社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－DPS－122（62）
2005－CSEC－28（62）
2005／3／23

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

柳沢信成†　加藤尚樹†　鈴木秀和†　渡邊晃†
†名城大学大学院理工学研究科 〒468-8502 愛知県名古屋市天白区塩釜口 1-501
†E-mail: m0432041@ccmailg.meijo-u.ac.jp，wtnbakr@ccmfs.meijo-u.ac.jp

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 NAPT が設置され自由に通信を行うことができない．この NAPT の通信制約を解決するプロトコルとして，NATF（NAT Free Protocol）がある．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環境にある端末同士が自由に通信を可能にする方式を提案し，実装方法を述べる．

Proposal and its implementation on the communication between terminals in independent private address areas via a global address area

Nobushige YANAGISAWA†　Naoki KATO†
Hidekazu SUZUKI†　Akira WATANABE†
†Graduate School of Science and Technology, Meijo University
1-501 Shiogamaguchi, Tenpaku-ku, Nagoya-shi, Aichi, 468-8502 Japan
†E-mail: m0432041@ccmailg.meijo-u.ac.jp，wtnbakr@ccmfs.meijo-u.ac.jp

Abstract　There are a global address area and a private address area in IPv4 world. It is not possible to communicate freely among them, because NAPT exists on the communication route. NATF (NAT Free Protocol) is a promising protocol to solve the restriction of NAPT. In this paper, a new method that enables the communication between terminals in independent private address areas via a global address area is proposed enhancing the idea of NATF.　The implementation method is also described.

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），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，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 が普及すれば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［2］と共存しつつ使い続けられると思われ, 上記課

題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 NAPT［3］が設置され，多くの場合ファイアウォールも併設される. 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，NAPT 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 NAPT 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 NAPT 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で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. 筆者らは端末側に新たにポート変換機能を追加することにより GA 空間からの通信開始を可能とする NATF（NAT Free Protocol）を提案してきた［4］.

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）と呼ぶ．一般に，異なる PA 空間ではアドレス管理が別々に行われており, アドレスの重複があり得る．したがって，このような通信を P2P で実現しようという発想はこれまでなかった．しかし，今後は各家庭内のプライベートアドレス端末同士の P2P 通信のニーズはあり得ると考えられ，これが実現できれば，今後のユビキタス社会において有益である.

CIPA では，NATF で拡張した端末の機能を NATF BOX に持たせる．すなわち，2 台の NATF BOX が，通信開始に先立って通信を行うポート番号の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とにより制約のない通信を行うことが可能になる.

以下，2 章に NATF，3 章に提案方式 CIPA について，4 章に実装方法，5 章にまとめを述べる.

2. NATF（NAT Free protocol）

NATF は，端末・DNS・NATF BOX が連携し，GA 空間の端末から PA 空間の端末への通信を可能とするプロトコルである．NATF BOX とは NATF が適用されたアドレス変換装置のことである．NATF の環境を図 1，シーケンスを図 2 に示す．DNS には，あらかじめドメイン内に存在する端末のプライベートアドレスを, LIP と呼ぶ拡張レコードとして登録しておく必要がある．DNS は，端末から PC2 に対する IP アドレスの問合せがあると，A レコードと共に, LIP レコードを付加して応答する．ここで，A レコードは NATF BOX のグローバルアドレス，LIP レコードは PC2 のプライベートアドレスとなる. 端末 PC1 の IP 層において，この DNS 応答から送信元 IP アドレス，宛先 IP アドレス，LIP アドレスを記憶する NLT(NATF List Table)を作成する(図 3). DNS 応答を IP 層から上位層へ渡す際, LIP レコード(PC2)を削除し，A レコード（NATF BOX）のみを渡す．アプリケーションでは，通信相手はグローバルアドレスを持つ NATF BOX であると認識する．次に，PC1 は NATF BOX に対して通信を行うが，通信開始に先立ち， NATF プロトコルを実行する．PC1 は，

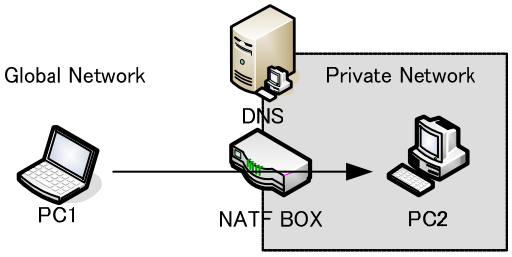

図 1　NATF の環境

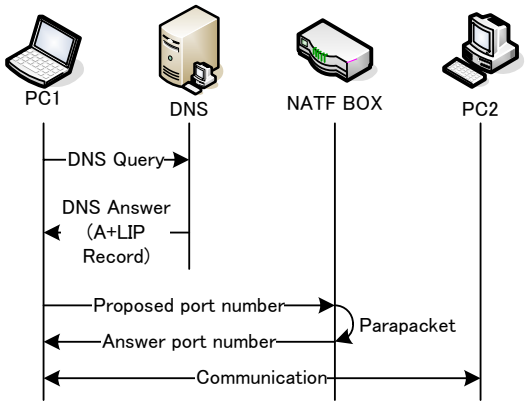

図 2　NATF のシーケンス

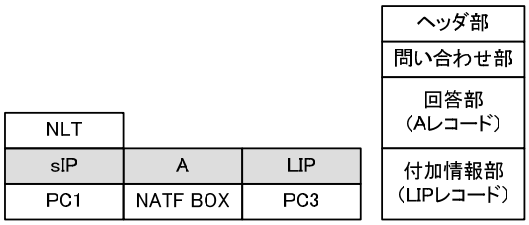

図 3　NLT と DNS 応答

最初のパケットを一時待避し，NATF BOX にポート番号報告指示パケットを送信する．このパケットには，PC1 が通信中に使う送信元アドレス・ポート番号，宛先アドレス・ポート番号，LIP，プロトコルの情報が含まれる．NATF BOX はポート報告指示パケットの情報から，疑似パケットと呼ぶ仮想的なパケットを作成する．これは，NATF BOX に NAPT テーブルを強制的に作成させるためのもので，PC2 から PC1 へパケットが送信されたものと見せかける．これにより NATF BOX のアドレス変換テーブルが生成される．NATF BOX はここで割り当てたポート番号を，ポート番号報告応答

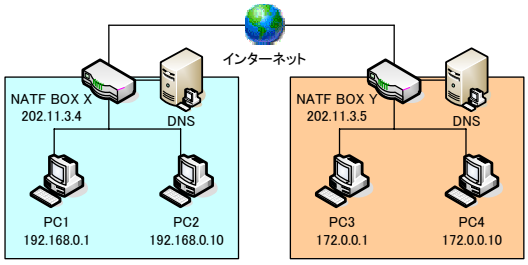

図 4　CIPA の環境

パケットを用いて PC1 に送り返す．疑似パケットは破棄される．PC1 は，報告されたポート番号を元に，FAT（natF Address Translation）テーブルを生成し，NLT を削除する．PC1 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 は疑似パケットで生成された NAPT アドレス変換テーブルを用いて IP アドレス変換、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．

3. CIPA の方式

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 4 に示す．CIPA では通信端末は一般端末でよく，NATF BOX 同士が NATF プロトコルを実行する．送信元の NATF BOX は NAPT アドレス変換とポート番号の変換（FAT ポート変換）を行う．

PC1 から PC3 へ接続を開始する場合の通信の流れを図 5 に示す．PC1 は PC3 に関する DNS 問合せを行う．DNS は，A レコードとして NATF BOX Y のグローバルアドレス（202.11.3.5）を，LIP レコードとして PC3 のプライベートアドレス（172.0.0.1）の情報を載せた DNS 応答を返す．NATF BOX X がこの DNS 応答を受け取ったら NLT を作成する．次に，応答パケットから LIP レコードを

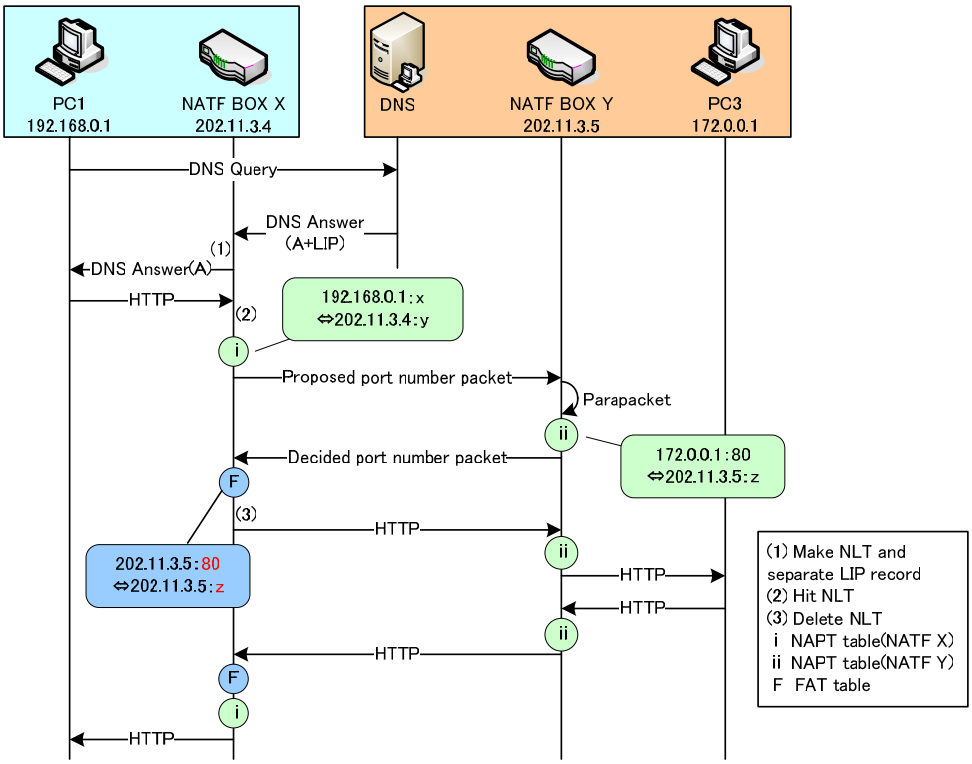

図 5　CIPA のシーケンス

分離し，A レコードのみを載せた DNS 応答を PC1 へ送る．PC1 は DNS 応答を受け取ったら通信を開始する．

以下 IP アドレスとポート番号の関係を，順を追って示す．スラッシュの左側は送信元 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．PC1 側の OS から動的に割り当てられた送信元ポート番号を x，PC3 を WEB サーバと仮定し，宛先ポート番号を 80 とすると，PC1 が送信するパケットは次のようになる．

192.168.0.1：x／202.11.3.5：80

NATF BOX X はこのパケットを受信すると，NLT を検索する．該当する NLT が存在したら NATF BOX 間で NATF プロトコルを実行するための準備を行う．NATF BOX X は一般の NAPT の手順に従い NAPT テーブルを生成し，最初のパケットのアドレス変換を行っておく．NAPT テーブルの情報は以下のように生成される．y は NAPT により動的に割り当てられたポート番号である．

{192.168.0.1：x⇔202.11.3.4：y}

したがって，アドレス変換後のパケットは

202.11.3.4：y／202.11.3.5：80

となる．NATF プロトコルを実行するため，このパケットは NATF BOX X 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 NATF BOX Y に送る．このパケットには，送信元 IP アドレス：ポート番号（202.11.3.4：y）と LIP：ポート番号（172.0.0.1：80）とプロトコル情報が含まれる．

NATF BOX Y は上記ポート番号報告指示パケットを受信したら，疑似パケットを作成する．疑似パケットの IP アドレスとポート番号の内容は以下の通りである．

172.0.0.1：80／202.11.3.5：z

このパケットにより，NATF BOX Y は NAPT テーブルを生成する．テーブルの内容は以下の

通りである．z は NAPT により動的に割り当てられたポート番号である．

{172.0.0.1：80⇔202.11.3.5：z}

テーブル生成後，疑似パケットは破棄する．NATF BOX Y はこの変換ポート番号（z）を，ポート番号報告応答パケットを用いて NATF BOX X に送る．

NATF BOX X 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．

{202.11.3.5：80⇔202.11.3.5：z}

FAT テーブル作成後，NLT を削除し NATF プロトコルは終了する．

NATF BOX X は，待避していたパケットを FAT テーブルよりポート変換し，NATF BOX Y 宛に送信する．このパケットの内容は次のとおりである．

202.11.3.4：y／202.11.3.5：z

このパケットを受信した NATF BOX Y では，NAPT アドレス変換後，PC3 に送信する．パケットの内容は以下のように変わる．

202.11.3.4：y／172.0.0.1：80

PC3 から PC1 への応答は，上記と逆の変換で行われる．

以下，NATF BOX X では NAPT アドレス変換と FAT ポート変換，NATF BOX Y では NAPT アドレス変換をすることで，通信を行う．

このように，NATF BOX 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 NATF BOX が２つのポート番号の変換を行う．PC1 は NATF BOX Y と，PC3 は NATF BOX X と通信しているようにみえる．

4.実装

4.1　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．natd を組み込んだ FreeBSD カーネル内に，表 1 のモジュールを組み込むことで実現方法を検討中である．図 6 のように，natd や IP 層で行われる既存の処理に一切の変更を加えず，IP 層の 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形を採用する．今回は NATF BOX に組み込むモジュールと，ポート番号報告指示パケット・ポート番号報告応答パケットのパケットフォーマットについて説明する．

4.2　試作モジュールと機能

NATF の機能を実現するための試作モジュールとその機能を表 1 に示す．試作システムは FAT ポート変換モジュール，拡張 DNS モジュール，NLT モジュール，疑似パケットモジュールで構成される．

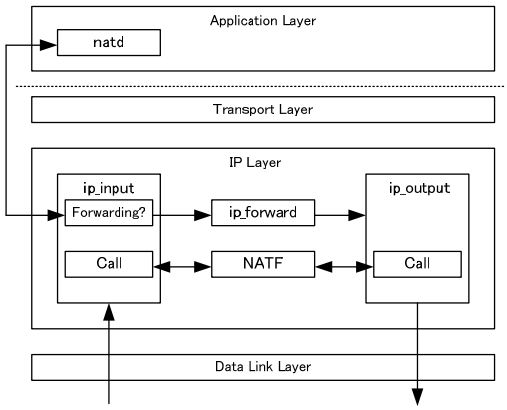

図 6　実装場所

表 1　モジュールと機能

モジュール	機能
FAT ポート変換	FAT テーブルを生成し，ポート変換を行う
拡張 DNS	NLT の更新を行う．また，A レコードと LIP レコードを分離し，後者を削除する
登録／応答	ポート番号報告／登録パケットを生成・送信する
NLT	NLT の作成／検索を行う
疑似パケット	疑似パケットを用いて NAPT テーブルを生成する

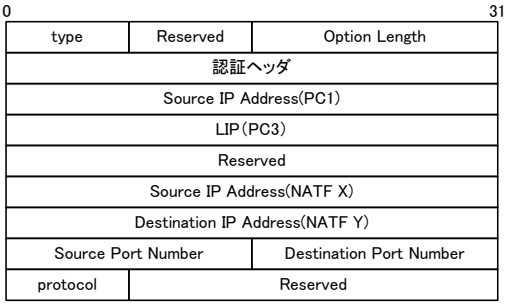

図 7　ポート番号報告登録パケットのフォーマット

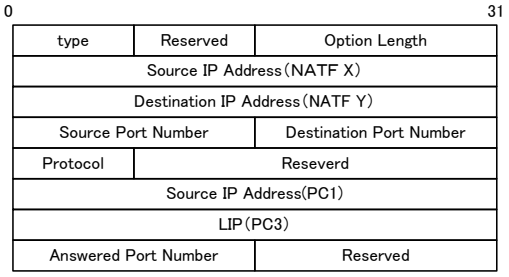

図 8　ポート番号報告応答パケットのフォーマット

4.3　パケットフォーマット

ポート番号報告指示パケット, ポート番号報告応答パケットのフォーマットを図 7, 図 8 に示す．これらは ICMP 上に定義されている．ポート番号報告指示パケットは, 疑似パケットを作成するために必要な送信元アドレス・ポート番号と，宛先アドレス・ポート番号，プロトコル情報で構成される. ポート番号報告応答パケットは, 上記に加え, NATF BOX Y の NAPT テーブルで変換されるポート番号で構成される．

5. まとめ

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 の提案を行い, 実装方法を述べた．今後は，実装を完了し，システムとしての動作検証と性能評価を行う．

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] 加藤尚樹，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”, WinF2004 論文集 pp.222-225, September 2004

[5] 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 pp.217-221, September 2004

[6] 鈴木秀和，渡邊晃，“フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の仕組み”, 研究報告「コンピュータセキュリティ」No.2004-CSEC-026 , July 2004

[7] 竹内元規，渡邊晃，“モバイル端末の移動透過性を実現する Mobile PPC の提案”, 研究報告「モバイルコンピューティングとユビキタス通信」No.2004-MBL-030 , July 2004

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

[9] P. Mockapetris, “DOMAIN NAMES - IMPLEMENTATION AND SPECIFICAT -ION”, RFC1035, November 1987

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