情報処理学会論文誌 Vol. 50 No. 6 1610–1621 (June 2009)

ダミーアドレスからのコネクショントレースバックによるワーム早期抑制手法

稲場太郎[†1] 川口信隆[†1] 岡田謙一[†2]

近年，ネットワークワームによる被害が多数報告されてきた．なかでも主流はスキャニングワームであり，この種のワームへの対策については各所で議論されている．ところが，それらの対策では検知や抑制が困難な内部アドレスリストを用いたワームも登場してきている．本論文では，この内部アドレスリストを用いたワームに対する検知・抑制手法を提案する．具体的には，各ホストにダミーアドレスを持たせ，そこへのコネクションをワームと判断することによってワームの検知を行う．また，そのコネクション元のホストだけでなく，そこから接続履歴を遡ることで次々に感染疑惑のあるホストを発見し，感染疑惑ホストをネットワークから切断することによってワームの根絶を目指す．我々が行ったシミュレーション評価の結果，比較的遅いワームを感染台数 1%以下，切断台数 5%以下の段階で根絶でき，本提案手法の有用性が示された．

Worm Containment Using Traceback from Dummy Addresses

Taro Inaba,[†1] Nobutaka Kawaguchi[†1] and Kenichi Okada[†2]

Most of existing network worms have used address scanning to find vulnerable hosts. Recently, however, worms with more effective propagation strategies have emerged. Among the worms, we focus on the worms that exploit address lists obtained from infected hosts to find other vulnerable hosts effectively. In this paper, we propose a method that detects and contains such worms that try to infect all hosts in an enterprise network. In our method, a detection system inserts some dummy addresses into the address lists of hosts in the network. Then, the system detects the existence of worms when a host tries to open a connection to a dummy address, and then traces back the connection logs to find potentially infected hosts and removes them from the network. Computer simulation results showed our method detected and contained worms with less than 1% infected hosts and less than 5% removed hosts.

1. はじめに

近年，CodeRed など，様々なワームによる被害が報告されてきた[1]．これまでの主流はスキャニングワームであり，これは無作為に感染ターゲットとなるアドレスを作成し，感染コネクションを張るものであった．この種のワームに対する抑制手法は，各所で議論されてきた[2),3)]．これらの手法は主にネットワークの挙動を観察し，通常状態と異なる場合にワームと判断するものであった．しかし，近年ではスキャン以外の感染手法をとるネットワークワームが登場しており，検知，抑制が困難となっている．

その中でも我々は感染ホストの内部にあるアドレス情報を用いて感染活動を行うワームに注目した[4]．内部アドレス情報を用いた感染活動の場合，スキャンと異なり確実に存在するアドレスに対してコネクションを張ることができるため，感染の広がりが速いにもかかわらず，感染コネクションは通常コネクションに紛れ込んでしまう．したがって，ネットワークの異常を発見してこのワームの存在を検知することは困難である．

そこで本論文では，ダミーアドレスという通常では使用されないアドレスを用い，そこからのコネクショントレースバックによってワームを抑制する手法を提案する．本手法ではエンタープライズネットワークを対象とし，全ネットワークを監視する Address Monitoring Server（AMS）によってワームを抑制する．AMS は各ホストのアドレスリストにダミーアドレスを持たせ，このアドレスへの接続が行われることでワームの存在を検知する．これは，通常の挙動では各ホストはダミーアドレスにはコネクションを張ることはないという考えに基づくものである．そしてこのアドレスにコネクションを張ったホストはワームに感染していると判断し，このホストをネットワークから切断する．また，AMS は全ホスト間の接続履歴を保持しており，これを利用して接続をトレースバックし，感染の疑いがあるホストを次々にネットワークから切断する．この作業を繰り返し，最終的にはアクティブな感染ホストを根絶させることを目指す．

以降の本論文の構成は以下のとおりである．まず，2 章において提案の背景に触れる．3 章でダミーアドレスからのコネクショントレースバックによるワームの早期抑制手法を提案

†1 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
†2 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

し，4 章で提案手法のシミュレーション評価について述べる．そして 5 章は本論文のまとめとする．

2. 背　　景

2.1 スキャニングワーム

近年登場した，CodeRed などのワームはアドレススキャンにより感染活動を行ってきた[1]．アドレススキャンはターゲットとなるアドレスをランダムに探索するものであり，実際には存在しないアドレスに対するコネクションも多く行うこととなる．したがって，ワームの感染速度に対してネットワークのトラフィック量が非常に多くなり，この特徴を利用してワーム検知，抑制が可能となる．

Williamson は，この種のワームを抑制するウイルススロットルという手法を提案している[2]．この手法は，接続頻度がある値を超えるトラフィックに対して遅延を発生させ，ワームの感染活動を遅らせようというものである．通常の通信においてこの速度を超えてしまった場合も遅延が生じるが，通常の通信があまりに頻繁に行われることはほとんどないので，遅延の影響は最小限に抑えられる．それに対し，ワームのトラフィックの場合はつねに高速で感染活動を行っているので遅延の影響が甚大となり，感染を遅らせることができる．

Schechter らは，スキャニングワームの早期検知抑制手法を提案している[3]．この手法では，スキャンパケットのほとんどが初めての接続先に対するものであることを利用し，ワームを検知する．また，ワーム検知システムが監視した結果の仮説検定と信頼性ベースの速度規制によってワームを抑制する．これにより，非常に低い誤検知率でワームの早期抑制が実現できる．

2.2 内部アドレスリストを利用したワーム

スキャニングワームは，ランダムに作成したアドレス宛てに感染活動を行おうとする．このため，実在しないアドレス宛てへの送信やコネクション確立要求が出されるため，非常に非効率的である．また，先述のような抑制手法も多数議論されている．このような現状に対し，より効率的な感染活動を行うネットワークワームが登場してきた[4),5]．その 1 つが内部アドレスリストを利用したネットワークワームである．

この種のワームは，感染したホスト内部のアドレスリスト（E メールのアドレス帳，ARP キャッシュなど）を用いてターゲットの発見を行う．したがって確実に存在するアドレスに対して感染コネクションを張ることができ，効率的であると同時に低い接続頻度でも感染活動を行うことが可能となる．通常のコネクションに紛れる形で感染活動が行われると，先述の抑制手法はまったく効果がなくなる．そこで，これらのワームを抑制する手法の確立が必要である．

内部アドレスリストを用いて感染活動を行うワームの一種に，E メールワームがある．Huang らは，ユーザの E メールソフトのアドレス帳にそれぞれ 1 つダミーアドレスを加えることによって，E メールワームを捕え，感染を抑える手法を提案している[6]．

この手法では，まずサーバ側が未使用のダミーアドレスを用意し，各ユーザのアドレス帳にこれを登録させる．ダミーアドレスにメールが送信された場合，そのユーザとシグニチャがブラックリストに加えられ，このリストによって E メールワームを抑制する．

この手法は，E メールワームに特化しており，他の内部アドレスリストを用いたワームについては議論されていない．また，コネクションごとに攻撃コードのシグニチャを変更して感染活動を行うワーム（ポリモフィックワーム）の場合，シグニチャベースの抑制手法は効果を発揮しない．したがって，より効率的な抑制手法が必要となる．

2.3 ボット攻撃

内部アドレスリストを利用したワームと同様に，ネットワークの監視のみでは存在検知が難しい攻撃として，近年ボットが猛威を振るっている．ボットは感染させたホストマシンを外部から操作することで攻撃活動を行うものであり，その活動はマシンのユーザやネットワーク管理者からは見えにくいため，検知や駆除が困難となっている．

このボットに対し，サイバークリーンセンターではおとりマシンであるハニーポットを用いることで感染マシンを検知，ボットの駆除を行う活動を展開している[7]．この活動では，ハニーポットに対して感染活動を行ったマシンはボットに感染していると判断し，協力 ISP を通じてその感染ホストを特定する．そしてそのホストに対して注意喚起メールを送信し，対策を促すことを行う．

この活動は「おとり」を用いるという点で本提案手法と類似している．しかしながら，本提案手法と異なり，インターネット上に広がるボットを対象としているため，感染ホストの特定には ISP の協力が必要となる．また，トレースバックを行うことによる感染疑惑ホストの発見も難しい．本提案手法ではエンタープライズネットワーク内での感染活動を対象としているため，対象ネットワークのログを集めることができ，即座に感染ホストを特定するとともにトレースバックを行うことが可能となっている．

3. ダミーアドレスからのコネクショントレースバックによるワーム早期抑制手法

3.1 検知対象ワーム

本提案手法での検知・抑制の対象となるのは，エンタープライズネットワーク内で感染活動を行い，内部アドレスリストからランダムに感染対象ターゲットとなるホストを選択して感染コネクションを張るワームである．ここでいう内部アドレスリストとは，E メールソフトのアドレス帳，インスタントメッセンジャの登録アドレス，ARP キャッシュ，他のホストへの接続履歴など，ホストマシンの内部に保存されているアドレスリストのことを指す．また，本手法では，シグニチャベースの抑制手法[6] を回避できるポリモフィックワームも検知が可能となる．

3.2 ワームの抑制手法

本提案手法の概略を**図 1** に示す．

本手法では，各ホストのアドレスリスト中に通常は存在しない「ダミーアドレス」を挿入し，そこへ張られたコネクションからトレースバックを行うことでワームの抑制を行う．まず，各ホストのユーザがアドレスリストにダミーアドレスをいくつか追加する．このダミーアドレスに対して接続がなされると Address Monitoring Server（AMS）と呼ばれるサーバが検知し，ワームがネットワーク中に存在すると判断する．その後，接続元のホストを感染ホストと見なし，ネットワークから切断し，以後の感染の広がりを抑える．これは，ダミーアドレスに対してコネクションを張るのはワームに感染しているホストのみであるという考えに基づいたものである．

また，ワームは絶えず増殖を繰り返すものであるため，ここで切断されたホストがワームの起源である確率は低い．したがって，このホストだけを切断したのではワームは根絶されず，他の感染ホストも発見する必要がある．そこで AMS は各ホスト間の接続履歴情報を利用する．起源以外の感染ホストは必ず事前に他の感染ホストからコネクションを受けていると考えられるので，ダミーアドレスにコネクションを張ったホストからトレースバックすることによって感染の疑いが高いホストを発見することができる．ここで抽出した感染疑惑ホストもネットワークから切断し，結果的に 1 度のダミーアドレスに対するコネクションから複数のホストが切断されることになる．AMS はダミーアドレスにコネクションが張られるたびにこの作業を繰り返し，ワームの根絶を目指す．

図 2 にはコネクショントレースバックの概略を示す．

この図では，ホスト H が時刻 t_3 においてダミーアドレスにコネクションを張っている．AMS はまずこのホストを感染ホストと見なし，ネットワークから切断する（以降，ホスト

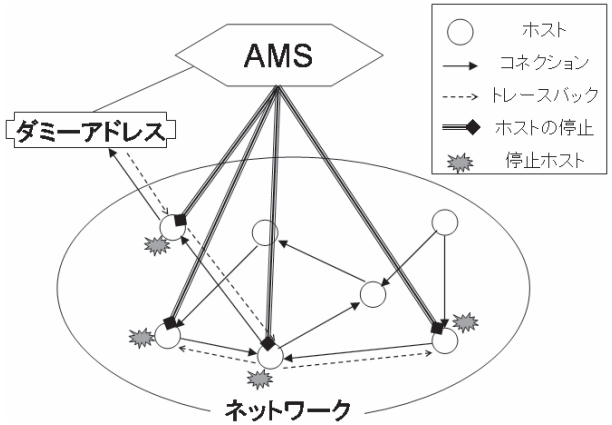

図 1　提案手法概略
Fig. 1　Outline of proposal.

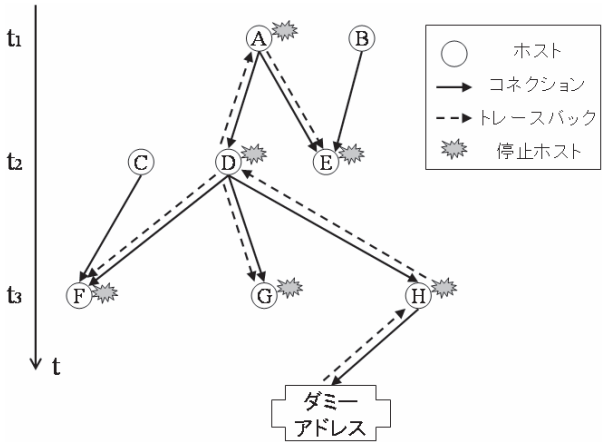

図 2　コネクショントレースバックの概略
Fig. 2　Outline of connection traceback.

をネットワークから切断することをホストの停止と呼ぶ）．続けて，このホストから接続履歴を遡ることによって感染疑惑のあるホストを次々に発見し，停止させていく．図 2 では，ホスト D，F，G，A，E が遡られ，停止させられている．この作業はダミーアドレスに対してコネクションが張られるたびに行われ，すべての感染ホストが停止するまで続けられる．

3.2.1　AMS

本提案手法において AMS は以下の役割と機能を持っている．

- エンタープライズネットワークの全ホストを監視し，各ホストのアドレスリストにダミーアドレスを挿入させる．
- 全ホスト間の接続履歴を保持している．
- 任意のホストをネットワークから切断する．

ダミーアドレスの挿入の手順としては，まず AMS にはダミーアドレスとして複数の IP アドレスを保持させる．続いて，それらのアドレスを各ホストがインストールという形で挿入できるようなプログラムを用意し，外部メディアに保存する．各ホストのユーザはその外部メディアを用いてプログラムを実行することでダミーアドレス挿入を実現する．各ホストがダミーアドレスを挿入しているかどうかは AMS に通知され，挿入していないホストに対しては AMS が挿入を促す．

具体的な追加先としては，たとえば IPmessenger[8] があげられる．このアプリケーションは LAN 内でのインスタントメッセージサービスを可能にするもので，企業内での従業員同士のコミュニケーションに広く利用されている．このアプリケーションを起動すると，その LAN 内にあるホストのうち IPmessenger にログインしているホストのホスト名と IP アドレスがリスト状に表示される．ワームは，このリストを活用してターゲットとなるホストのアドレスを取得することが可能となるため，このリストはダミーアドレス挿入の対象となる．ここに AMS が保持している複数のダミー IP アドレスを追加することでダミーアドレスの挿入を実現する．

また，ホスト名と IP アドレスの対応が記述され，アドレス解決に用いられる hosts ファイル（WindowsXP では C:\WINDOWS\system32\drivers\etc にある）もダミーアドレスの追加先の 1 つとして考えられる．この hosts ファイルには LAN 内のサーバなどの IP アドレスが記載されることがあるため，このファイルを利用することでワームは次なるターゲットを発見することができる．そこで，このファイルにダミーとなる IP アドレスを記載しておき，それを適当なホスト名（たとえば mailserver など）と関連付けておくことで，hosts ファイルを利用するワームを捕えるためのダミーアドレス挿入が実現できる．

接続履歴の収集は，中継スイッチやルータにおけるポートミラーリングによって実現する．具体的な手法としては，Cranor らの提案している Gigascope などがあげられる[9]．この際に収集する情報は通信プロトコル，ポート番号，コネクション時刻，ソースホストのアドレス，ディスティネーションホストのアドレスであり，これらの情報が AMS に蓄えられる．本研究ではエンタープライズネットワークを対象としているため，これらの情報収集に際するプライバシの問題は大きくないと考える．

ホストをネットワークから切断する際には，スイッチにおけるフィルタリング機能を利用し，特定の MAC アドレスを持つホストからの通信をすべて遮断するようにする．

3.2.2　ワームの検知

本論文においては，ワームが存在しない状況ではダミーアドレスに対して接続が行われることはないと想定し，ダミーアドレスに対するコネクションが張られた時点でワームの存在を検知する．

3.2.3　ホスト停止によるワームの抑制

ダミーアドレスに対するコネクション（以降，ダミーコネクションという）が張られると，AMS はその送信元のホストをネットワークから切断する．このことを，ホストの停止と呼ぶ．ここで，ダミーコネクションを張ったホストのみの停止による，ワームの抑制性能について述べる．

今，あるネットワークにおいて内部アドレスリストを用いるワームに感染しているホストが 1 つあるとする．このワームに感染したホストは，Δt ごとに 1 つのアドレスに対して感染コネクションを張る．また，このネットワークにあるホストはすべてアドレスリストに n 個の通常アドレスと d 個のダミーアドレスを持っている．

感染しているホストが初めてコネクションを張ったときにダミーアドレスに当たる確率 P_d は，

$$P_d = \frac{d}{d+n} \tag{1}$$

となる．この場合，他のホストにコネクションを張ることなくワームを根絶することができる．

ところが，

$$P_n = 1 - \frac{d}{d+n} \tag{2}$$

の確率でダミーアドレス以外のアドレスにコネクションを行い，ワームは根絶されることな

く増殖する．次の Δt 後には 2 つのホストから感染コネクションが張られる．これらが両方ダミーコネクションを張ってワームが完全に停止する確率 P_{d2} は，

$$P_{d2} = \frac{d^2}{(d+n)^2} \tag{3}$$

であり，この P_{d2} は P_d より小さい．以降ワームが増殖していくたびにこの確率はさらに小さくなり，ワームの根絶が非常に困難となる．したがって，この方法のみで早期にワームを根絶させるためには，d の値を大きくして P_d の値を大きくしなければならない．しかし，ダミーアドレス数が大きくなると各ホストや AMS にとって負担となる．そこで，ダミーアドレス数を抑えつつ，ワームの早期抑制を実現することが求められる．

3.2.4　トレースバックによるワーム抑制

前述のとおり，ダミーコネクションを張ったホストのみを停止していたのでは，ワームの根絶は困難である．

本研究ではこれを解決するために，各ホストの接続履歴を参照し，これに従って感染の疑いがあるホストを追跡し，次々と停止させていくことを提案する．

ここで，感染の疑いがあるホストを以下の 3 種類に分類する．

- トリガホスト（TH）
  ダミーコネクションを張ったホスト
- 感染源疑惑ホスト（SH）
  トリガホストの感染源となりうるホスト
- 感染疑惑ホスト（IH）
  感染の疑いのあるホストからコネクションを受けたホスト

SH は，過去に TH へ対してコネクションを行ったホストのことを指す．また SH へ対して過去にコネクションを行ったホストも TH の元の感染源となりうるので，SH となる．

感染の疑いのあるホスト（TH，SH，IH）からのコネクションを受けたホストは，感染源ではないと推定されるが感染の疑いが強い．このホストを IH と呼ぶ．

図 3 は AMS から見るホスト A〜H の接続履歴の例を示している．

各アルファベットはホストを表し，Dum はダミーアドレスを表す．また，t_n は時刻を表す．t_n と t_{n+1} の間の時間を 1 ut とする．矢印は，始点の時刻においてあるホストから別のホストまたはダミーアドレスへの接続がなされたことを示す．図 3 ではホスト G が時刻 t_6 においてダミーコネクションを張っている．この場合，ダミーコネクションを張ったホスト G は TH となり，AMS がただちに停止させる．また，感染ホスト G から接続をトレースバックすることで SH，IH を発見し，停止させていく．

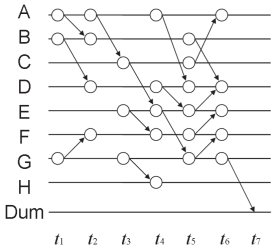

図 3　接続履歴
Fig. 3　Connection Histories.

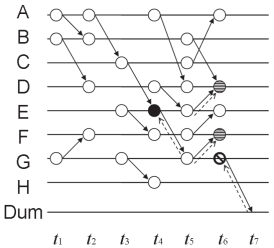

図 4　$N_t = 1$ の場合のトレースバック
Fig. 4　Traceback in case of $N_t = 1$.

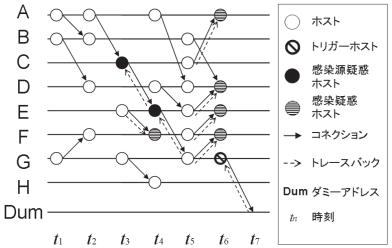

図 5　$N_t = 2$ の場合のトレースバック
Fig. 5　Traceback in case of $N_t = 2$.

ここで，トレース数とトレース時間というパラメータを設け，それぞれ N_t と T_t で表す．N_t は，SH を何ホップ遡って発見するか，ということを示す値であり，T_t は 1 ホップあたりどれくらいの時間（ut 単位）を遡るか，ということを表す．たとえば $N_t = 1$，$T_t = 2$ ut とした場合のトレースバックの様子を**図 4** に示す．まず，TH である G が t_4 以降にコネクションを張った先のホストは IH となり，停止される．図 4 ではホスト F がこれにあたる．また，TH である G に t_4 以降にコネクションを張った元のホストは SH となり，停止される．図 4 ではホスト E がこれにあたる．さらに，これらの感染の疑いのあるホストが現在までにコネクションを張った先のホストは IH となり，停止される．図 4 では SH である E が時刻 t_5 にコネクションを張った先のホスト D が IH となる．

$N_t = 2$ とした場合は，同様の作業を TH だけでなく $N_t = 1$ のときの SH に対しても行う．このときの様子を**図 5** に示す．ここでは，$N_t = 1$ の場合に加え，さらに SH であるホスト C，IH であるホスト A が停止される．

以降，N_t を増やした場合も同様で，SH をさらに遡って感染の疑いのあるホストを停止させていく．

以上のアルゴリズムをフローチャートで表したものを**図 6** に示す．なお，このフローチャートは 1 つのダミーコネクションが張られた場合のトレースバックアルゴリズムを表している．また，C [num] は AMS に蓄えられている時系列順に並んだ接続履歴を表し，num が大きいほど時間的に後の履歴となる．

ここで，本アルゴリズムの計算量について考える．図 6 において，サブルーチン trace-

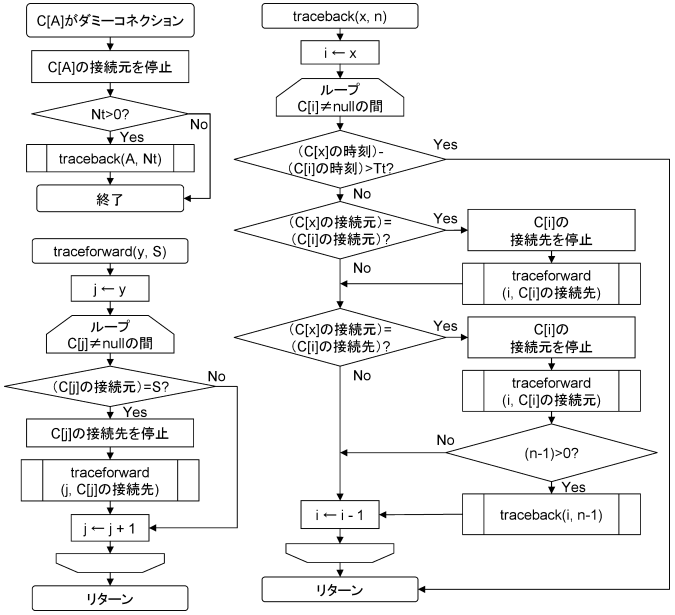

図 6　トレースバックアルゴリズム
Fig. 6　Algorithm of traceback.

back と traceforward は再帰的に呼び出され，トレースすべきコネクションを探しているが，traceback は自身からのみ呼ばれるのに対し，traceforward は traceback と traceforward の両方から呼び出される．したがって本アルゴリズムにおいて最も多く呼び出されるサブルーチンは traceforward であり，その中でループ部分が最も実行回数の多い場所であると考えられる．この部分の実行回数は，ホスト数を H，各ホストの平均コネクションインターバルを I，トレース数を N_t，トレース時間を T_t とし，$A = T_t/I$ とすると式 (4) のようになる（付録 A.1）．

$$N_e = 2H \sum_{i=1}^{N_t} \left\{ \left(e^A - 1\right)^i - A^i \right\} \tag{4}$$

ここで，式 (4) のオーダを求めるために $\sum$ をはずすことを考えるが，オーダを求める観点からは i が N_t 以外の場合は除外して考えてよい．したがって，式 (4) のオーダは以下のようになる．

$$O\left(He^{AN_T}\right) \tag{5}$$

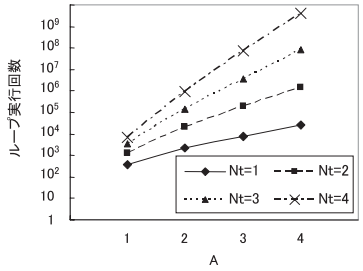

図 7　各 A，N_t におけるループ実行回数
Fig. 7　Number of loop running.

また，式 (4) に具体的な値を代入し，実際のトレースバックにかかる大まかな計算量を算出する．$H = 500$ とし，N_t と A を動かしたときのループ実行回数は**図 7** のようになる．

式 (5)，図 7 より，本アルゴリズムの計算量には N_t と A が大きく影響することが分かる．本提案手法の実用に向けては，これらのパラメータをできるだけ小さくする必要がある．

3.3　停止ホストの復旧

本手法では感染コネクションと通常コネクションの区別がつかないため，AMS は実際には感染していないホストも停止させる可能性がある．ワームの早期抑制のためには誤停止となるホストの犠牲はやむをえないものであるが，これらのホストは大変な迷惑を被るため，早期にネットワークに復旧させる必要がある．しかしながら，誤停止された各ホストはネットワークから切断されているため，管理者がスタンドアロンの状態で精査し，感染していないことが確認されたら再接続，という手順をふまなければならない．この作業は大きなコストがかかるため，復旧という視点から見ても停止ホスト台数はできるだけ少なくする必要がある．

4.　シミュレーション評価

本提案の有用性を示すために，C 言語でシミュレーションシステムを構築し，評価を行った．

4.1　シミュレーション条件

本シミュレーションでは 10,000 台のホストが存在するネットワークを想定する．また監

表 1　シミュレーションパラメータ
Table 1　Simulation Parameter.

シミュレーション回数：R	30 回
全ホスト数：N	10,000 台
初期感染台数：I_0	5 台
BA モデルにおける初期ホスト数：M_0	6
BA モデルにおける初期リンク数：M	5
平均アドレス数：A	15
ダミーアドレス数：d	5
平均通常コネクション間隔：B_n	5 ut
平均感染コネクション間隔：B_i	5 ut
平均ホスト停止遅延時間：Q	2 ut

視サーバは全ホストのコネクションログをリアルタイムに取得できるものとする．本シミュレーションにおける時間の単位を ut とする．シミュレーション開始時は，全 10,000 台のホストが通常の接続のみを行っている状態でシミュレーション時間を 100 ut 経過させ，その時点で 5 台のホストをワームに感染させる．ダミーコネクションが張られるたびにトレースバックによって感染している疑いがあるホストを停止させていく．活動中の感染ホストがすべてなくなった時点での，ワーム感染台数，誤停止台数，停止台数を記録する．このシミュレーションを各 30 回ずつ行い，その平均値によって評価を行う．

シミュレーションパラメータは**表 1** のように設定した．なお，ここで示す値はデフォルト値であり，特に断りのない限りはこの値でシミュレーションを行う．

ホストどうしのリンクは，実世界に近いものとなるよう，スケールフリーネットワーク[10]とした．スケールフリーネットワークを構築するため，BA モデル[11] に従ってホストの生成を行った．なお，BA モデルに従ってホストを生成した結果の平均アドレス数が 15 個となる．

1 台のホストがアドレスリスト中のあるホストに通常のコネクションを張る間隔，1 台の感染ホストが感染コネクションを張る間隔に対して平均値を設け，それぞれ $N(5\,\mathrm{ut}, 2\,\mathrm{ut}^2)$ の正規分布に従って変動する．なお，本提案手法では感染コネクション速度が通常通信と同程度である場合を想定しているため，この 2 つは同じ値とした．また，ホストを停止するのにかかる時間は，$N(2\,\mathrm{ut}, 1\,\mathrm{ut}^2)$ に従う．

4.2　N_t，T_t の設定

ここでは，N_t と T_t の最適値について考える．本手法は通常コネクションと同程度の速度でワームコネクションが張られることを想定しているので，以下の 2 つの条件についてシミュレーションを行い，最適値を決定した．なお，B_i はワームコネクションインターバルを，B_n は通常コネクションインターバルを表す．

- inf-5: $B_i = 1\,\mathrm{ut}$, $B_n = 5\,\mathrm{ut}$
- inf-1: $B_i = 5\,\mathrm{ut}$, $B_n = 5\,\mathrm{ut}$

ここで，inf-5 はワームが通常コネクションの 5 倍の速度で感染活動を行う場合を表し，inf-1 はワームと通常コネクションが同程度の速度の場合を表す．それぞれの条件に対し N_t と T_t を様々に変化させ，そのときの感染台数と誤停止台数，停止台数を記録した．感染台数に関する結果は**図 8** に示し，誤停止台数に関する結果は**図 9** に，停止台数に関する結果は**図 10** に示す．なお，N_t に関しては 3 以上の値では相当数の誤停止ホストが確認されたため，1

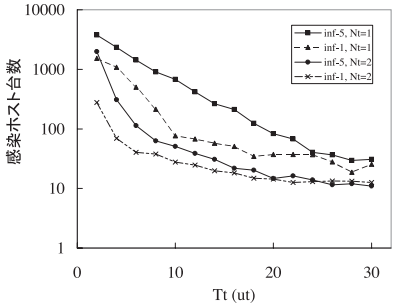

図 8　各 T_t，N_t に対する感染台数
Fig. 8　Infected hosts vs. T_t, N_t.

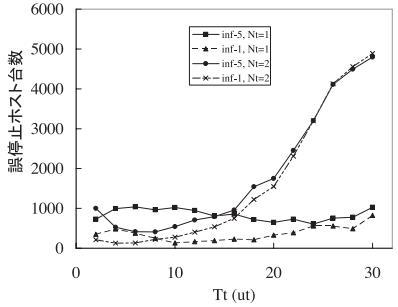

図 9　T_t，N_t に対する誤停止台数
Fig. 9　Falsely removed hosts vs. T_t, N_t.

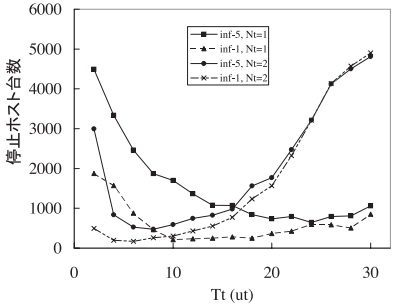

図 10　各 T_t，N_t に対する停止台数
Fig. 10　Removed hosts vs. T_t, N_t.

と 2 の場合のみ示している．

いずれの場合においても感染台数は T_t が増加するにつれて減少しているが，誤停止台数は極小を持つグラフとなっている．特に $N_t = 2$ の場合は T_t の増加に対する誤停止台数の増加が著しい．したがって，感染台数と誤停止台数の間にはトレードオフの関係が見てとれる．その結果として，停止台数に関しても極小を持つグラフとなっている．本手法では，停止台数をなるべく低く抑えることを目標としているので，停止台数が最小となるパラメータを最適値とする．$N_t = 1$ の場合を考えると，inf-5 における最適値は T_t が 10 から 14 程度のときであり，inf-1 における最適値は T_t が 24 から 26 のときである．これに対し，$N_t = 2$ の場合は inf-1 も inf-5 も T_t が 6-8 程度のときに最適となっており，条件による最適値のばらつきが少ない．したがって，$N_t = 2$，$T_t = 8$ を最適値としてデフォルトパラメータに採用する．

4.3　評 価 方 法

トレースバックによる効果

トレースバックを行う場合（本提案手法）と行わない場合での比較を行った．両者においてダミーアドレス数 d を変動させ，その際の感染，誤停止，停止ホスト台数を比較することによって評価を行った．なお，本提案手法におけるトレース時間 T_t は d によって最適値が異なるため，各 d における最適な T_t を用いて評価を行った．

各感染速度における効果

感染コネクションインターバル B_i を変化させ，様々な速度のワームにおいて感染，誤停止，停止ホスト台数がどれだけ抑えられるか評価を行った．

脆弱ホスト率に対する効果

ワームはホスト内の脆弱性を利用して感染活動を行う．これに対し，OS のバージョンアップやパッチの利用によって脆弱性を持たないホストが存在することも考えられる．そこで，全ホスト中に含まれる脆弱ホストの割合が本提案手法に与える効果を評価した．

ダミーアドレス保持率に対する効果

本提案手法はネットワーク内のすべてのホストがダミーアドレスを保持していることが前提となっている．ここで，全体の 80%のホストのみがダミーアドレスを保持している場合の効果について保持率 100%の場合と比較することで評価を行った．

4.4　シミュレーション結果

トレースバックによる効果

シミュレーション結果を**図 11**，**図 12**，**図 13** に示す．

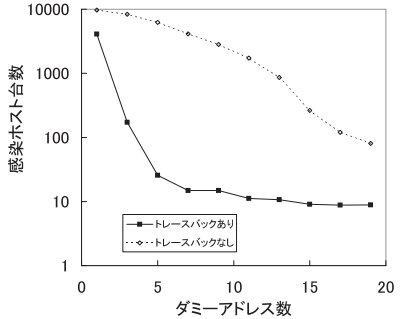

図 11　各手法における感染台数
Fig. 11　Number of infected hosts.

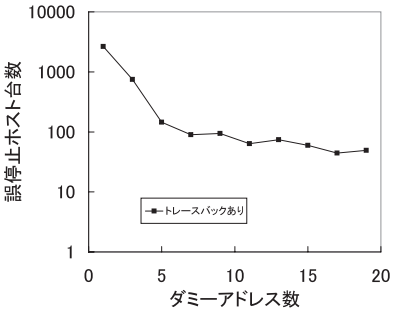

図 12　トレースバックありの場合の誤停止台数
Fig. 12　Number of falsely removed hosts with traceback.

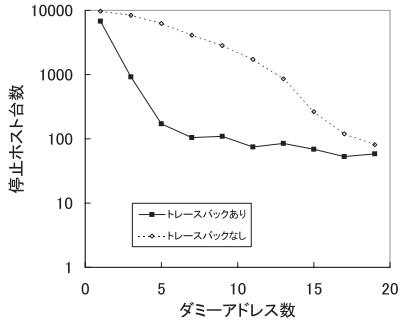

図 13　各手法における停止台数
Fig. 13　Number of removed hosts.

図 11 は，感染台数についての結果である．トレースバックを用いることによって感染台数が大幅に減少することが分かる．ダミーアドレス数 5 のときで比較すると，トレースバックなしでは 7,000 台（70%）あまりが感染した段階でワームが根絶するが，トレースバックありでは 100 台（1%）以下に抑えることが可能となっている．トレースバックを行わなくてもダミーアドレス数が 20 個近く挿入されれば感染台数を 1,000 台以下に抑えることができているが，ホストの持つ平均アドレス数が 15 個であることを考えるとこのダミーアドレス数は多すぎる．少ないダミーアドレス数でも感染台数を抑えることができるという点で，

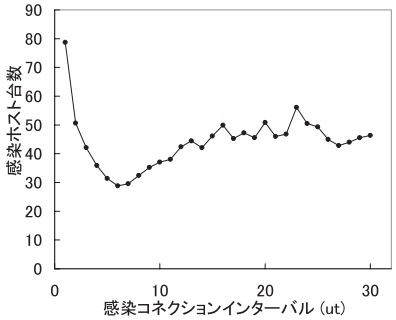

図 14　各感染インターバルにおける感染台数
Fig. 14　Infected hosts vs. infection interval.

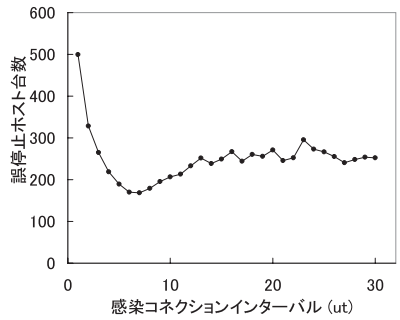

図 15　各感染インターバルにおける誤停止台数
Fig. 15　Falsely removed hosts vs. infection interval.

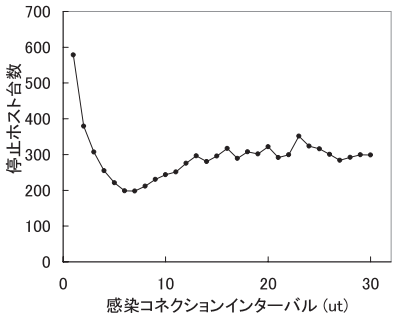

図 16　各感染インターバルにおける停止台数
Fig. 16　Removed hosts vs. infection interval.

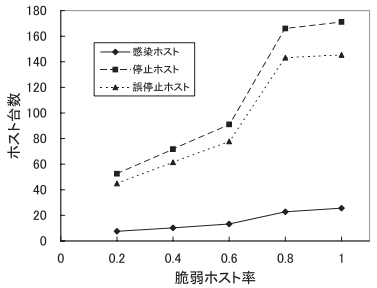

図 17　各脆弱ホスト率に対する結果
Fig. 17　vs. Vulnerable host rate.

トレースバックを行う本提案手法は優れている．

図 12 は，誤停止台数についての結果であり，図 13 は停止台数についての結果である．トレースバックを行わない手法は当然ながら誤停止は存在しないため，図 12 はトレースバックを行った場合のみの結果となる．3 つのグラフと比較すると，すべて同じような形のグラフになっており，ダミーアドレス数を増加させて感染台数を減少させた場合，それにともなって誤停止台数，停止台数も減少していることが分かる．また，トレースバックを行う手法の場合，感染台数に対して誤停止台数はつねに 6 倍程度となっている．これは一見多い数字であるが，誤停止が 0 であるトレースバックなしの手法よりも大幅に感染台数を減少させることが可能となっており，結果として図 13 に示される総停止台数も大幅に減少している．また，感染が見つかった段階ですべてのホストをネットワークから切断し，それ以降の拡散を防ぐという単純な手法と比較すると，誤停止台数は 20 分の 1 以下となっている．以上より，トレースバックを行う本提案手法は誤停止ホストをともなうものの，結果として少ない総停止台数でワームの根絶が可能となることが分かる．

各感染速度における効果

シミュレーション結果を**図 14**，**図 15**，**図 16** に示す．

図 14〜図 16 は，感染コネクションインターバル B_i を 1 ut から 30 ut まで動かしたときの感染台数，誤停止台数，停止台数を表している．これは通常コネクションと比較すると 1/6 倍から 5 倍の速度である．

結果は，3 つのグラフすべてにおいて感染コネクションインターバルが 5 ut 程度の場合が最もパフォーマンスが良くなっている．これは，本シミュレーションにおけるパラメータがインターバル 5 ut の場合の最適値であるためと考えられる．それよりも高速な（インターバルが短い）ワームでは感染台数，停止台数が著しく増加しているのに対し，比較的遅い（インターバルが長い）ワームでは感染ホスト 50 台程度（全体の 0.5%），停止ホスト 300 台程度（3%）の安定した値が得られている．これは，ワーム速度が遅くなった場合，全ホスト抑制までに時間がかかるのと同時にワームの感染の広がりにも時間がかかるためと考えられる．高速なワームに本提案手法は対応できない結果となっているが，通常の 5 倍以上もの速度で感染活動を行うワームの場合，必然的にトラフィックに異常をきたすと考えられるので，他のアプローチによる抑制が可能となる．

以上の結果より，本提案手法は感染インターバルが長く，通常コネクションに紛れ込んでしまうワームに対して特に有効であるといえる．

脆弱ホスト率に対する効果

シミュレーション結果を**図 17** に示す．

図 17 に示されるように，脆弱ホスト率が減少すればするほど感染ホスト台数，停止ホスト台数，誤停止ホスト台数も減少していることが分かる．直感的に考えると，本提案手法では感染の有無を問わずトレースバックを行うので，脆弱ホストが減少した場合誤停止ホストが増加しそうであるが，そうはなっていない．これは，脆弱ホスト率の減少にともなってワームの拡散速度が低下し，結果的にワームが広がりを見せる前にすべての感染ホストが

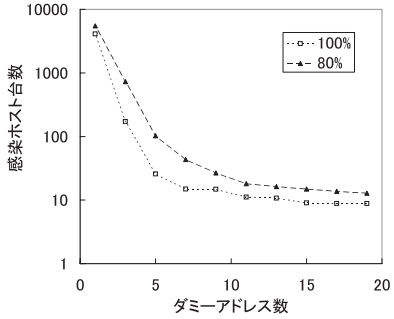

図 18　ダミーアドレス保持率を変化させた場合の感染台数
Fig. 18　Infected hosts vs. percentage of hosts having dummy addresses.

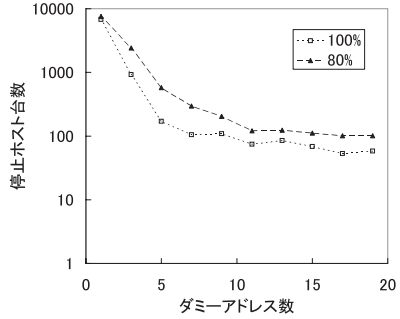

図 19　ダミーアドレス保持率を変化させた場合の停止台数
Fig. 19　Removed hosts vs. percentage of host having dummy addresses.

停止させられてしまうためと考えられる．この結果より，本提案手法は脆弱ホスト率が低い場合でも有効であることが示された．

ダミーアドレス保持率に対する効果

シミュレーション結果を**図 18** と**図 19** に示す．

ダミーアドレス保持率 80%の場合，100%の場合と比較してパフォーマンスが大きく劣る結果となっている．たとえば $d = 5$ の場合，80%における感染ホスト，停止ホストは 100%の場合の 4 倍になってしまっている．100%の場合と同等のパフォーマンスを得るためには，およそ 2 倍のダミーアドレス数を必要としており，わずか 20%の保持率の差が結果に大きく影響していることが分かる．したがって，本提案手法ではすべてのホストにダミーアドレスを挿入することは非常に重要であるといえる．

5. ま と め

本論文では，内部アドレスリストを利用して感染活動を行うワームに対して有効な検知・抑制手法を提案した．各ホストにダミーアドレスと呼ばれる通常では使用されないアドレスを挿入し，ここへのコネクションをワームと判断することで検知が行える．また，ダミーアドレスにコネクションを張ったホストと，事前にそのホストへコネクションを張っているホストは感染疑惑があると判断し，ネットワークから切断することで以後の感染の広がりを抑える．これをダミーアドレスへのコネクションが張られるたびに繰り返し，最終的にワームの根絶を目指す．

本提案手法のシミュレーション評価を行ったところ，通常コネクションと同等か，もしくは遅い速度で感染活動を行うワームに対して特に有効であり，感染台数を全体の 1%以下，停止台数を 5%以下に抑制することが可能となった．したがって，本提案手法は内部アドレスリストを利用して感染活動を行うワームの検知・抑制に有効であることが示された．

謝辞　本研究の一部はセコム科学技術振興財団の支援により行われた．

参 考 文 献

1) Zou, C.C., Gong, W. and Towsley, D.: Code Red Worm Propagation Modeling and Analysis, *Proc. 9th ACM Conference on Computer and Communications Security* (2002).
2) Williamson, M.M.: Throttling Viruses: Restricting propagation to defeat malicious mobile code, *Proc. Computer Security Applications Conference* (2002).
3) Schechter, S.E., Jung, J. and Berger, A.W.: Fast Detection of Scanning Worm Infections, *Recent Advances in Intrusion Detection* (2004).
4) Weaver, N., Paxson, V., Staniford, S. and Cunningham, R.: A Taxonomy of Computer Worms, *Proc. 2003 ACM Workshop on Rapid Malcode* (2003).
5) Symantec.com: W32.Bropia, Symantec.com (online). available from http://www.symantec.com/avcenter/venc/data/w32.bropia.html (accessed 2009-3-9)
6) Huang, C., Jhonson, N.L., Janies, J. and Liu, A.X.: On Capturing and Containing E-mail Worms, *Proc. International Performance Computing and Communications Conference* (2006).
7) Cyber Clean Center 注意喚起活動について，Cyber Clean Center（オンライン）．入手先 https://www.ccc.go.jp/activity/index.html (参照 2009-3-9)
8) IP Messenger, IP Messenger（オンライン）．入手先 http://www.ipmsg.org/ (参照 2008-3-9)
9) Cranor, C., Gao, Y., Johnson, T., Shkapenyuk, V. and Spatscheck, O.: Gigascope: a stream database for network applications, *Proc. ACM Special Interest Group on Management Data* (2003).
10) Zou, C.C., Towsley, D. and Gong, W.: Email Worm Modeling and Defense, *Proc. 13th International Conference on Computer Communications and Networks* (2004).
11) 石田晋哉，荒川伸一，村田正幸：べき乗則に従う WDM ネットワークにおける論理トポロジー設計，電子情報通信学会技術研究報告，PN2003-27 (2003).

付　　録

A.1　提案アルゴリズムの計算量

図 6 における traceforward 中の検索実行回数 N_e は，以下の式で表される．

$$N_e = \sum_{i=1}^{N_t} \left\{ O_{tf}(i) \times T_{tf}(i) \times \frac{H}{I} \right\} \tag{6}$$

H：ホスト台数
N_t：トレース数
T_t：トレース時間
I：1 台のホストの平均コネクションインターバル
$O_{tf}(i)$：i 回 traceback した際の traceforward 起点数
$T_{tf}(i)$：i 回 traceback された起点ホスト 1 台からの再帰を含む総検索時間
$\frac{H}{I}$：単位時間あたりの総接続履歴数

では $O_{tf}(i)$ を考える．1 台のホストから 1 回の traceback によってさかのぼられるホスト台数は，T_t の間に張られるコネクション数と等しいので，$\frac{T_t}{I}$ となる．しかしながら，図 6 に示されるように，T_t 内に含まれるコネクションのうちトリガホストがディスティネーションホストになっているものに対しては traceback を，トリガホストがソースホストとなっているものに対しては traceforward を行うので，結果的に起点となるホスト数は $2 \times \frac{T_t}{I}$ となる．2 回以上 traceback が行われる場合，それぞれのホストから再帰的にさかのぼる．したがって，$O_{tf}(i)$ は以下の式で表される．

$$O_{tf}(i) = 2 \times \left(\frac{T_t}{I} \right)^i \tag{7}$$

続いて $T_{tf}(i)$ を考える．

ある 1 台のホストから traceforward が開始され，その開始時刻を 0，終了時刻を F とする．ここで，時刻 t における traceforward 実施ホスト数を $J(t)$ とすると，時刻 (t) における $J(t)$ の増加量は $J(t)$ とコネクション速度（コネクションインターバル I の逆数）の積になるので，$J(t)$ に関する以下の微分方程式が成り立つ．

$$\frac{J(t)}{dt} = \frac{1}{I} J(t) \tag{8}$$

この微分方程式の解は以下のようになる．

$$J(t) = C \exp\left(\frac{t}{I}\right) \quad (C \text{ は任意定数}) \tag{9}$$

$J(0) = 1$ より，式 (9) は以下のようになる．

$$J(t) = \exp\left(\frac{t}{I}\right) \tag{10}$$

ここで，1 台のホストからの traceforward による総検索時間は $J(t)$ と微小時間 dt の積を時刻 0 から F まで積分したものと考えられ，以下の式で表される．

$$\int_0^F \exp\left(\frac{t}{I}\right) dt \tag{11}$$

式 (11) を解くと以下のようになる．

$$I\left(\exp\left(\frac{F}{I}\right) - 1\right) \tag{12}$$

次に，F のとりうる範囲と確率について考える．$i = 1$ の場合を考えると，F のとりうる範囲は $0 \le F \le T_t$ であり，その確率密度関数 $P(F)$ は一様に $\frac{1}{T_t}$ となる．したがって，F の確率分布を考慮した $i = 1$ の場合の総検索時間 $T_{tf}(1)$ は以下のようになる．

$$T_{tf}(1) = \int_0^{T_t} I\left(\exp\left(\frac{F}{I}\right) - 1\right) \frac{1}{T_t} dF \tag{13}$$

式 (13) を解くと以下のようになる．

$$T_{tf}(1) = \frac{I^2}{T_t}\left(\exp\left(\frac{T_t}{I}\right) - 1\right) - I \tag{14}$$

続いて $i = 2$ のときを考える．F のとりうる範囲と確率密度関数 $P(F)$ は以下のようになる．

$$0 \le F \le 2T_t \tag{15}$$

$$P(F) = \begin{cases} \dfrac{F}{{T_t}^2} & (0 \le F \le T_t) \\ \dfrac{2T_t - F}{T_t^2} & (T_t \le F \le 2T_t) \end{cases} \tag{16}$$

したがって，$T_{tf}(2)$ は以下のようになる．

$$T_{tf}(2) = \int_0^{T_t} I\left(\exp\left(\frac{F}{I}\right) - 1\right) \frac{F}{T_t^2} dF + \int_{T_t}^{2T_t} I\left(\exp\left(\frac{F}{I}\right) - 1\right) \frac{2T_t - F}{T_t^2} dF \tag{17}$$

式 (17) を解くと，以下のようになる．

$$T_{tf}(2) = \frac{I^3}{{T_t}^2}\left(\exp\left(\frac{T_t}{I}\right) - 1\right)^2 - I \tag{18}$$

$i = 3$ 以降も同様の計算を行うと，$T_{tf}(i)$ は結局以下のように定義される．

$$T_{tf}(i) = \frac{I^{i+1}}{{T_t}^i}\left(\exp\left(\frac{T_t}{I}\right) - 1\right)^i - I \tag{19}$$

式 (7)，(19) より，式 (6) は以下のようになる．

$$\begin{aligned} N_e(i) &= \sum_{i=1}^{N_t}\left[2 \times \left(\frac{T_t}{I}\right)^i \times \left\{\frac{I^{i+1}}{{T_t}^i}\left(\exp\left(\frac{T_t}{I}\right) - 1\right)^i - I\right\} \times \frac{H}{I}\right] \\ &= 2H\sum_{i=1}^{N_t}\left[\left\{\exp\left(\frac{T_t}{I}\right) - 1\right\}^i - \left(\frac{T_t}{I}\right)^i\right] \end{aligned} \tag{20}$$

(平成 20 年 8 月 7 日受付)
(平成 21 年 3 月 6 日採録)

稲場　太郎（学生会員）

2007 年慶應義塾大学理工学部情報工学科卒業．現在，同大学大学院理工学研究科修士課程在学中．ネットワークセキュリティ，デジタルフォレンジックスに関する研究に従事．

川口　信隆（正会員）

2005 年慶應義塾大学大学院理工学研究科開放環境科学専攻前期博士課程修了．2008 年同大学院理工学研究科開放システム科学専攻後期博士課程修了．博士（工学）．ネットワークセキュリティ，デジタルフォレンジックスの分野に興味を持つ．IEEE 会員．

岡田　謙一（フェロー）

慶應義塾大学理工学部情報工学科教授，工学博士．専門は，CSCW，グループウェア，ヒューマン・コンピュータ・インタラクション．情報処理学会誌編集主査，論文誌編集主査，GW 研究会主査等を歴任．現在，情報処理学会 MBL 研究会運営委員，BCC 研究グループ主査，日本 VR 学会理事，CS 研究会委員長．情報処理学会論文賞（1996，2001，2008 年），情報処理学会 40 周年記念論文賞，日本 VR 学会サイバースペース研究賞，IEEE SAINT'04 最優秀論文賞を受賞．情報処理学会フェロー，IEEE，ACM，電子情報通信学会，人工知能学会各会員．