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

鈴木秀和[†] 宇佐見庄五[†] 渡邊晃[†]

インターネット利用形態の多様化により，IP 電話やマルチメディア通信など個人間を主体とした P2P 通信の需要が高まっている．しかし通信相手ノードが NAT 配下に存在する場合，インターネット側から通信を開始することができない．このため NAT 配下のノードとコネクションを確立する NAT 越え技術が要求されている．これまでの NAT 越え技術は，アプリケーションに依存した限定的な方式が多く提案されている．また，特有の装置を導入し，パケットのカプセル化や中継転送を行うなどの方式も提案されているが，P2P 通信の特徴を大きく損なうなどの課題がある．本論文では，このような課題を解決するため，外部ノードから NAT に対してマッピング処理を指示する外部動的マッピング方式を提案する．これを実現するためのプロトコルとして NAT-f（NAT-free protocol）を定義した．提案方式は，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．

Proposal and Implementation of NAT-f for Realizing NAT Traversal Communication with External Dynamic Mapping Method

Hidekazu Suzuki,[†] Shogo Usami[†] and Akira Watanabe[†]

There are growing demands for P2P communications like IP telephony and multimedia communications due to the diversification of the Internet. However, we cannot initiate communications to nodes located behind a Network Address Translator (NAT) from the Internet side. Therefore there needs a NAT traversal technology that can establish connections between the nodes. Previous technologies often depend on applications and are not versatile enough. Alternatives that do not depend on applications severely spoil the efficiency of P2P communications because they need a specific server that relays packets. In this paper, we propose an external dynamic mapping method to solve the NAT traversal problem. We also define NAT-free protocol (NAT-f) to realize the method. NAT mapping is created with the negotiation between an external node and NAT in advance of the communication. The kernel in the external node translates the address/port numbers in the sending packets into the mapped-address. We have implemented the trial system and evaluated initial delay and throughput between end nodes. As a result, the increase of the initial delay time was less than 1 ms, and the throughput performance was as same as the case that proposed method is not implemented.

1. はじめに

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，企業や家庭など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）[1] 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

† 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University

インフラの普及にともない，IP 電話やマルチメディア通信など，個人間の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 のマッピングをあらかじ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うえ，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とされず，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/v6 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．ここで本論文における NAT とは，ポート番号の変換も行う NAPT（Network Address Port Translator）[2] 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表記する．

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，両ノードがとも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（NAT 外側の IP アドレスとマッピング時に割り当てられたポート番号の組み）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 のほかに，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，内部ノード側はあらかじ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ったりす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，内部ノードへの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．

我々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．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，2 章で既存の NAT 越え技術を分類し，その概要と課題について整理する．3 章で外部動的マッピング方式を提案する．4 章では NAT-f の実装について述べ，5 章で提案方式の動作検証と性能評価の結果

を示す．最後に 6 章でまとめる．

2. NAT 越え問題に関わる関連研究

本章では既存の 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），両ノードがとも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 3) 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☆に対応できることから，すで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☆☆に対応できなかったりするなどの課題がある．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）[4] がある．このほか，Hole punching を利用した IPv6 over UDP/IPv4 技術として Teredo [5] 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[6)~8)] 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，すべての通信が 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ったりす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）[9] 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）[10] や NAT Port Mapping Protocol [11] がある．

(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）[12] 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）[13] 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）[14] 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）[15] 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 トンネる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いっさい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）[16] がある．

(6) IP ルーティング拡張方式

この方式は IP のルーティング方式を拡張することに

☆ 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．
☆☆ 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．

表 1 NAT 越えの既存技術とその実装箇所
Table 1 Existing technologies for NAT traversal and their implementation point.

	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 *
ICE	APP	SIP 拡張	✓	✓	—	STUN/TURN サーバ
AVES	NET	トンネリング	—	—	✓	AVES 対応 DNS サーバ，waypoint
4+4	NET	IP ルーティング拡張	✓	✓	✓	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
* 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）

表 2 NAT 越え技術の要求条件と既存技術の満足度
Table 2 Requirements for NAT traversal and satisfaction degree of the existing technologies.

	アプリケーションレベル				ネットワークレベル	
	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	×	○	×	○	—*	—*
遅延	○	×	○	△	×	○
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．

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 のすべ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が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，ほかのシステムに影響を及ぼす可能性がある．IP ルーティング拡張方式として，4+4[17] や IPNL（for IP Next Layer）[18] などがある．

表 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，すでに多くの NAT に実装されているため，導入は容易である．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を問わず，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，すべてのカーネルを改造しなければならない．

表 2 に本論文における 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 ○，△，× は各要求条件を満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，ほか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．たとえば，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 および 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 はすべ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 19) 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 は上記でも述べたとおり，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．したがって，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 は独自のルーティング処理を行うが，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[17]．したがって，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．

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．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．

3.1 初期登録情報

図 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☆，IN への機能実装は行わない．ダイナミック DNS（以下 DDNS）サーバ[20] は IN の名前解決のために利用する．NAT 越えのための機能はいっさい不要であり，すでに運用されているものをそのまま利用できる．

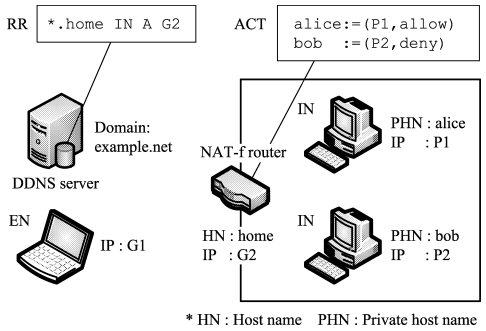

図 1 提案方式のシステム構成と初期設定情報
Fig. 1 System configuration and initialization.

表 3 IN が複数存在する場合の DNS 登録パターン
Table 3 DNS registration patterns when plural INs exist.

方法	登録内容			
wildcard A	*.home	IN	A	G2
CNAME	home	IN	A	G2
	alice.home	IN	CNAME	home
	bob.home	IN	CNAME	home
複数の A	alice	IN	A	G2
	bob	IN	A	G2

事前準備として，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 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が利用できる場合，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[21]．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．

☆ NAT-f の機能を実装した NAT を NAT-f ルータと呼ぶ．

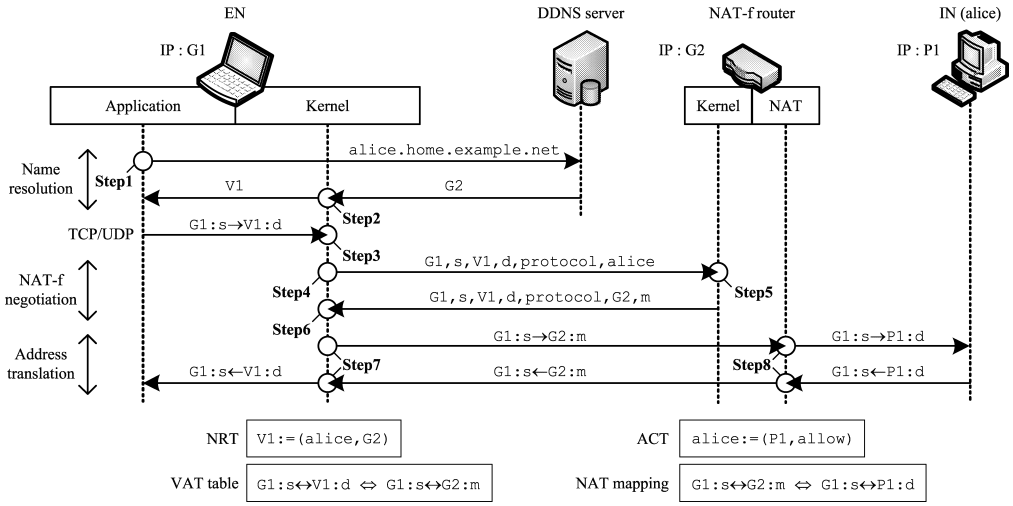

図 2 提案方式における通信シーケンス
Fig. 2 Communication sequence with the proposal method.

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．本論文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

$$alice := (P1, allow) \qquad bob := (P2, deny)$$

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☆．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．

3.2 動作概要

図 2 に alice への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．ここで，

$$A : a \rightarrow B : b$$

は IP アドレス A のノードから IP アドレス B のノードへの通信で，送信元/宛先ポート番号がそれぞれ a, b であることを意味する．提案方式における通信は以下に示す (1)〜(3) の 3 フェーズから構成される．

(1) **DNS 名前解決処理**

Step1：EN は IN 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 アドレスを応答する．

Step2：EN はカーネルにおいて DNS 応答パケットをフッキングして，取得した NAT-f ルータの IP アドレスを仮想 IP アドレス $V1$ に書き換える．このとき仮想 IP アドレスは，

$$V1 := (alice, G2)$$

のように IN 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 3.3 節で述べる．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**

Step3：EN は宛先 IP アドレスが仮想 IP アドレスとなっている TCP/UDP パケットを送信する際，カーネルにおいて送信元/宛先 IP アドレスとポート番号，およびプロトコルタイプより仮想アドレス変換（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

☆ 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．

了時に生成される．該当する情報が存在すれば，Step7 の動作を，該当する情報が存在しなければ，Step4 の動作を行う．

Step4：宛先 IP アドレス $V1$ より NRT を参照して仮想 IP アドレスに関連付けられた情報を取得する．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実行する．EN はネゴシエーションのトリガとなった TCP/UDP パケットの送信元/宛先 IP アドレスとポート番号，プロトコルタイプ，および IN のプライベートホスト名をマッピング要求パケットに載せて NAT-f ルータに通知する．

Step5：NAT-f ルータはマッピング要求パケットから，IN の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

$$G1 : s \leftrightarrow G2 : m \Longleftrightarrow G1 : s \leftrightarrow P1 : d$$

のようにマッピング情報を生成する．これは IP アドレス/ポート番号が $G1 : s$ および $P1 : d$，すなわち EN と IN（alice）間の通信に対応するマッピングアドレスが $G2 : m$ 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マッピング応答パケットに載せて EN へ応答する．

Step6：EN は NAT-f ルータからマッピング応答パケットを受信すると，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

$$G1 : s \leftrightarrow V1 : d \Longleftrightarrow G1 : s \leftrightarrow G2 : m$$

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

Step7：復帰した TCP/UDP パケットは 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 $V1 : d$ から $G2 : m$ に変換された後，送信される．

Step8：NAT-f ルータはすでにマッピングされているため，通常の NAT 処理により宛先 IP アドレスとポート番号を $G2 : m$ から $P1 : d$ に変換して，該当する IN（alice）へパケットを転送する．

以上の処理により，EN から IN へのパケット転送が完了する．IN 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，マッピング情報および 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 IN 間の通信は Step7，Step8 を繰り返す．このようにして EN から IN 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．

3.3 同時通信

図 1 において EN が alice と通信中に同一ホームネットワークの 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 $V2$ に書き換えて，アプリケーションに報告する．アプリケーションは宛先 IP アドレスを alice 宛なら $V1$，bob 宛なら $V2$ として設定するため，カーネルでは NAT 配下のどのノードに対して 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できる．NAT-f ルータは Step5 で示したとおり，EN から通知された IN 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アドレスが割り当てられる．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．

3.4 異なるホームネットワーク内の IN 間の通信

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．

図 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．IN（alice と carol）は同一のプライベート IP アドレス $P1$ とする．図 3 と図 2 の Step は対応しており，同一の処理を行う．NAT-f router1 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（Step2.5）が行われる．

Step2.5：TCP/UDP パケットは通常の NAT 処理に従って，送信元 IP アドレスとポート番号が $P1 : s$ から $G1 : m$ に変換された後，カーネルに渡される．このとき，生成されるマッピング情報は

$$P1 : s \leftrightarrow V1 : d \Longleftrightarrow G1 : m \leftrightarrow V1 : d$$

のように示される．

以後，3.2 節の手順により，NAT-f router2 には NAT マッピング情報

$$G1 : m \leftrightarrow G2 : n \Longleftrightarrow G1 : m \leftrightarrow P1 : d$$

が，NAT-f router1 には VAT エントリ

$$G1 : m \leftrightarrow V1 : d \Longleftrightarrow G1 : m \leftrightarrow G2 : n$$

が生成される．carol から alice への通信は，NAT-f router1 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 router2 において，NAT

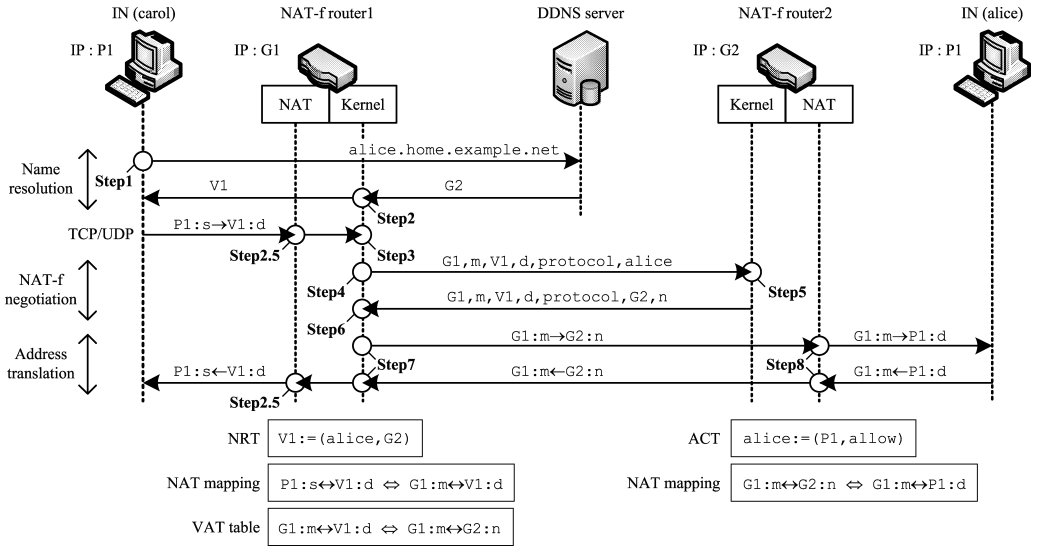

図 3 プライベートネットワーク間の通信シーケンス
Fig. 3 Sequence of Private-to-Private communication.

によるアドレス変換が行われ，alice 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．

3.5 既存技術との比較

表 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，ほかのネットワークレベルの技術との互換性がある．また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．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 アドレスととも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すべ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 を除くほか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，きわめて短時間で終了し，通信開始時のほかの処理に与える影響はほとんどない．NAT-f にかかる時間とスループットの実測値に関しては，5.2 節で示す．

表 4 NAT 越え要求条件に対する提案方式の満足度
Table 4 Satisfaction degree of the proposed method to the requirements for NAT traversal.

	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	○
遅延	○
スループット	○

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．これは今後いっそう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 はこのサーバに問合せを行えば，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 1 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 を DDNS に置き換えた構成になる．DNS はすべ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 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，ルーティングに必要な 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 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 でもかまわない．

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 の IP 層に実装した．**図 4** に EN の実装概要を示す．NAT-f モジュールは IP 層の入出力関数 `ip_input()`，`ip_output()` から呼び出され，NAT-f 対応の処理を行い，パケットを元の場所に差し戻す．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，カーネル内にとどめておく．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ともなう遅延を最小限に抑えることが可能になる．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作成する．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．

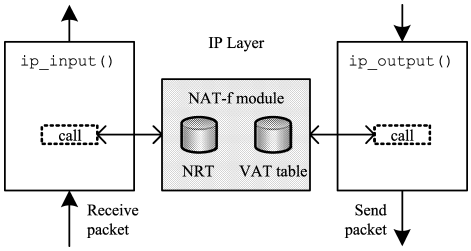

図 4 EN の実装概要
Fig. 4 Implementation on EN.

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☆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 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 デーモン natd ☆☆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

☆ 通常，クラス C である 192 が割り当てられている．
☆☆ 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．

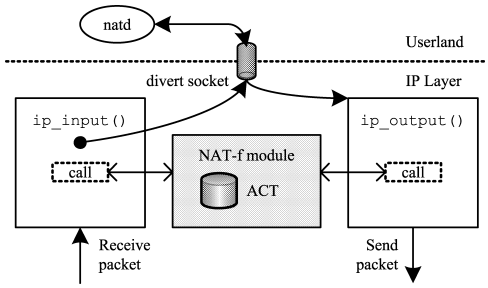

図 5 NAT-f ルータの実装概要
Fig. 5 Implementation on NAT-f router.

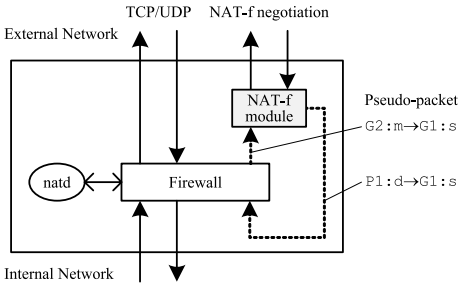

図 6 疑似パケットによる NAT マッピング手法
Fig. 6 NAT mapping method with pseudo-packet.

ドレス変換処理が行われる．**図 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 ACT の内容から

$P1 : d \rightarrow G1 : s$

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と呼ぶ．疑似パケットは IN（alice）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 NAT-f モジュールへ渡され，NAT-f ネゴシエーションの応答パケットの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，ネゴシエーションの応答パケットが生成された後，破棄される．この手法によると，NAT 処理にはいっさいの変更を必要としないため，Packet Filter などほかの 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 秒，コネクション確立後の TCP が 86,400 秒（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 によるアドレス割当て時や，NAT-f ルータが NBNS（NetBIOS Name Server）[22] 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 ネゴシエーションは ICMP ECHO パケットをベースとした制御パケットを用いて行われる．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ことを示すエントリ

$G1 : s \leftrightarrow V1 : d \Longleftrightarrow G1 : s \leftrightarrow G2 : d$

を生成し，VAT テーブルに格納する．以後，EN は NAT-f ルータとの通信を開始する．

5. 動作検証と評価

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．

5.2 性能評価

図 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 MB 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 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）[23] 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 [24] 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

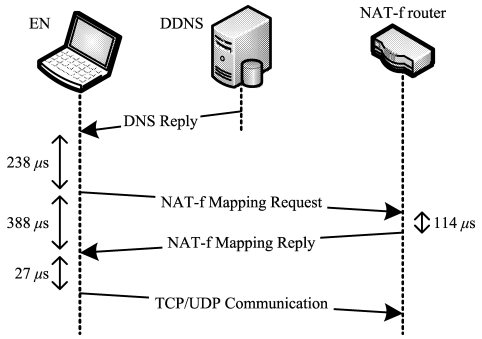

図 7 通信開始時のオーバヘッド
Fig. 7 Overhead when EN starts communication.

表 5 Netperf によるスループット測定値
Table 5 Throughput on the proposal method using Netperf.

Message Size (Bytes)	TCP Stream (Mbps)		UDP Stream (Mbps)	
	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

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 7 に通信開始時のオーバヘッドを示す．EN が DNS 応答パケットを受信してからマッピング要求パケットを送信するまでの時間は 238 μs であった．このうち，DNS 応答書き換え処理（Step2）が 7.13 μs，NAT-f ネゴシエーション開始処理（Step3+Step4）が 3.38 μs を占めていた．また EN がマッピング要求パケットを送信してからマッピング応答パケットを受信するまでの時間は 388 μs であり，NAT-f ルータのネゴシエーション処理時間（Step5）は 114 μs であった．EN はマッピング応答パケット受信後，27 μs 後（Step6+Step7）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 μs☆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分かる．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 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 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．

☆ $238 + 388 + 27 \fallingdotseq 650\,\mu$s

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 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 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．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[25] がある．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．マッピング要求パケットは図 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 ルータはマッピング要求パ

ケットに対してのみ正しく動作することができる．また，大量のマッピング要求パケットを送りつけられる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シーケンスの中でクッキーの交換を行ったりする[26]などの方式を導入する必要があると考えられる．

6. まとめ

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 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．

今後はDoS攻撃への防止対策や，認証および暗号化技術の適用などについて検討を行っていく予定である．

参考文献

1) Egevang, K. and Francis, P.: The IP Network Address Translator (NAT), RFC 1631, IETF (1994).
2) Srisuresh, P. and Holdrege, M.: IP Network Address Translator (NAT) Terminology and Considerations, RFC 2663, IETF (1999).
3) Ford, B., Srisuresh, P. and Kegel, D.: Peer-to-Peer Communication Across Network Address Translators, *Proc. USENIX Annual Technical Conference*, pp.179–192 (2005).
4) 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).
5) Huitema, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs), RFC 4380, IETF (2006).
6) Guha, S. and Francis, P.: Simple Traversal of UDP Through NATs and TCP too (STUNT), Internet-draft, IETF (2004). draft-guha-STUNT-00.txt
7) Guha, S. and Francis, P.: Characterization and Measurement of TCP Traversal through NATs and Firewalls, *Proc. ACM International Measurement Conference* (*IMC*), pp.199–211 (2005).
8) Takeda, Y.: Symmetric NAT Traversal using STUN, Internet-draft, IETF (2003). draft-takeda-symmetric-nat-traversal-00.txt
9) Rosenberg, J., Mahy, R. and Huitema, C.: Traversal Using Relay NAT (TURN), Internet-draft, IETF (2005). draft-rosenberg-midcom-turn-08.txt
10) UPnP Forum: *Internet Gateway Device* (*IGD*) *Standardized Device Control Protocol V 1.0* (2001). http://www.upnp.org/standardizeddcps/igd.asp
11) Cheshire, S., Krochmal, M. and Sekar, K.: NAT Port Mapping Protocol (NAT-PMP), Internet-draft, IETF (2006). draft-cheshire-nat-pmp-02.txt
12) 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).
13) Guha, S., Takeda, Y. and Francis, P.: NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity, *Proc. ACM SIGCOMM workshop on Future Directions in Network Architecture* (*FDNA*), pp.43–48 (2004).
14) Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, Internet-draft, IETF (2006). draft-ietf-mmusic-ice-12.txt
15) Ng, T., Stoica, I. and Zhang, H.: A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces, *Proc. USENIX Annual Technical Conference*, pp.319–332 (2001).
16) Kondo, K.: Capsulated Network Address Translation with Sub-Address (C-NATS), Internet-draft, IETF (2003). draft-kuniaki-capsulated-nats-05.txt
17) 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).
18) Francis, P. and Gummadi, R.: IPNL: A NAT-Extended Internet Architecture, *Proc. ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.69–80 (2001).
19) 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，Vol.103, No.716, pp.67–70 (2004).
20) Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).
21) Lewis, E.: The Role of Wildcards in the Domain Name System, RFC 4592, IETF (2006).
22) NetBIOS Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board and End-to-End Services Task Force: Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods, RFC 1001, IETF (1987).
23) Intel Corp.: *Using the RDTSC Instruction for Performance Monitoring* (1998). http://developer.intel.com/drg/pentiumII/appnotes/RDTSCPM1.htm
24) Jones, R.: Netperf: A network performance monitoring tool. http://www.netperf.org/netperf/NetperfPage.html
25) 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).
26) Harkins, D. and Carrel, D.: The Internet Key Exchange (IKE), RFC 2409, IETF (1998).

(平成 19 年 3 月 28 日受付)
(平成 19 年 9 月 3 日採録)

鈴木 秀和（学生会員）

2004 年名城大学理工学部情報科学科卒業．2006 年同大学大学院理工学研究科情報科学専攻修了．現在，同大学院理工学研究科電気電子・情報・材料工学専攻後期課程に在学中．ネットワークセキュリティ，モバイルネットワーク等の研究に従事．修士（工学）．2006 年 IEEE 名古屋支部学生奨励賞受賞．2006 年 DICOMO2006 松下温賞受賞．2007 年 DICOMO2007 ヤングリサーチャー賞受賞．電子情報通信学会所属．

宇佐見庄五（正会員）

1997 年名古屋工業大学知能情報システム学科卒業．2002 年同大学大学院博士後期課程修了．同年同大学院ベンチャービジネスラボラトリ研究員．2004 年より名城大学理工学部情報工学科講師．現在，同大学助教．量子情報理論，符号理論の研究に従事．博士（工学）．電子情報通信学会所属．

渡邊 晃（正会員）

1974 年慶應義塾大学工学部電気工学科卒業．1976 年同大学大学院工学研究科修士課程修了．同年三菱電機株式会社入社後，LAN システムの開発・設計に従事．1991 年同社情報技術総合研究所に移籍し，ルータ，ネットワークセキュリティ等の研究に従事．2002 年名城大学理工学部教授，現在に至る．博士（工学）．電子情報通信学会，IEEE 各会員．