推薦論文

DNSの正引き応答を利用したパケットフィルタリングによるコンピュータワームの増殖抑制

岡本 剛†

本論文では，DNSの正引き応答を利用したパケットフィルタリングを提案する．インターネット上のホストの通信において，ユーザはDNSを利用するが，コンピュータワームはDNSを利用しない．この特徴に基づいて，提案手法は，DNSの正引き応答から得られたIPアドレスへの送信を許可し，それ以外のIPアドレスへの送信をWilliamsonのウイルス・スロットル（*virus throttle*）により処理する．シミュレーション実験では，提案手法は，ウイルス・スロットルによって発生する送信の遅延時間を飛躍的に短縮し，コンピュータワームの増殖を速やかに阻止できることを示した．さらに，DNSを利用しないコンピュータワームであれば，たとえ感染しても，提案手法はコンピュータワームの増殖だけを阻止し，ユーザは継続して通信できることを示した．

Packet Filtering Using DNS Responses against Worm Propagation

Takeshi Okamoto†

I propose a new approach to stop the propagation of worms. The approach discriminates between user's packets and worm's packets according to IP addresses contained in DNS responses. The method takes advantage of the fact that users accessing the Internet usually use DNS servers, while computer worms do not. Outgoing packets to resolved IP addresses are permitted, while outgoing packets to unresolved IP addresses are restricted (delayed or stopped) by the *virus throttle* method. Simulation results showed my approach shortens the delay time and the time to stop worm propagation, compared with the *virus throttle* method. Furthermore, my approach enables users to connect to only resolved IP addresses, even if their computers are infected with a computer worm which does not use a DNS server.

1. はじめに

コンピュータワーム（以下，ワーム）とは，ネットワーク上で利用されるサービスのセキュリティホール（欠陥）を不正に利用してネットワーク上を自己増殖するプログラムである[1]．ワームは，ユーザがワームのプログラムを実行しなくてもワーム自身の機能により増殖できるが，コンピュータウイルスは，ユーザがコンピュータウイルスを実行しなければ増殖できない．ワームの増殖にはユーザが介在しないため，ワームはウイルスに比べて驚異的な速度で広がる．たとえば，2001年7月19日（UTC）頃に放たれたCodeRedv2は14時間に359,000台以上のコンピュータに感染し[2]，2003年1月25日（UTC）頃に放たれたSlammerは30分間に75,000台以上のコンピュータに感染したと報告されている[3]．

ワームの対策は，「侵入阻止」と「増殖阻止」に分けられる．前者は，侵入検知ソフトやアンチウイルスソフトなどのセキュリティ対策ソフトで利用されている不正検出に基づく手法（たとえば，文献4)）やIngressフィルタリング[5]が含まれる．後者は，Egressフィルタリング[5]や，ハニーポットを応用したラブレア[6]やパケットフィルタリングを応用した．Williamsonのウイルス・スロットル[7)~9)]がある．

アンチウイルスソフトは，ネットワークからコンピュータに入力されるデータを監視し，ワームの侵入を防ぐ．しかし，アンチウイルスソフトは，主に過去に発見されたワームの情報に基づいているため，新種のワームを検出するには，そのワームを検出するためのデータが必要となる．そのデータが多数のユーザに

† 神奈川工科大学情報学部情報ネットワーク工学科
Department of Network Engineering, Kanagawa Institute of Technology

本論文の内容は2003年12月のコンピュータセキュリティ研究会にて報告され，CSEC研究会前主査により情報処理学会論文誌への掲載が推薦された論文である．

配布されるまでに，ワームが放たれると，大規模の感染を避けられない．

ウイルス・スロットルは，ユーザとワームに関する通信の違いに基づいて，ワームの増殖を阻止する．Williamsonが着目した違いは，次の2つに要約できる．

- ワームに感染したコンピュータが単位時間に接続するホスト数は，未感染時と比べると非常に多い．
- ユーザは最近に接続したホストと通信する傾向があるが，ワームは過去に接続したことがないホストに接続しようとする．

ウイルス・スロットルは，この違いに基づいて，ワームから送信されたと考えられるパケットの送信を遅延させる．遅延させたときに発生するパケットの待ち行列（キュー）が，閾値以上の長さに達したとき，ウイルス・スロットルはすべての送信を停止する．文献8)によると，Linux上で実装されたウイルス・スロットルは，Nimdaの増殖を0.25秒，Slammerの増殖を0.02秒で阻止した．

しかし，ウイルス・スロットルでは，ワームに感染していなくても遅延することがある．文献9)によると，送信されたTCPの全SYNパケット中，2.14%のパケットが遅延し，1接続あたり最大で5秒間も遅延する．また，ワームに感染すると，すべての送信が拒否されるため，たとえ感染を発見しても，セキュリティホールを修復するプログラムをダウンロードできない．感染したコンピュータしか所有しないユーザにとって，ワームの駆除やセキュリティホールの修復は容易でない．

そこで本論文では，ウイルス・スロットルによる送信の遅延時間を短くし，ワームに感染した後でも，ユーザによる通信だけを通す手法を提案する．本手法が着目するユーザとワームに関する通信の違いは，Williamsonが着目した2つの違いに加えて，「ユーザの通信にはDNSサーバとの通信が含まれるが，ワームの通信には含まれない」という違いである．本手法は，これらの違いに基づいて，ユーザとワームの通信を識別する．

シミュレーション実験では，本手法は，ウイルス・スロットルによって発生する送信の遅延時間を飛躍的に短縮し，ワームの増殖を速やかに阻止できることを示す．さらに，DNSを利用しないワームであれば，たとえ感染しても，本手法はワームの増殖だけを阻止し，ユーザは継続して通信できることを示す．

2. 正引き応答を利用したパケットフィルタリング

ユーザはインターネット上のホストと通信するとき，接続先のホストをホスト名で指定することが多い．そのため，ユーザの通信には，名前解決に関連した正引き要求と正引き応答が含まれる．力武らは，DNSとTCPコネクションに関する挙動の解析を行い，正引き応答の後に，正引き応答で得られたIPアドレスへの接続要求が続く傾向が強いことを確認している[10)]．一方，CodeRedv2など2001年から2003年に大規模な感染が確認されたワームは，接続先のホストをホスト名ではなく，IPアドレスで指定する．つまり，ワームの通信には正引き要求は含まれない．

しかし，正引き要求が含まれなかった通信がワームによる通信とは限らない．名前解決を行わない正当な通信には，DNSサーバやブロードキャストアドレスへの通信がある．このほかに，ユーザがIPアドレスを直接指定したときや，ホームページのリンクがIPアドレスで指定されているときなども名前解決は行われない．

名前解決を行わない通信におけるユーザとワームの違いは，一定時間内に接続するホスト数にあると考えられる．ウイルス・スロットルはこのホスト数に基づいてユーザとワームの通信を識別することから，これらの通信には，ウイルス・スロットルが有効であると考えられる．

そこで，本手法は，DNSサーバからの正引き応答を受信し，その正引き応答に含まれるIPアドレスへの送信を許可する．それ以外のIPアドレスへの送信は，ウイルス・スロットルにより識別する．ただし，本手法は，DNSサーバと通信できることを前提としているので，DNSサーバとの通信にはウイルス・スロットルを適用しない．

本手法は，ユーザが利用するコンピュータとDNSサーバの間に導入すればよい．最も単純な導入は，ユーザが利用するコンピュータへの導入である．一方，**図1**のように，複数のコンピュータとDNSサーバの間に，本手法を導入したパケットフィルタリングコンピュータを設置すれば，1台のコンピュータで複数のコンピュータに有効である．この構成では，パケットフィルタリングコンピュータがLAN内のコンピュータごと（送信元IPアドレスごと）にパケットの処理を行う．以下では，ユーザが利用するコンピュータに本手法を導入した場合について，詳細な処理の流れを述べる．

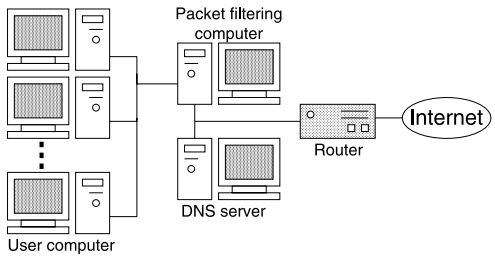

図 1 本手法の導入例
Fig. 1 Example of a packet filtering system.

2.1 受信パケットの処理

受信処理の対象は，事前に登録された DNS サーバ（ポート 53）からの正引き応答パケット（TCP と UDP の両方）とする．正引き応答を受信したら，正引き応答に含まれる IP アドレスをリスト（リゾルブセット）に登録する．正引き応答に複数の IP アドレスが含まれる場合，最初の 1 つだけを登録する．ただし，次の条件のいずれかを満たすとき，IP アドレスを登録しない．

- 登録された IP アドレスの総数が r 個以上のとき
- 正引き応答から得られた IP アドレスの個数が最近 1 秒間に w 個以上あったとき

本論文では，r をリゾルブセットの大きさ，w をリゾルブセットウインドウの大きさとよぶ．第 1 の条件は，メモリの消費と IP アドレスの検索時間の増大を防ぐためである．第 2 の条件は，ユーザとワームの名前解決を識別するためである．第 2 の条件が満たされた状態で，ウイルス・スロットルのキュー（次節参照）が一杯になれば，リゾルブセットを空にして，これ以降すべての IP アドレスを登録しない．つまり，この時点ですべての送信が拒否される．

リゾルブセットに登録された IP アドレスの有効期限は，正引き応答を受信してから 24 時間とする．24 時間を超えるとその IP アドレスはリゾルブセットから削除される．正引き応答で得られた IP アドレスがリゾルブセットに含まれる場合は，その IP アドレスの有効期限を 24 時間延長する．

2.2 送信パケットの処理

送信処理の対象は，SYN フラグが 1 の TCP パケット（3 way handshake の接続要求パケット）と，UDP パケットとする．フィルタリングルールの適用順序は次のとおり．

(1) 宛先 IP アドレスが事前に登録された DNS サーバの IP アドレスであり，宛先ポートが 53 であれば，送信を許可する．
(2) 宛先 IP アドレスが正引き応答から得られた IP アドレスであれば（リゾルブセットに含まれる IP アドレスであれば），送信を許可する．
(3) それ以外の IP アドレスであれば，ウイルス・スロットルにより処理する．

ウイルス・スロットル[7)]

ウイルス・スロットルでは，最近に通信した n 個の宛先 IP アドレスのリスト（ワーキングセット）を保持している．送信するパケットの宛先 IP アドレスがワーキングセットに含まれるなら，送信を許可する．また，ワーキングセットに含まれないが，ワーキングセットに空きがあれば，送信を許可し，その宛先をワーキングセットに追加する．それ以外の IP アドレスのとき，そのパケットをキューに追加する．キューに追加したとき，キューの長さが閾値 q を超えたら，キューからの送信を停止し，それをユーザに知らせる．これ以降，ウイルス・スロットルはすべての送信パケットを破棄する．しかし，本手法では，リゾルブセットに登録された IP アドレスには送信できるので，これ以降でも，すでに接続が確立したホストや，名前解決されたホストと通信できる．ただし，本手法では，キューが一杯の状態で，正引き応答から得られた IP アドレスの個数が 1 秒間に w 個以上あった場合には，DNS を利用するワーム（4 章参照）に感染していると判断し，すべての送信を停止する．

キューの先頭パケットはタイムアウト時間（d 秒）ごとに送信される．送信時に，ワーキングセット内で最も長い間通信していない IP アドレスを 1 つ削除し，送信したパケットの宛先 IP アドレスを追加する．新しく追加された IP アドレスと同じ宛先のパケットがキューに含まれるなら，それらをただちに送信する．キューからパケットを送信したとき，キューが空になり，ワーキングセットが一杯であれば，ワーキングセット内で最も長い間通信していない IP アドレスを 1 つ削除して，ワーキングセットに空きを 1 つ用意する．

3. シミュレーション実験

3.1 実験データ

本手法の有効性を評価するため，5 人のユーザと 4 種のワームからパケットを採取した．5 人のユーザは互いに異なるブロードバンド・ネットワークに属し，日常的にインターネットを利用している．ユーザ A と C のコンピュータは，プライベートアドレスが割り当てられ，NAPT を利用する．ユーザ B，D，E のコンピュータは，インターネットサービスプロバイダの DHCP サーバから 1 つのグローバルアドレスが割り当てられる．**表 1** に，ユーザが利用した OS の種類，

表 1 採取したパケットの属性
Table 1 Property of captured packets.

ユーザ	OS	日数	送信パケット数
A	Windows® XP	78	28,505
B	Windows® 98	40	81,982
C	Windows® XP	72	79,916
D	Windows® 2000	63	19,234
E	Windows® 2000	55	24,175

採取した日数とそのパケット数を示す．

実験に用いたワームは CodeRedv2，CodeRedII☆，Slammer，Blaster である．ワームのパケットは，VMware® Workstation 4 のゲスト OS（Windows® 2000 Server）から採取した．VMware® により仮想のネットワークを用意し，仮想のコンピュータ（Windows® 2000 Server）をワームに感染させる．仮想コンピュータの通信速度は，10 Mbps である．仮想のネットワークは現実のネットワークから切り放されているため，実験中にワームのパケットがインターネットに流出することはない．実験に用いたワームは増殖を始めると大量にパケットを送信し始めるので，増殖を開始してから約 1 時間だけ採取した．ただし，Slammer は，1 秒間に約 3,000 パケットもの莫大な量を送信するので，約 1 分間で採取を終了した．

ユーザとワームから採取したパケットは，次の 3 種類である．

- DNS サーバから受信した正引き応答パケット
  （`port 53 and dst host hostname`）
- 送信した TCP の接続要求パケット
  （`tcp[13]&18=2 and src host hostname`）
- 送信した UDP のパケット
  （`udp and src host hostname`）

これらのパケットは，WinDump 3.8α（パケットキャプチャドライバーは WinPcap 3.0 である）により採取した．カッコ内は WinDump のオプション（条件式）を示している．`hostname` は WinDump を実行したコンピュータのホスト名または IP アドレスである．

3.2 パケットの宛先 IP アドレスと正引き応答から得られた IP アドレスの関係

送信パケットの宛先 IP アドレスと正引き応答から得られた IP アドレスの関係を調べた．**表 2** に，送信されたパケットの宛先 IP アドレスが正引き応答に含まれた割合を示す．表 2 から，多くの宛先 IP アドレスは正引き応答に含まれることが確認された．つまり，宛先 IP アドレスと正引き応答に含まれる IP アドレスには強い相関がある．

☆ CodeRedII は，CodeRedv2 と同じセキュリティホールを不正に利用するが，プログラムコードは異なる．

表 2 宛先 IP アドレスが正引き応答に含まれた割合
Table 2 Rate of destination IP addresses included in DNS responses among all destination IP addresses.

User A	User B	User C	User D	User E
89.66%	98.79%	96.74%	97.25%	92.65%

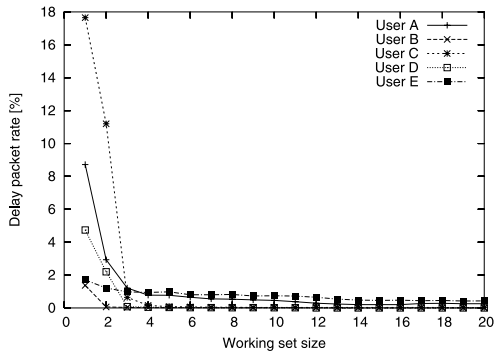

図 2 ワーキングセットの大きさ n と遅延パケットの割合
Fig. 2 Working set size vs. delay packet rate.

正引き応答に含まれなかった IP アドレスは，主に，DNS サーバやネットワークプリンタなど LAN 内の IP アドレスであった．LAN 外の IP アドレスには，少数ではあるが，インスタント・メッセンジャやオンラインゲームなどアプリケーション固有の IP アドレスや，接続先のサーバが指定した IP アドレスがあった．本手法では，これらの IP アドレスを宛先とするパケットは，ウイルス・スロットルにより処理される．

3.3 パラメータの設定

3.3.1 ウイルス・スロットルのパラメータ

ウイルス・スロットルの性能を決めるパラメータには，ワーキングセットの大きさ n と，タイムアウト時間 d，キューの閾値 q がある．ここから，実験データを用いて，本手法の一部として機能するウイルス・スロットルのパラメータを求める．

ワーキングセットの設定

ワーキングセットは，遅延なくパケットを送信できる宛先 IP アドレスのリストである．ワーキングセットが小さければ，宛先の IP アドレスがワーキングセットに含まれる可能性が低くなるため，パケットの送信は遅延しやすくなる．つまり，ワーキングセットの大きさ n によって，遅延するパケット数は変化する．

そこで，適切なワーキングセットの大きさ n を求めるために，ワーキングセットの大きさ n と遅延パケット数の関係を調べた．**図 2** に，ユーザごとのワーキングセットの大きさ n と，全パケットに占める遅延パケットの割合を示す．ここで設定したタイムアウト

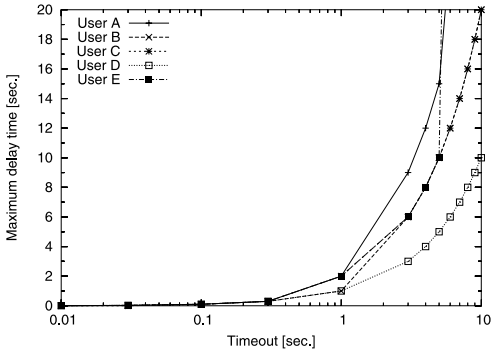

図 3 タイムアウト時間 d と最大待ち時間
Fig. 3 Timeout vs. maximum delay time.

時間 d は 1 秒間である．遅延パケットの割合は，ワーキングセットの大きさ n が 3 以上になるとほとんど変化しない．これは，最近に通信した IP アドレスと相関がある IP アドレスの個数は 3 個程度であることを意味する．一方，ウイルス・スロットル単体では，その値はおおよそ 6 であった．したがって，本手法では，ワーキングセットの大きさ n を 3 とし，ウイルス・スロットル単体では 6 とする．

タイムアウト時間の設定

タイムアウト時間 d は，キューの先頭パケットが送信される時間間隔であるとともに，ワーキングセットの IP アドレスが新しい IP アドレスに置き換えられる時間間隔である．つまり，タイムアウト時間 d は単位時間あたりに送信可能な新しい IP アドレスの個数 $(1/d)$ を与える．タイムアウト時間 d が短くなるにつれて，遅延パケットは減少するが，ワームの増殖を止めるまでにかかる時間が長くなる．特に，タイムアウト時間 d が，ワームの増殖間隔（ワームが新しいホストへ接続しようとする時間間隔）より短くなると，ワームの増殖を止められない．たとえば，Blaster は，約 0.04 秒間隔（表 4 参照）で増殖するので，Blaster の増殖を阻止するには，タイムアウト時間 d を 0.04 秒より大きくしなければならない．今後のワームにも対応するために，タイムアウト時間 d を 0.04 秒より，できるだけ長くする必要がある．しかし，タイムアウト時間 d が長くなるにつれて，遅延パケットが増加するとともに，送信待ちの時間が著しく長くなる．

そこで，0.01 秒から 10.0 秒までのタイムアウト時間 d について，ユーザが 1 つの接続に待たされる最大待ち時間を調べた．その結果を**図 3** に示す（グラフの横軸は，対数目盛りである）．最大待ち時間は，タイムアウト時間 d が 0.1 秒付近から緩やかに増加し，1.0 秒付近から急激に増加している．本手法では，最大待ち時間が急激に増加する直前のタイムアウト時間（1.0 秒）をタイムアウト時間 d として設定する．ウイルス・スロットル単体でも，図 3 と同様の結果が得られたので，ウイルス・スロットル単体のタイムアウト時間 d も 1.0 秒とする．この値は，文献 8) の実装実験で設定された値と同じである．

キューの設定

ウイルス・スロットルでは，ワーキングセットに含まれない IP アドレスを宛先とするパケットは，キューに追加される．キューの長さが閾値 q を超えたとき，キューのパケットは破棄され，これ以降，パケットはキューへ追加されない．したがって，その閾値 q はユーザの通常の操作で発生したキューの最大サイズより大きくしなければならない．

そこで，ワーキングセットの大きさ n とタイムアウト時間 d を上述の値に設定して，キューの最大サイズを調べた．各ユーザのキューの最大サイズは，本手法では 2 から 25（平均値は 12.2），ウイルス・スロットル単体では，28 から 153（平均値は 69.0）であった．様々なユーザの操作やアプリケーションの影響を避けるため，本手法では，実験で得られた最大値 25 より十分に大きな値として閾値 q を 100 に設定する．ウイルス・スロットル単体では，最大値が 153 であったので，200 に設定する．

3.3.2 リゾルブセットの設定

リゾルブセットは正引き応答から得られた IP アドレスのリストである．リゾルブセットには，IP アドレスを 1 秒間に最大 w 個（リゾルブセットウインドウの大きさ）まで登録できる．リゾルブセットウインドウに入りきらなかった IP アドレスを宛先とするパケットは，名前解決されていない IP アドレスとして判断され，ウイルス・スロットルで処理される．つまり，リゾルブセットウインドウの大きさ w によって，ウイルス・スロットルで処理されるパケット数は変化する．前節で述べたように，ウイルス・スロットルは送信を遅延させることがあるので，リゾルブセットウインドウの大きさ w は，遅延パケット数に影響を与える．

そこで，遅延パケットの個数を最小化するリゾルブセットウインドウの大きさ w を求めるために，リゾルブセットウインドウの大きさ w と遅延パケットの割合を調べた．その結果を**図 4** に示す．遅延パケットの割合は，リゾルブセットウインドウの大きさ w が 12 以上になると変化しなかった．したがって，本実験では，リゾルブセットウインドウの大きさ w を 12 に設定する．

リゾルブセットに登録できる IP アドレスの個数は，

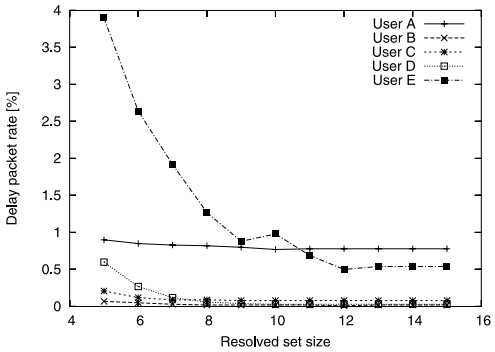

図 4 リゾルブセットウインドウの大きさ w と遅延パケットの割合
Fig. 4 Resolved set size vs. delay packet rate.

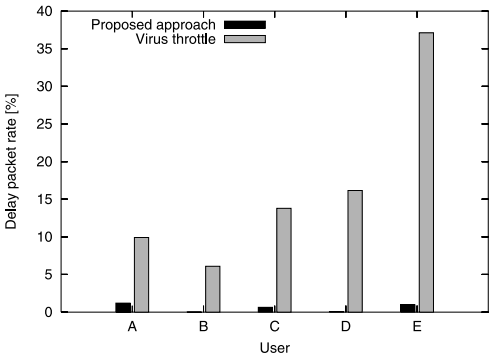

図 5 各ユーザの遅延パケットの割合
Fig. 5 Delay packet rate.

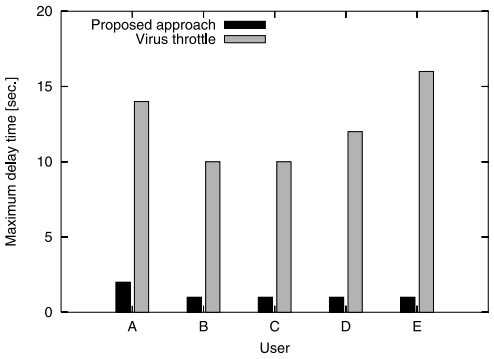

図 6 各ユーザの最大待ち時間
Fig. 6 Maximum delay time.

表 3 実験パラメータの一覧
Table 3 Experimental parameters.

Approach	n	d	q	r	w
Proposed approach	3	1	100	300	12
Virus throttle	6	1	200	–	–

リゾルブセットウインドウの大きさ w だけでなく，リゾルブセットの大きさ r によって制限される．そのため，リゾルブセットの大きさ r は，少なくとも，ユーザの通常の操作で登録された IP アドレスの個数より大きくしなければならない．5 人のユーザについて，リゾルブセットへ登録された IP アドレスの最大個数を調べると，その個数は 94 から 247（平均値は，163.2）であった．ユーザの操作やアプリケーションの影響を避けるため，本手法では，リゾルブセットの大きさ r を 300 に設定する．

3.4 シミュレーションによる性能評価

前節で求めたパラメータ（**表 3**）を設定して，シミュレーションにより，ユーザとワームに対する本手法の性能を評価する．

3.4.1 ユーザに対する性能評価

ユーザ 5 人の実験データに対して，本手法とウイルス・スロットル単体によって発生する通信の遅延を比較した．シミュレーション評価では，実験データについて，連続する接続要求間の因果関係☆を調べることは困難なので，ユーザの接続要求はすべて独立していると仮定した．**図 5** に各ユーザの通信で発生した遅延パケットの割合を示す．ユーザごとに，左側の棒グラフが本手法，右側の棒グラフがウイルス・スロットル単体の結果を示している．正引き応答を利用することによって，パケットはほとんど遅延しないようになったことが分かる．

通信の遅延に関する評価は，遅延パケットの割合だけでなく，ユーザが 1 つの接続に待たされる最大待ち時間も重要な評価尺度である．**図 6** に各ユーザの最大待ち時間を示す．本手法では，最大待ち時間が 1 秒から 2 秒（平均値は，1.2 秒）であり，その待ち時間が発生する頻度も非常に少ないことから，この最大待ち時間は，実用に耐えられると考えられる．一方，ウイルス・スロットル単体では，最大待ち時間は 5 秒から 16 秒（平均値は，10.2 秒）であった．ただし，この最大待ち時間は，過大評価されていることに注意されたい．シミュレーション評価では，すべての接続要求は他の接続要求と独立していると仮定したためである．たとえば，ワーキングセットが一杯の状態で，1 秒間に異なる 5 個の IP アドレスに接続要求があったとする．シミュレーションでは，接続要求がすべて独立していると仮定したので，最大待ち時間は 5 秒になるが，仮に，1 つ前の IP アドレスから次の IP アドレスを取得して接続する要求が 5 つ続いた場合は，1 つ前の IP アドレスが許可されるまで次の接続は要求されないので，最大待ち時間は 1 秒になる（合計待ち時

☆ 連続する 2 つの接続要求について，1 つ目の IP アドレスと接続したことが原因で，2 つ目の IP アドレスと接続する関係を意味する．

間は 5 秒である）．このように，本実験の最大待ち時間は，過大評価されているため，最悪の場合における最大待ち時間として考える必要がある．正確な最大待ち時間は，本手法を実装して実際に遅延させた状態で評価する必要がある．

本手法が最大待ち時間を短縮できた要因は，次のとおりである．送信の遅延は，ウイルス・スロットルで発生するため，できるだけウイルス・スロットルで処理するパケット数を減らさなければならない．本手法は，パケットの宛先が名前解決されたホストであれば，そのパケットを遅延なく送信し，名前解決されていなければ，そのパケットをウイルス・スロットルで処理する．実験データは，約 90%以上のホストが名前解決され，約 10%以下のホストが名前解決されていない（表 2 参照）．つまり，本手法では，約 90%のホストへのパケットを遅延なく送信し，残りのホストへのパケットだけをウイルス・スロットルで処理する．このように，ウイルス・スロットルで処理するパケット数を減らしたため，本手法は，最大待ち時間を短縮できた．

図 5 と図 6 に関して，ウイルス・スロットル単体の結果は，文献 8) の結果よりも悪い．この原因の 1 つは，実験に用いたパケットのプロトコルやポートの違いがあげられる．本研究では，TCP と UDP の全ポートのパケットを評価したが，文献 8)，9) では，TCP の全ポートか TCP のポート 80 のパケットだけを評価していた．もう 1 つは，実験対象のネットワーク環境やアプリケーションなど利用環境の違いがあげられる．

次に，本手法を導入したとき，パケットの送信が遅延しやすくなるアプリケーションについて考察する．本手法は，正引き応答から得られなかった IP アドレスを宛先とするパケットに対してウイルス・スロットルを用いるため，名前解決を必要としないアプリケーションの通信は遅延しやすい．たとえば，Winny や WinMX^{TM} など P2P ネットワークを利用するアプリケーションがその 1 つである．P2P アプリケーションは，接続先のホストを主に IP アドレスで指定し，さらに，単位時間あたりに通信するホスト数は従来のアプリケーションよりも多い．ウェブページの閲覧など日常の作業中に，P2P アプリケーションを利用した場合は，それを利用しない場合よりも遅延が発生しやすくなるため，待ち時間が長くなると予想される．そこで，約 1 週間，P2P アプリケーションを起動した状態で日常作業を行い，採取したパケットを調べた．その結果，遅延パケットの割合は，P2P アプリケーションを利用しないときと比べて約 16 倍の値（20.13%）に増加した．しかし，本手法の総待ち時間は，ウイルス・スロットル単体の 61.5%であり，本手法の最大待ち時間は 2 秒間であった．したがって，P2P アプリケーションを利用しても，本手法は実用に耐えられると考えられる．本手法の最大待ち時間が長くならなかった原因は，実験に用いた P2P アプリケーションが，異なるホストと通信を頻繁に行わなかったためである．

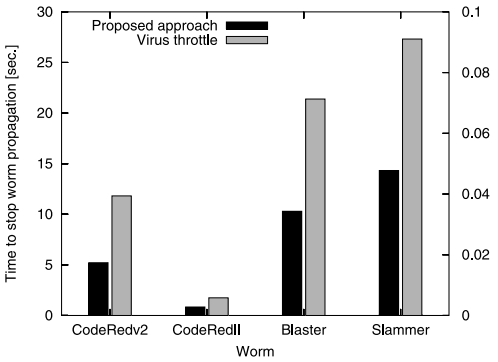

図 7 ワームの増殖停止に要した時間
Fig. 7 Time to stop worms.

3.4.2 既知のワームに対する性能評価

実在する 4 種のワームに対して，増殖を阻止するまでにかかる時間を評価した．その結果を**図 7** に示す．ワームごとに，左側の棒グラフは本手法，右側の棒グラフはウイルス・スロットル単体の結果を示している．ただし，Slammer は，他の 3 種のワームと比べて，非常に早く阻止できたため，Slammer に関する時間のスケールは右側の縦軸に従う．本実験で用いたワームは，DNS を利用しないので，増殖を阻止するまでにかかる時間は，主に閾値 q の大きさで決まる．本手法の閾値 q は，ウイルス・スロットル単体の 2 分の 1 であるため，本手法はウイルス・スロットル単体よりもワームの増殖を約 2 倍早く阻止した．

次に，ユーザとワームの通信が混在している場合，すなわち，ユーザがワームに感染したコンピュータを使用した場合について考える．本実験では，ユーザ A が Blaster に感染したことを想定して，ユーザ A が 1 時間に送信したパケット列の中に，Blaster が 1 時間に送信したパケット列を挿入した．そのデータを本手法により評価した結果，9.07 秒でワームの増殖は停止し，ユーザは名前解決が行われたホストのみ継続して通信できることを確認した．

4. 今後のワームに関する考察

近い将来に作成される可能性があるワームについて考察する．まず，本研究で考察すべきワームは，DNS

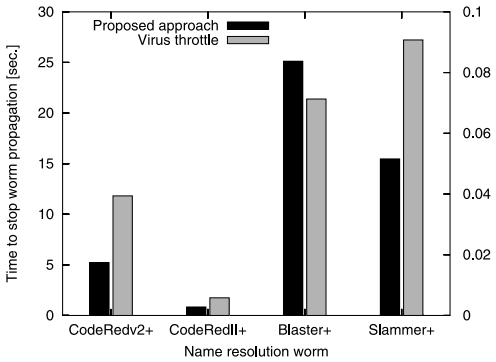

図 8 DNS を利用するワームの増殖停止に要した時間
Fig. 8 Time to stop name resolution worms.

を利用するワーム（ホスト名で増殖先のホストを指定するワーム）である．本手法は正引き応答から得られた IP アドレスに基づいて，ユーザとワームの通信を識別しているため，DNS を利用するワームに対して性能を評価することは重要である．DNS を利用するワームとして，次のようなワームが考えられる．

- ランダムに増殖先のホスト名を生成するワーム
- 感染したホスト内のデータから増殖先のホスト名を探し出すワーム[1)]
- Google^{TM} など外部のサーバから増殖先のホスト名を探し出すワーム[1)]

しかしながら，このようなワームは，2004 年 3 月時点で，Windows® 上で動作するものは確認されていない☆．

そこで，前節で評価した 4 種のワームについて，これらのワームが DNS を利用したと仮定して，ワームから送信されたパケット列に対して，TCP の接続要求ごとに正引き要求と正引き応答のパケットを追加した．そのデータを評価した結果を**図 8** に示す．前節で示したワームと区別するために，DNS を利用するワームはワーム名の後ろに「+」を付け加えた．

Blaster+以外のワームに対して，本手法は，DNS を利用しないワームと同等の性能が得られた．Blaster+に対して，本手法の性能がウイルス・スロットルより低い原因は，Blaster の増殖速度が関係している．**表 4** に，オリジナルのワームの最大増殖速度を示す．単位は 1 秒間あたりの接続要求パケット数 pps（packets per second）である．本実験では，リゾルブセットウインドウの大きさ w を 12 に設定しているため，毎秒 13 個以上の IP アドレスへの接続がなければ，ウイルス・スロットルで処理されない．Blaster+の増殖速度は最大でも 25 pps であることから，Blaster+のパケットはウイルス・スロットルで処理されずに送信されることが多い．そのため，本手法は，ウイルス・スロットル単体と比べて，Blaster+を停止するまでに多くの時間を要した．

☆ 感染したホストから増殖先のホスト名を探し出すワームは，1988 年に作成されているが，このワームは Sun Microsystems の Sun3 システムと 4BSD 系の UNIX® が動作している VAX^{TM} コンピュータにしか感染できない[11)]．

表 4 ワームの増殖速度（単位は pps）
Table 4 Speed of worm propagation.

CodeRedv2	CodeRedII	Blaster	Slammer
102	292	25	3,151

ここで，ワームの増殖速度が，リゾルブセットウインドウの大きさ w より小さい場合について考える．本手法は，リゾルブセットに登録できる IP アドレスの個数を r 個に制限している．そのため，ワームの増殖速度が遅くても，リゾルブセットに登録された IP アドレスの個数が r に達したら，それ以降の新しいホストへの接続要求はウイルス・スロットルにより処理される．たとえば，本実験では，リゾルブセットの大きさ r を 300 に設定しているので，ユーザやワームが 24 時間以内に 300 台のホスト名を名前解決すれば，それ以降の新しいホストへの接続要求は，たとえ名前解決されていようとも，名前解決されていないホストとして解釈され，その接続要求はウイルス・スロットルにより処理される．このリゾルブセットの大きさ r は，IP アドレスの有効期限（24 時間）を短くすれば小さくできる．実験データについて，性能を低下させない範囲で調整した結果，有効期限を 30 分間にすると，リゾルブセットの大きさ r を 200 まで減らせた．

次に，DNS を利用しない従来のワームについて考察する．本手法では，DNS を利用しないワームの通信は，ウイルス・スロットルによって処理される．ウイルス・スロットルは，「急速に広がるワーム」の増殖を抑える手法であるため，増殖速度が遅いワームに対して，期待した性能が得られないことがある．たとえば，コンピュータが別の新しいホストと接続したとき，そのホストへ増殖を試みるワームである[1),12)]．このようなワームは，ユーザの操作と連動しているため，ユーザとワームの識別が難しい．しかし，文献 9) によると，そのようなワームは，増殖速度が遅いため，従来の「侵入阻止」の手法，すなわち，アンチウイルスソフトなどのセキュリティ対策ソフトでも効果が得られるとしている．

最後に，本手法の機能を無効にするワームについて考察する．このワームは，図 1 のように，パケット

フィルタリングを行うコンピュータと，ユーザが利用するコンピュータを分離することによって，本手法の機能をワームから保護できると考えられる．図 1 の構成であれば，1 台のコンピュータにより複数のコンピュータへ本手法を適用できる．

5. おわりに

本論文では，正引き応答を利用したパケットフィルタリングを提案した．シミュレーション実験では，ユーザ 5 人と，実在する 4 種のワームから採取したパケットを用いて，ワームに対する有効性を評価した．その結果，提案手法は，ウイルス・スロットルによる送信の遅延時間を飛躍的に短縮し，ワームの増殖を速やかに阻止した．さらに，DNS を利用しないワームであれば，たとえ感染しても，提案手法は，ワームの増殖だけを阻止し，ユーザは名前解決されたホストであれば継続して通信できることを確認した．また，近い将来に作成される可能性のあるワームをいくつか取り上げ，それらに対する本手法の有効性について考察した．今後は，プロトタイプの実装およびその性能評価を行い，提案手法の実用化を目指したい．

謝辞 本研究は，文部科学省科学研究費補助金若手研究 B（課題番号：15700075）の助成による．

参考文献

1) Weaver, N., Paxson, V., Staniford, S. and Cunningham, R.: A Taxonomy of Computer Worms, *Proc. ACM Workshop on Rapid Malcode* (*WORM*) *2003* (2003).
2) Moore, D. and Shannon, C.: The Spread of the Code-Red Worm (CRv2), *The Cooperative Association for Internet Data Analysis* (*CAIDA*) (2001).
3) Moore, D., Paxson, V., Shannon, C., Staniford, S. and Weaver, N.: The Spread of the Sapphire/Slammer Worm, *The Cooperative Association for Internet Data Analysis* (*CAIDA*) (2003).
4) Symantec Corp.: Understanding Heuristics: Symantec's Bloodhound Technology, Symantec White Paper Series, Vol.XXXIV (1997).
5) Ferguson, P. and Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, RFC 2827 (2000).
6) Liston, T.: Welcome to My Tarpit: The Tactical and Strategic Use of LaBrea (2001).
7) Williamson, M.M.: Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code, *Proc. ACSAC Security Conference*, pp.61–68 (2002).
8) Twycross, J. and Williamson, M.M.: Implementing and Testing a Virus Throttle, *Proc. 12th USENIX Security Symposium*, pp.285–294 (2003).
9) Williamson, M.M., Twycross, J., Griffin, J. and Norman, A.: Virus Throttling, Virus Bulletin, Vol.3, pp.8–11 (2003).
10) 力武健次，野川裕記，管谷史昭，中尾康二，下條真司：DNS と TCP コネクションに関する挙動の解析，*CSS2003*, pp.521–526 (2003).
11) Spafford, E.H.: The Internet Worm: Crisis and Aftermath, *Comm. ACM*, Vol.32, No.6, pp.678–687 (1989).
12) Staniford, S., Paxson, V. and Weaver, N.: How to Own the Internet in Your Spare Time, *Proc. 11th USENIX Security Symposium*, pp.149–167 (2002).

(平成 16 年 3 月 26 日受付)
(平成 16 年 9 月 3 日採録)

推薦文

「ワームは DNS をひかない」という仮定のもとでワーム増殖を担うパケットをフィルタリングする手法を提案している．そのフィルタリングポリシーには十分な新規性があり，かつ，実験により裏付けしているため有効性も高い．そのワーム拡散抑止効果は健全なネットワーク社会の実現に寄与するところが大変大きいと考えられるので，ここに研究会推薦論文として推薦したい． （CSEC 研究会前主査 岡本 栄司）

岡本 剛（正会員）

昭和 48 年生．平成 14 年豊橋技術科学大学大学院工学研究科博士後期課程修了，平成 14 年より神奈川工科大学情報学部情報ネットワーク工学科助手．コンピュータウイルスの伝播防止等コンピュータセキュリティの研究に従事．工学（博士）．平成 12 年電気通信普及財団テレコムシステム技術学生賞受賞．電子情報通信学会会員．