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

加藤　尚樹　　柳沢　信成　　鈴木　秀和
宇佐見　庄五　　渡邊　晃
名城大学大学院理工学研究科
Naoki KATO　Nobushige YANAGISAWA　Hidekazu SUZUKI
Shogo USAMI　Akira WATANABE
Graduate School of Science and Technology, Meijo University

1. はじめに

ユビキタス社会においてはどこにいても自由に通信できることが求められる．しかし，IPv4 の世界ではインターネットで用いられるグローバルアドレス空間と組織内で用いられるプライベートアドレス空間があり，両者を接続するためにアドレス変換装置（以下 NAT）が存在し，その間の通信に制約がある．その理由は，NAT のアドレス変換テーブルが，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で始まる場合のみに生成されるためで，グローバルアドレス空間からプライベートアドレス空間へ通信を開始することができない．この制約を緩和するため NAT にはアドレス変換テーブルを静的にあらかじめ生成しておく IP フォワード機能があるが，ポート番号 1 個に対して 1 台の端末しか設定できないうえ，動的に変更できないので汎用性に欠ける．

これまで，企業ネットワークにおいては NAT と共にファイアーウォールが併設され，内側からの通信開始のみを許可するのが一般的であったため，このような制約は表に出ることはなかった．しかし，今後は家庭にもネットワークが導入されていくことが想定される．よって，外出先からインターネットを通じて家庭内のネットワーク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，上記のような NAT の制約を除去することは有益である．

グローバルアドレス空間からプライベートアドレス空間への通信開始を汎用的に可能にしようとする方式として，STUN[1]，AVES[2]，IPv4+4[3]，NATS[4-8]などがある．

STUN（Simple Traversal of User Datagram Protocol（UDP）Through Network Address Translators）はあらかじめプライベート空間側の端末がインターネット上に公開された STUN サーバに利用可能なポートと NAT のグローバル IP アドレスを登録し，グローバル空間側の端末が通信開始時に STUN サーバに問合せることによって NAT の制約を除去する方式である．しかし，インターネット上に第三のサーバをおく必要があり，UDP 通信に限定されるという課題がある．

AVES（Address Virtualization Enabling Service）は waypoint と呼ぶ装置をインターネット上に設置し，waypoint と NAT BOX が協調することにより，プライベート空間側の端末へパケットを転送する方式である．しかし，STUN 同様，第三の機器をインターネット上に配置する必要があることや，DNS に改良を加える必要があるという課題がある．

IPv4+4 は，DNS から通信相手及び通過するアドレス変換装置の IP アドレスを得て，IP ヘッダを多重化し，経路上で複数の IP ヘッダを入れ替えていくことにより，通信を可能とする方式である．この方式では全てのルータに IPv4+4 を実装する必要があり，カプセル化によるオーバヘッドが発生する．

NATS（Network Address Translation with Sub-Address）は DNS と連携してサブアドレスと呼ばれる新しい IP アドレス体系を定義し，NAT 上で IP in IP Tunneling[5]を用いてパケットをカプセル化／デカプセル化する方式である．しかし，全パケットに対してカプセル化／カプセル解放処理を行うため，NATS BOX に高い負荷がかかる．また，プライベートアドレス空間からの DNS 問い合わせを NATS BOX が監視し，パケットのフッキング処理を行う必要がある．

本稿では，端末と NAT が協調して NAT テーブルを強制的に生成させ，かつ端末側がポート番号の変換を行うことにより，NAT が生成した NAT テーブルをそのまま使用した通信を可能とする NATF（NAT Free Protocol）[9]を提案する．NATF は端末および既存の NAT BOX に若干の改造を加えることで実現可能である．

以下 2 章では NAT における問題点，3 章で既存技術による解決，4 章で NATF の概要，5 章で実装方法について述べ，6 章でまとめる．

2. NAT における問題点

NAT にはアドレス変換のみを行う狭義の NAT と，アドレス変換とポート変換を行う NAPT（Network Address Port Translator）がある．前者は複数のグローバルアドレスをプールして通信ごとにアドレスを変える必要があるが，後者は 1 つのグローバル IP アドレスを用いてプライベート空間の複数の端末を同時接続可能であり，通常は NAPT が用いられることが多い．以下の記述では，用語として NAT を用いるが動作としては NAPT を対象とする．

以下に NAT の原理とその課題を述べる．図 1 に NAT の動作を示す．プライベートアドレス空間に所属する端末がグローバルアドレス空間に所属する WEB サーバへ HTTP 通信を開始するものとする．NAT BOX は NAT 機能が搭載されたアドレス変換装置である．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．はじめにクライアントは宛先を IP アドレス GA1，ポート番号 80，送信元を IP アドレス PA1，ポート番号 X として送信する（①）．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 BOX では送信元を IP アドレス GA2，ポート番号 Y へと変換して中継する（②）．Y は NAT BOX が動的に選んだ任意のポート番号である．このとき NAT BOX はこの変換の関係を記した NAT テーブルを生成する．上記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，ポート番号 Y，送信元 IP アドレス GA1，ポート番号 80 として返

信する（③）．NAT BOX がこのパケットを受信すると，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，クライアントがこれを受信する（④）．以後の通信は NAT テーブルに従って，NAT BOX 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から通信を開始する場合を図 2 に示す．グローバルアドレス空間に所属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 通信を開始するものとする．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においては無効な値であり送信ができない（①）．また，仮に NAT BOX のグローバル IP アドレスを知ることができて，NAT BOX までパケットを送信できたとしても，NAT BOX には NAT テーブルが存在しないためパケットは破棄される（②）．即ち，プライベートアドレス空間にサーバ，グローバル空間にクライアントが存在するシステムは構築できない．NAT にはあらかじめ NAT テーブルを静的に設定して，グローバルアドレス空間からの通信開始を可能とする IP フォワードと呼ぶ機能がある．しかしこの方法では 1 つのポートに対してサーバを 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．

3. 既存技術による解決とその課題

上記制約を除去する技術である STUN や AVES はインターネット上に第三の装置が必要であり，今後の P2P 通信の発展を考えるとこのような構造は好ましくない．また，1 点障害に弱いことや負荷の集中が懸念される．IPv4+4 は全てのルータに機能を追加しなければならないため現実的な解決策とは言えない．上記のような理由から，既存技術としては NATS が最も現実的で我々のコンセプトに近いと考えられるので，NATF との比較対象として以下に詳細に説明する．

図 3 に NATS の動作を示す．図 2 と同様，グローバル空間に端末，プライベート空間に WEB サーバ，その間に NATS 機能を搭載したアドレス変換装置(以後 NATS BOX)が配置される．NATS 機能を利用するにあたって，端末，NATS BOX，DNS サーバに機能が追加される．まず端末は DNS による名前解決を行う（①）．このとき通常の A レコード問合せによる IP アドレスの取得とともに，NATS 独自のアドレス体系であるサブアドレスを取得する（②）．サブアドレスとは NATS BOX のグローバル IP アドレス GA2 と WEB サーバの IP アドレス PA1 を組にしたものである．取得したサブアドレスを元に宛先 PA1，送信元 GA1 のパケットを宛先 GA2，送信元 GA1 の IP ヘッダでカプセル化して送信する（③）．これを NATS BOX が受信するとカプセル解放処理を行い，WEB サーバへと転送する（④）．WEB サーバは応答パケットを宛先 GA1，送信元 PA1 として送信する（⑤）．このパケットを NATS BOX が受け取ると，送信元を PA1 から GA2 へと書き換えた IP ヘッダでカプセル化して端末へと転送する（⑥）．以後の通信は同様の処理によって行われる．NATS では全通信パケットに対して NATS BOX がパケットのカプセル化／デカプセル化を行う必要があり，処理が NATS BOX に集中するため負荷が大きい．またサブアドレスを DNS サーバに登録する必要があり，これを取得するため DNS シーケンスに変更を加える必要があるなどの課題がある．

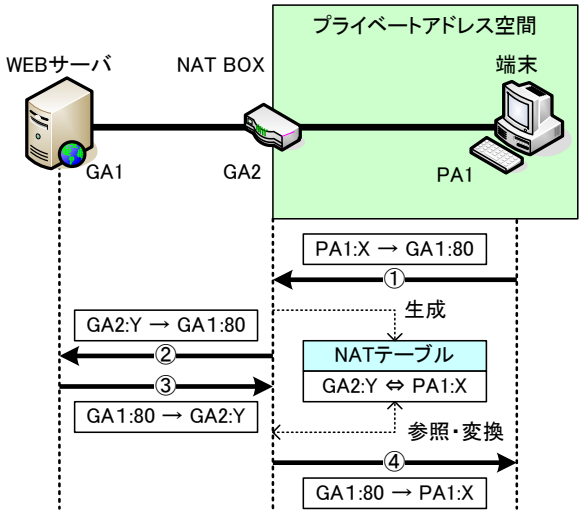

図 3: NAT の動作

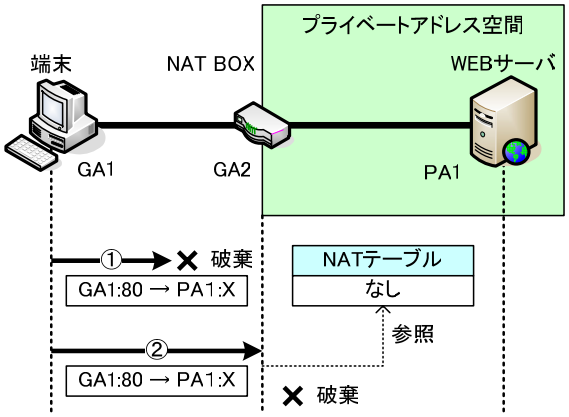

図 3: NAT の制約

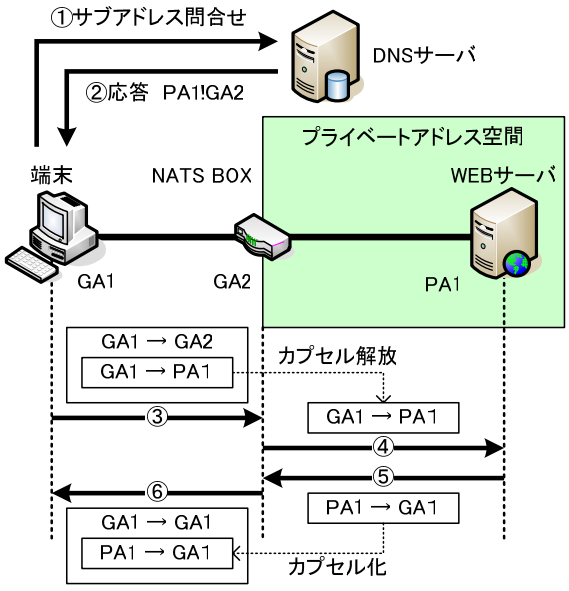

図 3: NATS の動作

4. NATF

4.1. 構成と初期情報

NATF を利用する最初のユーザとして，自宅のインターネット環境を整え，ネットワークを構成できるパワーユーザを対象とし，インターネット上からホームネットワークの機器へアクセスすることを想定している．

NATF の構成を図 4 に示す．図 3 と同様，グローバル空間に端末，プライベート空間に WEB サーバ，その間に NATF 機能を搭載した NAT BOX（以下 NATF BOX）が配置されている．プライベートアドレス空間のドメイン名を『home.com』とし，NATF BOX のグローバル IP アドレス GA2 を DNS サーバに登録しておく．また WEB サーバのホスト名は『www』とし，FQDN は『www. home.com』となる．端末と NATF BOX には以下の内容をあらかじめ登録しておく必要がある．即ち，端末にはアクセスしたい NATF BOX 配下の端末のホスト名『www』とそのプライベートアドレス空間のドメイン名『home.com』を組とした NRDB（Name Resolution Data Base）を，NATF BOX には配下のホスト名『www』とその IP アドレス『PA1』を組とした APDB（Access Permission Data Base）を登録する．

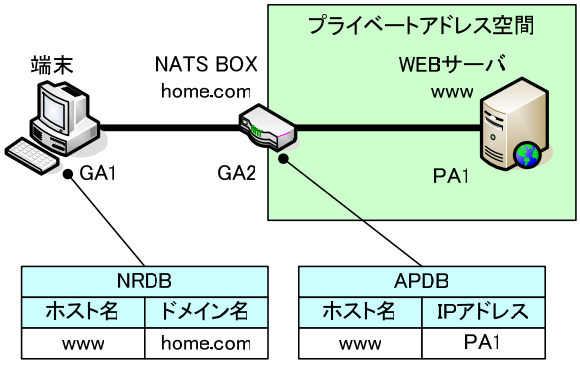

図 6: NATF の構成と初期情報

4.2. 動作概要

動作概要を図 5 に示す．まず端末は DNS に対し，『home.com』に対する DNS 問合せを行い， NATF BOX のグローバル IP アドレス GA2 を得る．次に，通信に先立って端末と NATF BOX 間でネゴシエーションを行う．このネゴシエーションにより，NATF BOX で強制的に NAT テーブルを生成し，端末側では NAT テーブルにあわせたポート変換テーブル（以下 FAT テーブル）を生成する．NATF ネゴシエーションが終了すると，端末でのポート番号変換処理と NATF BOX での通常の NAT 処理によって通信が行われる．

以降 4.3 節で NATF における DNS 名前解決，4.4 節で NATF ネゴシエーションについて詳しく述べる．

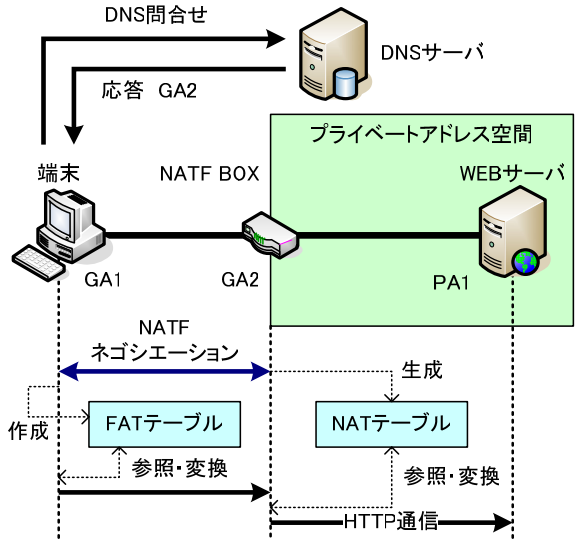

図 6: NATF の動作概要

4.3. DNS 問合せ時の端末の動作

図 6 に DNS 問合せ時の端末の動作を示す．端末のアプリケーションから OS に対し『www.home.com』の問合せを依頼すると，OS では NRDB 検索を行う．問合せ内容が NRDB 内のホスト名+ドメイン名にヒットした場合，問合せ内容のホスト名の部分を除去してドメイン名だけで DNS サーバへ問合せを行う．DNS サーバにはドメイン名『home.com』に対する A レコードとして GA2 が登録されているため端末は GA2 を取得することができる．アプリケーションへ応答を返す前に，名前解決の結果を一時的に記憶しておく．ヒットしなかった場合は NATF を適用する必要がないと判断し，そのまま DNS 問合せを行う．

この機能によって DNS サーバに変更を加えることなく，アプリケーションに通信相手を NATF BOX であることを認識させることができる．

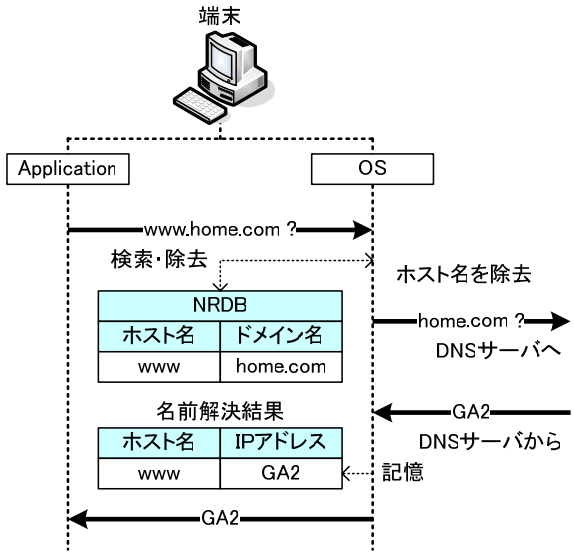

図 6: DNS 問合せ時の端末の動作

4.4. NATF ネゴシエーション

NATF ネゴシエーションは ICMP を利用した 1 往復のシーケンスからなり， NATF BOX の NAT テーブルを強制的に生成することと，NAT テーブル生成時に NATF BOX が動的に選んだポート番号を端末に通知して端末の FAT テーブルを生成することが目的である．図 7 に NATF ネゴシエーションの動作を示す．アプリケーションからのパケットが OS に

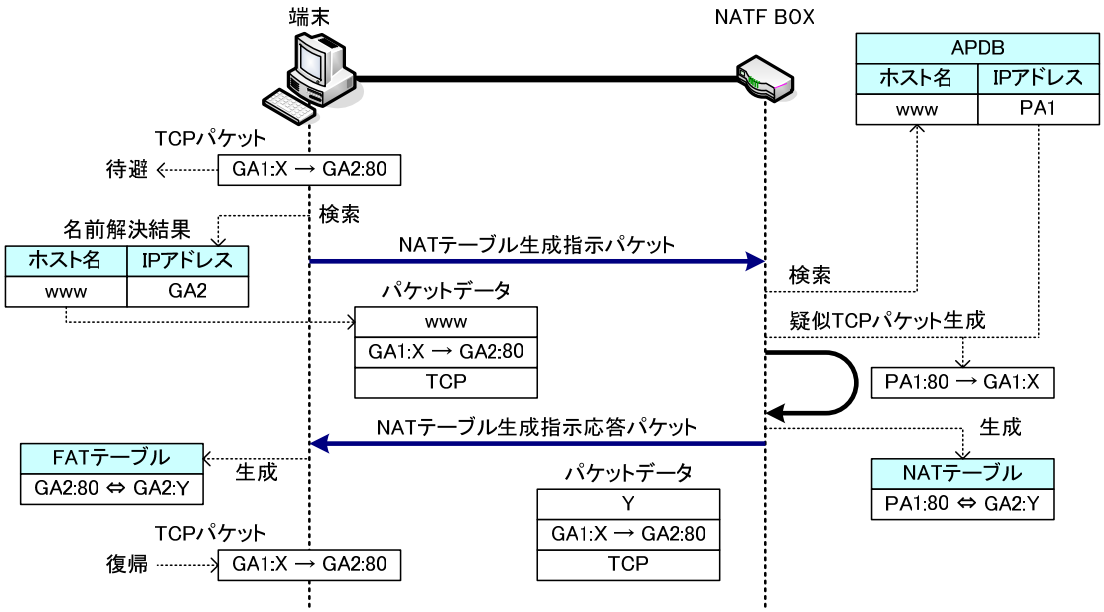

図 8: NATF ネゴシエーションの動作

渡されると，先ほど記憶した名前解決の結果を宛先 IP アドレスで検索する．ヒットした場合，NATF ネゴシエーションを開始する．以後，NATF ネゴシエーションのトリガーとなったパケットを第 1 パケットと呼ぶ．第 1 パケットの宛先 GA2，ポート番号 80，送信元 IP アドレス GA1，ポート番号 X，プロトコルタイプ『TCP』と，名前解決結果の検索結果よりホスト名『www』を NAT テーブル生成指示パケットとして NATF BOX へ送信する．第 1 パケットは NATF ネゴシエーションが終了するまで OS 内に待避される．

NATF BOX では NAT テーブル生成指示パケットを受信すると，APDB を受信したホスト名『www』で検索する．ヒットした場合，検索結果の IP アドレス PA1 と，受信した第一パケットの情報から擬似パケットを作成する．疑似パケットは受信した情報に含まれているプロトコルタイプと同じパケットとして生成され，送信元 IP アドレス PA1，ポート番号 80，宛先 IP アドレス GA1，ポート番号 X が設定される．これは端末の第 1 パケットの応答パケットに相当するものに見せかけたパケットである．疑似パケット生成後，自分宛に送信し，強制的に NAT テーブルを生成する．NAT テーブルを生成するとき，擬似パケットの送信元ポート番号 80 は Y に変換される．ポート番号 Y は NAT テーブル生成指示応答パケットとして端末に送信され，端末では 80 と Y の変換を指示する FAT テーブルが生成される．その後，待避していた第 1 パケットを復帰させて NATF ネゴシエーションが終了する．

5. 実装

NATF の機能は端末，NATF BOX ともに IP 層に実装され，共通化することができる．実装に利用した OS は IP 層の情報が豊富な FreeBSD である．図 8 に NATF の実装概要を示す．NATF は IP 層の入出力関数 ip_input( )，ip_output( )から NATF を呼び出して処理を行う．NATF で処理された通信パケットは元の位置に戻され，既存の IP 層の処理には一切影響を与えない．NATF BOX は NAT デーモンの natd が動作しており，IP 層から divert ソケットを通じてアドレス変換

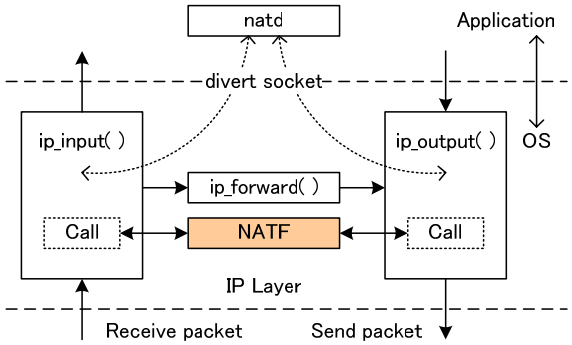

図 7: NATF の実装概要

の処理を行う．

5.1. 端末における処理

図 9 に端末における NATF 処理フローを示す．NATF のモジュールが呼び出されるとパケットを判別し，適切なモジュールを呼び出して処理を行う．UDP の 53 番ポートであった場合，DNS 問合せに関するパケットであるため，NRDB を検索する．NRDB にヒットした場合，DNS 書き換えモジュールに処理を渡す．そうでない場合は通常の問合せであるため，何もせずにリターンする．ICMP パケットであった場合，さらに NATF ネゴシエーションパケットかどうかを判別する．NATF ネゴシエーションパケットであった場合，FAT に情報を登録した後，ネゴシエーションパケットを破棄する．NATF ネゴシエーションパケットでなかった場合は何もせずにリターンする． TCP または 53 番ポート以外の UDP パケットであった場合は，FAT を検索し，ヒットした場合はポート番号変換モジュールに処理を渡し，宛先ポート番号を変換する．そうでなかった場合は何もせずにリターンする．

5.2. NATF BOX における処理

図 10 に NATF BOX における NATF 処理フローを示す．NATF BOX においても端末と同様に，パケットの種類を判別して適切なモジュールを呼び出す．ICMP パケットであった場合，さらに NATF ネゴシエーションパケットかどうかを判別する．NATF ネゴシエーションパケットであった場合，擬似パケット生成モジュールを呼び出す．NATF ネゴシエーションパケットでなかった場合は何もせずにリターンする．TCP または UDP パケットであった場合，さらに擬似パケットかどうかを判別する．擬似パケットであった場合，擬似パケットの情報から NAT テーブル生成指示応答パケットを生成して送信する．その後，疑似パケットを破棄する．擬似パケットでなかった場合はそのままリターンする．

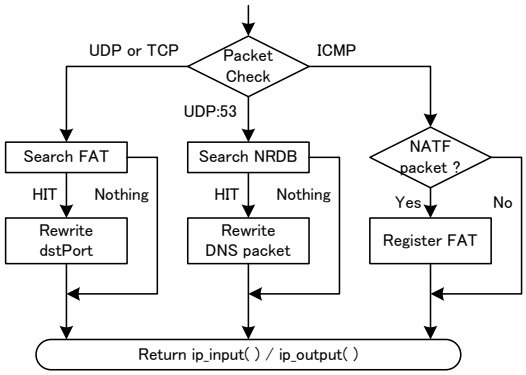

図 10：端末における NATF 処理フロー

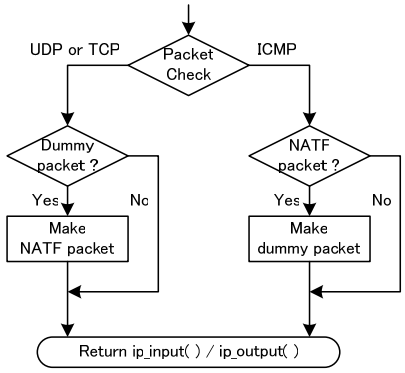

図 10：NATF BOX における NATF 処理フロー

6. まとめ

本稿ではグローバルアドレス空間からプライベートアドレス空間内の複数の端末へアクセスを開始することができる通信方式 NATF を提案し，実装について報告した．

今後は，NATF の実装を完了して，動作検証を行い，機能の有効性について確認する．

参 考 文 献

[1] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489 (2003).

[2] T.S.Eugene Ng, I.Stoica, H.Zhang, "A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces", USENIX 2001 (2001).

[3] Z. Turanyi, A. Valko, "IPv4+4", ICNP2002 (2002).

[4] Kuniaki Kondo, "Capsulated Network Address Translation with Sub-Address(C-NATS)", Internet Draft (2002).

[5] Kuniaki Kondo, "Possibility of NATS Communications Summary", http://www.nats-project.org/com-possibility-sum.html

[6] Kuniaki Kondo, "Capsulated NATS Protocol Overview", http://www.nats-project.org/presentations/Capsulated-NATS-Overview.pdf

[7] Kuniaki Kondo, "NATS Address Translation Practice", http://www.nats-project.org/presentations/NATS_Address_Translation_Practice.pdf

[8] Kuniaki Kondo, "NATS の適用範囲とプロトコルの概要", http://www.nats-project.org/presentations/NATS-exp-Generic.pdf

[9] 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"，情報技報，2005-DPS-122，pp.351-356 (2005).