TTLを用いた能動的NAT検出手法の実装と評価

三村 守[†] 中村康弘[†]

LAN内でNAT接続された機器を識別・検出するために，パケットがNATルータを経由した際にTTLが減少することを利用する手法が提案されている．しかしながらこの手法では，監視対象ホストから検出システムまでのホップ数が既知である必要がある．本研究では，TTLを能動的に取得することにより，ホップ数が既知でない場合においてもNAT接続された機器を検出しうる手法を提案するとともに，既存のNAT検出手法と比較検討し，提案手法の有効性を示す．

Development and Evaluation of an Active NAT Detection Method Using TTL

Mamoru Mimura[†] and Yasuhiro Nakamura[†]

To detect NAT devices or hosts in LAN, a method has been proposed, which uses the characteristic that TTL decreased by NAT router when a packet is forwarded. However, number of the hop count from the target host to the detection system must be known in that method. This paper proposes a method which can detect NAT devices or hosts by getting TTL actively, when the hop count isn't known. It is compared with the other NAT detection method, and the effectivity of proposed method is shown.

1. はじめに

近年，コンピュータネットワークを介した情報漏洩やウィルス感染が問題となっている．これらの問題が起きる要因の1つに，ネットワーク利用者の故意または過失により，脆弱性対策等を十分に行っていない機器をネットワークに接続することが考えられる．このような問題への対策として，古くはKerberos等の機器認証，MACアドレスや利用者認証に基づく機器認証および近年では検疫ネットワークを構築する手法が用いられている．しかしながら，十分な検証のうえで接続が許可された承諾機器であっても，複数のネットワークインタフェースを持ち，NAT（Network Address Translator）[1]機能を持つルータとして動作させることで，検証が十分ではない未承諾機器を当該LANに接続できてしまう．こうした背景から，不正なNAT接続機器を検出する技術が必要となる．この論文では，まず既存のNAT検出関連技術について考察した後，Phaal[3]の手法を基に，トレースパケットを送信して能動的にIPヘッダ内のTTL値を取得し，これによりNAT検出を行う能動的検出手法を提案する．さらに提案手法を実装し，検証実験および従来手法との比較検討により有効性を示す．

† 防衛大学校情報工学科
Department of Computer Science, National Defense Academy

2. 関連技術

図1を用いてNATによる不正接続の例を示す．未承諾機器検出システムはネットワーク内のパケットを監視し，データリンク層から得られるMACアドレスにより承諾機器と未承諾機器を識別する．ホストA，Bは承諾機器であり，ネットワークの利用を許可する．ホストCは未承諾機器であるため，未承諾機器検出システムによって検出され，その通信は無力化される．しかし，承諾機器であるホストAがNATルータとして動作し，未承諾機器であるホストDをホストAに接続した場合，ホストDはホストAを介してネットワークの利用が可能となる．既存のシステムではこのようなNATを利用した不正接続を検出することができない．

本章ではネットワーク内の機器を識別するためのいくつかの既存技術について，NAT接続機器検出への適用可能性を検討する．

2.1 TTLを用いたNAT検出技術

IPヘッダ（**図2**）に含まれるTTL（Time To Live）の値はオペレーティングシステム（以下OS）やそのプロトコルにより特定の初期値を持つ（**表1**）．TTL

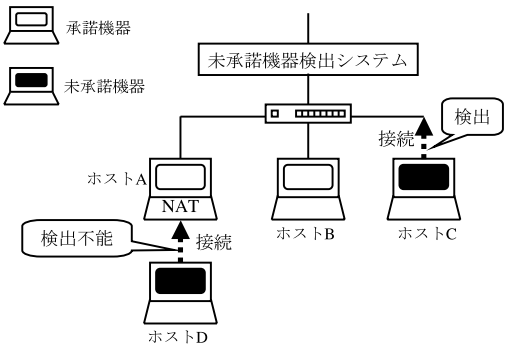

図 1 NAT による不正接続
Fig. 1 Disapproval connection by NAT.

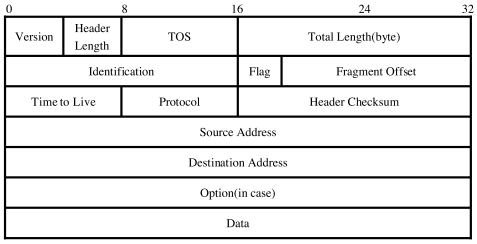

図 2 IP ヘッダ
Fig. 2 IP header.

表 1 デフォルト TTL 値
Table 1 Default TTL values.

OS	ICMP	UDP	TCP
Linux 2.6	64	64	64
FreeBSD 5.4	64	64	64
OpenBSD 3.9	64	64	64
Solaris 2.6	255	255	255
AIX 4.2	255	60	64
WindowsXP	128	128	128

はルータを経由するごとに 1 減算され宛先ホストに到達する．これは RFC791 [2] で規定されており，この仕組みを利用した NAT 検出手法がすでに提案されている[3]．**図 3** を例にこの手法を説明する．監視対象であるホスト A は LAN の運用ポリシに反し，NAT ルータとして機能している．検出システムでは監視対象ホストまでのホップ数が 1 であることが既知とする．検出システムはホスト A から送信されたパケットを傍受すると，途中のルータ R で TTL が 1 減算されるためホップ数が 1 であることを確認できる．以後 TTL を継続監視する．NAT の内側から送信されたパケットを傍受した場合には，TTL はホスト A でさらに 1 減算されるためホップ数は 2 となり，既知のホップ数と一致しないため，NAT 接続された機器が存在することを検出できる．この手法は負荷が小さく，仕組みが単純であるため，導入が容易である．しかし，監視

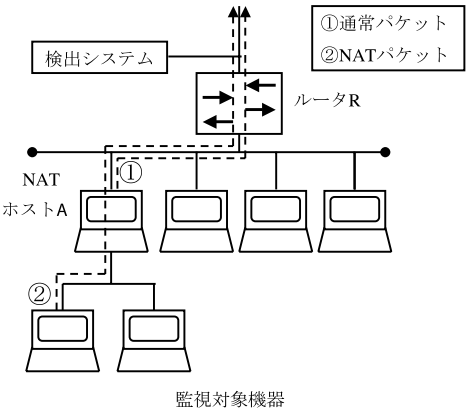

図 3 TTL を用いた既存の NAT 検出手法
Fig. 3 A previous NAT detection method using TTL.

表 2 OS Fingerprint
Table 2 OS Fingerprint.

パケット	ヘッダ	Linux	Free BSD	Open BSD	Solaris	AIX	Windows XP
SYN	TTL	64	64	64	255	64	128
	Window	5840	65535	16384	8760	16384	64240
	IPid	1 ずつ増加※	1 ずつ増加	ランダム	1 ずつ増加	1 ずつ増加	1 ずつ増加
	Length	60	60	64	44	44	48
	TOS	0x00	0x00	0x10	0x00	0x10	0x00

※ コネクション確立まではランダム

対象の IP アドレスから検出システムの設置場所までのホップ数が既知でなければならない．したがって，ホップ数が変化しない静的ネットワークでは全ホストから検出システムまでのホップ数をあらかじめ調べておく必要があり，ホップ数が変化する動的ネットワークには導入することができない．また，あらかじめ監視対象 IP アドレスと正しいホップ数のデータベースを作成する必要があるため，DHCP [4] を利用する場合等，IP アドレスが動的に変化する環境に導入することは困難である．

2.2 オペレーティングシステムの識別技術

一般に IP ヘッダに含まれる，パケット長，TTL，初期ウィンドウ値，IPid の変化，オプション等は OS ごとに規則性がある[5]．**表 2** に TCP SYN パケットの OS Fingerprint の一例を示す．この情報を利用すれば，パケットを受動的に観測することで，その送信元ホストの OS を識別することが可能となる．これを NAT 検出技術として利用すれば，同一の送信元 IP アドレスが複数の異なる OS を使用していることを，NAT として検出する手法が考えられる．しかし，NAT の

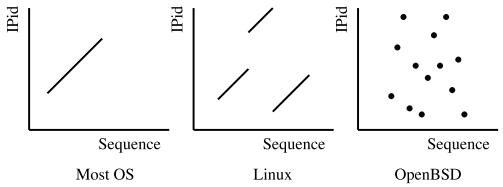

図 4 IPid の規則性
Fig. 4 Regulations of IPid.

内側で複数のホストが同一の OS を使用している場合には，これらを区別することができない．また，同一のホストに複数の OS を共存させている場合や，仮想マシン等で別の OS を動作させている場合についても区別できず，NAT として検出してしまう．

2.3 IPid を用いた NAT 検出技術

特定の OS においては IPid が 1 ずつ増加する性質を用いて，ホップ数に無関係に NAT の内側に接続されているホストの数を検出することができる[6]．ほとんどの OS では IPid は 1 ずつ増加するため，観測した IPid を IP アドレスごとに到着順にプロットすることで，値の連続性を観測することができる（**図 4**）．図中の線分を数えることで，対象 IP アドレスで動作するホスト台数を検出することができる．しかし，IPid の値は OpenBSD においてはつねにランダムに変化し，また，Linux においてはセッション確立まではランダムに変化する性質があるため，この手法を用いることができない．これらの OS については OS の識別技術を用いて分離する方式が提案されている．また，パケット観測数が精度に大きく影響するという問題もある[7]．さらに，同一のホストに複数の OS を共存させている場合や，仮想マシン等で別の OS を動作させている場合についても区別することができない．IPid が 1 ずつ増加する性質は OS の実装に依存するものであり，実装が変更された場合にはこの技術は利用できなくなる．IPid は経路途中でパケットが分割された場合，宛先ホストが分割されたパケットを再構築する際に参照する識別子であり，必ずしも 1 ずつ増加させる必要はない．また，値の変化について RFC に規定はないため，今後異なる実装が行われる可能性もある．

2.4 TCP ポート番号を用いた NAT 検出技術

NAT ルータはパケットを中継する場合，受信したパケットの TCP ポート番号を比較的高い番号に変換して送信するものがある．この変換後のポート番号に規則性があれば，NAT 検出に利用できる可能性がある．しかし，Linux ルータ等，機器によってはポート番号の変換を行わずにパケットを転送するため，ポート番号を利用して NAT 検出を行うことは困難である．

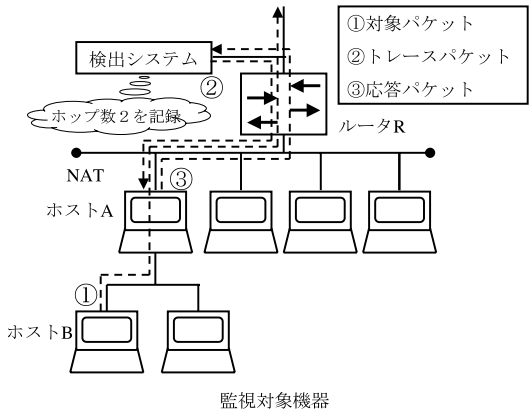

図 5 TTL を用いた能動的 NAT 検出手法（1）
Fig. 5 An active NAT detection method using TTL (1).

2.5 Clock Skew を用いたリモート機器識別

TCP オプションに含まれるタイムスタンプ値を取得し，正確な時刻との差分をとることにより，クロック周波数のずれ（Clock Skew）を得ることができる．Clock Skew の時間的変化は機器固有の量となることを利用し，リモート機器識別を行う手法が提案されており[8]，これを NAT 検出に応用することが考えられる．Clock Skew を用いる方法はホップ数に無関係であり，偽装されにくいが，高い検出精度は得られない．Linux と FreeBSD は TCP セッション開始時に送信する TCP SYN パケットにタイムスタンプオプションを付加するが，他のほとんどの OS は標準では付加しないため，Clock Skew を受動的に観測することができない．また，可変クロックの CPU には対応できないという問題もあり，NAT 検出への適用は難しい．

3. TTL を用いた能動的 NAT 検出手法

前章で述べた IP ヘッダの TTL を用いた NAT 検出技術を利用するためには，監視対象機器から検出システムまでのホップ数が既知である必要があった．ホップ数が既知でない場合には，検出システムから当該機器に対して能動的にパケットを送信し，その応答パケットからホップ数を取得すればよい．ここでは検出システムから当該機器に対して送信するパケットをトレースパケットと呼び，この能動的手法をアクティブ NAT 検出手法と呼ぶ．

図 5 を例に提案手法を説明する．監視対象であるホスト A は LAN の運用ポリシに反し，NAT ルータとして機能している．検出システムは NAT の内側にあるホスト B から送信されたパケットを傍受すると，その TTL からホップ数が 2 であることを記録するとともに，送信元 IP アドレスに対しトレースパケット

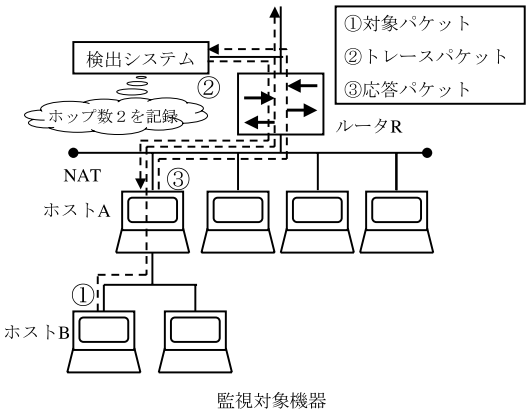

図 6 TTL を用いた能動的 NAT 検出手法（2）
Fig. 6 An active NAT detection method using TTL (2).

を送信する．このときのトレースパケットの宛先は，ホスト A の IP アドレスとなる．ホスト A からの応答パケット内の TTL 値から，ホスト A までのホップ数が 1 であることが検出できる．これが記録してある対象パケットのホップ数より小さい場合には，先の送信パケットは NAT の内側から中継されたものと判定できる．

一方，**図 6** のようにホスト A 自身や NAT を実施していないホストから送信されたパケットを傍受した場合，その TTL からホップ数が 1 であることを記録し，同様にトレースパケットを送信する．そして応答パケットからホップ数が 1 であることが分かり，先に記録したホップ数と一致するので，NAT の内側から送信されたパケットではないと判定できる．

3.1 前提条件

TTL を用いた手法で NAT を検出するためには，次の 2 つの前提条件がある．どちらの条件も OS やルータの実装に依存するものであるが，現在のところ，ほぼすべての OS や機器がこの前提条件に基づいて実装されており，実装を変更するための合理的理由はない．

- IP ヘッダの TTL は，ルータを経由するごとに 1 減算される．
- 監視対象ホストが送信するパケットの IP ヘッダの TTL は，OS 固有の定まった値である．

また，能動的 NAT 検出手法ではさらに次の前提条件が必要となる．

- 能動的に監視対象ホストから IP ヘッダを含む応答パケットが得られる．

3.2 トレースパケット

TTL を能動的に取得するためには，IP ヘッダを含む応答パケットを傍受する必要がある．たとえば，Ping 等で利用されている ICMP エコー要求を監視対象 IP アドレスに送信し，応答パケットとして受信する ICMP エコーから TTL を取得する方法が考えられる．ICMP エコーの TTL は，TCP や UDP と異なる場合があるが，これは既存のオペレーティングシステムの識別技術により，ホップ数に変換することが可能である．ほかにも TTL を含む応答パケットを取得するためのトレースパケットとして，TCP や UDP パケットの利用も考えられる．トレースパケットの候補として，以下の 3 つを検討する．トラフィックの増加を最小限とするため，1 つのトレースパケットに対し，1 つの応答パケットが得られることが望ましい．

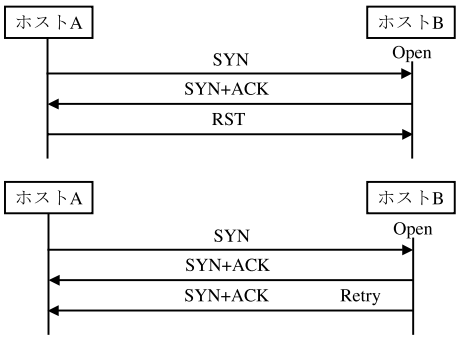

図 7 Open Port への TCP SYN 送信
Fig. 7 Sending TCP SYN to open ports.

3.2.1 ICMP

ICMP ではエコー要求のほか，タイムスタンプ要求，情報要求，アドレスマスク要求等のメッセージタイプを用いることができる．往復の経路が同一の場合には，ホップ数も同一になることから，時間超過メッセージも利用できる．しかしながら，ICMP の発着信がフィルタリングされている場合には応答を得ることができない．IPv4 における ICMP の利用は任意であり，プロトコル自身の脆弱性も多数報告されていることから，使用を許可していないネットワークもある．よって ICMP はトレースパケットとして技術的には利用可能であるが，セキュリティ性を考慮すると，利用には適さないものと考える．

3.2.2 TCP

一般に開いているポートに SYN パケットを送信すると，さらにその SYN+ACK パケットを受信することができる（**図 7**）．しかしここで通信は完結せず，応答を返さない場合には，SYN+ACK パケットが再送されてしまう．応答として RST パケットを送信すれば通信は完結するが，他の手法よりもトラフィックが増加してしまう．一方，閉じているポートに SYN パケットを送信すると，RST+ACK パケットを受信することができる（**図 8**）．この手法は応答パケットを

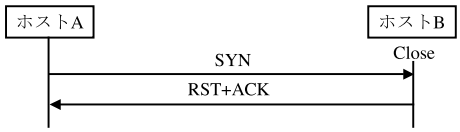

図 8 Close Port への TCP SYN 送信
Fig. 8 Sending TCP SYN to close ports.

得るために利用可能であり，トラフィックの増加も最小限となる．ただし，異常検知型の IDS（Intrusion Detection System）が動作している場合には，動作に影響する可能性がある．また，フィルタリングされているポートからは，応答を得ることはできない．

3.2.3 UDP

開いているポートおよびフィルタリングされているポートに UDP パケットを送信しても，応答を得ることはできない．閉じているポートに UDP パケットを送信すると，ICMP 宛先到達不能メッセージの応答を得ることができる．この手法も ICMP と同様の理由により，利用には適さないものと考える．

3.2.4 ま と め

トレースパケットととしては ICMP，TCP および UDP が利用可能であり，TCP 以外では ICMP パケットを利用する必要がある．しかし，ICMP はセキュリティ上の理由から許容していない LAN も存在する．したがって，LAN の運用上 ICMP を許容していない場合には，TCP を利用する必要がある．ICMP を許容している場合には，いずれのプロトコルも利用可能である．

3.3 パケットフィルタリングの問題

Windows XP の ICF（Internet Connection Firewall）や，Linux の iptables 等のファイアウォールを用いてパケットフィルタリングを実施しているホストからは，一般に応答を得ることができない．そこで，ここでは利用者の多い ICF が導入されているホストに対し，能動的に TTL が取得できるかどうかを検討した．

3.3.1 例外パケットの送信

SYN+FIN フラグまたは SYN+RST フラグを立てたパケットを送信すると，SYN+ACK パケットを返信する OS やルータがある．フィルタリングにおいて SYN フラグしか評価せず，このような例外パケットを通してしまうパケットフィルタも存在する．また，パケットをフラグメント化することにより，パケットフィルタを回避できる場合もある．このような例外パケットを用いて，ICF が有効になっているホストから応答が得られるかどうか実験を実施したが，開いているポートからも，閉じているポートからも応答を得ることはできなかった．ICF では例外パケットには応答しない実装が行われているものと判断できる．

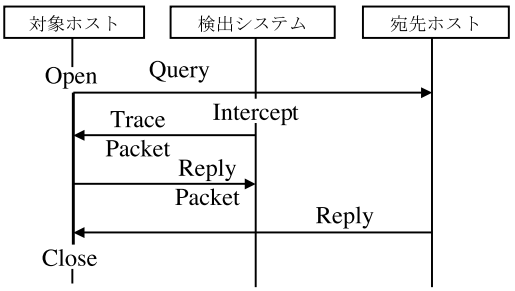

図 9 パケットの傍受
Fig. 9 Interception of packets.

表 3 トレースパケット
Table 3 The trace packets.

Proto	Trace	Dest	Reply
ICMP	ECHO REQUEST	-	ECHO
UDP	ANY	CLOSE PORT	ICMP UNREACHABLE
TCP	SYN	CLOSE PORT	RST+ACK

3.3.2 パケットの傍受

図 9 に示すように，ファイアウォールを有効としているホスト内部からの通信について，パケットを傍受し，トレースパケットを送信して応答を得ることができるか検証する．ファイアウォールは内部から送信されるパケットの通過を許可する場合，その応答パケットの受信も許可する必要がある．応答を許可しなければ，相互に通信を成立させることができないからである．特にデータグラム通信ではコネクションを確立しないため，一定時間ポートを開放状態にして応答を待つ必要がある．ICF を有効にしているホストからの TCP 接続要求に対し，応答が到着する前に，送信元ポートに別の送信元ポートから TCP パケットを送信したが，応答を得ることはできなかった．同様に UDP ポートに対し，通常の UDP パケット，Loose Source and Record Route パケット，More Fragment パケット等を送信したが，応答を得ることはできなかった．ICF では受信部だけではなく，送信部でもフィルタリングを実施しているものと考えられる．このように，ICF が有効なホストからは，開いているポートに SYN パケットを送信した場合，すなわち許可された正当な接続要求を送った場合にのみ応答を得ることができた．

トレースパケットの種類を**表 3** にまとめる．確実に応答を得るためには，監視対象ホストでトレースパケットに利用するプロトコルの送受信を許可する必要

がある．

3.4 トレースパケット送信のタイミング

トレースパケット送信のタイミングとしては，定期的に監視対象 IP アドレスにポーリングする方式と，監視対象 IP アドレスからの送信を傍受するたびに送信する方式が考えられる．定期的に監視対象 IP アドレスにポーリングする方式では，通信が常時傍受できないホストに対しても，能動的に TTL を取得することができる．しかし，そのようなホストで NAT が実施されていた場合，NAT の内側のホストからの通信も傍受できないため TTL を取得することができない．また，LAN の環境によっては，電源が入っていない等の理由により応答を返さないホストも存在することから，無駄なトラフィックの増加を招く．送信元が監視対象 IP アドレスに該当するパケットを傍受するたびに送信する方式では，通信を傍受できないホストの TTL を取得することはできないが，電源が入っていない等の理由により応答を返さないホストへパケットを送信することはなく，無駄なトラフィックの増加は生じない．よって本手法では，対象パケットを傍受するたびにトレースパケットを送信する方式を採用する．

3.5 受動的 NAT 検出

トレースパケットに応答しないホストについては，傍受したパケットのホップ数をシステムに記録し，同一ホストの別パケットのホップ数と比較することにより NAT の判定を行う．送信元ホストごとに TTL から求めたホップ数を記録する．以後，継続して TTL からホップ数を算出し，同一送信元ホストとして記録されている値と異なるホップ数が検出された場合，NAT を実施しているホストと判定する．この手法はトレースパケットを送信しない場合にも利用することができる．ここでは，この受動的手法をパッシブ NAT 検出手法と呼ぶ．

この手法は受動的に動作が可能であり，ホップ数が既知である必要はないという利点がある．ただし，異なるホップ数を検出するためには，NAT の内側のホストだけでなく，NAT ホスト自身から送信されたパケットも傍受する必要がある．このため，市販の NAT ルータ等を用いて，NAT の内側に接続したホストが主に通信を実施するような環境では検出が難しい．

3.6 検出アルゴリズム

図 10 にトレースパケットを送信するアクティブ＋パッシブ NAT 検出手法の提案アルゴリズムを示す．検出システムは，監視対象 IP アドレスから送信されたパケットを傍受すると，送信元 IP アドレスに対してトレースパケットを送信するとともに，IP アドレスとホップ数を記録する．送信元ホストから応答パケットが得られた場合には，そのホップ数と先に記録しているホップ数を比較し，ホップ数が異なれば NAT と判定する．応答が得られない場合には記録したホップ数を保持し，別に傍受したパケットの TTL から異なるホップ数を検出した場合には NAT と判定する．

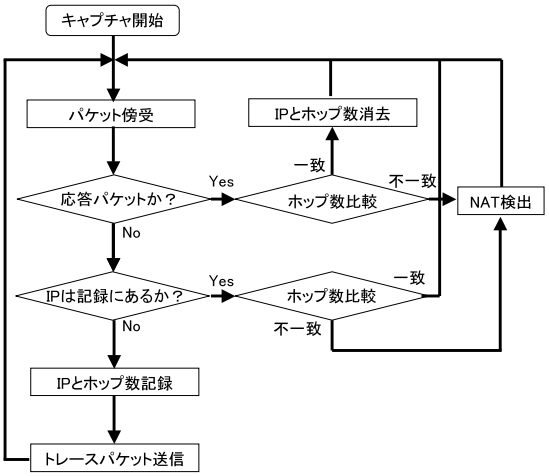

図 10 アクティブ＋パッシブ NAT 検出手法
Fig. 10 The active + passive method.

4. 検証実験

4.1 実　　装

前章で示したアクティブ＋パッシブ NAT 検出手法を実装した検証プログラムを，C 言語と pcap ライブラリを利用して作成した．このプログラムは Linux，FreeBSD，AIX 等の pcap ライブラリが利用可能な OS で動作する．また，比較検証のため，パッシブ NAT 検出手法および TTL を用いた従来の NAT 検出手法を実装したプログラムも作成した．以後，次の 3 つのプログラムを用いて検証実験を行い，結果を比較検討する．

- アクティブ＋パッシブ NAT 検出手法（図 10）
- パッシブ NAT 検出手法（**図 11**）
- TTL を用いた従来の NAT 検出手法（**図 12**）

4.2 実験用ネットワークでの検証実験

作成したプログラムを実験用ネットワークで動作させ，ホップ数が変化する動的ネットワークおよび IP アドレスが動的に変化するネットワークでも NAT 検出ができることを検証する．

4.2.1 実 験 環 境

検証実験を実施するネットワークの構成を**図 13** に示す．各ホスト間はイーサネットで接続されており，図中に示す IPv4 アドレスを保持している．このうち

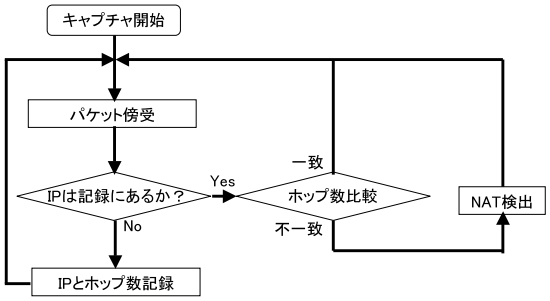

図 11 パッシブ NAT 検出手法
Fig. 11 The passive method.

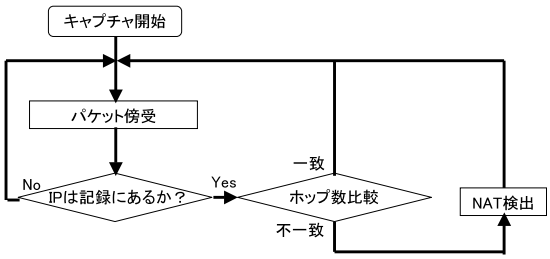

図 12 TTL を用いた従来の NAT 検出手法
Fig. 12 The previous method.

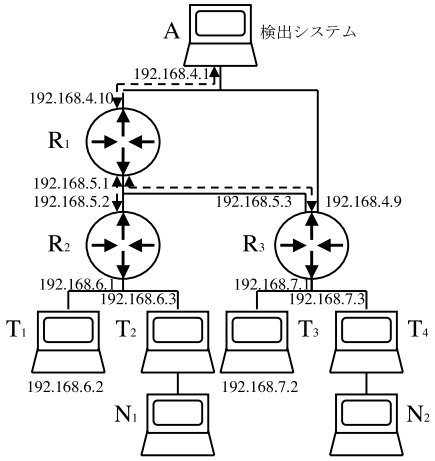

図 13 実験用ネットワークの構成
Fig. 13 The network for the experiment.

ホスト R_1，R_2，R_3 はルータであり，2 つ以上のインタフェースを備えている．ルータ R_2，R_3 からホスト A への経路はルータ R_1 を経由し，A から各ホストへの経路はこの逆順となっている．各ホストの OS には Fedora Core 5（Linux2.6 カーネル）を用い，ホスト A で作成したプログラムを動作させ，ホスト T_1，T_2，T_3，T_4 を監視対象ホストとする．監視対象のうち，ホスト T_2，T_4 は LAN の運用ポリシに反して NAT を実施しており，それぞれホスト N_1，N_2 の通信を中継している．すなわちホスト T_2，T_4 も 2 つのインタフェースを備えており，ルータとして動作している．

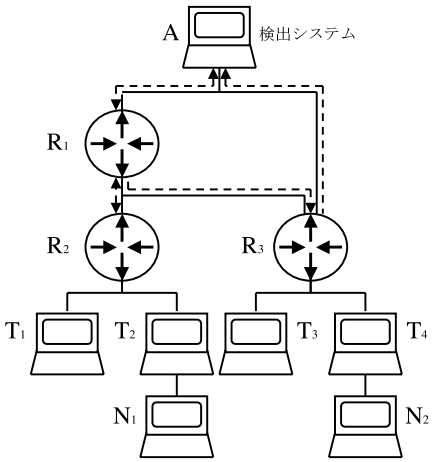

図 14 変更後の実験用ネットワークの経路
Fig. 14 The network after changing the route.

表 4 動的ネットワークでの実験結果
Table 4 Results in the dynamic routing network.

手法	経路	T_1	T_2	T_3	T_4
アクティブ	変更前	○	○	○	○
＋パッシブ	変更後	○	○	○	○
パッシブ	変更前	○	○	○	○
	変更後	○	○	△	△
従来手法	変更前	○	○	○	○
	変更後	○	○	×	×

4.2.2 動的ネットワークでの検証

ホスト T_1，T_2，T_3，T_4，N_1，N_2 から A にパケットを送信し，ホスト N_1，N_2 から送信されたパケットから，ホスト T_2，T_4 が NAT として検出されることを確認する．また，ホスト T_1，T_3 が NAT として検出されないことを確認する．次にルータ R_3 のルーティングテーブルを変更し，ホスト A への経路をルータ R_1 から直接 A に変更し，同様の結果が得られることを確認する．ルーティングテーブル変更後の実験用ネットワークの経路を**図 14** に示す．経路変更後，ホスト T_3，T_4 から A までのホップ数は 2 から 1 に変化する．ホスト A から各ホストへの経路には変更はない．検証実験の結果を**表 4** に示す．表中の○は検出可能，△は状況により検出可能，×は検出不能を表す．アクティブ NAT 検出手法では，経路を変更する前後で同様に，ホスト T_2，T_4 を NAT として検出し，ホスト T_1，T_3 を NAT として検出しないことを確認した．パッシブ NAT 検出手法では，傍受したパケットの送信時期が経路変更時期をまたがない場合には，アクティブ方式同様に検出が可能であった．従来手法では監視対象ホスト T_1，T_2，T_3，T_4 のホップ数を

2 として登録した場合，経路変更前はホスト T_3，T_4 を NAT として検出することができた．しかし，経路変更後はホスト T_2，T_4 は NAT として検出するが，ホスト T_3 も NAT として誤検出した．しかも，ホスト T_4 の NAT 検出は，ホスト T_4 から送信されたパケットによるものである．ホスト N_2 から送信されたパケットは，NAT として検出されなかった．この実験結果から，提案手法ではホップ数が変化する動的ネットワークでも NAT 検出が可能であることを確認した．

4.2.3 IP アドレスが動的に変化する環境での検証

ルータ R_2，R_3 で DHCP サーバを動作させ，ホスト T_1，T_2 に 192.168.6.2〜254，ホスト T_3，T_4 に 192.168.7.2〜254 の IP アドレスを動的に割り当てる．このときに，ホスト T_1，T_2，T_3，T_4，N_1，N_2 から A を経由するパケットを送信し，ホスト T_2，T_4 が NAT として検出されることを確認する．また，ホスト T_1，T_3 が NAT として検出されないことを確認する．検証実験の結果，どのように IP アドレスを設定してもホスト T_2，T_4 は NAT として検出され，ホスト T_1，T_3 は NAT として検出されないことを確認した．この結果から，提案手法では IP アドレスが動的に変化する環境でも NAT 検出が可能であることを確認した．

4.3 実環境への適用実験

作成したプログラムを大学校内の既存の LAN 内に設置し，当該 LAN に接続され実運用している機器を対象に実験を行い，提案手法の実効性を検証する．また，トレースパケットおよび応答パケットによるトラフィックの増加量を観測し，結果を考察する．

4.3.1 実験環境

実験 LAN の論理トポロジを**図 15** に示す．監視対象から検出システムまでのホップ数は 1 であり，動的ルーティングは実施していない．LAN に 100BASE-T イーサネットで接続され，IPv4 アドレスが割り当てられた 172 台の機器を監視対象とする．監視対象機器は PC，ワークステーション，NAT ルータ等で構成されており，OS の割合は Windows が 59.9%，MacOS が 14.0%，Linux が 21.5%，残りはその他の OS である．監視対象機器には 6 台の NAT 接続機器が含まれている．監視対象機器は大学校内で研究用に利用されており，その制御は利用者に依存するため，電源が入っていない場合もある．多くの監視対象からのパケットを得るため，ここでは LAN の DNS サーバに検出システムを設置した．DNS サーバの OS は AIX であり，実験 LAN 外からも約 5,000 台のホストに利用されている．DNS サーバで受信した DNS クエリの

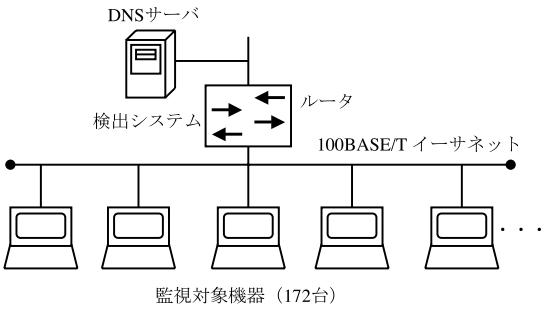

図 15 実験 LAN の論理トポロジ
Fig. 15 Logical topology of the experimental network.

表 5 NAT 検出結果とパケット観測数
Table 5 Results of NAT detection and number of packets.

回	アクセスホスト数	手法	NAT 検出率	パケット合計	DNS クエリ	トレースパケット	応答パケット
1	104 / 172	アクティブ＋パッシブ	4/5	2273704	1274763	5489	5429
		アクティブ	3/5				
		パッシブ	4/5			-	-
		従来手法	5/5				
2	111 / 172	アクティブ＋パッシブ	5/6	2312774	1193188	5840	5774
		アクティブ	4/6				
		パッシブ	4/6			-	-
		従来手法	6/6				
3	112 / 172	アクティブ＋パッシブ	4/5	2190326	1164770	5323	5257
		アクティブ	3/5				
		パッシブ	4/5			-	-
		従来手法	5/5				

TTL からホップ数を算出し，送信元 IP アドレスに対してトレースパケットを送信する．トレースパケットは，閉じているポートに TCP SYN パケットを送信し，RST+ACK フラグが立った応答パケットを得る方法を用いた．

4.3.2 実験内容

実験 LAN において先に示した 3 つの手法で検出システムを動作させ，NAT 接続機器の検出を試みる．動作時間は LAN の利用者が多い平日の午前 8 時 30 分から午後 5 時 30 分までを含む，午前 6 時から午後 6 時までの 12 時間とし，3 回実験を実施する．従来手法において検出のために必要なホップ数は 1 に設定する．

4.3.3 実験結果

運用実験の NAT 検出結果とパケット観測数を**表 5** に示す．アクセスホスト数は監視対象である 172 台のホストのうち，検出システムの動作中にアクセスがあったホストの数を表す．パケット合計は検出システムが傍受したパケットの総数であり，DNS クエリはそのうちの DNS クエリパケットの数である．トレース

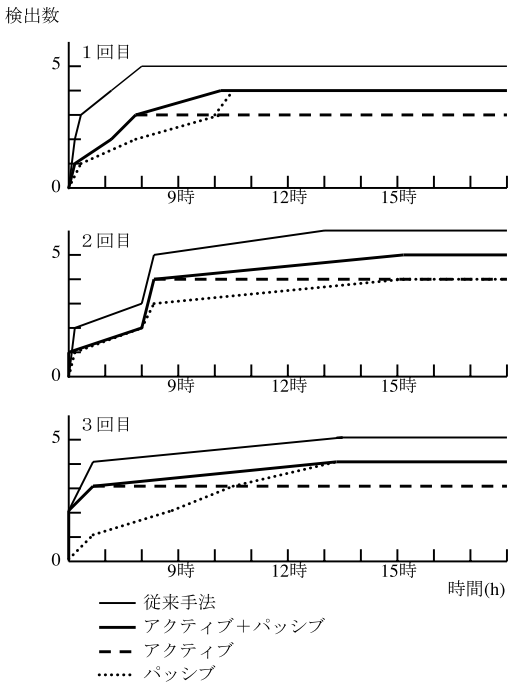

図 16 NAT 検出数と検出時間の推移
Fig. 16 NAT detection count versus the detection time.

パケットは傍受した DNS クエリの送信元ホストが監視対象に該当するホストに送信したトレースパケットの数であり，応答パケットはそれに対する応答パケットの数である．これらはいずれもパケット合計数に含まれている．監視対象には実際に 6 台の NAT 接続機器が含まれており，内訳は Linux を搭載した PC が 2 台と，ルータ専用機器が 4 台であった．検出率の分母は，6 台の NAT 接続機器のうち，検出システムにアクセスがあったホスト数である．各手法ごとの NAT 検出数と検出時間の推移を**図 16** に示す．提案手法についてはホップ数を事前に調査することなく，従来手法と同様に実環境での運用が可能であることが確認できた．

5. 評　　価

アクティブおよびパッシブ NAT 検出手法と従来の NAT 検出手法を比較検討し，提案手法を評価する．

5.1 検 出 精 度

TTL を用いた手法については良好であり，前提条件を満たしていれば，NAT 接続を検出することができる．アクティブ NAT 検出手法の検出率は 60.0〜66.7%であった．すべて NAT 接続機器の検出に失敗した原因は，応答パケットが得られないホストが存在したためであった．よってトレースパケットの種類を変更したり増やしたりすることで，検出精度を向上させることができると考えられる．2 回目においては応答パケットが得られないホストから，パッシブ NAT 検出手法によってホップ数の異なるパケットを傍受できたため，検出精度が向上していることを観測することができた．パッシブ NAT 検出手法については，NAT の内側に接続されたホストのパケットのほかに，NAT ホスト自身のパケットも傍受する必要があるため，すべての NAT 接続機器の検出には失敗しており，検出精度は 60.0〜80.0%にとどまっている．検出に失敗した機器はルータ専用機器であり，NAT ホスト自身であるルータ専用機器からのパケットを傍受することができなかったことが原因である．TTL を用いた手法の検出精度はネットワークの利用状況に依存するが，このようにアクティブ NAT 検出手法とパッシブ NAT 検出手法を組み合わせることで，検出精度を向上させることができる場合がある．IPid を用いた手法についてはパケットの観測量が精度に大きく影響し，実用段階にあるとはいい難い．

5.2 検 出 時 間

図 16 から TTL を用いた従来手法が最も早く，アクティブ+パッシブ NAT 検出手法，アクティブ NAT 検出手法，パッシブ NAT 検出手法の順に早く検出することができる．TTL を用いた従来手法ではあらかじめホップ数を記録しているため，パケット傍受後ただちに検出することができる．また，アクティブ NAT 検出手法では応答パケット受信後に検出することができる．従来手法との時間差は 1 秒未満であり，ほとんど検出時間では劣らない．パッシブ NAT 検出手法については，NAT ホスト自身からのパケットも傍受する必要があり，検出には時間を要する．今回の実験では従来手法に比べ，1 秒未満〜4 時間 11 分遅れて NAT 接続機器を検出している．検出時間についてもネットワークの利用状況に依存し，利用状況によっては検出時間は大きく変動する．IPid を用いた手法については，IPid 値の連続性が得られるまで分析を続ける必要があるため，やはり検出には時間を要する．

5.3 ホスト負荷

TTL を用いた手法についてはアルゴリズムが単純で，扱うデータ量が少ないため負荷は低い．今回の実験では約 5,000 台のホストが利用する DNS において 172 台のホストを監視対象として実験を実施し，DNS の動作に不具合を与えることなく NAT 接続機器を検出することができた．IPid を用いた手法についてはアルゴリズムが複雑であり，膨大なデータを処理する必要があるため負荷は高い．

5.4 ネットワーク負荷

アクティブ NAT 検出手法ではトレースパケットとその応答パケットの数だけトラフィックが増加する．表 5 のアクティブ NAT 検出手法におけるトラフィックの増加量について検討する．今回の実験ではトレースパケットと応答パケットによるトラフィックの増加はパケット数全体の 0.48〜0.50%であり，本手法がトラフィックに与える影響はきわめて小さいものと判断できる．また，応答パケットの受信率は 98.8〜98.9%であり，監視対象のパケットフィルタリングの実施率はあまり高くなかったことが推定される．TTL からホップ数を抽出する対象パケットを絞り込むことで，トラフィックの増加量を制御することができる．これらの相関関係はネットワークの利用状況に依存するものであり，適切にトラフィックを制御するためには，トラフィックをよく分析してトレース対象のパケットを制限する必要がある．パッシブ NAT 検出手法および従来手法は受動的に動作するため，トラフィックの増加は生じない．

5.5 システムの秘匿

アクティブ NAT 検出手法はトレースパケットを送信するため，システムの存在を秘匿することは難しい．受動的に動作する他の手法については，比較的容易にシステムの存在を秘匿することができる．

5.6 対応ネットワーク環境

アクティブ NAT 検出手法および IPid を用いた手法は，DHCP 環境や経路が動的に変更される場合にも適用することができる．また，パッシブ NAT 検出手法では，傍受したパケットが経路変更前後にまたがって送信された場合には対応できないが，あらかじめ IP アドレスとホップ数の記録は必要とせず，DHCP 環境には適用することができる．このように，提案手法は幅広いネットワーク環境に適用可能である．あらかじめ監視対象 IP アドレスとホップ数の記録が必要な従来手法は，DHCP 環境や経路が動的に変更される場合に適用することができない．

5.7 対 応 OS

提案手法を含む TTL を用いた手法はすべての OS に適用可能である．また今回の実験によって，パッシブ NAT 検出手法については PC による NAT は比較的早期に検出できるが，ルータ専用機器の検出には時間がかかり，検出できない場合もあることが分かった．IPid を用いた手法は Linux および OpenBSD に適用することができない．

5.8 実装依存性

TTL を用いた手法は前提条件が少なく，その条件も RFC に規定されている仕様であるため，実装依存性は低いといえる．IPid を用いた手法は判定できる OS が限定されており，しかも IPid が 1 ずつ増加するという仕様には規定がないため，実装依存性が大きいといえる．

5.9 ま と め

以上の考察から，提案したアクティブ＋パッシブ NAT 検出手法は多少のトラフィックの増加は生じるものの，幅広いネットワーク環境において低負荷で NAT 検出が可能な実効的手法であると考える．TTL を用いた各 NAT 検出手法にはそれぞれ特徴があり，ネットワーク環境への適不適がある．ネットワーク環境や運用ポリシを考慮し，効果的な運用をするために最善の検出手法を選択する必要がある．

5.10 応 用

提案手法の応用としては，ネットワークを常時監視し，不正 NAT 接続を検出して無力化するシステムが考えられる．TCP RST によるコネクションの強制切断等の手法を用いることで，通信経路外からの通信の無力化も可能である．また，TTL による NAT 検出手法は IPv6[9] ネットワークにおいても，TTL の代わりに HOP LIMIT を用いることで，NAT やトランスレータ[10] の検出に適用可能である．

6. おわりに

本研究では既存の NAT 検出技術について考察し，IP ヘッダの TTL を能動的に取得して NAT 検出を行う手法を提案した．作成したプログラムを用いて，ホップ数が既知でない場合や IP アドレスが動的に変化するネットワークにおいても NAT 接続された機器を検出することが可能であることを実証した．さらに，作成したプログラムを実環境に適用することにより，その実効性を確認した．本手法は，ホップ数が既知でない場合においても，高い精度で NAT 接続された機器を検出することができる．また，トラフィックはわずかに増加するが，ホップ数が変化する動的ネットワークにも IP アドレスが動的に変化するネットワークにも適用できる．ただし，能動的に TTL を取得できるかどうかは監視対象ホストに依存する．確実に監視対象ホストから能動的に TTL を取得する手法については今後の課題である．

提案した NAT 検出手法は，OS やルータの実装等のいくつかの前提条件に基づいており，検出システムの存在が明らかな場合，意図的に検出を回避される可能性もある．たとえば，TTL を減算しない NAT ルータは提案手法では検出することができない．既存の

NAT 検出手法についてもそれぞれの前提条件があるため，検出率は OS の実装に依存してしまう．このため，OS の実装に依存しない技術の研究や，検出システムの存在や検出アルゴリズムを秘匿することも重要であると考える．

参 考 文 献

1) Srisuresh, P. and Egevang, K.: Traditional IP Network Address Translator RFC3022 (2001).
2) Information Sciences Institute, University of Southern California: Internet Protocol RFC791 (1981).
3) Phaal, P.: Detecting NAT Device using sFlow. http://www.sflow.org/detectNAT/
4) Droms, R.: Dynamic Host Configuration Protocol RFC2131 (1997).
5) Dostoevsky, F.: Remote OS Detection via TCP/IP Fingerprinting (2nd Generation). http://insecure.org/nmap/osdetect/
6) Bellovin, S.M.: A Technique for Counting NATted Hosts, *Proc. Internet Measurement Workshop 2002*, pp.267–272 (2002).
7) 高橋輝壮，甲斐俊文，篠原克幸：IPid を用いた NAT 検出手法の考察，情報処理，2006-CSEC-32, Vol.2006, No.26, pp.97–102 (2006).
8) Kohno, T., Broido, A. and Claffy, K.C.: Remote physical device fingerprinting, *IEEE Symposium on Security and Privacy* (2005).
9) Deering, S. and Hinden, R.: Internet Protocol, Version 6, RFC2460 (1998).
10) Hagino, J. and Yamamoto, K.: An IPv6-to-IPv4 Transport Relay Translator, RFC3142 (2001).

(平成 19 年 3 月 13 日受付)
(平成 19 年 7 月 3 日採録)

三村 守（学生会員）

1978 年生．2001 年防衛大学校情報工学科卒業．同年海上自衛隊入隊．現在，防衛大学校理工学研究科前期課程在学中．ネットワークセキュリティに興味を持つ．

中村 康弘（正会員）

1959 年生．1987 年防衛大学校理工学研究科オペレーションズリサーチ専攻修了．同年同大学校電気工学科助手兼共同利用電子計算機室勤務．1994 年同大学校情報工学科助教授．2005 年同大学校学術情報センター情報通信技術研究室勤務（現職）．2007 年同大学校情報工学科准教授（現職）．コンピュータネットワークセキュリティ，電子透かし，パターン認識に関する研究に従事．工学博士．電子情報通信学会，画像電子学会各会員．