高品質インターネット　4－1
(2002. 7. 26)

MACトレースバック：Hash-Based IPトレースバック拡張方式の提案

櫨山　寛章　　大江　将史　　門林　雄基

† 奈良先端科学技術大学院大学　〒630–0101 奈良県生駒市高山町 8916–5
E-mail: †{hiroa-ha,masafu-o,youki-k}@is.aist-nara.ac.jp

あらまし　分散型サービス妨害攻撃（Distributed Denial of Service Attack）への対策としてIPトレースバック技術が注目されている．IPトレースバック技術のなかでもパケット単位での追跡が可能なHase-Based IPトレースバック[2]に本研究では注目した．先行研究はAS（Autonomous System）内部のトレースバック技術としての有効性が期待できる．しかし，先行研究では攻撃の流入点となっているルータまでしか特定できず，攻撃ノードの特定には至っていない．そこで，本研究では攻撃ノードの特定に役立つ情報も記録するようにHash-Basd IPトレースバックの拡張方式を提案する．
キーワード　分散サービス妨害攻撃，Hash-Based IPトレースバック，MACトレースバック

MAC Traceback : An extention of Hash-Based IP Traceback

Hiroaki HAZEYAMA, Masafumi OE, and Yuuki KADOBAYASHI

† Graduate School of Information Science, Nara Institute of Science and Technology　Takayamacho 8916–5, Ikoma-si, Nara, 630–0101 Japan
E-mail: †{hiroa-ha,masafu-o,youki-k}@is.aist-nara.ac.jp

Abstract　Hash-Based IP Traceback is a technique to generate audits trails for traffic within a network. It can detect the true Attack path of Distributed Denial of Service attack. This technique has an effectiveness to find a route to the attacker in an Intra-Domain network. However, it cannot detect true sources of the attack because of its algorithm of tracing. It only detect the nearest router of attack node. So, we propose an extended technique of Hash-Based IP Traceback to get datalink addresses to help detecting the true source of an attack.
Key words　Distributed Denial of Service Attack, Hash-Based IP Traceback, MAC Traceback

1. はじめに

近年，インターネット上で行われるサービスを妨害する攻撃が頻発するようになり，深刻な被害を与えている．これはサービス妨害攻撃（DoS攻撃：Denial of Service Attack）[?]と呼ばれ，インターネットの脅威となっている．

サービス妨害攻撃で典型的な攻撃手法は，攻撃フローと呼ばれる大量のトラフィックを被害者（攻撃目標）に対して送信し，サーバ資源やネットワーク資源を消費させる攻撃である．また，近年，分散的に多地点からサービス妨害攻撃を行う分散型サービス妨害攻撃（Distributed Denial of Service Attack）が攻撃手法として用いられるようになり，その被害は甚大なものとなっている．

この攻撃手法による被害を最小限に抑えるには，攻撃フローの発生源（攻撃ノード）を短時間で特定しネットワークから隔離することが必要である．しかし，一般的に攻撃に用いられるIPパケットの送信元アドレスは偽装されている．よって，送信元アドレスを用いたtracerouteやルーティング情報を用いた攻撃ノードの特定は不可能である．このため，これまで攻撃フローの追跡はルータの持つフィルタリング機能やサービス妨害検知機能などを使い，手作業によって行っている．しかし，インターネットは商用ISPや企業，研究・学術機関といった様様なポリシーを持った機関が相互接続して成り立っており，その相互接続は国際的に行われている．そのため，手動追跡では，その境界を越えた追跡を行う際のコストや国際的な協力が必要となり，追跡時間の短縮に対する大きな障害となっている．

この問題点の解決として，IPトレースバック技術が注目されている．IPトレースバックとは，攻撃フローが通過した経路（攻撃経路）を特定する技術であり，偽装された送信元アドレスであっても真の発信元の特定を目指す技術である[3]．

現在提案されているIPトレースバック技術のなかでも，Hash-Based IPトレースバック[2]はパケット単位での追跡が可能なことから，AS内部の攻撃ノードを特定する技術としての有効性を期待できる．しかし，先行研究では攻撃フローの流

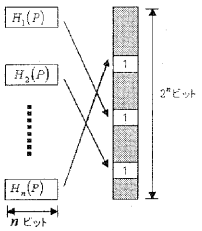

図 1　ダイジェストテーブル

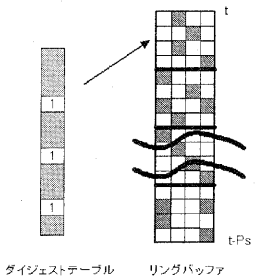

図 2　リングバッファへの保存

入点となっているルータまでの攻撃経路を特定するに留まっており，攻撃ノード自体を特定するには至っていない．

そこで，本研究では，流入点であるルータ以降の追跡に役立つ情報も記録する Hash-Based IP トレースバックの拡張方式を提案する．

2. 関連研究

この章では，先行研究である Hash-Based IP トレースバック [2] の概要と問題点を指摘する．

2.1 Hash-Based IP トレースバックの概要

Hash-Based IP トレースバック [2] は Snoeren らによって提案された手法である．この手法では，ルータ上のエージェント，もしくはルータの外部に設置する機器 [4] によって，ルータ上を通過するパケットの記録を効率よく行うことにより，攻撃フローの通過した経路を特定する手法である．この手法ではルータ上で Bloom Filter [5] と呼ばれるデータ構造を用いる．経路上で IP ヘッダの不変な部分（20 バイト）とペイロードの先頭部分（8 バイト）である P を入力値として k 個の異なるハッシュ関数 H を用いて得た k 個のハッシュ値（パケットダイジェスト）を生成する．そして，図 1 に示すように，パケットダイジェストの示す値をダイジェストテーブルと呼ばれる 2^m ビットのビットマップに変換して記録する．

ダイジェストテーブルは記録するパケットの許容量を超えるまで，もしくは更新時間までパケットの通過記録に使用される．更新時にダイジェストテーブルは使用された k 個のハッシュ関数と使用された時間とともにルータ上のリングバッファに記録される（図 2）．k 個のハッシュ関数はルータごとに独立して決められており，また，ダイジェストテーブルの保存時に更新されることで IP パケットのハッシュ衝突をできる限り避ける工夫がなされている．

ルータ上の追跡対象パケットの通過記録は，パケットが通過したと思われる時間間隔で使用された k 個のハッシュ関数にパケットを通し，ハッシュ関数に対応するダイジェストテーブルとの比較により検証する．k 個のハッシュ値の示す位置にあるダイジェストテーブルのビットが一つでも 0 の場合はそのパケットは記録されなかったとみなされ，すべてのビットが 1 ならば，高確率でルータ上を通過したことになる．この手法ではデータ収集と問い合わせのアーキテクチャSPIE（Source Path Isolation Engine）が用意されている．攻撃経路の追跡はSPIE の追跡マネージャが適用範囲内の各ルータに対象パケットの通過記録を検証するように要求し，図 3 に示すように被害ノードの上流ルータから 1 ホップずつ特定していく．こうして攻撃経路を構成し，攻撃フローの流入点となっている最も攻撃ノードに近いルータを特定する．

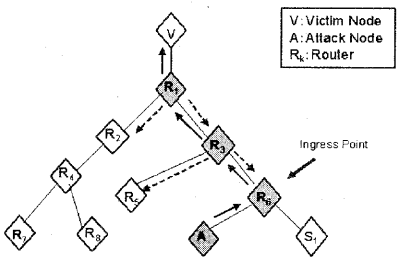

図 3　Hash-Based IP トレースバックにおける攻撃経路の特定

この手法はパケット単位での追跡が行えることから，攻撃ノードに近づくにつれて相対的にトラフィック量の減る分散型サービス妨害攻撃 [1] の攻撃ノードが組織内部の LAN にいる場合においても攻撃経路の特定にその有効性が期待できる．

2.2 問題点

先行研究では，適用範囲内において攻撃ノードに最も近いルータまで特定でき，組織内部のネットワークにおける追跡技術としての有効性を期待できる．組織内部にいる攻撃ノードの追跡を行う場合，追跡により特定する流入点のルータは末端のルータとなる．

しかし，ダイジェストテーブルに記録される情報はパケットが通過した事実のみである．このため，末端ルータが複数の LAN と接続している場合，どの LAN に攻撃ノードが存在しているのかは即座にわからない．また，ルータが下流に一つの LAN としか接続していない場合でも，その LAN が無線ネットワークや広域 LAN 型サービス網などの広域なレイヤ 2 ネットワークを構成している場合，その大きさのため，攻撃ノードの特定には労力と時間がかかる．特定した末端ルータでフィル

タリングをかけたとしても，攻撃が収束するまでの間はLAN上の他のノードのトラフィックに対して影響を与えてしまう．フィルタリングによる影響を最小限に抑えるには，短時間で攻撃ノードを特定し，ネットワークから隔離する必要がある．そのためには，攻撃ノードの存在するLAN，および攻撃ノードを識別する情報を記録するようにHash-Based IPトレースバックを拡張する必要がある．

3. 拡張方式

本章では，2章で指摘したHash-Based IPトレースバック[2]の問題点を解決する拡張方式を提案する．

提案する拡張方式では，ルータ上を流れたパケットのダイジェストと対応させて，パケットが送られてきたネットワークの情報，パケットを送信したノードに関する情報を記録する．

3.1 前提条件

議論を進めていく上で，問題を簡単にするために，前提条件を仮定する．

- パケットの流入点となっているルータまでは従来のHash-based IPトレースバックの方式で特定する．
- 流入点となっているルータは末端のルータであり，接続する近隣ルータでパケットが通過していないことが従来方式で確認されている．
- 末端ルータに接続している下流のLANはMACアドレスで宛先解決をしており，ブリッジMIB[6]を採用している．

3.2 記録する情報

まず，記録する情報について考察する．末端の流入点となっているルータまでの攻撃経路を特定できた場合，攻撃経路の追跡を続けるために必要となる情報は，攻撃ノードの存在するLANを識別する情報と攻撃ノードを識別する情報である．LANを識別する情報を取得することで追跡範囲の絞込みを行い，攻撃ノードを識別する情報により攻撃ノードの特定を行う．流入点となっているルータの接続するネットワークがイーサネットネットワークの場合，この2つの情報は流入パケットのイーサネットフレームにある宛先MACアドレスと送信元MACアドレスから得ることができる．

MAC（Media Access Control）アドレスはIEEEで管理されており，ネットワークでホストを識別するために設定されるハードウェアアドレスである．イーサネットネットワークでは各ネットワークカードに固有のMACアドレスを割り当て，イーサネットネットワーク上でのアドレス解決を行いパケットを転送している．

ルータに流入したパケットのイーサネットフレームにおいて，宛先MACアドレスはルータのパケットが流入したネットワークインターフェースに付けられたMACアドレスである．つまり，宛先MACアドレスからパケットが到着したネットワークインターフェースがわかり，そのインターフェースが接続するネットワークが判明する．また，送信元MACアドレスはパケットを送信したノードのネットワークインターフェースに付いているMACアドレスであり，固有に識別されるものである．つまり，送信元MACアドレスはノード自身を固有に識別できる情報である．

NI-ID	MACアドレス
0001	a
0010	b
0011	c
0100	d

(a)

Port-ID	Port-番号
0000001	1
0000010	2
0000011	3
0000100	4

(b)

図4 NI-IDテーブル（a），Port-IDテーブル（b）

ダイジェストテーブルに記録されたパケットとの対応を取るためにパケットのハッシュ値の中で最初に生成されたものを代表ハッシュ値として選出し，代表ハッシュ値と2つのMACアドレスを組にしてデータベースに保存する．これをルータに流入する全てのパケットに対して行う．データベースはダイジェストテーブルと対応しており，ダイジェストテーブルとともに保存され，新たなデータベースが次のダイジェストテーブルのために設けられる．

この方式で記録すると，追跡時においてダイジェストテーブルの検査に使用したハッシュ値からデータベースに保存されている宛先MACアドレスと送信元MACアドレスを取得でき，2つのMACアドレスを元にノードの存在するネットワークとノードのMACアドレスを特定できる．しかし，ダイジェストテーブルの記録用に生成されるハッシュ値の大きさを24ビットとした場合，パケットあたり記録する情報量の増分は，代表ハッシュ値24ビット，宛先・送信元MACアドレスが各48ビット，合計120ビットになる．パケットの平均長を1000ビットと仮定すると，このデータベースに必要な記憶容量はリンク容量の約12.0%となる．この値は1Gbpsのリンクに対して，秒間15Mバイト必要となり記憶容量に対する負荷が高い．そこで，MACアドレスと同じ意味を持つ別の情報を記録する．

3.2.1 攻撃ノードの存在するLANを識別する情報

流入パケットの宛先MACアドレスがルータのネットワークインターフェースが持つMACアドレスを指すことから，攻撃パケットの流入したルータのネットワークインターフェースを割り出せる．しかし，MACアドレスは48ビットで表現されるため，記録においてその情報量は大きく負荷が高い．

そこで，ルータの各ネットワークインターフェースを識別するネットワークインターフェース識別子（NI-ID）を宛先MACアドレスの代わりに記録する．ルータにおいて，各ネットワークインターフェースに対し固定でNI-IDを割り振り，NI-IDとネットワークインターフェースの持つMACアドレスの変換表（NI-IDテーブル）を保持する．（図4（a））

また，NI-IDは8ビットで表現され，最大256のネットワークインターフェースを識別できる．今日のルータにおいて，物理インターフェースだけでなく，VPNやVLANといった論理インターフェースも用いるため，この値を用いる．

3.2.2 攻撃ノードの特定を行うための情報

次に流入パケットの送信元アドレスと同じ意味をもつ情報を考える．これは，ルータの各ネットワークインターフェースが

接続しているスイッチのポート番号で表現する．

スイッチでは，MACアドレスと送信先ポートの対応表がフォワーディングデータベースに記録されている．この対応表は，イーサネットにおいてパケットを受信したときに，そのパケットが到着したポートとそのパケットに含まれている送信元MACアドレスを元にしてスイッチ上で自動的に生成される．パケットを転送する際に今度は記憶されている送信元MACアドレスを宛先MACアドレスとして，一致するMACアドレスに対応するポートに対してのみパケットを転送する．つまり，接続しているスイッチのフォワーディングデータベースにルータに流入したパケットの送信元アドレスと同一のMACアドレスがその送信先ポートに記憶されている．この送信先ポートが追跡対象パケットの通過したポートである．

よって，流入パケットの送信元MACアドレスを元にスイッチのフォワーディングデータベースに記録された同一のMACアドレスを検索し，その送信先ポートを調べることで，スイッチのどのポートをパケットが通過したかを特定でき，特定したポートに攻撃ノードが接続していることがわかる．このポートとMACアドレスの対応表はスイッチのブリッジMIB[6]から取得できる．

また，スイッチのアドレスラーニングの特性から，MACアドレスが偽装されていた場合においてもスイッチ上のフォワーディングデータベースに一定時間記録されるため，対応するMACアドレスが記憶されているポートを攻撃パケットが通過したことが判明する．

パケットの通過したポートを表す，記録に用いる情報にはポート番号識別子Port-IDを用いる．このPort-IDはルータの各インターフェースが接続するスイッチのポートに対して固定して割り振られる．まず，ルータが接続しているスイッチのMIB情報からポートと送信先MACアドレスの対応表を取得する．そして，各ポートに対してPort-IDを割り当て，Port-IDとポート番号の変換表（Port-IDテーブル（図4 (b)）），Port-IDとMACアドレスの対応表（Port-MACテーブル）を保持する．（図5）この対応表はルータの各ネットワークインターフェースごとに個別に作成され，ネットワークインターフェース識別子NI-IDとともに保持される．

このため，ルータの複数のネットワークインターフェースが同じスイッチに接続していても，NI-IDとあわせて表現することで，流入パケットがどのネットワークインターフェースとどのスイッチのポートを通過したかが一意に識別できる．ポート番号識別子は8ビットで表現し，最大256ポートを識別できる．この値は，末端のLANに使われるスイッチで256を超えるポート数を持つスイッチを利用するとは考えにくいことから設定した．また，ポート数が256を超える場合，4つのポートに同じPort-IDを割り当てて，Port-IDとポートの変換表から攻撃パケットを通過した可能性のあるポートを割り出す．

3.3 記録方法

ルータが接続するネットワークと攻撃ノードの通過したポートを識別する情報として，それぞれネットワークインターフェース識別子とポート番号識別子を設定した．しかし，先に説明したデータベースの形でこれらのノードに関する情報を保存すると，パケットあたり40ビット，リンク容量の約4.0%の記憶容量必要であり，1Gbpsのリンクに対し毎秒5Mバイト記録することになる．そこで，必要となる記憶容量を削減するために，追加するノード情報に対してもBloom Filter[5]を用いる．

NI-ID	Port-ID	MACアドレス
0001	0000001	A
	0000001	B, F
	0000011	C
	0000100	D, E, G

図5 Port-MACテーブル

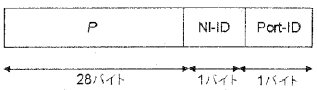

図6 ノード情報用ダイジェストテーブルの入力値D

2つの識別子はパケットの通過記録を保持するダイジェストテーブルに対応させた，ノード情報用のダイジェストテーブルに記録する．まず，ルータに流入したパケットのイーサネットフレームにある宛先MACアドレス，送信元MACアドレスに対応するネットワーク識別子NI-IDとポート番号識別子Port-IDを取得する．詳しく説明すると，NI-IDテーブル（図4 (a)）から該当する宛先MACアドレス，つまりパケットの流入したネットワークインターフェースのMACアドレスに割り振られたNI-IDを取得する．そして，取得したNI-IDを持つスイッチのPort-MACテーブル（図5）から該当する送信元MACアドレスに一致するMACアドレスと組になっているPort-IDを取得する．

つぎに，パケット用のダイジェストテーブルに用いられるハッシュ関数への入力値PとNI-ID，Port-IDを組み合わせた計30バイトのDをノード情報用の入力値として生成する．そして，図6に示すように，入力値Dについてノード情報用のk個のハッシュ関数を用いてハッシュ値を出し，その値をノード情報用のダイジェストテーブルに変換して記録する．

ノード情報用のハッシュ関数はパケット用のダイジェストテーブルに用いられるハッシュ関数と同じ大きさのハッシュ値を生成する．また，用いるハッシュ関数の数はパケット用のダイジェストテーブルと同じ個数をパケット用のハッシュ関数とは独立して用意する．

ノード情報用のダイジェストテーブルはパケット用のダイジェストテーブルと対応しており，ダイジェストテーブルの更新時にパケット用のダイジェストテーブルと使用ハッシュ関数の組とともに，ノード情報用のダイジェストテーブルとそれに

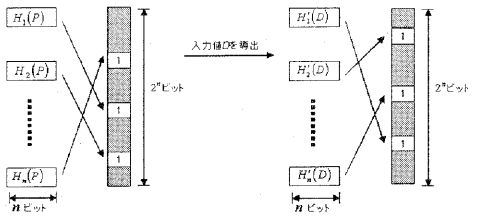

図 7　ノード情報用ダイジェストテーブル

使用されたハッシュ関数の組を対応させ，同じタイムスタンプを押して保存される．

3.4　追跡時の情報の取得

追跡時においてノード情報の取得は次のように行われる．まず，追跡対象時間から使用されたハッシュ関数を割り出し，追跡対象パケットのハッシュ値を計算してパケット用のダイジェストテーブルに記録されているかを調べる．パケット用のダイジェストテーブルに記録されていた場合，パケットダイジェストの入力値と同じ IP ヘッダの不変な部分 28 バイト，ネットワークインターフェース識別子，ポート番号識別子の組を入力値としてハッシュ関数を作成し，ノード情報用のダイジェストテーブルに記録されているかを調べる．

ノード情報用のダイジェストテーブルに記録されていた場合，ネットワークインターフェース識別子が示すルータのネットワークインターフェースとポート番号識別子が示す接続先のスイッチのポートを追跡対象パケットが通過したことになる．記録されていない場合，インターフェース識別子とポート番号識別子の値を変えて再びノード情報用ダイジェストテーブルを調べる．

4.　考　　察

4.1　必要となる記憶容量

新たにノード情報用のダイジェストテーブルを保持することで増加する記憶容量について考察する．

拡張方式で使用するノード情報用のダイジェストテーブルは使用ハッシュ関数以外はすべて同じパラメータの値を使うので，同じ大きさのダイジェストテーブルを 2 つ保持するため，必要な記憶容量は 2 倍となる．先行研究 [2] で示されている，使用ハッシュ関数の数を 3，ダイジェストテーブルの大きさ m を 2M バイト，Bloom Filter [5] の最大記録パケット数 n を $n = m/5$，パケットの平均長を 1000 ビットとした場合，パケット用のダイジェストテーブルに必要な記憶容量はリンク容量の約 0.5 %であるので，拡張方式に必要な記憶容量はリンク容量の約 1.0 %となる．末端ルータが 1Gbps のイーサネットを 3 本収容する場合，必要となる記憶容量は秒間約 4M バイトになる．

必要な記録容量が 2 倍となるため，負荷は高い．しかし，拡張方式を適用するのは組織内部の末端に位置するルータのみである．また，拡張方式のノード情報は攻撃ノードを特定し攻撃の早期収束を目的として記録するものなので，攻撃発生時以外は記録する必要がない．そこで，攻撃時に稼動する拡張機能として使用する．

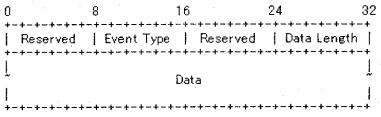

図 8　Event Reply フィールド

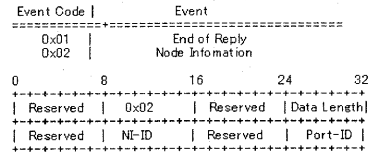

図 9　Event Reply フィールドの拡張

5.　おわりに

攻撃ノードの特定に必要となる情報を保持するように Hash-Based IP トレースバックの拡張方式，MAC トレースバックを提案し，追加情報の記録による記憶容量の増分をリンク容量の 0.5 %までに抑えた．

拡張方式ではノード情報の特定にネットワーク識別子とポート番号識別子を入力値として，追跡対象パケットが通過したルータのネットワークインターフェースとスイッチのポートを検索する．このとき，検索回数は，最悪の場合，ルータの各ネットワークインターフェースが接続するスイッチのポート数の総和となる．実装において，この検索にかかる時間がリングバッファの更新時間内に終わる，実用に耐えうるものであるかの検証が必要となる．

また，Port-MAC テーブル（図 5）の更新時間についての考察が必要である．スイッチのポートに記憶される MAC アドレスは一定時間保持される．攻撃ノードが MAC アドレスを偽装していたとしても一定時間は保持される．MAC アドレスが対応表に記録されていない度に更新を行うと，更新にかかる時間がボトルネックになると予測され，送信元 MAC アドレスをパケットごとに変化させる攻撃が安易に予測される．しかし，そのような攻撃手法をとられた場合，一定間隔での対応表の更新によって偽装された MAC アドレスは対応表に記録され，急激に MAC アドレスが増加したポートに攻撃ノードが接続していることが判明する．また，分散サービス妨害攻撃の場合，連続してパケットが送信されているため，対応表にない場合は記録せず，一定時間の更新で十分に対応できる．ただし，ping of

death などの単一パケットによりシステムの脆弱性をつく攻撃手法に MAC アドレスの変更を組み合わせて行われた場合，対応できないため，Port-MAC テーブルの更新時間についてより詳細な考察が必要である．

実装において，追加情報の記録だけでなく，SPIE アーキテクチャでのメッセージプロトコル[7] に対しても変更を加えなければならない．ここでは，未定義の Event Reply フィールド（図 8）を利用し，拡張機能が稼動していた場合にインターフェース識別子とポート番号識別子を送信する（図 9）．

今後，実装し上記にあげた評価項目の検証をを行う予定である．

文　献

[1] Computer Emergency Response Team. CERT Incident Note IN-99-07, "Distributed Denial of Service Tools", http://www.cert.org/incident_notes/IN-99-07.html

[2] Snoren, A. C., Partridge, C., Sanches, L. A., Jones, C. E., Tchakountio, F., Kent, S. T., and Stayer, W. T., "Hash-Based IP Traceback" in Proceedings of SIGCOMM '01, San Diego, 2001.

[3] Savage, S., Wetherall, D., Karlin, A., and Anderson, T. Practical network support for IP traceback. In Proc. ACM SIGCOMM'00 (Aug. 2000), pp. 295-306.

[4] Sanchez, L. A., Milliken, W. C., Snoeren, A. C., Tchakountio, F., Jones, C. E., Kent, S. T., Partridge, C., and Strayer, W. T. Hardware support for a hash-based IP traceback. In Proc. Second DARPA Information Survivability Conference and Exposition (June 2001).

[5] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. Communications of ACM 13, 7 (July 1970), 422-26.

[6] E. Decker, P. Langille, A. Rijsinghani, K. McCloghire, "Definitions od Managed Objects for Bridges" RFC 1493 http://www.ietf.org/rfc/rfc1493.txt, 1993

[7] C. Partridge, C. Jones, D. Waitzman, A. Snoeren New Protocols to Support Internet Traceback http://www.ir.bbn.com/documents/internet_drafts/draft-partridge-ippt-discuss-00.txt (November 2001) (expired)