社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

観測トラヒックの統計的性質を利用した DDoS Attack の検出方法

大下　裕一[†]　　阿多　信吾[††]　　村田　正幸[†††]

† 大阪大学 大学院情報科学研究科
†† 大阪市立大学 大学院工学研究科
††† 大阪大学 サイバーメディアセンター
E-mail: †y-ohsita@ist.osaka-u.ac.jp, ††ata@info.eng.osaka-cu.ac.jp, †††murata@cmc.osaka-u.ac.jp

あらまし　近年，公開サーバに対する分散サービス拒否 (DDoS) 攻撃はますます深刻を増しており，早急な対策が望まれる．特に，TCP の仕様を悪用した SYN Flood 攻撃は，簡単な方法で容易にサーバを停止状態にできることから，現在最も多く利用されている．SYN Flood 攻撃の検出は，SYN パケットの到着レートを用いて行うのが一般的であるが，到着レートは時刻により変化することから，通常トラヒックと攻撃トラヒックを明確に区別することは難しい．そこで本稿では，通常トラヒックの統計的性質を利用した，SYN Flood 攻撃の検出手法を提案する．提案手法では，トラヒックモニタを用いて正常トラヒックの到着レートの特性を統計的手法によりモデル化する．その結果に基づき，攻撃トラヒックの検出アルゴリズムを新たに提案する．性能評価より，正常トラヒックの到着レート変動が正規分布によりモデル化できることを示す．さらに提案手法が，攻撃トラヒックの存在を時間的変動を考慮しつつより明確に検出できることを示す．

キーワード　DDoS 攻撃, SYN Flood, 統計解析, 正規分布, トラヒックモニタリング

Detecting Distributed Denial of Service Attacks by utilizing statistical analysis of TCP SYN packets

Yuichi OHSITA[†], Shingo ATA[††], and Masayuki MURATA[†††]

† Graduate School of Information Science and Technology, Osaka University
†† Graduate School of Engeneering, Osaka City University
††† Cyber Media Center, Osaka University
E-mail: †y-ohsita@ist.osaka-u.ac.jp, ††ata@info.eng.osaka-cu.ac.jp, †††murata@cmc.osaka-u.ac.jp

Abstract　Recently DDoS (Distributed Denial of Service) attacks to public servers become more serious. SYN Flood attacks which misuse the specification of TCP (Transmission Control Protocol) are used most frequently since the malicious attackers can easily generate attacking traffic to make public servers unavailable. More quick and accurate defence mechanisms against DDoS traffic (especially SYN Flood) are more important to keep survive of services. One of difficult problems of detecting SYN Flood traffic is that server nodes or firewalls cannot distinguish SYN packets of normal TCP connections from SYN Flood attacked packets. Moreover since the rate of traffic may vary by time, we cannot use an explicit threshold of SYN arrival rates to detect the SYN Flood traffic. In this paper we introduce more accurate detection mechanism of SYN Flood traffic by taking time variance of arrival traffic into consideration. We first investigate the statistics of arrival rates of both normal TCP SYN packets and SYN Flood attack packets. We then propose a new detection mechanism based on the statistics of SYN arrival rates. Our results have shown that we can model the arrival rate of normal TCP SYN packets to the normal distribution. By using our analytical results we show that our proposed mechanism can detect SYN Flood traffic more quickly and accurately regardless of time variance of the traffic.

Key words　Distributed Denial of Service (DDoS), SYN Flood, Statistical Analysis, Normal Distribution, Traffic Monitoring

1. はじめに

近年，インターネットの急速な発展により，ネットワークを介した，様々なサービスが提供され，その利便性は増すばかりである．その一方で，悪意を持った第三者がサービスを提供する計算機に攻撃を行い，一般ユーザの利用を妨げるサービス拒否 (Denial of Service; DoS) 攻撃が深刻な問題となってきている．実際に，yahoo! や amazon といった大手サイトもこれらの攻撃の被害にあっており，多大な損失がでている [1].

攻撃の技術も近年ますます高度化しており，各地に分散された複数の攻撃ノードが同時に同じサーバを攻撃する DDoS (Distributed DoS) 攻撃と呼ばれるものが主流になりつつある．DDoS 攻撃においては，攻撃者は計算機のもつ脆弱性を悪用して複数の端末に不正に侵入し，攻撃を実行するプログラムを実行可能状態で待機させる．そして，一斉に攻撃命令を送ることにより，複数端末から同時に攻撃が開始される．このため，各端末が生成する攻撃トラヒックがさほど影響を与えないものであっても，同時に攻撃を行う端末が非常に多ければ，サーバへの影響は深刻になる．

DDoS 攻撃にも多種多様なものが確認されており，攻撃対象ホストに対して許容量を超える接続要求を行い一般ユーザが新たな接続を確立できなくさせる SYN Flood 攻撃や，不要な ICMP パケットを意図的に攻撃対象ホストに大量に送信することで帯域を占有する Smurf 攻撃などがある．なかでも，SYN Flood 攻撃は，簡単な方法でサーバを接続拒否状態に陥らせることができるため，現在最も多く行われており， DoS 攻撃の約 90% に該当する [2].

通常，TCP における接続確立は 3-way handshake と呼ばれる図 1 (a) に示されるような手順で行われる．まず，送信元ホスト (Client) から宛先ホスト (Server) へ SYN パケットを送信する．次に，宛先ホストから送信元ホストへ SYN パケットに対する受信確認である SYN/ACK パケットを送信する．そして，送信元ホストから宛先ホストへ ACK パケットを送り接続を確立する．その後実際のデータの転送を行う．

宛先ホスト側において，SYN/ACK パケットを送信してからそれに対する ACK パケットを受け取るまでは half-open 状態と呼ばれ，送受信バッファの確保等を行うなど送信元ホストとの通信に必要な準備を行う．ホストの資源は有限であるため，half-open 状態の TCP 接続数には上限値が設定されており，これらは backlog-queue と呼ばれる待ち行列で管理されている．上限値を超える接続要求 (SYN) パケットに対しては，これ以上接続ができないことを示す受信拒否 (RST) パケットを送信元に返す.

SYN Flood 攻撃はこれを悪用した攻撃であり，図 1 (b) に示されるように，攻撃ホストは送信元フィールドを詐称した SYN パケットを送り出す．SYN パケットを受信したサーバは，詐称されたアドレスに対して SYN/ACK パケットを送信することになる．詐称されたアドレスが実際に存在するのであれば，詐称されたアドレスに該当するホストは RST パケットを送信し接続要求を破棄するが，詐称されたアドレスがネットワーク上に存在しない場合，宛先ホストは破棄されたことを知ることができず，タイムアウトになるまで ACK パケットの到着を待ち続けることになる．その結果，SYN Flood 攻撃を受けているホストでは backlog-queue に攻撃パケットが次々と蓄積され，結果的に一般ユーザからの接続要求のほとんどが backlog-queue に空きがないために拒否される．

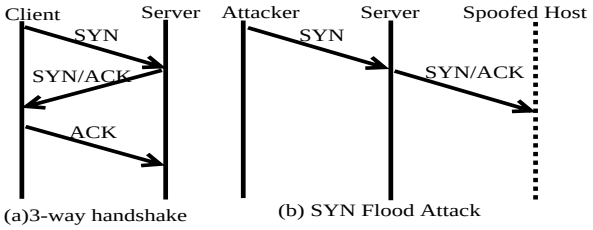

図 1　3-way handshake と SYN Flood の概要

SYN Flood に用いられるパケットは，送信元アドレスが詐称されていることをのぞけば通常の TCP SYN パケットと相違がないため，攻撃対象ホストにおいて通常の TCP SYN パケットと攻撃パケットを区別することは容易ではない．このため，一般的に SYN Flood の攻撃を検出し防御することは難しく，現在でも様々な対策が考えられている．

SYN Flood 攻撃を通常のトラヒックと区別するもっとも確実な方法として，ルータにおいて各 TCP コネクションごとに状態を保持し，3-way handshake が完了した正常な TCP フローのパケットのみを送信先ホストに渡す方法が考えられる．一部のファイアウォール製品では，SYN パケットを受信すると送信先ホストの代理で SYN/ACK パケットを返送し，そのACK を受信したパケットに限ってサーバに実際の SYN パケットを届ける方法でサーバへの攻撃を軽減させるものもある [3]. しかしながら，これらの方法はファイアウォールやルータにおいて TCP フローごとに状態を保持しなければならないため，コストが高い．さらに，瞬間的な攻撃については上記の方法で回避することが可能なものの，長期にわたり継続的に攻撃トラヒックが発生した場合，ファイアウォールが保持できる状態数以上のトラヒックを防ぐことはできない．このような場合，可能な限り早く攻撃トラヒックを特定した上で，ルータでパケット棄却，レート制御などのフィルタを設定する必要がある．

攻撃トラヒックの検出手法として，これまでにもルータ上で双方向のトラヒックを観測し，そのレートの不整合性から攻撃を検出する方法 [4]，TCP コネクションの先頭パケット (SYN) と終端パケット (FIN あるいは RST) の数の差より攻撃を検出する方法 [5] などがある．しかしながら，これらの手法ではサーバが攻撃の被害を受けるまで検出できない，攻撃の検出までに最低でも TCP コネクションが終了する程度の時間はかかる等の問題点がある．また，これらの方法は正常なトラヒックを含むトラヒック全体の時間的性質を考慮していないために，一時的に負荷所集中した場合に攻撃と誤検出されることもある．このため，これらの性質を考慮することで，さらに短期間で攻撃トラヒックを検出できる可能性がある．

以上のことから，本稿ではトラヒックの時間的変動を考慮す

ることで，より高速かつ確実に攻撃トラヒックを検出する手法を新たに提案する．本研究では，提案手法により特定できた攻撃トラヒックに対しては，ルータにおいてパケット棄却あるいはレート制御のフィルタを設定することで攻撃に対する防御を行うものとする．したがって，本稿で対象とする攻撃トラヒックは，サーバを長期的にサービス拒否状態に陥らせることが可能であるものとする．具体的な攻撃トラヒックの定義については 3. で説明する．

以降，2. で提案手法で必要となるトラヒックの統計的解析方法およびその結果を示し，提案する攻撃トラヒック検出手法のアルゴリズムについて説明する．そして，3. でシミュレーションによる提案手法の評価を行い，最後に 4. でまとめと今後の課題について述べる．

2. 実トラヒックの統計的解析と攻撃トラヒックの検出手法

本章では，通常トラヒックのモデル化に必要となる，トラヒックの収集，ならびに統計的分析結果について説明する．さらに，攻撃トラヒックの検出アルゴリズムについて説明する．

2.1 実トラヒックのモニタリングと分類

実トラヒックの統計情報を得るために，トラヒックモニタを用いて実トラヒックの収集を行う．大阪大学と外部をつなぐ 1000Base-SX の光ファイバケーブルを光スプリッタにより分波し，それぞれをキャプチャマシンのギガビット NIC のデータ受信側 (RX) に接続する．そして，トラヒックモニタ上で時系列に到着パケットのヘッダを記録する．パケットの記録には tcpdump を用いる．

次に，得られたパケットを TCP フローごとに分類し，それぞれが正常な TCP 通信で用いられたパケットであるかどうかを分類する．まず，同一の送信元アドレス，送信元ポート番号から，同一の宛先アドレス，宛先ポート番号に送信されているパケットの列を一つのフローとして考える．そして，フローごとに以下の 5 つのグループに分類する．

グループ N　3-way handshake が完了し，データの送受信が確認された後，終端パケット (FIN/RST) により終了しているフロー．正常に通信が行われたフローを表す．

グループ Rs　宛先ホストからの SYN/ACK 到着前に RST により終端しているフロー．宛先ホストにおいてサービスが提供されていない状態などが原因と考えられる．

グループ Ra　送信元ホストが SYN/ACK に対する ACK を送る前に RST により終端しているフロー．何らかの理由ですでに存在しないホストに対して SYN/ACK パケットを送った場合などが考えられる．

グループ Ts　SYN/ACK パケットが送られる前にタイムアウトが発生．SYN/ACK パケットの消失，詐称アドレスを用いた SYN パケットの送信などが考えられる．

グループ Ta　SYN/ACK パケットに対する ACK パケットが送られる前にタイムアウトが発生．ACK パケットの消失などが考えられる．

以上の分類後，各グループのトラヒックについて到着レートの時間的変動を調べる．

表 1　フローの分類

グループ	フローの個数	割合 (%)
グループ N	18,147,469	85.1
グループ Rs	622,976	2.9
グループ Ra	75,432	0.3
グループ Ts	2,435,228	11.4
グループ Ta	2,009	0.0

2.2 取得トラヒックの時間的性質とモデル化

本稿ではサンプルトラヒックとして，2003 年 5 月 20 日 17 時 55 分から 5 日間の記録を用いる．サンプルトラヒックは，外部から学内向けのトラヒックの平均レートが約 12.0 Mbps，学内から外部への送信レートは平均約 22.4 Mbps であり，送信レートが高い昼間の時間帯では学内向けが約 37 Mbps，学外向けが約 55 Mbps である．また，サンプル内に含まれる TCP パケットは 1,983,116,637 個，そのうち SYN パケットは 21,615,220 個である．サンプル内のフロー数は SYN パケットの再送のため，SYN パケットの個数よりやや少ない 21,283,114 個である．

はじめに，サンプルトラヒックを上で述べたグループごとに分類した結果を表 1 に示す．ここでは，タイムアウト時間を 180 秒とする．すなわち，終端パケットが到着していない各フローについて最終パケット到着時刻から 180 秒経過したフローは，すでに終了したものとして判断する．

次に，すべてのフロー，グループ N のフロー，およびそれ以外のフローについて，SYN パケットの到着レートの時間的変動を比較した結果をそれぞれ図 2，3，4 に示す．これらの結果より，トラヒック全体において急激に到着レートが上昇している箇所は，正常に通信が行われていないトラヒックが原因であると考えることができる．しかしながら一方では，正常トラヒックだけを着目しても時間帯によってレートが大きく異なることから，単一の SYN 到着レートを閾値として攻撃トラヒックの検出を行うと，数多くの正常トラヒックを攻撃と誤検出することがわかる．さらに，正常トラヒックとそれ以外のトラヒックのレート変動を比較すると，正常でないトラヒックのレート分布は非常にすそのが広いと考えられるが，正常トラヒックの変動についてはそれほど分布のすその部分が大きくないと考えられる．

このことを示すため，正常トラヒックを正規分布によりモデル化することを考える．正常トラヒックのフローについて，一定間隔ごとに到着レートを求め，その平均および分散を導出する．次に，これらをパラメータとした正規分布を考える．

正規分布の分布関数は以下の式で与えられる．ここで，ζ，σ はそれぞれ到着レートの平均，分散を表す．

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} exp[\frac{-(y-\zeta)^2}{2\sigma^2}]dy \qquad (1)$$

ただし，式 (1) は $(-\infty, \infty)$ 領域での正規分布であり，一方到着レートは $[0, \infty)$ の範囲で変動することから，本稿では式 (1) の非負領域のみを用いる．すなわち，

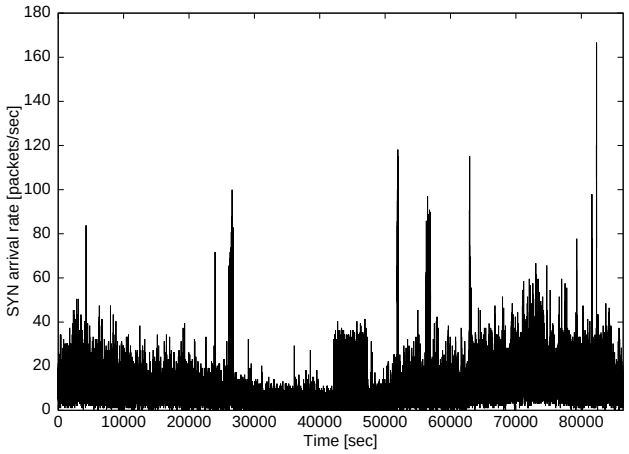

図 2　SYN パケットの到着レートの時間変動 (すべてのフロー)

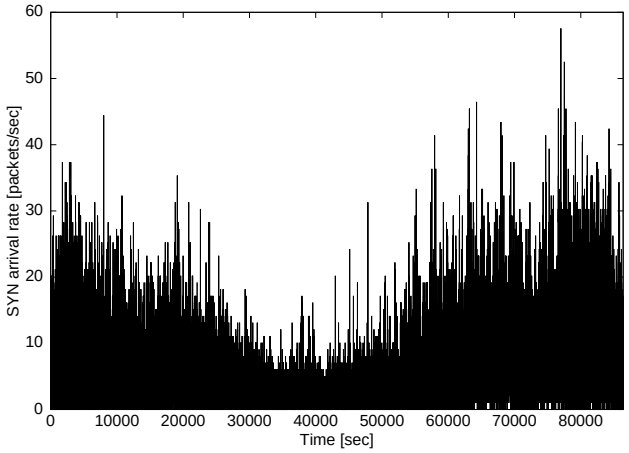

図 3　SYN パケットの到着レートの時間変動 (グループ N)

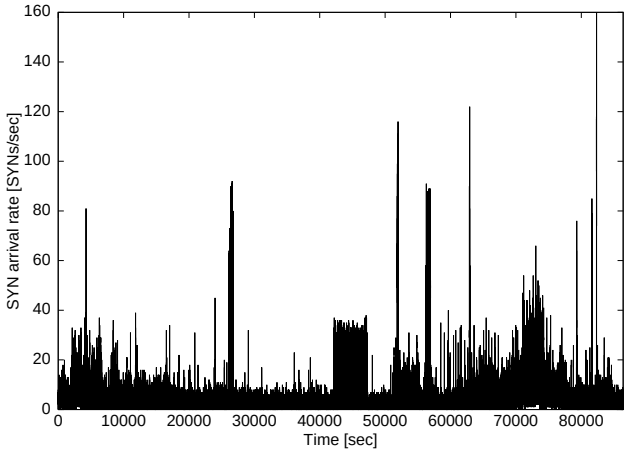

図 4　SYN パケットの到着レートの時間変動 (グループ Ra, Rs, Ta, Ts)

$$G(x) = \frac{F(x) - F(0)}{1 - F(0)} \tag{2}$$

で与えられる分布 $G(x)$ によってモデル化を行う．

取得したトラヒックのうち，正常なトラヒックに分類されたグループ N のトラヒックについて，式 (2) を用いてモデル化を行った結果を図 5 に示す．図は，3 時，9 時，19 時における SYN トラヒックレートの累積分布と，その平均および分散を

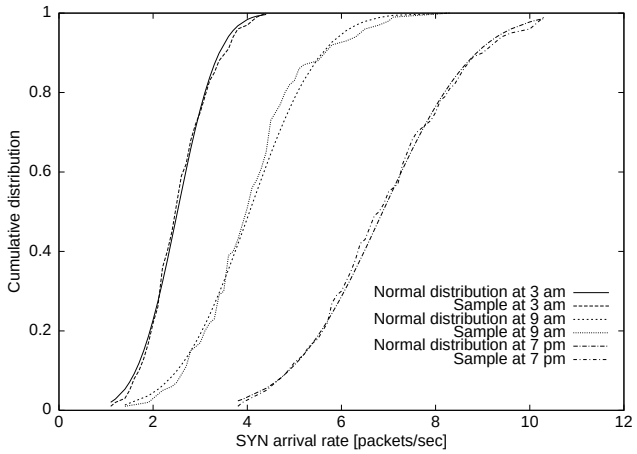

図 5　正常なフローにおける SYN パケットの到着レートの分布と同一平均，分散を持つ正規分布

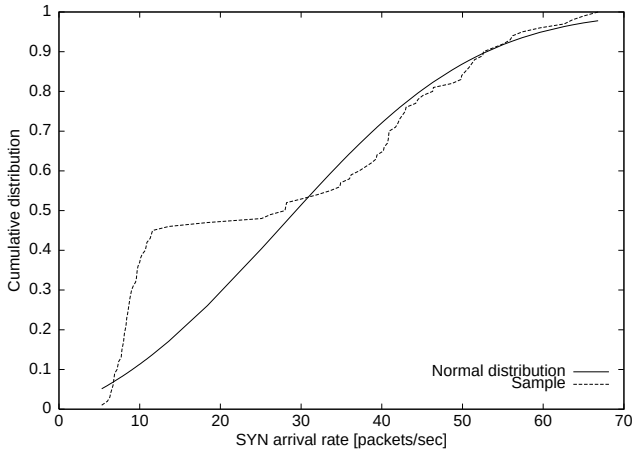

図 6　攻撃開始地点における SYN パケットの到着レートの分布

もとに作成した正規分布の比較を示している．この結果より，正常トラヒックの SYN レート分布は，ほぼ正規分布によりモデル化が可能であることがわかる．また，異なる時間帯においても同様の結果が得られていることより，時間帯に依存せず適用可能であることがわかる．

一方，攻撃トラヒックを含む全体の SYN レート分布は正規分布とは大きく異なっている．図 6 に攻撃が行われた時間帯におけるトラヒック全体の SYN レート分布と，その時の正規分布によるモデル化結果との比較を示す．このように，高い SYN レートが継続して発生すると，分布のすその部分が大きくなり，結果として正規分布ではモデル化できないことが示されている．

2.3　トラヒック変動を考慮した攻撃トラヒックの検出手法

以上のことより，正常でないトラヒックを検出するためには，SYN レートの分布を調べ，正規分布からどの程度離れているかを調べることが有効な手法であることがわかる．そこで，正規分布からの平均二乗誤差を導出し，その値の大きさによって攻撃箇所を特定する方法を考える．はじめに，モデル化に用いる SYN レートのサンプル数を n，直近 n 個の SYN 到着レートを $r_1, r_2, \ldots, r_n$ とする．ただし，$r_i (1 \leqq i \leqq n)$ はあらかじめ昇順にソートされているものとする．このとき平均二乗誤差 D を以下の式で与える．

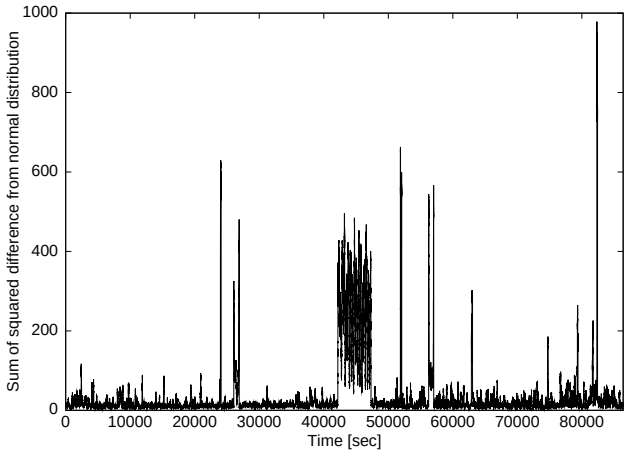

図 7　平均二乗誤差の変動 (すべてのフロー)

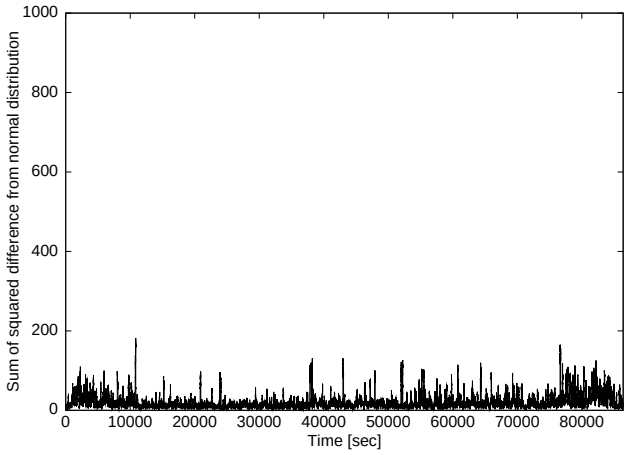

図 8　平均二乗誤差の変動 (正常なフロー)

$$D = \frac{\sum_{i=1}^{n}(G(r_i) - i/n)^2}{n} \tag{3}$$

以上の方法で D を各 SYN レート計算ごとに行い，その時間的変動を示した結果が図 7，8 である．このうち，図 7 がトラヒック全体に対する D であり，図 8 が正常トラヒックを対象とした結果である．これらの結果より，正常トラヒックについては時間による影響をほとんど受けずに平均二乗誤差が安定して穏やかに変動していることが見受けられる．一方，すべてのトラヒックに着目すると，平均二乗誤差が急激に増大している点が複数見受けられる．これらはすべて正常なトラヒックではないと判断されたものであり，その影響によって平均二乗誤差が急激に上昇することが観測できる．以上より，トラヒックレートを正規分布でモデル化し，その平均二乗誤差を観測することによって，比較的容易に正常でないトラヒックを検出することが可能である．

3. 提案手法の性能評価

3.1 対象とする攻撃トラヒック

本稿で対象とする攻撃トラヒックは，サーバをサービス拒否状態にできるトラヒックとする．サービス拒否が行われるのは，backlog-queue に空きがない場合である．待ち行列に待機できる数はオペレーティングシステムにより異なり，代表的な OS での backlog queue のパラメータは表 2 の通りである．ここで，タイムアウトとは half-open 状態を解除するまでのタイムアウト時間であり，これ以上経過した half-open 状態のコネクションは，サーバ側で強制的に解除される．したがって，サーバをサービス拒否状態にするためには，タイムアウト時間内に最大長を超える SYN パケットが到着することが必要である．本稿では，サーバとして Linux を仮定し，180 秒間に 1,024 個を超える half-open 状態が発生した場合に攻撃であると定義する．本稿で取得した 5 日間のトラヒックデータにおいて攻撃であると定義された箇所は全部で 10 箇所である．

表 2　backlog queue の標準設定

OS	最大長	タイムアウト (sec)
Linux	1,024	180
Solaris	1,024	240
Windows 2000 server	200	40

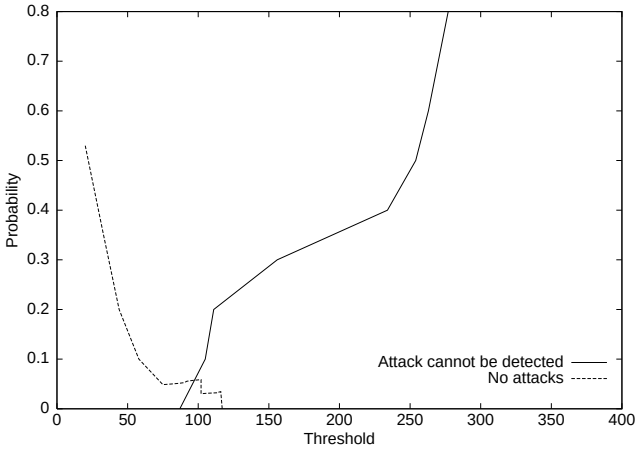

図 9　閾値と攻撃未検出率，誤検出率

3.2 攻撃の検出率

本稿では，実トラヒックデータをもとにしたシミュレーションにより，提案手法の検出性能を評価する．また，評価指標として攻撃未検出率および誤検出率を定義する．これらはそれぞれ以下の式で与えられる．

$$\text{攻撃未検出率} = \frac{\text{攻撃検出ができなかった箇所}}{\text{定義されている攻撃箇所の総数}} \tag{4}$$

$$\text{誤検出率} = \frac{\text{検出されたが実際は攻撃ではなかった箇所}}{\text{攻撃と検出された総数}}$$

はじめに，閾値 B を変化させたときの攻撃未検出率および誤検出率を図 9 に示す．この結果より，閾値が約 90 以下であればすべての攻撃を検出することが可能である．また，このとき 5% ほど誤検出が含まれているが，これはすべて同一ホストから同一サーバに対して大量のコネクション要求 (20 SYNs/sec) が出されている箇所 (2 箇所) であり，サーバの負荷軽減，公平な資源提供の観点からもレート制御などを行っても差し支えない部分であるともいえる．したがって，これらの箇所を攻撃と同様に扱うことができれば，誤検出率も 0 であると考えることができる．

3.3 検出可能な攻撃トラヒックレート

次に，提案手法がどの程度の低い攻撃トラヒックを検出でき

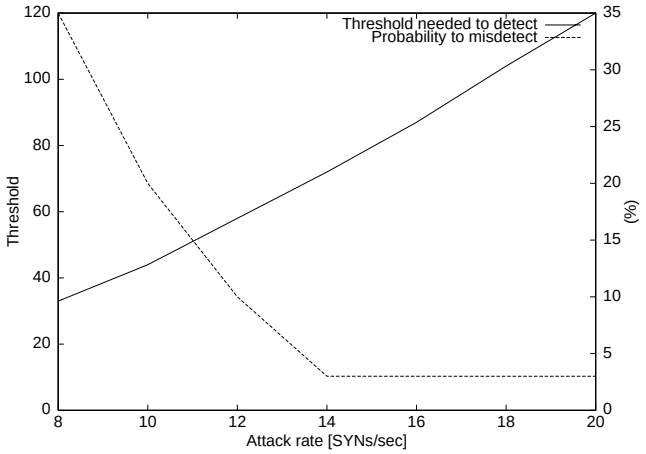

図 10　攻撃のレートと検出するのに必要な閾値，誤検出率

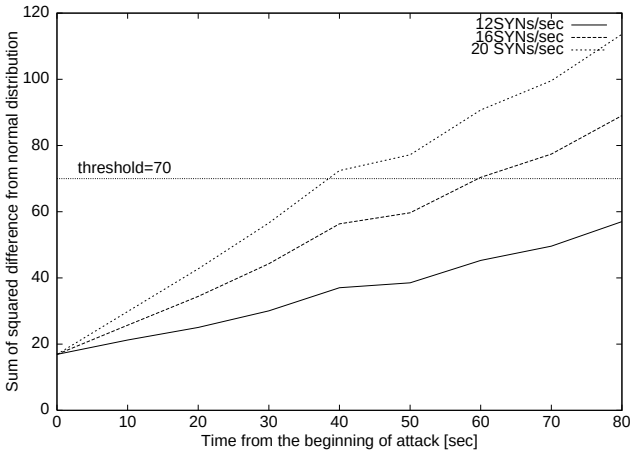

図 11　攻撃開始後の残差平方和の変化

るかについて検討を行う．ただし，大阪大学において観測された攻撃トラヒックはすべてレートが高いものであったため，ここでは擬似的に低いレートの攻撃トラヒックを生成し，正常トラヒックのデータに混在させることによって作成した擬似トラヒックを用いて評価を行う．

図 10 は，挿入した攻撃トラヒックのレートを変化させたときに，検出が可能となる閾値の値，ならびにその時の誤検出率を示したものである．低いレートの攻撃トラヒックを検出するためには，閾値を低い値に設定する必要があるが，その結果正常なトラヒックの変動であっても攻撃であると誤検出する確率が増大する．ただし，今回のシミュレーション結果では 14 SYNs/sec 以上の攻撃トラヒックであれば，誤検出をしない閾値を設定することができることがわかる．また，攻撃レートが上昇しても誤検出がゼロとならないのは，前節と同じく高い SYN レートを観測した箇所が存在するためである．

最後に，攻撃検出までにかかる時間を調べるため，攻撃開始時点からの平均二乗誤差の時間変化を図 11 に示す．ここで，対象とする攻撃トラヒックのレートをそれぞれ 12 SYNs/sec, 16 SYNs/sec, 20 SYNs/sec とする．この図より，攻撃が発生すると平均二乗誤差が徐々に増加していることがわかる．しかしながら，誤検出が発生しない閾値 70 を設定しても，16 SYNs/sec の攻撃を検出するのに必要な時間は 60 秒程度であり，検出までに送られる攻撃トラヒックの SYN パケットの総数は 960 個で Linux サーバの backlog-queue がすべて攻撃で埋まる前に検出できることがわかる．

4. まとめと今後の課題

本稿では，トラヒックの到着レートの統計的な性質を利用し，トラヒックの時間的変動を考慮に入れた新しい攻撃トラヒック検出手法を提案した．シミュレーション評価によって，提案手法がより高速かつ確実に攻撃トラヒックの検出を行うことができることを示した．今後の課題としては，動的な閾値の設定，サンプリング時間の最適化などがあげられる．

文　献

[1] “CNN. Cyber-attacks batter web heavyweights,” available at http://www.cnn.com/2000/TECH/computing/02/09/cyber.attacks.01/.

[2] D. Moore, G.M. Voelker, and S. Savage, “Inferring internet Denial-of-Service activity,” Proceedings of the 2001 USENIX Security Symposium, pp.9–22, August 2001.

[3] T. Darmohray, and R. Oliver, “Hot spares for DoS attacks,” The Magazine of USENIX and SAGE, vol.25, no.4, p.3, July 2000.

[4] J. Mirkovic, D-WARD : DDoS network attack recognition and defence, Ph.D thesis, Computer Science Department, University of California, Los Angels, June 2003.

[5] H. Wang, D. Zhang, and K.G. Shin, “Detecting SYN flooding attacks,” Proceedings of IEEE INFOCOM 2002, vol.3, pp.1530–1539, June 2002.