平成20年度　修士論文

邦文題目

端末に依存しないNAT越え通信に関する研究

英文題目

Research on the NAT Traversal communication independent of user terminals

情報科学専攻

(学籍番号: 073432029)

宮崎　悠

名城大学名城大学大学院理工学研究科

内容要旨

インターネットの利用形態が多様化し，いつでもどこからでもネットワークにアクセスしたいという需要が高まっている．そこでは外出先からでも家庭内や企業内の端末に自由にアクセスしたいというニーズが考えられる．しかし，家庭内や企業内のネットワークはプライベートアドレスで構築される場合が一般であり，通信経路上に必ず NAT が存在する．このような環境ではインターネット側の端末からプライベートアドレスの端末に対して通信を開始することができないという問題がある．これは NAT 越え問題と呼ばれる．これまでの NAT 越え技術は，アプリケーションに依存した方式や，特有の装置を導入し，トンネリングや中継を行う方式も提案されているが，これらはユーザ端末に機能を実装する必要がある．本論文では，DNS サーバと NAT ルータを改造し，両者が連携することにより，ユーザ端末に機能を実装することなく NAT 越えを実現する方式を提案する．これを実現するプロトコルとして NTS (NAT Traversal Support) 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，名前解決を行った DNS サーバが事前に情報を与えることで，外部ノードからの通信により NAT テーブルを生成し，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，動作検証，測定を行った結果，通信開始時のオーバヘッドは 1ms 以内であり，通常の NAT ルータに対してほとんど影響のないスループットを実現することを確認した．

目 次

図 目 次

表 目 次

第1章　はじめに

IPv4 ネットワークでは IP アドレスの枯渇を回避するため，家庭内や企業内のネットワークはプライベートアドレスで構築する形態が一般となっている．それらのネットワークとインターネットの間にはアドレス変換装置 (以下 NAT：Network Address Translator) [1] が必要である．しかし，このような環境ではインターネット側の端末からプライベートアドレス空間の内部が見えなくなるため，NAT の外側の端末から内側の端末へ通信を開始することができないという制約がある．これは NAT 越え問題と呼ばれている．これまで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端末側から通信を開始していた．ファイアウォールでもこのような通信形態のみを許可するのが一般的であったため，NAT の制約が表面化することはなかった．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通信が増加し，外出先から家庭内の端末に自由にアクセスしたいというニーズが十分に考えられる．このため IPv4 ネットワークにおいて NAT 越え問題を解決する必要性は高まっている．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 配下の端末を内部ノード，NAT の外部に存在する端末を外部ノードと表記する．

NAT のアドレス変換テーブル（NAT テーブル）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生成される．また，そもそも外部ノード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，内部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 テーブルを予め静的に設定しておく方法があるが，ポート番号 1 つに対して 1 台の内部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．

IPv4 アドレスの枯渇を回避するために IPv6 が検討されているが，IPv4 が既に広く浸透しており，IPv4 と IPv6 の互換性がないことから，未だに IPv6 技術の導入はほとんど進んでいない．また，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，NAT の利用は今後も避けられない．現段階の IPv4 における解決方法として，ISP（インターネットサービスプロバイダ）等の電気通信事業者レベルで NAT を行い，プライベート IP アドレスを利用する Carrier Grade NAT [3] が検討されている．これはサービスプロバイダレベルで NAT を使用し，できるだけ一般家庭へプライベート IP アドレスを割り当てる方法である．しかし，より多数の端末で NATS ルータの IP アドレスを共有

することになるため，NAT 越え問題に加え利用可能なポート数が制限されるという課題がある．今後の利用形態の多様化を考慮すれば，IPv4 における NAT の制約を除去することは有益である．

NAT 越え問題を解決する為にこれまで様々な解決手法が提案されているが，大別するとアプリケーションレベルの解決方法とネットワークレベルの解決方法に分類できる．アプリケーションレベルの解決方法とは，インターネット上に専用の特殊なサーバを用意し，エンドノードの通信を行うアプリケーションが NAT 越え通信のために特殊なやり取りを講じることで解決する手法である．内部ノードからのアクションにより，NAT ルータではアドレス・ポート変換のマッピングが行われ，専用サーバにそのマッピング情報が通知される．外部ノードは実装されたアプリケーションにより，専用サーバから内部ノードのマッピング情報を取得し，その情報に対応するパケットを送信することにより NAT 越え通信を実現する．この方式は使用する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情報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方法とは，NAT に独自の機能を実装することで NAT 越え通信を実現する方法である．エンドノードや専用サーバに機能を実装し，それらが情報交換を行い，ユーザが使用したアプリケーションにより生成されたパケットを変換するか，NAT のマッピング機能を独自の処理に置き換えて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．そのためアプリケーションに依存しない汎用性を提供できる．またアプリケーションレベルの解決方法のように，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，独自処理によりる通信遅延の増加やスループットが低下など，解決方法に特化した専用機器が必要になるなどの課題がある．

今後も様々なネットワークサービスが誕生することが予想され，それらは各ネットワークの内外に囚われず利用できることが望まれる．そこで我々はネットワークレベルの解決方法に着目する．ネットワークレベルの解決方法として 4+4 [4] や NAT-f (NAT-free protocol) [5]，AVES(Address Virtualization Enabling Service) [6] 等がある．以後，外部ノードを EN（External Node），内部ノードを IN（Internal Node），NAT 越え通信実現に必要な専用サーバを SS（Special Server）と略する．

4+4 では 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，プライベート IP アドレスとグローバル IP アドレスを入れ替えて転送することにより IN への通信を可能とする．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り，プライベート IP アドレスも登録/通知できるように DNS も変更する必要があるため，実用難易度に課題がある．

NAT-f では EN と NAT に機能を実装し，EN からの通信開始時に動的に NAT のマッピングを行うことにより NAT 越え通信を実現する．EN が通信を開始する際，DNS の名前

解決をトリガにして EN と NAT の間でマッピングに必要な情報を交換し，強制的に NAT のマッピングを行う．その後 EN 側で NAT のマッピングに合わせたパケットを生成することにより IN との通信を可能とする．NAT-f は SS が不要できるが，EN のカーネルに実装が必要であり，一般ユーザに適用することは困難である．

AVES では AVES 対応 DNS サーバと waypoint と呼ぶ SS を導入し，EN は AVES 対応 DNS サーバに IN の名前解決を行う．AVES 対応 DNS サーバは waypoint と情報交換してから EN へ waypoint のアドレスを返すことで，EN は 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 のプライベート IP アドレス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送信する．NAT はそのパケットをデカプセリングして IN へ転送することにより NAT 越え通信を実現する．

今後は情報家電やモバイル端末の普及により，ユーザが勝手に機能を追加できない端末との通信も要求されることも予想されるため，AVES の様に可能な限りユーザの使用する端末には手を加えないことが望ましい．しかし，AVES は中継転送やカプセリング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．そこで本論文では改造した DNS サーバ [7, 8] と NAT ルータが協調し，外部端末からの通信により NAT テーブルを生成することで，エンドのユーザ端末に機能を追加することなく，かつエンドエンドで NAT 越え通信を実現する方式を提案する．

提案方式を FreeBSD 上に実装し, 動作検証および性能測定を行った．NTS サーバによる名前解決時のオーバヘッドおよびエンドノード間の通信のスループットを評価した結果，事実上問題ない性能を有することを確認した．

以降，2 章で提案技術と目的が同じである AVES について詳細に説明し，分析する．3 章 で本論文の提案技術を説明し，4 章 で実装について述べる．5 章 で提案方式の動作検証と性能評価の結果を示し，最後に 6 章 でまとめる．

第2章　既存技術(AVES)とその課題

本論文と同様の目的を持つ既存技術として AVES があるので，その詳細と課題について述べる．AVES ではインターネット上に AVES 対応 DNS サーバと waypoint と呼ばれる専用サーバを配置し，エンドノードは waypoint を経由して通信を行う．IN は通信を受けるに当たり DNS に自身の FQDN と IP アドレスに加え，NAT ルータの IP アドレスを関連づけて登録しておく．EN は AVES 対応 DNS サーバをプライマリ DNS として設定し，名前解決を依頼する必要がある．

図 2.1 に AVES の動作を示す．EN が DNS サーバに IN(*alice*) について問い合わせると，DNS サーバは waypoint に *alice* についてのルート確認情報を送信する．ルート確認情報には *alice* のプライベート IP アドレス “PA1”，NAT ルータのグローバル IP アドレス “GA2” および EN の IP アドレス “GA1” が含まれる．waypoint はこれを受理したら DNS に肯定応答を返す．DNS サーバが waypoint から受理メッセージを受け取ると，waypoint の IP アドレス”GA3”を EN に応答する．次に EN は waypoint に対して通信を開始する．waypoint はパケットの宛先アドレスを *alice* のプライベート IP アドレス PA1 に変換し，NAT ルータのグローバル IP アドレス GA2 で IP-in-IP カプセリングし，NAT ルータへ送信する．NAT は上記パケットを受信すると，カプセル化を解除し *alice* へ送信する．*alice* からの応答は直接 EN へ送信されるが，NAT ルータは送信元 IP アドレスを waypoint の GA3 に変更してから送信する．以後，同様にして三角経路での通信を行う．

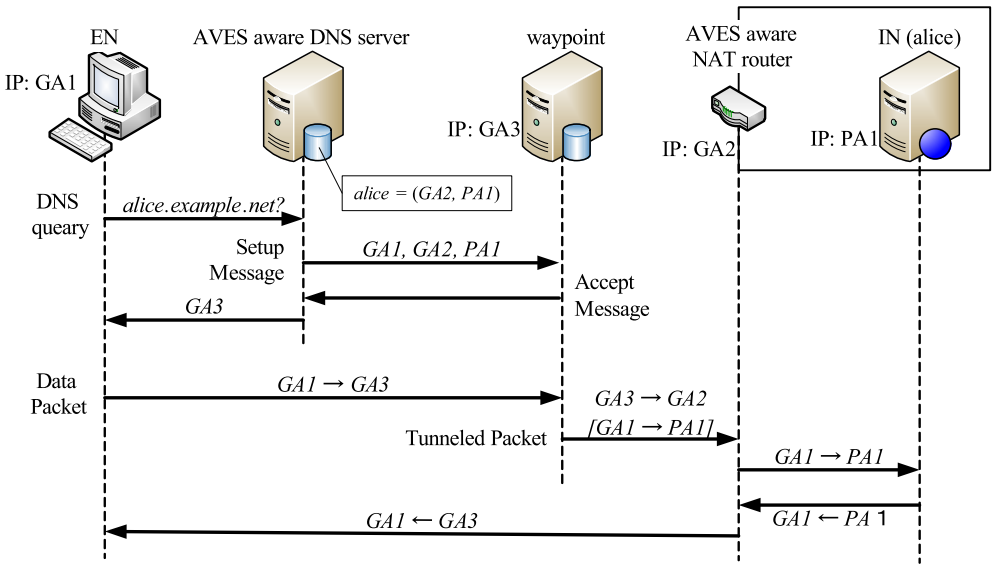


図 2.1　AVES の動作

AVES はユーザ端末には機能を実装をせずに NAT 越えを実現できるという利点があるが，第三の特殊な装置が必要で，かつ DNS を改造する必要がある．また経路が冗長になることや，IP-in-IP カプセリングによるパケット冗長が発生するなどの課題がある．更に，IN からの返信は NAT 処理時にパケットを本来の送信元とは違う waypoint から送信するため，イングレスフィルタリング [9] などのセキュリティが講じられていた場合，通信が届かない可能性がある．

第3章　提案方式

本提案方式を NTSS(NAT-Traversal Support system) と呼び，本方式で使用する改造した DNS サーバを NTS サーバ，改造した NAT ルータを NTS ルータ，実行するプロトコルを NTS プロトコルと呼ぶ．EN と IN には機能を実装する必要がなく，今後普及する情報家電やネットワークに幅広く対応することができる．IN は事前にサーバとの余計な通信を行う必要はなく，EN と IN 間の通信は一切余分な処理を行わないため，End-to-End 通信の利点を損なうことなく NAT 越え通信を実現することができる．

3.1　ネットワーク構成と事前設定

NTSS を実現するネットワーク構成を図 3.1 に示す．インターネットとプライベートネットワークの間に NTS ルータ，NTS ルータと協調する NTS サーバ，IN の名前解決を行う Dynamic DNS(以下 DDNS) サーバ [10] が必要である．DDNS サーバには NAT 越えのための機能は不要であり，既に運用されている DDNS サービスをそのまま利用できる．ここで EN，NTS ルータのグローバル IP アドレスをそれぞれ GA1，GA2 とし，IN(*alice*)，IN(*bob*) のプライベート IP アドレスをそれぞれ PA1，PA2 とする．*alice* および *bob* は IN のホスト名である．

事前設定として，EN のプライマリ DNS として NTS サーバを登録しておく．更に NTS ルータへ PHL(Private Host List) と呼ぶテーブルに以下の様に IN の FQDN(Fully Qualified

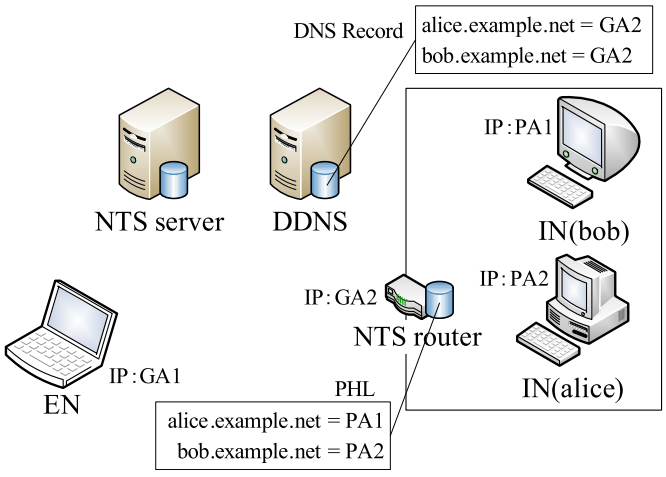


図 3.1　想定されるネットワーク構成

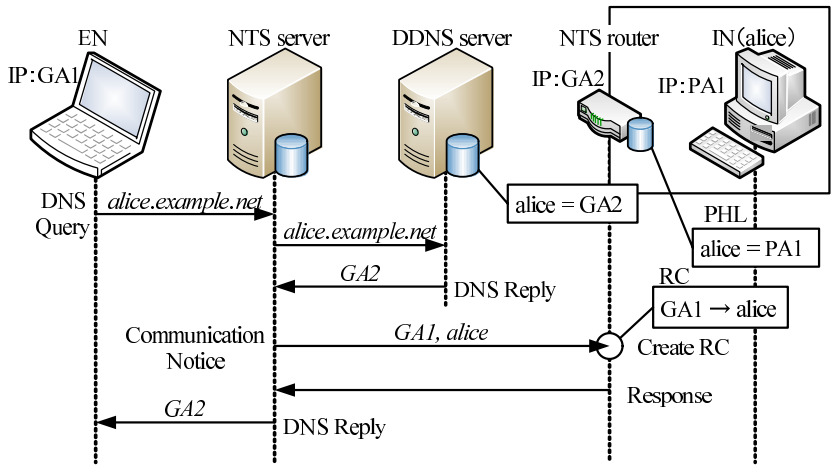


図 3.2　DNS 名前解決シーケンス

Domain Name) とプライベート IP アドレスを対応づけて登録しておく．

$$alice.example.net := PA1 \qquad bob.example.net := PA2$$

NTS ルータは PHL より IN の FQDN と NTS ルータのグローバル IP アドレスを DDNS サーバに登録する．ここで DDNS サーバを利用する理由は，一般の家庭ネットワークに割り当てられるグローバル IP アドレスは可変の場合が多いためである．使用するプライベートネットワークが固定でグローバル IP アドレスが割り当てられている場合はこの限りではない．

以降の動作説明では EN から IN(*alice*) へ通信を開始する場合の例を，名前解決時，通信開始時にわけ，それぞれ説明する．

3.2　DNS 名前解決

図 3.2 に EN から IN(alice) へ通信を開始する場合の名前解決シーケンスを示す．EN は IN(*alice*) と通信を開始するに当たり，alice.example.net の名前解決を NTS サーバへ依頼する．NTS サーバは DDNS サーバより NTS ルータの IP アドレス *GA*2 を取得する．実際には NTS サーバが DDNS サーバの IP アドレスを知るために DNS のしくみを利用するが，図 3.2 ではこのシーケンスは省略して記載されている．次に NTS サーバは *GA*1 から *alice* への接続依頼があることを NTS ルータ（*GA*2）に通知する．この通知を受け取った NTS ルータは PHL を参照し，*GA*1 から *PA*1 へ通信があるということを RC(Request Cache) へ記憶しておく．NTS ルータは NTS サーバへ通知応答を返す．最後に NTS サーバは EN に対して NTS ルータのアドレス *GA*2 を応答する．

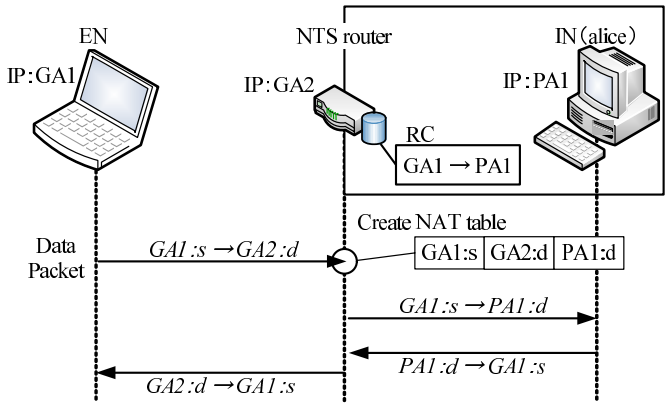


図 3.3　通信開始シーケンス

3.3　通信開始

EN は通信相手のアドレスがわかったので，*GA*2(NTS ルータ) に対して通信を開始する．図 3.3 に通信開始シーケンスを示す．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．

EN は取得した IP アドレス *GA*2 への通信を開始する．NTS ルータはパケットを受け取ると RC の内容を確認する．RC に該当するデータがあれば，受け取ったパケットと RC の情報から宛先/送信元 IP アドレスとポート番号，プロトコルタイプがわかるので，次のような NAT テーブルを動的に生成し，RC を削除する．

$$GA1 : s | GA2 : d | PA1 : d$$

この NAT テーブルの意味は，$GA1 : s$ との通信では NAT の $GA2 : d$ と IN の $PA1 : d$ が対応していることを意味する．つまり，$GA1 : s$ から $GA2 : d$ 宛に返信されたパケットは，NAT において宛先が $PA1 : d$ に変換されて *alice* に送信される．これに対する *alice* からの応答パケットは上記と逆の変換を行い EN へ送信される．

第4章　実装

提案システムの検証を行うため，プロトタイプシステムとして，NTS サーバモジュールを FreeBSD のアプリケーションに，NTS ルータモジュールを FreeBSD の NAT デーモンに実装した．

4.1　NTS サーバの実装

図 4.1 に NTS サーバの実装概要を示す．NTS サーバには DNS サーバ機能として BIND をインストールし，これを任意 (ここでは 10053 番) のポートでリッスンするように設定する．また，NTS サーバ処理モジュールは通常の DNS アプリケーションと同様に TCP/UDP の 53 番ポートでリッスンしておくことで DNS パケットを処理する．図における黒破線矢印は DNS に付随する通信，青矢印は NTS における通信を表し，lower layer の矢印の先は通信相手を表す．

NTS サーバはパケットを受信すると，通常の DNS 処理は BIND へ受け渡す．DNS リクエストパケットを受け取った BIND は通常の DNS 機能により名前解決を行い，NTS サーバモジュールへ DNS レスポンスパケットを返す．NTS サーバモジュールは上記 DNS レスポンスパケットより，IN の所属する NTS ルータのアドレスが分かるので，“FQDN に対応する IP アドレスへ EN から通信要求がある” ことを通知する．この際，NTS ルータモジュールは DNS パケットのトランザクション ID より DNS パケットやネゴシエーションを管理する．ネゴシエーション後，NTS サーバは EN（DNS 名前解決依頼元）に対して DNS レスポンスパケットを返信する．上記手順により，NTS サーバはあたかも通常の DNS サーバの様に振る舞う．

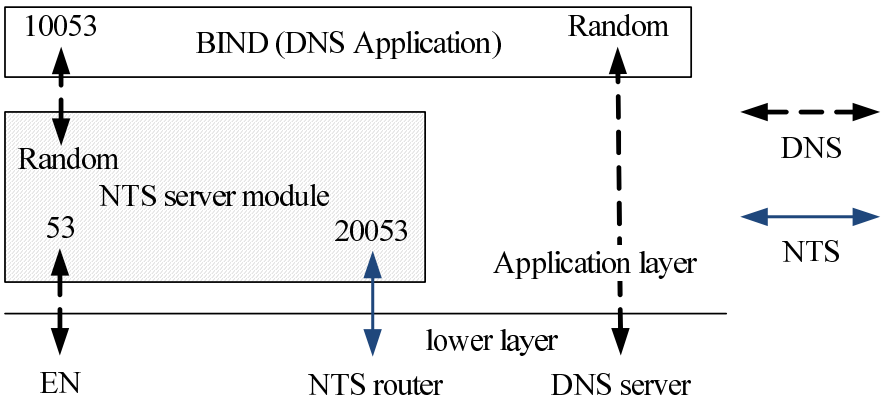


図 4.1　NTS サーバの実装概要

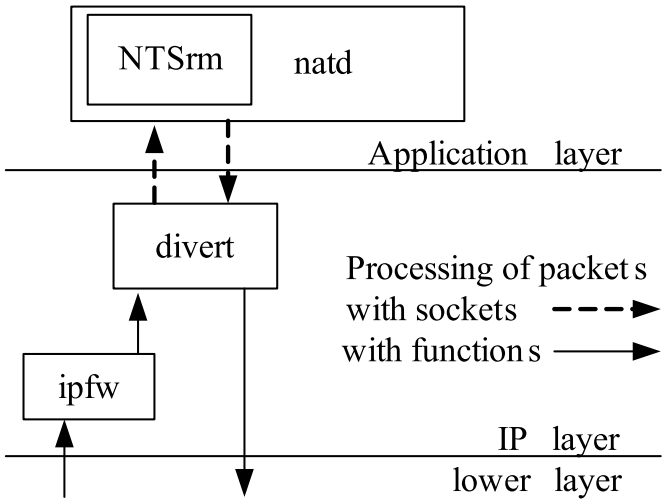


図 4.2　NTS ルータの実装概要

4.2　NTS ルータの実装

図 4.2 に NTS ルータの実装概要を示す．NTS ルータを実現するにあたり，NTS router モジュールを NAT デーモン natd[1]内に実装した．ipfw はファイアウォールの動作を行うモジュールである．divert は natd のパケット取り出しをサポートするソケットである．

FreeBSD では通常の NAT として動作する時は以下の様になる．パケットを受信すると下位層から IP 層の ipfw に渡され，ipfw はそのパケットを divert に渡す．natd は divert からソケットを介してパケットを取り出し，テーブル生成やパケットのアドレス変換など NAT に関わる動作を行った後に，ソケットを介して divert に戻す．divert はパケットを下位層に渡し，アドレス変換されたパケットが送信される．NTS ルータの実装では，natd 内に NTS router モジュールを組み込み，natd が受け取ったパケットを常時監視し，NTS サーバとのネゴシエーションや，受信パケットの変換処理等の機能を実現した．このようにすることで natd が持つ NAT としての様々な機能をそのまま利用できるようにした．通常の NAT 変換では FTP や SIP(Session Initiation Protocol) [11] のように，パケットのペイロード部分に通信を行う IP アドレスやポート番号などの制御情報が書かれていた場合，その通りに通信を行うことは不可能である．そこで，この方法で実装することにより現行の natd では標準となっている，ペイロード部分まで監視して NAT 処理を行うことで上記問題を解決する技術 ALG(Application Level Gateway) をそのまま利用することができる．

4.3　NAT テーブル生成方法

NTS ルータにおいて，natd 機能を流用するにあたり，以下のような工夫をした．図 4.3 に NAT テーブルの生成方法を示す．

[1]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．

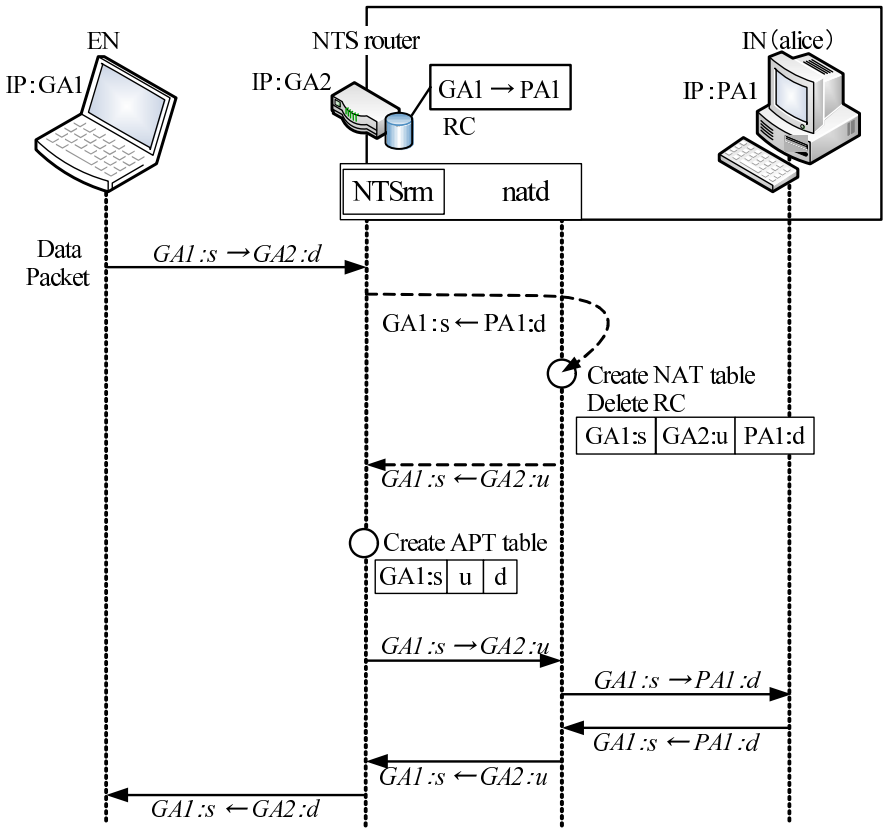


図 4.3　NAT テーブル生成方法

EN は DNS 名前解決により得たアドレスへ向けて最初のパケット “GA1:s → GA2:d” を NTS ルータへ送信する．NTS ルータは受け取ったそのパケットに該当する NAT テーブルがない場合，RC を参照する．その受信パケットは名前解決時に生成された RC に該当するので，RC の内容と受信パケットの内容から “PA1:d → GA1:s” のような擬似パケットを作成し，natd の処理に渡す．すると natd は IN から EN へ送信されるパケットを受信したものと判断して，下記のような NAT テーブルを生成する．

$$GA1 : s|GA2 : u|PA1 : d$$

この NAT テーブルの意味は，$GA1 : s$ との通信では $GA2 : u$ と $PA1 : d$ が対応していることを意味する．ここで u は NAT 内の空ポートの中から割り当てられたポート番号である．しかし EN からのパケットは $GA2 : d$ であり，ポート番号が一致しない．そこで NTS router モジュールにおいて次のような APT（Address Port Translate）テーブルを生成し，更にポート変換を行う．

$$GA1 : s|u|d$$

すなわち，“$GA1 : s \rightarrow GA2 : d$” パケットは APT テーブにより “$GA1 : s \rightarrow GA2 : u$” に変換してから natd へ渡す．先ほど生成された NAT テーブルにより “$GA1 : s \rightarrow PA1 : d$” に変換されて *alice* へパケットを送信される．逆方向の通信でも同様に，NAT 処理後に APT テーブルで変換することにより EN へ通信を行うことができる．つまり NTS ルータでは natd

で生成された NAT テーブルと APT テーブルを合わせることにより，必要な NAT テーブルを実現する．

NAT テーブルの生存時間は，UDP が 300 秒，コネクション確立後の TCP が 86400 秒 (24 時間) であり，APT テーブルにも同様の値を適用する．

第5章　動作検証と評価

図 3.1 のシステム構成において，EN と IN が通信を行う場合の動作検証と性能測定を行った．性能測定に使用した各装置の仕様は CPU が Pentium4 3.0GHz，メモリが 512MB である．またネットワーク環境は 100BASE-TX の Ethernet であり，EN，NTS ルータ，NTS サーバ，DNS サーバをスイッチで接続した．

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

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をネットワークアナライザ Ethereal を用いて測定した．次に，NTS ルータにおける APT テーブルの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 [12] 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，NTS サーバの代わりに通常の DNS サーバを用い，通常の natd で IN 側から EN 側へ通信を行った場合も測定した．DNS の名前解決はシーケンスに差がないように，NTS サーバ，DNS サーバともに IN の A レコードを持たせた．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．

図 5.1 に通信開始時のオーバヘッドを示す．EN が DNS クエリを送信してから，NTS ネゴシエーションを経て DNS の応答を得るまでの時間は 841.8μs であった．このうち，NTS サーバが NTS ルータとのネゴシエーションにかかった時間が 265.2μs を占めていた．また，通常の DNS サーバによる名前解決に掛かる時間は 336.0μs であった．提案方式における通信には EN 側では処理などを行うことはないので，提案方式による純粋なオーバヘッドは 505.8μs[1] となり，提案方式は事実上，通信開始に影響を与えないことがわかる．

表 5.1 に Netperf によるスループット測定値を示す．NTS 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

[1] 841.8-336.0≒505.8μs

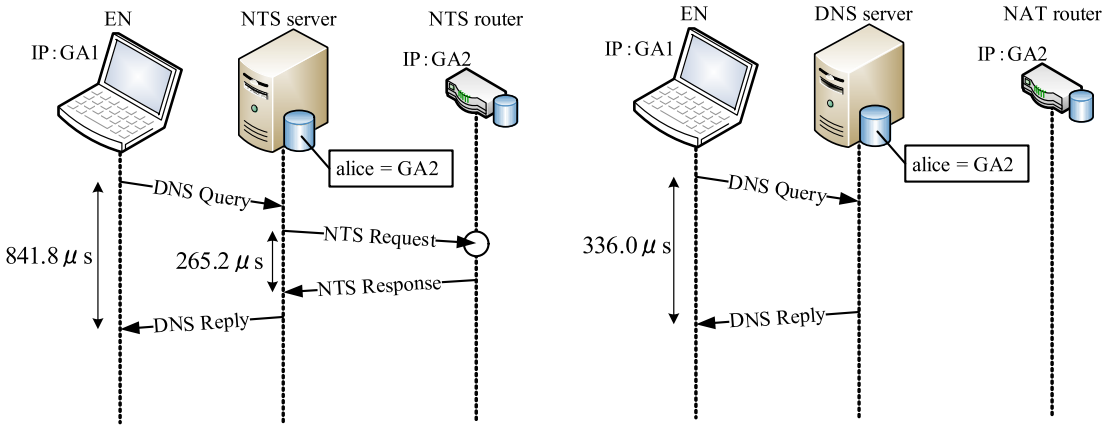


図 5.1　Ethreal による通信開始時のオーバヘッド測定値

表 5.1　Netperf によるスループット測定値

Message Size (Bytes)	TCP Stream (Mbps)		UDP Stream (Mbps)	
	NTS	NAT	NTS	NAT
64	94.1	94.1	49.3	49.3
128	94.1	94.1	66.0	66.0
256	94.1	94.1	79.6	79.6
512	94.1	94.1	88.9	88.9
1024	94.1	94.1	94.4	96.4
1472	94.1	94.1	96.4	96.4

NTS：提案方式による NAT 越え通信 (EN→IN)
NAT：通常の通信 (EN←IN)

められなかった．NTS では NAT のテーブル変換が APT で一回増えるだけであり，カプセリング等を行う方式より高スループットを得られることが実証できた．

5.3　セキュリティに関する考察

プライベートネットワークは NAT により内部 IP アドレスが隠蔽されていたため，外部から特定の IN を標的とした攻撃から防ぐという効果もあった．故に企業ネットワークでは簡易的なセキュリティ対策の為に NAT を利用することもある．そのため，NAT 越え技術により IN のセキュリティは脅威にさらされる可能性がある．提案方式は IN が外部からの通信を許容する場合，DNS に名前登録をしている．これは自分の IP アドレスを公開しているのと同意であり，IN がグローバル IP アドレスを取得した場合と同様の状況になる．また，PHL によりアクセス制御も行っているため，アクセスが許可されていない IN に対して，NTS ルータが外部からの指示でマッピングされることはない．IN の外部公開を辞める場合は，その IN に関する PHL を削除すれば，通常の NAT 配下にある端末と同じ状態になる．その他，NTS ルータは実装上 natd をそのまま利用することができるの

で，通常の NAT 同様 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．更に NTS ルータ管理者は特定の NTS サーバからの通知のみ許可するように設定しておくことで，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．

第6章　まとめ

本論文ではユーザ端末の改造が不要な NAT 越えを実現する方式を提案した．提案方式では EN の通信開始に先駆けて，NTS サーバと NTS ルータが協調することにより NTS ルータが動的に NAT テーブルを生成することで NAT 越え通信を可能にする．各端末間の通信はエンドエンドで行うことができ，今後のユビキタス社会に有益なシステムと考えられる．

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，性能測定により提案方式によるオーバヘッドは無視できることを示した．

今後は更なる検討を行う．

謝辞

本研究にあたり，多大なる御指導と御教授を賜りました，渡邊 晃 教授には心から感謝いたします．

本論文を作成するにあたり，快く査読を引き受けて下さり，熱心にご指摘を頂きました，高橋 友一 教授に感謝の意を表します．

本論文を作成するにあたり，快く査読を引き受けて下さり，熱心にご指摘を頂きました，宇佐見 庄五 准教授に感謝の意を表します．

また，本研究を進めるにあたり，常日頃からの御意見ならびに御助言を受け賜りました，博士後期課程 鈴木 秀和 氏に深謝いたします．

最後に，本研究を進めるにあたり，数々の有益な御助言や御討論を賜りました，渡邊研究室の諸氏に感謝します．

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] Nishitani, T. and Miyakawa, S.: Carrier Grade Network Address Translator (NAT) Behavioral Requirements for Unicast UDP, TCP and ICMP, Internet-draft, IETF (2008). draft-nishitani-cgn-00.txt.

[4] 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).

[5] 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.

[6] 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).

[7] P.Mockapetris: DOMAIN NAMES - CONCEPTS AND FACILITIES, RFC 1034 (1987).

[8] P.Mockapetris: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION, RFC 1035 (1987).

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

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

[11] 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).

[12] Jones, R.: Netperf: a network performance monitoring tool. http://www.netperf.org/netperf/NetperfPage.h

[13] Rosenberg, J., Mahy, R. and Wing, D.: Session Traversal Utilities for NAT (STUN), RFC 5389, IETF (2008).

[14] Rosenberg, J., Mahy, R. and Huitema, C.: Traversal Using Relay NAT (TURN), Internet-draft, IETF (2005). draft-rosenberg-midcom-turn-08.

[15] UPnP Forum: *Internet Gateway Device (IGD) Standardized Device Control Protocol V 1.0* (2001). http://www.upnp.org/standardizeddcps/igd.asp.

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

[17] Cheshire, S., Krochmal, M. and Sekar, K.: NAT Port Mapping Protocol (NAT-PMP), Internet-draft, IETF (2006). draft-cheshire-nat-pmp-02.txt.

[18] Kondo, K.: Capsulated Network Address Translation with Sub-Address (C-NATS), Internet-draft, IETF (2003). draft-kuniaki-capsulated-nats-05.txt.

研究業績

学術論文

なし

国際会議

1. Miyazaki Yutaka, Suzuki Hidekazu and Watanabe Akira, “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), Oct.2007.

2. Miyazaki Yutaka, Suzuki Hidekazu and Watanabe Akira, “Proposal of a NAT traversal system independent of user terminals and its implementation,” Proceedings of the IEEE International Region 10 Conference 2008 (TENCON2008), Nov.2008.

国内会議

1. 宮崎 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, Sep.2006.

2. 宮崎 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, Mar. 2007.

研究会・大会等

1. 宮崎 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.

2. 宮崎 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.

展示会・その他

1. 鈴木秀和, 宮崎 悠, 細尾幸宏, 渡邊晃, 2008 年 9 月 16 日から 18 日に東京国際フォーラムで行われた “イノベーション・ジャパン 2008-大学見本市” にて同研究室から出展した “MobilePPC および NAT-f” の展示において説明員として参加.

付 録A　NATの動作と種類

NAT には，IP アドレスのみを変換する NAT（Network Address Translator）と IP アドレス変換に加え，ポート番号変換も行う NAPT（Network Address Port Translator）がある．NAT はグローバル IP アドレスとプライベート IP アドレスを対応づけるだけなので，複数のプライベートアドレス空間の端末が，同時にグローバルアドレス空間上の端末と通信ができるのは NAT の保持するグローバル IP アドレスの数だけに制限される．一方，NAPT はポート番号を用いて通信の判別を行うため， NAPT に 1 つだけグローバル IP アドレスを割り当てれば，複数のプライベートアドレス空間の端末がグローバルアドレス空間の端末と同時に接続できる．NAPT は NAT より汎用性が高いので多く使われているが，NAPT は広義の NAT に含まれるため，以後 NAPT を含めて NAT と呼ぶ．ただし，本稿における NAT の動作説明は全て NAPT のそれを指すものとする．

A.1　NAT の動作

NAT の動作説明の例として，クライアント端末から異なるアドレス空間に存在する WEB サーバへの HTTP 通信を挙げる．NAT router は NAT 機能が搭載された装置である．PA はプライベート IP アドレス，GA はグローバル IP アドレスを示す．

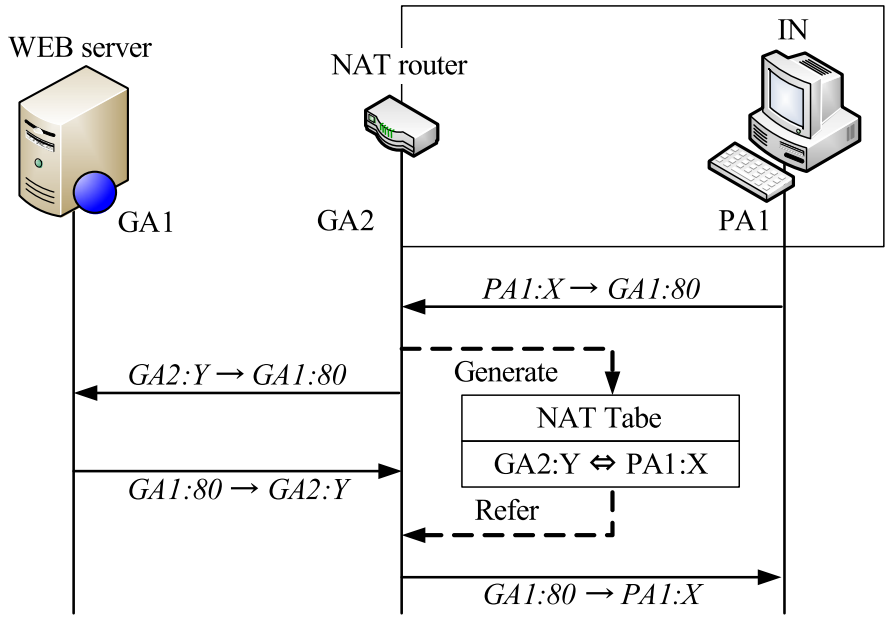


図 A.1　NAT の動作（内 → 外）

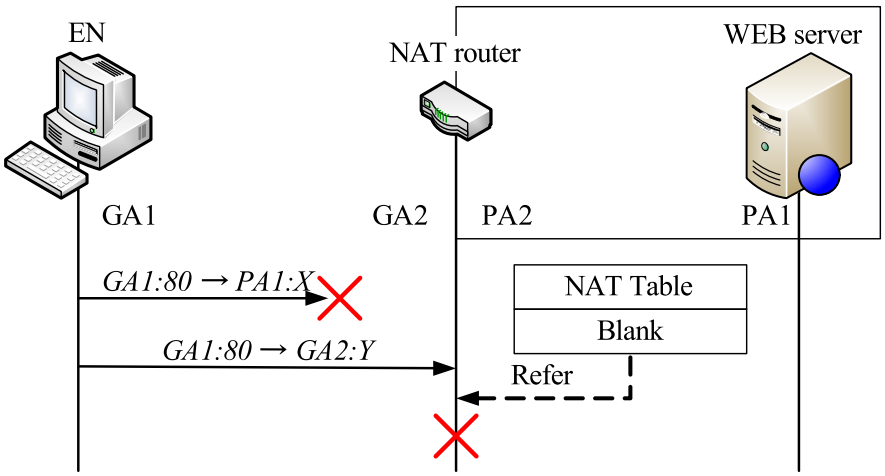


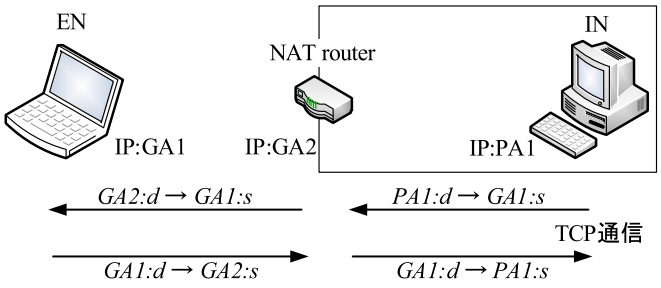
図 A.2　NAT の動作（外 → 内）

まずプライベートアドレス空間に存在する端末からグローバルアドレス空間に存在する WEB サーバに通信を開始する場合の NAT の動作を図 A.1 に示す．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．はじめに EN は宛先を IP アドレス GA1，ポート番号を 80，送信元を IP アドレス PA1，ポート番号を X として送信する．X はクライアントの OS が動的に選んだ任意のポート番号である．NAT router では送信元を NAT router の IP アドレス GA2，ポート番号 Y へと変換して中継する．Y は NAT router が動的に選んだ任意のポート番号である．このとき NAT router はこの変換の関係を記した NAT テーブルを生成する．このパケットを受信した WEB サーバは，応答パケットを宛先 IP アドレス GA2，宛先ポート番号 Y，送信元 IP アドレス GA1，送信元ポート番号 80 として返信する．このパケットは NAT router が受信し，NAT テーブルに従って宛先を IP アドレス PA1，ポート番号 X に書き換えて中継し (4)， クライアントがこれを受信する．以後の通信は NAT テーブルに従って，NAT router がアドレス変換を行うことにより，通信が行われる．

次に，グローバルアドレス空間に存在する端末がプライベートアドレス空間に所属する WEB サーバへ HTTP 通信を開始する場合の NAT の動作を図 A.2 に示す．まず WEB サーバはプライベート IP アドレスであるため，グローバルアドレス空間から見ると無効な値であり，インターネット上に送信ができない (1)．また，仮に NAT router のグローバル IP アドレスを知ることができて，NAT router までパケットを送信できたとしても，NAT router には，まだ NAT テーブルが存在しないため NAT router はどこにパケットを中継すれば良いのか判断できないため，破棄される (2)．即ちプライベートアドレス空間にサーバ，異なるアドレス空間にクライアントが存在するシステムは一般的に構築できない．ただし，NAT で静的にあらかじめ NAT テーブルを手動で記述しておくポートフォワーディングと呼ぶ機能を利用すればこの限りではない．しかしこの方法では，1 つの



NAT Table						
Remote		Global		Local		Protocol
address	port	address	port	address	port	
		GA2	s	PA1	s	TCP

図 A.3　Full Cone NAT

ポートに対して 1 台しか設定できないことや動的に変更が不可能なため柔軟性に欠けるなどの欠点がある．

A.2　NAT の種類

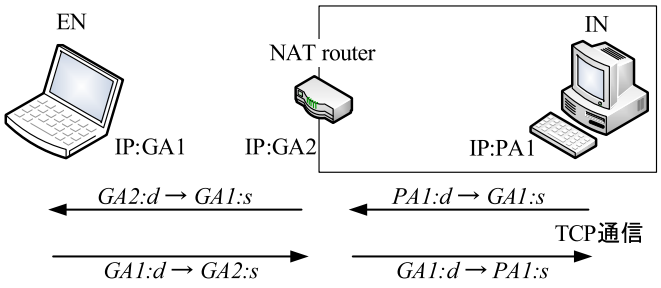
NAT には NAT 処理の変換方法から “Cone NAT”，“Redistricted cone NAT”，“Port Redistricted cone NAT” と “Symmetric NAT” に分けられる．次の節でそれぞれの詳細な違いを説明する．

A.2.1　Full Cone NAT

Full Cone NAT の NAT テーブル作成法を図 A.3 に示す．Full Cone NAT は，NAT ルータの自身のポートが，どの IN のどのポートと対応しているかだけを保持する．その為，そのテーブルを保持している間は，NAT のそのポートへ通信を行えば，第三者でも IN へ通信を行うことができる．また，その際の NAT ルータの外部ポートは IN の送信元ポートと同じ値が割り当てられる．しかし，NAT ルータ内に複数の IN が存在し，同じポートからパケットを送信しようとした場合，二台目の IN が割り当てられる NAT ルータの外部ポートは違う値になる．その場合の NAT テーブルは後に示す Symmetric NAT と同様であり，Redistricted cone NAT と Port Redistricted cone NAT も同様な動作で処理する．

A.2.2　Redistricted Cone NAT

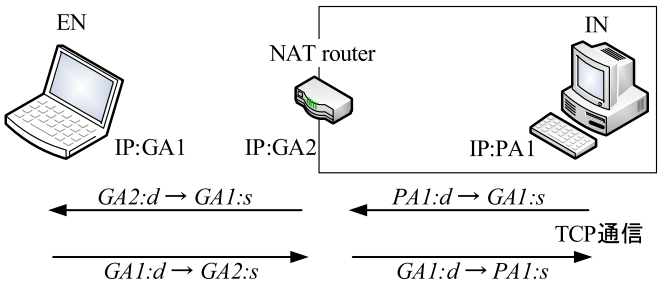
Redistricted Cone NAT の NAT テーブル作成法を図 A.4 に示す．Redistricted Cone NAT では，Full Cone NAT に加えて EN のアドレスも対応させて覚えておく．それにより，NAT テーブルは生成された EN-IN 間のみで使用されることになり，Full Cone NAT よりセキュリティを高めることができる．



NAT Table

Remote		*Global*		*Local*		*Protocol*
address	*port*	*address*	*port*	*address*	*port*	
GA1		GA2	s	PA1	s	TCP

図 A.4　Redistricted Cone NAT



NAT Table

Remote		*Global*		*Local*		*Protocol*
address	*port*	*address*	*port*	*address*	*port*	
GA1	d	GA2	s	PA1	s	TCP

図 A.5　Port Redistricted Cone NAT

A.2.3　Port Redistricted Cone NAT

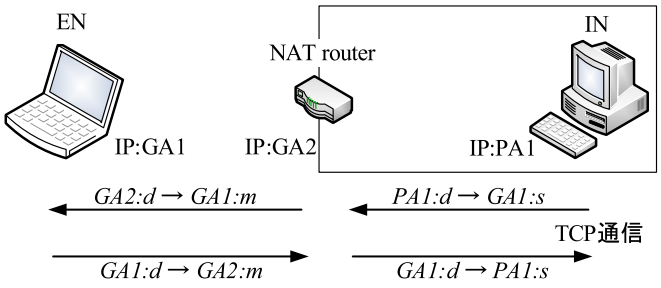
Port Redistricted Cone NAT の NAT テーブル作成法を図 A.5 に示す．Port Redistricted Cone NAT では，Redistricted Cone NAT に加え更にポート番号も対応させて保持する．おれにより，EN が NAT 等を使用してアドレスを共有している場合でもその端末の通信のみを通すことができ，更にセキュリティを高めることができる．

A.2.4　Symmetric NAT

Symmetric NAT のテーブル作成法を図 A.6 に示す．Symmetric NAt では，NAT ルータの外部ポートに IN の送信元ポートと違う番号が割り当てられる．

A.3　NAT Table の所持時間

NAT の使用できるポート番号は限られているため，これを有効に使用するために，生成された NAT Table には通信プロトコルに応じて TTL (Time To Live) が設定されている．



NAT Table						
Remote		Global		Local		Protocol
address	port	address	port	address	port	
GA1	d	GA2	m	PA1	s	TCP

図 A.6　Symmetric NAT

表 A.1　各通信プロトコルにおける NAT Tabel の TTL

Protocol	TCP	UDP	ICMP	DNS
TTL(sec)	86400	300	86400	60

各通信プロトコルにおける，NAT Table の TTL を表 A.1 に示す．また，TTL を設定し，NAT テーブルを定期的にリセットすることで不用意な通信を遮断することもできる．

A.4　Carrier Grade NAT

IPv4 アドレスは 32bits の 0 と 1 の羅列であり，使用できる数に上限があり，近年は急速なインターネットの普及により，IPv4 アドレスの枯渇が懸念されている．そこで，IP 層を拡張することにより，IP アドレスを 128bits にする IPv6 の検討がなされている．しかし，IP 層に変更を加えるため，通信を行う全ての端末に実装しなければならないことから，実現が遅れている．

その対応策として CGN(Carrier Grade NAT) が検討されている．CGN とは，その名の通りキャリア，つまりプロバイダレベルで NAT を使用する技術である．図 A.7 に CGN の構成例を示す．通常，プロバイダのネットワークはグローバル IP アドレスにより構築される．CGN の場合は，プロバイダのネットワークを NAT を使用してインターネットに接続し，プライベート IP アドレスにより構築することにより，IPv4 アドレスを節約する．CGN はプロバイダがインターネットに接続する機器に NAT 機能を入れ，会員へはプライベート IP アドレスを割り当てるだけなので，比較的容易に実現することができる．しかし，ポート番号は有限であり[1]，一つのグローバル IP アドレスをより多数で使用すると，1 ユーザあたりのポート数が制限されるという課題がある．また，多段 NAT 構成に

[1]WELL KNOWN PORT NUMBERS:0-1023, REGISTERED PORT NUMBERS:1024-49151, DYNAMIC PRIVATE PORTS:49152-65535

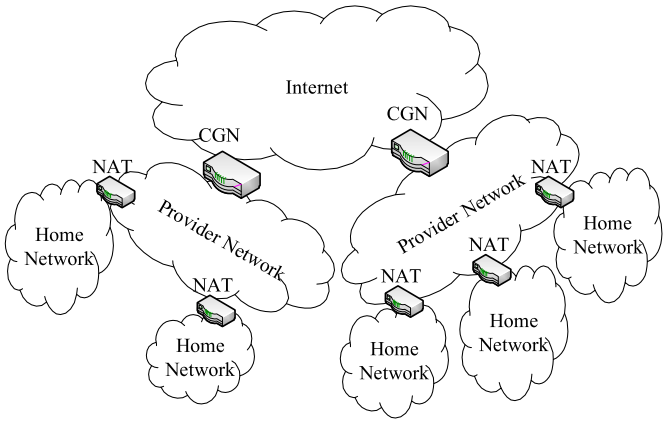


図 A.7　CGN の構成例

より UPnP や STUN など，いくつかの NAT 越え技術は利用できなくなる.

付 録B　その他のNAT越え関連研究

B.1　アプリケーションレベルの解決方法

アプリケーションレベルの解決手法として STUN (Session Traversal Utilities for NAT) [13] や TURN（Traversal Using Relay NAT）[14]，UPnP（Universal Plug and Play）[15] 等がある．

B.1.1　STUN

STUN は Hole punching という技術を利用した方式である．Hole punching [16] とは IN が予め NAT の外部へ通信を行い，それにより NAT へ穴を開けるかのように NAT テーブルを生成しておく方法である．STUN ではその NAT の情報を得て通信を行うことで NAT 越え通信を可能とする技術である．

ここで STUN の実際のシーケンスを図 B.1 へ示す．通信を受けたい IN は STUN サーバへ情報登録を行う．そこへは IN にマッピングされた NAT テーブルの情報を登録する．EN は STUN アプリケーションにより IN の情報を取得する．そして，EN は送信するパケットをその情報に合うように生成して送信することで，NAT ルータは IN からの通信に対する返信であるかのようにそのパケットを IN へ転送することができる．

この方式は Cone NAT[1] に対応しており，既に実用化されている．しかし，通信に利用するアプリケーションが STUN に対応している必要があり，STUN は IN が NAT テーブ

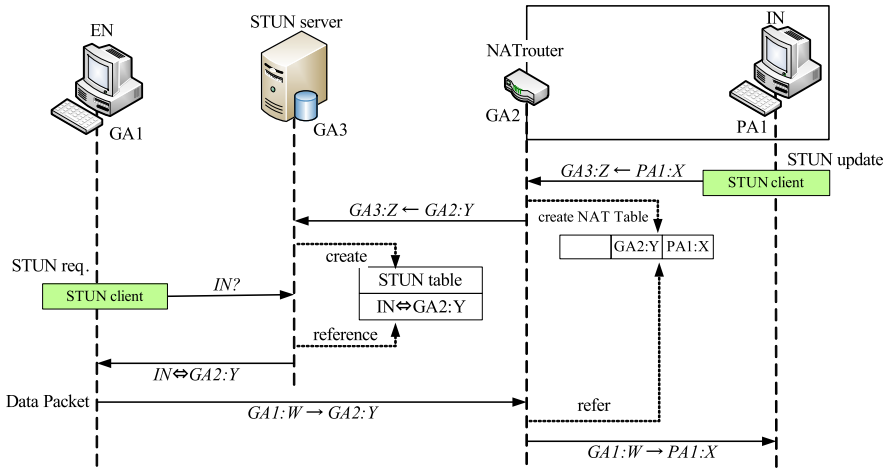


図 B.1　STUN の動作

[1]IN の送信元ポート番号から NAT に割り当てられるポート番号が変化しない型式．

ルに載るであろう情報を登録するため，Symmetric NAT[2] に対応できないなどの課題がある．Symmetric NAT に対する研究もされてはいるが，確実ではなく，以前課題が残っている．また，A.3 節で記述した通り，NAT のマッピングは timeout があるため，定期的に Keep Alive しておく必要がある．

B.1.2 TURN

TURN は IETF（Internet Engineering Task Force）では STUN の追加機構として定義されており，専用のサーバ (TURN サーバ) を中継することで 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 TURN サーバを経由するため，サーバにネットワーク負荷や処理負荷が集中し，スループットも低下する．また，ネットワーク上では EN の通信相手は TURN サーバになっていることや，経路が冗長になることなどから，今後さらに普及する Peer-to-Peer 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．

B.1.3 UPnP

UPnP はネットワーク版のプラグアンドプレイであり，パソコンや周辺機器，家電製品などをネットワークを通じて接続し，相互に機能を提供しあうための技術仕様である．その一部として，IN が実装された NAT にポートフォワーディングを動的に行うことができる．NAT 越え通信を行うには，IN が NAT に設定された NAT テーブルの情報を取得し，アプリケーションサーバとして用意された専用サーバへ通知する．EN は IN に関する NAT テーブルの情報を取得することで，その情報に対応したパケットを送信することで NAT 越え通信を実現する．

UPnP は IN と NAT ルータへアプリケーションを導入しなければならないが，デファクトスタンダードとして多くの NAT ルータに導入されているため，比較的容易に使用することが出来る．しかし，多段 NAT には対応できないため，CGN 等の技術が普及すると利用できなくなるという課題がある．

類似技術として NAT-PMP（NAT Port Mapping Protocol）[17] がある．

B.2 ネットワークレベルの解決方法

ネットワークレベルの解決方法として 4+4 [4] や NAT-f (NAT-free protocol) [5]，NATS（NAT with Sub-Address）[18] 等がある．

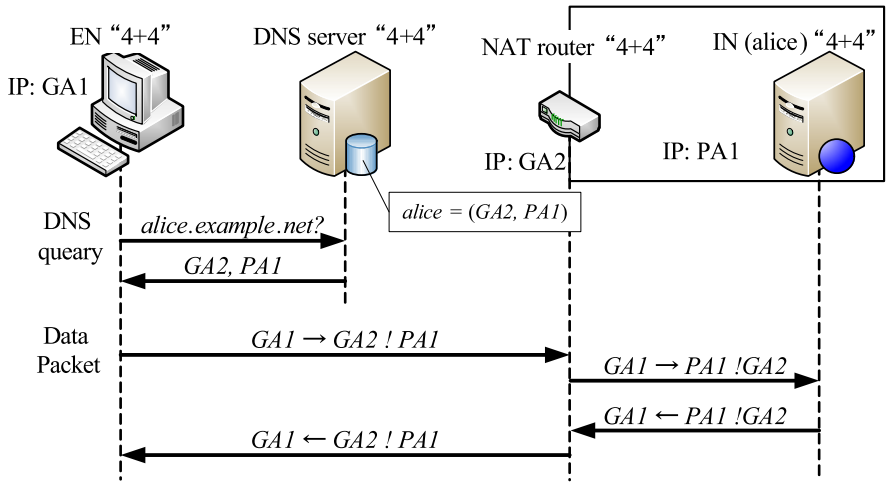


図 B.2　4+4 の動作

B.2.1　4+4

4+4 では 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．図 B.2 に 4+4 を利用した NAT 越え通信のシーケンスを示す．4+4 を導入するにあたり，EN，IN，NAT ルータの他に使用する DNS サーバにも実装が必要である．実装した DNS サーバには NAT のグローバル IP アドレスと *alice* のプライベート IP アドレスを登録しておく．EN は DNS より *alice* の登録情報を得ると，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 アドレスを記載して送信する．NAT は EN からのパケットを受信すると，NAT 処理を行わず，プライベート IP アドレスとグローバル IP アドレスを入れ替えて転送することにより IN への通信を可能とする．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り，プライベート IP アドレスも登録/通知できるように DNS も変更する必要があるため，実用難易度に課題がある．

B.2.2　NAT-f

NAT-f では EN と NAT に機能を実装し，EN からの通信開始時に動的に NAT のマッピングを行うことにより NAT 越え通信を実現する．図 B.3 に NAT-f を利用した NAT 越え通信のシーケンスを示す．事前設定として，EN と NAT ルータに NAT-f を実装する．NAT-f ルータには ACT(Access Control Tabel) として IN のホスト名，プライベート IP アドレスと通信の可否を登録する．

EN は通信を開始するにあたり，*alice* の名前解決を行う．EN が実際に送信するパケットをトリガに，そのパケットを待避してから通信に必要な情報を NAT-f ルータと交換し，強制的に NAT テーブルを生成する．その情報には EN のグローバル IP アドレスとポート，仮想アドレスという IN を識別するためのアドレスとポート，通信プロトコル，ホス

[2]IN の送信元ポート番号から NAT に割り当てられるポート番号が変化する型式

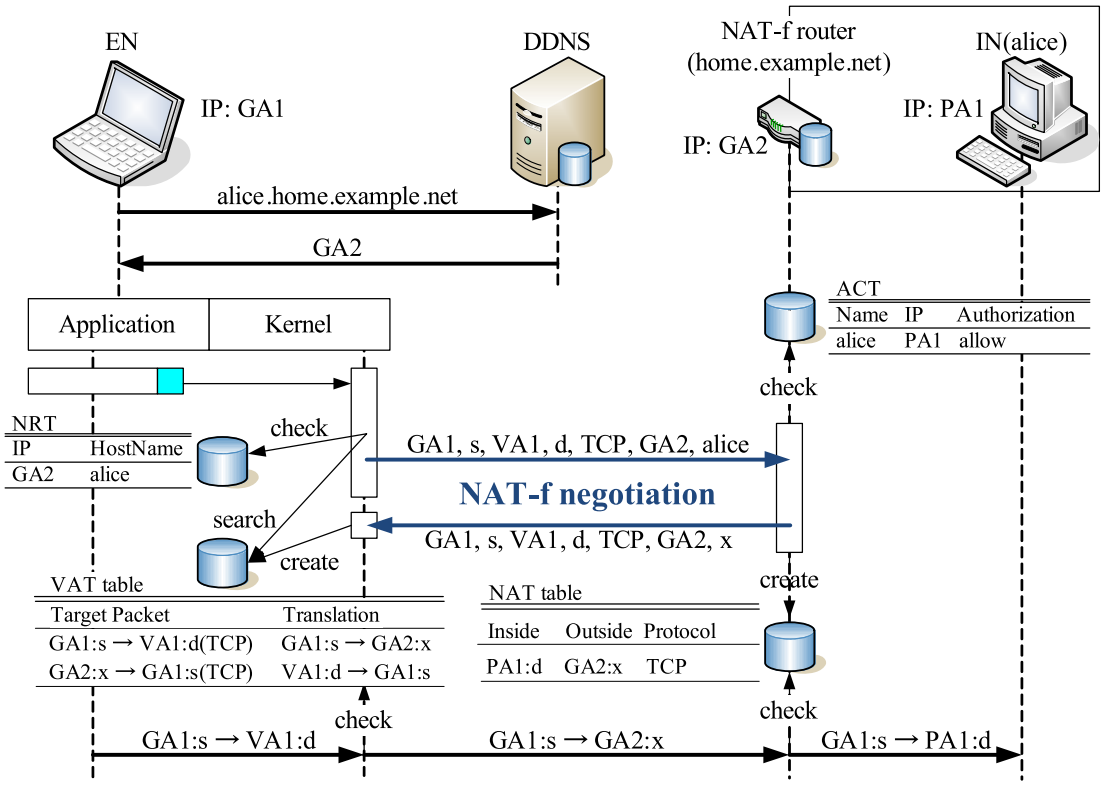


図 B.3　NAT-f の動作

ト名，NAT に割り当てられたポートが含まれる．その後，EN は待避していたパケットを復帰し，カーネルで NAT テーブルに合わせて変換して送信することで，NAT 越え通信を実現する．NAT-f は専用のサーバが不要であるが，EN のカーネルに実装が必要であり，一般ユーザに適用することは困難であり，導入に課題がある．

B.2.3　NATS

NATS は独自のサブアドレスを定義し，拡張した IP ヘッダにサブアドレスを記載して通信を行うことで，IP-in-IP トンネリングのように通信を行う技術である．EN と DNS サーバ，NAT ルータに実装が必要である．図 B.4 に NATS のシーケンスを示す．事前設定として NATS 対応 DNS サーバには *alice* の FQDN と対応させて NATS ルータの IP アドレスと同時に *alice* のサブアドレスを登録する．この DNS 登録は NATS ルータが自動に登録することで，IN は NATS 対応の実装を行う必要はないとしている．また，NATS ルータにはそのサブアドレスと対応するプライベート IP アドレス登録する．

alice と通信を行いたい EN は NATS 対応 DNS サーバに *alice* の名前解決を行う．NATS 対応 DNS サーバは登録されているグローバルアドレス *GA2* とサブアドレス *SA1* を返す．EN は NATS ルータへの通信において，拡張 IP ヘッダにサブアドレス *SA1* を入れ，GA2 宛のパケットでカプセリングして通信を行う．これを受け取った NATS ルータは，拡張 IP ヘッダの *SA1* から *PA1* 宛の通常の IP ヘッダに変換して転送する．これにより，IN へ

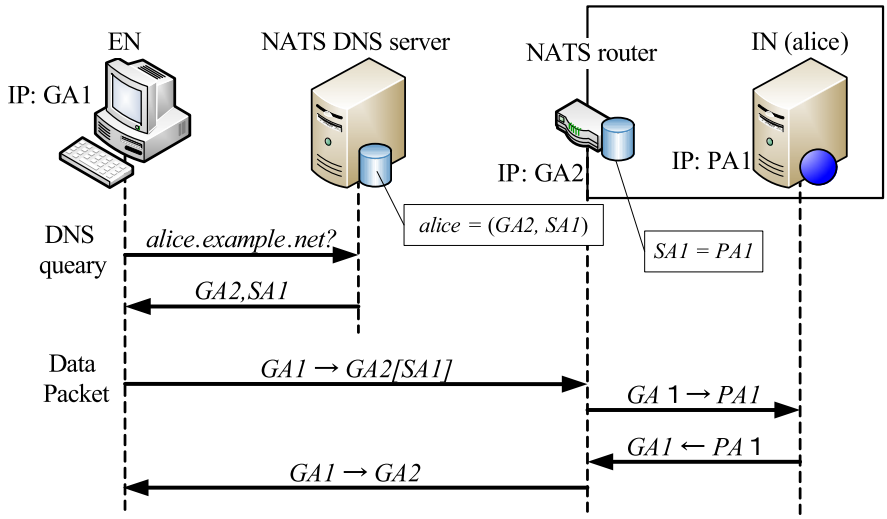


図 B.4　NATS シーケンス

実装せずに NAT 越え通信を可能としている．

しかし，EN はカーネルを改造しないといけないほか，DNS サーバを改造し，NATS ルータには NAT とは別の処理を定義しているため，導入が難しいという課題がある．また，EN と NATS ルータ間の通信はトンネリングにより通信を行うため，スループットが低下するなどの課題もある．

付 録C　Session Initiation Protocol

SIP(Session Initiation Protocol) [11] はその名の通り，二つ以上のクライアント間でセッションを制御するためのプロトコルである．SIP は，端末（UA：User Agent）間でセッションの生成，変更，切断を行うのみのプロトコルで，セッション上で交換されるデータそのものについては定めていない．従って，アプリケーションが SIP によって制御されたセッション上で，音声のやり取りを行えば IP 電話，音声と映像ならばテレビ会議，テキストメッセージならばインスタントメッセンジャーというように，幅広い応用が可能である．

次に，例として *alice* が *bob* へ IP 電話をかける場合のシーケンスを図 C.1 に示す．*alice* と *bob* は UA であり，各 UA の所属する SIP プロキシサーバがインターネット上に存在する．SIP プロキシサーバは，公衆電話網で言う交換機のようなものであり，UA やプロキシからのリクエストを受け取り，適切な UA/プロキシへ送信を行う．SIP における UA の識別は URI(Uniform Resource Identifier) を使用する．実際はメールアドレスの様な形式であるが，図 C.1 では “alice”“bob” に相当する．各 SIP プロキシには *alice* や *bob* の URI が登録されている．

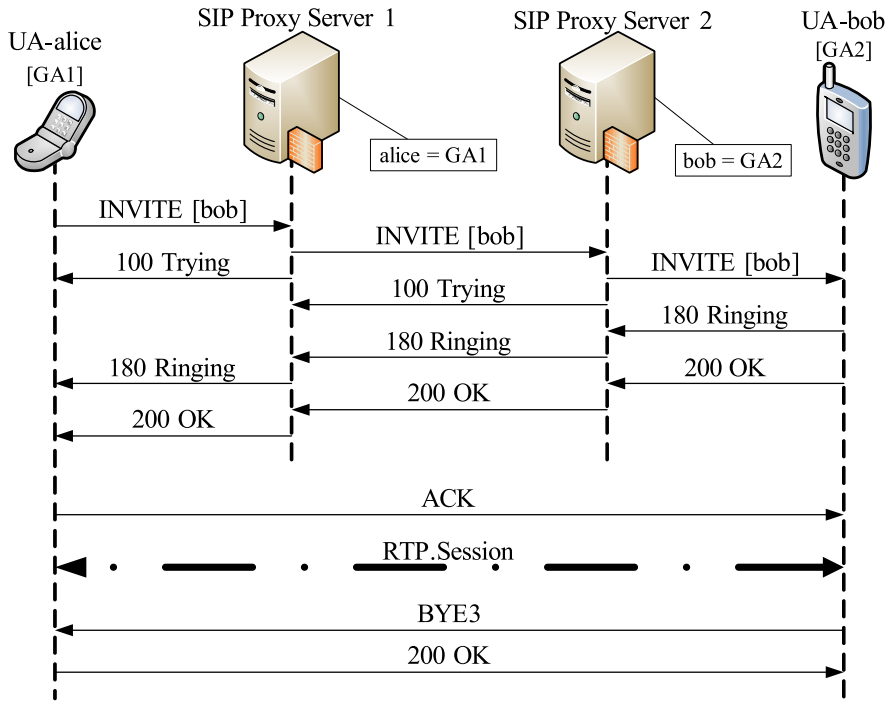


図 C.1　SIP シーケンス例

表 C.1　SIP におけるステータスコード

1xx	暫定応答	要求への処理状況
2xx	成功	要求承認
3xx	リダイレクト	要求を完了させるためには更なる処理が必要
4xx	クライアントエラー	要求の構文誤り，要求が実行できない
5xx	サーバエラー	サーバ上でのエラー
6xx	グローバルエラー	要求はいかなるサーバにおいても処理できなかった

セッションの確立は UA からの INVITE メッセージ送信から始まる．INVITE を受信したプロキシ 1 では，宛先が *bob* であることから，*bob* のプロキシ 2 へ INVITE メッセージを送信する．また，プロキシ 1 は *alice* へプロキシ 2 への INVITE を実行中であることを通知する暫定応答 100Triyng を送信する．この “100” とは，要求に対する結果を示すステータスコードであり，表 C.1 に示すように HTTP で定めたステータスコードを拡張した仕様となっている．プロキシ 2 では，受信した INVITE から配下の *bob* へ INVITE メッセージを送信し，プロキシ 1 へ 100Trying を送信する．INVITE メッセージを受信した *boib* は，電話のベルを鳴らすなど相手からの呼び出し処理を行い，合わせて発信元 *alice* へ呼び出し中であることを伝えるための暫定応答 180Ringing を応答する．*bob* は受話器を取るなどによって，200OK をプロキシ 2，プロキシ 1 を経由して *alice* へ送信する．*alice* は，*bob* からの 200OK を元に ACK を応答することで *alice* と *bob* の間にセッションが生成される．そして *bob* の受話器を戻すなどの動作から，BYE 要求要求と 200OK によってセッションが終了し，通話が終了する．

付 録D　提案方式補足

D.1　同時通信

提案方式において，EN が同一 NTS ルータ内の複数の端末に通信を行う場合の動作を記す．TCP/IP において，通信の識別には IP アドレスとポート番号，プロトコルが使用される．EN が複数のコネクションを張ろうとする場合も，同様に送信元ポートを複数使用して通信を行う．NTS ルータにおいて NAT 処理を行う場合にも，送信元ポート番号を監視しているため，図 3.1 のネットワーク構成において EN が *alice* と *bob* に通信を行った場合に通信を間違って繋げることはない．

EN が名前解決を行ってから NTS ルータが通信を受け取る間に，複数のプロセスにより同一 NTS ルータ内の他の IN へ通信を行っていた場合，異なる IN へ通信を送ってしまう可能性がある．それは 3.2 節で説明した名前解決時の NTS サーバと NTS ルータ間のネゴシエーションについては，“EN の IP アドレス，IN の FQDN” しか通知しないためである．しかし，その他のプロセスも同様に RC を作成しているはずであり，同一経路で通信しているなら通信の前後関係も同じはずなので，問題になる可能性は少ないと予想される．

D.2　イニシエータが NAT 配下に存在する場合

提案方式において，EN が NAT 配下にいる場合の考察を記す．図 D.1 に NAT 配下の端末から通信要求があった場合のシーケンスを示す．ここでイニシエータが存在する NAT

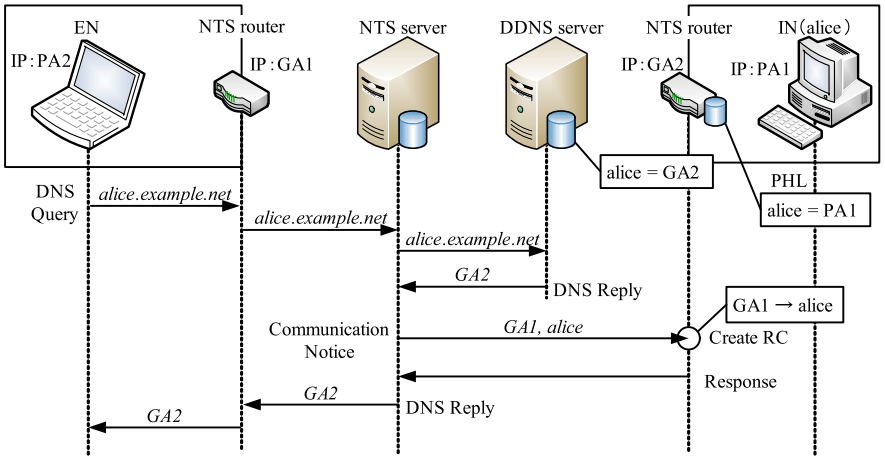


図 D.1　NAT 配下の端末からの DNS 名前解決シーケンス

ルータが NTS ルータとなっているが，通常の NAT ルータでも問題はない．イニシエータが NAT 配下にいる場合でも，実際に通信に使用する送信元 IP アドレスは NAT ルータのグローバル IP アドレスである．よって，DNS クエリの送信元も，相手 IN との通信に使用される送信元も同一 IP アドレスとなるため，イニシエータが NAT 配下に存在する場合でも問題ないことがわかる．

しかし，EN が同じ NTS ルータ内の複数の異なる端末に通信を行った場合は D.1 節で説明したことと同じ問題がある．

D.3　プライマリ DNS 設定

3 章 で説明したように，提案方式では EN は通信に先立って NTS サーバで名前解決を行わなければならない．それは，NTS サーバは NTS ルータへ”*GA*1 から alice へ通信要求がある”ということを伝えなければならないが，NTS サーバが EN より直接名前解決を行われなければ,”*GA*1 から”という情報が得られないためである．しかし，ユーザが故意にその設定を変えることができない場合も考えられる．その場合をサポートするために，プライマリ DNS サーバが NTS サーバでない場合も NAT 越え通信を行える方法を検討した．

そのためには IN は NTS サーバに名前登録を行う必要がある．その場合の名前解決シーケンスを図 D.2 に示す．EN が “alice.example.net” の名前解決を行った時に，DNS のリクエストは最終的に NTS サーバに名前解決依頼がフォワードされてくる．名前解決依頼を受け取った NTS サーバは alice が GA2 だとわかっているので，*GA*2 へは “誰かから alice へ通信要求がある” ということを通知する．その後，NTS ルータは一定秒以内に通信を受け取った場合 NAT テーブルを alice と対応付け，転送することで NAT 越え通信を可能とする．

しかしこの方法では，NTS ルータが同時にいくつかの通信を受け取った場合，間違っ

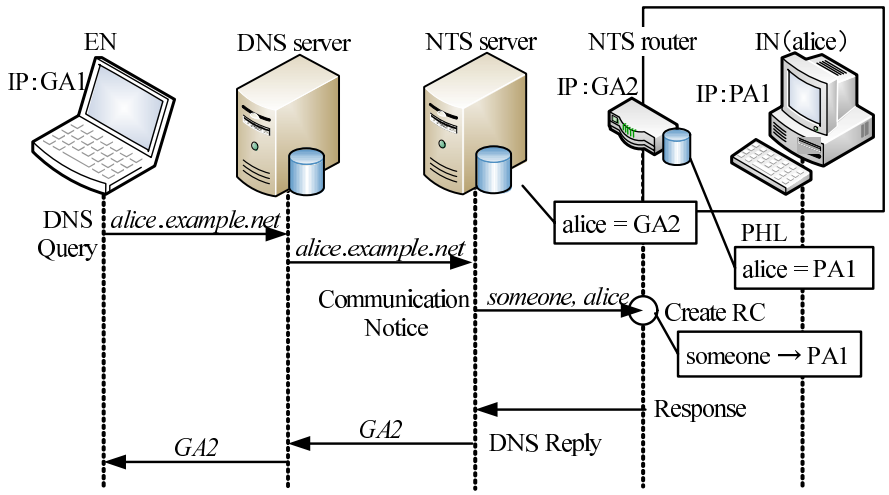


図 D.2　名前解決シーケンス (B 案)

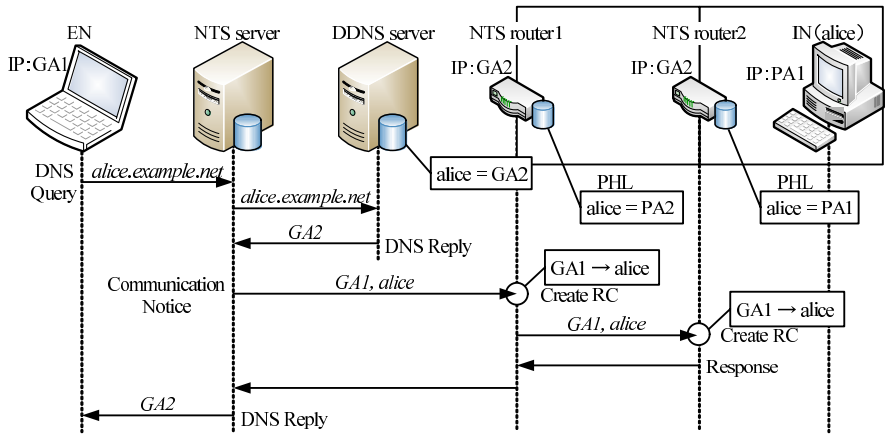


図 D.3　多段 NAT 時の名前解決シーケンス

た通信を alice に送ってしまう可能性があるので，RC の有効時間を適切に選定する必要がある．

D.4　多段 NAT

STUN や UPnP の様に，NAT のマッピング情報をサーバに登録するような方式の場合，NAT が重複配置されるような多段 NAT には対応できない．しかし，企業や学校の様な，一つの組織内に複数のネットワークグループが存在する構成も予想される．そこで，提案方式を多段 NAT に対応できる方法を考えた．

事前設定としては，DNS サーバには外側の NTS ルータのグローバル IP アドレスを IN1 の FQDN と対応させて登録する．IN の存在する内側の NTS ルータには，PHL に IN のプライベート IP アドレスと FQDN を対応させて登録する．外側の NTS ルータには，PHL に内側の NTS ルータのプライベート IP アドレスと FQDN を対応させて登録する．

次に多段 NAT 時の名前解決シーケンスを図 D.3 に示す．EN は IN(*alice*) と通信を開始するに当たり，alice.example.net の名前解決を NTS サーバへ依頼する．NTS サーバは DDNS サーバより NTS ルータの IP アドレス *GA*2 を取得する．次に NTS サーバは *GA*1 から *alice* への接続依頼があることを NTS ルータ 1(*GA*2) に通知する．この通知を受け取った NTS ルータ 1 は PHL を参照し，*GA*1 から *PA*2 へ通信があるということを RC(Request Cache) へ記憶しておく．更に NTS ルータ 1 は受け取った通信通知を NTS ルータ 2*PA*2 へ転送し，NTS サーバへ通知応答を返す．これにより NTS ルータ 2 にも RC として *GA*1 から *PA*1 へ通信があるということを記憶する．最後に NTS サーバは EN に対して NTS ルータのアドレス *GA*2 を応答する．

通信開始時には 3.3 節と同じことが NTS ルータ 1 と NTS ルータ 2 のそれぞれで行われ，*alice* と通信することができる．

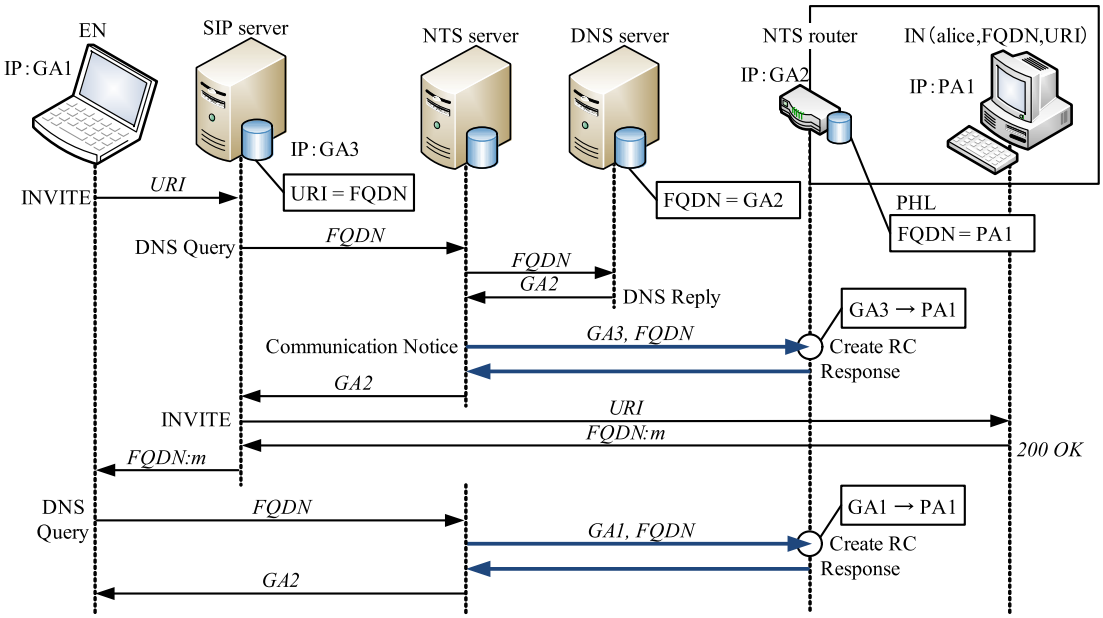


図 D.4　NTS による INVITE シーケンス

D.5　SIP への対応

提案方式では DNS クエリを NTS サーバが受け取ることで NAT 越え通信を可能としているため，SIP の様にプロトコルがアプリケーションにより，SIP サーバより IN の情報を得て通信を開始する場合，NAT 越え通信を提供することができない．そこで，提案方式における，SIP の NAT 越え通信についても検討した．

事前設定としては，本論文で説明した通り，EN は IN へ NTS を利用して NAT 越え通信を行える状況であるとする．更に SIP サーバのプライマリ DNS も NTS サーバに設定しておく．ここで IN(*alice*) は URI と対応させて FQDN を登録しておく．この登録は SIP アプリケーションに設定しておくことで自動で行われる．

図 D.4 に NTS を利用した INVITE シーケンスを示す．この図では簡易的に各 UA(EN,IN) の SIP プロキシは同一である様に記載した．更に *alice* の FQDN と URI を区別するため，それぞれ “*FQDN*” と “*URI*” とした．

始めに EN は自 SIP サーバへ URI への INVITE メッセージを送信する．INVITE メッセージを受け取った SIP サーバは，URI が FQDN であると分かる．ここで SIP は INVITE メッセージを送る *FQDN* が IP アドレスではなく FQDN となっているため，NTS サーバへ名前解決を行う．NTS サーバは *GA*3 から *FQDN* の DNS クエリを受け取ったので，*GA*2 へ *GA*3 から *alice* へ接続以来があることを通知する．この通知を受け取った NTS ルータは PHL を参照し，*GA*1 から *PA*2 へ通信があるということを RC へ記憶し，返答する．NTS サーバは *FQDN* に対する応答 *GA*2 を返す．SIP サーバはここで *URI* が *GA*2 であるとわかったので，*GA*2 へ *URI* への INVITE メッセージを送信する．NTS ルータは先ほど生成された RC により，NAT テーブルを生成して INVITE メッセージを IN(*alice*) へ転送する．INVITE メッセージを受け取った IN(*alice*) は SIP アプリケーションにより通信に

使用する *FQDN* とポート番号を応答する．更に SIP サーバは EN へ応答を転送する．ここで，EN は IN のデータ通信に使用するアドレスとして FQDN を応答されているため，NTS サーバへ名前解決を依頼する．以降は本文で説明した通り，NTSS により NAT 越え通信を行うことができる．