec

表 5.6　Netperf によるスループット測定値

Message Size (Byte)	TCP Stream (Mbit/s)		UDP Stream (Mbit/s)	
	EN → IN	EN ← IN	EN → IN	EN ← IN
64	93.2	93.1	49.3	49.3
128	93.2	93.2	66.0	66.0
256	93.2	93.2	79.6	79.6
512	93.2	93.2	88.8	88.8
1024	93.2	93.2	94.4	96.4
1472	93.2	93.2	96.4	96.4

EN → IN：提案方式による NAT 越え通信
EN ← IN：提案方式を実装しない通常の通信

における再送処理が発生することはない.

2. **EN-IN 間のスループット**
   表 5.6 に Netperf によるスループット測定値を示す．NAT-f 実装時，未実装時のスループットは TCP，UDP とも，どのメッセージサイズにおいても，両者の間には有意差が認められなかった．提案方式は通常の NAT マッピング処理と同等のスループットが得られており，EN 内における仮想アドレス変換処理によるオーバヘッドは無視できるほど小さい．また，アプリケーションレベルの解決手法とも同等の性能であり，カプセル化を行うトンネリング方式より高スループットを得られることが実証できた.

5.5.3　セキュリティに関する考察

既存のホームネットワークは NAT により内部の IP アドレスが隠蔽されていたため，特定の IN を標的とした攻撃を外部から実行することが困難であった．これは NAT により簡易的なセキュリティ対策が施された状態といえる．そのため，NAT 越え技術により IN はセキュリティ脅威にさらされる可能性が高くなる.

提案方式は ACT によるアクセス制御を行っているため，アクセスが許可されていない IN に対して，NAT-f ルータ外部からの指示でマッピングされることはない．またマッピングが生成された後は図 5.6 に示すように，EN と IN 間の通信に対してファイアウォールによるフィルタリング処理が行われる．NAT-f ルータ管理者は外部へ提供するサービスを制御することにより，不正アクセスなどの脅威から IN を保護することができる．本来，NAT はセキュリティのための機能ではないため，個々の IN がウィルス対策やパーソナルファイアウォールによりセキュリティ対策を施すことが重要である.

さらに通信の安全性を向上させるために，相手認証や暗号化通信を行うことが考えられる．本研究では NAT やファイアウォールとの親和性が高い暗号通信方式として，PCCOM（Practical Cipher Communication）[35] を提案している．3 章で述べたとおり，PCCOM はパケットのフォーマット

を変えずに本人性確認とパケットの完全性保証を実現しており，NAT 越え暗号化通信を可能としている．この技術は高スループットが得られることや，IP 層において動作するため，提案方式の利点を損なうことなく適用できる．

近年ファイアウォールにおいて DoS 攻撃を防止するために，ICMP Echo に応答しないようにフィルタを設定する場合がある．Mapping Request メッセージは図 5.6 に示すように，NAT-f ルータではファイアウォール処理を行う前に NAT-f モジュールの処理が実行される．そのため，NAT-f ルータは Mapping Request に対してのみ正しく動作することができる．また，大量の Mapping Request を送りつけられる DoS 攻撃の可能性が考えられる．この場合はパケットの送信元 IP アドレスの値を検証したり，NAT-f シーケンスの中で文献 [65] のようにクッキーの交換を行うなどの方式を導入する必要があると考えられる．

5.6 結論

NAT 越え通信を実現するための方式として外部動的マッピング方式を提案し，これを実現するためのプロトコルとして NAT-f を定義した．外部動的マッピング方式は内部ノードへの通信に先立ち，外部ノードが NAT-f ネゴシエーションにより NAT のマッピング処理を動的に実行させる．ネゴシエーション完了後，外部ノードは送信パケットの宛先をマッピングアドレスになるようにアドレス変換することにより NAT 越え通信を実現する．提案方式はアプリケーションに依存せず，専用のサーバが不要である．

プロトタイプシステムの実装を行い，複数の内部ノードと同時に通信できることを実証した．提案方式の評価を行った結果，通信開始時の遅延増加は 1 msec 以下であり，スループットは提案方式を実装しない場合と比べ，同等であることを確認した．

第6章　提案アーキテクチャによる応用研究

6.1　概要

これまでの移動透過性の研究は将来のネットワークを見越して，IPv6を前提としたものが多かった [16, 97–109]．しかし，IPv6は当初予想していたような普及をしていない．また，IPv6が普及を始めたとしても当分の間はIPv4とIPv6が混在することが想定されている．したがってIPv4においても移動透過性を実現することは，移動体通信の利便性向上の観点から大きな意義がある．

IPv4ネットワークではアドレス枯渇問題のため，全ての移動ノードMN（Mobile Node）にグローバルIPアドレスを割り当てることは困難であり，プライベートIPアドレスを積極的に利用する必要がある．Mobile IPv4 [15] においてはプライベートIPアドレスの利用を想定した方式がいくつか提案されている [42, 91, 162]．また筆者らが提案しているMobile PPC（Mobile Peer-to-Peer Communication protocol）[45] においても，MNにプライベートIPアドレスが割り当てられた場合の移動透過性が検討されている [128]．これらの提案はMNの通信相手ノードCN（Correspondent Node）がグローバルネットワーク上のサーバであることを想定しており，MNからCNに対して通信を開始後，MNがプライベートネットワークとグローバルネットワークの間を移動するケースを実現している．

しかし，近年DLNA（Digital Living Network Alliance）[163] に対応した情報家電機器が充実しつつあり，ユーザは外出先からこれらのコンテンツを利用したいという要求が高まっている．この場合，従来の考え方とは逆に，CNがプライベートネットワークに存在し，MNがグローバルネットワークに存在することになる．このようなケースでは，CNの直前にNAT（Network Address Translator）[21] が存在するため，一般にはMNからCNへ通信を開始することができない．これはNAT越え問題と呼ばれており，IPv4における大きな課題となっている．したがって，従来の移動透過性技術だけでは上記のようなニーズに対応することができない．

本章では5章にて提案したNAT越え技術NAT-f（NAT-free protocol）[46] を既存の移動透過性プロトコルと組み合わせることにより，新たな通信ケースを実現する応用手法について提案する．NAT-fはグローバルネットワーク上のノードとNATが通信に先立ちネゴシエーションを行い，NATマッピングを動的に生成する．その後の通信はNATに割り当てられたマッピングアドレス[1]を介してエンドツーエンド通信を確立する．グローバルネットワーク上のMNが通信中に移動すると，MNと所定の機器が移動透過性プロトコルを実行することにより通信を継続する．NAT-fと移動透過性プロトコルは処理タイミングが異なるため，独立性が高く，容易に組み合わせることができる．

[1] NATでマッピングされたIPアドレスとポート番号の組．

以下，6.2 節では既存技術の概要と IPv4 ネットワークにおける通信ケースを整理する．6.3 節において NAT-f を Mobile PPC に組み合わせた場合の通信手順ついて述べ，6.4 節でシステムの実装概要を示す．6.5 節において実装したシステムの評価，及びセキュリティや対応可能な通信ケースに関する考察を行う．6.6 節では提案アーキテクチャの他システムへの応用例について示す．最後に 6.7 節でまとめる．

6.2 既存技術

6.2.1 IPv4 ネットワークにおける移動パターン

図 6.1 に MN の移動パターンの例を示す．従来の研究では，CN はグローバルネットワークに存在することが前提である．MN の移動前，移動後のネットワークの組み合わせにより以下の 4 つのパターンが検討されている．

Pattern 1: グローバルネットワークからグローバルネットワークへの移動
Pattern 2: グローバルネットワークからプライベートネットワークへの移動
Pattern 3: プライベートネットワークからグローバルネットワークへの移動
Pattern 4: プライベートネットワークから異なるプライベートネットワークへの移動

Pattern 1 は最も基本的な移動パターンである．Pattern 2，Pattern 3 は移動前もしくは移動後の通信経路上に NAT が介在することになる．Pattern 4 はプライベートネットワークが階層的に構築された環境での移動が想定される．

以下に，Mobile IP と Mobile PPC による上記四つの移動パターンの実現方法を示す．

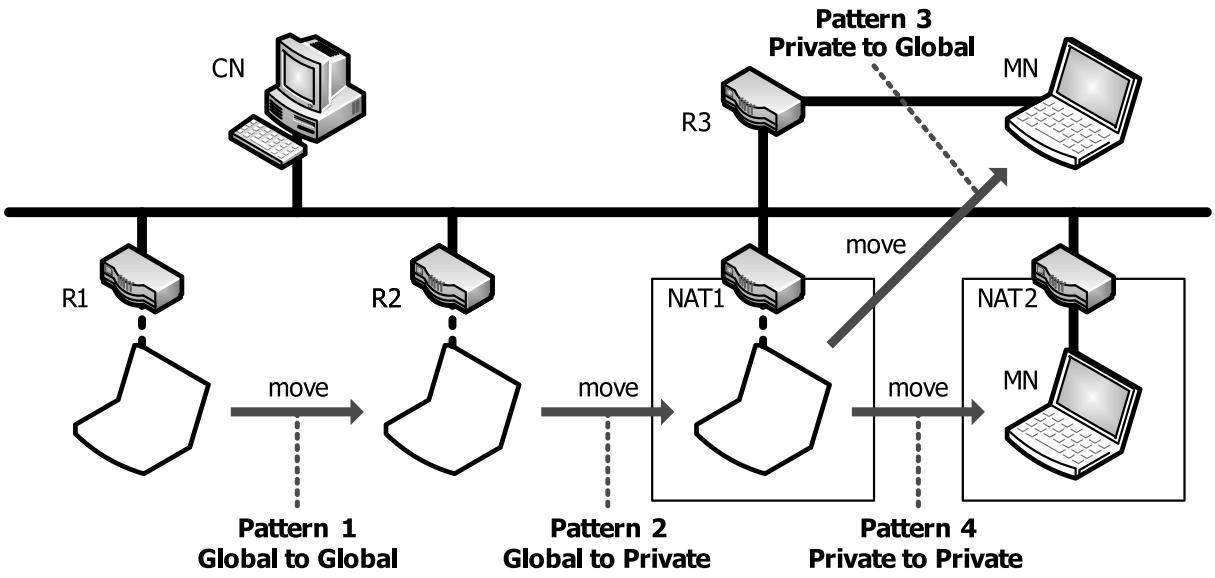

図 6.1　既存技術による移動パターン

6.2.2 Mobile IP による実現

図 6.2 に Mobile IP のシーケンスを示す．MN は移動しても変化しないホームアドレス HoA（Home Address）を送信元として，CN と直接通信を行う．MN が通信中に移動して DHCP [113] などにより新しい IP アドレスである共存気付アドレス CCoA（Co-located Care-of Address）が割り当てられると，HA（Home Agent）に対して BU（Binding Update）を送信する．HA は MN の HoA と CCoA の対応関係を Mobility Binding Table に保存する．以後，MN から CN への通信パケットは CN へ直接送信される．CN からの返信は HA が代理受信し，CCoA を用いた IP-in-IP カプセル化により MN へ転送される．

Pattern 2 を想定した技術として，Mobile IP Traversal of NAT [42] がある．MN はプライベートネットワークに移動してグローバルネットワーク上の HA に BU を送信する際，オプションとして UDP トンネルを要求する．BU 処理を終えた後，MN は CN 宛のパケットを UDP-in-IP による逆方向トンネルを形成して，HA へ送信する．HA はデカプセル化後，CN へ転送する．CN から MN への通信は逆の手順により，HA を経由して送信される．

Pattern 3 の移動パターンを実現する方法として，Reverse Tunneling for Mobile IP [91] を利用する方法がある．これはネットワークトポロジーの整合性を図り，Ingress Filtering 問題 [112] を解決するための手法である．HA の機能を NAT に搭載し，かつこの技術を応用することにより，MN の HoA にプライベート IP アドレスを割り当てることができる．MN は HoA を送信元として CN と通信を開始するが，NAT により HoA から HA のグローバル IP アドレスに変換される．MN が移動して BU 処理を終えた後，MN から CN への通信パケットは逆方向トンネリングにより HA を経由して送信される．

文献 [162] では Pattern 4 の移動パターンを想定した方式が提案されている．NAT に独自機能を実装した GRA（Global Roaming Agent）と呼ぶ装置を導入し，配下にプライベートネットワークの Mobile IP 網を構築する．Mobile IP 網内に設置された HA や FA（Foreign Agent）と連携することにより，GRA は MN がどの Mobile IP 網に接続しているかを管理したり，Mobile IP 網間のパケット転送を行う．

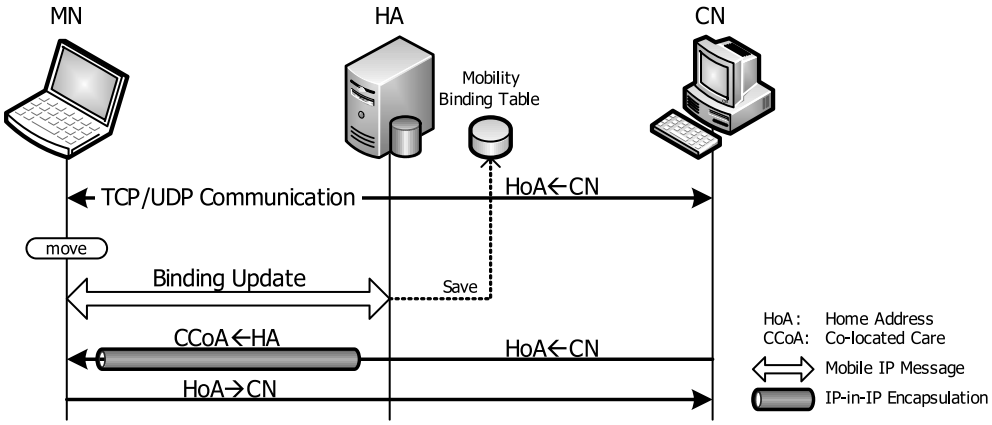

図 6.2　Mobile IP シーケンス

6.2.3 Mobile PPC による実現

Mobile PPC は HA のようなプロキシ装置を必要としないエンドツーエンド方式の移動透過性プロトコルである．図 6.3 に Mobile PPC の基本シーケンスを示す．本章では 4 章に示した Mobile PPC の基本シーケンスに，付録 E.1 で詳述する認証鍵共有シーケンスを含めて説明する．Mobile PPC では，通信の開始時は DDNS（Dynamic DNS） [111] により CN の IP アドレスを取得する．MN と CN は通信開始に先立ち，Cookie 交換及び Diffie-Hellman（以下 DH）鍵交換による 2 往復のネゴシエーションにより認証鍵を共有する [124]．更に通信パケットの通信識別子 CID（Connection ID）[2]を用いて，CIT（Connection ID Table）と呼ぶアドレス変換テーブルを IP 層に生成しておく．

MN が通信中に移動して IP アドレスが変化した際，CN に対して移動前後の IP アドレスの関係を CIT Update（以下 CU）処理により直接通知し合い，CIT を更新する．CU 処理では通信開始時に共有しておいた認証鍵による相手認証を行う．その後，両ノードは移動前に確立したコネクションに関する全ての TCP/UDP パケットに対して，更新した CIT に基づいたアドレス変換処理を行う．これにより，IP 層より上位では移動前の IP アドレスとして処理される．その結果，上位層から IP アドレスの変化を隠蔽することができる．

Mobile PPC では Pattern 2 から Pattern 4 に対応するため，NAT 越え技術として知られている Hole Punching [22, 23] の原理を導入する方法を提案している [128]．通信開始時の認証鍵共有処理または移動後の CU 処理において通信経路上に NAT の存在を確認すると，MN から CN に対して Hole Punching を実行し，NAT にマッピング情報を生成する．MN は CN からの応答により NAT の外側に割り当てられた IP アドレスとポート番号を取得する．これにより，CN は NAT のアドレス変換に対応した CIT を生成することができる．

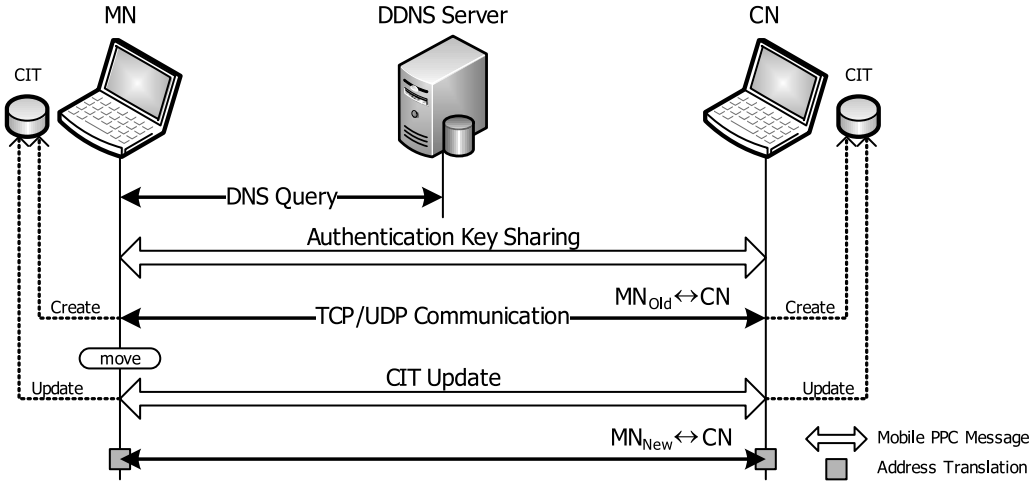

図 6.3　Mobile PPC シーケンス

[2]TCP コネクション，または UDP ストリームを識別するための情報であり，送信元/宛先 IP アドレス，ポート番号とプロトコルタイプの 5 つの値の組からなる．

表 6.1　IPv4 ネットワークにおける通信ケースの定義

MN の移動パターン	CN の位置	
	Global Network	Private Network
Pattern 1	Case 1	Case 5
Pattern 2	Case 2	Case 6
Pattern 3	Case 3	Case 7
Pattern 4	Case 4	Case 8

6.2.4　新たなニーズへの対応と通信ケースの定義

近年，外出先からホームネットワーク内の情報家電機器と通信するための研究が盛んに行われている [164, 165]．このような場合，既存技術の前提とは異なり，CN はプライベートネットワーク内に存在することになる．

そこで本章では MN の移動パターンに CN の位置を組み合わせた通信ケースを定義する．表 6.1 に IPv4 ネットワークにおける通信ケースを示す．既存技術は CN がグローバルネットワークに存在することを前提としているため，Case 1 から Case 4 に対応している．Case 5 から Case 8 のような新たな通信ケースを実現するためには，MN から CN に対する NAT 越えを実現する必要がある．

6.3　NAT-f と移動透過性プロトコルの融合

本章では CN 側の NAT 越え問題を解決するために，5 章で提案した NAT-f を移動透過性プロトコルと融合することにより，新たな通信ケースを実現する．NAT-f は通信開始ノードと NAT が連携することにより，NAT 配下のノードに対して通信を開始できる技術である．プライベートネットワークに存在する既存のノードをそのまま利用できるという利点があり，本研究が対象とするホームネットワークへの導入に適している．NAT-f は Mobile IP，Mobile PPC のどちらとも共存することが可能であるが，ここでは Mobile PPC を中心にその方法を述べる．NAT-f は通信開始時，Mobile PPC は移動時にアドレス変換テーブルを生成する．そのため，両者の技術には独立性があり，大きな修正を加えることなく組み合わせることができる．

6.3.1　システム構成

図 6.4 に本章において想定するシステム構成を示す．グローバル IP アドレス gIP_{MN} を持つ MN が，NAT-f ルータ配下のプライベートネットワークに存在する CN へ通信を開始する．以後，本章では NAT-f ルータを HGW（Home Gateway）と記載する．その後，MN は CN と通信中にルータ R1 配下のネットワークから R2 の配下ネットワークに移動して，新しいグローバル IP アドレス $gIP_{MN}{}^2$ を取得したことを想定する．R1 と R2 は DHCP サーバ機能を有していると仮定する．MN と HGW はそれぞれ NAT-f と Mobile PPC を実装しており，CN はこれら機能を有さない一般ノードでよい．

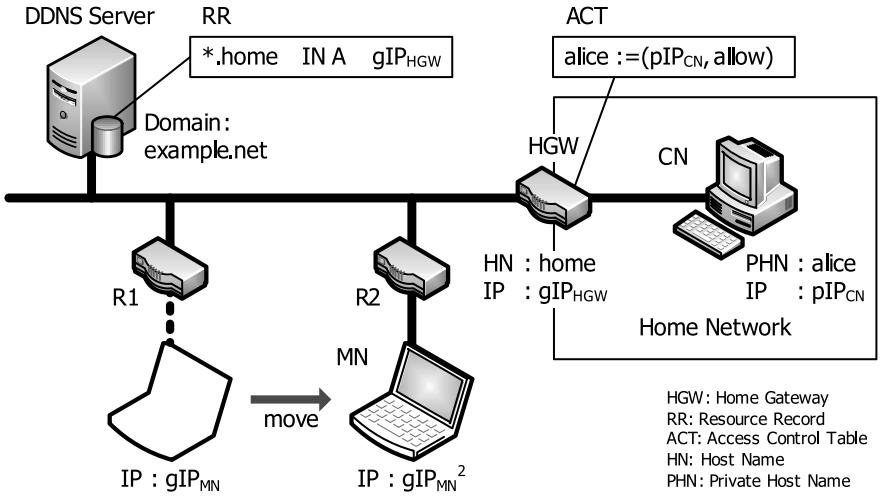

図 6.4　システム構成と事前設定

NAT-f を利用するための必要な事前設定として，DDNS サーバには HN_{HGW} と HGW のグローバル IP アドレス gIP_{HGW} の対応関係がワイルドカード A レコードとして，HGW には PHN_{CN} と CN のプライベート IP アドレス pIP_{CN} の対応関係がアクセス制御テーブル ACT（Access Control Table）に登録されているものとする．

6.3.2　NAT-f による通信開始手順

図 6.5 に通信開始時における NAT-f シーケンスを示す．以下に，MN が NAT-f により CN と通信を確立するまでの手順について述べる．

Step S-1: MN は CN へ通信を開始する際，$FQDN_{HGW}$ “*home.example.net*” の先頭に PHN_{CN} を付加した $FQDN_{CN}$ “*alice.home.example.net*” を用いて，DDNS サーバに名前解決の依頼を行う．DDNS サーバは DNS リソースレコードを検索し，$FQDN_{CN}$ に対応する IP アドレス gIP_{HGW} を応答する．

Step S-2: MN は受信した DNS A Reply を IP 層に一時待避させてから，DNS A Reply に記載されている IP アドレス gIP_{HGW} 宛に Support Check Request メッセージを送信する．ここで，Support Check ネゴシエーションは通信相手[3]が GE（GSCIP Element）であるかを確認するために，新たに追加したシーケンスである．Support Check Request を受信した GE（本章では MN と HGW が該当）は自身が NAT-f や Mobile PPC に対応しているか否かを応答するために，Support Check Response を返信する．

MN は Support Check Response を確認し，宛先が NAT-f 対応 NAT ルータであれば待避させた DNS A Reply に記載された IP アドレスを仮想 IP アドレス vIP_{CN} に書き換え，アプリケーションには CN の IP アドレスを vIP_{CN} として通知する．ここで仮想 IP アドレスとは外部ノー

[3]DNS の名前解決により取得した IP アドレスを持つノードを指す．

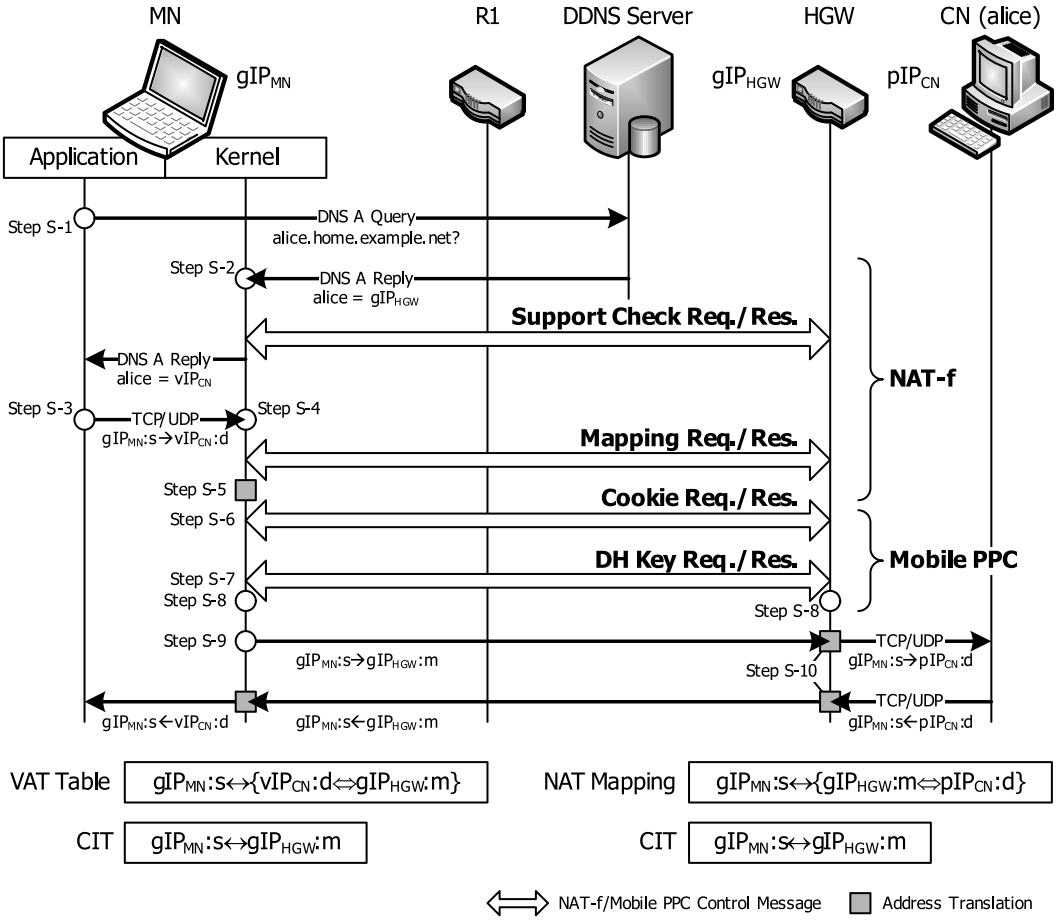

図 6.5　MN が通信を開始する時の NAT-f シーケンス

ド（MN）が HGW 配下の内部ノード（CN）を一意に特定するために割り当てるアドレスであり，$FQDN_{CN}$ から生成する．これにより MN の IP 層では，送信パケットの宛先仮想 IP アドレスから HGW 配下のどのノードと通信したいのかを判断することが可能になり，CN との通信に必要な NAT マッピング情報を HGW に生成させることができる．仮想 IP アドレスを導入することにより，HGW 配下の複数のノードと同時に通信することが可能となる．

なお，宛先が NAT-f 対応 NAT ルータではない場合[4]，待避させた DNS 応答をそのまま上位層へ渡し，取得した IP アドレス gIP_{HGW} をそのままアプリケーションへ通知する．以後は NAT-f に関する処理は一切行われず，通常通り通信を開始する．この場合，CN がグローバルネットワークに存在する場合が想定される．

Step S-3: MN のアプリケーションは仮想 IP アドレスを宛先として送信処理を行い，下記パケットが IP 層に渡される．

$$gIP_{MN} : s \rightarrow vIP_{CN} : d \quad [proto] \tag{6.1}$$

[4]Support Check Request は GSCIP 関連プロトコルの制御メッセージと同様に ICMP Echo Request の上で定義されているため，宛先ノードまたは宛先 NAT が GE ではない，すなわち NAT-f に対応していない場合は Support Check Request とメッセージデータが同じ ICMP Echo Reply が応答される．MN は Support Check Request の応答の違いにより，宛先が NAT-f 対応 NAT ルータか否かを判断する．

MN は IP 層において，送信パケットの宛先が仮想 IP アドレスの場合，パケットの CID（gIP_{MN}，s，vIP_{CN}，d，$proto$）を用いて VAT（Virtual Address Translation）テーブルと呼ぶアドレス変換テーブルを参照する．VAT テーブルとは，仮想 IP アドレスと HGW における NAT のマッピングアドレスとの変換関係を示すテーブルで，以後の NAT-f マッピング処理完了時に生成される．MN が CN に初めて通信する場合は VAT テーブルに該当するエントリがないため，マッピング処理を開始する．

ここで，送信パケットの宛先が仮想 IP アドレスでない場合は，VAT テーブルの参照や以下のマッピング処理など NAT-f 一連の処理を行わない．Support Check Response により CN が Mobile PPC に対応していることがわかったら Step S-6 の Mobile PPC 認証鍵共有処理へ移行し，Mobile PPC に対応していない場合は Step S-9 の処理へ移行して通信を開始する．

Step S-4: MN は送信しようとしていた TCP/UDP パケットをカーネル内に一時待避し，Mapping Request を HGW へ送信する．Mapping Request には待避したパケットの CID と，仮想 IP アドレスに対応する PHN_{CN} が記載される．HGW は Mapping Request を受信すると，通知された PHN_{CN} をキーとして ACT を検索しに対応する IP アドレスを取得する．その後，通知された CID と ACT から取得した CN の IP アドレス pIP_{CN} から以下に示す NAT マッピング情報を生成する．

$$\text{HGW:} \qquad gIP_{MN} : s \leftrightarrow \{gIP_{HGW} : m \overset{NAT}{\Longleftrightarrow} pIP_{CN} : d\} \quad [proto] \tag{6.2}$$

これは CN のポート番号 d と NAT のポート番号 m がマッピングされたことを示しており，CN から MN へ通信を開始した場合に HGW で生成される NAT マッピング情報と同様のものである．HGW は $gIP_{HGW} : m$ をマッピングアドレスとして Mapping Response に記載して MN へ応答する．

MN は Mapping Response を受信すると，仮想 IP アドレスとマッピングアドレスの変換関係を示すエントリ

$$\text{MN:} \qquad gIP_{MN} : s \leftrightarrow \{vIP_{CN} : d \overset{VAT}{\Longleftrightarrow} gIP_{HGW} : m\} \quad [proto] \tag{6.3}$$

を生成し，VAT テーブルに格納する．その後，先ほど待避した TCP/UDP パケットを復帰させ，マッピング処理を完了する．

Step S-5: 復帰した TCP/UDP パケットは式 (6.3) に示す VAT テーブルエントリに基づいて，宛先 IP アドレス・ポート番号が $vIP_{CN} : d$ から $gIP_{HGW} : m$ に変換される．ここで，提案方式では Mobile PPC と組み合わせるため，Mobile PPC の通信開始時の処理，すなわち認証鍵共有処理を行う．認証鍵共有処理は通常の Mobile PPC と同じだが，VAT テーブルに基づいて変換された通信パケットをトリガとして実行する点が異なる．

Step S-6: MN はアドレス変換された TCP/UDP パケットを再度カーネル内に待避してから，HGW へ Cookie Request メッセージを送信して Cookie 交換を行う．MN は HGW からの Cookie Response を受信後，待避していたパケットを復帰させる．以上の処理を終えると，MN は

TCP/UDP 通信を開始し，Step S-9 の処理へ移行すると同時に，そのバックエンドで Step S-7 の DH 鍵交換処理を実行する．

Step S-7: MN は DH 秘密鍵及び DH 公開鍵を生成し，DH Key Request メッセージにより DH 公開鍵を DH Key Request メッセージにより HGW へ送信する．HGW も同様に DH 秘密鍵及び DH 公開鍵を生成後，DH Key Response により DH 公開鍵を応答する．

Step S-8: MN と HGW は DH 鍵交換後，自身の DH 秘密鍵と相手の DH 公開鍵から認証鍵を生成する．

Step S-9: MN は Step S-5 において復帰した TCP/UDP パケットから移動前の CID 情報として下記のような CIT エントリを生成する．

$$\text{MN:} \qquad gIP_{MN} : s \leftrightarrow gIP_{HGW} : m \quad [proto] \tag{6.4}$$

上記エントリを CIT に登録後，復帰したパケットを HGW へ送信する．

Step S-10: HGW は MN からのパケットを受信後，MN と同様に受信パケットの CID から以下に示す CIT エントリを生成する．

$$\text{HGW:} \qquad gIP_{MN} : s \leftrightarrow gIP_{HGW} : m \quad [proto] \tag{6.5}$$

その後，Step S-4 で生成した NAT マッピング情報に基づいて，式 (6.3) のように当該パケットの宛先 IP アドレス・ポート番号を $gIP_{HGW} : m$ から $pIP_{CN} : d$ に変換し，CN へ転送する．

以上の処理により，MN から CN への通信開始が完了する．ここまでのアドレス変換処理による TCP/UDP パケットの IP アドレス及びポート番号の遷移を図 6.6 にまとめる．CN から MN への応答パケットは上記と逆の変換処理を行う．すなわち，CN から MN への応答パケット

$$pIP_{CN} : d \longrightarrow gIP_{MN} : s \quad [proto] \tag{6.6}$$

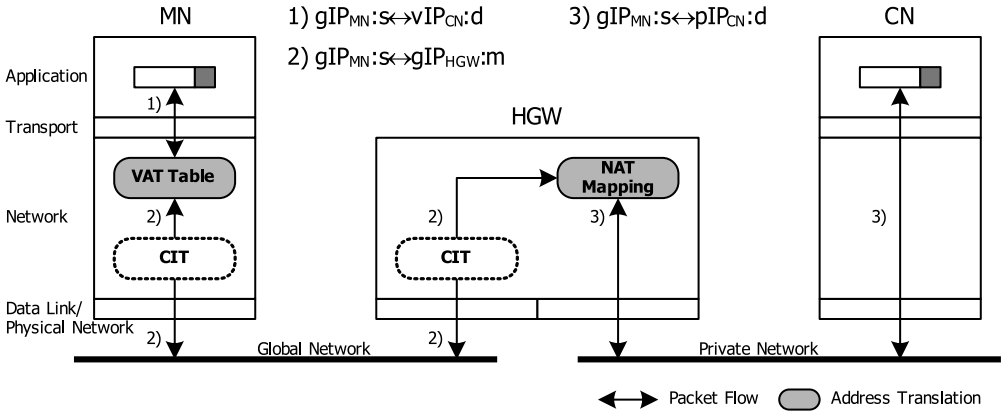

図 6.6　MN 移動前における IP アドレス/ポート番号の遷移

は HGW の NAT マッピングに基づいて，送信元が $gIP_{HGW} : m$ に変換される．その後，MN の IP 層で VAT テーブルに基づいて送信元が $vIP_{CN} : d$ に変換される．MN が移動前であるため，CIT に基づくアドレス変換は行われない．以後，上記アドレス変換処理が TCP/UDP パケット送受信ごとに繰り返し実行される．

6.3.3 NAT-f と Mobile PPC の融合による通信継続手順

図 6.7 に MN が R2 配下へ移動した後の通信シーケンスを示す．MN は CN と通信中に移動した場合は，既存の Mobile PPC による CU 処理を実行した後，VAT テーブル，NAT マッピング及び CIT に基づく 3 種類のアドレス変換を同時に実行する．以下に，MN と CN が通信を継続するまでの手順について述べる．

Step M-1: MN は別のネットワークに移動したことを検知すると，DHCP 処理を実行して新しい IP アドレス $gIP_{MN}{}^2$ を取得する．

Step M-2: アドレス取得後，MN は HGW に対して CU 処理を行う．HGW に送信する CU Request には移動前 IP アドレス gIP_{MN} と移動後 IP アドレス $gIP_{MN}{}^2$ が記載され，通信開始時に共有した認証鍵を用いて署名を付加する．CU Request を受信した HGW は認証処理を終えた後，式 (6.5) に示す CIT エントリを

$$\text{HGW:} \qquad \{gIP_{MN} : s \overset{CIT}{\Longleftrightarrow} gIP_{MN}{}^2 : s\} \leftrightarrow gIP_{HGW} : m \quad [proto] \tag{6.7}$$

のように更新してから，CU Response を応答する．

MN は CU Response を受信したら，HGW と同様に式 (6.4) に示す自身の CIT エントリを更

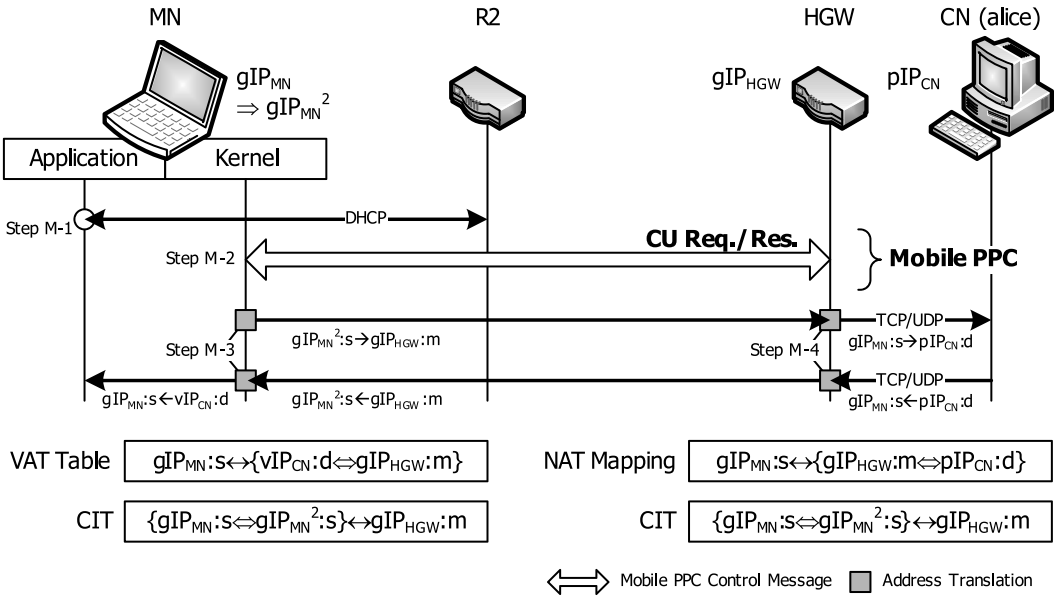

図 6.7　MN 移動後における Mobile PPC シーケンス

新する．

$$\text{MN:} \qquad \{gIP_{MN} : s \overset{CIT}{\Longleftrightarrow} gIP_{MN}{}^{2} : s\} \leftrightarrow gIP_{HGW} : m \quad [proto] \tag{6.8}$$

以上により，CU 処理は完了する．

Step M-3: 上位層から渡された TCP/UDP パケット

$$gIP_{MN} : s \rightarrow vIP_{CN} : d \quad [proto] \tag{6.9}$$

は，式 (6.3) に示す VAT テーブルエントリに基づくアドレス変換，及び式 (6.8) に示す CIT エントリに基づくアドレス変換が行われる．すなわち，上記パケットは送信元が移動前から移動後の IP アドレスへ，宛先が仮想 IP アドレスからマッピングアドレスへ変換され，最終的に

$$gIP_{MN}{}^{2} : s \rightarrow gIP_{HGW} : m \quad [proto] \tag{6.10}$$

として HGW へ送信される．

Step M-4: HGW は受信パケットに対して，式 (6.7) の CIT エントリに基づくアドレス変換，及び式 (6.2) の NAT マッピング情報に基づくアドレス変換が行われる．すなわち，上記パケットは送信元が MN の移動後から移動前の IP アドレスへ，宛先がマッピングアドレスから CN のプライベート IP アドレスへ変換され，最終的に

$$gIP_{MN} : s \rightarrow pIP_{CN} : d \quad [proto] \tag{6.11}$$

として CN へ転送される．

以上の処理により，MN の上位アプリケーション，HGW の NAT アドレス変換処理部および CN は，移動が発生して MN の IP アドレスが変化したことに気づくことなく，通信を継続することができる．ここまでのアドレス変換処理による TCP/UDP パケットの IP アドレス及びポート番号の遷移を図 6.8 にまとめる．なお，CN から MN への通信は通信開始時と同様に上記と逆の手順で

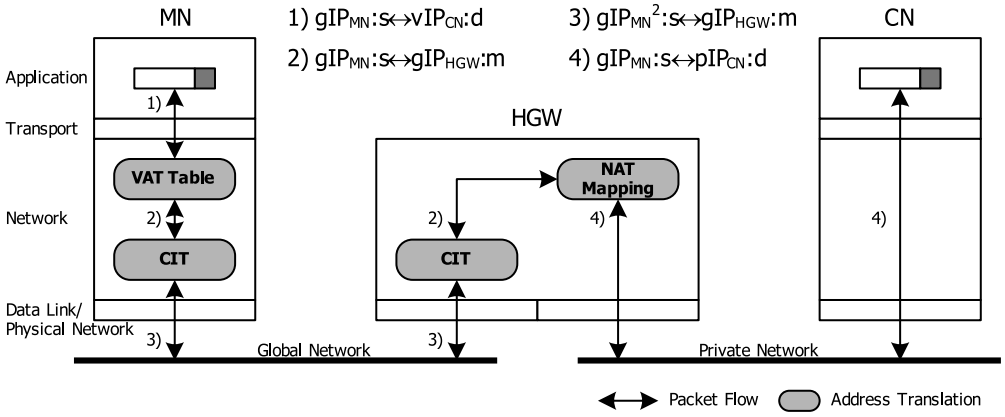

図 6.8　MN 移動後における IP アドレス/ポート番号の遷移

アドレス変換を行う．すなわち，HGW の NAT マッピング情報に基づいて送信元が $pIP_{CN} : d$ から $gIP_{HGW} : m$ に，さらに CIT に基づいて宛先が gIP_{MN} から ${gIP_{MN}}^2$ に変換される．その後，MN の IP 層で CIT に基づいて宛先が ${gIP_{MN}}^2$ から gIP_{MN} に，VAT テーブルに基づいて送信元が $gIP_{HGW} : m$ から $vIP_{CN} : d$ に変換される．

6.4 実装

NAT-f と Mobile PPC を組み合わせた方式を確認するために，FreeBSD 6.1-RELEASE を用いてプロトタイプシステムを実装した．以下に MN 側と HGW 側の実装概要をそれぞれ示す．

6.4.1 MN の実装概要と移動検知処理

図 6.9 に MN おけるカーネルモジュールの実装を示す．NAT-f モジュール及び Mobile PPC モジュールは IP 層に実装され，IP 入出力関数 `ip_input()`，`ip_output()` から呼び出される．特にパケット送信時は `ip_output()` においてルーチング処理が行われるが，両モジュールはそれ以前に呼び出されて所定の動作を行う．

6.3 節に示した処理手順を可能とするため，NAT-f モジュールの呼び出しインタフェースを Mobile PPC 呼び出しインタフェースより上位になるように変更した．このような変更は両モジュールが `ip_input()`, `ip_output()` から独立した実装となっているため, 容易に実現可能である．NAT-f と Mobile PPC の主処理は従来のまま利用するが，融合するに当たり以下の機能追加・修正を施し

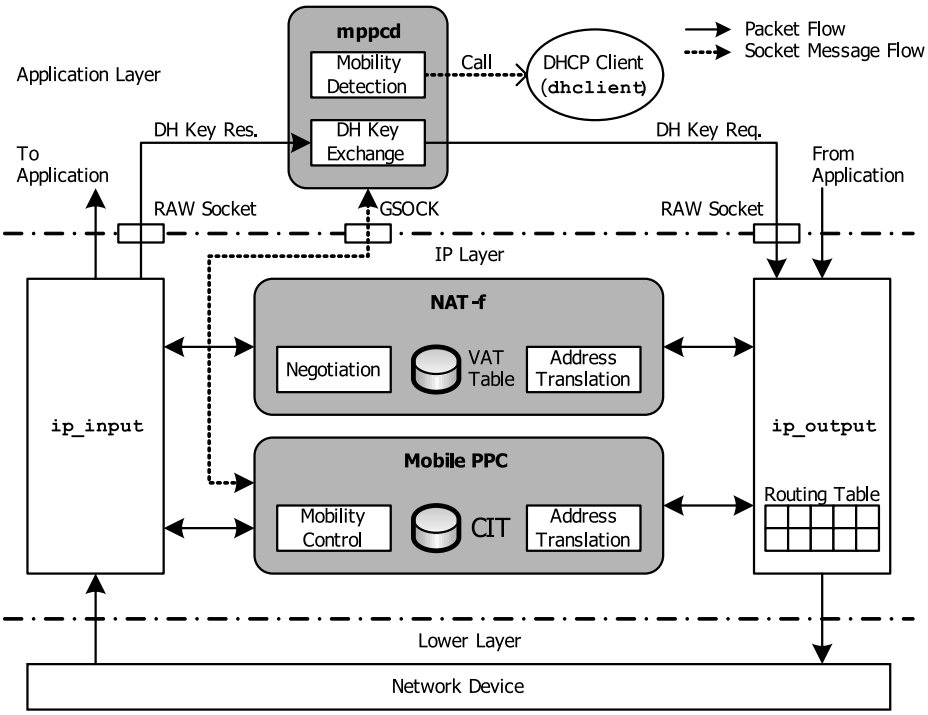

図 6.9　MN におけるカーネルモジュールの実装

た．NAT-f モジュールのネゴシエーション処理に Mobile PPC 対応フラグを設定する機能を追加した．本章のように両モジュールが融合されている場合は Support Check Response にフラグを設定し，NAT-f モジュールの処理を完了したら Mobile PPC モジュールが呼び出される．Mobile PPC モジュールが実装されていない場合はフラグが設定されていないため，NAT-f 単体の処理だけが実行される．なお，MN の IP レベルにおける通信相手が NAT-f に対応していない場合は 6.3.2 項（Step S-2）の DNS 書き換え処理を行わないため，以後の NAT-f 処理は実行されない．このような工夫により，NAT-f と Mobile PPC を融合した場合だけでなく，NAT-f，Mobile PPC 単体の機能でも動作するようにした．

Mobile PPC の DH 鍵交換処理はユーザランドで動作するデーモン `mppcd` が行う．Cookie Response を受信した MN は，Mobile PPC カーネルモジュールから GSOCK（GPACK Socket）[5]を通じて `mppcd` に DH 鍵交換処理を指示する．`mppcd` は RAW socket を用いて DH Key Request/Response を送受信する．認証鍵を生成後，GSOCK を通じて Mobile PPC カーネルモジュールに登録する．なお，DH 鍵交換処理は OpenSSL [73] を利用し，RFC3526 [166] と RFC4306 [77] で定義されている DH グループ 1，2，5 及び 14 の素数と原始根がシステム共通のパラメータとして実装した[6]．

IPv4 ネットワークでは，IPv6 ルータが定期的に送信する RA（Router Advertisement）のような仕組みがないため，MN は移動を検知する手段がない．Mobile IPv4 では FA からのエージェント広告により移動の検知は可能であるが，Mobile PPC では FA のような装置を想定していないため，MN が自律的に解決する手段が必要となる．また，4 章 の 図 4.10 に示した従来の Mobile PPC は，IP アドレス取得後に行うアドレス重複確認の終了と同時にカーネルの ARP 関数から CU 処理を開始する仕組みであった．そこで IP 入出力関数以外のカーネル関数を変更することなく，かつ移動を自律的に検知して CU 処理をカーネルモジュールに指示する移動検知モジュールを `mppcd` に実装した．

図 6.10 に移動検知処理の仕組みを示す．`mppcd` は定期的にネットワークデバイスのリンク状

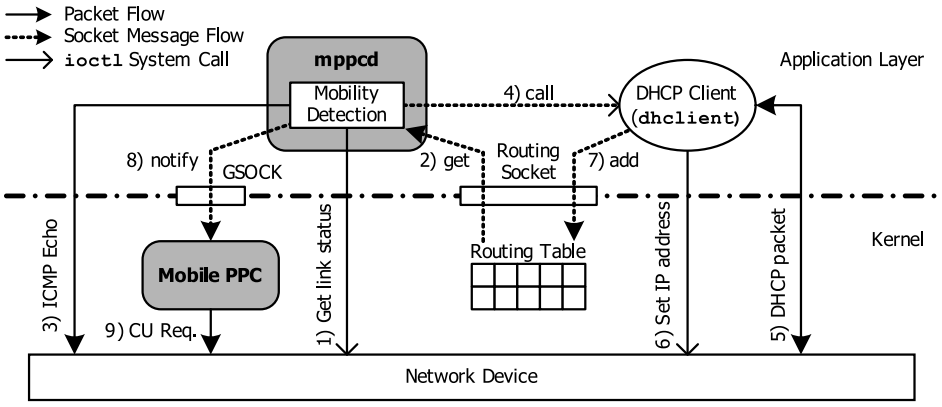

図 6.10　移動検知処理の仕組み

[5]ユーザランドと GPACK カーネルモジュール間のデータ受け渡しを実現するために実装したソケットインタフェース．

[6]素数のサイズは順に 768，1024，1536，2048 bit であり，原始根は 2 である．

態を監視し，ネットワークに接続したと判断したらルーチグテーブルから取得したゲートウェイの IP アドレスを用いて `ping` を実行する．一定時間内に応答を受信できなかった場合，異なるネットワークに接続したと判断し，`dhclient` [7]を実行する．以上の処理が完了したら，Mobile PPC カーネルモジュールに対して CU 処理を指示する．Mobile PPC カーネルモジュールは `mppcd` からの指示を受けると，CU Request を生成して該当する CN へ送信する．これにより，Mobile PPC モジュールのカーネル依存度を少なくし，かつ自律的に移動検知から CU 処理を実行することを可能とした．

6.4.2 natd の拡張

図 6.11 に NAT-f と Mobile PPC に対応した HGW のモジュール実装の概要を示す．これまで NAT-f 及び Mobile PPC の機能はカーネルに実装していたが，今回，アプリケーションレベルで動作する `natd` [8]を拡張することにより実現した．

`natd` は Divert socket を経由して送受信パケットのアドレス変換を行う．受信パケットが NAT-f または Mobile PPC に関するメッセージであれば，各モジュールに処理を移す．Mapping Request の場合，`natd` のマッピングを行う処理を呼び出し，マッピング情報を生成後，MN 宛の Mapping Response を Raw socket 経由で送信する．Mobile PPC に関するメッセージの場合は，Mobile PPC モジュールを呼び出して処理を行い，NAT-f と同様に各 Response メッセージを RAW socket 経由で送信する．

なお，NAT のマッピング情報と CIT を統合することも可能だが，今後の各プロトコルの拡張を考慮し，あえて各モジュールの独立性を確保した．この方法では HGW は MN 移動後の通信パケットに対して，CIT と NAT によるアドレス変換を 2 回行うことになる．

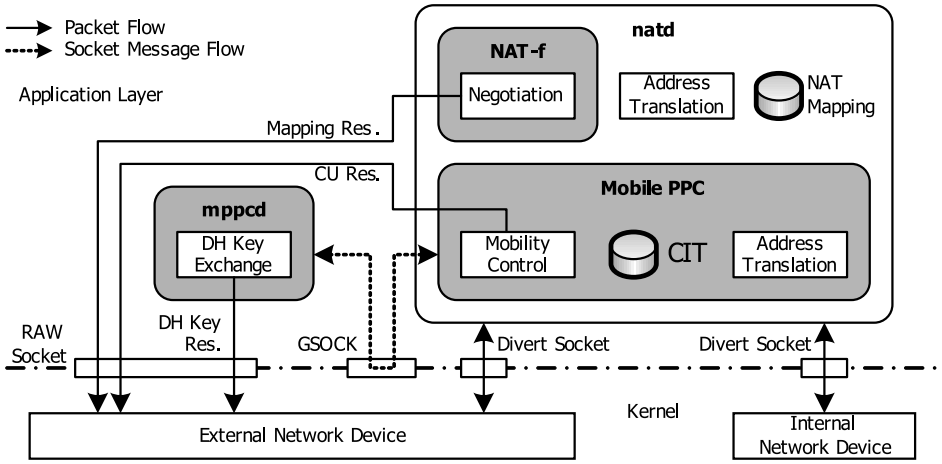

図 6.11　natd の拡張によるモジュールの実装

[7]FreeBSD に搭載されている DHCP クライアント．IP アドレス，ゲートウェイ情報の設定から二重アドレスチェックを行う．

[8]FreeBSD に搭載されている NAT アプリケーション．

表 6.2　装置仕様

	MN	CN	HGW
CPU	Pentium M	Core2 U7600	Geode LX800
	1.73 GHz	1.20 GHz	500 MHz
Memory	512 MByte	2037 MByte	256 MByte
NIC	100Base-TX	100Base-TX	100Base-TX
OS	FreeBSD 6.1	Windows Vista	FreeBSD 6.1

6.4.3　アドレス変換に伴うチェックサム

VAT テーブル及び CIT に基づくアドレス変換処理は，IP アドレスとポート番号のみを変換する．IP ヘッダ及び TCP/UDP ヘッダのチェックサムは差分計算により修正する．これは NAT のアドレス変換と原理は同様であり，RFC3022 [21] に準じた演算を行う．

6.5　評価と考察

MN と HGW で行われるアドレス変換処理が，MN と CN のエンドツーエンドのスループットに与える影響を明らかにするために，Iperf [167] を用いて TCP/UDP スループットを測定した．また，通信開始時に発生するオーバヘッド及び `mppcd` による移動検知から通信継続までに要する時間，すなわち通信断絶時間を測定した．

測定環境は図 6.4 に示す構成とし，HGW，DDNS サーバ及び R1/R2 をスイッチで接続した．表 6.2 に各装置の仕様を示す．MN の移動は UTP ケーブルを R1 から R2 につなぎなおすことでエミュレートした．Cookie と認証鍵の生成に用いるハッシュ関数には MD5 を使用し，DH 鍵交換における DH グループは Group 1 とした．

6.5.1　スループット性能

Iperf により MN から CN に対して TCP トラフィックを 60 秒間送信した．NAT-f と Mobile PPC を実装したシステムにおいて，移動前と移動後のスループットを測定した．また比較のため，同一装置により NAT-f と Mobile PPC を実装していない通常のシステムにおいても測定した．この場合は HGW に予め静的マッピングを設定し，MN が CN へ通信を開始できるようにした．

表 6.3 に TCP スループット測定結果を示す．未実装時のスループットが 69.3 Mbit/s であったのに対して，実装時の移動前は 69.1 Mbit/s，移動後は 67.9 Mbit/s であった．実装時の移動前は MN において VAT テーブルに基づくアドレス変換処理が加わるが，スループットに対する影響はほとんどないといえる．実装時の移動後は，更に MN と HGW において CIT に基づくアドレス変換が加わるため，未実装時のスループットから約 2 % 低下していた．低下の要因は HGW における CIT のアドレス変換処理にあることがわかった．

表 6.3　Iperf による TCP スループット測定値

	スループット
NAT-f/Mobile PPC 未実装時	69.3 [Mbit/s]
NAT-f/Mobile PPC 実装時（移動前）	69.1 [Mbit/s]
NAT-f/Mobile PPC 実装時（移動後）	67.9 [Mbit/s]

表 6.4　MN の通信開始時に発生する処理時間の内訳

処理内容	該当 Step	処理時間
a) DNS 応答書き換え	Step S-2	2.74 [msec]$^{\star 1}$
b) マッピング処理	Step S-3〜Step S-5	5.49 [msec]$^{\star 2}$
c) Cookie 交換	Step S-6	3.28 [msec]$^{\star 2}$
d) DH 鍵交換	Step S-7	98.72 [msec]$^{\star 2}$
e) 認証鍵生成（MN）	Step S-8	5.40 [msec]
f) 認証鍵生成（CN）	Step S-8	38.92 [msec]
通信開始までの総オーバヘッド（a+b+c）		11.51 [msec]

$^{\star 1}$ 処理時間 +1RTT（RTT は MN〜DDNS 間の RTT）
$^{\star 2}$ 処理時間 +1RTT（RTT は MN〜HGW 間の RTT）

なお，測定で使用した装置は 100BASE-TX の Ethernet で接続していたが，提案方式の実装の有無に関わらず 70 Mbit/s 程度のスループットしか得られなかった．この原因は NAT アプリケーションに `natd` を採用したためである．図 6.11 に示すように，`natd` はユーザランドで動作し，Divert socket により受信したパケットを IP 層から取得する．このとき，カーネルではフラグメントされたパケットを再構築する処理が発生する．また，ソケット層ではカーネルとユーザランド間のメモリコピーがパケット毎に発生する．これらの処理によって，`natd` はスループットが大幅に低下してしまう[9]．

以上の結果より，NAT-f と Mobile PPC を組み合わせても，スループットの低下は十分に小さく，実用上問題ないといえる．

6.5.2　通信開始時のオーバヘッド

表 6.4 に通信開始時に発生するオーバヘッドとその内訳を示す[10]．MN が最初の TCP/UDP パケットを送信する際，カーネルに一時待避させてから実際に送信されるまでに行われる処理は，表 6.4 のうち NAT-f による DNS 応答書き換え，マッピング処理，及び Mobile PPC の Cookie 交換の合計処理である．従って，通信開始までのオーバヘッドは上記処理時間の合計，すなわち 11.51 msec

[9] HGW の `natd` を無効にしてルータとして動作させた場合，MN と CN 間の TCP スループットは 92.2 Mbit/s であった．

[10] 表 6.4 および表 6.5 における数値は実験環境における小さな RTT（Round Trip Time）値によるものである．実環境における RTT の値についてはたとえば文献 [79, 80] を参照のこと．

表 6.5　通信断絶時間の内訳

処理内容	該当 Step	処理時間	
ネットワーク移動	Step M-1（手動）	1.64	[sec]
移動検知	Step M-1 `mppcd`	28.70	[msec]
アドレス取得	Step M-1（`dhclient`）	2.11	[sec]*1
アドレス重複確認	Step M-1（Kernel）	0.69	[sec]
CU 処理	Step M-2（Mobile PPC）	41.26	[msec]*2
総通信断絶時間		4.51	[sec]

*1 処理時間 +2RTT（RTT は MN〜R2 間の RTT）
*2 処理時間 +1RTT（RTT は MN〜HGW 間の RTT）

となる．認証鍵共有処理の後半部分（DH 鍵交換と認証鍵生成）は TCP/UDP 通信のバックエンドで行われるため，通信開始時のオーバヘッドには含まれない．

上記結果より，6.6.1 項に示した NAT-f と Mobile IP を組み合わせたシステムにおいても，通信開始時に発生するオーバヘッドは十分許容できる範囲であるといえる．

6.5.3　通信断絶時間

表 6.5 に移動時の通信断絶時間とその内訳を示す．表中の処理内容はそれぞれ下記の間の処理である．

- ネットワーク移動：UTP ケーブル抜線〜挿線
- 移動検知：リンク確立判断〜ping タイムアウト
- IP アドレス取得：DHCP Discover 送信〜IP アドレス設定
- アドレス重複確認：Gratuitous ARP 送信〜タイムアウト
- CU 処理：CU Request 送信〜CIT 更新

通信断絶時間の合計は 4.51 sec であった．このうち，ネットワークの移動に 1.64 sec を要しているが，実際は無線 LAN における L2 ハンドオーバに該当するため，50〜400 msec になると推測される [117]．上記時間を除いた通信断絶時間に着目すると，DHCP によるアドレス取得と Gratuitous ARP によるアドレス重複確認の合計が 97.6 % を占める結果となった．一方，`mppcd` やカーネルモジュールによる CU 処理，すなわち提案方式特有の処理時間は十分に短いことがわかる．

上記の結果より，移動に伴うパケットロスを減らすためには，アドレス取得に関する処理時間を抑えることが課題となる．この課題については文献 [134] において別途検討済みである．

6.5.4　セキュリティに関する考察

攻撃者は Mapping Request や CU Request に記載されている MN の送信元 IP アドレスを改ざんすることにより，セッションのハイジャックを試みることが考えられる．Mobile PPC では MN と

HGW は通信開始時に認証鍵を共有しているため，CU Request/Response に署名を付加することにより，メッセージ完全性を保証できる．Mobile PPC の安全性は上記認証鍵の共有方法に依存する．本論文では自宅や友人などある特定のネットワークに対してアクセスすることを想定としているため，MN と HGW との間で事前に秘密鍵を共有することが可能である．したがって，MN と HGW は認証を伴った認証鍵共有を実行することが可能であり，中間者攻撃を防止できる．NAT-f の Mapping 処理においても MN と HGW が事前共有鍵を保持することにより，Mapping Request/Response の暗号化や認証を行うことが可能である．

事前共有鍵に基づくシステムは導入が比較的容易であるが，鍵の管理が煩雑になったり，不特定の相手と鍵を共有することが困難であるなどの課題がある．提案方式の適用対象を不特定なネットワークにまで拡張するには，PKI（Public Key Infrastructure）によるディジタル署名認証や公開鍵認証などの手法を利用する必要がある．

6.5.5 各通信パターンへの対応

移動透過アーキテクチャの実用性を評価する上で，対応可能な通信ケースの広さは重要な指標と考えられる．本章では 6.2.4 項に示した Case 5 の実現方法を取り上げたことになる．提案方式が他の通信ケースを実現する可能性について考察した．

Mobile PPC では 6.2.3 項で述べたように，hole punching 処理を導入することにより Case 2 から Case 4 を実現できる手法を提案済みである．提案方式は移動透過性プロトコルの機能をそのまま利用しているため，そのまま Case 2 から Case 4 に対応することができる．

Case 6 はグローバルネットワークからプライベートネットワークへの移動のため，Case 5 とは移動後の処理だけが異なる．移動後の処理に着目すると，Mobile PPC に関する処理しか行わない．従って，Case 2 と同様の処理を行うことにより Case 6 は実現可能である．

Case 7 については，文献 [128] の hole punching の手法を NAT-f のマッピング処理に対して導入することにより，実現可能であると考えられる．

Case 8 は Case 4 の実現方法と同じ考え方で対応することが可能である．すなわち，Case 7 の通信開始時の処理と Case 6 の移動後の処理を組み合わせることで実現できる．

6.6 他システムへの応用

これまでは，NAT-f と Mobile PPC の融合について述べてきた．これらは本論文で提案するグループ通信アーキテクチャ GSCIP（Grouping for Secure Communication for IP）を構成するプロトコルとして位置づけられている．本節では，提案アーキテクチャ GSCIP は他の様々なシステムに応用可能であること示す．特に移動透過性に関するシステムに着目して議論する．

6.6.1 Mobile IP への応用

図 6.12 に NAT-f と Mobile IP を組み合わせたシーケンスを示す．MN の HoA を HoA_{MN}，移動後の CCoA を $CCoA_{MN}$ とし，MN と HGW に NAT-f 機能が実装されているものとする．MN は通常の NAT-f の手順により，下記の VAT テーブル及び NAT マッピングを生成後，VAT テーブルに基づくアドレス変換を行い CN への通信を開始する．

$$\text{MN:} \quad HoA_{MN} : s \leftrightarrow \{vIP_{CN} : d \overset{VAT}{\Longleftrightarrow} gIP_{HGW} : m\} \quad [proto] \tag{6.12}$$

$$\text{HGW:} \quad HoA_{MN} : s \leftrightarrow \{gIP_{HGW} : m \overset{NAT}{\Longleftrightarrow} pIP_{CN} : d\} \quad [proto] \tag{6.13}$$

HGW は上記マッピング情報に従ってアドレス変換を実行し，MN からの通信パケットを CN へ転送する．

MN が CN と通信中に別のネットワークに移動すると，通常の Mobile IP の手順により HA と BU 処理を行い，Mobility Binding Table を登録する．MN から CN への通信パケットは，送信元 IP アドレスが常に HoA であるため，これを受信した HGW は MN 移動前に生成された NAT マッピングの情報に従って CN へ転送することができる．CN からの応答パケットは宛先 IP アドレスが HoA と

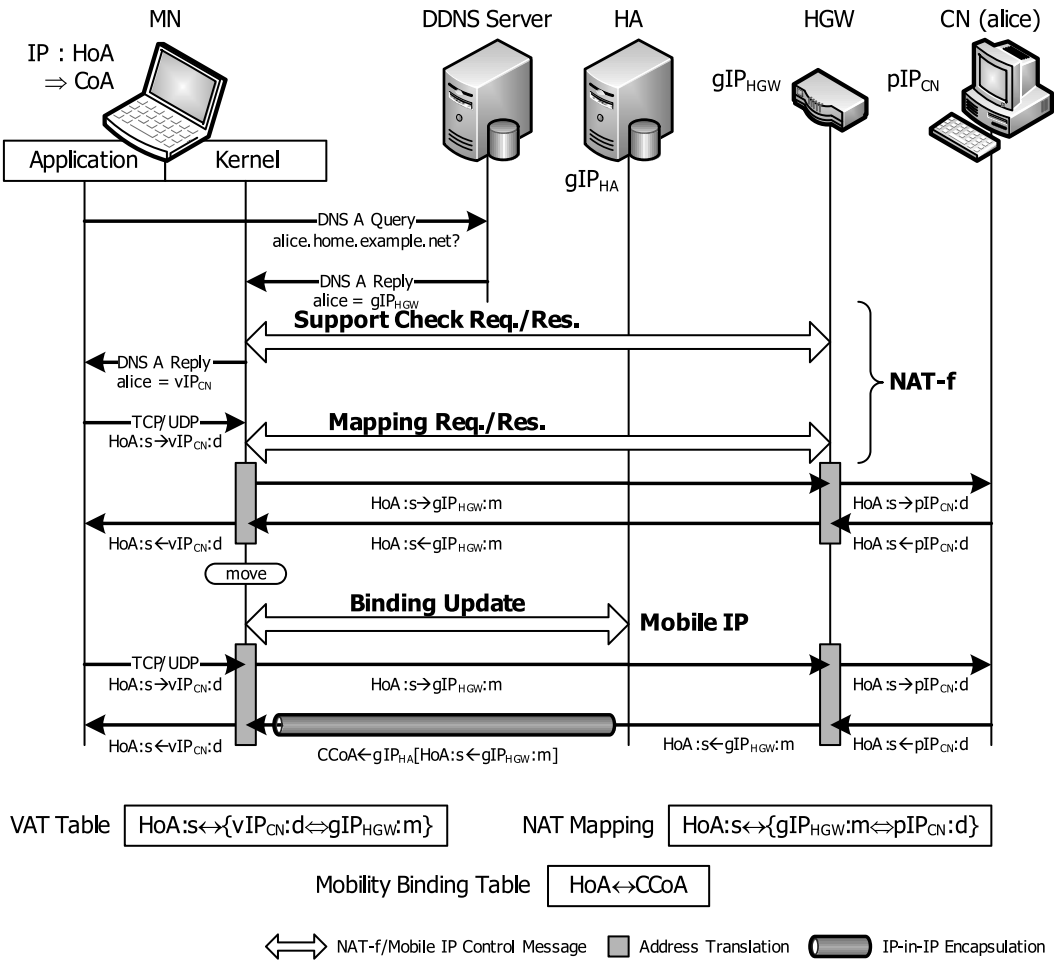

図 6.12　Mobile IP と NAT-f を組み合わせたシーケンス

なるため，HA が代理受信する．その後，HA は Mobility Binding Table の情報に基づいて，受信パケットを IP-in-IP カプセル化してから MN の CCoA 宛へ転送する．上記パケットを受信後，MN はデカプセル化してから VAT テーブルに基づいて送信元を HGW のマッピングアドレス $gIP_{HGW} : m$ から仮想 IP アドレス $vIP_{CN} : d$ へアドレス変換して上位層へ渡す．

以上の手順により，Mobile IP においても NAT-f を組み合わせることにより CN がプライベートネットワークに存在しても通信の開始及び継続を実現できる．HGW 及び CN は通信相手となる MN のアドレスを HoA として認識しているため，MN 移動後のネットワークに FA が設置されている場合や，Reverse Tunneling を行う場合においても，NAT-f を適用することが可能である．

6.6.2 ネットワークモビリティへの応用

移動透過性プロトコルは，これまで述べてきたホスト単位の移動透過性（ホストモビリティ）を実現する技術のほかに，ネットワーク単位の移動透過性（ネットワークモビリティ）を実現する技術がある．ネットワーク単位の移動透過性技術として，Mobile IP をベースとした NEMO（Network Mobility） [168, 169] や，Mobile PPC をベースとした Mobile NPC [170] がある．これらは移動透過性プロトコルを実装したルータ MR（Mobile Router）の配下にモバイルネットワークを形成し，移動透過性プロトコルを実装しない一般のノード LFN（Local Fixed Node）を収容する．MR が移動して IP アドレスが変化しても，配下の LFN はグローバルネットワーク上の CN と確立していた通信を継続できる．

NEMO は Mobile IP の拡張として定義されており，プロトコル上の違いは NEMO を運用することを示すフラグと，モバイルネットワークの情報をやりとりするためのオプションが制御情報に追加されたことである [171]．MR と HA が扱うアドレスに関する情報は，MR の HoA と気付けアドレス CoA（Care-of Address），及びモバイルネットワークのプレフィックス情報である．これらは MR 側から見ると全て送信元側の情報であり，宛先側の情報は含まれない．一方，提案方式特有のアドレスである仮想 IP アドレスは，MR 側から見ると宛先側の情報として用いられる．従って，NAT-f は Mobile IP と同様に NEMO 固有の制御に影響を及ぼすことなく組み合わせて利用することができる．提案方式を NEMO に応用する場合は，MR に NAT-f を実装して仮想アドレス変換処理などを行う．これにより，LFN はプライベートネットワークに存在する CN との通信を開始，継続することができる．

Mobile NPC は本章における NAT と同様に，NAT-f と Mobile PPC を実装した NAT ルータにより移動プライベートネットワークを構築する．LFN から外部ノードに対しては通常の通信と同様に自由に通信を開始することができる．移動プライベートネットワークが移動して NAT ルータの外側 IP アドレスが変化すると，NAT ルータは配下の LFN が通信している全ての外部ノードに対して CU 処理を行う．CIT と NAT マッピング情報に基づくアドレス変換を組み合わせることにより，LFN が確立していたコネクションを維持することができる．

なお，文献 [172] では動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol）をベースとして，Mobile PPC と NAT-f の機能を統合した形で Mobile NPC を実装している．動作検

証の結果，統合された各プロトコルは正常に動作することが確認されており，統合に伴う性能の劣化もないことが確認されている．

6.6.3 IPv4/IPv6 混在環境への応用

IPv4 ネットワークは当分の間，IPv6 ネットワークと混在することが想定される．このような混在したネットワークにおいて移動透過性を実現する技術として DSMIP（Dual Stack Mobile IP）[173] や DSMPPC（Dual Stack Mobile PPC） [174] がある．DSMIP は MN が IPv4/IPv6 の両者に対応することにより，IPv6 ネットワーク，IPv4 グローバルネットワーク，及び IPv4 プライベートネットワークの間を移動することを実現しており，IP Mobility の普及に適した解決策である．しかし DSMIP においても，CN は IPv6 ノードであること，または IPv4 グローバルアドレスが割り当てられていることを前提としており，IPv4 プライベートネットワーク内のノードと通信を開始することができないという課題が残されている．このようなシステムにおいても，提案方式を応用することにより上記課題を解決できると考えられる．

DSMPPC は IPv4/IPv6 プロトコルスタックに対応した Mobile PPC である．Mobile PPC は IPv4 と IPv6 で同じ方式，すなわち移動前後の IP アドレスを変換することにより移動透過性を実現できる．これは現在検討段階の技術であるが，文献 [174] では IP アドレスの変換だけでなく，IPv4 ヘッダと IPv6 ヘッダの変換を行うことにより混在環境における移動透過性の実現を狙っている．

6.7 結論

本章では，これまで検討の対象となっていなかった移動透過性の通信ケース，すなわち MN が宅外からホームネットワーク内の CN に通信を開始し，MN が移動した場合においても通信を継続できる方式を提案した．NAT-f を適用することにより，NAT-f と移動透過性プロトコルが互いに独立性を保持しつつ，容易に組み合わせることができることを示した．また，提案アーキテクチャがネットワークモビリティや IPv4/IPv6 混在環境における移動透過性を実現する既存技術に応用可能であることを示した．NAT-f と Mobile PPC を組み合わせたシステムを実装して性能評価を行った結果，スループットの低下は十分に小さく，実用上問題ないことを確認した．

第7章　結論

7.1　総括

いつでも，どこでも，何でも，誰でもアクセスが可能なユビキタスネットワークを実現するためには，暗号化通信，移動通信，エンドツーエンド通信を同時に行うことが重要である．本研究では柔軟性とセキュリティを兼ね備えたセキュア通信グループを構築できる FPN（Flexible Private Network）と呼ぶシステムに，位置透過性，移動透過性，アドレス空間透過性の機能を追加し，上記 3 つの通信を実現するネットワークの概念として拡張した．FPN を段階的に実現するための一連の通信アーキテクチャとして GSCIP（Grouping for Secure Communication for IP）を提案し，独立した 3 つの透過性を実現するプロトコルを統一した考え方に基づいて設計した．

2 章 では GSCIP の主要プロトコルである動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol）を提案した．DPRP は FPN の実現に必須となる位置透過性，すなわちネットワーク構成の変化に動的に対応する機能を実現するためのプロトコルである．DPRP は通信に先立って通信相手が同一のセキュア通信グループのメンバか確認し，ノード間の暗号化通信に必要な動作処理情報テーブルを動的に生成する役割を持つ．DPRP を FreeBSD に実装し，通信開始時に発生するオーバヘッドが TCP/UDP 通信にほとんど影響を与えないことを確認した．また，ネットワーク構成が変化した際に発生するコストを評価し，管理負荷を大幅に軽減できることを示した．

3 章 では NAT やファイアウォールと共存でき，かつオリジナルパケットのフォーマットを変えないまま本人性確認とパケットの完全性保証を実現する暗号通信方式 PCCOM（Practical Cipher Communication）を提案した．PCCOM はセキュア通信グループのメンバ間で行う暗号化通信を実現する方式である．本人性確認とパケット全体の完全性保証は，共通秘密鍵とパケットの内容から生成した疑似データと呼ぶ値を用いて，独自の TCP/UDP チェックサム再計算を行うことにより実現した．PCCOM の有効性を確認するために試作システムを FreeBSD 上に実装し，NAT やファイアウォールとの親和性が高いことを確認した．また，スループットを測定した結果，パケットフォーマットを変えないことによる性能上の効果があることを確認した．

4 章 ではモバイルノードの IP アドレスが変化しても通信を継続できる移動透過性を実現する Mobile PPC（Mobile Peer-to-Peer Communication protocol）を提案した．Mobile PPC は移動前後の IP アドレスの違いをノード間で共有し，アドレス変換処理によりアドレスの変化を隠蔽する方式である．また移動透過性プロトコルを実装しない一般ノードとの上位互換性を有しており，段階的な普及が可能である．Mobile PPC を FreeBSD 上に実装し検証をした結果，エンドツーエンドでかつ高スループットを維持したままで移動透過通信が行えることを示した．

5 章 ではエンドツーエンド通信の障害となる NAT 越え問題を解決し，アドレス空間透過性を実

現する外部動的マッピング方式を提案した．提案方式を実現するためのプロトコルとして NAT-f（NAT-free protocol）を定義し，外部ノードが NAT 配下のノードに通信を開始する際，NAT とネゴシエーションを行うことにより，NAT にマッピング処理を行わせる．外部ノードはカーネルにおいて，NAT でマッピングされた情報に一致するようにアドレス/ポート変換を行うことにより，NAT 越え通信を実現する．プロトタイプシステムの実装を行い，エンドノード間の初期遅延およびスループットを評価した結果，通信開始時の遅延増加は 1 ms 以下であり，スループットは提案方式を実装しない場合と比べ，同等であることを確認した．

6 章 では NAT- f を既存の移動透過性プロトコルと融合する方式を提案し，本研究の応用例について示した．移動ノードは通信開始時に NAT-f によりプライベートネットワーク内の通信相手ノードと通信を開始し，移動時は既存の移動透過性プロトコルを用いて通信を継続する．NAT-f と移動透過性プロトコルは異なるタイミングで処理を行うため，互いに独立性を保持しつつ，かつ容易に組み合わせることができる．NAT-f と Mobile PPC を実装したシステムを評価した結果，所定の機能を実行できること，かつ通信開始時のオーバヘッド及びスループットの低下は十分に小さいことを確認した．この応用研究は，外出先ノードからホームネットワーク内の情報家電機器などと自由に通信を行いながら移動通信がしたいというユーザの要求を満たすことができ，従来研究では実現できなかった新たな通信スタイルを確立することができた．

以上の研究成果より，提案アーキテクチャは本論文で設定した 5 つのユビキタスネットワークの要求仕様，すなわち

- 高セキュリティと低管理負荷の両立
- ノードの位置に依存しない柔軟性
- アプリケーションに依存しない汎用性
- 特殊なサーバの導入回避
- 低遅延・高スループット

を全て満たすことができ，先行研究に対する優位性を示すことができた．また，暗号化通信，移動通信，エンドツーエンド通信を IPv4 ネットワークにおいて同時に実現できることを確認し，ユビキタスネットワークを実現できるアーキテクチャとしての有効性を示した．

7.2 今後の課題

ユビキタスネットワークを早期に実現するためには，一般ユーザが提案システムを容易に利用できる環境を整備する必要がある．現在，提案アーキテクチャをオープンソースとして公開することを検討しており，さらに一般ユーザが既存の環境で体験できるよう Live CD や Windows OS に対応したパッケージ [175] の提供など，普及に関する具体的方法を検討，推進していくことが重要である．ただし，GSCIP を構成するプロトコルはそれぞれ独立しており，各プロジェクトによりプロトコルの拡張が行われている．そのため，プロトタイプとして実装した GPACK に最新のプロトコル仕様を反映仕切れておらず，提供可能なパッケージを早期に完成させる必要がある．

また，ユビキタスネットワークは IPv4 ネットワーク環境だけではなく，IPv6 ネットワーク環境でその真価を発揮すると考えられている．本論文で提案した GSCIP アーキテクチャは IPv4/IPv6 の両者に対応可能な設計となっているが [174, 176, 177]，現時点では IPv4 プロトコルスタックに対応した実装まで完了している．今後は IPv6 プロトコルスタックに対応した GSCIPv6 を実装し，IPv4 ネットワークと IPv6 ネットワークをまたがって動作する仕様に拡張する必要がある．

これまではデファクト標準に基軸を置き，研究開発を行ってきた．GSCIP アーキテクチャをベースとして各プロトコルを提案してきたが，IETF（Internet Engineering Task Force）がこれまでに標準化してきた暗号化通信や移動通信，NAT 越え通信を実現するプロトコルに対して，それぞれ優位性を示すことができた．今後はインターネット・ドラフトの提出など，提案アーキテクチャの標準化を目指したデジュール標準の活動も併せて行うことが望ましいと考えられる．

GSCIP では定期的にグループ鍵を更新することを想定しているため，通信グループのメンバ数 n が増加すると鍵配送におけるオーバヘッドの増加が懸念される．これについてはメンバ数 n に対して $\log n$ の通信でよい方式や，n に依存しない方式が既に知られている [178, 179]．また鍵更新が頻繁でかつ時間がかかるため，DoS 攻撃の対象にされる危険性があり，今後検討が必要と考えられる．

FPN の適用範囲をインターネットに拡大する場合，セキュア通信グループを定義する GMS の運用管理方法を検討する必要がある．企業ネットワークでは管理者が部署や役職に応じてセキュア通信グループを定義することは容易であるが，インターネットでは不特定多数のメンバが存在し，それらを一元管理することが難しい．ユーザが自主的にセキュア通信グループを定義したい場合，管理者に依頼する方法では即効性がなく，オンデマンドでセキュア通信グループを定義できる仕組みが重要だと考えられる．

また，個々のプロトコルに関しては以下のような課題が残されている．

本論文における DPRP は同一アドレス空間でのみ動作する仕様を提案した．ホームネットワークを含むインターネットに FPN を適用する場合，異なるアドレス空間をまたがった DPRP ネゴシエーションを実現する必要がある．現在，DPRP に NAT-f を統合することにより上記ネゴシエーションを実現するシステムを実装している [180]．プロトタイプシステムを構築して動作確認まで完了しているが，実環境で運用する場合に必要となる機能についてはさらに検討やプロトコル仕様の拡張が必要である．

Mobile PPC は通信開始時に DH 鍵交換により認証鍵を共有する．DH 鍵交換は一般に中間者攻撃に対して脆弱性があるため，認証を伴った処理を行う必要がある．FPN システム環境下では互いにグループ鍵を所持しているため問題ないが，FPN の枠外で利用する場合は別途対策が必要である．Mobile PPC は特有のサーバを導入しないため，DDNS サーバを有効に活用する方法が考えられる．移動ノードは DDNS サーバに対して名前登録を行う際，共通鍵により認証を行うことを想定している．そのため，DH 公開鍵を DDNS サーバを経由することにより，中間者攻撃の可能性を限りなく小さくすることができると思われる．

NAT-f は TCP/UDP パケットをトリガとしてマッピング処理を行うが，その対象をユニキャストパケットに限定している．例えば DLNA はネットワーク上から情報家電製品を発見するためにマ

ルチキャストが利用される．現在，遠隔地からホームネットワークに設置された DLNA 機器と通信するためプロトコル仕様を拡張している [181]．今後は DLNA に特化するだけでなく，マルチキャストに汎用的に対応できる方式を検討する必要がある．

謝辞

本研究を遂行するにあたり，多大なる御指導，御鞭撻を賜りました，名城大学理工学部情報工学科の渡邊晃教授に心より厚く御礼申し上げます．また，本論文をまとめるにあたり，有益な御助言をして頂きました，名城大学理工学部情報工学科の高橋友一教授，名城大学理工学部情報工学科の田中敏光教授，福井工業大学工学部電気電子工学科の鹿間敏弘教授に心より御礼申し上げます．

本研究における暗号通信方式に対して，様々な御助言，御検討を頂きました宮崎大学工学部情報システム工学科の岡崎直宣准教授に心より感謝致します．また，移動透過性およびアドレス空間透過性に関する論文執筆時において，様々な御助言，御検討を頂きました名城大学理工学部情報工学科の宇佐見庄五准教授に心より感謝致します．そして日々の研究活動に対して様々な御指導を頂きました元名城大学理工学部情報工学科教授の小川明先生，山本新先生に心より感謝致します．

なお，本研究は平成 20 年度より日本学術振興会科学研究費補助金（特別研究員奨励費 20・1069）の助成を受けたものです．ここに記して感謝致します．

本研究を遂行するにあたり，様々な御検討，御協力を頂いた，渡邊研究室の皆様に感謝致します．とりわけ，本研究テーマに関する深い議論をして頂いた，

- 加藤尚樹氏（現在，日本コムシス株式会社勤務）
- 竹内元規氏（現在，日本コムシス株式会社勤務）
- 竹尾大輔氏（現在，三菱電機情報ネットワーク株式会社勤務）
- 保母雅敏氏（現在，株式会社日立情報システムズ勤務）
- 増田真也氏（現在，NTT ソフトウェア株式会社勤務）
- 柳沢信成氏（現在，アイホン株式会社勤務）
- 坂本順一氏（現在，株式会社東芝勤務）
- 瀬下正樹氏（現在，日本電気株式会社勤務）
- 金本綾子氏（現在，ブラザー工業株式会社勤務）
- 束　長俊氏（現在，株式会社日立情報システムズ勤務）
- 葛谷章一氏（現在，株式会社トヨタデジタルクルーズ勤務）
- 後藤裕司氏（現在，名城大学大学院理工学研究科情報科学専攻修士２年）
- 今村圭佑氏（現在，名城大学大学院理工学研究科情報科学専攻修士２年）

に心より感謝致します．

最後に，研究を進めていく中，いつも暖かく支えて頂いた両親に心より感謝致します．

参考文献

[1] ユビキタスネット社会の実現に向けた政策懇談会（編）：u-Japan 政策-2010 年ユビキタスネット社会の実現に向けて-，総務省 (2004).
http://www.soumu.go.jp/s-news/2004/pdf/041217_7_bt2_all.pdf

[2] 総務省：情報通信白書（平成 20 年版） (2008).
http://www.johotsusintokei.soumu.go.jp/whitepaper/ja/h20/pdf/20honpen.pdf

[3] Dierks, T. and Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246, IETF (2008).

[4] Lehtinen, S. and Lonvick, C.: The Secure Shell (SSH) Protocol Assigned Numbers, RFC 4250, IETF (2006).

[5] Ylonen, T. and Lonvick, C.: The Secure Shell (SSH) Protocol Architecture, RFC 4251, IETF (2006).

[6] Ylonen, T. and Lonvick, C.: The Secure Shell (SSH) Transport Layer Protocol, RFC 4253, IETF (2006).

[7] Ylonen, T. and Lonvick, C.: The Secure Shell (SSH) Authentication Protocol, RFC 4252, IETF (2006).

[8] Ylonen, T. and Lonvick, C.: The Secure Shell (SSH) Connection Protocol, RFC 4254, IETF (2006).

[9] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L. and Repka, L.: S/MIME Version 2 Message Specification, RFC 2311, IETF (1998).

[10] Dusse, S., Hoffman, P., Ramsdell, B. and Weinstein, J.: S/MIME Version 2 Certificate Handling, RFC 2312, IETF (1998).

[11] Callas, J., Donnerhacke, L., Finney, H., Shaw, D. and Thayer, R.: OpenPGP Message Format, RFC 4880, IETF (2007).

[12] Kent, S. and Seo, K.: Security Architecture for the Internet Protocol, RFC 4301, IETF (2005).

[13] Gleeson, B., Lin, A., Heinanen, J., Armitage, G. and Malis, A.: A Framework for IP Based Virtual Private Networks, RFC 2764, IETF (2000).

[14] 寺岡文男：インターネットにおけるノード移動透過性プロトコル，電子情報通信学会論文誌 D-I， Vol. J87-D-I, No. 3, pp. 308–328 (2004).

[15] Perkins, C.: IP Mobility Support for IPv4, RFC 3344, IETF (2002).

[16] Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, RFC 3775, IETF (2004).

[17] The 3rd Generation Partnership Project: 3GPP home page. http://www.3gpp.org/

[18] Leung, K., Dommety, G., Yegani, P. and Chowdhury, K.: WiMAX Forum/3GPP2 Proxy Mobile IPv4, Internet-draft, IETF (2008). http://tools.ietf.org/draft/draft-leung-mip4-proxy-mode/draft-leung-mip4-proxy-mode-09.txt

[19] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K. and Patil, B.: Proxy Mobile IPv6, RFC 5213, IETF (2008).

[20] The WiMAX Forum: WiMAX Forum. http://www.wimaxforum.org/

[21] Srisuresh, P. and Egevang, K.: Traditional IP Network Address Translator (Traditional NAT), RFC 3022, IETF (2001).

[22] Ford, B., Srisuresh, P. and Kegel, D.: Peer-to-Peer Communication Across Network Address Translators, *Proceedings of The Annual Conference on USENIX Annual Technical Conference*, pp. 179–192 (2005).

[23] Srisuresh, P., Ford, B. and Kegel, D.: State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs), RFC 5128, IETF (2008).

[24] Muller, A., Carle, G. and Klenk, A.: Behavior and Classification of NAT Devices and Implications for NAT Traversal, *IEEE Network*, Vol. 22, No. 5, pp. 14–19 (2008).

[25] Park, C., Jeong, K., Kim, S. and Lee, Y.: NAT Issues in the Remote Management of Home Network Devices, *IEEE Network*, Vol. 22, No. 5, pp. 48–55 (2008).

[26] Eyeball Networks Inc.: *NAT Traversal for VoIP and Internet Communications using STUN, TURN and ICE* (2007). http://www.eyeball.com/technology/whitepapers/EyeballAnyfirewallWhitePaper.pdf

[27] Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF (1998).

[28] IPv6 普及・高度化推進協議会（編）：IPv6 接続サービスの提供状況に関する調査の結果について，総務省 (2007). http://www.v6pc.jp/pdf/070330_h18_ipv6.pdf

[29] Okazaki, N., Park, M., Watanabe, A., Seno, S., Ideguchi, T. and Yabe, M.: Realization Method of Flexible Private Network System, *Transactions of The Institute of Electrical Engineers of Japan. C*, Vol. 120-C, No. 8, pp. 1242–1249 (2000).

[30] 渡邊　晃，厚井裕司，井手口哲夫，横山幸雄，妹尾尚一郎：暗号技術を用いたセキュア通信グループの構築方式とその実現，情報処理学会論文誌，Vol. 38, No. 4, pp. 904–914 (1997).

[31] 今村圭祐，鈴木秀和，渡邊　晃：GSCIP と IPsec を併用したリモートアクセス方式の提案と評価，マルチメディア，分散，協調とモバイル（DICOMO2007），Vol. 2007, No. 1, pp. 468–472 (2007).

[32] 筒井章博，藤井伸朗，川村龍太郎，依田育生：次世代ホームネットワーク技術，電子情報通信学会誌，Vol. 89, No. 12, pp. 1067–1072 (2006).

[33] 鈴木秀和，竹内元規，加藤尚樹，増田真也，渡邊　晃：フレキシブルプライベートネットワークを実現するセキュア通信アーキテクチャ GSCIP の提案，DICOMO2005 シンポジウム論文集，Vol. 2005, No. 6, pp. 441–444 (2005).

[34] 鈴木秀和，渡邊　晃：フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装と評価，情報処理学会論文誌，Vol. 47, No. 11, pp. 2976–2991 (2006).

[35] 増田真也，鈴木秀和，岡崎直宣，渡邊　晃：NAT やファイアウォールと共存できる暗号通信方式 PCCOM の提案と実装，情報処理学会論文誌，Vol. 47, No. 7, pp. 2258–2266 (2006).

[36] 保母雅敏，渡邊　晃：IC カードを用いた重要情報の配送方式 SPAIC の検討，マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集，Vol. 2005, No. 6, pp. 9–12 (2005).

[37] 束　長俊，鈴木秀和，渡邊　晃：非接触型 IC カードを用いた認証方式 SPAIC の提案，マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集，Vol. 2007, No. 1, pp. 1332–1337 (2007).

[38] Shu, C., Suzuki, H. and Watanabe, A.: Proposal of an Authentication Method “SPAIC” using a Non-contact Type IC Card, *Proceedings of IEEE 7th International Symposium on Communications and Information Technologies* (*ISCIT2007*), pp. 1470–1475 (2007).

[39] 今村圭佑，鈴木秀和，後藤裕司，渡邊　晃：セキュア通信アーキテクチャ GSCIP を実現するグループ管理サーバの実装と運用評価，マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集，Vol. 2008, No. 1, pp. 1516–1522 (2008).

[40] Huttunen, A., Swander, B., Volpe, V., Diburro, L. and Stenberg, M.: UDP Encapsulation of IPsec Packets, RFC 3948, IETF (2005).

[41] Kivinen, T., Swander, B., Huttunen, A. and Volpe, V.: Negotiation of NAT-Traversal in the IKE, RFC 3947, IETF (2005).

[42] Levkowetz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519, IETF (2003).

[43] Wing, D., Rosenberg, J. and Tschofenig, H.: Discovering, Querying, and Controlling Firewalls and NATs, Internet-draft, IETF (2007).
http://tools.ietf.org/id/draft-wing-behave-nat-control-stun-usage-05.txt

[44] Tschofenig, H. and Bajko, G.: Mobile IP Interactive Connectivity Establishment (M-ICE), Internet-draft, IETF (2008). http://tools.ietf.org/id/draft-tschofenig-mip6-ice-02.txt

[45] 竹内元規，鈴木秀和，渡邊　晃：エンドエンドで移動透過性を実現する Mobile PPC の提案と実装，情報処理学会論文誌，Vol. 47, No. 12, pp. 3244–3257 (2006).

[46] 鈴木秀和, 宇佐見庄五, 渡邊　晃：外部動的マッピングにより NAT 越え通信を実現する NAT-f の提案と実装，情報処理学会論文誌，Vol. 48, No. 12, pp. 3949–3961 (2007).

[47] Gordon, L., Loeb, P., Lucyshyn, W. and Richardson, R.: 2004 CSI/FBI Computer Crime and Security Survey, Technical report, Computer Security Institute (2004).

[48] 荒井正人，鍛　忠志，伊藤浩道，手塚　悟，佐々木良一：企業情報向けグループ暗号システム，情報処理学会論文誌，Vol. 40, No. 12, pp. 4378–4387 (1999).

[49] 岡田浩一，冨士　仁：個人単位の VPN を実現するネットワークサービス「VPN-exchange」，CSS2001 論文集，情報処理学会，pp. 67–72 (2001).

[50] 辻本孝博，唐澤　圭，藤崎智宏，三上博英：IPv6 IPSec による End-to-End VPN 構築方式に関する考察，情報処理学会研究報告，2001-CSEC-014, Vol. 2001, No. 75, pp. 205–210 (2001).

[51] Kourai, K., Hirotsu, T., Sato, K., Akashi, O., Fukuda, K., Sugawara, T. and Chiba, S.: Secure and Manageable Virtual Private Networks for End-users, *Proceesings of Annual IEEE Conference of Local Computer Networks* (*LCN 2003*), pp. 385–394 (2003).

[52] 藤田範人，石川雄一，岩田　淳，飯島明夫：DNS を用いたスケーラブルな VPN アーキテクチャ，電子情報通信学会 2004 年総合大会講演論文集，Vol. 2004, No. 2, p. 200 (2004).

[53] Rodeh, O., Birman, K., Hayden, M. and Dolev, D.: Dynamic Virtual Private Networks, Technical Report TR98-1695, Department of Computer Science, Cornell University (1998).

[54] 加島伸吾，後藤幸功，荒木啓二郎：DVPN の提案と応用，マルチメディア, 分散, 協調とモバイル（DICOMO2003）シンポジウム論文集，Vol. 2003, No. 9, 情報処理学会，pp. 365–368 (2003).

[55] 堀　賢治，吉原貴仁，堀内浩規：ピアツーピア型レイヤ 2 インターネット VPN 自動設定方式の実装と評価，情報処理学会第 67 回全国大会論文集，pp. 485–486 (2004).

[56] Kindred, D. and Sterne, D.: Dynamic VPN Communities: Implementation and Experience, *Proceesings of 2nd DARPA Information Survivability Conference and Exposition II* (*DISCEX II '01*), Vol. 1, pp. 254–263 (2001).

[57] 萱島　信，寺田真敏，藤山達也，小泉　稔，加藤恵理：多重ファイアウォール環境に適したVPN 構築方式の提案，電子情報通信学会論文誌 D-I，Vol. J82-D-I, No. 6, pp. 772–778 (1999).

[58] 岡山聖彦，山井成良，石橋勇人，安倍広多，松浦敏雄：代理ゲートウェイを用いた SOCKS ベースの階層的 VPN 構成法，情報処理学会論文誌，Vol. 42, No. 12, pp. 2860–2868 (2001).

[59] Leech, M., Ganis, M., Lee, Y., Kuris, R., Koblas, D. and Jones, L.: SOCKS Protocol Version 5, RFC 1928, IETF (1996).

[60] Wong, C., Gouda, M. and Lam, S.: Secure Group Communications Using Key Graphs, *IEEE/ACM Transaction on Networking*, Vol. 8, No. 1, pp. 16–30 (2000).

[61] 鎌田　実，川瀬徹也，渡邊　晃，笹瀬　巌：部門 VPN 構成下におけるマルチキャスト通信方式の提案とその評価，電子情報通信学会論文誌 B，Vol. J82-B, No. 11, pp. 2061–2073 (1999).

[62] Amir, Y., Kim, Y., Nita-Rotaru, C., Schultz, J., Stanton, J. and Tsudik, G.: Secure Group Communication Using Robust Contributory Key Agreement, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 5, pp. 468–480 (2004).

[63] Harney, H., Meth, U., Colegrove, A. and Gross, G.: GSAKMP: Group Secure Association Key Management Protocol, RFC 4535, IETF (2006).

[64] 渡邊　晃，井手口哲夫，笹瀬　巌：イントラネット閉域通信グループの物理的位置透過性を可能にする動的処理解決プロトコルの提案，電子情報通信学会論文誌 D-I，Vol. J84-D-I, No. 3, pp. 269–284 (2001).

[65] Harkins, D. and Carrel, D.: The Internet Key Exchange (IKE), RFC 2409, IETF (1998).

[66] Kent, S.: IP Encapsulating Security Payload (ESP), RFC 4303, IETF (2005).

[67] Kent, S.: IP Authentication Header, RFC 4302, IETF (2005).

[68] Maughan, D., Schertler, M., Schneider, M. and Turner, J.: Internet Security Association and Key Management Protocol (ISAKMP), RFC 2408, IETF (1998).

[69] Orman, H.: The OAKLEY Key Determination Protocol, RFC 2412, IETF (1998).

[70] National Institute of Standards and Technology: Specification for the ADVANCED ENCRYPTION STANDARD (AES), FIPS 197, U.S. Department of Commerce (2001).

[71] Rivest, R.: The MD5 Message-Digest Algorithm, RFC 1321, IETF (1992).

[72] Plummer, D. C.: An Ethernet Address Resolution Protocol, RFC 826, IETF (1982).

[73] The OpenSSL Project: The Open Source toolkit for SSL/TLS. http://www.openssl.org/

[74] Ethereal: A Network Protocol Analyzer. http://www.ethereal.com/

[75] Jinmei, T., Yamamoto, K., Hagino, J., Sumikawa, M., Inoue, Y., Sugyo, K. and Sakane, S.: An overview of the KAME network software: Design and implementation of the advanced internet-working platform, *Proceesings of INET'99* (1999).
http://www.isoc.org/isoc/conferences/inet/99/proceedings/4s/4s_2.htm

[76] The KAME project: KAME. http://www.kame.net/

[77] Kaufman, C.: Internet Key Exchange (IKEv2) Protocol, RFC 4306, IETF (2005).

[78] Intel Corporation: *Using the RDTSC Instruction for Performance Monitoring* (1997).
http://cs.smu.ca/ jamuir/rdtscpm1.pdf

[79] 菊池　豊, 藤井資子, 山本正晃, 永見健一, 中川郁夫：遅延計測による日本のインターネットトポロジーの推定，電子情報通信学会技術研究報告，IA2007-27, Vol. 107, No. 151, pp. 103–108 (2007).

[80] 吉田　薫，藤井資子，菊池　豊，山本正晃，永見健一，中川郁夫，江崎　浩：ユーザ視点に基づいたブロードバンドインターネット環境における遅延・パケットロスの傾向分析，電子情報通信学会論文誌 B，Vol. J91-B, No. 10, pp. 1182–1192 (2008).

[81] Diffie, W. and Hellman, M.: New Directions in Cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644–654 (1976).

[82] Rescorla, E.: Diffie-Hellman Key Agreement Method, RFC 2631, IETF (1999).

[83] Zhang, Y. and Singh, B.: A Multi-Layer IPsec Protocol, *Proceedings of The 9th Conference on USENIX Security Symposium*, Vol. 9, p. 16 (2000).

[84] Braden, R., Borman, D. and Partridge, C.: Computing the Internet Checksum, RFC 1071, IETF (1988).

[85] Mallory, T. and Kullberg, A.: Incremental Updating of the Internet Checksum, RFC 1141, IETF (1990).

[86] Rijsinghani, A.: Computation of the Internet Checksum via Incremental Update, RFC 1624, IETF (1994).

[87] Krawczyk, H., Bellare, M. and Canetti, R.: HMAC: Keyed-Hashing for Message Authentication, RFC 2104, IETF (1997).

[88] Jones, R.: The Netperf Homepage. http://www.netperf.org/netperf/NetperfPage.html

[89] Perkins, C.: IP Encapsulation within IP, RFC 2003, IETF (1996).

[90] Calhoun, P. and Perkins, C.: Mobile IP Network Access Identifier Extension for IPv4, RFC 2794, IETF (2000).

[91] Montenegro, G.: Reverse Tunneling for Mobile IP, revised, RFC 3024, IETF (2001).

[92] Perkins, C., Calhoun, P. and Bharatia, J.: Mobile IPv4 Challenge/Response Extensions (Revised), RFC 4721, IETF (2007).

[93] Bhagwat, P., Maltz, D. and Segall, A.: MSOCKS+: an architecture for transport layer mobility, *Computer Networks*, Vol. 39, No. 4, pp. 385–403 (2002).

[94] Funato, D., Yasuda, K. and Tokuda, H.: TCP-R: TCP Mobility Support for Continuous Operation, *Proceedings of IEEE International Conference on Network Protocol* (*ICNP 1997*), pp. 229–236 (1997).

[95] Snoeren, A. and Balakrishnan, H.: An End-to-End Approach to Host Mobility, *Proceedings of The 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking* (*MobiCom 2000*), pp. 155–156 (2000).

[96] 松岡保静，吉村　健，大矢智之：エンドツーエンド型 IP ソフトハンドオーバ，電子情報通信学会論文誌 B，Vol. J86-B, No. 8, pp. 1369–1378 (2003).

[97] Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H. and Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks, *IEICE Transactions on Communications*, Vol. E84-B, No. 8, pp. 2076–2086 (2001).

[98] 相原玲二，藤田貫大，前田香織，野村嘉洋：アドレス変換方式による移動透過インターネットアーキテクチャ，情報処理学会論文誌，Vol. 43, No. 12, pp. 3889–3897 (2002).

[99] Nordmark, E. and Bagnulo, M.: Shim6: Level 3 Multihoming Shim Protocol for IPv6, Internet-draft, IETF (2007). http://www.shim6.org/draft-ietf-shim6-proto-09.txt

[100] Arkko, J. and Beijnum, I. V.: Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming, Internet-draft, IETF (2007).
http://www.shim6.org/draft-ietf-shim6-failure-detection-09.txt

[101] Bagnulo, M.: Hash Based Addresses (HBA), Internet-draft, IETF (2008).
http://www.shim6.org/draft-ietf-shim6-hba-06.txt

[102] Moskowitz, R. and Nikander, P.: Host Identity Protocol (HIP) Architecture, RFC 4423, IETF (2006).

[103] Moskowitz, R., Nikander, P., Jokela, P. and Henderson, T.: Host Identity Protocol, RFC 5201, IETF (2008).

[104] Jokela, P., Moskowitz, R. and Nikander, P.: Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP), RFC 5202, IETF (2008).

[105] Laganier, J., Koponen, T. and Eggert, L.: Host Identity Protocol (HIP) Registration Extension, RFC 5203, IETF (2008).

[106] Laganier, J.: Host Identity Protocol (HIP) Rendezvous Extension, RFC 5204, IETF (2008).

[107] Nikander, P. and Laganier, J.: Host Identity Protocol (HIP) Domain Name System (DNS) Extension, RFC 5205, IETF (2008).

[108] Nikander, P., Henderson, T., Vogt, C. and Arkko, J.: End-Host Mobility and Multihoming with the Host Identity Protocol, RFC 5206, IETF (2008).

[109] Stiemerling, M., Quittek, J. and Eggert, L.: NAT and Firewall Traversal Issues of Host Identity Protocol (HIP) Communication, RFC 5207, IETF (2008).

[110] ITU-T: *One-way Transmission Time*, ITU-T Recommendation G.114 (2003).

[111] Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).

[112] Ferguson, P. and Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, IETF (2000).

[113] Droms, R.: Dynamic Host Configuration Protocol, RFC 2131, IETF (1997).

[114] Internet Systems Consortium: ISC BIND. https://www.isc.org/software/bind

[115] 石山政浩，國司光宣，河野通宗，寺岡文男：移動体通信プロトコル LIN6 における後方互換性拡張の一方式，電子情報通信学会技術研究報告，IA2005-25, Vol. 102, No. 362, pp. 23–28 (2002).

[116] Internet Systems Consortium: ISC DHCP. https://www.isc.org/sw/dhcp

[117] Mishra, A., Shin, M. and Srbaugh, W.: An Empirical Analysis of the IEEE802.11 MAC Layer Handoff Process, *ACM SIGCOMM Computer Communication Review*, Vol. 33, No. 2, pp. 93–102 (2003).

[118] 小川猛志，伊東　匡：DHCP をベースとしたシームレスハンドオーバ方法の研究，電子情報通信学会論文誌 B，Vol. J88-B, No. 11, pp. 2228–2238 (2005).

[119] Moore, N. and Daley, G.: Fast Address Configuration Strategies for the Next-Generation Internet, *Proceedings of the Australian Telecommunications, Networks, and Applications Conference* (*ATNAC 2003*) (2003).

[120] Narten, T., Nordmark, E. and Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6), RFC 2461, IETF (1998).

[121] 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, Vol. 105, No. 80, pp. 13–18 (2005).

[122] Thomson, S. and Narten, T.: IPv6 Stateless Address Autoconfiguration, RFC 2462, IETF (1998).

[123] The Portland State University Secure Mobile Networking Project: PSU Mobile-IP. http://www.cs.pdx.edu/research/SMN/

[124] 瀬下正樹，渡邊　晃：Mobile PPC における認証方式の実装，マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集 (II)，Vol. 2006, No. 6, pp. 809–812 (2006).

[125] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, RFC 3261, IETF (2002).

[126] Stewart, R.: Stream Control Transmission Protocol, RFC 4960, IETF (2007).

[127] Kohler, E., Handley, M. and Floyd, S.: Datagram Congestion Control Protocol (DCCP), RFC 4340, IETF (2006).

[128] 鈴木秀和，渡邊　晃：Hole Punching を用いた NAT 越え Mobile PPC の設計，情報処理学会研究報告，2008-MBL-045, Vol. 2008, No. 44, pp. 69–74 (2008).

[129] Huitema, C.: Multi-homed TCP, Internet-draft, IETF (1995). http://tools.ietf.org/id/draft-huitema-multi-homed-01.txt

[130] Wong, K. D., Dutta, A., Schulzrinne, H. and Young, K.: Simultaneous Mobility: Analytical Framework, Theorems and Solutions, *Wireless Communications & Mobile Computing*, Vol. 7, No. 5, pp. 623–642 (2007).

[131] Liu, Q., Li, S., He, H. and Wang, B.: A Multi-binding Solution for Simultaneous Mobility of MIPv6, *Proceedings of the 2nd IEEE International Symposium on Service-Oriented System Engineering* (*SOSE2006*), pp. 143–146 (2006).

[132] Lin, J. L. and Pan, J. Y.: Hand-Around: A Handoff Evolution with Monami6, *Proceedings of The 3rd International Conference on Wireless Communications, Networking and Mobile Computing* (*WiCOM2007*), pp. 1775–1778 (2007).

[133] 清水智行，中村素典，美濃導彦：NAPT を越えた端末の移動時の TCP コネクション維持による移動透過性保証プロトコル，情報処理学会研究報告，2002-DPS-107-5, Vol. 2002, No. 32, pp. 25–30 (2002).

[134] 金本綾子，鈴木秀和，伊藤将志，渡邊　晃：IPv4 移動体通信システムにおけるパケットロスレスハンドオーバの提案，情報処理学会論文誌，Vol. 50, No. 1, pp. 133–143 (2009).

[135] 張　冰冰，鈴木秀和，渡邊　晃：プロキシ中継型 Mobile PPC の検討，マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集，Vol. 2008, No. 1, pp. 1588–1592 (2008).

[136] 葛谷章一，瀬下正樹，渡邊　晃：プロキシを利用した Mobile PPC の検討，電子情報通信学会 2007 年総合大会講演論文集，B-7-164, Vol. 2007 年_通信，No. 2, p. 254 (2007).

[137] Rosenberg, J., Weinberger, J., Huitema, C. and Mahy, R.: STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), RFC 3489, IETF (2003).

[138] Huitema, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs), RFC 4380, IETF (2006).

[139] Guha, S. and Francis, P.: Simple Traversal of UDP Through NATs and TCP too (STUNT), Internet-draft, IETF (2004).
http://nutss.gforge.cis.cornell.edu/pub/draft-guha-STUNT-00.txt

[140] Guha, S. and Francis, P.: Characterization and Measurement of TCP Traversal through NATs and Firewalls, *Proceedings of The Internet Measurement Conference* (*IMC 2005*), pp. 199–211 (2005).

[141] Y.Takeda: Symmetric NAT Traversal using STUN, Internet-draft, IETF (2003). http://tools.ietf.org/draft/draft-takeda-symmetric-nat-traversal/draft-takeda-symmetric-nat-traversal-00.txt

[142] Rosenberg, J., Mahy, R. and Matthews, P.: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN), Internet-draft, IETF (2008).
http://tools.ietf.org/id/draft-ietf-behave-turn-10.txt

[143] UPnP Forum: *Internet Gateway Device* (*IGD*) *Standardized Device Control Protocol V 1.0* (2001).
http://www.upnp.org/standardizeddcps/igd.asp

[144] Cheshire, S., Krochmal, M. and Sekar, K.: NAT Port Mapping Protocol (NAT-PMP), Internet-draft, IETF (2008). http://tools.ietf.org/id/draft-cheshire-nat-pmp-03.txt

[145] Guha, S., Takeda, Y. and Francis, P.: NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity, *Proceedings of The ACM SIGCOMM Workshop on Future Directions in Network Architecture* (*FDNA 2004*), pp. 43–48 (2004).

[146] Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, Internet-draft, IETF (2007).
http://www.ietf.org/internet-drafts/draft-ietf-mmusic-ice-19.txt

[147] Ng, T., Stoica, I. and Zhang, H.: A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces, *Proceedings of The USENIX Annual Technical Conference 2001*, pp. 319–332 (2001).

[148] Kondo, K.: Capsulated Network Address Translation with Sub-Address (C-NATS), Internet-draft, IETF (2003).
http://tools.ietf.org/draft/draft-kuniaki-capsulated-nats/draft-kuniaki-capsulated-nats-05.txt

[149] Turányi, Z., Valkó, A. and Campbell, A.: 4+4: An Architecture for Evolving the Internet Address Space Back Toward Transparency, *ACM SIGCOMM Computer Communication Review*, Vol. 33, No. 5, pp. 43–54 (2003).

[150] Francis, P. and Gummadi, R.: IPNL: A NAT-Extended Internet Architecture, *ACM SIGCOMM Computer Communication Review*, Vol. 31, No. 4, pp. 69–80 (2001).

[151] 日高　稔，高瀬誠実，奥　智行：ネットワークプロセッサを用いたIPv6 over IPv4トンネル機能の評価，電子情報通信学会技術研究報告，CS2003-169, Vol. 103, No. 720, pp. 67–70 (2004).

[152] Lewis, E.: The Role of Wildcards in the Domain Name System, RFC 4592, IETF (2006).

[153] 柳沢信成，加藤尚樹，鈴木秀和，渡邊　晃：異なるプライベートアドレス空間端末の通信（CIPA）の提案，マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集，Vol. 2005, No. 6, pp. 369–372 (2005).

[154] 吉原貴仁，茂木信二，堀内浩規：ユビキタス・ネットワーク実現に向けたサービスゲートウェイの実装と評価，情報処理学会論文誌，Vol. 44, No. 12, pp. 3038–3049 (2003).

[155] 杣　信吾，古谷信司，佐藤浩司，横谷哲也，下笠　清：ホームゲートウェイによる情報家電連携サービスの検討，電子情報通信学会技術研究報告，CS2008-6, Vol. 108, No. 82, pp. 1–5 (2008).

[156] 古川隆弘, 滝澤基行, 湊　透, 島田裕一：NGNに向けたホームゲートウェイとプラットフォームの検討，電子情報通信学会技術研究報告，CS2006-41, Vol. 106, No. 304, pp. 29–34 (2006).

[157] 加藤尚樹，柳沢信成，鈴木秀和，宇佐見庄五，渡邊　晃：インターネットから家庭ネットワークへの接続を可能とするNATFプロトコルの検討と実装，情報学ワークショップ2005（WiNF2005）論文集，Vol. 3, pp. 142–146 (2005).

[158] 加藤尚樹，柳沢信成，鈴木秀和，渡邊　晃：アドレス空間の違いを意識しない通信を可能とするNATF（NAT Free protocol）の検討と実装，マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集，Vol. 2005, No. 6, pp. 373–376 (2005).

[159] OpenBSD: PF: The OpenBSD Packet Filter. http://www.openbsd.org/faq/pf/

[160] Network Working Group: Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods, RFC 1001, IETF (1987).

[161] Network Working Group: Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detail Specifications, RFC 1002, IETF (1987).

[162] 井戸上彰，久保　健，横田英俊：プライベートアドレスを使用するモバイルネットワーク間のローミング手順とその実装，情報処理学会論文誌，Vol. 44, No. 12, pp. 2958–2967 (2003).

[163] Digital Living Network Alliance: *DLNA Networked Device Interoperability Guidelines Expanded* (2006). http://www.dlna.org/

[164] Oh, Y.-J., Lee, H.-K., Kim, J.-T., Paik, E.-H. and Park, K.-R.: Design of an Extended Architecture for Sharing DLNA Compliant Home Media from Outside the Home, *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 2, pp. 542–547 (2007).

[165] 茂木信二，田坂和之，テープウィロージャナポンニワット，堀内浩規：情報家電の広域DLNA通信方式の提案，電子情報通信学会技術研究報告，NS2007-13, Vol. 107, No. 6, pp. 71–76 (2007).

[166] Kivinen, T. and Kojo, M.: More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), RFC 3526, IETF (2003).

[167] NLANR/DAST: Iperf - The TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/projects/Iperf/

[168] Devarapalli, V., Wakikawa, R., Petrescu, A. and Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, RFC 3963, IETF (2005).

[169] Leung, K., Dommety, G., Narayanan, V. and Petrescu, A.: Network Mobility (NEMO) Extensions for Mobile IPv4, RFC 5177, IETF (2008).

[170] 坂本順一，鈴木秀和，渡邊　晃：ネットワーク単位の移動透過性を実現するMobile NPCの実装と評価，マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集 (II)，Vol. 2006, No. 6, pp. 821–824 (2006).

[171] 島慶　一，湧川隆次：WIDEプロジェクトと最新インターネット技術研究動向:3.WIDEプロジェクトにおけるIPv6モビリティ技術の研究開発，情報処理学会論文誌，Vol. 46, No. 8, pp. 879–886 (2005).

[172] 坂本順一：ネットワーク単位の移動透過性を実現するMobile NPCの提案と評価，名城大学修士論文，名城大学大学院理工学研究科 (2007).

[173] Soliman, H.: Mobile IPv6 support for dual stack Hosts and Routers (DSMIPv6), Internet-draft, IETF (2007). http://tools.ietf.org/id/draft-ietf-mip6-nemo-v4traversal-06.txt

[174] 寺澤圭史, 鈴木秀和, 渡邊　晃：IPv4/IPv6 混在環境で移動透過性を実現する Mobile PPC の検討, マルチメディア, 分散, 協調とモバイル（DICOMO2008）シンポジウム論文集, Vol. 2008, No. 1, pp. 1593–1599 (2008).

[175] 細尾幸宏, 鈴木秀和, 渡邊　晃：GSCIP の Windows への実装に関する検討, マルチメディア, 分散, 協調とモバイル（DICOMO2008）シンポジウム論文集, Vol. 2008, No. 1, pp. 616–621 (2008).

[176] 増田真也, 鈴木秀和, 渡邊　晃：IPv4/IPv6 混在環境における暗号通信方式の考察, マルチメディア, 分散, 協調とモバイル（DICOMO2005）シンポジウム論文集, Vol. 2005, No. 6, pp. 693–696 (2005).

[177] 金本綾子, 瀬下正樹, 竹内元規, 渡邊　晃：IPv6 環境での移動透過性を実現する Mobile PPCv6 の検討, 平成 17 年度電気関係学会東海支部連合大会論文集 (2005).

[178] Boneh, D., Gentry, C. and Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys, *Proceesings of CRYPTO 2005*, Lecture Notes in Computer Science, Vol. 3621, Springer Berlin, pp. 258–275 (2005).

[179] Halevy, D. and Shamir, A.: The LSD Broadcast Encryption Scheme, *Proceesings of CRYPTO 2002*, Lecture Notes in Computer Science, Vol. 2442, Springer Berlin, pp. 145–161 (2002).

[180] 後藤裕司, 鈴木秀和, 渡邊　晃：NAT を越えてグループ通信が可能な拡張 DPRP の提案, マルチメディア, 分散, 協調とモバイル（DICOMO2008）シンポジウム論文集, Vol. 2008, No. 1, pp. 593–600 (2008).

[181] 鈴木秀和, 渡邊　晃：NAT-f を用いたホームネットワーク間相互接続方式の検討, マルチメディア, 分散, 協調とモバイル（DICOMO2008）シンポジウム論文集, Vol. 2008, No. 1, pp. 1675–1682 (2008).

[182] Sun Microsystems: MySQL. http://www.mysql.com/

[183] The Apache Software Foundation: Apache. http://www.apache.org/

[184] キーストリーム株式会社：技術 1：省電力チップとは? http://www.keystream.co.jp/tech/

[185] Cohen, J. and Aggarwal, S.: General Event Notification Architecture Base, Internet-draft, IETF (1998). http://news.gnus.org/internet-drafts/draft-cohen-gena-p-base-01.txt

[186] Goland, Y. Y., Cai, T., Leach, P., Gu, Y. and Albright, S.: Simple Service Discovery Protocol/1.0 Operating without an Arbiter, Internet-draft, IETF (1999). ftp://ftp.pwg.org/pub/pwg/ipp/new_SSDP/draft-cai-ssdp-v1-03.txt

[187] W3C: SOAP Specifications. http://www.w3.org/TR/soap/

[188] Debique, K., Igarashi, T., Kou, S., Moonen, J., Ritchie, J., Schults, G. and Walker, M.: *Content-Directory:1 Service Template Version 1.01*, UPnP Forum (2002).
http://www.upnp.org/standardizeddcps/documents/ContentDirectory1.0.pdf

[189] 武藤大悟，吉永　努：ルールベースアクセス制御機能を持つ DLNA 情報家電の遠隔共有支援機構，情報処理学会論文誌，Vol. 49, No. 12, pp. 3985–3996 (2008).

[190] 小山卓視，呉　敬源，武藤大悟，吉永　努：Mobile‐Wormhole Device：DLNA 情報家電の相互遠隔接続支援機構の携帯端末への応用，情報処理学会研究報告，2008-UBI-017, Vol. 2008, pp. 1–8 (2008).

[191] Haruyama, T., Mizuno, S., Kawashima, M. and Mizuno, O.: Dial-to-Connect VPN System for Remote DLNA Communication, *Proceedings of 5th IEEE Consumer Communications and Networking Conference* (*CCNC2008*), pp. 1224–1225 (2008).

[192] 春山敬宏，水野伸太郎，山田孝二，水野　修：VPN を介した情報家電サービス利用方式の提案，情報処理学会研究報告，2006-UBI-012, Vol. 2006, pp. 1–6 (2006).

[193] Kim, T., Oh, Y. J., Lee, H. K., Paik, J. E. H. and Park, K. R.: Implementation of the DLNA Proxy System for Sharing Home Media Contents, *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 1, pp. 139–144 (2007).

[194] 吉川　貴，三宅基治，竹下　敦：モバイル連携ホームゲートウェイシステム，情報処理学会研究報告，2006-ITS-027, Vol. 2006, pp. 97–102 (2006).

[195] Oh, Y. J., Lee, H. K., Kim, J. T., Paik, E. H. and Park, K. R.: Design of an Extended Architecture for Sharing DLNA Compliant Home Media from Outside the Home, *IEEE Transactions on Consumer Electronics*, Vol. 53, No. 2, pp. 542–547 (2007).

[196] 小川将弘，早川裕志，小板隆浩，佐藤健哉：グローバルネットワーク環境における UPnP 機器連携の実現，マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol. 2007, pp. 125–133 (2007).

[197] Venkitaraman, N.: Wide-Area Media Sharing with UPnP/DLNA, *Proceedings of 5th IEEE Consumer Communications and Networking Conference* (*CCNC2008*), pp. 294–298 (2008).

研究業績

○印は本論文に関係ある文献及び口頭発表を示す.

学術論文（査読あり）

○ 1. 鈴木秀和, 渡邊　晃, “プライベートネットワーク内のノードを通信相手とした移動透過性の実現方式,” 電子情報通信学会論文誌 (B), Vol.J92-B, No.1, pp.109–121, Jan. 2009.

○ 2. 金本綾子, 鈴木秀和, 伊藤将志, 渡邊　晃, “IPv4 移動体通信システムにおけるパケットロスレスハンドオーバの提案,” 情報処理学会論文誌, Vol.50, No.1, pp.133–143, Jan. 2009.

3. 播磨宏和, 伊藤将志, 鈴木秀和, 岡崎直宣, 渡邊　晃, “L2-based IP トレースバック方式の提案と実装,” 情報処理学会論文誌, Vol.49, No.6, pp.2200–2211, Jun. 2008.

○ 4. 鈴木秀和, 宇佐見庄五, 渡邊　晃, “外部動的マッピングにより NAT 越え通信を実現する NAT-f の提案と実装,” 情報処理学会論文誌, Vol.48, No.12, pp.3949–3961, Dec. 2007.

5. 竹尾大輔, 伊藤将志, 鈴木秀和, 岡崎直宣, 渡邊　晃, “コネクションベース方式による踏み台攻撃検出手法の提案,” 情報処理学会論文誌, Vol.48, No.2, pp.644–655, Feb. 2007.

○ 6. 竹内元規, 鈴木秀和, 渡邊　晃, “エンドエンドで移動透過性を実現する Mobile PPC の提案と実装,” 情報処理学会論文誌, Vol.47, No.12, pp.3244–3257, Dec. 2006.

○ 7. 鈴木秀和, 渡邊　晃, “フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装と評価,” 情報処理学会論文誌, Vol. 47, No. 11, pp.2976–2991, Nov. 2006.（推薦論文）

○ 8. 増田真也, 鈴木秀和, 岡崎直宣, 渡邊　晃, “NAT やファイアウォールと共存できる暗号通信方式 PCCOM の提案と実装,” 情報処理学会論文誌, Vol.47, No.7, pp.2258–2266, Jul. 2006.

国際会議（査読あり）

○ 1. H. Suzuki and A. Watanabe, “Design of NAT Traversal for Mobile PPC Applying Hole Punching Technology,” Proceedings of the IEEE International Region 10 Conference 2008 (TENCON2008), O28-2, Hyderabad, India, Nov. 2008.

2. Y. Miyazaki, H. Suzuki and A. Watanabe, “Proposal of a NAT Traversal System Independent of User Terminals and its Implementation,” Proceedings of the IEEE International Region 10 Conference 2008 (TENCON2008), P16-10, Hyderabad, India, Nov. 2008.

○ 3. K. Imamura, H. Suzuki and A. Watanabe, “A Proposal for a Remote Access Method using GSCIP and IPsec,” Proceedings of the IEEE International Region 10 Conference 2007 (TENCON2007), WeCM-O4.2, 219, pp.1–4, Taipei, Taiwan, Oct. 2007.

○ 4. Y. Goto, H. Suzuki and A. Watanabe, “Researches on Extended Dynamic Process Resolution Protocol that Can Traverse NAT,” Proceedings of the IEEE International Region 10 Conference 2007 (TENCON2007), ThCN-O7.6, 206, pp.1–4, Taipei, Taiwan, Oct. 2007.

5. Y. Miyazaki, H. Suzuki and A. Watanabe, “A Proposal for a NAT Traversal System that Does Not Require Additional Functions at Terminals,” Proceedings of the IEEE International Region 10 Conference 2007 (TENCON2007), FrCN-O12.2, 262, pp.1–4, Taipei, Taiwan, Oct. 2007.

○ 6. H. Suzuki, Y. Goto and A. Watanabe, “External Dynamic Mapping Method for NAT Traversal,” Proceedings of the IEEE 7th International Symposium on Communications and Information Technologies 2007 (ISCIT2007), pp.723–728, Sydney, Australia, Oct. 2007.

○ 7. C. Shu, H. Suzuki and A. Watanabe, “Proposal of an Authentication Method “SPAIC” using a Non-contact Type IC Card,” Proceedings of the IEEE 7th International Symposium on Communications and Information Technologies 2007 (ISCIT2007), pp.1470–1475, Sydney, Australia, Oct. 2007.

○ 8. H. Suzuki and A. Watanabe, “Implementation and Evaluation of Dynamic Process Resolution Protocol Actualizing Location Transparency,” Proceedings of The 2006 International Symposium on Information Theory and its Applications (ISITA2006), pp.284–289, Seoul, Korea, Oct. 2006.

○ 9. K. Enomoto, H. Suzuki and A. Watanabe, “Researches on Mobile Communications over a Private Address Area and a Global Address Area,” Proceedings of The 2006 International Symposium on Information Theory and its Applications (ISITA2006), pp.635–637, Seoul, Korea, Oct. 2006.

○ 10. S. Masuda, H. Suzuki, N. Okazaki and A. Watanabe, “Proposal for a Practical Cipher Communication Protocol That Can Coexist with NAT and Firewalls,” Proceedings of The International Conference on Information Networking (ICOIN2006), LNCS, Vol.3961, pp.713–722, Sendai, Japan, Jan. 2006.

国内会議（査読あり）

1. 宮﨑　悠, 鈴木秀和, 渡邊　晃, “端末に依存しない NAT 越えシステムの提案と実装,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.587–592, Jul. 2008.

○ 2. 後藤裕司, 鈴木秀和, 渡邊　晃, “NAT を越えてグループ通信が可能な拡張 DPRP の提案,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.593–600, Jul. 2008.

○ 3. 細尾幸宏, 鈴木秀和, 渡邊　晃, “GSCIP の Windows への実装に関する検討,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.616–6621, Jul. 2008.

○ 4. 今村圭祐, 鈴木秀和, 後藤裕司, 渡邊　晃, “セキュア通信アーキテクチャ GSCIP を実現するグループ管理サーバの実装と運用評価,” マルチメディア, 分散, 協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.1516–1522, Jul. 2008.

○ 5. 張　冰冰, 鈴木秀和, 渡邊　晃, “プロキシ中継型 Mobile PPC の検討,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.1588–1592, Jul. 2008.

○ 6. 寺澤圭史, 鈴木秀和, 渡邊　晃, “IPv4/IPv6 混在環境で移動透過性を実現する Mobile PPC の検討,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.1593–1599, Jul. 2008.

○ 7. 鈴木秀和, 渡邊　晃, “NAT-f を用いたホームネットワーク間相互接続方式の検討,” マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム論文集, Vol.2008, No.1, pp.1675–1682, Jul. 2008.

8. 宮﨑　悠, 鈴木秀和, 渡邊　晃, “端末の機能追加が不要な NAT 越え方式の提案,” マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.409–413, Jun. 2007.

○ 9. 今村圭祐, 鈴木秀和, 渡邊　晃, “GSCIP と IPsec を併用したリモートアクセス方式の提案と評価,” マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.468–472, Jun. 2007.

○ 10. 鈴木秀和, 金本綾子, 渡邊　晃, “NAT-f の移動透過通信への拡張,” マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.857–865, Jun. 2007.

○ 11. 金本綾子, 鈴木秀和, 渡邊　晃, “Mobile PPC におけるパケットロスレスハンドオーバの提案,” マルチメディア, 分散, 協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.866–872, Jun. 2007.

○ 12. 束　長俊, 鈴木秀和, 渡邊　晃, “非接触型 IC カードを用いた認証方式 SPAIC の提案,” マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.1332–1337, Jun. 2007.

○ 13. 後藤裕司, 鈴木秀和, 渡邊　晃, “NAT 越えを可能にする DPRP の検討,” マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集, Vol.2007, No.1, pp.1373–1377, Jun. 2007.

○ 14. 鈴木秀和, 渡邊　晃, “アドレス空間透過性を実現する NAT-f の実装と評価,” マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集, Vol.2006, No.6, pp.453–456, Jul. 2006.

○ 15. 榎本万人, 鈴木秀和, 坂本順一, 渡邊　晃, “プライベートアドレス空間とグローバルアドレス空間を跨る移動透過性の検討,” マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集, Vol.2006, No.6, pp.813–816, Jul. 2006.

○ 16. 坂本順一, 鈴木秀和, 渡邊　晃, “ネットワーク単位の移動透過性を実現する Mobile NPC の実装と評価,” マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム論文集, Vol.2006, No.6, pp.821–824, Jul. 2006.

○ 17. 竹内元規, 鈴木秀和, 渡邊　晃, “エンドエンドで移動透過性を実現する Mobile PPC の実装と評価,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.125–128, Jul. 2005.

○ 18. 坂本順一, 鈴木秀和, 竹内元規, 渡邊　晃, “Mobile PPC を利用したネットワーク単位の移動通信の提案,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.133–136, Jul. 2005.

○ 19. 柳沢信成, 加藤尚樹, 鈴木秀和, 渡邊　晃, “異なるプライベートアドレス空間端末の通信（CIPA）の提案,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.369–372, Jul. 2005.

○ 20. 加藤尚樹, 柳沢信成, 鈴木秀和, 渡邊　晃, “アドレス空間の違いを意識しない通信を可能とする NATF（NAT Free protocol）の検討と実装,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.373–376, Jul. 2005.

○ 21. 鈴木秀和, 竹内元規, 加藤尚樹, 増田真也, 渡邊　晃, “フレキシブルプライベートネットワークを実現するセキュア通信アーキテクチャ GSCIP の提案,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.441–444, Jul. 2005.

○ 22. 増田真也, 鈴木秀和, 渡邊　晃, “IPv4/IPv6 混在環境における暗号通信方式の考察,” マルチメディア，分散，協調とモバイル（DICOMO2005）シンポジウム論文集, Vol.2005, No.6, pp.693–696, Jul. 2005.

研究会・大会等（査読なし）

1. 平田祐二, 鈴木秀和, 渡邊　晃, “ボットネットによる不正メールの送信を防止するための検討,” 平成 20 年度電気関係学会東海支部連合大会論文集, O-077, Sep. 2008.

○ 2. 水谷智大, 鈴木秀和, 渡邊　晃, “Mobile PPC における仮想インタフェースの提案,” 平成 20 年度電気関係学会東海支部連合大会論文集, O-147, Sep. 2008.

○ 3. 近藤千華, 鈴木秀和, 渡邊　晃, “宅外モバイル機器の移動透過性を可能とする遠隔 DLNA 通信方式の検討,” 平成 20 年度電気関係学会東海支部連合大会論文集, O-204, Sep. 2008.

○ 4. 三浦健吉, 鈴木秀和, 渡邊　晃, “NAT-f を利用した SIP の NAT 越え通信の検討,” 平成 20 年度電気関係学会東海支部連合大会論文集, O-499 Sep. 2008.

○ 5. 鈴木秀和, 渡邊　晃, “Hole Punching を用いた NAT 越え Mobile PPC の設計,” 情報処理学会研究報告, 2008-MBL-45, Vol.2008, No.44, pp.69–74, May. 2008.

○ 6. 金本綾子, 鈴木秀和, 渡邊　晃, “端末移動時におけるパケットロスレスハンドオーバの提案,” 情報処理学会研究報告, 2008-MBL-44, Vol.2008, No.18, pp.91–98, Mar. 2008.

○ 7. 細尾幸宏, 鈴木秀和, 渡邊　晃, “GSCIP の Windows への実装に関する検討,” 情報処理学会第 70 回全国大会講演論文集, 3P-3, pp.1-185–1-186, Mar. 2008.

○ 8. 寺澤圭史, 鈴木秀和, 渡邊　晃, “IPv4/IPv6 混在環境における Mobile PPC の検討,” 情報処理学会第 70 回全国大会講演論文集, 6Z-2, pp.3-247–3-248, Mar. 2008.

○ 9. 張　冰冰, 鈴木秀和, 渡邊　晃, “プロキシ中継型 Mobile PPC の検討,” 情報処理学会第 70 回全国大会講演論文集, 6Z-3, pp.3-249–3-250, Mar. 2008.

○ 10. 細尾幸宏, 鈴木秀和, 渡邊　晃, “GSCIP の Windows への実装に関する検討,” 平成 19 年度電気関係学会東海支部連合大会論文集, O-308, Sep. 2007.

○ 11. 張　冰冰, 鈴木秀和, 渡邊　晃, “プロキシ中継型 Mobile PPC の検討,” 平成 19 年度電気関係学会東海支部連合大会論文集, O-312, Sep. 2007.

○ 12. 寺澤圭史, 鈴木秀和, 渡邊　晃, “IPv4/IPv6 混在環境における Mobile PPC の検討,” 平成 19 年度電気関係学会東海支部連合大会論文集, O-313, Sep. 2007.

13. 三根健司, 鈴木秀和, 渡邊　晃, “Windows API の監視による未知ウイルス検出手法の検討,” 平成 19 年度電気関係学会東海支部連合大会論文集, O-372, Sep. 2007.

14. 間宮領一, 鈴木秀和, 渡邊　晃, “ボットネットによるスパムメール送信防止方法の検討,” 平成 19 年度電気関係学会東海支部連合大会論文集, O-373, Sep. 2007.

○ 15. 束　長俊, 鈴木秀和, 渡邊　晃, “非接触型 IC カードを用いた重要情報の配送方式 SPAIC の提案,” 電子情報通信学会 2007 年総合大会講演論文集, A-7-9, p.213, Mar. 2007.

○ 16. 今村圭祐, 鈴木秀和, 渡邊　晃, “GSCIP と IPsec を併用したリモートアクセス方式の提案,” 電子情報通信学会 2007 年総合大会講演論文集, B-7-134, p.224, Mar. 2007.

○ 17. 後藤裕司, 鈴木秀和, 渡邊　晃, “NAT 越えが可能な DPRP の検討,” 電子情報通信学会 2007 年総合大会講演論文集, B-7-137, p.227, Mar. 2007.

18. 宮﨑　悠, 鈴木秀和, 渡邊　晃, “端末の機能追加が不要な NAT 越え方式の提案,” 電子情報通信学会 2007 年総合大会講演論文集, B-7-199, p.289, Mar. 2007.

○ 19. 佐本章悟, 鈴木秀和, 渡邊　晃, “GSCIP における構成要素 GEA の検討,” 平成 18 年度電気関係学会東海支部連合大会論文集, O-427, Sep. 2006.

20. 宮﨑　悠, 鈴木秀和, 渡邊　晃, “端末の改造が不要な NAT 越え方式の提案,” 平成 18 年度電気関係学会東海支部連合大会論文集, O-429, Sep. 2006.

○ 21. 後藤裕司, 鈴木秀和, 渡邊　晃, “グローバルアドレスとプライベートアドレス空間を跨る DPRP の検討,” 情報処理学会第 68 回全国大会講演論文集, 3R-3, pp.3-609–3-610, Mar. 2006.

○ 22. 榎本万人, 鈴木秀和, 坂本順一, 渡邊　晃, “プライベートアドレス空間とグローバルアドレス空間を跨る移動通信の検討,” 情報処理学会第 68 回全国大会講演論文集, 5R-3, pp.3-645–3-646, Mar. 2006.

○ 23. 加藤尚樹, 柳沢信成, 鈴木秀和, 宇佐見庄五, 渡邊　晃, “インターネットから家庭ネットワークへの接続を可能とする NATF プロトコルの検討と実装,” 情報学ワークショップ 2005（WiNF2005）論文集, Vol.3, pp.142–146, Sep. 2005.

○ 24. 竹内元規, 鈴木秀和, 渡邊　晃, “移動通信プロトコル Mobile PPC の実装とその評価,” 平成 17 年度電気関係学会東海支部連合大会論文集, O-225, Sep. 2005.

○ 25. 坂本順一, 鈴木秀和, 竹内元規, 渡邊　晃, “ネットワーク単位の移動透過性を実現する Mobile NPC とその実装,” 平成 17 年度電気関係学会東海支部連合大会論文集, O-227, Sep. 2005.

○ 26. 鈴木秀和, 渡邊　晃, “動的処理解決プロトコル DPRP の性能評価,” 平成 17 年度電気関係学会東海支部連合大会論文集, O-229, Sep. 2005.

○ 27. 榎本万人, 坂本順一, 鈴木秀和, 渡邊　晃, “異なるアドレス空間を跨る移動通信の検討,” 平成 17 年度電気関係学会東海支部連合大会論文集, O-230, Sep. 2005.

○ 28. 後藤裕司, 鈴木秀和, 渡邊　晃, “異なるアドレス空間をまたがる DPRP の検討,” 平成 17 年度電気関係学会東海支部連合大会論文集, O-231, Sep. 2005.

○ 29. 鈴木秀和, 渡邊　晃, “フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装,” 情報処理学会研究報告, 2004-CSEC-028, Vol.2005, No.33, pp.199–204, May. 2005.

○ 30. 加藤尚樹, 柳沢信成, 鈴木秀和, 渡邊　晃, “アドレス空間の違いを意識しない通信方式 NATF の提案と実装,” 情報処理学会研究報告, 2004-DPS-122, Vol.2005, No.33, pp.351–356, May. 2005.

○ 31. 柳沢信成, 加藤尚樹, 鈴木秀和, 渡邊　晃, “グローバルアドレス環境を挟んだプライベートアドレス端末同士の通信の提案と実装,” 情報処理学会研究報告, 2004-DPS-122, Vol.2005, No.33, pp.357–362, May. 2005.

○ 32. 竹内元規, 鈴木秀和, 渡邊　晃, “モバイル端末の移動透過性を実現する Mobile PPC の実装,” 情報処理学会研究報告, 2004-MBL-032, Vol.2005, No.28, pp.29–35, May. 2005.

○ 33. 坂本順一, 鈴木秀和, 竹内元規, 渡邊　晃, “Mobile PPC を利用したネットワーク単位の移動通信の提案,” 情報処理学会第 67 回全国大会講演論文集, 5U-7, pp.3-649–3-650, Mar. 2005.

○ 34. 坂本順一, 鈴木秀和, 竹内元規, 渡邊　晃, “Mobile P2P を利用した移動ネットワークの提案,” 平成 16 年度電気関係学会東海支部連合大会論文集, O-382, Sep. 2004.

○ 35. 鈴木秀和, 渡邊　晃, “フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の仕組み,” 情報処理学会研究報告, 2004-CSEC-026, Vol.2004, No.75, pp.259–266, Jul. 2004.

○ 36. 鈴木秀和, 渡邊　晃, “GSCIP を構成する DPRP の仕組みの検討,” 情報処理学会第 66 回全国大会講演論文集, 5V-1, pp.3-479–3-480, Mar. 2004.

○ 37. 鈴木秀和, 渡邊　晃, “イントラネットに柔軟な閉域通信グループを実現する動的処理解決プロトコル DPRP の検討,” 平成 15 年度電気関係学会東海支部連合大会論文集, 359, pp.180, Oct. 2003.

外部資金獲得実績

1. 2008 年度　日本学術振興会特別研究員 DC2（2 年間）
   研究課題：安全性と柔軟性を両立するフレキシブルプライベートネットワークの実現
   科学研究費補助金（特別研究員奨励費）：20・1069

受賞歴

1. 2008 年 7 月　マルチメディア，分散，協調とモバイル（DICOMO2008）シンポジウム　ヤングリサーチャー賞

2. 2007 年 6 月　マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム　ヤングリサーチャー賞

3. 2006 年 7 月　マルチメディア，分散，協調とモバイル（DICOMO2006）シンポジウム　松下温賞（最優秀プレゼンテーション賞）

4. 2006 年 5 月　情報処理学会東海支部　学生論文奨励賞

5. 2006 年 2 月　電気関係学会東海支部連合大会　IEEE Nagoya Section Student Paper Award

展示会（審査あり）

1. 2008 年 9 月 16 日～18 日　イノベーション・ジャパン 2008
   東京国際フォーラムで開催された国内最大規模の産学マッチング・フェア（独立行政法人科学技術振興機構，独立行政法人新エネルギー・産業技術総合開発機構主催）．研究テーマ「ユビキタス社会を実現するフレキシブルプライベートネットワークの研究」として出展し，Mobile PPC と NAT-f を融合したシステムのデモンストレーションを実施した．その結果，数社より機材提供や製品開発などに関する話を頂き，成果を上げることができた．

図 7.1　イノベーション・ジャパン 2008 出展ブースの様子

付 録A　表記法

本論文で用いる記号を表 A.1 に定義する．

表 A.1　本論文共通の記法

記号	意味
IP_n	ノード n の IP アドレス
${IP_n}^m$	$(m-1)$ 回移動した後のノード n の IP アドレス（m は正整数）
gIP_n	ノード n のグローバル IP アドレス
pIP_n	ノード n のプライベート IP アドレス
vIP_n	ノード n の仮想 IP アドレス
$A:p$	IP アドレス A，ポート番号 p の組
$FQDN_n$	ノード n の FQDN
HN_n	ノード n のホスト名
PHN_n	ノード n のプライベートホスト名
$proto$	プロトコルタイプ（TCP/UDP の区別）
CK	GSCIP におけるシステム共通鍵
GKx	GSCIP におけるグループ鍵（x は鍵番号）
DGK	DPRP により決定したグループ鍵
CKI	GSCIP におけるシステム共通鍵 CK の鍵情報
$GKIx$	GSCIP におけるグループ鍵 GKx の鍵情報
$DGKI$	DPRP により決定したグループ鍵 DGK の鍵情報
$SK_{m,n}$	DH 鍵交換によりノード m とノード n の間で生成される共通鍵
AK	Mobile PPC における認証鍵
$PrivKey_n$	ノード n の秘密鍵
$PubKey_n$	ノード n の公開鍵
CKY_n	Mobile PPC の認証処理においてノード n が生成する Cookie
R_n	Mobile PPC の認証処理においてノード n が生成する乱数値
T_n	Mobile PPC の認証処理において CKY_n が生成された時刻
$E_K(M)$	鍵 K を用いてメッセージ M を暗号化
$D_K(M)$	鍵 K を用いてメッセージ M を復号
$h(M)$	メッセージ M のハッシュ値
$M1 \parallel M2$	メッセージ $M1$ とメッセージ $M2$ の結合
$S \rightarrow D$，$D \leftarrow S$	S から D への通信
$S \leftrightarrow D$	S と D 間の通信
$S \Rightarrow D$	S から D への変化
$\{S \overset{T}{\Leftrightarrow} D\}$	変換テーブル T に基づく S から D，または D から S へのアドレス変換

付 録B　提案アーキテクチャの要素技術

B.1　NATの種類

NAT はアドレス変換の仕組みの違いに応じて，以下の 4 種類に分類できる．

1. Full Cone NAT

   プライベートネットワークに存在する IN が EN1 のポート番号 $d1$ に対して通信を開始すると，NAT は IN 側のトランスポートアドレス $pIP_{IN} : s$ に対して，ポート番号 $m1$ をマッピングする．Full Cone NAT は IN のポートと NAT のマッピングされたポートだけを 1 対 1 に対応させるため，EN 側のトランスポートアドレスはどんな値でも構わない．

   従って，EN1 からの応答パケット（図 B.1 (a)）はもちろんのこと，EN1 が異なる送信元ポート番号 $d2$ からマッピングアドレス $gIP_{NAT} : m1$ へパケットを送信すれば，IN のポート番号 s へ到達する（図 B.1 (b)）．また，EN1 とは関係ない異なるノード EN2 もマッピングアドレス $gIP_{NAT} : m1$ へパケット送信すれば，送信元ポート番号が何であれ IN へパケットを送信することができる（図 B.1 (c)，(d)）．

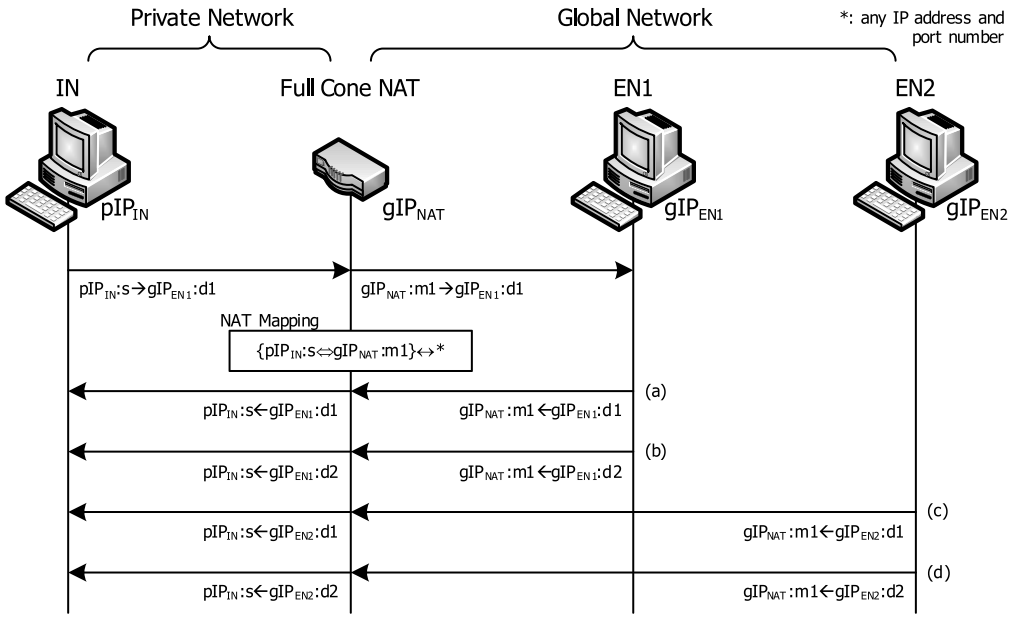

図 B.1　Full Cone NAT

2. Restricted Cone NAT

プライベートネットワークに存在する IN が EN1 のポート番号 $d1$ に対して通信を開始すると，NAT は IN 側のトランスポートアドレス $pIP_{IN}:s$ に対して，$gIP_{NAT}:m1$ をマッピングする．Restricted Cone NAT は IN のポートに加えて，EN の IP アドレス gIP_{EN1} を関連付けて，ポート番号 $m1$ をマッピングする．

従って，EN1 からの応答パケット（図 B.2 (a)）はもちろんのこと，EN1 が異なる送信元ポート番号 $d2$ からマッピングアドレス $gIP_{NAT}:m1$ へパケットを送信すれば，IN のポート番号 s へ到達する（図 B.2 (b)）．この仕組みは Full Cone NAT と全く同じである．ただし，EN1 とは関係ない異なるノード EN2 が上記マッピングアドレス $gIP_{NAT}:m1$ へパケット送信しても，NAT は EN 側の IP アドレスが異なるため IN へパケットを転送しない（図 B.2 (c-1)）．IN が EN2 に対して一度通信を開始すれば，EN2 に対するマッピングテーブルが追加生成されるため，その後は EN2 側から IN へパケットを送信することが可能になる（図 B.2 (c-2)，(d)）．ここで，IN から EN1 と EN2 への通信に着目すると，送信元トランスポートアドレスはいずれも $pIP_{IN}:s$ である．Restricted Cone NAT の場合，IN 側のトランスポートアドレスが同一のパケットに対しては新たなポートをマッピングしない．すなわち，IN から EN2 への通信に対しても，NAT は IN から EN1 への通信と同じポート番号 $m1$ をマッピングする．

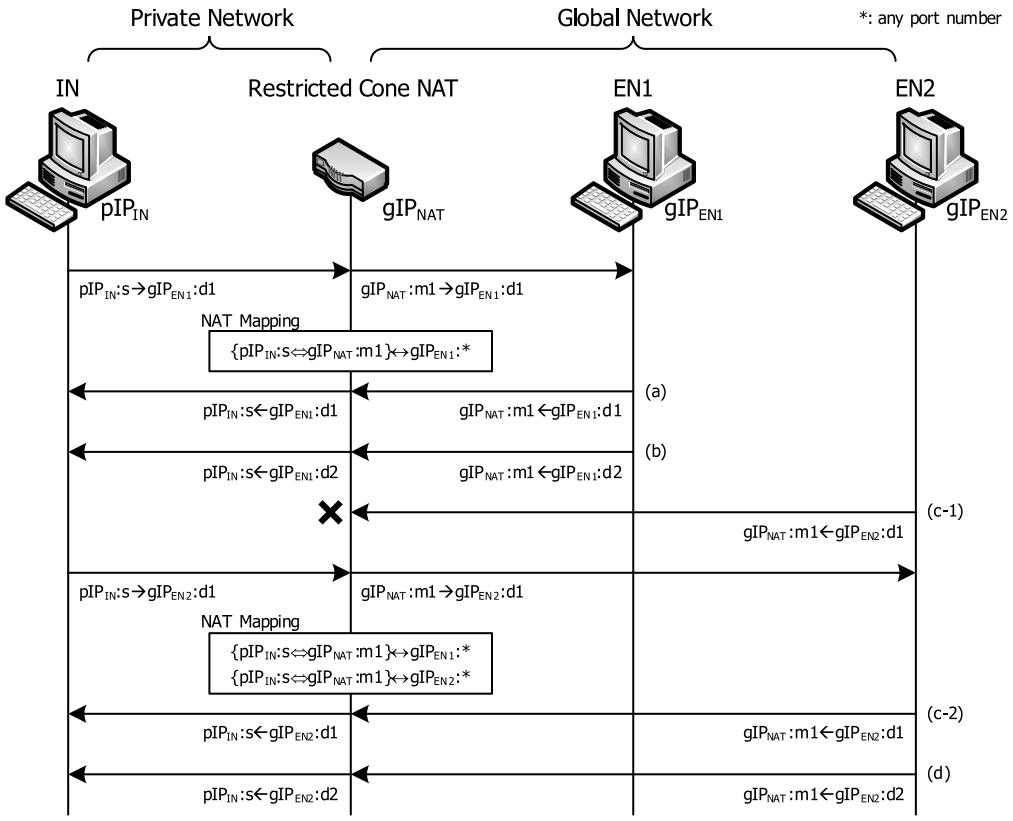

図 B.2　Restricted Cone NAT

3. Port Restricted NAT

プライベートネットワークに存在する IN が EN1 のポート番号 $d1$ に対して通信を開始すると，NAT は IN 側のトランスポートアドレス $pIP_{IN}:s$ に対して，$gIP_{NAT}:m1$ をマッピングする．Restricted Cone NAT は IN のポートに加えて，EN1 側のトランスポートアドレス $gIP_{EN1}:d1$ を関連付けて，ポート番号 $m1$ をマッピングする．

従って，EN1 からの応答パケット（図 B.3 (a)）は IN へ到達するが，EN1 が異なる送信元ポート番号 $d2$ からマッピングアドレス $gIP_{NAT}:m1$ へパケットを送信しても IN には届かない（図 B.3 (b)）．すなわち，Restricted Cone NAT からポート番号の制限が加えられたものといえる．

ここで，Port Restricted Cone NAT は Restricted Cone NAT と同様に，IN 側のトランスポートアドレスが同一のパケットに対しては新たなポートをマッピングしない．そのため，EN1 とは関係ない異なるノード EN2 が IN までパケットを送信したい場合，NAT にマッピングテーブルが生成されていない場合は届かず（図 B.3 (c-1)），生成されていれば IN まで転送される（図 B.3 (c-2)）．ただし，EN2 の送信元ポート番号が異なれば，やはり IN までパケットを送信することはできない（図 B.3 (d)）．

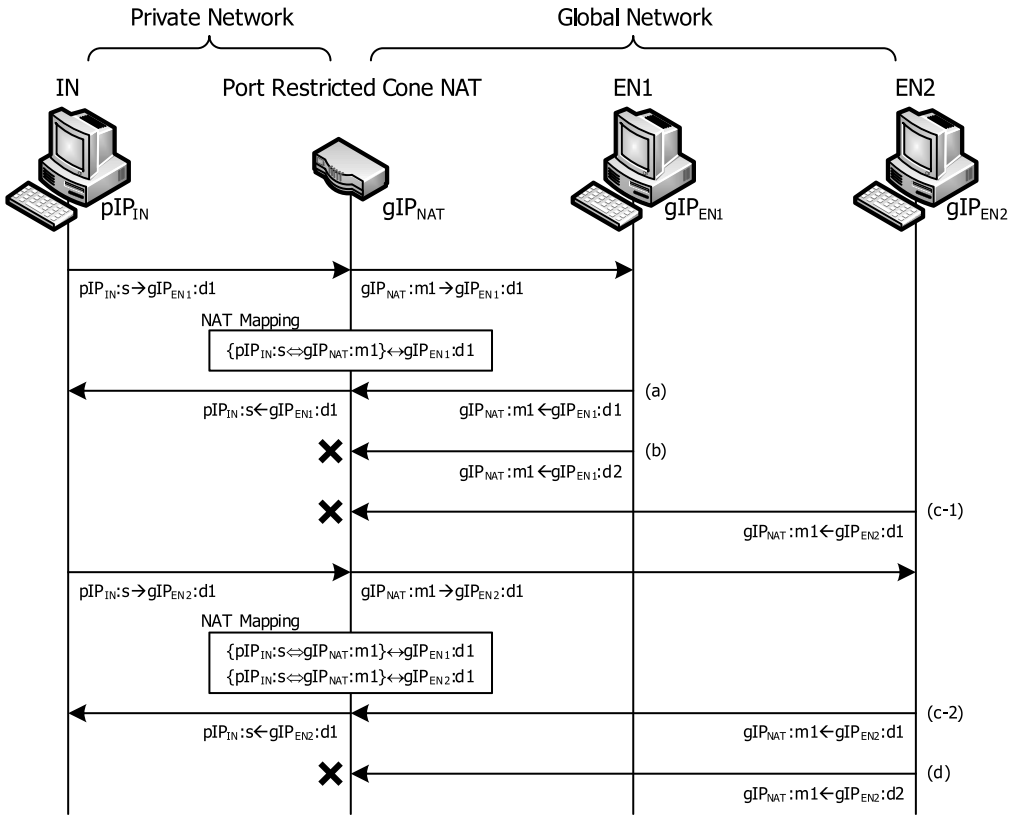

図 B.3　Port Restricted Cone NAT

4. Symmetric NAT

プライベートネットワークに存在する IN が EN1 のポート番号 $d1$ に対して通信を開始すると，NAT は IN 側のトランスポートアドレス $pIP_{IN} : s$ に対して，$gIP_{NAT} : m1$ をマッピングする．Symmetric NAT は Port Restricted Cone NAT と同様に，IN のポートに加えて EN1 側のトランスポートアドレス $gIP_{EN1} : d1$ を関連付けて，ポート番号 $m1$ をマッピングする．

Port Restricted Cone NAT と異なる点は，マッピングされるポート番号にある．Port Restricted Cone NAT は IN 側の送信元トランスポートアドレスが同一の通信に対しては同じポート番号 $m1$ をマッピングした．一方，Symmetric NAT では IN 側と EN 側のトランスポートアドレスのペアが完全に一致しない場合は，異なるポート番号 $m2$ をマッピングする．

従って，EN1 や EN2 からの応答パケット（図 B.4 (a-1)，(a-2)）は IN へ到達するが，その他のパケットは一切マッピングアドレスに送信しても IN には届かない（図 B.4 (b)，(c-1)，(c-2)，(d)）．

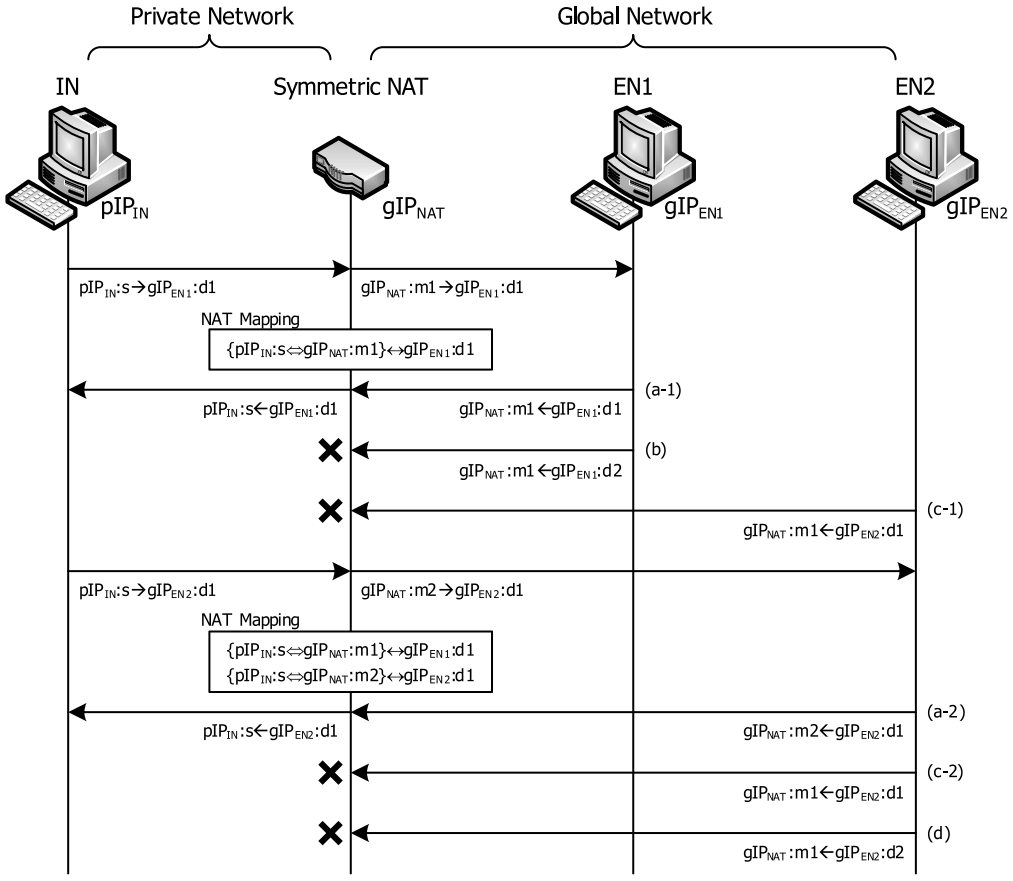

図 B.4　Symmetric NAT

B.2 Dynamic DNS

B.2.1 Dynamic DNS の仕組み

Dynamic DNS（DDNS）は動的に割り当てられる IP アドレスと，そのホスト名の対応を動的に登録，管理する仕組みである．一般にホームネットワークをインターネットに接続する場合，動的に変化する IP アドレスが契約ユーザに割り当てられる．このような環境において，ホームネットワーク内にサーバを設置し，各種サービスを提供する場合に DDNS の仕組みがよく利用されている．

DDNS は RFC2136 [111] で仕様が規定されており，最も一般的な DNS サーバアプリケーションである BIND [114] のバージョン 9 以降や，Microsoft 社の Windows 2000 Server の Active Directory などに実装されている．

図 B.5 に DDNS の基本的な動作を示す．ここでは，Host A が *example.com* ドメインを管理する DDNS サーバ（IP アドレス：198.76.29.20）に，ホスト名 *alice* に関する A レコードを登録・更新する場合を例に挙げ，その仕組みを示す．

(1) Host A は自身の FQDN “*alice.example.com*” に対する SOA（Start Of Authority）レコードをプライマリ DNS サーバに問い合わせる．Host A のプライマリ DNS サーバは再帰問い合わせを行う．

(2) *example.com* を管理する DDNS サーバは，*alice.example.com* の SOA レコードを返答する．

(3) Host A は SOA レコードに記載されている DDNS サーバの名前 “*ns.example.com*” を取得し，こ

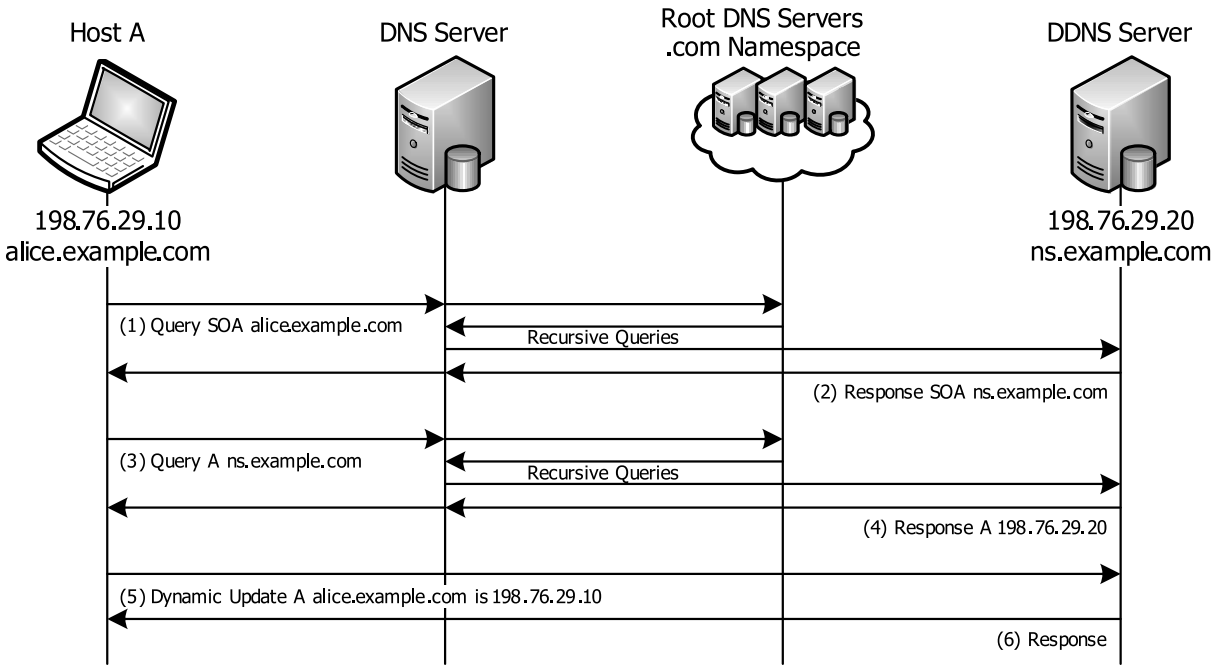

図 B.5　DDNS の動作概要（nsupdate）

れに対する A レコードを問い合わせる．

(4) DDNS サーバは，*ns.example.com* の A レコードを返答する．

(5) Host A は A レコードから DDNS サーバの IP アドレス（198.76.29.20）を取得し，自身の FQDN と IP アドレス（198.76.29.10）の関係を登録・更新するために update メッセージを送信する．

(6) update メッセージを受信した DDNS サーバは，自身のリソースレコードに通知された情報を登録し，結果を応答する．

上記の登録・更新の仕組みは DNS の update メッセージを利用したものであり，ホストが登録したい IP アドレスを通知する．従って，ホストがホームネットワークなどに存在し，プライベート IP アドレスを取得していた場合は，NAT に割り当てられたグローバル IP アドレスを知る必要がある．

そこで，update メッセージを利用しない方法として，図 B.6 のように DDNS サーバに WWW サービスを稼働させ，HTTP により登録・更新する方法がある．この方法は多くの DDNS サービスプロバイダが採用されており，登録したいホストは NAT の配下にいるのか，あるいは外側にいるのかを気にする必要がない．以下，DDNS サーバと WWW サーバは同一機器であるが，DNS と HTTP の違いにあわせて，表記を分ける．

(1) Host A は DDNS サービスプロバイダが運営する WWW サーバの FQDN “*www.example.com*” を問い合わせる．

(2) DDNS サーバは *www.example.com* の A レコードを返答する．

(3) Host A は A レコードから WWW サーバの IP アドレス（198.76.29.20）を取得し，HTTP でアクセスする．一般的に WWW サーバはホスト名（あるいはユーザ名）とパスワードを入力するためのフォームが用意してあり，ユーザは自信の名前とパスワードを入力し，ログインする．

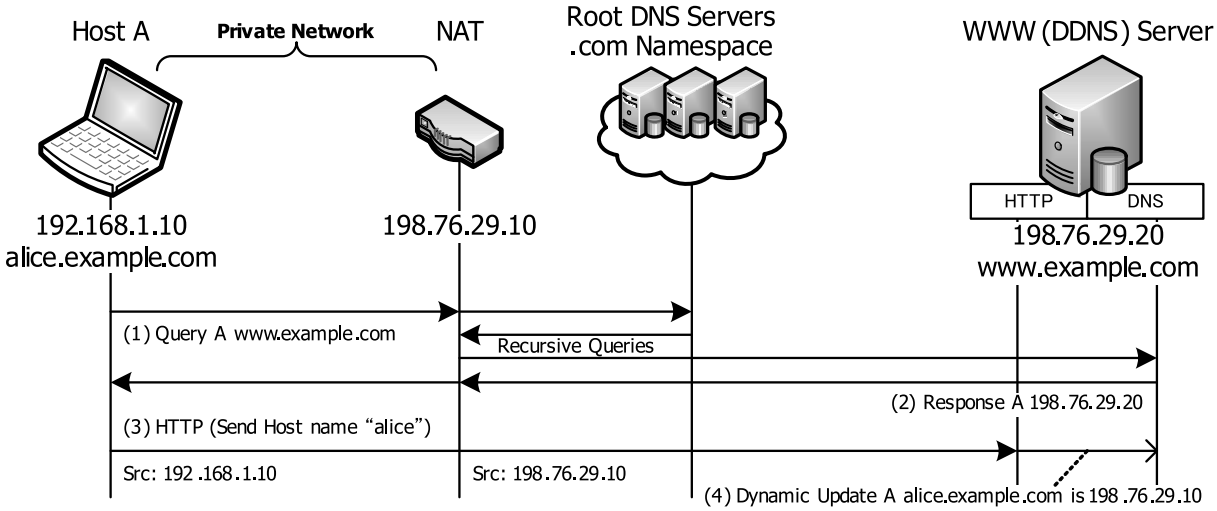

図 B.6　WWW サービスによる DDNS 登録・更新方法

(4) WWW サーバはユーザの認証に成功すると，入力された名前と受信した HTTP パケットの送信元 IP アドレス（198.76.29.10）を用いて，DDNS サーバに登録・更新の処理を行う．

以上のように，登録したいホストがプライベートネットワークに存在しても，DDNS サーバには必ず NAT のグローバル IP アドレスが登録される．

B.2.2 サービスプロバイダー一覧

提案アーキテクチャ（特に Mobile PPC と NAT-f）では，通信開始時に通信相手ノードの IP アドレスを解決するために DDNS サーバを利用する．提案アーキテクチャでは既に運用されている既存システムをそのまま利用することが可能である．ここでは A レコード，CNAME レコード，およびワイルドカード機能のサポートの有無に着目して，既存の DDNS サービスプロバイダの一部を紹介する．下記表中の “○” は当該機能を標準サポートしていることを，“△” は別途料金が発生することを意味している．

表 B.1 国内の DDNS サービスプロバイダ

サイト名	A	C	W	URL
Dynamic DO!.jp	○	—	△	http://ddo.jp/
マイドメイン	○	○	○	http://my-domain.jp/
JSpeed	○	○	○	http://ddns.j-speed.net/
Earth Dynamic DNS	○	—	○	http://mydns.to/
@nifty	△	—	—	http://domain.nifty.com/
ZENNO. コム	○	—	○	http://zenno.com/
pcc	○	—	△	http://pcc.jp/
MyDNS.jp	○	○	○	http://www.mydns.jp/
JPN.ch	○	—	○	http://jpn.ch/
VALUE-DOMAIN	○	○	—	https://www.value-domain.com/
お名前.com	○	○	—	http://www.onamae.com/option/ddns/
BIGLOBE	△	—	—	http://ddns.biglobe.ne.jp/
USA	○	—	—	http://www.usa.ne.jp/
CyberGate -DDNS-	○	—	—	http://cybergate.planex.co.jp/ddns/
Graphy	△	△	—	http://www.dnsalias.jp/
ieServer.Net	○	—	—	http://www.ieserver.net/
IvyNetwork	△	—	△	http://dp-21.net/
ぷらら	△	—	—	http://www.plala.or.jp/access/guest/dyndns/
So-net	△	—	—	http://www.so-net.ne.jp/ddns/

A：A レコード登録　C：CNAME レコード登録　W：ワイルドカード機能
○：サポート　△：有償サポート　—：非対応

表 B.2　海外の DDNS サービスプロバイダ

サイト名	A	C	W	URL
DynDNS	○	—	○	http://www.dyndns.com/
theBBS	○	—	—	http://thebbs.org/dns/
yi.org	○	—	○	http://www.yi.org/
DtDNS	○	—	○	http://www.dtdns.com/
No-IP	○	—	○	http://www.no-ip.com/
DynUp	○	—	—	http://www.dynup.net/
ChangeIP.com	○	—	—	http://www.changeip.com/
Dynamx	○	—	○	http://www.dynam.ac/
miniDNS	○	○	—	http://www.minidns.net/
3domain	○	○	○	http://www.3domain.hk/
Microtech	○	—	—	http://www.mtgsy.net/
ZoneEdit	○	○	—	http://www.zoneedit.com/
CJB.NET	○	○	—	http://www.cjb.net/
ODS	○	○	—	http://www.ods.org/
DNS2GO	○	△	○	http://dns2go.deerfield.com/
EveryDNS.net	○	○	○	http://www.everydns.com/
regfly	○	○	—	http://www.regfly.com/
eNom	○	○	—	http://www.enom.com/
ThatIP	△	△	—	http://www.thatip.com/
DNSExit	○	○	—	http://www.dnsexit.com/
yi.org	○	○	○	http://www.yi.org/

A：A レコード登録　C：CNAME レコード登録　W：ワイルドカード機能
○：サポート　△：有償サポート　—：非対応

B.3　Diffie-Hellman 鍵交換

Diffie-Hellman 鍵交換 [81] は，1976 年に Whitfield Diffie と Martin E. Hellman により提案された暗号プロトコルである．Diffie と Hellman は公開鍵暗号の概念を提案し，事前の秘密の共有なしに，盗聴の可能性がある通信路を使っても安全に暗号鍵を共有することが可能になった．DH 鍵交換により共有された暗号鍵は，共通鍵暗号方式の鍵として使用可能である．

B.3.1　概要

図 B.7 に DH 鍵交換の仕組みを示す．十分に大きな素数 p と $g \in \mathbb{Z}_p^*$ を用意し，それぞれ公開されているものとする（$\mathbb{Z}_p^*$ は乗法群）．ここで，Alice と Bob が暗号鍵を共有する場合を想定する．Alice と Bob は互いに秘密鍵 r_A，r_B を選択する．

$$r_A \in \mathbb{Z}_{p-1} \qquad r_B \in \mathbb{Z}_{p-1} \tag{B.1}$$

Alice と Bob は互いに公開鍵 u_A を計算して，これを互いに送信する．

$$u_A = g^{r_A} \bmod p \qquad u_B = g^{r_B} \bmod p \tag{B.2}$$

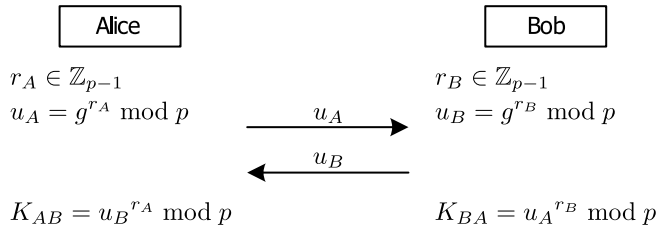

図 B.7　Diffie-Hellman 鍵交換

Alice と Bob は自身の秘密鍵と受信した相手の公開鍵から，以下の値を計算する．

$$K_{AB} = {u_B}^{r_A} \bmod p \qquad K_{BA} = {u_A}^{r_B} \bmod p \tag{B.3}$$

ここで，

$$K_{AB} = {u_B}^{r_A} \bmod p = g^{r_B \cdot r_A} \bmod p = g^{r_A \cdot r_B} \bmod p = {u_A}^{r_B} \bmod p = K_{BA} \tag{B.4}$$

となり，以後この $K_{AB} = K_{BA} = K$ を共通鍵暗号方式の鍵として使用する．

B.3.2　DH 鍵交換の安全性

盗聴に対する耐性

DH 鍵交換の安全性の概念として，

1. 秘密鍵 r_A，r_B の導出の困難性
2. 共有鍵 K の導出の困難性
3. 共有鍵 K の部分情報の導出の困難性

の 3 段階が考えられる．

第三者の Eve が Alice と Bob の通信を盗聴すると，Eve は p，g，u_A，u_A を取得することができる．ここで，p，g，u_A が与えられたとき，$u_A = g^{r_A} \bmod p$ を満たす r_A を求める問題を，離散対数問題（Discrete Logarithm Problem：DLP）という．例えば，p が 1024 bit 程度の素数であれば，DLP を解くには現行のスーパーコンピュータを用いても数百年以上という非現実的な時間がかかるといわれている．従って，秘密鍵 r_A，r_B の導出は困難であるといえる．

しかし DLP 問題が困難であるという仮定だけでは，DH 鍵交換は安全であるとは限らない．それは Eve が公開鍵 u_A，u_B から秘密鍵 r_A，r_B が計算できなくとも，$K = g^{r_A \cdot r_B} \bmod p$ を計算できる可能性があるためである．そこで，p，g，g^{r_A}，g^{r_B} が与えられたとき，$g^{r_A \cdot r_B}$ を求める問題を，計算 DH 問題（Computational Diffie-Hellman Problem：CDHP）という．p が十分に大きいとき，$g^{r_A \cdot r_B}$ を多項式時間で計算する方法は現在のところ存在しない．従って，共有鍵 K の導出は困難であるといえる．

最後に，p，g，g^{r_A}，g^{r_B}，g^{r_C} が与えられたとき，$g^{r_C} = g^{r_A \cdot r_B}$ か否かを識別する問題を，判定 DH 問題（Decisional Diffie-Hellman Problem：DDHP）という．CDHP では $g^{r_A \cdot r_B}$ の値を完全に求める

ことであったが，DDHP では $g^{r_A \cdot r_B}$ の値の部分情報，つまり 1 bit さえ求めればよいことを意味する．DDHP を解く効率的なアルゴリズムが存在しないという仮定のことを，DDH 仮定という．

以上のことから，DH 鍵交換は DDH 仮定が正しければ，受動的攻撃である盗聴に対して安全であるといえる．

中間者攻撃に対する耐性

DH 鍵交換は能動的攻撃である中間者攻撃に対しては安全でないことがわかっている．受動的攻撃はメッセージの盗聴のみを行うが，能動的攻撃は盗聴に加えてメッセージの改ざんなども行う．

図 B.8 に中間者攻撃について示す．Eve は Alice と Bob の通信に割り込んで，両者が交換する公開鍵 u_A，u_B を自身の公開鍵 u_E にすり替える．この結果，Alice と Bob は自身の秘密鍵 r_A，R_B と受信した公開鍵 u_E を用いて，暗号鍵

$$K_{AE} = {u_E}^{r_A} \bmod p \qquad K_{BE} = {u_E}^{r_B} \bmod p \tag{B.5}$$

を生成する．一方，Eve も同様に暗号鍵

$$K_{EA} = {u_A}^{r_E} \bmod p \qquad K_{EB} = {u_B}^{r_E} \bmod p \tag{B.6}$$

を生成する．ここで，

$$K_{AE} = {u_E}^{r_A} \bmod p = g^{r_E \cdot r_A} \bmod p = g^{r_A \cdot r_E} \bmod p = {u_A}^{r_E} \bmod p = K_{EA} \tag{B.7}$$

が成立するため，Alice は Eve と暗号鍵を共有したことになる．同様に $K_{BE} = K_{EB}$ より，Bob も Eve と暗号鍵を共有したことになる．従って，Alice と Bob 間の通信は暗号化されても Eve に解読されてしまう．

このような中間者攻撃を防ぐためには，Alice が Bob 本人と DH 鍵交換を行っていることを証明すればよい．何らかの手段で通信相手の認証を行う鍵交換を，認証鍵交換（Authenticated Key Exchange：AKE）と呼ぶ．AKE を実現する方式として，PKI（Public Key Infrastructure）を利用する PKI ベース AKE，パスワードを用いるパスワード AKE や，ID ベース暗号を用いた ID ベース AKE などが研究されている．

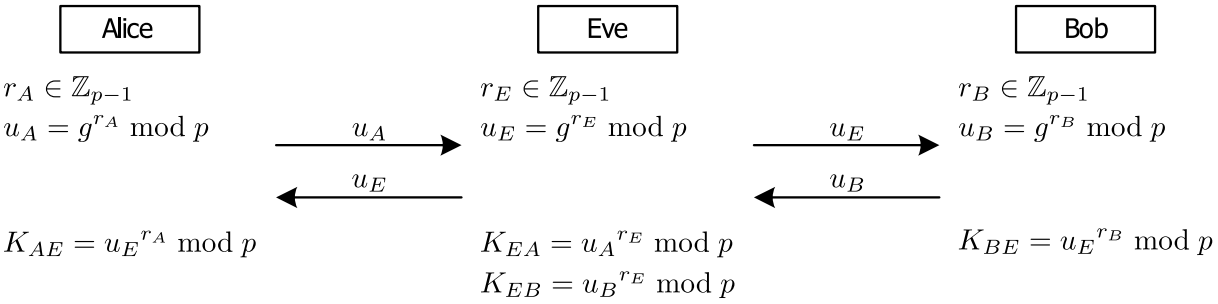

図 B.8　DH 鍵交換における中間者攻撃

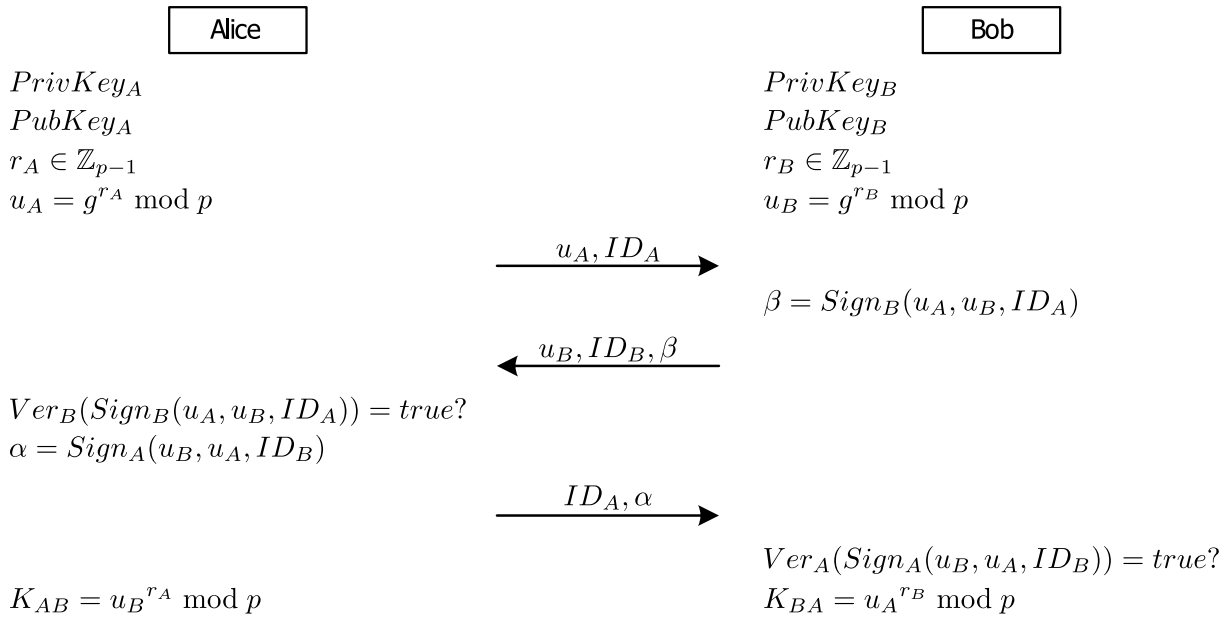

図 B.9　PKI ベース認証鍵交換の仕組み

図 B.9 にディジタル署名を用いた PKI ベース AKE の仕組みを示す．Alice と Bob は，システムで定義された署名方式の鍵生成処理を個別に行い，それぞれ固定の秘密鍵 *PrivKey* と公開鍵 *PubKey* を生成する．その後，公開鍵を PKI の手順に従って登録する．

Alice はランダムに $r_A \in \mathbb{Z}_{p-1}$ を選択し，$u_A = g^{r_A} \bmod p$ を計算した後，自身のユーザ ID ID_A と共に u_A を Bob に送信する．Bob も同様に $r_B \in \mathbb{Z}_{p-1}$ をランダムに選択し，$u_B = g^{r_B} \bmod p$ を計算する．さらに，受信した u_A と通信相手のユーザ ID ID_A に生成した u_B を併せて，秘密鍵 $PribKey_B$ により署名 $\beta = Sign_B(u_A, u_B, ID_A)$ を生成する．Alice には生成した u_B と自身のユーザ ID ID_B，署名 β を送信する．

Alice は受信したメッセージより通信相手のユーザ ID ID_B を取得し，何らかの方法により通信相手の公開鍵 $PubKey_B$ を認証された形で取得する．公開鍵を取得できたら，受信した署名 β の検証を行い，正しければ暗号鍵 $K_{AB} = u_B^{r_A} \bmod p$ を生成する．さらに，受信した u_B，ID_B と u_A から署名 $\alpha = Sign_A(u_B, u_A, ID_B)$ を生成し，Bob へ送信する．Bob も通信相手の公開鍵 $PubKey_A$ を取得し，署名 α を検証する．検証により通信相手を認証できたら，暗号鍵 $K_{BA} = u_A^{r_B} \bmod p$ を生成する．

以上のように，通信ペアが互いに相手を認証することにより，中間者攻撃が行われても第三者との暗号鍵の共有を防止することが可能になる．なお，Alice と Bob が生成した秘密鍵 r_A，r_B は上記共有鍵 $K = K_{AB} = K_{BA}$ の生成のみに一時的に使われる使い捨ての情報であり，暗号鍵 K が交換された後は直ちにメモリから削除される．

付 録C　メッセージフォーマット

ここでは，提案方式における各種プロトコルのメッセージフォーマットを示す．付録 C.1 では 2 章で示した DPRP のメッセージフォーマットをまとめる．一方，付録 C.2 では 6 章で示した Mobile PPC と NAT-f を融合した GSCIP における各種メッセージフォーマットをまとめる．

なお，現状では DPRP と GSCIP のメッセージフォーマットが独立しているが，Mobile PPC と NAT-f における制御メッセージで運ばれるデータを，DPRP メッセージのペイロードとして定義することにより，プロトコルの統合を計画している．

C.1　DPRP

C.1.1　DPRP ヘッダ

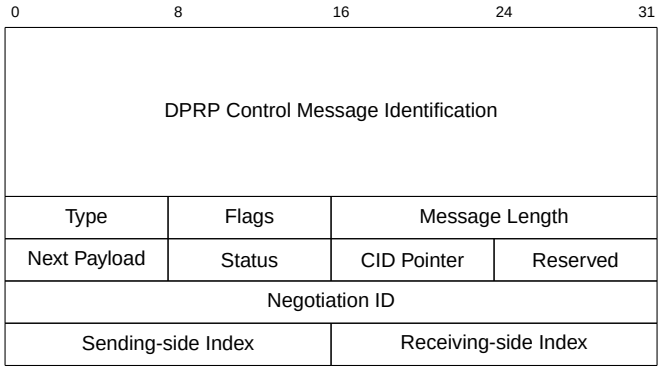

図 C.1　DPRP ヘッダフォーマット

DPRP Control Message Identification　(16 octets)

DPRP 制御メッセージを識別するための固定値．GPACK モジュールは ICMP Echo パケットのデータ部先頭 16 octets を検査し，以下の値であれば DPRP 制御メッセージと判断する．

`DPRP_ID  C734E6923B433BBFA54B2D91D44E059E`　固定値

Type　(1 octet)

DPRP 制御メッセージの種類を表す．

DPRP_TYPE_DDE	1	DDE メッセージ
DPRP_TYPE_RGI	2	RGI メッセージ
DPRP_TYPE_MPIT	3	MPIT メッセージ
DPRP_TYPE_CDN	4	CND メッセージ

Flags (1 octet)

オプショナルな処理を行った場合にフラグが設定される．通常の DPRP では，現時点で使用していない．

Message Length (2 octets)

DPRP 制御メッセージのメッセージ長を示す．

Next Payload (1 octet)

DPRP ヘッダ以降に続くペイロードの種類を示す．

DPRP_PLD_CID	1	通信識別子ペイロード
DPRP_PLD_NINFO	2	GE 情報ペイロード
DPRP_PLD_GAUTH	3	グループ認証ペイロード
DPRP_PLD_PINFO	4	動作処理情報ペイロード

Status (1 octet)

DPRP 制御メッセージの状態を示す．

DPRP_STS_OK	0	正常
DPRP_STS_NG	1	異常（エラー発生）

CID Pointer (1 octet)

DPRP 制御メッセージに複数の通信識別子が記載されている場合，何番目の通信識別子を対象に PIT を作成するか示す．

Reserved (1 octet)

未使用（予約）．

Negotiation ID (4 octets)

DPRP ネゴシエーションの識別子．1 回のネゴシエーションにおいて，DDE から CDN まで同じ値となる．

Sending-side Index (2 octets)

送信用 PIT を検索する際のハッシュ値が格納される．

Receiving-side Index (2 octets)

受信用 PIT を検索する際のハッシュ値が格納される．

C.1.2　DPRP ペイロードヘッダ

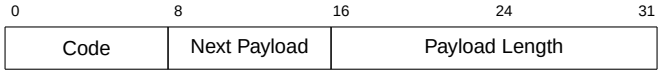

図 C.2　DPRP ペイロードヘッダフォーマット

Code　(1 octet)

DPRP ペイロードの種類を表す．DPRP ヘッダの Next Payload フィールドで定義された値（C.1.1 項）のいずれかが設定される．

Next Payload　(1 octet)

DPRP ヘッダ以降に続くペイロードの種類を示す．DPRP ヘッダの Next Payload フィールドで定義された値（C.1.1 項）のいずれかが設定される．

Payload Length　(2 octets)

DPRP ペイロード長を示す．

C.1.3　通信識別子ペイロード

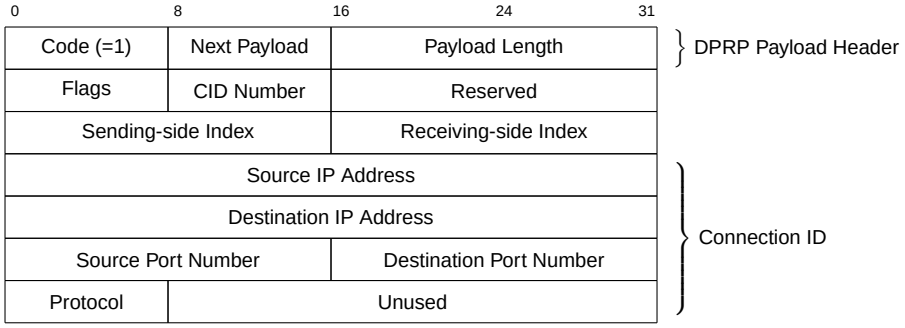

図 C.3　通信識別子ペイロードフォーマット

DPRP Payload Header　(4 octets)

DPRP ペイロードヘッダが設定される．ペイロードヘッダの `Code` には `1` が設定される．各フィールドの詳細は，図 C.2 を参照のこと．

Flags　(1 octet)

オプショナルな処理を行った場合にフラグが設定される．通常の DPRP では，使用していない．

CID Number　(1 octet)

通信識別子の番号を示す．DPRP ヘッダの CID Pointer フィールドが参照する値である．

Reserved (2 octets)

未使用（予約）.

Sending-side Index (2 octets)

送信用通信識別子のハッシュ値が格納される.

Receiving-side Index (2 octets)

受信用通信識別子のハッシュ値が格納される.

Connection ID (16 octets)

通信識別子が格納される．各フィールドの詳細は，図 C.4 を参照のこと.

通信識別子

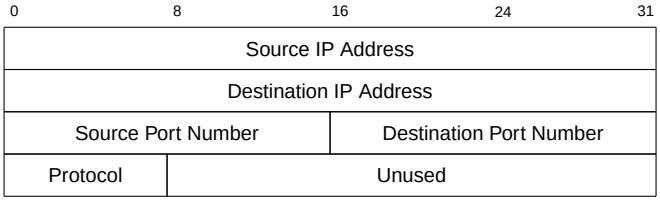

図 C.4　通信識別子フォーマット

Source IP Address (4 octets)

送信元 IP アドレスが設定される.

Destination IP Address (4 octets)

宛先 IP アドレスが設定される.

Source Port Number (2 octets)

送信元ポート番号が設定される.

Destination Port Number (2 octets)

宛先ポート番号が設定される.

Protocol (1 octet)

プロトコル番号が設定される.

Unused (3 octets)

未使用（ゼロパディング）.

C.1.4 GE 情報ペイロード

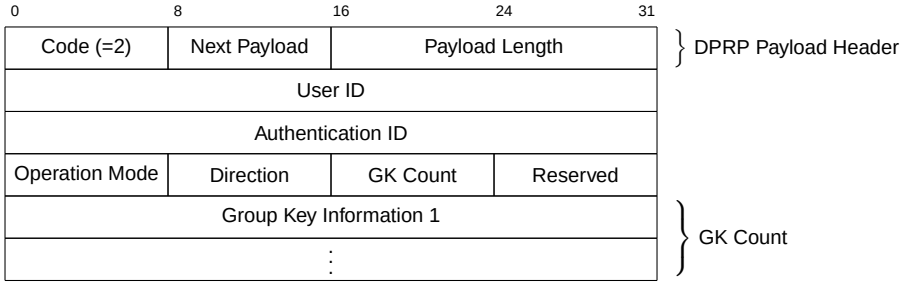

図 C.5　GE 情報ペイロードフォーマット

DPRP Payload Header　(4 octets)

DPRP ペイロードヘッダが設定される．ペイロードヘッダの `Code` には `2` が設定される．各フィールドの詳細は，図 C.2 を参照のこと．

User ID　(4 octets)

GE にログインしているユーザのユーザ ID が設定される．

Authentication ID　(4 octets)

動作処理情報の認証に用いる乱数値が設定される．

Operation Mode　(1 octet)

GE の動作モードを示す．

`GPACK_GEMODE_OP`	`1`	開放モード
`GPACK_GEMODE_CL`	`2`	閉域モード

Direction　(1 octet)

DPRP ネゴシエーションの方向情報を示す．

`DPRP_DIRECT_EDGE`	`1`	ネゴシエーションの終端
`DPRP_DIRECT_OUT`	`2`	DDE がネットワークから出る方向
`DPRP_DIRECT_IN`	`3`	DDE がネットワークへ入る方向

GK Count　(1 octet)

ペイロードに記載されているグループ鍵情報の数が設定される．

Reserved　(1 octet)

未使用（予約）．

Group Key Information　(4 × (GK Count) octets)

グループ鍵情報が GK Count の数だけ格納される．詳細な構造は，図 C.6 を参照のこと．

グループ鍵情報

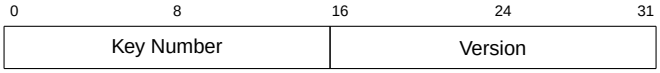

図 C.6　グループ鍵情報フォーマット

Key Number　(2 octets)

グループ鍵の番号が設定される．

Version　(2 octets)

グループ鍵のバージョンが設定される．

C.1.5　グループ認証ペイロード

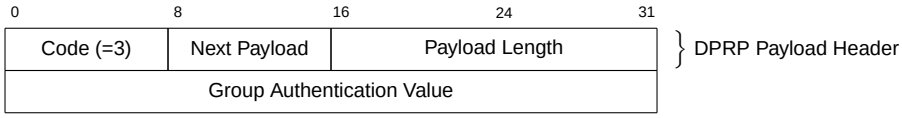

図 C.7　グループ認証ペイロードフォーマット

DPRP Payload Header　(4 octets)

DPRP ペイロードヘッダが設定される．ペイロードヘッダの Code には 3 が設定される．各フィールドの詳細は，図 C.2 を参照のこと．

Group Authentication Value　(4 octets)

決定したグループ鍵で暗号化されたネゴシエーション識別子が設定される．

C.1.6　動作処理情報ペイロード

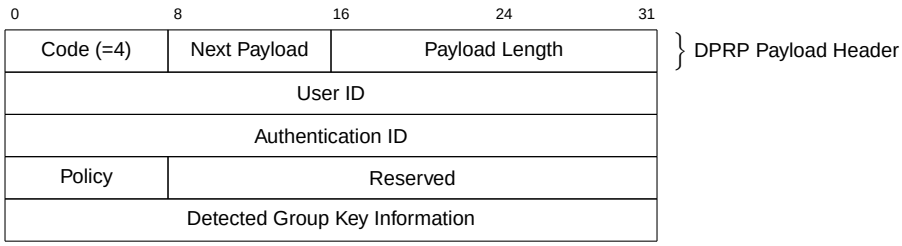

図 C.8　動作処理情報ペイロードフォーマット

DPRP Payload Header (4 octets)

DPRP ペイロードヘッダが設定される．ペイロードヘッダの `Code` には `4` が設定される．各フィールドの詳細は，図 C.2 を参照のこと．

User ID (4 octets)

RGI で受信したユーザ ID が設定される．

Authentication ID (4 octets)

RGI で受信した乱数値が設定される．

Policy (1 octet)

決定した処理内容を示す．

`PIT_PROC_ENCRYPT`	`1`	暗号化
`PIT_PROC_DECRYPT`	`2`	復号
`PIT_PROC_TRANSPARENT`	`3`	透過中継
`PIT_PROC_DISCARD`	`4`	破棄

Reserved (3 octets)

未使用（予約）．

Detected Group Key Information (4 octets)

決定したグループ鍵情報が設定される．詳細な構造は，図 C.6 を参照のこと．

C.2 GSCIP

C.2.1 GSCIP メッセージヘッダ

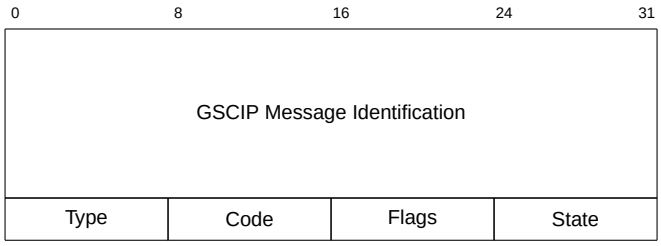

図 C.9 GSCIP メッセージヘッダフォーマット

GSCIP Message Identification (16 octets)

GSCIP メッセージを識別するための固定値．GPACK モジュールは ICMP Echo パケットのデータ部先頭 16 octets を検査し，以下の値であれば GSCIP メッセージと判断する．

`GSCIP_MSG_ID`	`C734E6923B433BBFA54B2D91D44E059E`	固定値

Type (1 octet)

GSCIP メッセージの種類を表す．Response メッセージと Reply メッセージの違いは，通信相手ノードが GSCIP 対応ノードか否かである．

GSCIP_TYPE_REQUEST	1	Request メッセージ
GSCIP_TYPE_RESPONSE	2	Response メッセージ（GSCIP 対応の場合）
GSCIP_TYPE_REPLY	3	Reply メッセージ（GSCIP 未対応の場合）

Code (1 octet)

GSCIP メッセージの種類を表す．実際の GSCIP メッセージは, Code フィールドと Type フィールドの組み合わせにより決定する．

GSCIP_CODE_SUPPORT_CHECK	1	Support Check メッセージ
GSCIP_CODE_MAPPING	2	Mapping メッセージ
GSCIP_CODE_BINDING	3	Binding メッセージ
GSCIP_CODE_COOKIE	4	Cookie メッセージ
GSCIP_CODE_DHKEY	5	DH Key メッセージ
GSCIP_CODE_CU	6	CU メッセージ

Flags (1 octet)

オプショナルな処理を行った場合にフラグが設定される．

GSCIP_FLAGS_PSEUDO	1	Binding メッセージにおける Pseudo メッセージ． NAT-f における疑似パケットであることを示す．

State (1 octet)

GSCIP メッセージの状態を示す．

GSCIP_STATE_OK	1	正常
GSCIP_STATE_NG_NO_ACT	2	異常（ACT エントリが見つからない）
GSCIP_STATE_NG_PERMIT	3	異常（アクセスが許可されていない）
GSCIP_STATE_NG_NAT_MAP	4	異常（NAT マッピング処理が失敗）

C.2.2 Support Check メッセージ

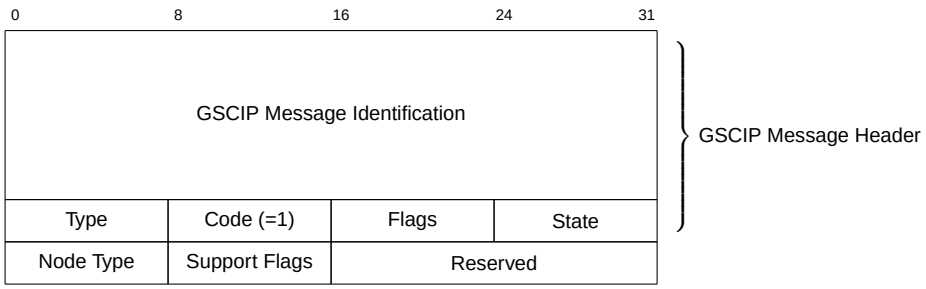

図 C.10　Support Check メッセージフォーマット

GSCIP Message Header (20 octets)

GSCIP メッセージヘッダが設定される．`Code` には `1` が設定される．各フィールドの詳細は，図 C.9 を参照のこと．

Node Type (1 octet)

GE のノードタイプを示す．

`GPACK_NODETYPE_GENERIC`	`0`	一般ノード
`GPACK_NODETYPE_GES`	`1`	GES
`GPACK_NODETYPE_GEN`	`2`	GEN

Support Flags (1 octet)

通信相手の GE が GSCIP に対応しているかを示す．

`GSCIP_SPTFLAG_NATF`	`0x01`	NAT-f 対応
`GSCIP_SPTFLAG_MPPC`	`0x02`	Mobile PPC 対応

Reserved (2 octets)

未使用（予約）．

C.2.3 Mapping メッセージ

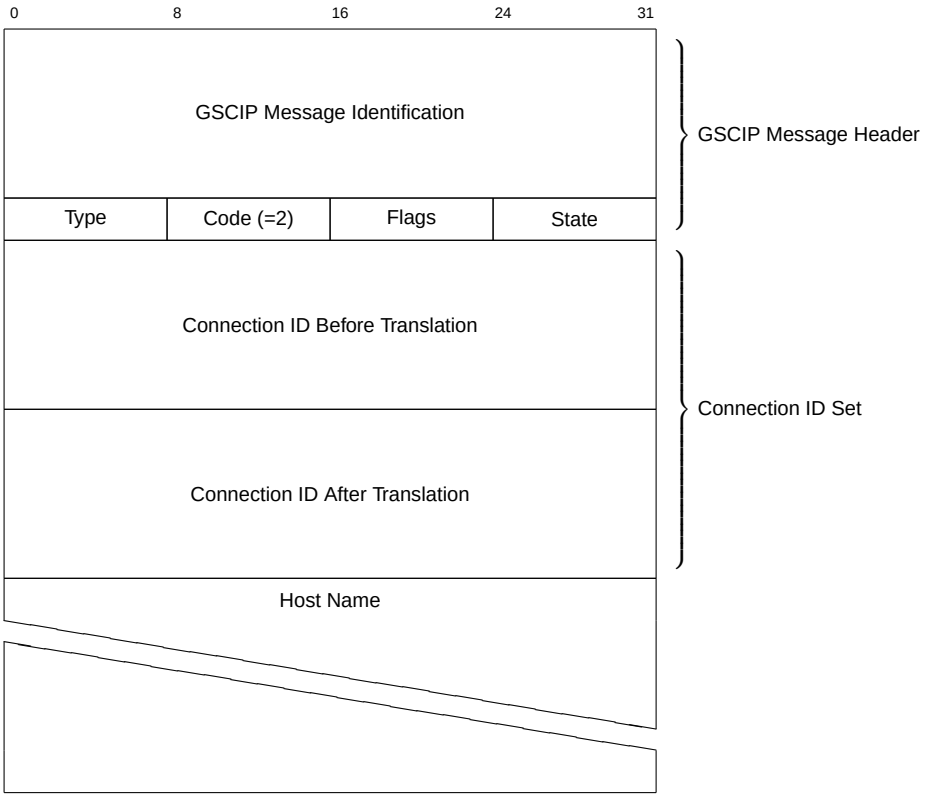

図 C.11 Mapping メッセージフォーマット

GSCIP Message Header (20 octets)

GSCIP メッセージヘッダが設定される．`Code` には `2` が設定される．各フィールドの詳細は，図 C.9 を参照のこと．

Connection ID Set (32 octets)

変換前と変換後の通信識別子の組が設定される．詳細な構造は，図 C.12 を参照のこと．

Host Name (64 octets)

ホスト名が格納される．

Connection ID Set

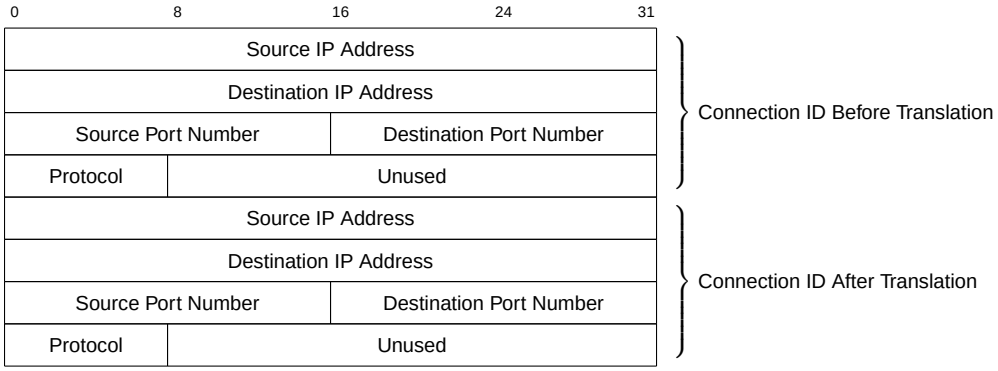

図 C.12　Connection ID Set フォーマット

Connection ID Before Translation (16 octets)

アドレス変換前の通信識別子が格納される．各フィールドの詳細は，図 C.4 を参照のこと．

Connection ID After Translation (16 octets)

アドレス変換後の通信識別子が格納される．各フィールドの詳細は，図 C.4 を参照のこと．

C.2.4 Cookie メッセージ

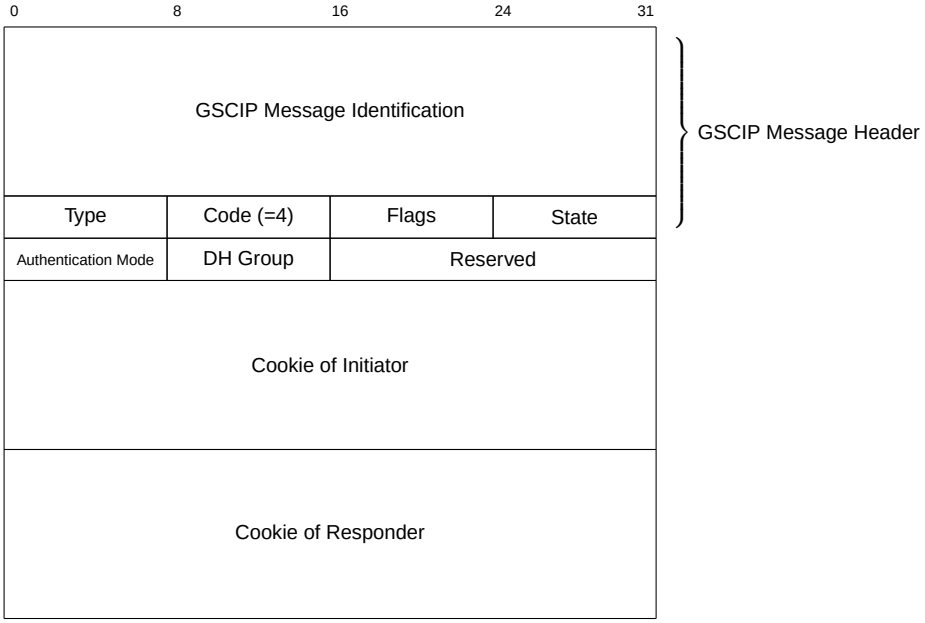

図 C.13　Cookie メッセージフォーマット

GSCIP Message Header　(20 octets)

GSCIP メッセージヘッダが設定される．`Code` には `4` が設定される．各フィールドの詳細は，図 C.9 を参照のこと．

Authentication Mode　(1 octet)

Mobile PPC の認証モードを示す．

`MPPC_AUTH_MODE_DKE`　`1`　DKE（Direct Key Exchange）モード

DH Group　(1 octet)

DH 鍵交換における DH グループを示す．グループ番号に応じて，DH 鍵交換で用いる素数 p のサイズが異なる．

`MPPC_DH_GROUP_1`　`1`　グループ 1（素数のサイズ：768 bits）
`MPPC_DH_GROUP_2`　`2`　グループ 2（素数のサイズ：1024 bits）
`MPPC_DH_GROUP_5`　`5`　グループ 5（素数のサイズ：1536 bits）
`MPPC_DH_GROUP_14`　`14`　グループ 14（素数のサイズ：2048 bits）

Reserved　(2 octets)

未使用（予約）．

Cookie of Initiator　(16 octets)

Mobile PPC 認証鍵共有ネゴシエーションの Initiator が生成した Cookie が格納される．

Cookie of Responder　(16 octets)

Mobile PPC 認証鍵共有ネゴシエーションの Responder が生成した Cookie が格納される．

C.2.5　DH Key メッセージ

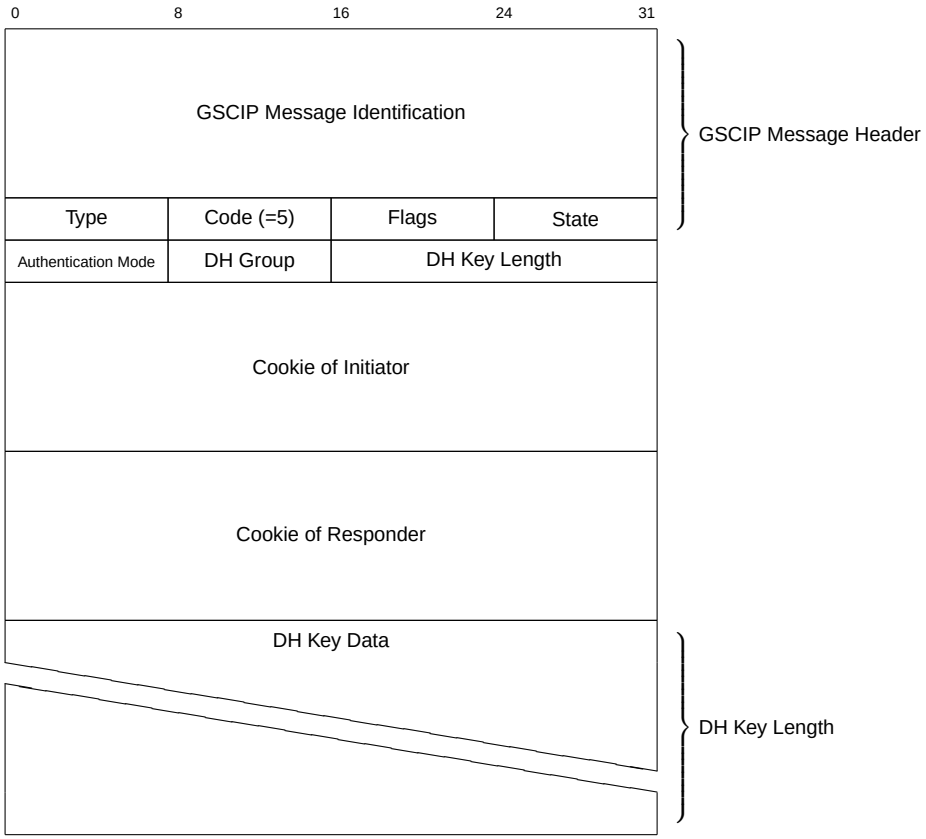

図 C.14　DH Key メッセージフォーマット

GSCIP Message Header　(20 octets)

GSCIP メッセージヘッダが設定される．`Code` には `5` が設定される．各フィールドの詳細は，図 C.9 を参照のこと．

Authentication Mode　(1 octet)

Mobile PPC の認証モードを示す．Cookie メッセージの Authentication Mode フィールドで定義された値（C.2.4 項）のいずれかが設定される．

DH Group　(1 octet)

DH 鍵交換における DH グループを示す．Cookie メッセージの DH Group フィールドで定義された値（C.2.4 項）のいずれかが設定される．

DH Key Length　(2 octets)

交換する DH 公開鍵の鍵長が設定される.

Cookie of Initiator (16 octets)

Mobile PPC 認証鍵共有ネゴシエーションの Initiator が生成した Cookie が格納される.

Cookie of Responder (16 octets)

Mobile PPC 認証鍵共有ネゴシエーションの Responder が生成した Cookie が格納される.

DH Key Data ((DH Key Length) octets)

DH 公開鍵のデータが格納される.

C.2.6　CU メッセージ

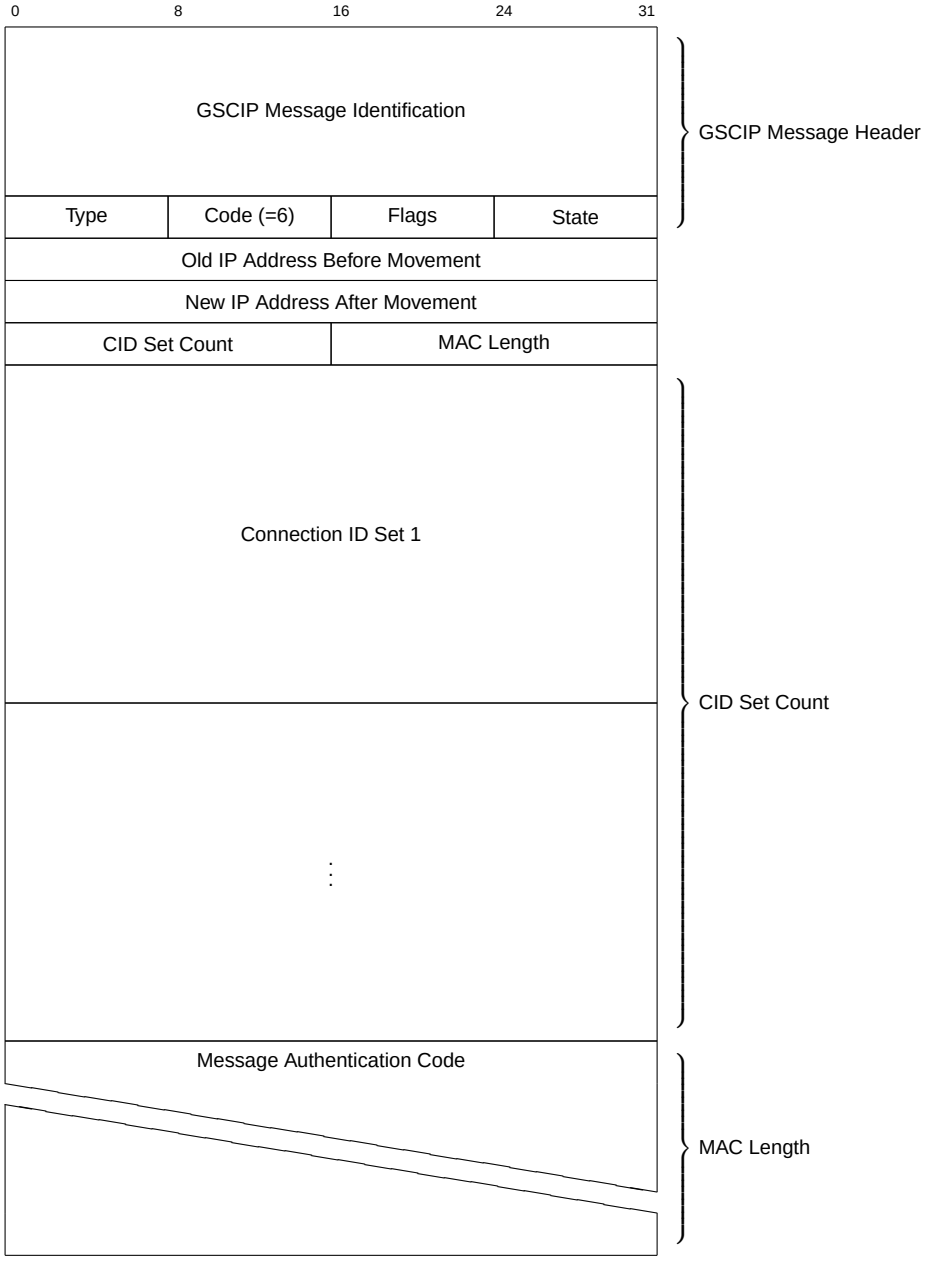

図 C.15　CU メッセージフォーマット

GSCIP Message Header　(20 octets)

GSCIP メッセージヘッダが設定される．`Code` には `6` が設定される．各フィールドの詳細は，図 C.9 を参照のこと．

Old IP Address Before Movement　(4 octets)

MN の移動前の IP アドレスが設定される．

New IP Address After Movement　(4 octets)

MN の移動後の IP アドレスが設定される．

CID Set Count　(2 octets)

CU に記載される通信識別子セットの数が設定される．

MAC Length　(2 octets)

CU に付加されるメッセージ認証コード（Message Authentication Code：MAC）のサイズが設定される．

Connection ID Set　(32 × (CID Set Count) octets)

MN が通知する変換前後の通信識別子の組が，CIT Set Count の数だけ記載される．詳細な構造は，図 C.12 を参照のこと．

Message Authentication Code　((MAC Length) octets)

メッセージ認証コードが格納される．

付 録D　GSCIPの関連研究

D.1　グループ管理装置 GMS

図 D.1 に GSCIP におけるグループ管理システムの構成を示す．グループ管理サーバ GMS（Group Management Server）は，以下の 3 つのコンポーネントから構成される．

データベース

GSCIP を利用するノード GE に関する情報（GE 情報）やグループ鍵を格納するデータベース．GMS デーモンプロセスと Web アプリケーションから参照される．プロトタイプシステムでは，MySQL [182] を使用している．

GMS デーモンプロセス

クライアントである各 GE に対して GE 情報やグループ鍵を配送したり，グループ鍵の更新などを行うサーバデーモン（Group Management Server Deamon：gmcd）．また，GE からの要求メッセージや，Web アプリケーションからの指示を処理する．プロトタイプシステムでは C 言語でプログラムされており，

Web アプリケーション

GSCIP 管理者が各種設定を行うためのインタフェースとして動作する Web アプリケーション．GE 追加登録や通信グループ構成の変更，グループ鍵の更新を受けつける．データベースの参照や GMS デーモンプロセスへの指示などは，PHP（Hypertext Preprocessor）で記述されたプログラムが行う．プロトタイプシステムでは，WWW サーバアプリケーションとして Apache [183] を利用している．

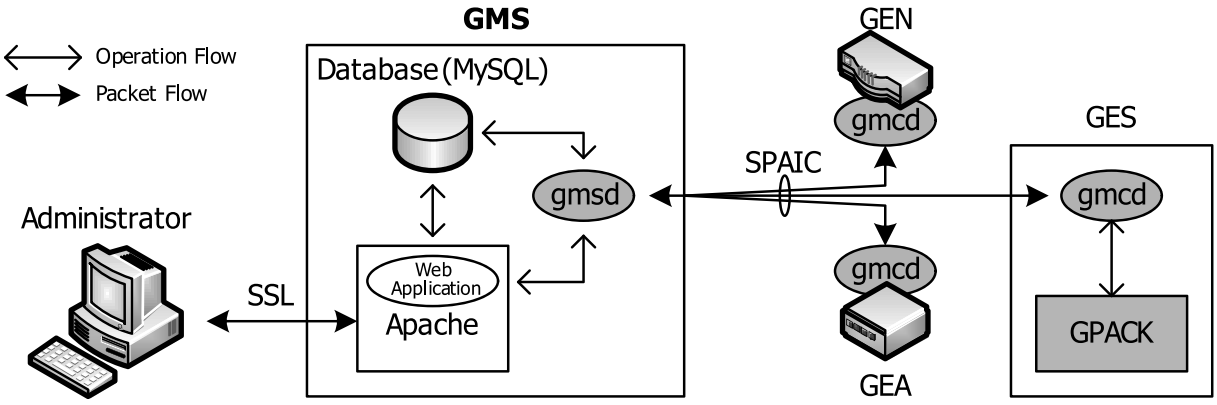

図 D.1　グループ管理システム構成

各 GE にはクライアント用デーモン（Group Management Client Deamon：gmcd）が動作しており，GMS とのメッセージ交換や，設定情報およびグループ鍵をカーネルモジュール GPACK（2.4 節を参照）へセットする機能がある．

なお，GE と GMS 間で交換されるメッセージの暗号化および認証には，GSCIP と親和性の高い SPAIC（Secure Protocol for Authentication with IC card） [36–38] を使用することを想定している．SPAIC の詳細については，D.2 節にて後述する．

D.1.1 グループ管理システムの動作

GMS で管理するデータベースを図 D.2 に示す．GE 情報テーブルは GE のユーザ ID，動作モード，GE がオンラインかオフラインであるかの状態，GE が GMS に対して定期的に送出するパケットの受信時間を記録するチェック時間で構成されている．GMS はこのチェック時間を用いて，GE の状態を記録している．グループ鍵が更新された場合は GE の状態を参照し，全オンライン GE に対してグループ鍵を即座に配送する．通信グループ情報テーブルには，通信グループ番号，グループ鍵のバージョン，グループ鍵長，グループ鍵で構成されている．所属グループ情報テーブルはどの GE がどの通信グループに所属しているかを示すテーブルで，ユーザ ID，通信グループ番号で構成されている．

図 D.3 に GMS から GE への配送情報を示す．GE は電源投入時に GMS と SPAIC による認証を行い，GE 毎に定義されている情報を取得する．この情報には，2 章で述べた動的処理解決プロトコル DPRP（Dynamic Process Resolution Protocol）[34] で使用するシステム共通暗号鍵 CK と，GE の動作モード，および所属する通信グループのグループ鍵とその情報が含まれる．これらの情報を受信した gmcd は，6.4.1 項で示した GSOCK（GPACK Socket）を利用してカーネルモジュール GPACK へセットする．GE があるノードに通信を開始すると DPRP が開始され，動作処理情報 PIT を生成後通信が開始される．

GMS は定期的あるいは管理者の要求により，システム共通暗号鍵とグループ鍵の更新を行う．これらの暗号鍵が更新された場合は，全オンライン GE に即座に配送し，GPACK へセットする．また，DPRP を開始して同一通信グループに所属しているにもかかわらずグループ鍵のバージョンが異なる場合は，GPACK から gmcd に対して鍵配送要求を指示し，gmcd が GMS から最新のグループ鍵を取得する．その後，再度 DPRP を開始して PIT 生成した後，新しいグループ鍵で通信が再開される．

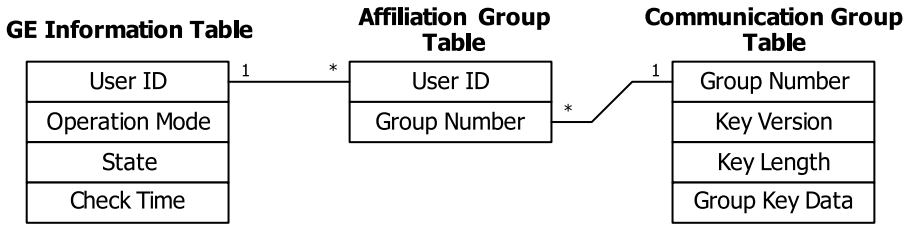

図 D.2　GMS データベース

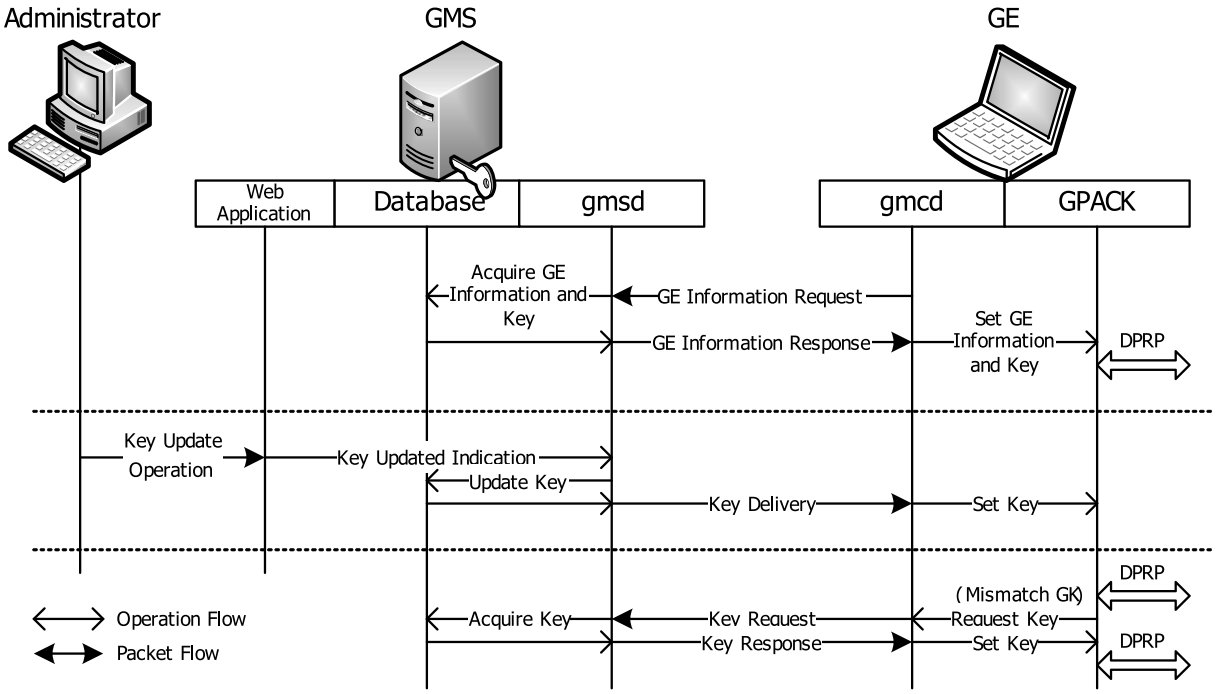

図 D.3　GMS から GE への配送シーケンス

D.2　認証方式 SPAIC

SPAIC（Secure Protocol for Authentication with IC card）は非接触型 IC カードを利用し，初期情報を一切持たないクライアントに対して重要情報を配送することを可能とするプロトコルである．SPAIC では IC カード，クライアント，サーバを独立したエンティティとし，これらを以下に示す 3 つの経路を環状で認証することによりクライアント/サーバ間の認証が行われる．

1. IC カードはパスワードや生体情報を用いてユーザ認証を行うことにより，クライアントを認証する．
2. サーバは IC カード秘密鍵から作成されたディジタル署名を検証することにより，IC カードを認証する．
3. クライアントはサーバ秘密鍵から作成されたディジタル署名を検証することにより，サーバを認証する．

GSCIP では GE と GMS 間でグループ鍵や GE 情報を配送する際に SPAIC を利用することを想定している．

図 D.4 に SPAIC シーケンスを示す．初期情報として，IC カードは所有ユーザの ID（*uID*），パスワード（*PW*），生体情報テンプレート *T* に加えて，IC カード秘密鍵/公開鍵（$Priv_{IC}$，Pub_{IC}）とサーバ公開鍵（Pub_S）が格納されている．サーバは各ユーザの ID と IC カード公開鍵およびサーバ秘密鍵（$Priv_S$）を保持している．

Step 1: ユーザは IC カードをリーダにかざすと，クライアントは IC カードに公開鍵を要求する．
IC カードは乱数 N_i を生成し，*uID*，Pub_{IC}，Pub_S と共にクライアントへ送信する．

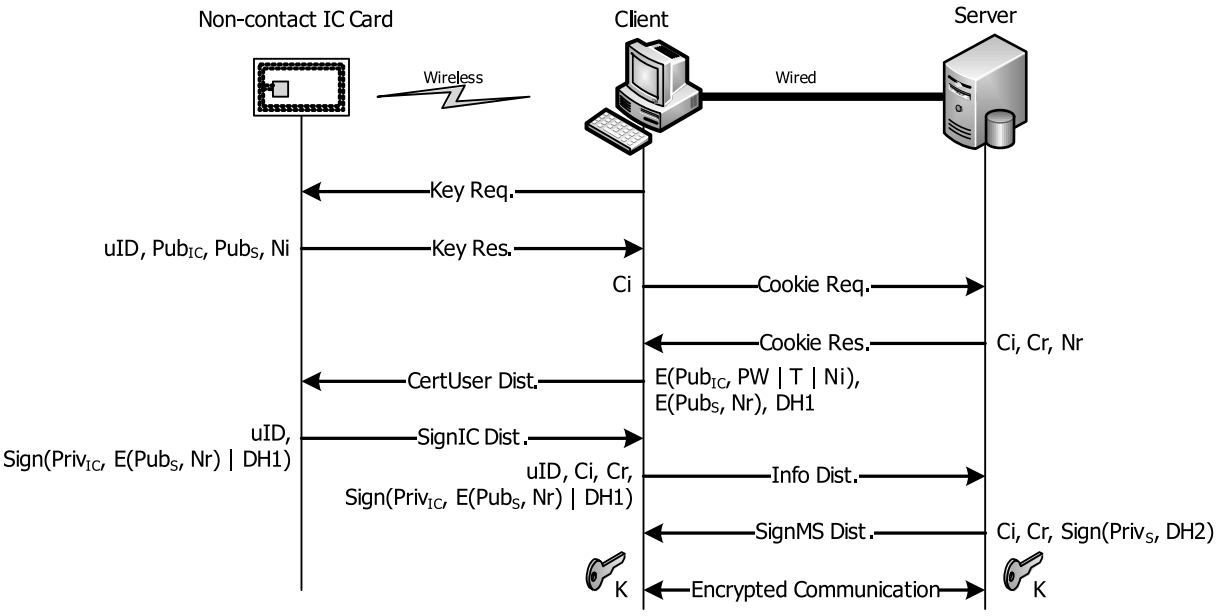

図 D.4　SPAIC シーケンス

Step 2: クライアントは後ほど行う DH 鍵交換に対する DoS 攻撃を防止するためのクッキー C_i を生成して，サーバへ送信する．サーバは乱数 N_r とクッキー C_r を生成し，受信した C_i と共にクライアントへ送信する．

Step 3: クライアントではログイン画面が表示され，ユーザが認証情報（パスワード PW と生体情報 T）を入力する．認証情報は IC カードから受信した N_i と結合され，IC カード公開鍵で暗号化される．

$$E(Pub_{IC}, PW \mid T \mid N_i)$$

同時にサーバから受信した N_r をサーバ公開鍵で暗号化する．

$$E(Pub_S, N_r)$$

また，DH 交換値 $DH1$ を生成する．以上の情報を IC カードへ送信する．
IC カードでは $Priv_{IC}$ を用いて PW，T，N_i を取り出してユーザ認証を行う．ユーザ認証後，受信した $E(Pub_S, N_r)$ に $DH1$ を付加して，IC カード秘密鍵でディジタル署名 $Sign$ を作成する．

$$Sign(Priv_{IC}, E(Pub_S, N_r) \mid DH1)$$

署名された情報はユーザ ID と共にクライアントへ送信される．

Step 4: クライアントは受信した情報とクッキー C_i，C_r をサーバに送信する．サーバはクライアントから送られてきたクッキーの正当性を確認する．また，uID に該当する IC カード公開鍵を読み出し，ディジタル署名の検証を行い，IC カードを認証する．その後，DH 交換値 $DH2$ を生成し，サーバ秘密鍵でディジタル署名を作成する．

$$Sign(Priv_S, DH2)$$

サーバは証明した DH 交換値とクッキーをクライアントへ送信すると同時に，$DH1$ と $DH2$ を利用して共通暗号鍵 K を生成する．

Step 5: クライアントは受信したクッキーの正当性を確認し，あらかじめ受信したサーバ公開鍵 Pub_S を用いてディジタル署名の検証を行い，サーバを認証する．その後，生成した $DH1$ と取得した $DH2$ から共通暗号鍵 K を生成する．

以降，クライアント/サーバ間の通信は共通鍵 K により暗号化される．

付 録E　Mobile PPCの関連研究

E.1　認証鍵共有処理

MN は CN との通信に先立ち，2 往復の認証鍵共有ネゴシエーションを行う．図 E.1 に認証鍵共有シーケンスを示す．認証鍵共有シーケンスはカーネルモジュールにおいて実行する Cookie 交換と，ユーザランドで動作するデーモン `mppcd` が実行する DH 鍵交換から構成される．

(1) MN のアプリケーションはストリームソケットまたはデータグラムソケットによりデータを CN へ送信する．ここで，カーネルに実装されている Mobile PPC モジュールは最初の TCP/UDP パケットを一時待避し，Cookie CKY_{MN} を生成して CN へ Cookie Request を送信する．Cookie は MN と CN の IP アドレス，乱数 R，及び生成時刻 T のデータを結合し，ハッシュ関数により出力される値として生成される．

$$CKY_{MN} = h(IP_{MN} \parallel IP_{CN} \parallel R_{MN} \parallel T_{MN}) \tag{E.1}$$

(2) Cookie Request を受信した CN は Cookie CKY_{CN} を生成し，受信した CKY_{MN} と共に Cookie Response を MN へ応答する．

$$CKY_{CN} = h(IP_{CN} \parallel IP_{MN} \parallel R_{CN} \parallel T_{CN}) \tag{E.2}$$

(3) MN は受信した C_{MN} を検証後，先ほど一時停止させた TCP/UDP 通信を開始してから，DH 鍵交換をバックエンドで行うため GSOCK を通じて `mppcd` に認証鍵を要求する．

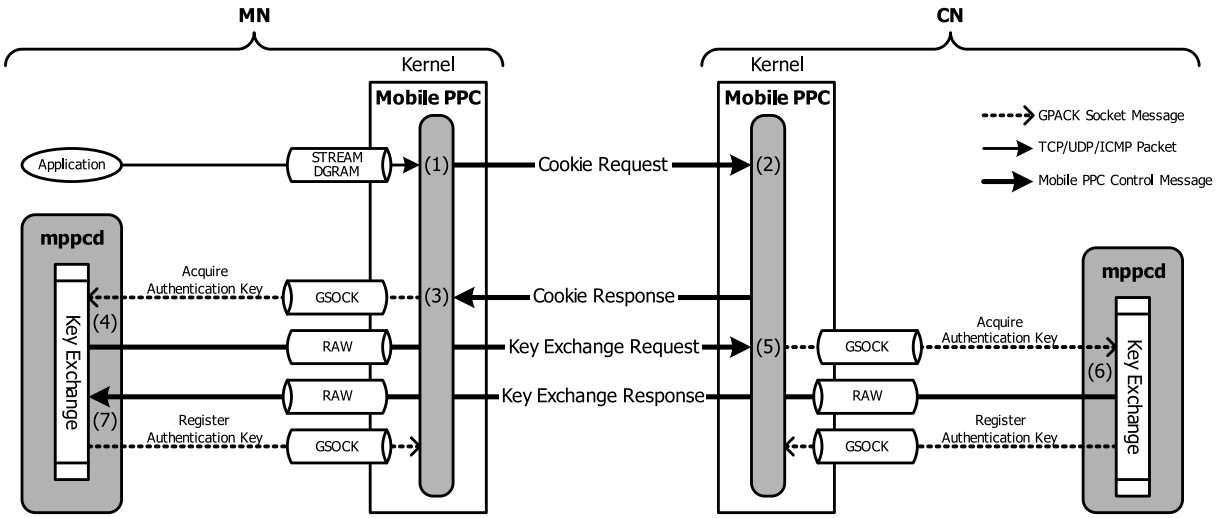

図 E.1　認証鍵共有シーケンスの詳細

(4) MN の mppcd は DH 秘密鍵 $PrivKey_{MN}$ をランダムに生成し，それに対応する DH 公開鍵 $PubKey_{MN}$ をシステム共通の素数 p 及び原始根 $g \in \mathbb{Z}_p^*$ により計算する．

$$PrivKey_{MN} \in \mathbb{Z}_{p-1} \tag{E.3}$$

$$PubKey_{MN} = g^{PrivKey_{MN}} \bmod p \tag{E.4}$$

その後，DH Key Request により CN へ DH 公開鍵と Cookie を送信する．

(5) DH Key Request を受信後，CN は受信した CKY_{CN} を検証する．CKY_{CN} が正しければ，mppcd に対して受信した MN の DH 公開鍵 $PubKey_{MN}$ を渡して認証鍵を要求する．

(6) CN の mppcd は MN と同様に DH 秘密鍵と DH 公開鍵を生成する．

$$PrivKey_{CN} \in \mathbb{Z}_{p-1} \tag{E.5}$$

$$PubKey_{CN} = g^{PrivKey_{CN}} \bmod p \tag{E.6}$$

上記 DH 公開鍵と Cookie を DH Key Response に記載して MN へ応答する．この後，CN は下記の手順に従って認証鍵を生成し，Mobile PPC カーネルモジュールに登録する．

(7) DH 鍵交換終了後，MN と CN は自身の DH 秘密鍵と受信した相手の DH 公開鍵により共通鍵 $SK_{MN,CN}$ を生成する．

$$SK_{MN,CN} = \begin{cases} {PubKey_{CN}}^{PrivKey_{MN}} \bmod p & \text{（MN 側）} \\ {PubKey_{MN}}^{PrivKey_{CN}} \bmod p & \text{（CN 側）} \end{cases} \tag{E.7}$$

さらに生成した共通鍵と交換した Cookie のハッシュ値を求め，最終的な認証鍵 AK を生成する．

$$AK = h(SK_{MN,CN} \parallel CKY_{MN} \parallel CKY_{CN}) \tag{E.8}$$

上記認証鍵は Mobile PPC カーネルモジュールに登録され，移動後の CU Request/Response の署名生成，及び検証のために用いられる．

E.2 Mobile PPC における NAT Traversal 処理

E.2.1 概要

IPv4 ネットワークにおいて移動通信を実現する場合，アドレス空間の違いを考慮する必要がある．MN と CN の通信経路上に NAT が介在すると，従来の Mobile PPC では以下のような課題が生じる．

課題 1: MN がグローバルネットワークからプライベートネットワークへ移動した場合，CN が MN から受け取る移動後 IP アドレスはプライベート IP アドレスのため，グローバルネットワーク上を正しくルーチングできない．

課題 2: 仮に MN の移動後 IP アドレスを NAT の外部 IP アドレスに変換できたとしても NAT までは正しくルーチングされるが，NAT マッピング情報が存在しないため，MN に到達できない．

課題 3: MN がプライベートネットワークからグローバルネットワークへ移動した場合，CN が受け取る移動前 IP アドレスはプライベート IP アドレスであるが，CN は MN の移動前 IP アドレスを NAT の外部 IP アドレスとして認識している．従って，CN は正しく CIT を更新することができない．

Hole punching [22, 23] は既存の NAT を変更することなく NAT 越え問題を解決できるという利点がある．上記手法を Mobile PPC に適用することにより，CN に対して MAPPED-ADDRESS [1]を通知し，課題 1 と課題 3 を解決できる．また，Binding 処理により NAT マッピング情報が生成されるため，課題 2 も解決できる．

ただし，Hole punching は Symmetric NAT [2] と呼ばれるタイプの NAT には適用できないうえ，TCP に対応できないという課題がある．またグローバルネットワーク上にサーバを設置する必要があり，さらには STUN（Simple Traversal of UDP Through NATs） [137] クライアントのアプリケーションにこの機能を実装しなければならない．これらの課題は Mobile PPC の利点，すなわち上位プロトコルに依存しない，特有のサーバを必要としないという特徴を損ねてしまう．

そこで，Hole punching の手法をそのまま適用するのではなく，上記利点を保持できるような工夫を追加する．Mobile PPC に関わるネゴシエーションにおいて MN と CN の通信経路上に NAT の存在を確認した場合，両ノード間で直接 Hole punching を実行する．このために，Mobile PPC に新たに Binding Request/Response メッセージを定義する．

以降，MN がグローバルネットワークからプライベートネットワークへ移動する場合と，その逆方向の移動について述べる．

E.2.2 アドレスに関する用語の定義

本方式で用いるアドレスを以下のように定義する．

- PREV-ADDRESS: CN から見た MN の移動前 IP アドレス・ポート番号の組
- MOVED-ADDRESS: CN から見た MN の移動後 IP アドレス・ポート番号の組
- MAPPED-ADDRESS: Hole punching により割り当てられた NAT 外側の IP アドレス・ポート番号の組
- REAL-PREV-ADDRESS: MN の移動前実 IP アドレス
- REAL-MOVED-ADDRESS: MN の移動後実 IP アドレス

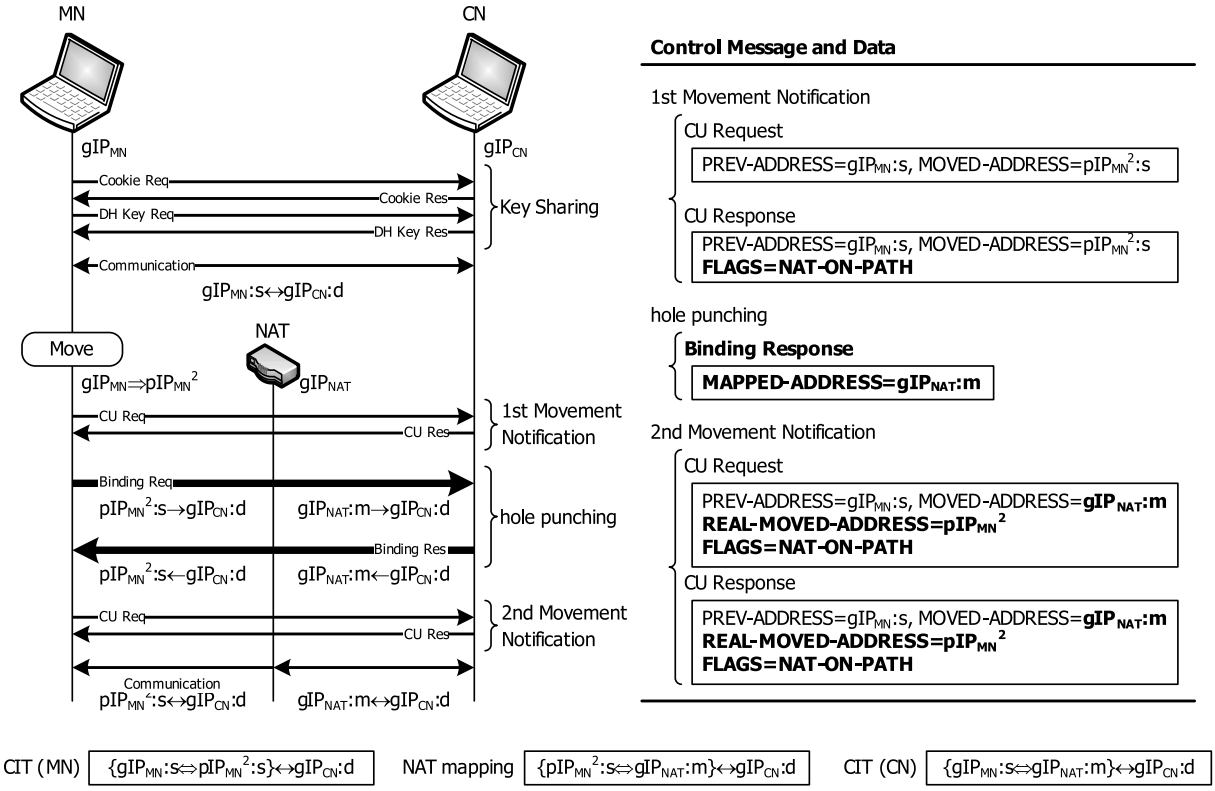

図 E.2　グローバルネットワークからプライベートネットワークへ移動する場合の通信シーケンス

E.2.3　プライベートネットワークへの移動

図 E.2 に MN がグローバルネットワークからプライベートネットワークへ移動する場合の通信シーケンスを示す．この場合，通信開始時には MN と CN の通信経路上に NAT が存在しないため，通常の Mobile PPC と同様に，MN と CN は通信に先立って DH 鍵交換を用いて認証鍵を共有する．その後，MN と CN は移動前の情報として式 (E.9) のような CIT を作成してから通信を開始する．

$$\text{MN/CN:} \qquad gIP_{MN} : s \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.9}$$

MN が通信中にプライベートネットワークに移動して，新しいプライベート IP アドレス “$pIP_{MN}{}^2$” を取得すると，CN に対して移動通知を行う．CN に送信する CU Request には，MOVED-ADDRESS として “$pIP_{MN}{}^2$” が記載され，通信開始時に共有した認証鍵を用いて署名を付加する．

CU Request を受信した CN は認証処理を終えた後，通知された MOVED-ADDRESS と CU Request の送信元 IP アドレスを比較する．CU Request の送信元 IP アドレスは NAT のグローバル IP アドレス “gIP_{NAT}” であるため，CN はアドレスの不一致から通信経路上に NAT が存在すると判断する．この場合，CN は CIT を更新せず，CU Response に新たに定義したフラグ NAT-ON-PATH を設定して応答する．CN は MN から次に Binding Request が送られてくることを期待して，通知された CID の宛先ポート “d” を一定時間監視する．

[1]NAT の外側に割り当てられた IP アドレスとポート番号の組.
[2]宛先が変化すると NAT の外側に割り当てられるポート番号も必ず変化する方式.

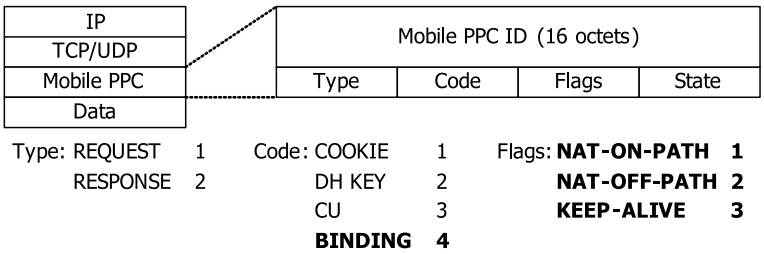

図 E.3　Binding Request/Response パケットフォーマット

MN は NAT-ON-PATH フラグが設定された CU Response を受信したら，CN に対して Binding Request を送信する．図 E.3 に Binding Request/Response のパケットフォーマットを示す．一般的な Hole punching では宛先ポート番号を特定の値に設定するが，提案方式では Symmetric NAT にも対応するため宛先ポート番号を以下のように決定する．すなわち，Binding Request の CID は通信パケットと同じ式 (E.10) とする[3].

$$pIP_{MN}{}^{2} : s \rightarrow gIP_{CN} : d \quad [proto] \tag{E.10}$$

NAT は Binding Request を転送する際，NAT の原理によりマッピング情報

$$\text{NAT:} \qquad \{pIP_{MN}{}^{2} : s \overset{NAT}{\Longleftrightarrow} gIP_{NAT} : m\} \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.11}$$

を生成する．CN は監視しているポート “*d*” 宛のパケットを受信するので，IP 層の Mobile PPC モジュールにおいて TCP/UDP ペイロード部に定義された Mobile PPC ヘッダを参照する．さらに新たに定義された Code と Flags の値（図中の太字部分）をチェックすることにより，Binding Request かどうか判別する．Binding Request であったら，送信元 IP アドレス “gIP_{NAT}” とポート番号 “*m*” を MAPPED-ADDRESS として，Binding Response のデータ部に記載して MN に応答する．

MN は上記 Binding Response を受信すると, 取得した MAPPED-ADDRESS “$gIP_{NAT} : m$” を MOVED-ADDRESS に変更し，MN のプライベート IP アドレス “$pIP_{MN}{}^{2}$” を REAL-MOVED-ADDRESS として CU Request の情報に追加する．その後，フラグに NAT-ON-PATH を設定して CN へ再度 CU Request を送信する．

上記 CU Request を受信した CN は認証処理を終えた後，従来の Mobile PPC と同様に自身の CIT を式 (E.12) のよう更新し，NAT-ON-PATH フラグをセットしたまま MN に CU Response を応答する．

$$\text{CN:} \qquad \{gIP_{MN} : s \overset{CIT}{\Longleftrightarrow} gIP_{NAT} : m\} \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.12}$$

MN は上記 CU Response を受信したら, MOVED-ADDRESS は参照せず REAL-MOVED-ADDRESS “$pIP_{MN}{}^{2}$” を用いて CIT を式 (E.13) のように更新する．

$$\text{MN:} \qquad \{gIP_{MN} : s \overset{CIT}{\Longleftrightarrow} pIP_{MN}{}^{2} : s\} \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.13}$$

以後，IP 層において更新した CIT に基づいてアドレス変換が行われる．MN は上位層から渡されたパケットの送信元 IP アドレスを，移動前の “gIP_{MN}” から移動後の “$pIP_{MN}{}^{2}$” へ変換して CN へ送

[3]Binding Request の L4 プロトコルヘッダの内容は，移動前の通信のプロトコルタイプを用いる．

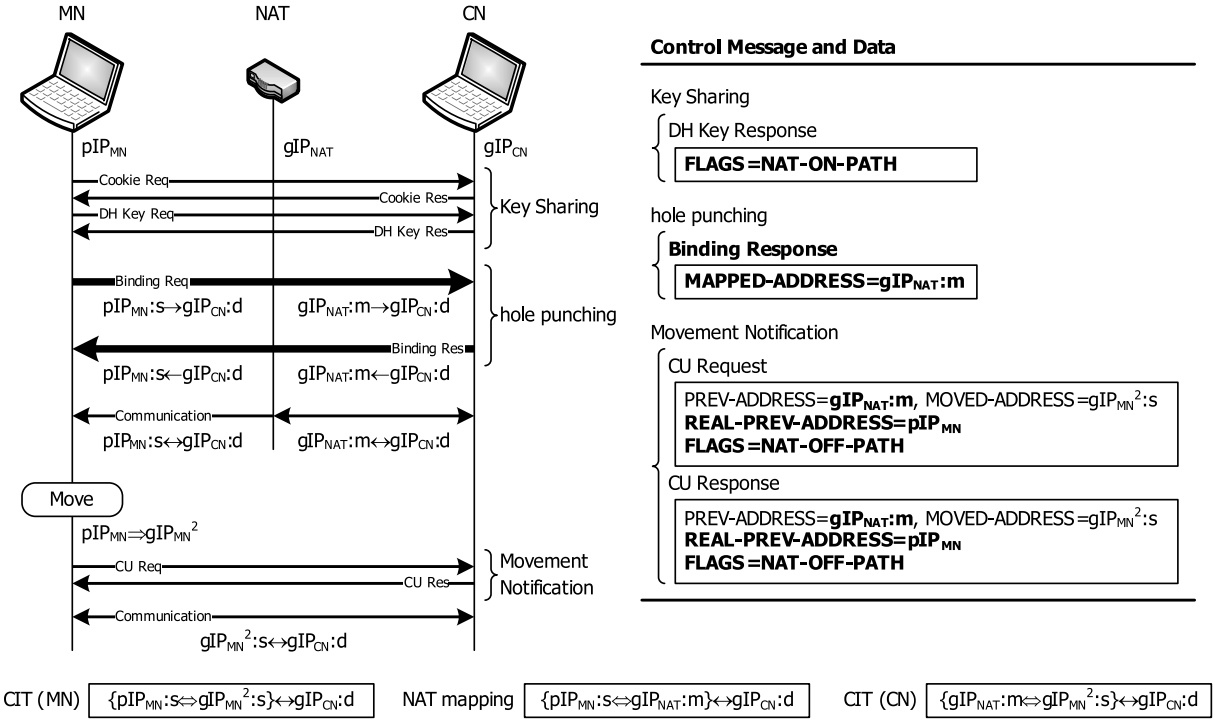

図 E.4　プライベートネットワークからグローバルネットワークへ移動する場合の通信シーケンス

信する．NAT はマッピング情報に基づいて，送信元 IP アドレスとポート番号を “$pIP_{MN}{}^2 : s$” から “$gIP_{NAT} : m$” へ変換して CN へ転送する．CN は受信したパケットの送信元を移動後の “$gIP_{NAT} : m$” から移動前の “$gIP_{MN} : s$” へ変換して上位層へ渡す．以上の動作により，グローバルネットワークからプライベートネットワークへ移動しても通信を継続することができる．

E.2.4　グローバルネットワークへの移動

次に，MN がグローバルネットワークからプライベートネットワークへ移動する場合の通信シーケンスを図 E.4 に示す．基本的な仕組みは E.2.3 項と同様である．通信開始時に，MN と CN は DH 鍵交換により認証鍵を共有する．上記シーケンスには MN のプライベート IP アドレスが含まれているため，CN はパケットの送信元 IP アドレスと比較することにより，NAT を経由していることがわかる．そこで CN は鍵交換シーケンスの最後の応答に NAT-ON-PATH フラグを設定する．MN は認証鍵を生成した後，CN と Binding 処理を実行して MAPPED-ADDRESS “$gIP_{NAT} : m$” を取得する．

MN が通信中にグローバルネットワークへ移動して新しい IP アドレス “$gIP_{MN}{}^2$” を取得すると，CN に対して移動通知を行う．MN は通信開始時に NAT-ON-PATH フラグを取得しているため，CU Request には MAPPED-ADDRESS “$gIP_{NAT} : m$” を PREV-ADDRESS に，MN の移動前プライベート IP アドレス “pIP_{MN}” を REAL-PREV-ADDRESS として記載する．その後，NAT-OFF-PATH フラグを設定して CN へ送信する．

CU Request を受信した CN は認証処理を終えた後，通常の Mobile PPC と同様に自身の CIT を

式 (E.14) のよう更新し，MN に CU Response を応答する．

$$\text{CN:} \qquad \{gIP_{NAT} : m \overset{CIT}{\Longleftrightarrow} gIP_{MN}{}^2 : s\} \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.14}$$

MN は NAT-OFF-PATH フラグが設定された CU Response を受信するので，送信元 IP アドレスが MOVED-ADDRESS “$gIP_{MN}{}^2$” となるように CIT を式 (E.15) のように更新する．

$$\text{MN:} \qquad \{pIP_{MN} : s \overset{CIT}{\Longleftrightarrow} gIP_{MN}{}^2 : s\} \leftrightarrow gIP_{CN} : d \quad [proto] \tag{E.15}$$

以後，MN は上位層から渡されたパケットの送信元 IP アドレスを，移動前の “pIP_{MN}” から移動後の “$gIP_{MN}{}^2$” へ変換してから CN へ送信する．CN では受信したパケットの送信元を移動後の “$gIP_{MN}{}^2 : s$” から，移動前の “$gIP_{NAT} : m$” へ変換して上位層へ渡す．以上の動作により，プライベートネットワークからグローバルネットワークへ移動しても通信を継続することができる．

E.2.5 考察

提案方式により，CN がグローバルネットワーク上に存在する場合，MN がグローバルネットワークとプライベートネットワークを相互に移動しても，通信を継続できることを示した．これ以外の移動パターンとして，MN が異なるプライベートネットワーク間を移動するケースが考えられる．この場合は，E.2.4 項の通信開始時のシーケンスと，E.2.3 項の移動後のシーケンスを組み合わせることにより，実現可能である．

提案方式では Hole punching に相当する Binding 処理を実際の通信相手と直接実行するため, STUN サーバのような装置は不要である．さらに実際の通信パケットと同じ CID を用いて Binding メッセージを生成しているため，TCP/UDP の両プロトコルと Symmetric NAT に対応できる．Binding 処理をカーネルレベルで実行しているため，アプリケーションを一切変更する必要はない．また，Hole punching の原理を適用しているため，プライベートネットワークが階層的に構成されていても動作可能である．

既存の NAT を利用して NAT 越え通信を実現するには，NAT のマッピング情報を維持するために Keepalive が必要となる．特に UDP 通信の場合，Binding 処理により生成されたマッピング情報は一定期間の無通信状態が発生すると消去されてしまい，通信の継続に影響を及ぼす．そのため，MN は CN に対して Keepalive パケットを定期的に送信する必要がある．Keepalive の送信間隔をマッピング情報の有効時間以下にすればよいが，各セッション単位で Keepalive を実行する必要があるため，CN に対する負荷が増加する可能性がある．マッピング情報保持時間は実装の種類や設定により異なるため，最適な値を検討する必要がある．

次に，本方式で導入した Binding 処理の安全性に関して考察する．攻撃者は Binding Response に記載されている MAPPED-ADDRESS を改ざんすることにより，セッションハイジャックを試みることが考えられる．Mobile PPC では MN と CN は通信開始時に認証鍵を共有しているため，Binding Request/Response に署名を付加することによりメッセージ完全性を保証できる．従って，Binding 処理の改ざんを検出することが可能である．

攻撃者が偽の Binding メッセージを送信することにより，セッションの妨害を試みることが考えられる．MN が CN へ Binding Request を送信後，ただちに攻撃者が偽の Binding Response を MN へ送信する．この攻撃も MN と CN 間で共有している認証鍵を用いることにより，メッセージ完全性の検証過程で検出することができる．

NAT によるアドレス変換を考慮すると，メッセージの完全性保証範囲は TCP/UDP ペイロード以降となる．そこで，攻撃者が正規の Binding Request を盗聴し，送信元を詐称した Binding Request を CN へ送信することが考えられる．この場合，CN は Binding Request を正規のメッセージと判断してしまい，詐称されたアドレスを MAPPED-ADDRESS として Binding Response を生成し，詐称されたアドレス宛（一般には攻撃者）へ応答する．

攻撃者は受信した Binding Response の送信元を CN に詐称して，NAT を経由して MN へ送信する．MN も Binding Response を正規のメッセージと判断してしまうため，以後の移動通知処理が完了すると攻撃者にセッションをハイジャックされてしまう．このような攻撃には，Binding Request のメッセージ部にシーケンス番号を設定して攻撃者による再送を防ぐ方法や，Ingress Filtering [112] を設定することで対策を講ずることが可能である．

E.3 パケットロスレスハンドオーバ

E.3.1 概要

Mobile PPC におけるパケットロスレスハンドオーバを実現するために，本研究ではデュアルインタフェース方式を採用している．デュアルインタフェース方式とはノードに無線インタフェースを複数保持させ，一方でパケットの送受信，もう一方で L2，L3 ハンドオーバを実行する．この方式は，移動ノードだけに処置をすればよくネットワークには変更が不要である．パケットロスも原理的になくすことが可能である．既存のデュアルインタフェース方式は電力消費の増加が課題となっていたが，提案方式では通信中でないカードを Sleep 状態にすることにより従来の課題を解決する．

図 E.5 に提案方式のエリア間ハンドオーバを示す．MN1 は 2 枚の無線 LAN カードを保持し，Old AP（Access Point）を介して Card 1 で通信を行っている．この状態では Card 2 はスリープ状態としている．スリープ状態とは省電力状態で，パケットやフレームの送信，受信を一切遮断した状態である．

MN1 は Card 1 で通信中に，接続中の Old AP の電波強度 RSSI（Received Signal Strength Indicaor）を定期的に測定する．RSSI は，Old AP から送信されるビーコンや，データパケットを受信したときに測定される．RSSI が低下して通信状態が不安定になる前にハンドオーバできるように，通信に適する閾値 α を設けておく．Old AP の RSSI が一定時間，閾値 α より低くなると，MN1 は Card1 による通信を維持しながら Card 2 のスリープ状態を解除する．次に Card 2 を用いてチャネルスキャンにより接続可能な AP を探索し，RSSI が最も高い AP を次に接続する New AP と定める．さらに，MN1 は New AP の ESS-ID の値を調べることによって，ネットワークが Old AP と同一か否かを判断する．

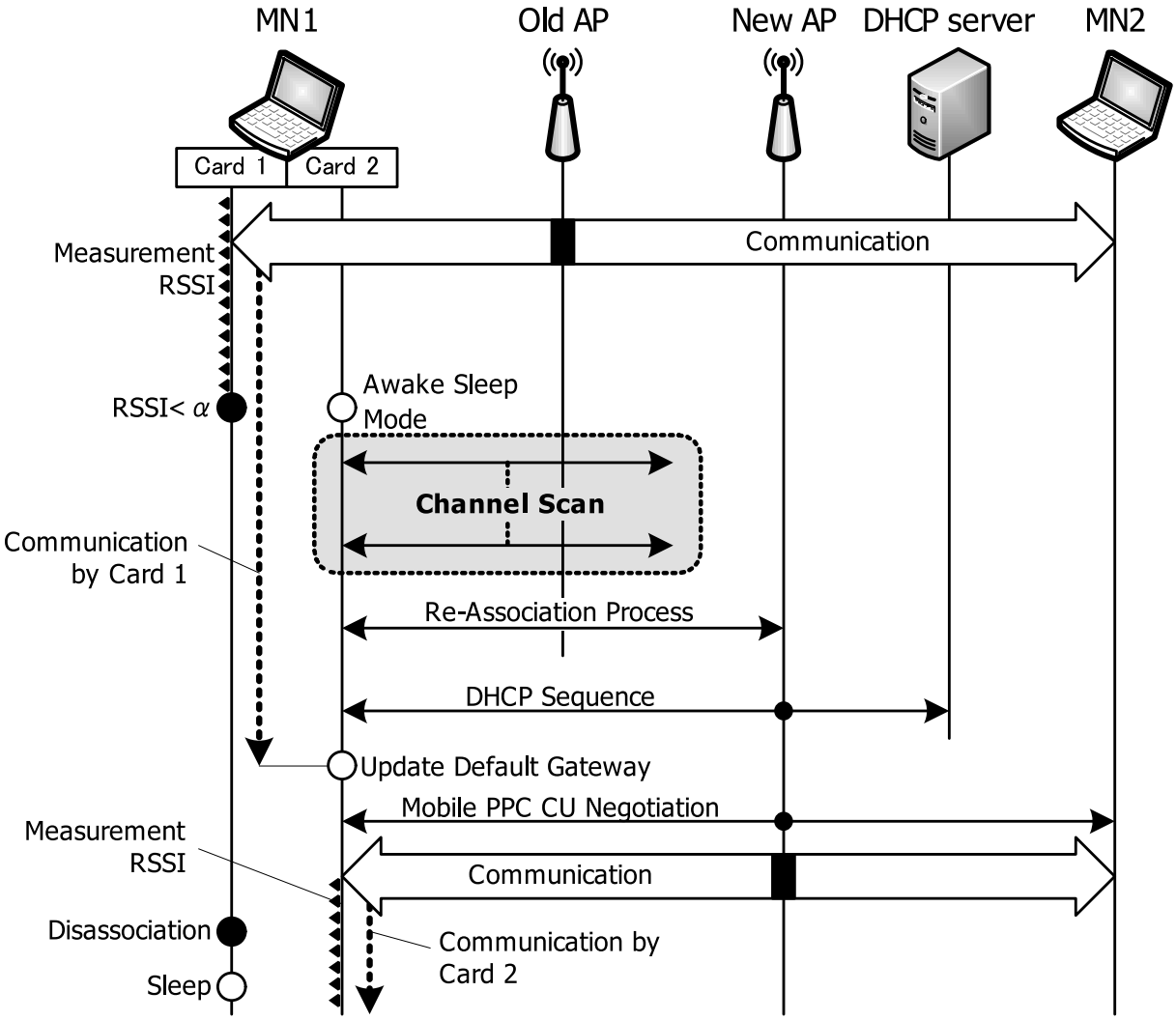

図 E.5　デュアルインタフェース方式によるエリア間ハンドオーバ

New AP と Old AP が異なるネットワークの場合，MN1 は Card1 による通信を継続しながら，Card 2 で再接続処理を行い New AP と接続し，DHCP サーバから新 IP アドレスを取得する．この様な仕組みを実現するためには，ルーチングテーブルに Old AP 側ネットワークのデフォルトルータ情報を維持しつつ，New AP 側ネットワークでアドレス取得をする必要がある．しかし，DHCP クライアントの種類によっては処理開始時にデフォルトルータの設定をクリアしてしまう場合がある．この場合，DHCP 処理は 2～数十秒を要するため，この期間は Card 1 側の通信を継続することができない．

そこで提案方式では，DHCP 処理実行時のデフォルトルータのクリアを無効とし，Mobile PPC の移動情報通知処理の直前にルーチングテーブル内のデフォルトルータの設定を更新する．これにより，DHCP 処理の時間に関わらず，Card 1 側の通信を継続することができる．Card 2 側で実行する DHCP 処理の送信はブロードキャストであるため，デフォルトルータの設定には影響されずに実行できる．ルーチングテーブルを更新後，Card 2 を用いて Mobile PPC の移動情報通知処理を実行して新 IP アドレスに対応する CIT を生成し，Card 2 を使用して通信を継続する．このとき

表 E.1　実験装置の仕様

	MN1	MN2
CPU	Pentium M 1.7 GHz	Pentium4 3.0 GHz
Memory	512 MByte	512 MByte
NIC	Intel 2915ABG (802.11g)	100BASE-TX
	Atheros 5212 (802.11g)	
OS	FreeBSD 6.1-RELEASE	FreeBSD 6.1-RELEASE

旧 IP アドレスに対応する CIT は，削除せず残しておく．以後の送信はすべて Card 2 から行われるが，受信は Card 1，2 のどちらからも可能である．

Card 1 は一定時間アソシエーションを維持した後に Old AP を切断する．受信したパケットの IP アドレスは新 IP アドレス宛の場合と旧 IP アドレス宛の場合がありうるが，CIT に基づいたアドレス変換が行われることにより，上位層には同一セッションの受信とみなされる．旧 IP アドレスに対応する CIT は，その後無通信状態となるためタイマにより自動的に消去される．MN1 は Card 1 と Old AP とのアソシエーションを切断した後は，Card 1 をスリープ状態にする．

ここで，図 E.5 中の移動情報通知処理を Card 1 側で実行することも可能であるが，移動情報通知は RSSI の高い Card 2 側で実行すべきと判断した．このため，移動情報通知処理の間に MN1 側から発生した送信パケットは，受信側 MN2 と CIT の内容が一致せず破棄される可能性がある．しかし，この時間は Mobile PPC では約 5 ms 程度であり，実用上の問題はないと判断できる．

E.3.2　性能評価結果

上記機能を実装した移動ノードを移動させてハンドオーバ処理を行わせ，所定の動作が可能であることを確認した．以下に試作の評価結果を示す．

提案方式の性能を測定するために，図 E.6 に示す試験環境でハンドオーバの実験を行った．DHCP サーバを搭載した 2 台の無線ルータ WR1，WR2 [4]によりサブネットが異なる 3 つのネットワークを用意した．表 E.1 に装置仕様を示す．MN1，MN2 には Mobile PPC を実装している．Iperf [167] により IP 電話（G.711）を想定したトラヒック，すなわちペイロード長 172 byte の UDP パケットを 50 packet/sec の頻度で双方向に送信しあう状況を作った．上記ストリームを流している際に，擬似的に MN1 が WR1 の無線セルから WR2 の無線セルへの移動を繰り返し，この間に発生するパケットロスを測定した．擬似的な移動を行わせるため，AP の電波強度はそのままとし，MN1 側で MN2 との通信開始後に取得した RSSI を閾値 α 未満となるように変化させて，カード切り替え処理を強制的に実行させた．

移動回数 20 回の測定結果は表 E.2 に示すとおり，すべての移動においてカードの切り替えに起因するパケットロスは MN1 から MN2，MN2 から MN1 の両方向とも 0 であった．

[4]BUFFALO 社製 WZR-G144NH.

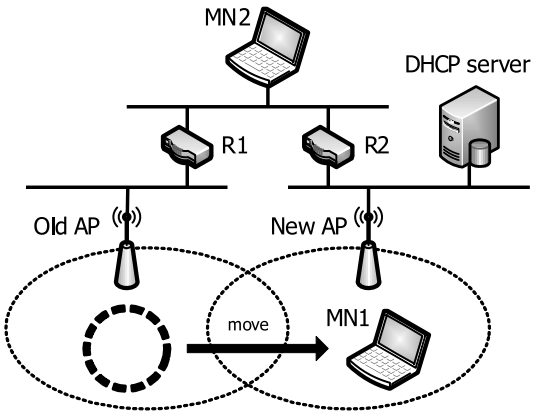

図 E.6　ハンドオーバ試験環境

表 E.2　カード切り替え時におけるパケットロスの測定結果

通信方向	送信パケット数（packet/sec）	パケットロス数
MN1→MN2	50	0
MN2→MN1	50	0

試行回数：20 回

E.3.3　電力消費に関する考察

デュアルインタフェース方式は，これまで電力消費が増加するという課題があった．しかしながら，本提案方式では通信中の無線 LAN カードを用いて RSSI の測定を行うため，チャネルスキャン実行側の無線 LAN カードは通常時はスリープ状態にしておけばよい．両カードが同時に動作するのはハンドオーバ時のみである．文献 [184] によると，無線 LAN チップの電力消費はパケット送信中が 543 mW，パケット受信中が 384 mW，受信待ち受け時が 263 mW であることが確認できる．それに対し，スリープ時の状態では無線 LAN カードへの漏れ電流のみで，電力消費は 57 μW とごくわずかとなる．このため，本提案方式では移動を繰り返さない限り無線 LAN カード 1 枚の場合と比較しても電力消費がほとんど増加することはない．

提案方式では，RSSI が閾値 α を下回る状態においてはチャネルスキャンを開始する．このとき，接続中の AP より電波強度の強い AP が見つからなかった場合は当該 AP との接続を維持し，チャネルスキャンを繰り返す必要がある．このような状況では，チャネルスキャンの周期 Ti を適切に設定し，待機中のカードのスリープを連続的に解除してしまうことがないようにする．ここでは，Ti を仮に 5 sec と設定した場合の電力消費の考察を行う．チャネルスキャンにかかる時間 Tcs は最大 600 msec なので [117]，チャネルスキャンを繰り返すときの電力消費は以下のように見積もることができる．ただし，チャネルスキャン実行側のカードの電力消費は通信中のカードと同程度と仮定する．

$$(Tcs + Ti)/Ti = 1.12 \tag{E.16}$$

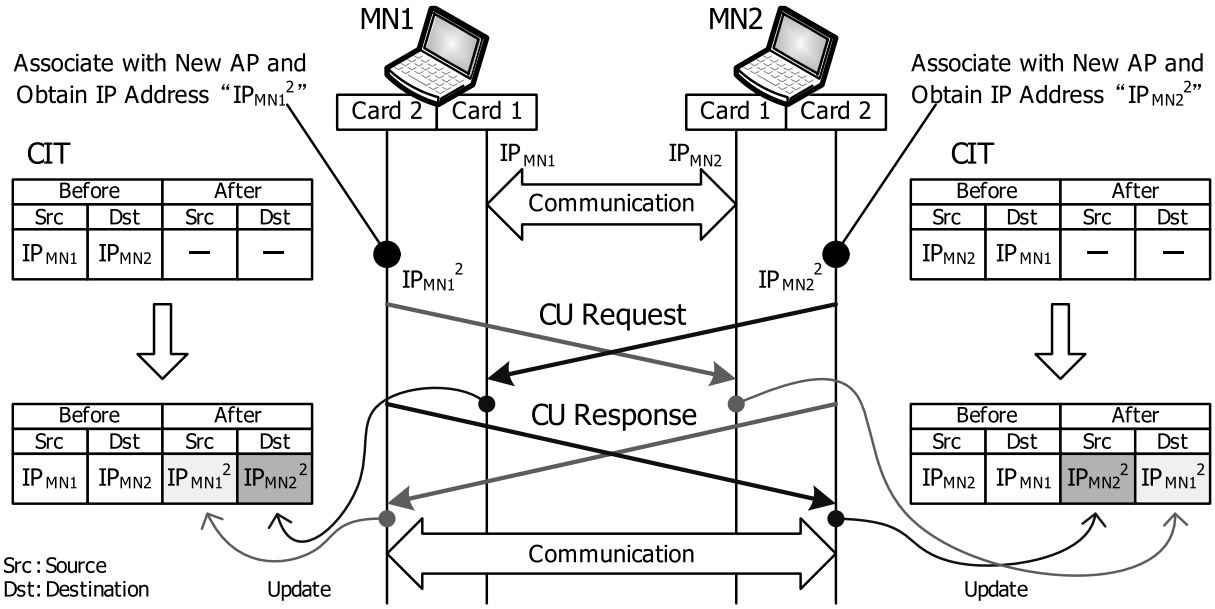

図 E.7　同時移動時における CIT の更新方法

すなわち，*Ti* が 5 sec の場合，カードの電力消費が最大 12 % 増加したのと同様となる．

E.3.4　Double Jump Problem の解決

2 台のエンドノードが全く同時に移動した場合，すなわち CU Request 送信後の CU Response 待ちの状態のときに相手側からの CU Request が到着するようなケースでは，従来のの Mobile PPC のままでは移動透過性を実現できなかった．デュアルインタフェース方式を導入することにより，同時移動が発生した時に生じる問題（Double Jump Problem）を解決できる．

図 E.7 に同時移動発生時に Mobile PPC の CIT が変化する様子を示す．いずれも Card 1（IP アドレス：IP_{MN1}，IP_{MN2}）で通信を行っており，新 IP アドレス IP_{MN1}^2，IP_{MN2}^2 がほぼ同時に Card 2 に割り当てられたものとする．CU Request はそれぞれ Card 2 側から送信され，通信相手ノードの Card 1 で受信される．CU Response はデフォルトルータが更新された後なので，Card 2 側から送信される．また CU Response の宛先は通信相手ノードの新 IP アドレスとするため，通信相手ノードの Card 2 で受信される．提案方式によりノードが CU 待ち状態時に 2 つの AP と接続しているため，上記のようにノードは旧 IP アドレス宛の CU Request を受信可能になった．

CIT の更新時に，これまでは CIT データの After 部分（移動後の自ノードと相手ノードの IP アドレス）をすべて更新していたため，同時移動時の CU Response 受信処理において，After/Destination を相手の移動前 IP アドレスに戻してしまう．そこで，CU Request 受信時は After/Destination（相手ノードの IP アドレス）部分のみ，CU Response 受信時は After/Source（自ノードの IP アドレス）部分のみを更新することとした．このような処理により同時移動を含む Mobile PPC の動作を統一的に扱うことが可能となり，通信の継続を実現できる．

E.4 プロキシ型 Mobile PPC

Mobile PPC を実装したプロキシサーバを GEP（GSCIP Element for Proxy）と呼ぶ．図 E.8 に GEP を利用する場合の Mobile PPC シーケンスを示す．MN と GEP は Mobile PPC を実装しており，MN が一般ノードである CN と通信を行うケースを想定する．

MN は CN の名前解決時に Support Check ネゴシエーションを行う．この結果，CN が Mobile PPC に非対応であることが確認できるので，以後 CN 宛の通信パケットを GEP に転送するように記憶する．MN のアプリケーションから IP 層に CN 宛 TCP/UDP パケットが渡されると，MN は GEP との間で認証鍵共有処理を実行する．さらに，CN 宛の通信パケットを GEP 宛となるようアドレス変換を行うために，MN と GEP は以下の CIT エントリを生成する．

$$\text{MN:} \qquad IP_{MN} : s \leftrightarrow \{IP_{CN} : d \overset{CIT}{\Longleftrightarrow} IP_{GEP} : d\} \quad [proto] \tag{E.17}$$

$$\text{GEP:} \qquad \{IP_{MN} : s \overset{CIT}{\Longleftrightarrow} IP_{GEP} : t\} \leftrightarrow \{IP_{GEP} : d \overset{CIT}{\Longleftrightarrow} IP_{CN} : d\} \quad [proto] \tag{E.18}$$

通常の Mobile PPC とは異なり，移動前から CIT に基づくアドレス変換処理が行われる．これにより，MN が CN へ送信する通信パケットは GEP へルーチングされ，GEP から CN へ転送される．CN は通信相手ノードを MN ではなく GEP として認識していることがポイントである．

MN 移動時は GEP に対して CU ネゴシエーションが行われ，CIT エントリが以下のように更新される．

$$\text{MN:} \qquad \{IP_{MN} : s \overset{CIT}{\Longleftrightarrow} {IP_{MN}}^2 : s\} \leftrightarrow \{IP_{CN} : d \overset{CIT}{\Longleftrightarrow} IP_{GEP} : d\} \quad [proto] \tag{E.19}$$

$$\text{GEP:} \qquad \{{IP_{MN}}^2 : s \overset{CIT}{\Longleftrightarrow} IP_{GEP} : t\} \leftrightarrow \{IP_{GEP} : d \overset{CIT}{\Longleftrightarrow} IP_{CN} : d\} \quad [proto] \tag{E.20}$$

上記 CIT エントリに基づいたアドレス変換を行うことにより，GEP は移動後の MN から受信した通信パケットを移動前と同様に CN へ転送する．上記パケットの送信元は常に GEP となり，MN の移動は CN に対して隠蔽される結果となる．

このように GEP を導入することにより，一般ノードと確立した通信に対しても移動透過性を保証できる．

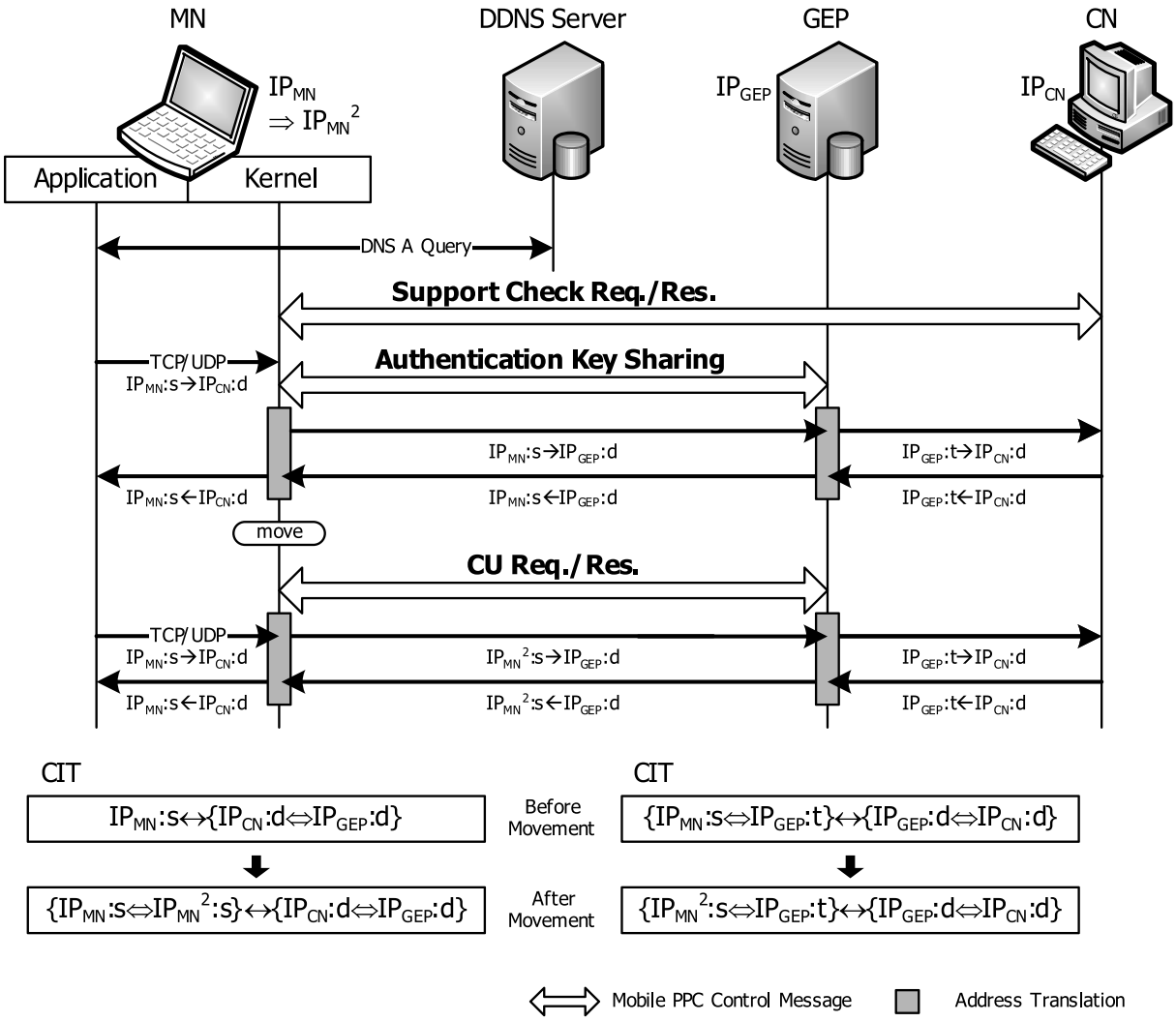

図 E.8　プロキシサーバを利用した Mobile PPC シーケンス

付 録F　NAT-fの関連研究

F.1　DLNA 機器の相互接続方式への応用

DLNA（Digital Living Network Alliance） [163] 準拠の情報家電機器が普及し，ユーザはホームネットワーク内の機器間で容易にメディアコンテンツを共有することができるようになった．しかし DLNA は規格上，同一ネットワーク内でしか利用することができない．

そこで，既に提案済みの NAT 越え技術 NAT-f（NAT-free protocol）に DLNA 機能を追加するアプローチにより，ホームネットワーク内外の DLNA 機器を相互に接続する方式を文献 [181] にて提案している．ここでは，DLNA の概要と課題を解説し，NAT-f の拡張方式について述べる．

F.1.1　DLNA

DLNA とは情報家電同士の相互接続に関するガイドラインを策定している標準化団体である．このガイドラインには各社製品が共通に対応すべきメディアフォーマット，情報家電の相互接続に用いる通信プロトコルやネットワークデバイスなどが規定されている．相互接続に用いる通信プロトコルとして，デバイスの検出や制御には UPnP（Universal Plug and Play） [143]，データ転送には HTTP がそれぞれ用いられている．

図 F.1 に DLNA 準拠の情報家電におけるデバイスの検出からコンテンツ再生までの一連の手順を示す．状態通知に関する手順は別途 GENA（General Event Notification Architecture） [185] で定義されているが，ここでは省略する．

(1) **デバイスの検出**

ユーザが DMP を起動すると，DMP は SSDP（Simple Service Discovery Protocol） [186] で定義された M-SEARCH メッセージをマルチキャストする．上記メッセージを受信した DMS は，自身の位置を示す URL（IP アドレスとポート番号）などの情報を 200 OK メッセージに含めて応答する．DMP は応答メッセージを受信することにより，ホームネットワーク内に存在する DMS を発見できる．

(2) **機器情報の取得**

DMS を発見後，DMP は取得した URL を宛先として HTTP GET メッセージを送信し，DMS から詳細な機器情報やサービス情報を XML ドキュメントとして取得する．以上の手順により，DMP の画面に DMS の情報が表示される．

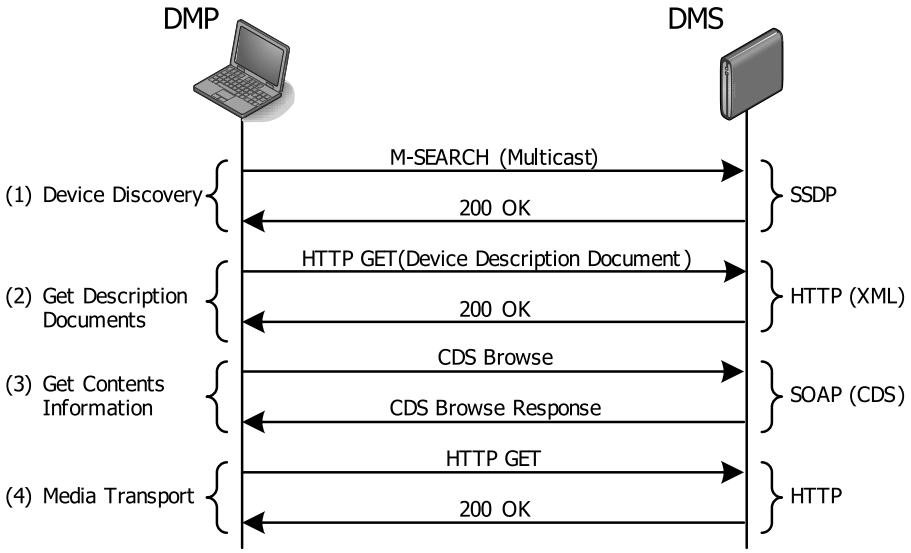

図 F.1　DLNA 準拠の情報家電の通信シーケンス

(3) **コンテンツの一覧情報の取得**

ユーザは表示された DMS を選択し, その DMS が保持するコンテンツを検索する．SOAP（Simple Object Access Protocol）[187] 及び CDS（Content Directory Service）[188] に従って Browse コマンドが送信され，DMS からコンテンツリストを取得する．

(4) **選択したコンテンツの伝送**

上記操作を繰り返し，ユーザは再生したいコンテンツを選択する．DMP と DMS 間は HTTP によりデータ転送が行われる．

F.1.2　DLNA の宅外利用における技術課題

DLNA ガイドラインは同一ホームネットワークにおける利用を前提としており，下記に示す技術課題がある．そのため宅外からホームネットワーク上の DMS のコンテンツを利用することができない．

課題 1: 手順 (1) の SSDP ではサイトローカルスコープマルチキャストアドレスを利用しているため，DMP と異なるネットワークに存在する DMS を検出することができない．

課題 2: 手順 (1) の DMS からの応答メッセージ内には DMS のプライベート IP アドレスが記載されている．従って，DMP は DMS に対して通信を開始できない．

課題 3: 手順 (2) で DMP は DMS に対して HTTP GET によりコンテンツを要求するが，DLNA 準拠の DMS は異なるネットワークからのアクセスを無視する規格となっている．従って，DMP はコンテンツを取得することができない．

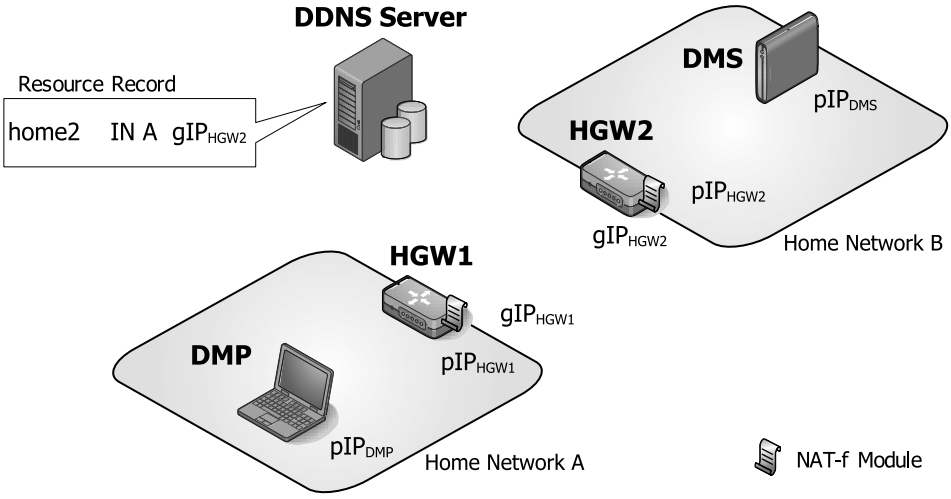

図 F.2　システム構成

課題 1 を解決するためには，新たな手法を導入して従来の SSDP による検出の仕組みを補完する必要がある．課題 2 はグローバルネットワーク上の端末からプライベートネットワーク内の端末に通信を開始することができるように，NAT 越え問題を解決する必要がある．課題 3 を解決するためには NAT 越え通信を実現し，かつ DMS に通信相手が同一ネットワーク上に存在しているように認識させる必要がある．

F.1.3　NAT-f の拡張による遠隔地 DLNA の相互接続方式

F.1.2 項に示した課題 1 を解決するために，NAT-f に新たに検索要求メッセージを定義する．DMS からの応答メッセージ内に記載されているプライベート IP アドレスを仮想 IP アドレスへ書き換える．これにより，課題 2 は NAT-f を適用することにより自ずと解決される．課題 3 を解決するために，パケットの送信元アドレスを変換できるように NAT-f を拡張する．

システム構成

図 F.2 にシステム構成を示す．本方式を実装した異なる HGW の配下に DLNA 準拠の DMP と DMS が設置されている．DDNS サーバには HGW2 の名前 “*home2*” とグローバル IP アドレス gIP_{HGW2} の対応関係が登録されているものとする．

このような構成において，DMP が DMS のコンテンツを共有するまでの手順を図 F.3 に示す．また，DMP と DMS 間の通信パケットの送信元及び宛先 IP アドレス，ポート番号が変化する様子を表 F.1 に示す．

1. **ユーザ認証手続き**

   ホームネットワーク A のユーザは HGW2 に対して認証手続きを行う．HGW1 は DDNS サーバより HGW2 の IP アドレスを取得してから，ユーザ ID とパスワードを送信する．HGW2

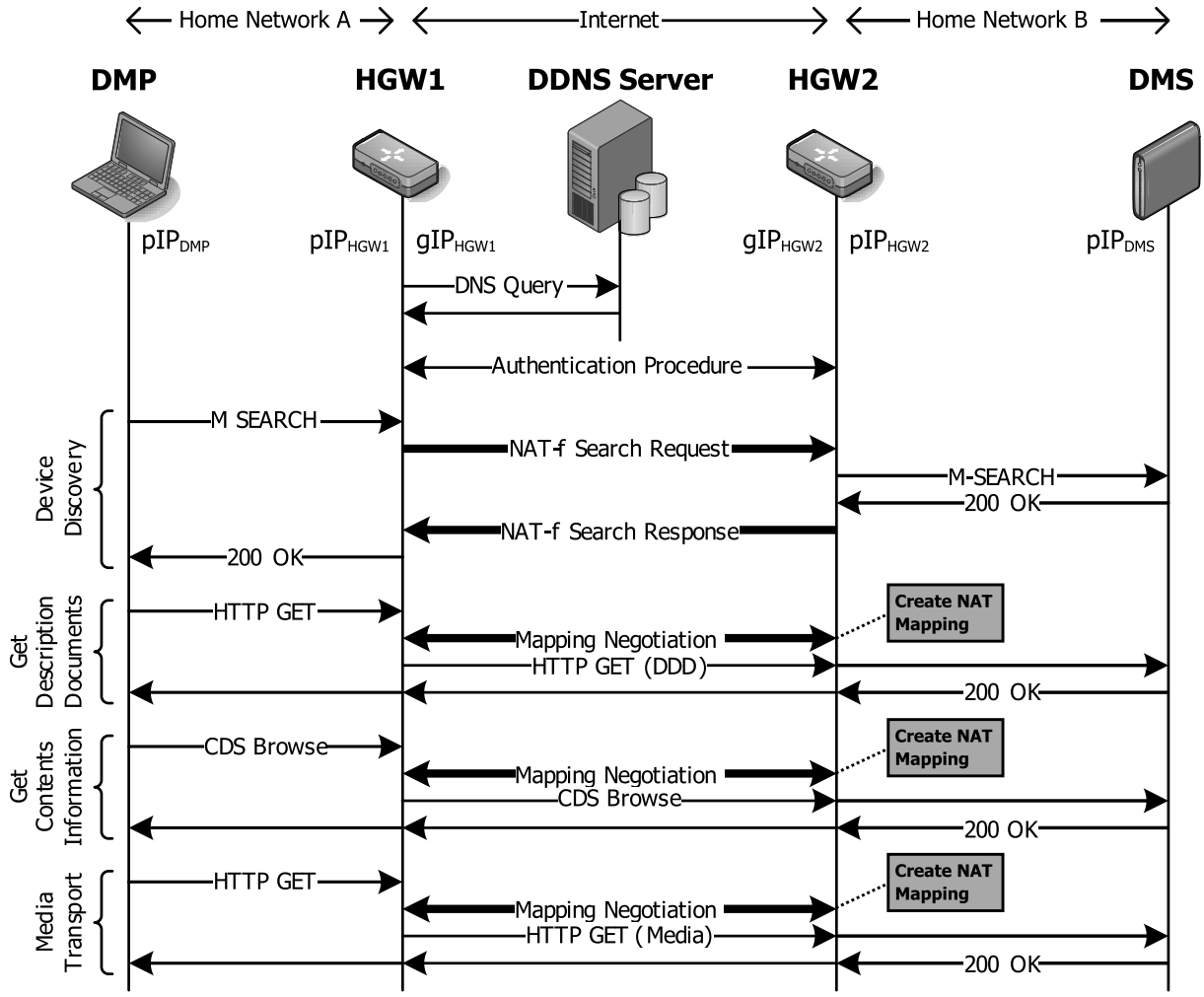

図 F.3　NAT-f によるホームネットワーク間相互接続シーケンス

はユーザ認証処理を行い，正規のユーザであればランダムなアクセスコードを生成し，返信する．HGW1 は受信したアクセスコードを記録する．ユーザ認証手続きにおける通信は TLS（Transport Layer Security） [3] により保護する．

2. **デバイスの検出**

ユーザ認証手続きを完了したら，DMP は通常の DLNA に基づく処理，すなわち M-SEARCH をマルチキャストする．HGW1 は M-SEARCH を受信すると，新たに定義した検出要求メッセージ Search Request を HGW2 へ送信する．上記メッセージには先ほど記録したアクセスコードが記載される．

HGW2 が Search Request を受信すると，記載されているアクセスコードを確認する．アクセスコードが正しければ，M-SEARCH を生成し，配下のホームネットワーク B へマルチキャストする．ここで，M-SEARCH の送信元を HGW2 とし，HGW2 が DMP の代理でデバイスの検出を行う．DMS からの 200 OK を受信した HGW2 は，これを内包した Search Response メッセージを生成し，HGW1 へ返信する．

表 F.1　DMP と DMS 間における通信パケットの送信元及び宛先の変遷

Message Type	Area		
	Home Network 1	Internet	Home Network 2
M-SEARCH	$pIP_{DMP} : s1 \rightarrow mIP : 1900$	— (NAT-f Search Request)	$pIP_{HGW2} : t1 \rightarrow mIP : 1900$
200 OK	$pIP_{DMP} : s1 \leftarrow vIP_{DMS} : 1900$	— (NAT-f Search Response)	$pIP_{HGW2} : t1 \leftarrow pIP_{DMS} : 1900$
HTTP GET	$pIP_{DMP} : s2 \rightarrow vIP_{DMS} : d1$	$gIP_{HGW1} : m1 \rightarrow gIP_{HGW2} : n1$	$pIP_{HGW2} : t2 \rightarrow pIP_{DMS} : d1$
200 OK (DDD)	$pIP_{DMP} : s2 \leftarrow vIP_{DMS} : d1$	$gIP_{HGW1} : m1 \leftarrow gIP_{HGW2} : n1$	$pIP_{HGW2} : t2 \leftarrow pIP_{DMS} : d1$
CDS Browse	$pIP_{DMP} : s3 \rightarrow vIP_{DMS} : d1$	$gIP_{HGW1} : m2 \rightarrow gIP_{HGW2} : n2$	$pIP_{HGW2} : t3 \rightarrow pIP_{DMS} : d1$
200 OK (Browse)	$pIP_{DMP} : s3 \leftarrow vIP_{DMS} : d1$	$gIP_{HGW1} : m2 \leftarrow gIP_{HGW2} : n2$	$pIP_{HGW2} : t3 \leftarrow pIP_{DMS} : d1$
HTTP GET (Media)	$pIP_{DMP} : s4 \rightarrow vIP_{DMS} : d2$	$gIP_{HGW1} : m3 \rightarrow gIP_{HGW2} : n3$	$pIP_{HGW2} : t4 \rightarrow pIP_{DMS} : d2$
200 OK (Media)	$pIP_{DMP} : s4 \leftarrow vIP_{DMS} : d2$	$gIP_{HGW1} : m3 \leftarrow gIP_{HGW2} : n3$	$pIP_{HGW2} : t4 \leftarrow pIP_{DMS} : d2$

mIP: Multicast address (239.255.255.250)　　*vIP*: Virtual IP address

Search Response を受信後, HGW1 は 200 OK を取り出し, メッセージ内に記載されている DMS のプライベート IP アドレス “pIP_{DMS}” を仮想 IP アドレス “vIP_{DMS}” に書き換える．vIP_{DMS} はホームネットワーク B のドメイン名と DMS のホストアドレスを用いて生成する．なお，200 OK メッセージに IP アドレスと併記されているポート番号（ここでは “$d1$” とする）はそのままとする．その後，HGW1 は割り当てた仮想 IP アドレスが DLNA 通信用であることを記録してから，200 OK を DMP に送信する．以上の処理により，DMP は他ホームネットワークに存在する DMS を検出することができる．

3. **NAT マッピングの生成**

DMP は 200 OK により取得した仮想 IP アドレス “$vIP_{DMS} : d1$” を宛先とした HTTP GET メッセージを送信する．仮想 IP アドレスのネットワークアドレスはホームネットワーク A と異なるため，上記パケットは必ずデフォルトゲートウェイである HGW1 へ送信される．

HGW1 は宛先が仮想 IP アドレスであるパケットを受信すると，NAT によるアドレス変換処理後，そのパケットを待避してから NAT-f のマッピングネゴシエーションを HGW2 に対して実行する．ここで，仮想 IP アドレスが DLNA 通信用であった場合，Mapping Request メッセージに DLNA 通信用のネゴシエーションであることを示すフラグを設定する．

上記フラグが設定された Mapping Request を受信した HGW2 は，通常とは異なる NAT マッピングを生成する．まず，宛先の DMS“$pIP_{DMS} : d2$” に対して，HGW2 の外部 IP アドレス・ポート番号 “$gIP_{HGW2} : n1$” をマッピングする．これは既存の NAT-f と同様である．さらに送信元の HGW1 の外部 IP アドレス・ポート番号 “$gIP_{HGW1} : m1$” を，HGW2 の内部 IP アドレス・ポート番号 “$pIP_{HGW2} : t1$” にマッピングする．その後，HGW2 は外側マッピングアドレス “$gIP_{HGW2} : n1$” を HGW1 に通知するため，Mapping Response メッセージを生成し送信する．

HGW1 は上記メッセージを受信したら，DMS に対して割り当てた仮想 IP アドレスを HGW2 の外側マッピングアドレスに変換するための VAT テーブルを生成する．

4. **DMS への NAT 越え通信**

マッピングネゴシエーションが完了したら，待避していた HTTP GET メッセージを VAT テー

ブルに基づいてアドレス変換して，HGW2 へ転送する．HGW2 は NAT マッピングに基づいて，送信元および宛先の両者をアドレス変換する．さらにメッセージ内に記載されている DMP の IP アドレス“pIP_{DMP}”を，HGW2 の内側 IP アドレス“pIP_{HGW2}”に書き換える．この結果，HGW1 から受信した HTTP GET メッセージは，送信元が HGW2 となり DMS へ転送される．

以上の処理により，DMS は同一ホームネットワークから要求があったと認識するため，通常の DLNA の手順に従って 200 OK メッセージを返答する．DMS からの応答メッセージは上記と逆の処理により，正しく DMP へ転送される．なお，200 OK に含まれる DMS の IP アドレスは，M-SEARCH に対する 200 OK と同様に，HGW1 において仮想 IP アドレスに書き換えられる．

以後のコンテンツの検索，及びコンテンツの再生時も同様の手順により，NAT 越え通信が行われる．このように提案方式は NAT-f の機能を拡張することにより，異なるホームネットワーク間において DLNA 準拠の情報家電のコンテンツを利用することができる．

F.1.4 関連研究

ホームネットワーク間でコンテンツの共有を可能とする既存技術には，以下のようなものが挙げられる．

W-DLNA [165] は，W-DLNA ゲートウェイ内に仮想的な DMP 及び DMS プロセスを生成し，相互に連携することにより異なるホームネットワークやインターネット上のモバイル機器から宅内の情報家電にアクセスすることを実現している．W-DLNA ゲートウェイ同士の SIP シグナリングにより生成された仮想 DMP 及び仮想 DMS をそれぞれ DMS 及び DMP に認識させる．コンテンツリストは SIP メッセージに内包することにより中継し，W-DLNA ゲートウェイ間で転送されるメッセージの送信元を仮想 DMP または仮想 DMS に変更する．これにより，DMP 及び DMS は通信相手が同一ネットワーク内に存在する場合と同様の手順でコンテンツを再生できる．

文献 [189] はワームホールデバイス（以後 WD）と呼ばれる装置をホームネットワーク内に設置し，異なる WD 同士が連携することにより，DMP は既存の通信手順のまま遠隔地にある DMS との通信を可能としている．接続先 WD に対しては SIP により機器情報の要求を行う．接続先 WD は自身のホームネットワーク内の DLNA 機器を検出し，その要約を応答する．WD は UPnP メッセージに含まれる IP アドレスを書き換えることにより，自身を DMP 及び DMS の通信相手となるように認識させる．DMP と DMS 間の通信は WD が中継することにより，コンテンツの再生を可能としている．この方式は携帯端末への応用も可能である．文献 [190] では WD の機能を携帯端末に実装することにより，モバイル機器からホームネットワーク内のコンテンツの再生を実現している．

Dial-to-Connect VPN システム [191,192] や DLNA Proxy システム [193] は，HGW 間に IPsec ESP によりセキュアな VPN 通信路を形成し，その上で DMP と DMS 間の通信を行う．

文献 [194] はモバイル GW と呼ぶ装置を導入し，DLNA 機器からの XML データを HTTP データに変換することにより，宅外の端末は WWW サービスと同じ要領でコンテンツを再生できるよ

うにしている．この方式は端末が DMP 機能を保持する必要はなく，WWW ブラウザを実装していればよいため，携帯電話などのモバイル機器での利用を想定している．

これらの他に，W-DLNA と同じく SIP により UPnP メッセージを転送する拡張 DLNA メディア共有システムアーキテクチャ [195]，UPnP メッセージを SOAP でカプセル化して転送する方式 [196]，Peer-to-Peer でコンテンツの共有を可能とする拡張 UPnP アーキテクチャ xUPnP [197] などがあり，その実現方法は多岐にわたっている．