目次

あらまし

IPv4 アドレス空間には，グローバルアドレス，プライベートアドレスの 2 つの空間があり，両者の間には NAT が設置され自由に通信を行うことができない．具体的には，NAT のアドレス変換の原理に起因して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に対して通信を開始することができない．IPv6 が普及すればアドレス変換が不要となり，このような課題は解決される可能性はあるが，IPv4 は当面の間 IPv6 と共存しつつ使い続けられると思われる．我々はこの NAT の通信制約を解決するプロトコルとして，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が可能である NATF（NAT Free Protocol）を提案している．本稿では NATF の考え方を拡張し，グローバルアドレス環境をはさんだ異なるプライベートアドレス空間に存在する端末どうしの通信を可能とする CIPA（Communication between terminals in Independent Private Address areas）の提案する．また提案方式の実装を行い提案方式の動作が可能であることを確認した．また性能測定を行い，通常の通信時にほとんど影響を与えないことを確認した．

1. はじめに

ユビキタス社会とは,いつでも誰でもどこからでも自由に通信できる社会である．しかし，IPv4 の世界においては IP アドレス空間としてグローバルアドレス空間（GA 空間）とプライベートアドレス空間（PA 空間）[1]の 2 つの空間があり，両者は自由に通信を行うことができない．具体的には，アドレス変換装置（NAT）のアドレス変換の原理に起因して GA 空間から PA 空間に対して通信を開始することができない．IPv6[2]が普及すればアドレス変換が不要となり，このような課題は解決される可能性はあるが，IPv4 は当面の間 IPv6 と共存しつつ使い続けられると思われる．また，IPv6 を適用したときに，内部のネットワークが見えてしまうのは問題であるとの指摘もあり,アドレス変換は IPv6 の時代でも使われる可能性がある．よって上記課題の解決を検討することは意味のあることと考えられる.

IPv4 の PA 空間と GA 空間との間にはアドレス変換装置 (Network Address Translation，以下 NAT)[3]の設置が必須で，多くの場合ファイアウォールに内蔵される.企業ネットワークにおいてはセキュリティーポリシーによりファイアウォールを用いて自主的に通信制限をかけるため，NAT による通信の制約は表に出てこない.しかし,今後家庭にもネットワークが普及していった場合,通信の利便性の向上が要求されるため NAT による通信の制約を除去することが望まれる.

GA 空間から PA 空間への通信の開始ができない理由は PA 空間から GA 空間に抜ける最初のパケットによってのみ NAT のアドレス変換テーブルが生成されるためである.事前に静的にテーブル内容を登録しておくことによりこの課題を解決する方法（ポートフォワーディング）[3]もあるが，ネットワーク構成やアプリケーションに応じてテーブルを設定する必要があり,自由な通信とは言えない.

NAT を越える通信方式の研究には, GA 空間上にサーバを用意するサーバ中継方式とサーバを使用しない P2P 方式がある.

サーバ中継方式は STUN[4]や AVES[5]，SoftEther[6]が挙げられる．STUN は端末と STUN サーバが予め NAT にテーブルを作らせておき，端末同士が通信を開始するとき，上記 NAT テーブルの情報を使い通信を行う．STUN の課題は特定の仕様をサポートする NAT でしか使えないこと,また UDP でしか通信できないことがあげられる．STUN を改良し TCP で通信を行う TCP Hole Punching[7]という技術もあるが，使用できる NAT がさらに限定される．AVES は waypoint と NAT が協調してパケットをカプセル化することにより NAT を越えるが，カプセル化によるオーバヘットが発生する．SoftEther は端末およ

びサーバ上にソフトウェアにより仮想 LAN カード，仮想 HUB といった仮想的な Ethernet ネットワークを構築することにより，あらゆるアプリケーションの通信が可能となる．しかし，通信端末間でアドレス環境を統一的に管理する必要であること，セキュリティを別途考慮する必要があることなどの課題がある．サーバ中継方式全般の欠点として，サーバを設置するためのコスト増加やサーバを中継するための遅延が生じる点があげられる．今後 P2P 通信が発展して行くことを考えると，サーバ中継方式は望ましい方式とは言えない．

P2P 方式として IPv4+4[8]や NATS[9]が挙げられる．IPv4+4 は IP ヘッダを改造して NAT が持つグローバル IP アドレスと端末が持つプライベート IP アドレスの 2 つの宛先・送信元アドレスを持たせることにより NAT を通過する．しかし，この方式では IPv4+4 を全端末および NAT に実装しなければならない．NATS は，独自のサブアドレスを定義し，DNS サーバと NAT が連携して IP in IP カプセリングにより NAT を通過する．しかし，サブアドレスを別途定義しなければならないことや，NAT がカプセル化を行うためのオーバヘッドが発生するなどの課題がある．

これらの課題を解決するため，我々は端末と NAT が強調することにより GA 空間からの通信開始を可能とする NATF（NAT Free Protocol）を提案している．本研究では，NATF の考え方を更に拡張し，グローバルアドレス環境をはさんで異なるプライベートアドレス環境にある端末同士の通信を可能にする方式を提案する．この方式を，ここでは CIPA（Communication between terminals in Independent Private Address areas；サイパ）と呼ぶ．

CIPA では，NATF で拡張した端末の機能を 2 台の NAT が実行する．すなわち，NAT どうしが，通信開始に先立って情報を事前に交換し，その情報を元に通信パケットの IP アドレス変換，ポート番号変換を行う．これにより異なるプライベートアドレス空間にいる端末どうしの通信を行うことが可能になる．

以下，2 章に NATS，3 章に NATF，4 章に CIPA の実現方式について，5 章に実装方法，6 章に評価実験結果，7 章にまとめを述べる．

2. 従来研究

本章では，従来研究の代表として NATS をとりあげ，その実現方法と課題を説明する．NATS は P2P 方式であること，NAT 越えを実現したい環境にだけ機能を追加すればよいなどの点で NATF と共通点がある．NATS は DNS サーバ，端末，NAT が連携し，独自のサブアドレスを用いることにより NAT 越えを実現する．サブアドレスを IP アドレスとは別に定義するため、同一のポート番号への接続もサブアドレス別に行う事ができ、ポート番号の衝突が発生しない。また特定のアプリケーションやプロトコルに依存しないため、IP を使った通信全般をサポートできる．

NATS の技術を利用して異なるプライベートアドレス空間にいる端末同士が通信を行おうとした場合の例を図 1 に示す．NATSBOX とは，NATS の機能を実装した NAT 装置である．DNS サーバには IP アドレスとサブアドレスを事前に登録しておき，問合せ時には両方のアドレスを返す．図 1 の構成においては，端末 A と端末 B は通常の端末でかまわない．端末 A が持つサブアドレスを SA,端末 B が持つサブアドレスを SB とする.SX,SY は NATSBOX がアドレス変換用に事前に管理しているサブアドレスである．

まず端末 A が端末 B に通信する際，DNS サーバに端末 B の IP アドレスの問合せを行なう．DNS サーバは NATSBOX2 がもつアドレス（B）とその内部にいる端末 B のサブアドレス（SB）を合わせて返す．NATSBOX1 はこの応答を端末 A に送信する際，DNS 応答の情報だけを取り出す．ここで NATS テーブル 1 を作成し応答のアドレス（B,SB）を別のサブアドレス（SX）に書き換える．

次に端末 A は端末 B 宛ての通信を開始する．この通信パケットの宛先アドレスは SX，送信元アドレスは SA である．NATSBOX に宛先アドレスを SX としたパケットが届いたら NATS テーブル 1 を検索する．もし一致する情報があればカプセル化を行い NATSBOX2 へ送る．カプセル化のアドレスは送信元アドレスが A，SA，宛先アドレスが B，SB となる．このパケットが NATSBOX2 へ届いたらデカプセル化を行なう．ここで NATS テーブル 2 を検索し一致する情報がなければ送信元アドレス（A，SA）を別のサブアドレス（SY）に書き換えるテーブルを追加する．そしてアドレスの書き換え端末 2 へフォワードする．ここでは送信元アドレスを SY，宛先アドレスを SB とし端末 2 に送信する．

以下 NATSBOX では通信パケットごとに NATS テーブルを検索しカプセル化，デカプセル化を行い通信する．

NATS はこのような原理であるため，新たにサブアドレスを定義し，その情報を DNS に登録しなければならない．また，全パケットに対してカプセル化，

デカプセル化を行うため，NATSBOX の負荷が大きいという課題がある．

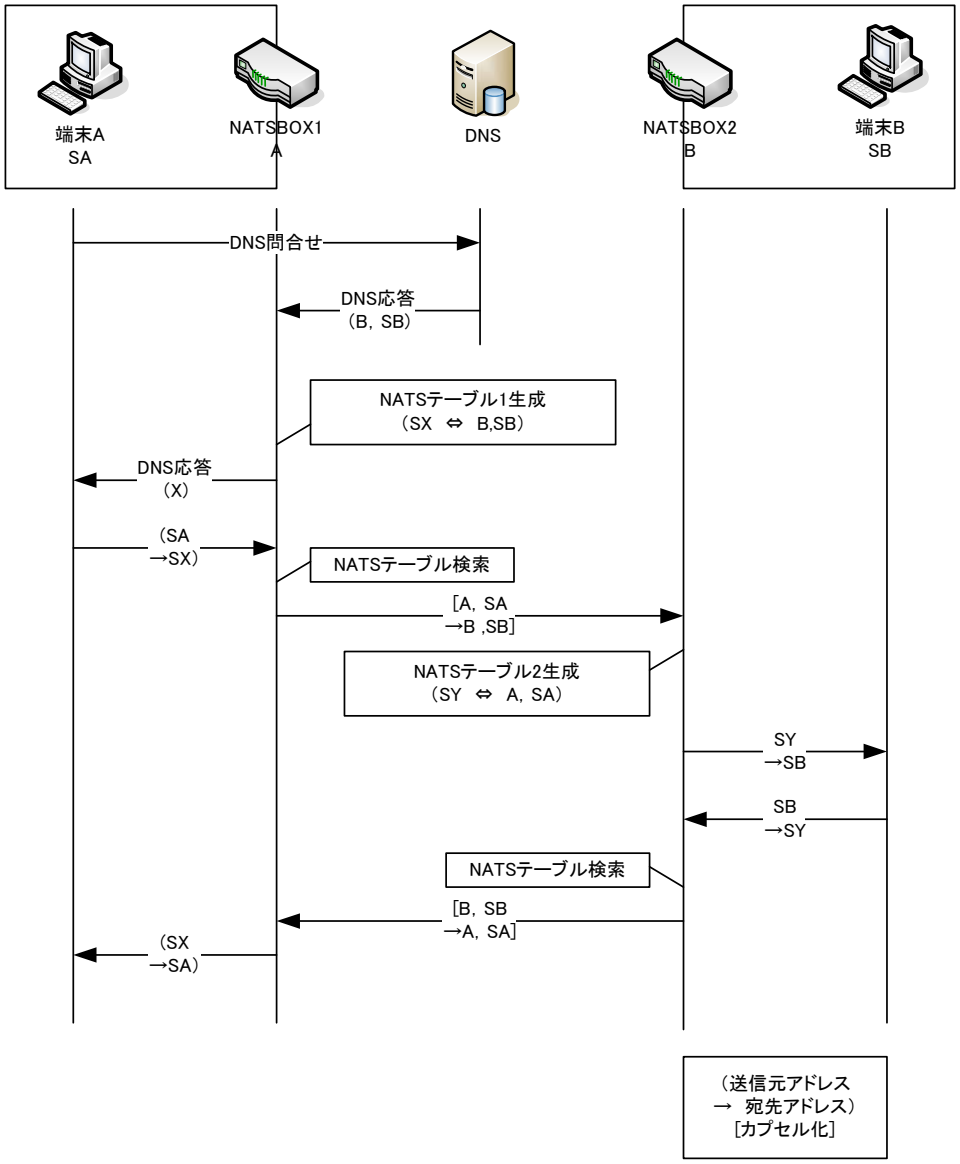

図 1　NATS を用いたプライベートアドレス端末どうしの通信

3. NATF

本章では CIPA のベースになる NATF について説明する．NATF は，端末と NAT が連携し，GA 空間の端末から PA 空間の端末への通信開始を可能とするプロトコルである．NATF では通信に先立ち，利用するポート番号の情報を交換することによって端末側でポート番号変換を行うため，パケット長を変化させる必要がなくオーバヘッドが少ないという利点がある．

図 2 に NATF の環境を示す．端末 A には NATF プロトコルを適用する．NATF が適用された NAT 装置を NATFBOX と呼ぶ．端末 B は通常の端末である．DNS サーバは既存の装置でよく改造は不要である．NATF のシーケンスを図 3 に示す．DNS サーバには，あらかじめ NATFBOX が持つグローバルアドレス（GA2）とドメイン名（natf.com）を登録しておく．NATFBOX1 の配下にいる端末 B にはホスト名（h）を持たせる．また初期設定として端末 A には通信先ホスト名とドメイン名を，NATFBOX にはホスト名とプライベート IP アドレスを登録しておく．端末 B は HTTP サーバとし，端末 A から端末 B へ通信する流れを説明する．

端末 A は DNS サーバに対して通信相手の(ホスト名 + ドメイン名)の IP アドレスを問い合わせる．ここではホスト名を h，ドメイン名を natf.com とする．端末 A では，問合せパケットが IP 層を通過する時に，ホスト名 + ドメイン名が登録されているか否かをチェックする．登録してあれば，DNS 問合せパケットの問合せ部からホスト名（h）を削除し，ドメイン名(natf.com)だけを DNS サーバに問い合わせる．DNS サーバはこの問い合わせに対し，NATFBOX2 のグローバルアドレス GA2 を応答する．

端末 A が上記 DNS 応答を受信した時は，IP 層にて IP アドレス（GA2）を保存する．次に DNS 応答の問合せ部にホスト名を追加し元の問合せ内容（ホスト名（h）+ドメイン名(natf.com)）に戻し，アプリケーションに渡す．これにより端末 A のアプリケーションは通信相手が NATFBOX であると認識する．

次に，端末 A は NATFBOX に対して通信を開始する．宛先アドレス：ポート番号は GA2:80 で送信元アドレス：ポート番号は GA1：a である．a は OS が自動的に割り振ったポート番号である．端末 A は最初の通信パケットを OS 内に一時待避し，NATF シーケンスを実行する．NATF シーケンスはポート番号報告指示とポート番号報告応答からなる．ポート番号報告指示パケットには端末 A が通信中に使う送信元アドレス（GA1）・ポート番号(a)，宛先アドレス(GA2)・ポート番号(80)，プロトコル(TCP)，通信相手のホスト名(h)の情報が含まれる．NATFBOX がポート番号報告指示を受信したらその内容と初期登録の情報により NAT テーブルを生成する．生成される NAT テーブルの詳細は図 6 のとおりである．NATFBOX はここでアドレス変換用に割り当てたポート番号

(x)を，ポート番号報告応答パケットを用いて端末 A に送り返す．端末 A はこの情報を元に通信パケットのポート番号を変換するためのテーブル（FAT テーブル(natF Address Translation)）を作成する．FAT テーブルの詳細は図 5 の通りである．

端末 A は以後の通信においては宛先ポート番号を，FAT テーブルを用いて変換する．一方，NATFBOX は上記 NAT テーブルを用いて IP アドレス変換とポート番号変換を行う．以上の動作により，GA 空間から PA 空間への通信が開始可能となる．

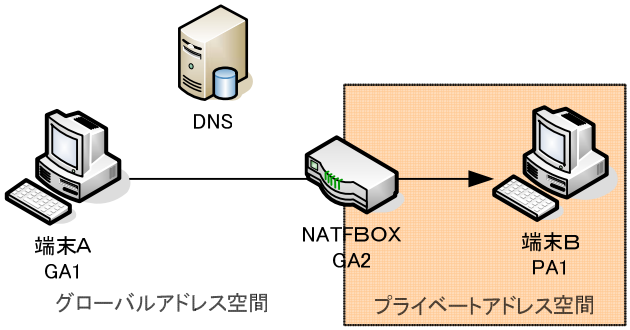

図 2　NATF の環境

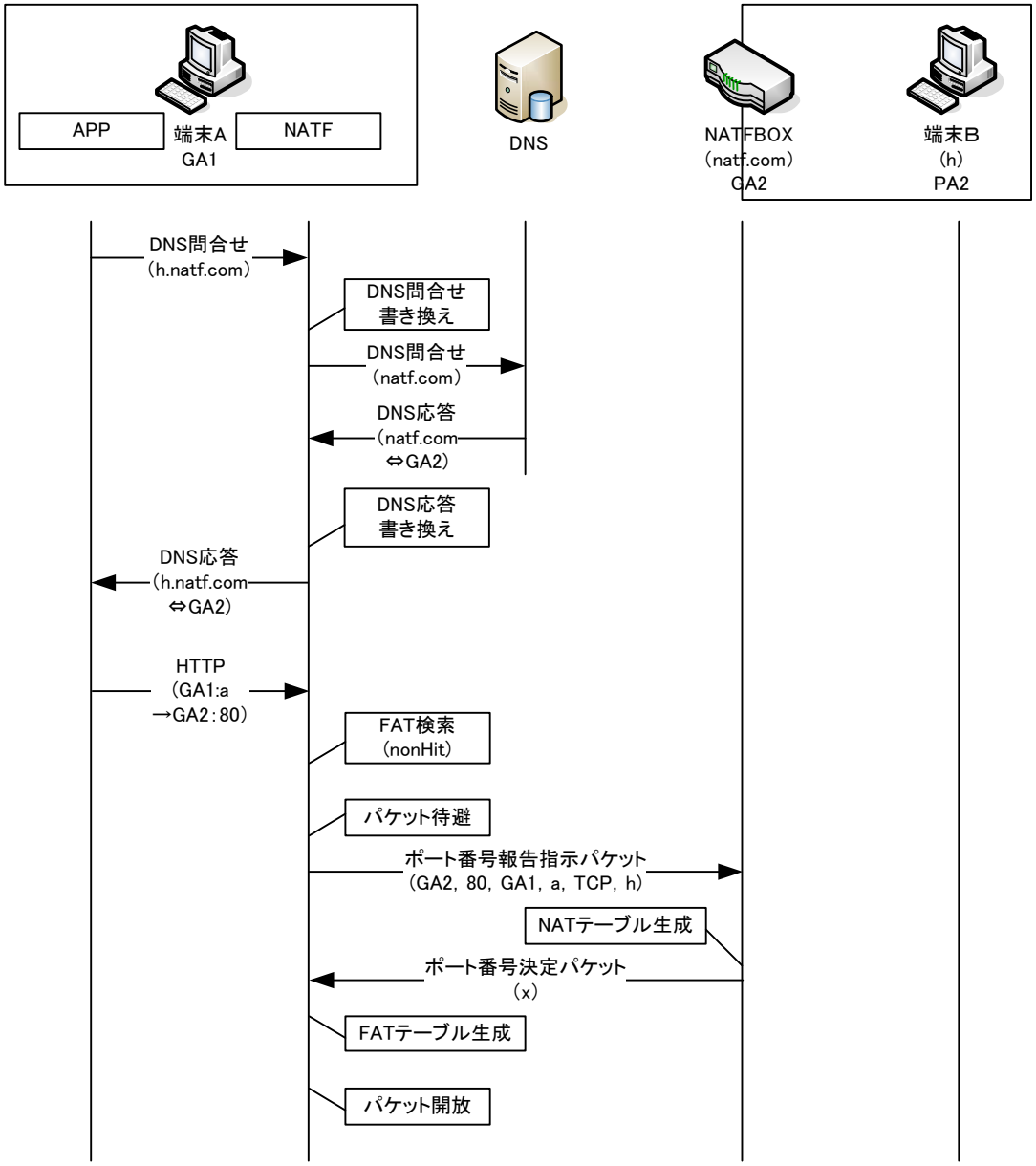

図 3　NATF プロトコルの流れ

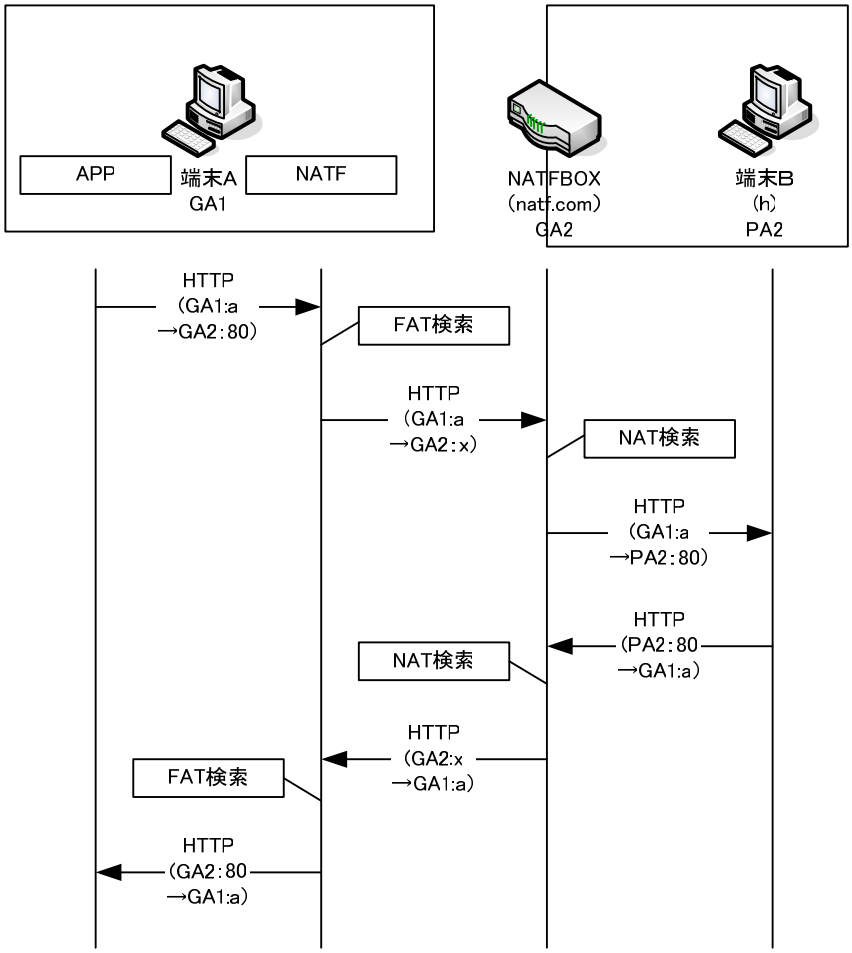

図 4　NATF プロトコル後の流れ

FATテーブル					
[in]	sIP	dIP	sPort	dPort	proto
	GA2	GA1	x	a	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	GA2	GA1	80	a	
[out]	sIP	dIP	sPort	dPort	proto
	GA1	GA2	a	80	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	GA1	GA2	a	x	

図 5　FAT テーブル

NATテーブル					
[in]	sIP	dIP	sPort	dPort	proto
	GA1	PA2	a	80	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	GA1	GA2	a	x	
[out]	sIP	dIP	sPort	dPort	proto
	PA2	GA1	80	a	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	GA2	GA1	x	a	

図 6　NAT テーブル

4. CIPA

CIPA は，NATF を拡張し，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を可能とする．提案システムの環境を図 7 に示す．CIPA では通信端末は一般端末でよく，NATFBOX 同士が NATF プロトコルを実行する．送信元の NATFBOX は NAT アドレス変換に加え FAT 変換を行う．図 7 のような環境では,プライベートアドレス空間のアドレスが重複する場合もあり得る。そこで本章では，端末 A と端末 B が同一のプライベートアドレスであっても通信が可能であることを示すため,実アドレスを用いて動作を説明する．端末 A,B のアドレスはともに 192.168.0.1 であるものとする．

端末 A から端末 B へ接続を開始する場合の通信の流れを図 8 に示す．初期設定として NATFBOX1 には通信先ホスト名(h)とドメイン名(natf.com)を，NATFBOX2 には内部に属する端末のホスト名(h)とプライベート IP アドレス(192.168.0.1)を設定する．

端末 A は端末 B に関する DNS 問合せを行う．このパケットが NATFBOX1 に届いたら NAT 処理後に問合せ部のホスト名(h)+ドメイン名（natf.com）からホスト名を削除し DNS に送信する．DNS は，A レコードとして NATFBOX2 のグローバルアドレス（2.2.2.2）を返す．

NATFBOX1 は DNS 応答を受信すると，NATFBOX2 の IP アドレスを保存する．そして DNS 応答の問合せ部を元のホスト名（h）とドメイン名（natf.com）に戻し，端末 A へ転送する．端末 A は通信相手が NATFBOX2 であるものと認識し通信を開始する．

以下 IP アドレスとポート番号の関係を，順を追って示す．斜線（/）の左側は送信元 IP アドレス：送信元ポート番号，右側は宛先 IP アドレス：宛先ポート番号である．端末 A 側の OS から動的に割り当てられた送信元ポート番号を a，端末 B を HTTP サーバと仮定し，宛先ポート番号を 80 とすると，端末 A が送信するパケットは次のようになる．

192.168.0.1：a / 2.2.2.2：80

NATFBOX1 はこのパケットを受信すると, NATF プロトコルを実行するための準備を行う. NATFBOX1 は一般の NAT の手順に従い NAT テーブル 1 を生成し，最初のパケットのアドレス変換を行っておく．NAT テーブル 1 の情報は以下のように生成される．b は NAT により動的に割り当てられたポート番号である（図 11）.

{192.168.0.1：a⇔1.1.1.1：b}

したがって，アドレス変換後のパケットは

1.1.1.1：b / 2.2.2.2：80

となる．次に NATF プロトコルを実行するため，このパケットは NATFBOX1

に一時的に待避しておき,ポート番号報告指示パケットを NATFBOX2 に送る．このパケットには，送信元 IP アドレス：ポート番号（1.1.1.1：ｂ）と宛先 IP アドレス：ポート番号（192.168.0.1：80）とプロトコル（TCP）とホスト名(h)の情報が含まれる．

NATFBOX2 は上記ポート番号報告指示パケットを受信したら，このパケットの情報と予め登録した情報をもとに NAT テーブル 2 を生成する．テーブルの内容は以下の通りである．ｘは NAT により動的に割り当てられたポート番号である（図 12)．

｛192.168.0.1：80⇔2.2.2.2：ｘ｝

NATFBOX2 はこの変換ポート番号（ｘ）を，ポート番号報告応答パケットを用いて NATFBOX1 に送る．

NATFBOX1 は変換ポート番号の情報を元に，FAT テーブルを生成する．FAT テーブルの情報は以下の通りである（図 9)．

｛1.1.1.1：a→2.2.2.2：80　⇔　1.1.1.1：a→2.2.2.2：ｘ｝

ポート番号変換テーブル作成後，NATF プロトコルは終了する．

NATF プロトコル終了後の通信の流れを図 8 に示す．NATFBOX1 は，待避していたパケットを NAT テーブルにより，アドレス/ポート番号変換後，ポート番号変換テーブルより更にポート変換し，NATFBOX2 宛に送信する．このパケットの内容は次のとおりである．

1.1.1.1：b / 2.2.2.2：ｘ

このパケットを受信した NATFBOX2 では，NAT アドレス変換後，端末 B に送信する．パケットの内容は以下のように変わる．

1.1.1.1：b / 192.168.0.1：80

端末 B から端末 A への応答は，上記と逆の変換で行われる．

以下，NATFBOX1 では NAT アドレス変換とポート番号変換，NATFBOX2 では NAT アドレス変換をすることで，通信を行う

このように，NATFBOX はアドレス変換用のポート番号をそれぞれ独立して生成し，送信側の NATFBOX が 2 つのポート番号の変換を行う．端末 A は NATFBOX2 と，端末 B は NATFBOX1 と通信しているようにみえる．

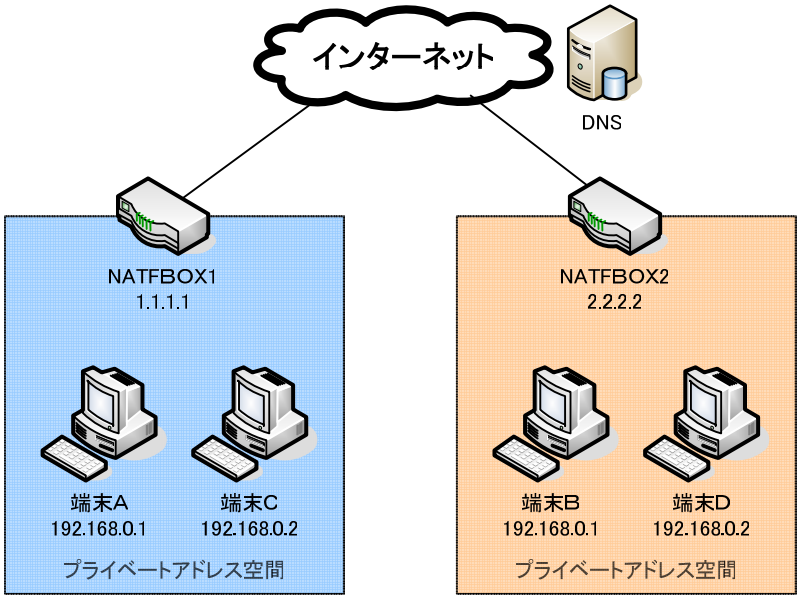
インターネット
DNS
NATFBOX1
1.1.1.1
NATFBOX2
2.2.2.2
端末A
192.168.0.1
端末C
192.168.0.2
プライベートアドレス空間
端末B
192.168.0.1
端末D
192.168.0.2
プライベートアドレス空間

図 7　CIPA の環境

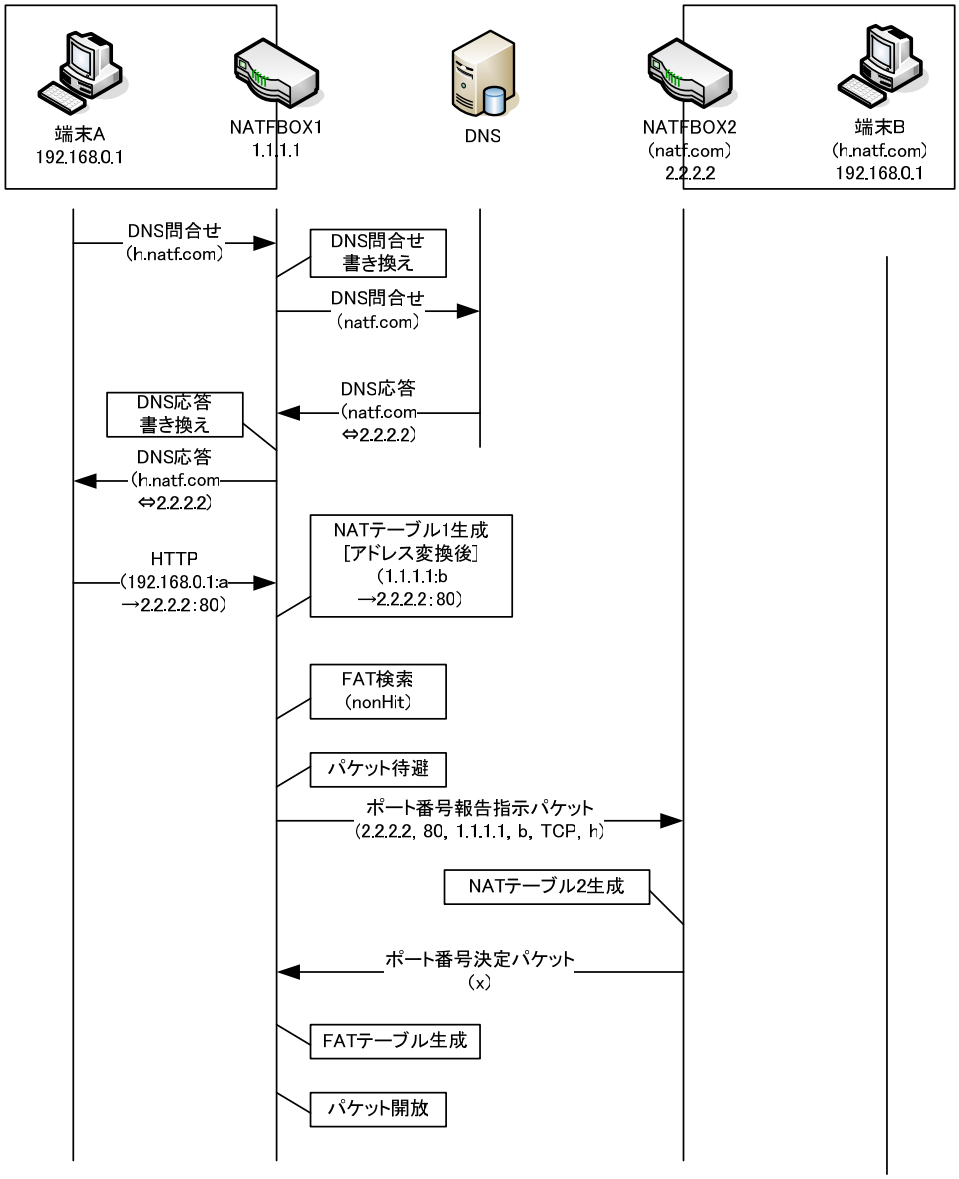

図 8　CIPA 時の NATF プロトコルの流れ

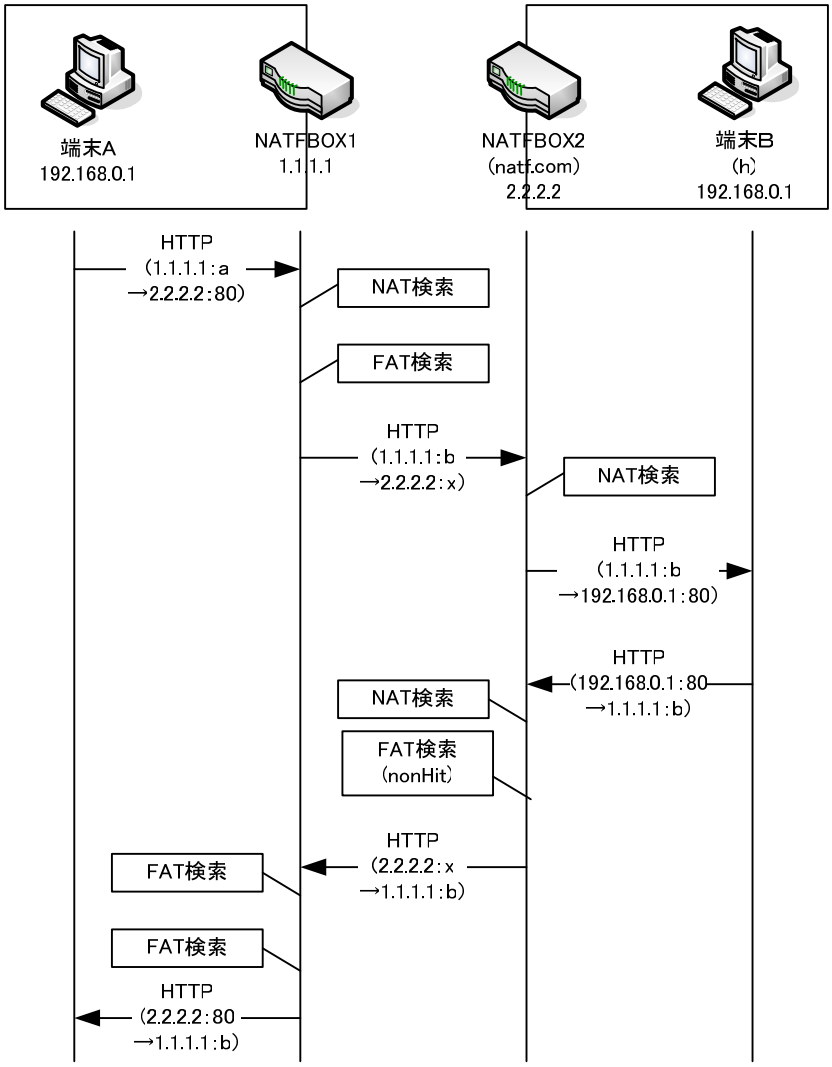

図 9　CIPA 時の NATF プロトコル後の流れ

FATテーブル1					
[in]	sIP	dIP	sPort	dPort	proto
	2.2.2.2	1.1.1.1	x	b	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	2.2.2.2	1.1.1.1	80	b	
[out]	sIP	dIP	sPort	dPort	proto
	1.1.1.1	2.2.2.2	b	80	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	1.1.1.1	2.2.2.2	b	x	

図 10　FAT テーブル 1

NATテーブル1					
[in]	sIP	dIP	sPort	dPort	proto
	2.2.2.2	1.1.1.1	80	b	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	2.2.2.2	192.168.0.1	80	a	
[out]	sIP	dIP	sPort	dPort	proto
	192.168.0.1	2.2.2.2	a	80	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	1.1.1.1	2.2.2.2	b	80	

図 11　NAT テーブル 1

NATテーブル2					
[in]	sIP	dIP	sPort	dPort	proto
	1.1.1.1	2.2.2.2	b	x	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	1.1.1.1	192.168.0.1	b	80	
[out]	sIP	dIP	sPort	dPort	proto
	192.168.0.1	1.1.1.1	80	b	tcp
⇔	t_sIP	t_dIP	t_sPort	t_dPort	
	2.2.2.2	1.1.1.1	x	b	

図 12　NAT テーブル 2

5. 実装

5.1 実装の概要

試作システムは，IP 層の詳細な処理フローに関する情報が多い FreeBSD を採用した．NATFBOX のモジュール構成を図 13 に示す．FreeBSD カーネル内に，NATF モジュールを組み込む．IP 層の ip_input，ip_output から NATF モジュールに処理を渡し，処理を終えたら差し戻す．既存の処理に一切の変更を加えない．NATF で使われるテーブルはハッシュテーブルとして実装する．natd は NAT をデーモンで動かすモジュールである．natd のアドレス変換はアプリケーション層で行われるが，NATF のポート番号変換は IP 層で行う．

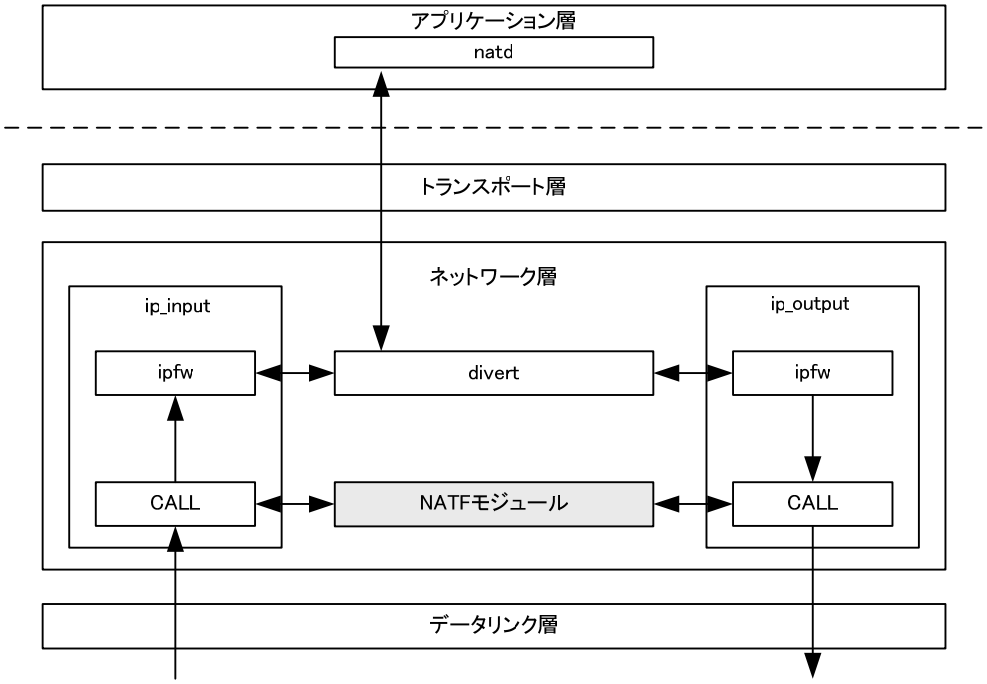

図 13　NATFBOX のモジュール構成

5.2　モジュールの機能

NATF は初期設定モジュール，ネゴシエーションモジュール，FAT 操作モジュール，ポート番号変換モジュールで構成される（図 14）.

初期設定モジュールとは DNS 問合せ/応答時に IP アドレスなどの情報を追加，検索，削除などを行うモジュールである．初期設定モジュールの詳細を 5.3 章で説明する.

ネゴシエーションモジュールとは，ポート番号報告指示パケット，ポート番号登録パケット，疑似パケットに関するモジュールである．疑似パケットとは通信相手である内部の PA 空間にいる端末から外部へ通信すると見せかけ NAT テーブルを生成させるパケットである.ポート番号報告指示パケットを受け取った NATFBOX はその情報を元に疑似パケットを生成する．疑似パケットが natd を通りテーブル作成後 ip_output で破棄し，NAT テーブルのポート番号の情報をポート番号登録パケットにのせて送信する．疑似パケットの詳細を 5.4 章で説明する.

FAT 操作モジュールとは， FAT テーブルを生成，検索，削除するモジュールである.

ポート番号変換モジュールとは．通信時に FAT テーブルを検索し，ポート番号の書き換えを行うモジュールである.

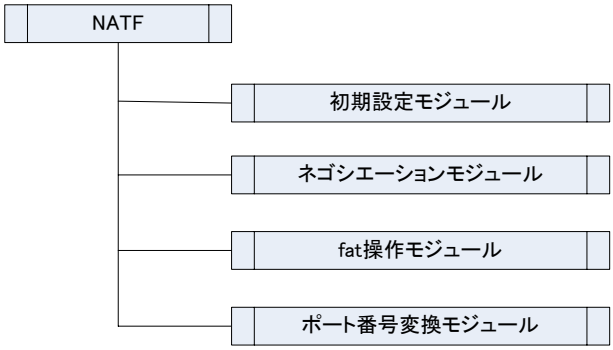

図 14　NATF モジュール

5.3 NATFBOX への初期登録と DNS 問合せ/応答時の処理

初期設定時に登録するテーブルと DNS 問合せ/応答時に作られるテーブルの内容について説明する．

送信元の NATFBOX の初期設定は NRDB（Name Resolution Data Base）と呼ぶテーブルに登録する．NRDB の詳細を図 16 に示す．NRDB は通信先のホスト名とドメイン名を合わせた FQDN のハッシュをキーとして FQDN とホスト名を登録する．

宛先の NATFBOX の初期設定は APDB（Access Permission Data Base）と呼ぶテーブルに登録する．APDB の詳細を図 19 に示す．APDB はホスト名のハッシュをキーとしてホスト名と IP アドレスを登録する．

DNS 問合せ時には RQT（Resolution Query Table）と呼ぶテーブルが生成される．RQT の詳細を図 17 に示す．RQT は DNS の Transaction ID のハッシュをキーとして Transaction ID と FQDN のハッシュを登録する．この時の FQDN のハッシュは NRDB の Hash No と一致する．

DNS 応答時には RRT（Resolution Response Table）と呼ぶテーブルが生成される．RRT の詳細を図 18 に示す．RRT は FQDN のハッシュをキーとして IP アドレスと Transaction ID のハッシュを登録する．この時の Transaction ID のハッシュは NRDB の Hash No と一致する．

図 15 に NRDB の検索，RRT と RQT の生成・検索の流れを示す．端末 A は端末 B に関する DNS 問合せを行う．このパケットが NATFBOX1 に届いたら問合せ部のホスト名(h)+ドメイン名(natf.com)のハッシュを元に NRDB 検索を行う．一致する情報があれば DNS の Transaction ID のハッシュ(200)をキーとして RQT に Transaction ID と FQDN のハッシュ（400）の登録を行う．ここでは Transaction ID を 300 とする．次に問合せ部からホスト名を削除し DNS に転送する．DNS は，A レコードとして NATFBOX2 のグローバルアドレス（2.2.2.2）を返す．

NATFBOX1 が DNS 応答を受信すると，NATFBOX1 の IP 層から上位層へ渡す際,回答部の IP アドレスのハッシュ(400)をキーとして IP アドレス(2.2.2.2)と Transaction ID のハッシュ（300）を RRT に作成する．Transaction ID のハッシュを用いて RQT 検索を行い FQDN のハッシュ（100）を取得する．FQDN のハッシュを用いて NRDB 検索を行い，ホスト名（h）を取得する．DNS 応答の問合せ部にこのホスト名を追加し元に戻してから，DNS 応答を端末 A へ転送する．端末 A は DNS 応答を受け取ると通信相手を NATFBOX2 と見なして通信を開始する．

通信開始後の第一パケットが NATFBOX1 に届いたら NAT 変換を行う．FAT 検索を行い一致する情報がなければ FAT を生成するために NATF プロトコル

を開始する．まず，第一パケットを待避し宛先 IP アドレスをハッシュ（400）キーとして RRT 検索を行い Transaction ID のハッシュ（200）を取得する．Transaction ID のハッシュを用いて RRT 検索を行い FQDN のハッシュ（100）を取得する．FQDN のハッシュを用いて NRDB 検索を行い，ホスト名（h）を取得する．このホスト名と通信開始時の宛先アドレス・ポート番号，送信元アドレス・ポート番号，プロトコル情報をポート番号指示パケットに載せて NATFBOX2 に送信する．

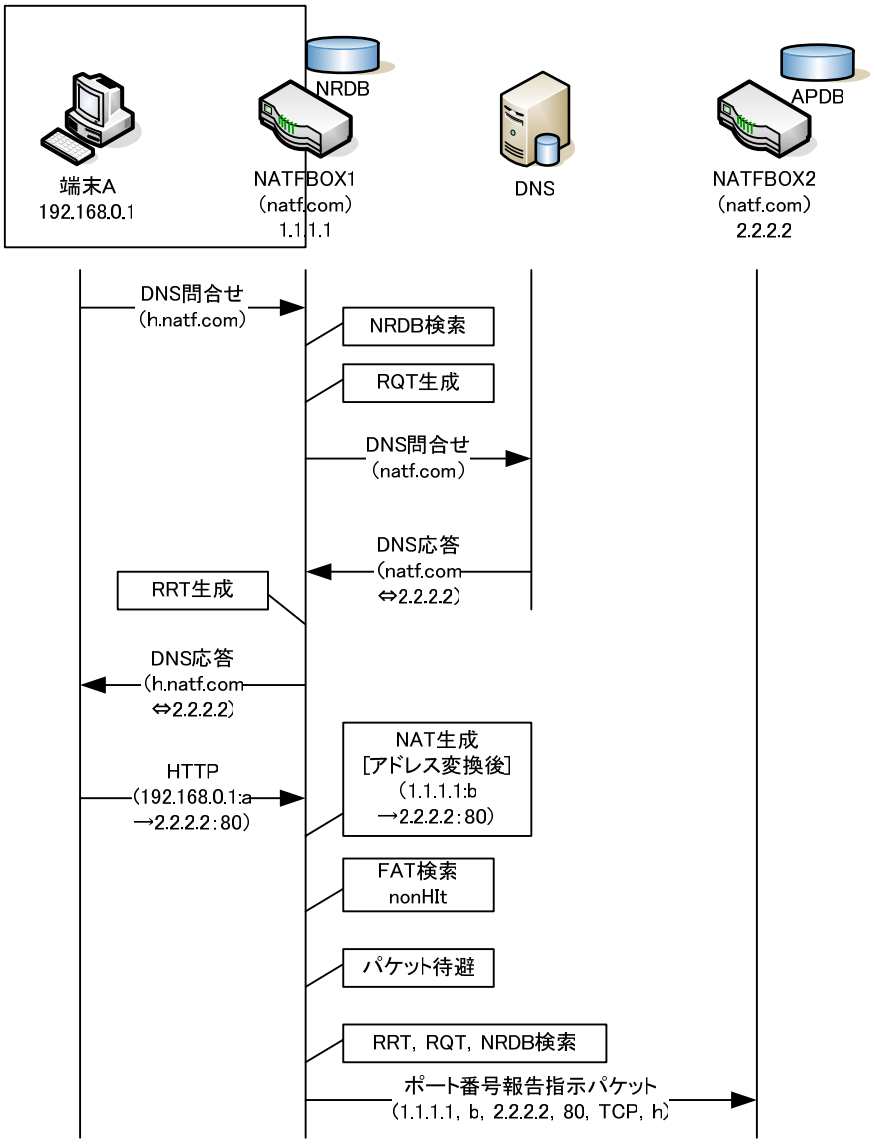

図 15　DNS 問合せからポート番号報告指示パケットまでの流れ

NRDB		
Hash No(FQDN hash)	FQDN	Host name
100	h.natf.com	h

図 16　NRDB

RQT		
Hash No.(Transaction ID hash)	Transaction ID	NRDB No.(FQDN hash)
200	300	100

図 17　RQT

RRT		
Hash No(IP address hash)	IP Address	RQT No.(Transaction ID hash)
400	2.2.2.2	200

図 18　RRT

APDB		
Hash No(Host name hash)	Host name	IP Address
500	h	192.168.0.1

図 19　APDB

5.4　疑似パケット

疑似パケットとは，通信相手である内部の PA 空間にいる端末が外部へ通信開始すると見せかけ，NATFBOX に NAT テーブルを強制的に生成させるパケットである．

疑似パケットの流れを図 20 に示す．まず，NATFBOX2 がポート番号報告指示パケットを受信する．ポート番号報告指示パケットの内容は，送信元アドレス（1.1.1.1），ポート番号（b），宛先アドレス（2.2.2.2），ポート番号（80），プロトコル（TCP），ホスト名（h）である．ホスト名より APDB 検索を行い，IP アドレス（192.168.0.1）を取得する．この IP アドレスとポート番号報告指示パケットの情報を元に疑似パケットを生成する．疑似パケットの内容は以下の通りである．

192.168.0.1：80　/ 1.1.1.1：b

このパケットに対して NAT 処理を実行させることにより，NAT テーブルが生成される．NAT テーブルの内容は以下の通りである（図 12）．

｛192.168.0.1：80⇔2.2.2.2： x ｝

NAT テーブル生成後，疑似パケットは破棄し，変換されたポート番号の情報 x をポート番号登録パケットに乗せて NATFBOX1 に返信する．

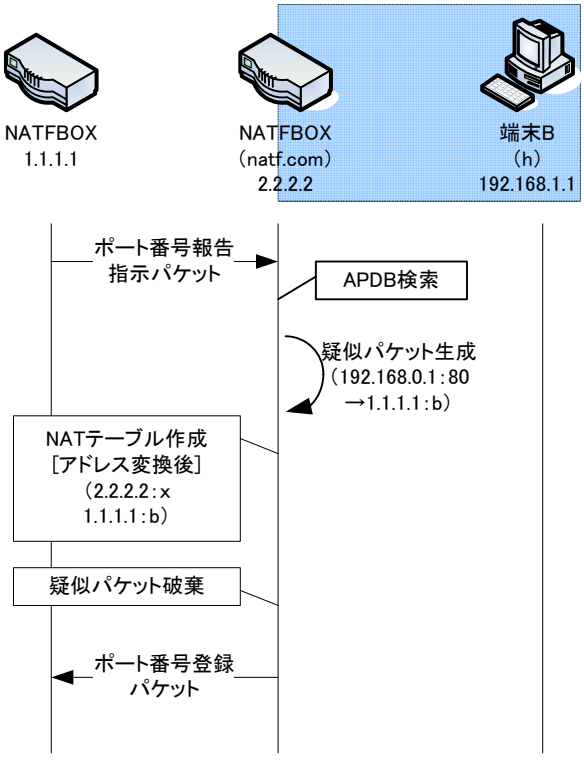

図 20　疑似パケットによる NAT テーブル生成の流れ

6. 評価実験

CIPA の試作システムとポートフォワーディングを用いたシステムの通信性能を測定し比較を行った．試験環境と端末の仕様を図 21 に示す．CIPA では DNS 問合せ・応答処理を行なった後の速度を計測した．

まず FTP を用いて 100Mbyte のファイルをダウンロードし転送時間を測定した．次に Netperf の UDP 測定機能を用いて 16Mbyte のファイルを送受信しスループットを測定した．ただし，Netperf では送受信するたびにポート番号が変更されるため，ポートフォワーディングでの測定はできない．測定結果は FTP，Netperf とも 20 回試行の平均値である．

測定結果を図 22 に示す．FTP を用いた CIPA の転送時間をスループットに変換すると約 61.1Mbps，ポートフォワーディングは約 62.1Mbps となり，CIPA はポートフォワーディングの約 98.5%のスループットであった．低下のの理由は NATFBOX1 における FAT 検索によるオーバヘッドが影響したものと考えられる．

次に Netperf を用いた UDP 測定では約 77.7Mbps のスループットであった．参考までに端末 AB 間を直接繋いだときの Netperf のスループットは約 78.0Mbps であった．

この結果より，CIPA が通信に与える影響はポートフォワーディング時とほとんど変わらないことを実証できた．

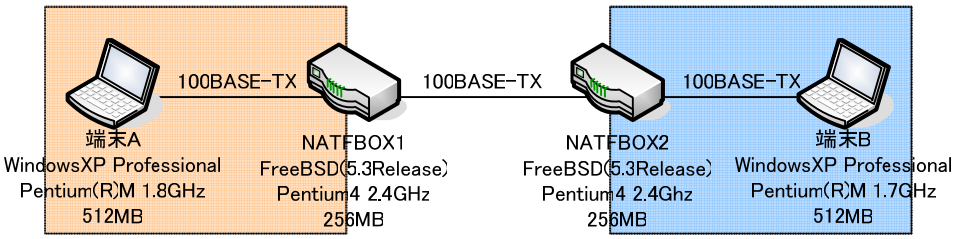

図 21　試験環境

	CIPA	ポートフォワーディング
FTP (TCP) (100M)	61.143Mbps	62.054Mbps
Netperf (UDP) (16M)	77.661Mbps	−

図 22　計測結果

7. おわりに

本稿では，NATF を拡張して，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方式 CIPA の提案を行った．CIPA は NATFBOX どうしが，通信開始に先立って情報を事前に交換し，その情報を元に通信パケットの IP アドレス変換，ポート番号変換をすることで通信を可能としている．CIPA ではカプセル化／デカプセル化やパケット長を変化させる必要がなくオーバヘッドが少ない．また，異なるプライベートアドレス空間に同一のプライベート IP アドレスを持つ端末があっても CIPA を利用することにより P2P 通信が可能であることを示した．そして CIPA の実装を行い，NAT を越えて通信できることを確認した．FTP を用いた性能測定ではポートフォワーディングと比べてオーバヘッドが極めて少ないことを確認した．

今後の課題は，IPv6 時や IPv4 と IPv6 が混合した環境で CIPA が適用できるよう検討していく．

謝辞

本研究にあたってご指導を頂きました名城大学大学院理工学研究科　渡邊晃教授には心から感謝致します．研究方法の初歩から，研究の内容，展開，論文の執筆に至るまで丁寧にご指導いただきました．また研究活動のみならず，各種活動の機会を与えて下さったことで，修士課程の 2 年間を有意義に過ごすことができました．本当にありがとうございました．

本研究をすすめるにあたり，研究内容に関して終始御熱心な御指導と御教示を賜わりました，名城大学大学院理工学研究科　小川明教授，山本新教授，宇佐見庄五講師に心より厚く御礼申し上げます．

渡邊研究室の皆様に心より深く感謝いたします．

最後に，ここには書ききれなかった方々を含め，学生生活の中でお世話になった全ての人たちに心より深く感謝いたします．

参考文献

[1] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, “Address Allocation for Private Internets”, RFC1918, February 1996

[2] S. Deering, R. Hinden, “Internet Protocol, Version6 (IPv6) specification”, RFC2460, December 1998

[3] P. Srisuresh, M. Holdrege, “IP Network Address Translator (NAT) Terminology and Considerations”, RFC2663, August 1999

[4] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, RFC3489, March 2003

[5] T.S.Eugene Ng, I.Stoica, H.Zhang, “A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces”,USENIX 2001

[6] SoftEther, http://www.softether.com/

[7] Bryan Ford, Pyda Srisuresh, Dan Kegel, “Peer-to-Peer Communication Across Network Address Translators”, USENIX 2005

[8] Z.turanyi, A.VAlko “IPv4+4”, ICNP 2002

[9] Kuniaki Kondo, “Capsulated Network Address Translation with Sub-Address(C-ＮＡＴS)”, Internet Draft, draft-kuniaki-cap sulated-ＮＡＴS-03.txt, December 2002

[10] P. Mockapetris , “ DOMAIN NAMES-IMPLEMEN-TATION AND SPECIFICATION”, RFC1035, November 1987

[11] S. Thomson, Y. Rekhter, J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE)”, RFC2136, April 1997

研究業績

1)柳沢信成, 渡邊晃,“DDNS を利用したターゲットの位置情報表示システム”，電気関係学会東海支部連合大会，Oct. 2003.

2)柳沢信成, 渡邊晃，“DDNS を利用したターゲットの位置情報表示システム”，情報処理学会　第 66 回全国大会，Mar.2004.

3)柳沢信成，渡邊晃，“グローバルアドレスをはさんだプライベートアドレス端末同士の通信”，WiNF2004 論文集，Vol.2，pp.217-221，Sep. 2004.

4)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，2005-DPS-122 ， pp.357-362 ， Mar.2005.

5)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どうしの通信方式 CIPA の提案”, DICOMO2005 シンポジウム論文集, Vol.2005, No.6，pp.369-372，Jul.2005.

6) 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， 2005-DPS-122.

7) 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",DICOMO2005 シンポジウム論文集，Vol.2005，No.6，pp.373-376，Jul.2005.

8) 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"，WiNF2005 論文集，pp.142-146，Sep.2005.

付録

1. 同時通信問題

NATF では複数台通信する際，同時に DNS 問合せと行なったとき問題が生じる可能性がある．その時の環境を図 23，シーケンスを図 24，初期設定の NRDB を図 25 に示す．具体的にいうと，PC1 が Host2 と Host3 へ同時にアクセスしようとした際に起きる．

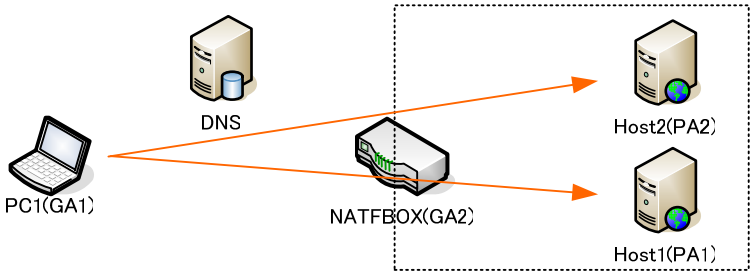

図 23　通信環境

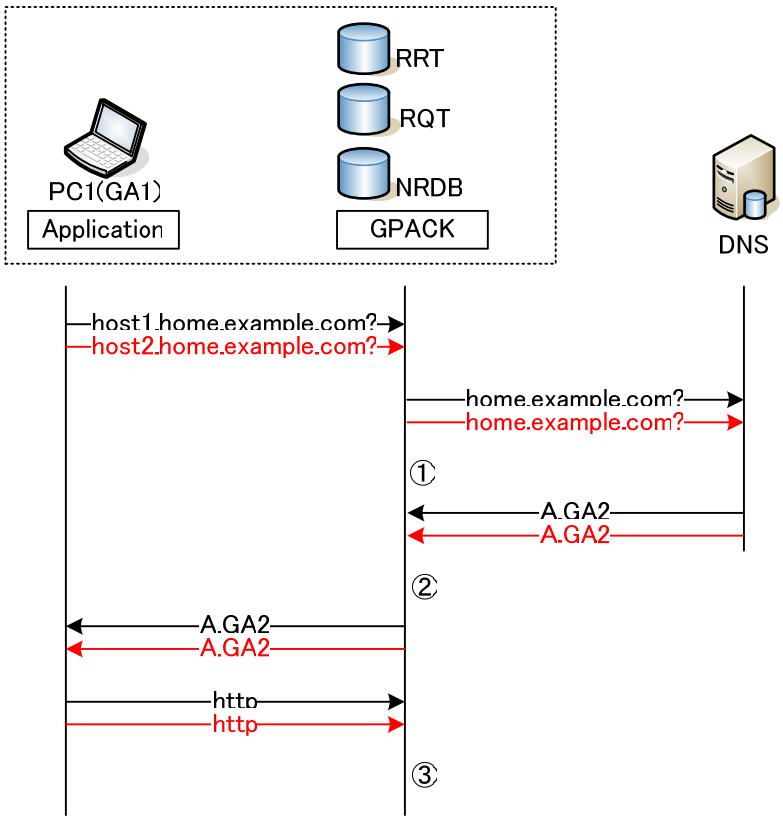

図 24　シーケンス

初期設定
<NRDB>

hash No	Host Name	Domain Name
1000	host1	home.example.com
1050	host2	home.example.com

図 25　NRDB

① DNS 問合わせ処理と RQT 作成

DNS 問い合わせを GPACK で監視し，問合せ FQDN のハッシュ値をキーとして NRDB を検索する．一致すれば Host Name 部分を削除して Domain Name と一致するよう書き換える(図 26).

次に，DNS Transaction ID のハッシュ値をキーとして RQT の作成を行う．

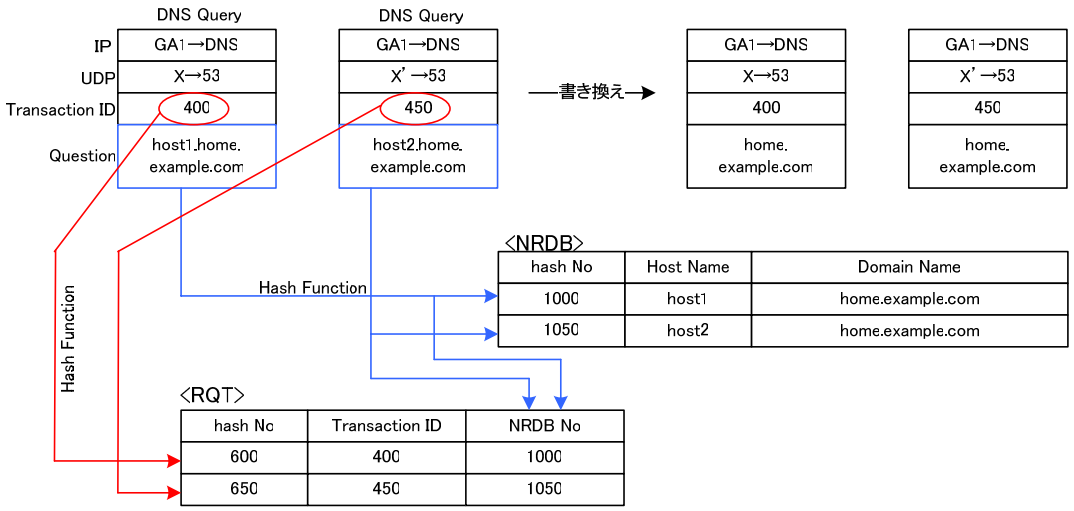

図 26　RQT 作成

② RRT の作成

DNS 応答より取得した IP アドレス（GA2）のハッシュ値をキーとして RRT を作成する．このとき，同時通信だと同じ値のハッシュ値のキーができてしまう．（図 27）

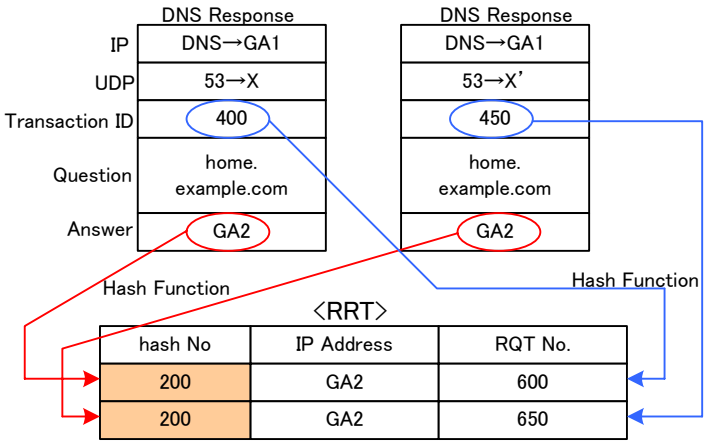

図 27　RRT 作成

③ NATF ネゴシエーションの実行

宛先アドレスのハッシュ値をキーとして RRT 検索を行うが，ハッシュが同じ値なので DPRP が正常に行われない（図 28）.

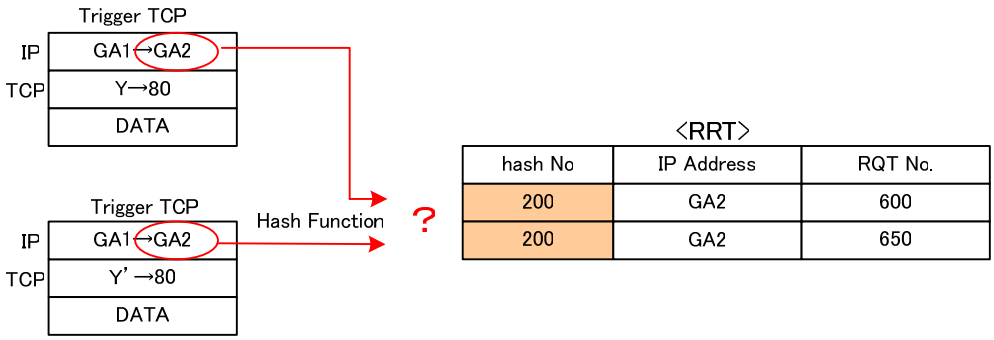

図 28　RRT 検索

解決策として DNS 応答を受信し RRT を登録する時に同じ IP アドレスだった場合，後から来た DNS 応答を破棄し，一台が NATF ネゴシエーションを終わった後，もう一台が DNS 再問合せで NATF ネゴシエーションを行なう.

2. 既存技術

2.1 ポートフォワーディング

ポートフォワーディングとは,事前に静的にテーブル内容を登録しておくことで外部から NAT へ来た特定のポートへのアクセスを，内部のホストのへのアクセスとしてフォワードするものである（図 29）．例えば，プライベートネットワーク内にあるサーバに外部からアクセスしたいという状況を考える．NAT において，外部からのアクセスを，内部サーバにポートフォワーディングすることにより，外部からも内部サーバにアクセスすることができる．

ポートフォワーディングは一般によく使われる方法であるが,ネットワーク構成の変化やアプリケーションごとにテーブルを設定し直さなければならず,自由な通信とは言えない．

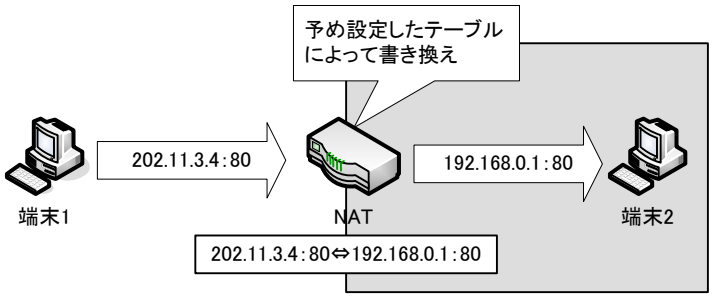

図 29 ポートフォワーディング

2.2 IPv4+4

IP アドレスは，IPv4 では 32bit だが，IPv4+4 ではさらにその下に 32bit 追加する（図 30）．例えば，ゲートウェイのグローバル IP が GA で，ホストのプライベート IP が PA であった場合，GA.PA と表す．この方式は，IPv4 ヘッダのカプセル化によって実現する．これによってグローバル IP アドレスとプライベート IP アドレスが含まれる．NAT ではこのヘッダを見て，NAT 通過時に GA と PA の IP アドレスを書き換えてやることによって通信が可能となる（図 31）．ポート番号を気にせず通信が行なえるが，問題点として全てのホストと NAT を改良しなければならない．また DNS には A レコードに GA と PA の追記をしなければならない．

<table>
<tr><td>Ver</td><td>Hlen</td><td>DS byre</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flag</td><td>Fragment offset</td></tr>
<tr><td>TTL</td><td colspan="2">Protocol 1</td><td colspan="2">Header Checsum 1</td></tr>
<tr><td colspan="5">Source Address 1</td></tr>
<tr><td colspan="5">Destination Address 1</td></tr>
<tr><td colspan="5">Source Address 2</td></tr>
<tr><td colspan="5">Destination Address 2</td></tr>
<tr><td>Protocol 1</td><td colspan="2">SPos</td><td>Dpos</td><td>Header Checsum 2</td></tr>
</table>

図 30　IPv4+4 のパケットフォーマット

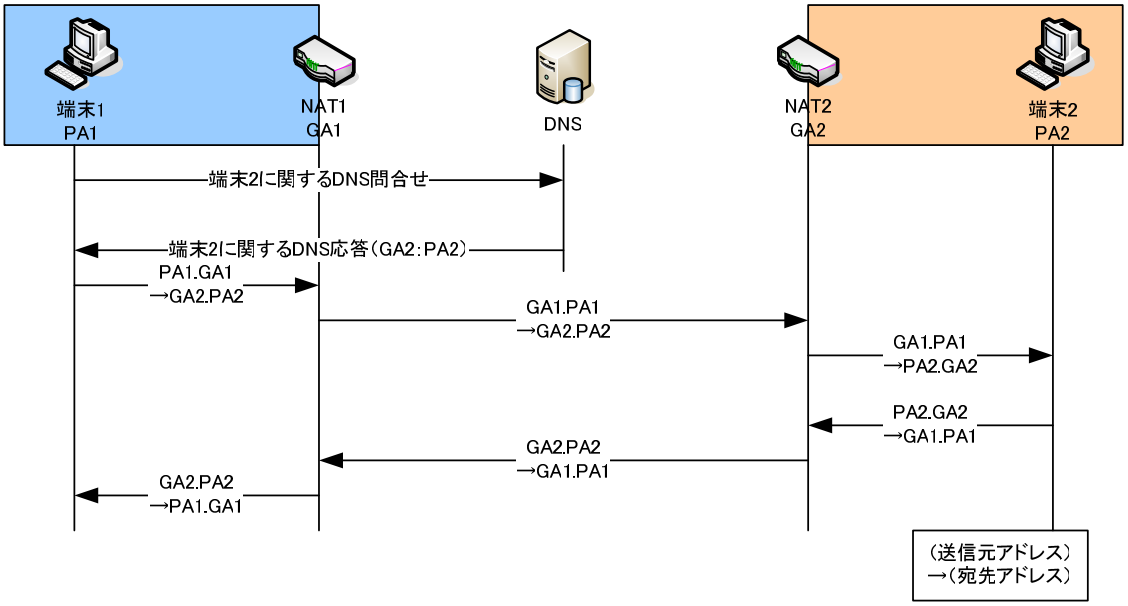

図 31　IPv4+4 の流れ

2.3 STUN(Simple Traversal of UDP through NATs)

この方式は UDP Hole Punching を用い NAT を通過する方式である．UDP Hole Punching とは，UDP を用いてあらかじめ内部から外向きに用意した通路を使って、外部から内向きの通信を行う方式である（図 32）．UDP Hole Punching とグローバル空間上に STUN サーバを用い NAT を超えた通信を実現している．通信の流れを図 33 に示す．端末 1 が端末 2 へ通信を行なう場合，端末 1 は STUN サーバへ端末 2 が属している NAT がどのポート番号が空いているか問い合わせる．そして STUN サーバから受け取ったポート番号の情報を元に端末 2 と通信を行なう．この方式の利点としては NAT や端末をそのまま使える点である．しかし，プロトコルは UDP，また NAT は Cone 型 NAT しか使えず，別途専用サーバが必要である．

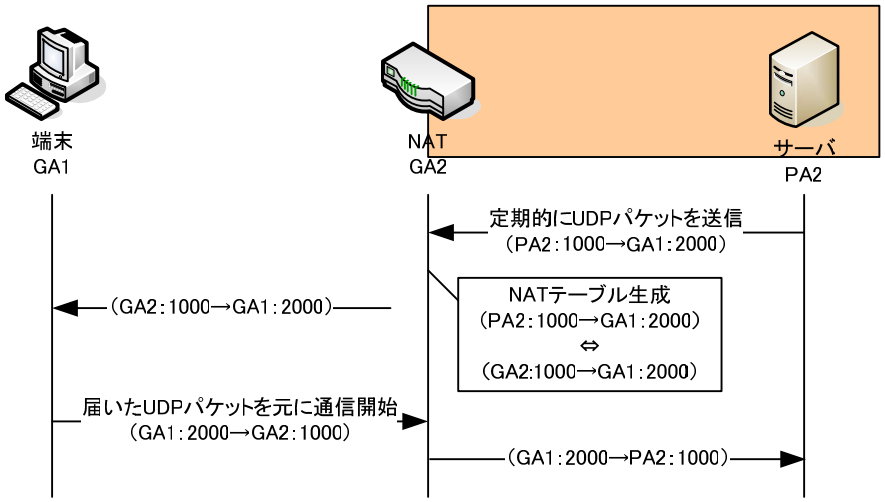

図 32　UDP Hole Punching

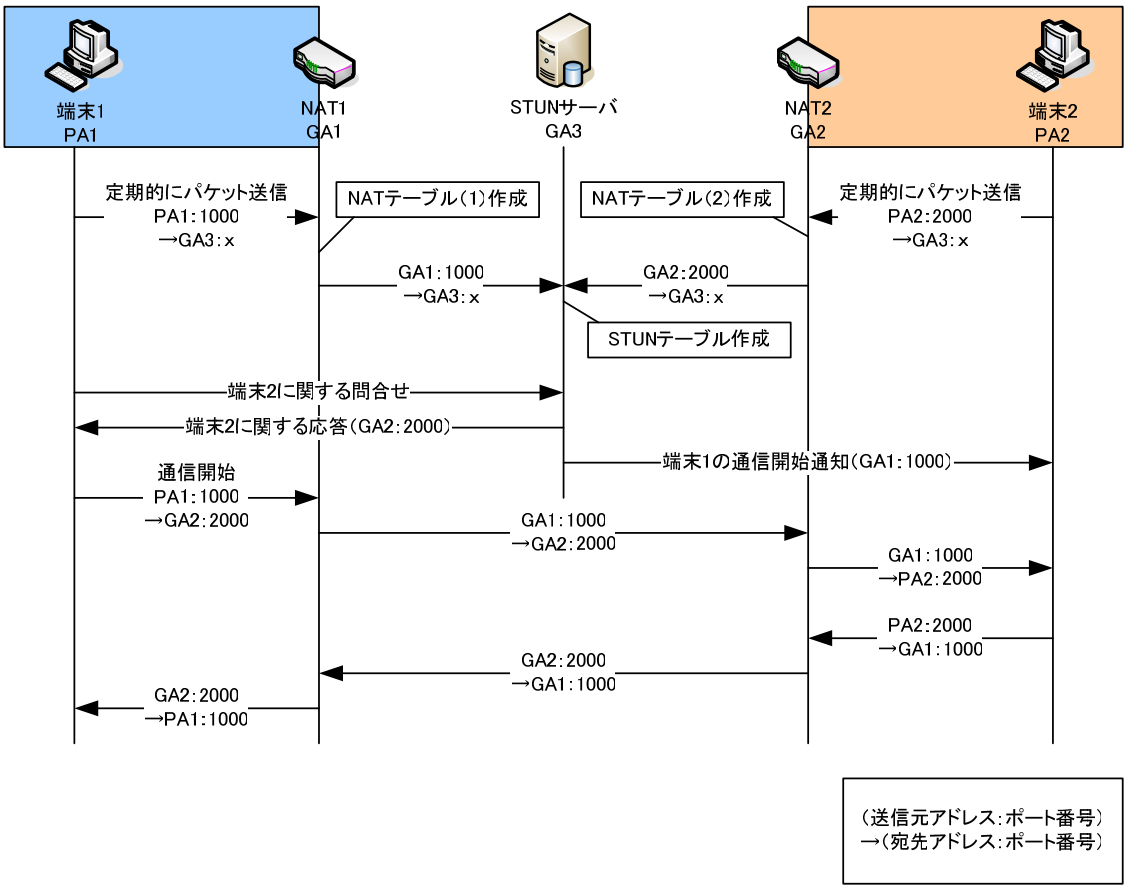

図 33　STUN の流れ

NATテーブル(1)	
変換前 送信元アドレス:ポート番号	変換後 送信元アドレス:ポート番号
PA1:1000	GA1:1000

図 34　NAT テーブル 1

NATテーブル(2)	
変換前 送信元アドレス:ポート番号	変換後 送信元アドレス:ポート番号
PA2:2000	GA2:2000

図 35　NAT テーブル 2

STUNテーブル	
ホスト名	IPアドレス:ポート番号
端末1	GA1:1000
端末2	GA2:2000

図 36　STUN テーブル

2.4 AVES (Address Virtualization Enabling Service)

AVES とはヘテロジーニアスな環境での通信方式である．ヘテロジーニアスな環境とは，プライベート IP アドレスと，グローバル IP アドレス，および IPv6 ホストが存在する空間のことを指している．そして，これらを相互に接続するにはどうすればよいかを考える．代表的な問題として，グローバル IP アドレスのホストからプライベート IP アドレスのホストへの接続がある．これが可能になれば，他も同様にして可能になる．AVES では，waypoint と，AVES-awareNAT（NAT を改良），AVES-awareDNS を必要とする．ただし，DNS は既存のものでも，性能を落とすが可能である．通信の流れを図 37 に示す．まず，プライベート IP ホストはそのホスト名を DNS に登録する．これによって，IP ホストからプライベート IP ホストへの指定が可能となる．そのため AVES-awareDNS では，waypoint，AVES-awareNAT，および B を知っていることとなる．A が B のホスト名を指定すると，AVES-awareDNS が waypoint，送信元アドレス，AVES-awareNAT の IP を組にして，waypoint に送り，セッションを張ることが可能か確認する．可能な場合は，waypoint のアドレスをホストに返し，ホストは waypoint→NAT→B というように通信を行うことが出来る．また，応答パケットは NAT→A と返ってくる．課題として waypoint に割り当てる IP アドレスの数によりプライベートホストへの同時接続数が制限される．また，三角経路を通り通信が冗長することと，waypoint と NAT 間でカプセル化を行なわなければならない．

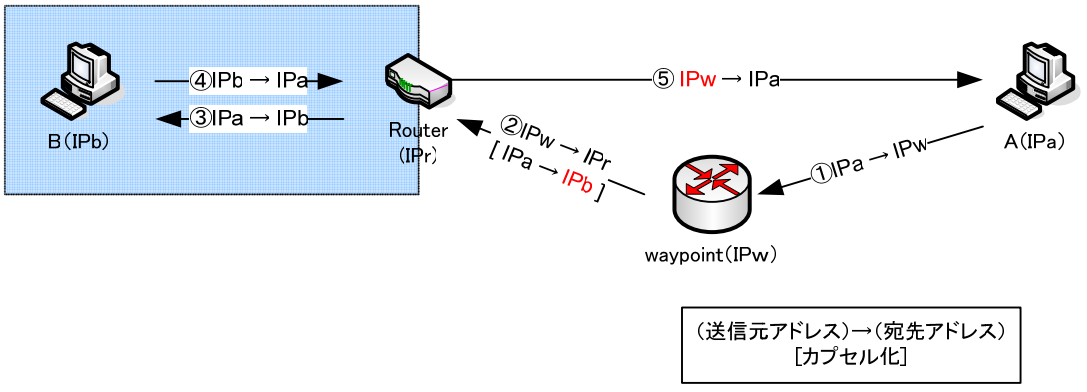

図 37　AVES の流れ

2.5 SoftEther

ソフトウェアとして実装された仮想的なスイッチングハブと仮想的な LAN カードにより，TCP/IP ネットワーク上に仮想的な Ethernet ネットワークを構築する．つまり物理的に離れたコンピュータ同士で Ethernet LAN を形成することができる（図 38）.

仮想 LAN カードはデバイスドライバになっており, OS やアプリケーションからは通常の LAN カードと同じように認識され，特別な対応は必要ない．また，仮想ハブも同様に通常のハブと同じように認識され、利用できる.

SoftEther 上の TCP/IP 通信は，既存の TCP/IP ネットワークの管理者が監視したり遮断したりすることができない. よって別途セキュリティ対策が必要である．またアドレス環境の統一管理が必要である.

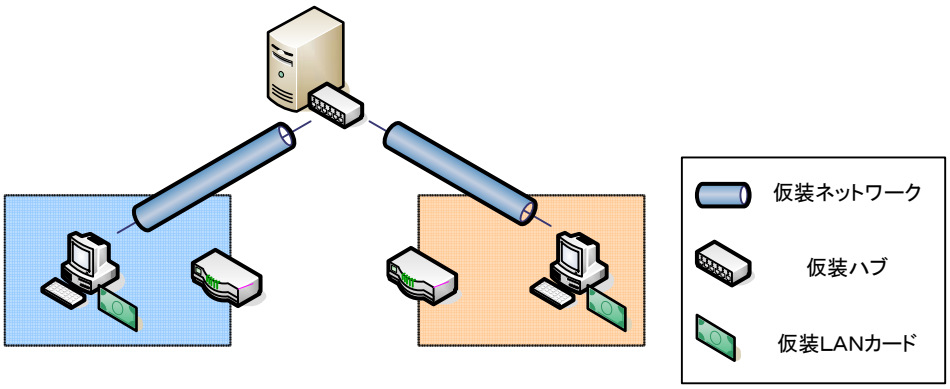

図 38　SoftEther

3. 応用例

提案方式の適用事例として位置情報取得システムが考えられる．このシステムではインターネットを用いてターゲットの位置情報を別な場所から常に把握することができる． このシステムは NATFBOX 配下にいる端末が自由にグローバルアドレス空間,プライベートアドレス空間に移動しても通信が行なえるという前提で説明する．図 39 に位置情報取得システムの構成を示す．監視端末はターゲットの位置を知りたいユーザが保持する端末である.ターゲットは，子供などが保持する小型端末で，将来的にはより小型になり常時保持が可能になることを想定する．ターゲットの位置情報の取得は GPS から取得する．DDNS（Dynamic DNS）はターゲットのホスト名と IP アドレスの関係をダイナミックに管理する装置である．

システムの動作は以下の通りで，図中の番号と説明の番号は同様の動作を示す．

①ターゲットが移動して IP アドレスが変化すると,ターゲットは新しい IP アドレスを DDNS サーバに通知する．

②監視端末はターゲットの位置情報を知りたいとき, DDNS サーバに対してターゲットのホスト名を基にターゲットの IP アドレスを問い合わせる．

③DDNS サーバが監視端末へターゲットの IP アドレスを渡す．

④監視端末は獲得した IP アドレスによって，ターゲットに対して位置情報を要求する．

⑤ターゲットは監視端末へ位置情報を返す．

⑥監視端末はターゲットの位置を画面に表示する．

このシステムはインターネットを用いてターゲットの位置を把握できるのが特徴であるが，図 40 のように監視者はプライベートアドレス空間におり，ターゲットは別の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移動する可能性がある.本稿の提案方式と組み合わせることによりターゲットがどのように移動しても位置情報を取得することができる．

もし監視端末とターゲットの両者に NATF 機能を実装すれば，監視端末，ターゲットともどの空間に移動しても位置情報を取得できる．ただし，プライバシの確保が重要であり，認証技術との組合せも必要となる．

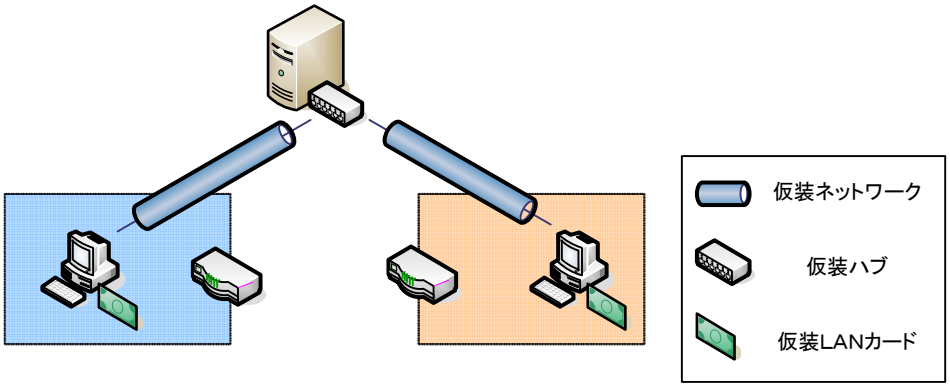

図 39　位置情報取得システムの構成

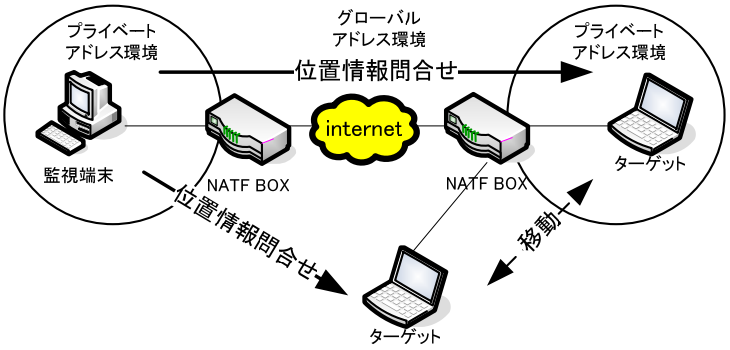

図 40　CIPA と位置情報取得システムの組み合わせ

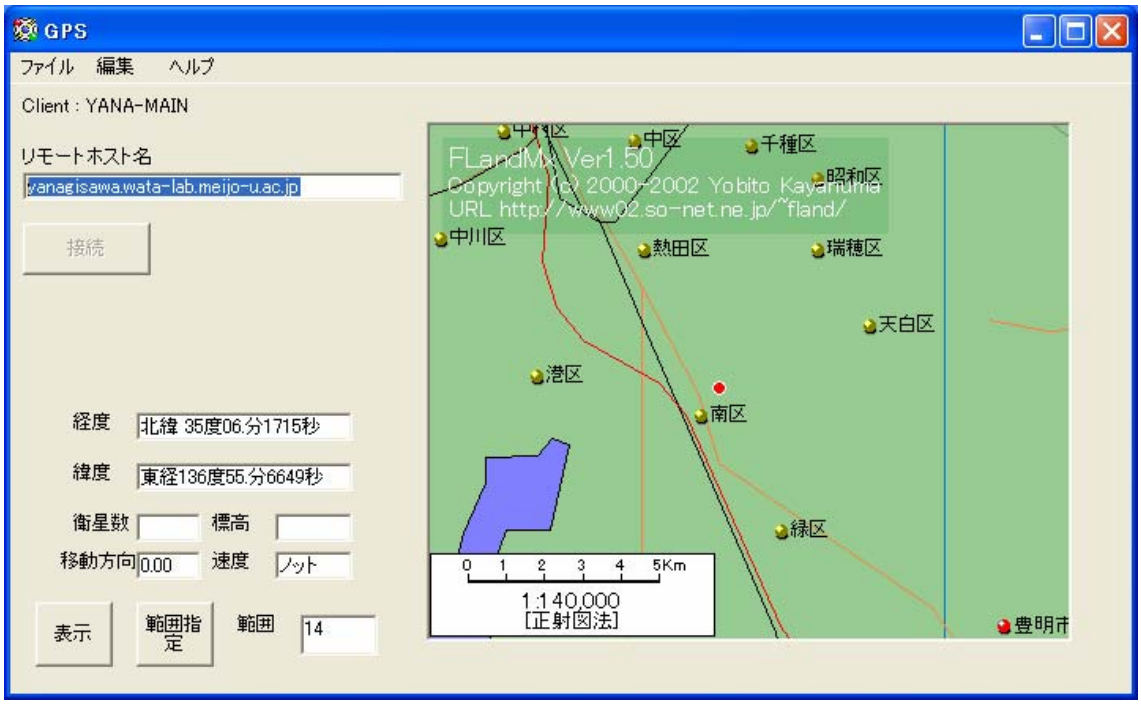

図 41　監視端末表示画面

4. NATF モジュール構成

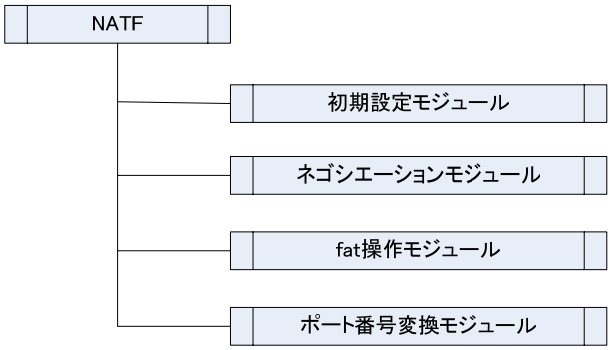

図 42　NATF モジュール

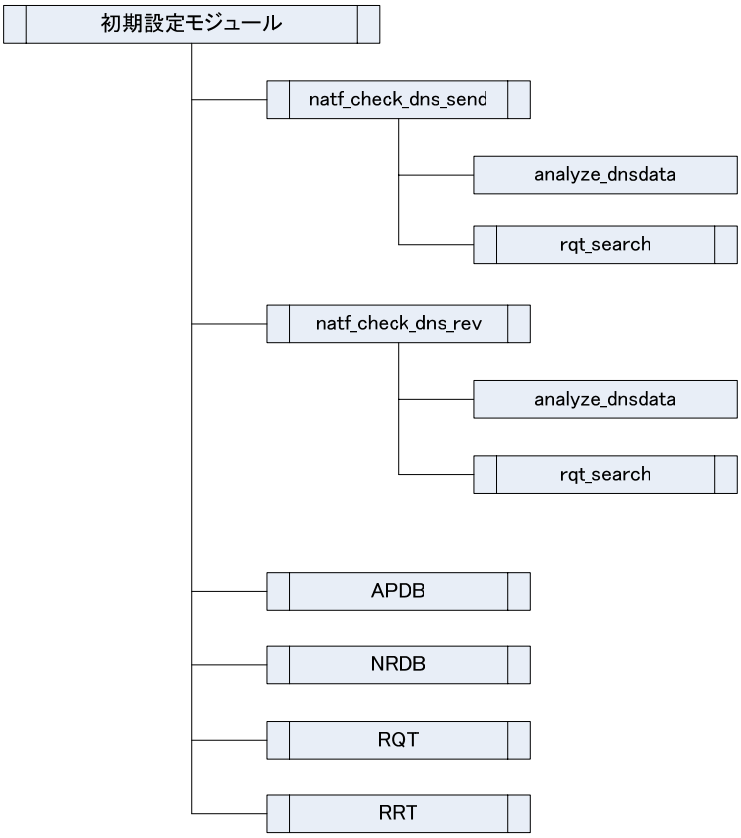

図 43　初期設定モジュール

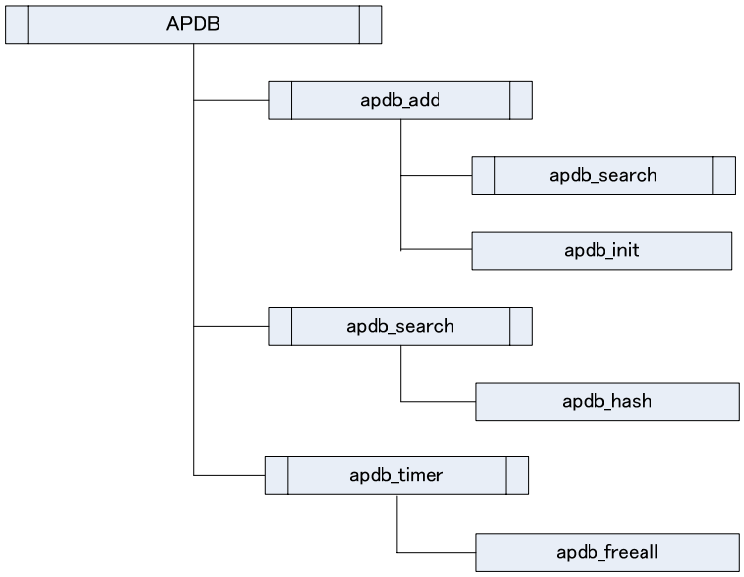

図 44　APDB モジュール

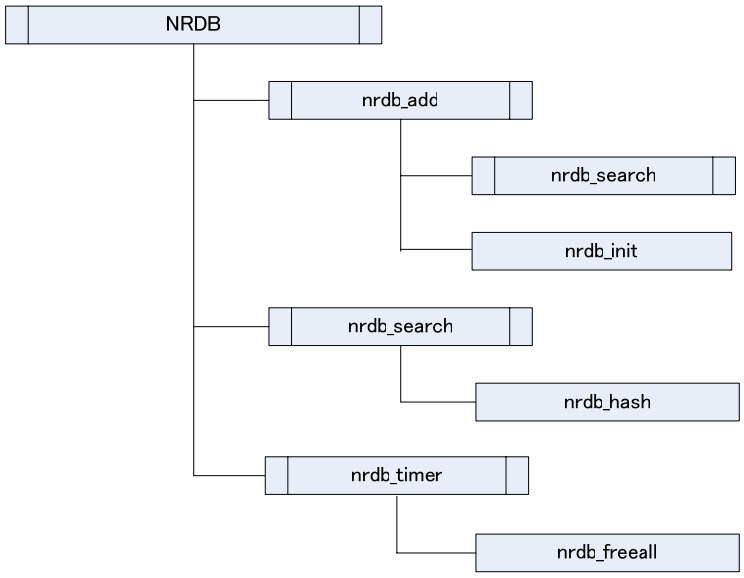

図 45　NRDB モジュール

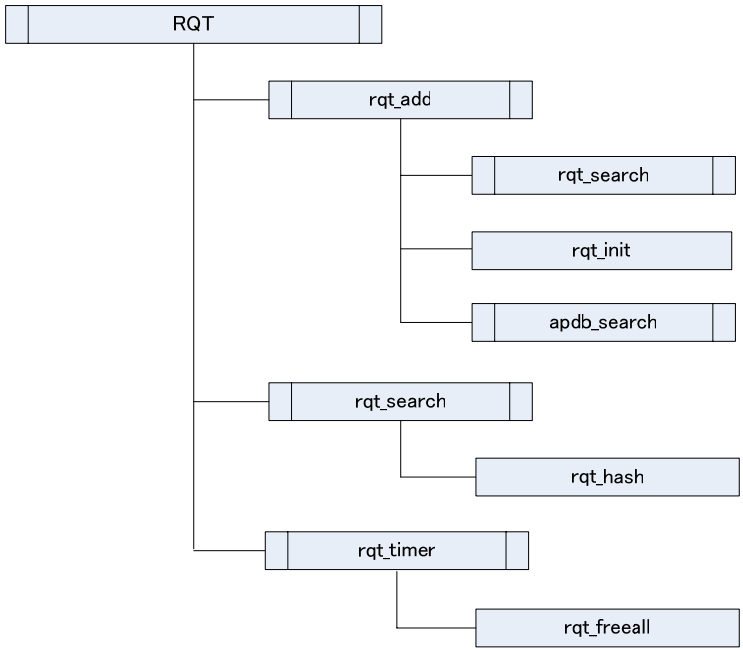

図 46　RQT モジュール

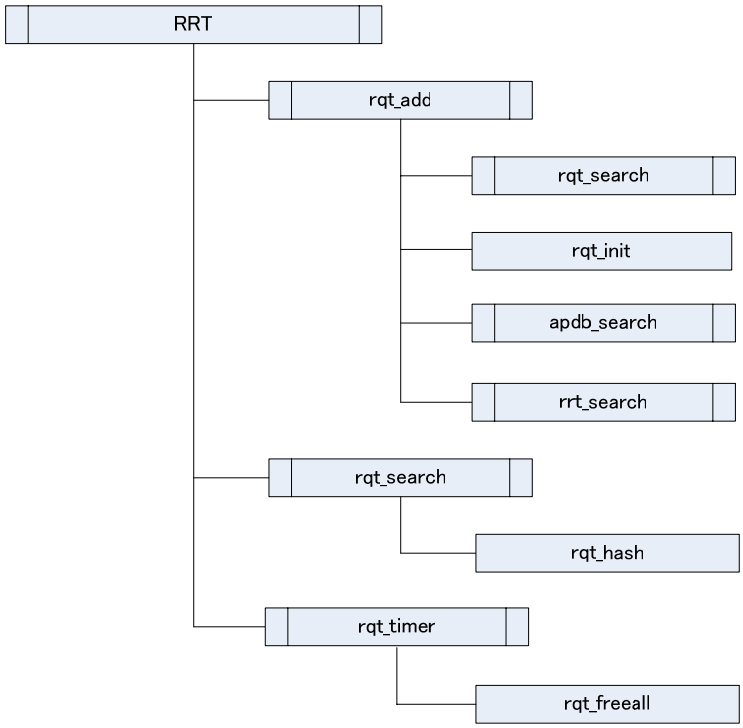

図 47　RRT モジュール

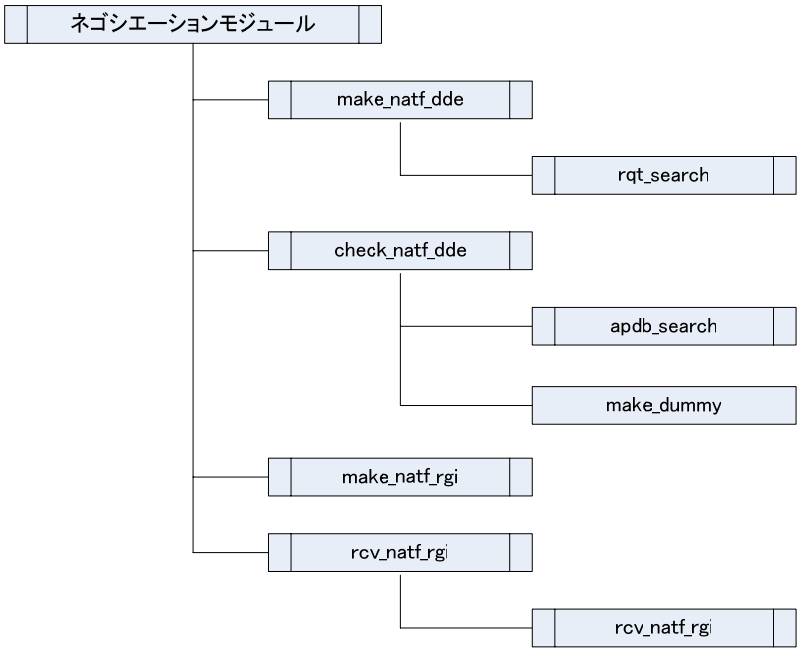

図 48　ネゴシエーションモジュール

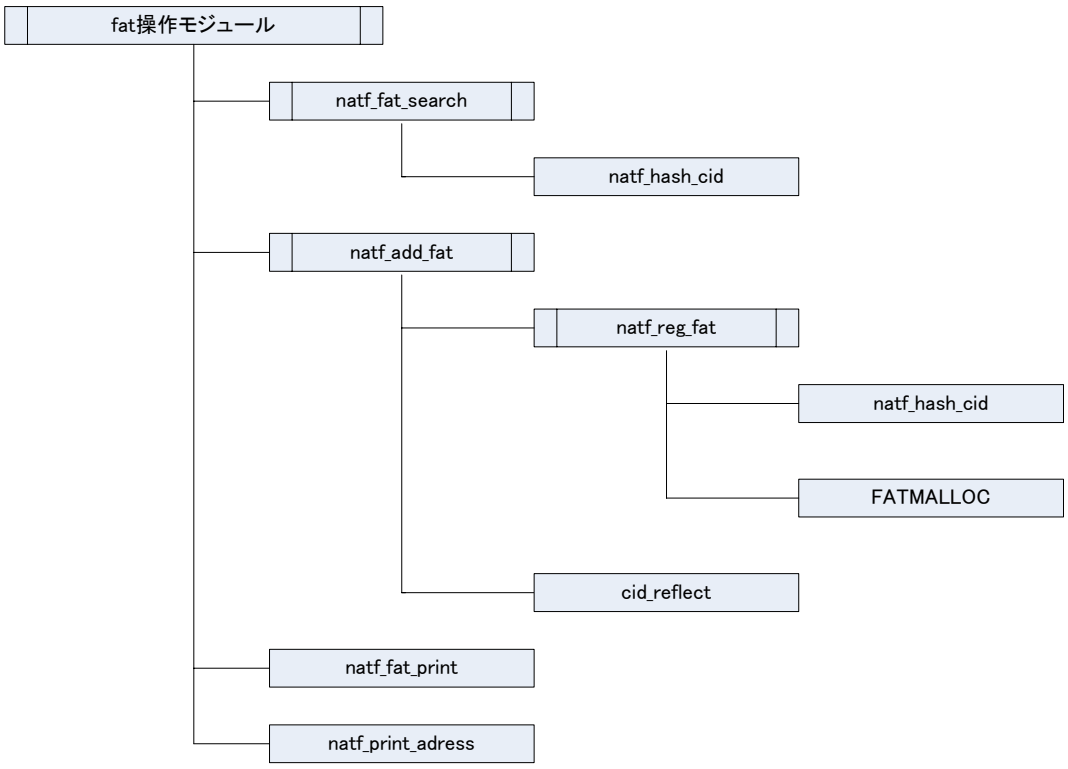

図 49　fat 操作モジュール

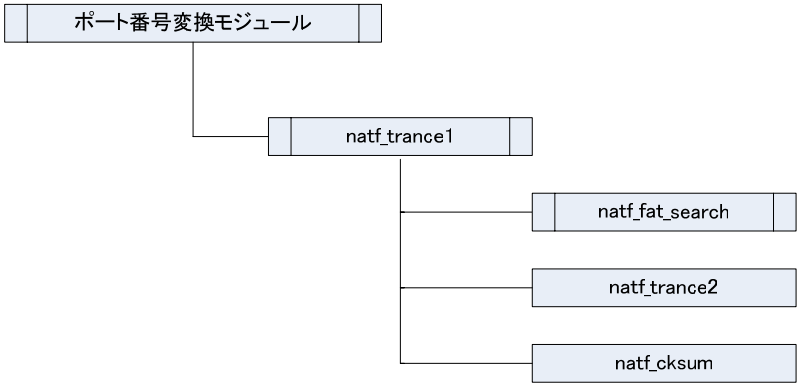

図 50　ポート番号変換モジュール

5. NATF モジュールの関数定義

表 1　初期設定モジュール

関数	説明
analyze_dnsdata	DNS パケットの解析
natf_check_dns_send	DNS 問合せパケット処理
natf_check_dns_rev	DNS 応答パケット処理
nrdb_hash	ホスト名とドメイン名をキーとしてハッシュを生成する
nrdb_add	nrdb に通信相手のホスト名とドメイン名を追加する
nrdb_search	nrdb を検索する
nrdb_init	
nrdb_freeall	
nrdb_timer	
apdb_hash	ホスト名をキーとしてハッシュを生成する
apdb_add	apdb に NATFBOX 配下にいるホスト名と IP アドレスを追加する
apdb_search	apdb を検索する
apdb_init	
apdb_freeall	
apdb_timer	
rqt_hash	Transaction ID をキーとしてハッシュを生成する
rqt_add	rqt に Transaction ID と，IP アドレスのハッシュを追加する
rqt_search	rqt を検索する
rqt_init	
rqt_freeall	
rqt_timer	
rrt_hash	IP アドレスをキーとしてハッシュを生成する
rrt_add	rrt に IP アドレスと，Transaction ID のハッシュを追加する
rrt_search	rrt を検索する
rrt_init	
rrt_freeall	
rrt_timer	

表 2　ネゴシエーションモジュール

関数	説明
make_natf_dde	ポート番号報告指示パケット生成処理
check_natf_dde	ポート番号報告指示パケット受信処理
make_dummy	ダミーパケット作成処理
make_natf_rgi	ポート番号応答パケット生成処理
rcv_natf_rgi	ポート番号応答パケット受信処理

表 3　fat 操作モジュール

関数	説明
natf_hash_cid	入力 CID からハッシュ値を計算する
natf_fat_serch	入力 CID を元に FAT レコードを検索
cid_reflect	CID を反転させる
natf_add_fat	CID と反転させた CID を natf_reg_fat に渡す
natf_reg_fat	FAT レコードを生成する
natf_print_fat	該当する FAT レコードを表示する
natf_print_address	IP アドレスを表示する
natf_set_table	FAT を初期化する

※CID…コネクションを管理する情報をまとめた識別子を指す．「Connection ID」の略である．CID は送信元/宛先 IP アドレス，送信元/宛先ポート番号，プロトコル番号の５つの情報から構成されている．

表 4　ポート番号変換モジュール

関数	説明
natf_trance1	主処理
natf_trance2	ポート番号付け替え処理
natf_cksum	チェックサム差分計算

6. パケットフォーマット

NATF のメッセージは，DPRP 制御パケットのオプション部に挿入される．DPRP 制御パケットについては，「DPRP 仕様書」を参照．メッセージパケットの構造を図 51 に示す．

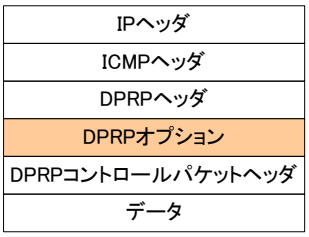

図 51　DPRP 制御パケットの構造

6.1 ポート番号報告指示パケット

DPRP オプション部分に以下の情報を付け加える．

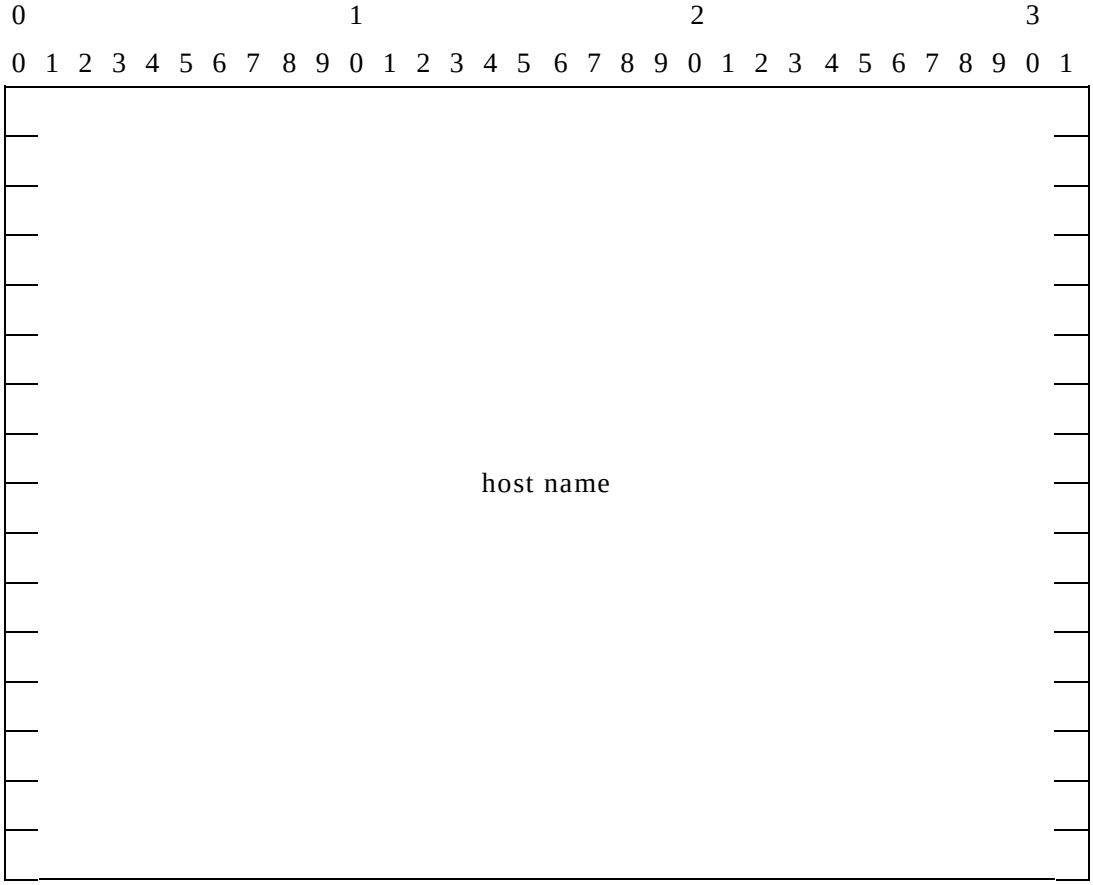

図 52　ポート番号報告指示パケットのフォーマット

表 5　ポート番号報告指示パケットのメッセージフィールド

フィールド	サイズ（byte）	値
Host Name	64	通信相手のホスト名情報

6.2 疑似パケット

ポート番号登録指示パケットより DPRP ヘッダ以下の情報を抜き取り TCP/UDP ヘッダ以下に貼り付け生成する.

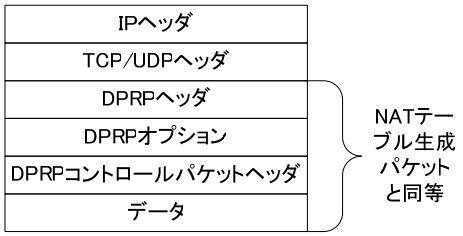

図 53　疑似パケットの構造

6.3 ポート番号登録指示パケット

DPRP オプション部分に以下の情報を付け加える.

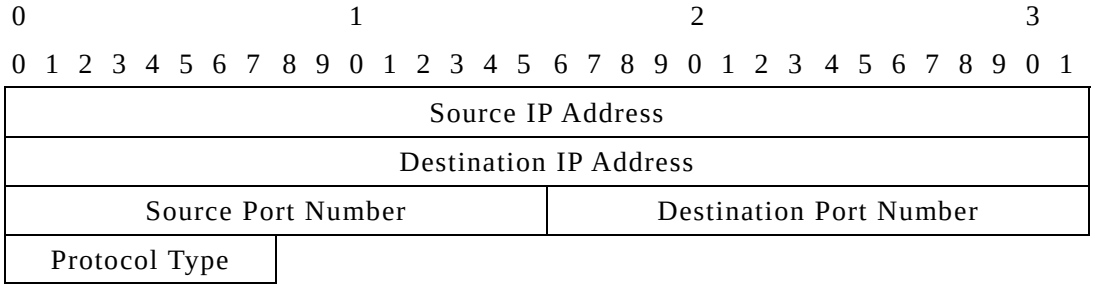

図 54　ポート番号指示パケットのフォーマット

表 6　ポート番号指示パケットのメッセージフィールド

フィールド	サイズ	値
Source IP Address	4	NAT 変換後の送信元 IP アドレス
Destination IP Address	4	NAT 変換後の宛先 IP アドレス
Source Port Number	2	NAT 変換後の送信元ポート番号
Destination Port Number	2	NAT 変換後の宛先ポート番号
Protocol Type	1	プロトコルタイプ