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.

分散 SYN Flood 攻撃防御のための構築可能なオーバーレイネットワーク

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
E-mail: †{y-ohsita,murata}@ist.osaka-u.ac.jp, ††ata@info.eng.osaka-cu.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 の接続要求を悪用した 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，現在最も多く行われている．SYN F lood 攻撃の既存の対策としては，ファイアウォールで代理応答を行い，攻撃パケットを遮断する手法があるが，ファイアーウォールの資源も有限であるため，大規模でレートの高い攻撃に耐えることは難しい．そこで本稿では，TCP Proxy のオーバーレイネットワークを用いた分散防御システムを提案する．提案手法では，オーバーレイネットワークを用い攻撃情報を伝播を行い，攻撃を受けているサーバ宛のパケットは各エッジで代理応答を行う．代理応答の結果，攻撃でないと確認されたフローについてはオーバーレイネットワークを用いてサーバに伝えることにより保護する．本稿ではシミュレーションにより提案手法が有効であることを確認する．
キーワード　DDoS 攻撃, SYN Flood, オーバーレイネットワーク, TCP Proxy

Deployable Overlay Network for defense against Distributed SYN Flood Attacks

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

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

Abstract　Distributed denial-of-service attacks on public servers have recently become more serious. More are SYN Flood attacks, since the malicious attackers can easily exploit the TCP specification to generate traffic making public servers unavailable. To assure that network services will not be interrupted, we need faster and more accurate defense mechanisms against malicious traffic, especially SYN floods. But single point deffense (ex. firewalls) cannot work at the large distributed attacks. In this paper, we introduce a distributed defense mechanism using overlay netwroks of TCP Proxies. This mechanism detects attacks near the victim servers and alert messages are sent via the overlay networks. And then TCP Proxies classify packets to the victim servers and blocks malicious traffic. The packets classified into legitimate traffic are protected by being sent via overlay networks. We simulate and verify our proposed method can effectively block malicious traffic and protect legitimate traffic.
Key words　Distributed Denial of Service (DDoS), SYN Flood, Overlay Network, TCP Proxy

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分散サービス拒否 (Distributed Denial of Service; DDoS) 攻撃が深刻な問題となっている．

DDoS は，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．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 [1] 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 [2] 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 [3].

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．

Ingress Filtering [4] は，組織のボーダールータにおいて組織外のアドレスを送信元アドレスとして使用しているパケットを不正パケットと見なし遮断する方法である．この方法は，組織が不正なパケットを送出しないための最低限の設定として近年広く利用されている．しかしながら，この手法では組織内アドレスをつけられたパケットを遮断することはできない．このためすべての攻撃トラヒックを遮断できないという問題がある．また，フィルタリングは組織の管理者が自発的に設定するものであり，最終的な判断は組織の管理者の判断に委ねられる．このため，攻撃が発生した場合に被害者あるいは被害者の属するネットワークが当該組織に対して対策を行うことができない．

SYN cache [5] や SYN cookie [6] はサーバにおける対策の一種である．SYN cache では，サーバは各アプリケーションごとに backlog queue を持つ代わりに，すべてのアプリケーションの half open 状態のコネクションを保持するグローバルハッシュテーブルを持つ．その結果，サーバはより多くの half open 状態のコネクションを保持することができ，攻撃の影響を軽減できる．しかしながら，このメカニズムは単に受付可能な half-open 状態の数を増加させるにすぎず，分散攻撃に対する根本的な解決にはなっていない．一方， SYN cookie [6] は backlog queue をサーバ内に持つ代わりに，SYN パケットの送信元アドレス，送信元ポート番号，あて先アドレス，あて先ポート番号，シーケンス番号から生成されたマジックナンバーを SYN/ACK パケットのシーケンス番号としてつける．そして，ACK パケットを受信した際に，その ACK と ACK パケットのヘッダ情報から生成したマジックナンバーとが一致している場合に SYN/ACK に対する ACK パケットであると確認する方法である．これにより，half-open 状態を保持することなく 3way-handshake を完了させることが可能である．しかしながら，この方法は， SYN パケットからシーケンス番号を生成する場合は，マジックナンバー生成がボトルネックとなりサービス拒否状態が発生しうる．また，half-open 状態を保持していないため，SYN/ACK パケットのロスに対して再送を行う手段が存在しないという問題もある．

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 [7]．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，コストが高く，非常に高いレートの攻撃には耐えることができない．

もともと分散 DoS 攻撃は世界中に広く分散された攻撃ホストからの攻撃が大量に集約されて大規模攻撃へと発展するメカニズムであり，攻撃トラヒックの量は分散攻撃ホスト数の増加によって容易に増大させることが可能であり，攻撃トラヒックのスケーラビリティが非常に高い．このため，従来の一点集中型の防御メカニズムはスケーラビリティという点では分散攻撃に対して大きく劣る．本来，分散攻撃に対する防御策は分散的に行われることが防御のスケーラビリティを考える上でも望ましいが，分散防御システムの連携等を考えた場合に解決すべき課題が多く，有効な解決策が見いだせていないのが現状である．これまでに提案された分散型防御システムとして以下が挙げられる．push back [8] は，攻撃を検出すると，上流ルータに対して送信レート制限要求を行う．この送信レート制限の要求をさらに上流ルータへと繰り返すことで，攻撃者の近くで送信レートを抑えることができる．しかし，この方法は，帯域を浪費する UDP Flood 等の攻撃の対策としては有効であるが，サーバ資源を浪費させる SYN Flood 攻撃の対策としては，大きな効果をあげることはできない．また，DefCOM [9] では，ルータ同士がオーバーレイで連携を行う際のフレームワークが提案されている．しかしながら，この連携も帯域を浪費させる攻撃が前提となっており，また，攻撃パケットと通常パケットの区別方法等の具体的な方法は示されていない．

本稿では，特に分散 SYN Flood 攻撃の対策として，オーバーレイネットワークを利用した分散防御システムを提案する．分散防御システムでは，攻撃の検出は比較的容易に行うことができるサーバの近くで行う．検出された攻撃に関する情報はオーバーレイネットワークを通じ，防御システムの各ノードに伝えられる．そして，ネットワークのエッジに設置された各防御ノードにおいて，攻撃パケットの識別を行い，攻撃パケットを遮断すると同時に，通常ユーザのパケットはオーバーレイネットワークを通じて伝送することによって，保護する．

以降，2. で提案する分散オーバーレイ防御メカニズムの概要について述べる．そして，3. でシミュレーションによる提案手法の評価により，提案手法の有効性について示す．最後に 4. でまとめと今後の課題について述べる．

2. 分散オーバーレイ防御メカニズム

従来の防御方法は図 1 に示されるように一箇所で行われるものであった．このように一箇所で攻撃パケットの判別を行うことは，ファイアウォールに負荷がかかるために，大規模な攻撃に対する耐性は乏しい．

そこで，サーバの近くのファイアウォール一点だけではなく，図 2 に示されるように，ネットワークの各入口など分散した箇所で攻撃に対する防御を行い，攻撃パケットと通常パケットの識別を行うことを考える．以降，攻撃パケットの遮断を行うノードを防御ノードと呼ぶ．すでに述べたとおり，防御ノードにおいて到着した SYN パケットが攻撃パケットか通常パケッ

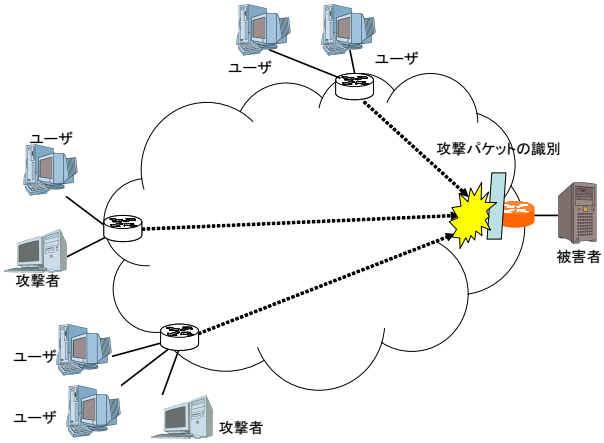


図 1　一点での防御

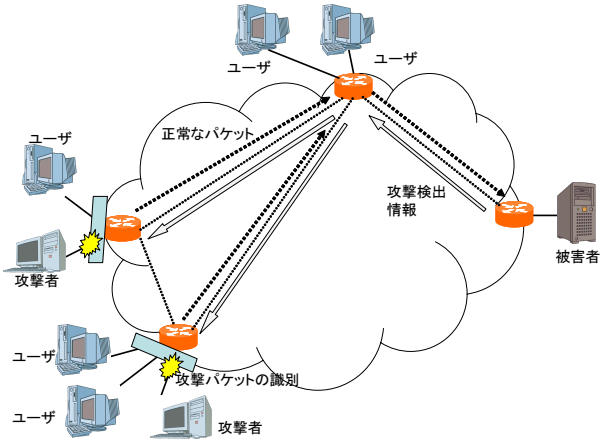


図 2　オーバーレイネットワークを用いた防御

トかを単独で識別することはできない．そこで，本稿では正常なトラヒックの TCP SYN パケットであるかを確認するため，SYN Proxy と同様に防御ノードにおいて SYN パケットの代理応答 (SYN/ACK) を行い，その ACK の受信を確認した後にサーバに対して SYN を送出する方法を用いる．また，攻撃トラヒックから正常トラヒックを保護するために，正常トラヒックは防御ノード同士が結合されたオーバーレイネットワーク上を転送する．本来すべての正常トラヒックをオーバーレイネットワーク上で転送することが望ましいが，防御ノードが高負荷になるとネットワーク全体が正しく動作しない可能性がある．そこで，オーバーレイネットワークで保護する正常トラヒックは攻撃の被害者向けのパケットのみを対象とし，その他のパケットについてはこれまで同様通常の IP ルーティングによる転送を行うことで，防御ノードの負荷を軽減しつつ攻撃トラヒックに対応できるようにする．

以上のメカニズムを提供するために必要な機能は次の通りである．

（1） 防御ノードにおける攻撃の検出
（2） 攻撃情報の通知
（3） 代理応答
（4） 正常トラヒックの伝送
（5） 攻撃終了の検知と通知

以下の各節ではそれぞれの機能の詳細について述べる．

2.1　防御ノードにおける攻撃の検出

大規模な DDoS 攻撃において，攻撃の検出をコアネットワークや攻撃者側で行うことは，攻撃パケットの数が少ないため難しい．そこで，攻撃パケットが多く，比較的検出が容易なサーバ側で攻撃の検出を行うことを考える．

攻撃トラヒックの検出手法として，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 [10]，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 [11]，送信元アドレスの数を用いた方法 [12] などがある．ここでは，以前に我々が提案した，SYN パケットの到着レートの時間変動と正規分布を用いたモデルと比較する方法 [13] を用いる．

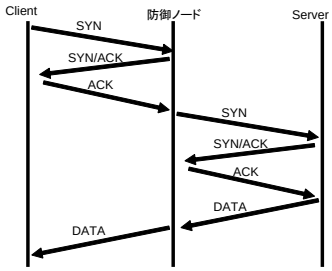


図 3　代 理 応 答

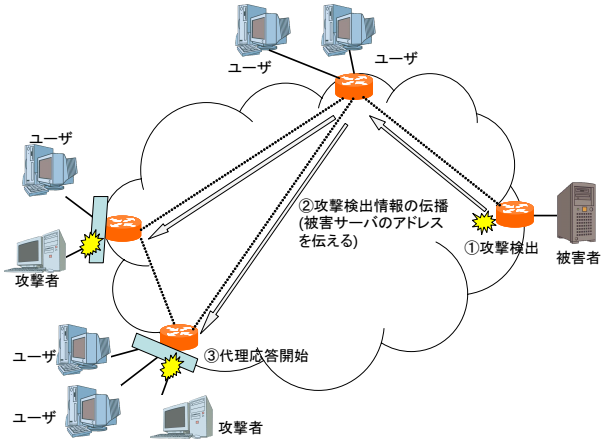


図 4　攻撃検出の手順

2.2　攻撃情報の通知

攻撃検出から代理応答が開始されるまでの手順を図 4 に示す．攻撃が検出されると，攻撃検出情報として，被害サーバの IP アドレスがオーバーレイネットワークを通して各防御ノードに伝えられる．そして，その攻撃検出情報を受け取った防御ノードでは被害サーバ宛のパケットに対する代理応答を開始する．

2.3　代 理 応 答

攻撃情報が伝えられた防御ノードでは，被害サーバ宛の SYN パケットに対する代理応答を開始する．つまり，防御ノードでは SYN パケット到着ごとに宛先アドレスを調べ，それが被害サーバ宛であることを確認すると，そのパケットの送信元アドレスに対して，SYN/ACK パケットを送信する．そして，クライアントから ACK パケットが送信された場合のみ，サーバ

表 1　フロー識別に用いられるデータ構造

ソースアドレス 32 bit			
宛先アドレス 32 bit			
受信側初期シーケンス番号 32 bit			
送信側初期シーケンス番号 32 bit			
ソースポート番号 16 bit		宛先ポート番号 16 bit	
再送タイマ			

とのコネクションを確立する．ここで，防御ノードでは，クライアントから送られた ACK パケットがどのフローに属する ACK パケットであるのか識別する必要がある．そこで，防御ノードでは，各 SYN パケットに対して，表 1 に示されるデータを保持し，ACK パケットが送られた際に，これらのデータを検索することで，どのフローに属するパケットであるのか識別を行う．

しかし，攻撃が行われている際には，大量の SYN パケットが送られるため，表 1 のデータを大量に保持する必要がある．そのため，そのような大量の SYN パケットが送られた場合にも，正常な接続要求は保持しつつ，メモリや CPU 等の資源を浪費することを防ぐ必要がある．

SYN Cache はフロー識別の用いるデータ構造を小さくし，ハッシュを用いて検索を高速化することにより，資源の浪費を防ぎ，サーバの攻撃への耐性を高める対策である．SYN Cache では，ソースアドレス，ソースポート番号，宛先アドレス，宛先ポート番号とランダムに生成されたシークレット番号ををMD5 ハッシュ関数にかけ，ハッシュ値を得る．そして，ハッシュテーブルの該当する箇所を検索する．ハッシュテーブルの該当箇所では，同じハッシュ値のものを保持するリストを持つ．そして，このリストの長さに上限を設け，その上限を超える接続要求があった際には，リスト内で最も古いエントリを削除することにより，新しい接続を受け入れる．

本稿では，防御ノードは 3-way handshake 未完了の接続要求を SYN Cache を用いて保持するものとする．

2.4　正常トラヒックの伝送

3-way handshake が完了したフローに関しては，攻撃ではないと考えられるため，オーバーレイネットワークを通じてサーバとのコネクションを確立し，パケットの転送を行う．ここでは，TCP Proxy [14] で用いられている手法を利用する．

TCP Proxy は，トランスポート層で品質制御を行う手法として提案されている手法である．この手法では，TCP Proxy 同士でオーバーレイネットワークを構築し，宛先に応じて次ホップの TCP Proxy を決め，そして，次ホップの TCP Proxy とコネクションを確立する．このように，クライアントからサーバまで，オーバーレイネットワーク上で 1 ホップずつ TCP コネクションを確立し，そのコネクションを用いて，クライアント・サーバ間のパケットの伝送を行う．

つまり，各正常なコネクションに対して，図 5 で示されるように，防御ノード間でコネクションを確立する．各防御ノードでは，転送元からのコネクションと転送先からのコネクションを結びつけることにより，クライアント・サーバ間の通信の中継を行う．

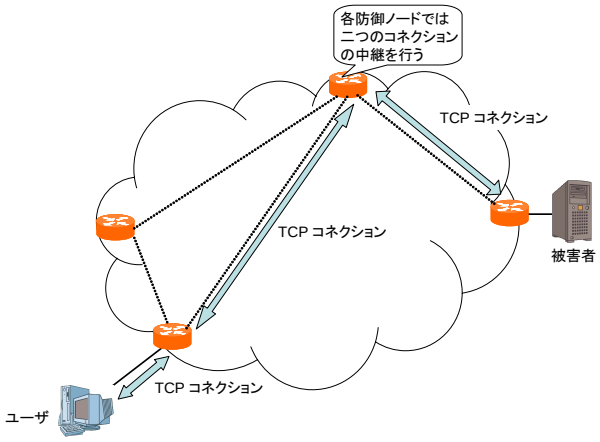


図 5　正常なコネクションの伝播方法

2.5　攻撃終了の検知と通知

分散箇所で代理応答を行うことにより，攻撃パケットを遮断することができると同時に 3-way handshake を完了したフローは保護することができる．しかし攻撃終了後は，SYN パケットに対する代理応答を終了することが防御ノードの負荷を考えると望ましい．そこで，各防御ノードは連携をとることにより，攻撃の終了を検出する必要がある．

各防御ノードでは，SYN パケットに対して代理応答を行っているため，タイムアウトとなった接続要求の数や SYN Cache からあふれた接続要求の数から，3-way handshake を完了していないフローの数を容易に知ることができる．そこで，このフロー数を利用して攻撃の終了を検出するものとする．この方法では，攻撃の可能性のあるパケットそのものの数をもとに攻撃終了を検出しているため，通常パケットの到着レートの変動の影響を受けずに攻撃終了の検出ができる．この方法は，検出までに接続要求がタイムアウトするまでの時間がかかるため，瞬時の検出が必要とされる攻撃検出に用いることはできないが，攻撃終了検出は攻撃検出ほどの迅速さは要求されないため用いることができる．

しかしながら，図 6 のように防御ノード B で攻撃パケットが混在している場合，防御ノード A が対策を終了すると，分散されて行われていた正常なパケットの識別をすべて防御ノード B が行わなければならなくなり，正常なパケットの保護が難しくなる．逆に，防御ノード A で攻撃パケットが混在し，防御ノード B では攻撃パケットが混在していない場合は，攻撃パケットは防御ノード A で遮断されているため，防御ノード B で攻撃対策を終了しても問題はない．

そこで，代理応答の終了は，ある防御ノードにおいて攻撃終了を検出し，かつ，より被害サーバ側の防御ノードで攻撃パケットが混在していない場合に行うとする．

そこで，攻撃終了検出の手順を図 7 に示す．まず，サーバが直接つながっている防御ノードにおいて，3-way handshake を終了していないコネクション数が十分少なくなり攻撃終了を検出する．攻撃終了を検出すると，オーバーレイネットワークを用いて隣接する防御ノードに攻撃終了検出情報を伝播する．

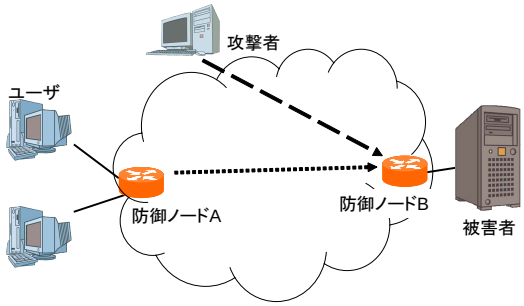


図 6　攻撃終了検出で問題となる例

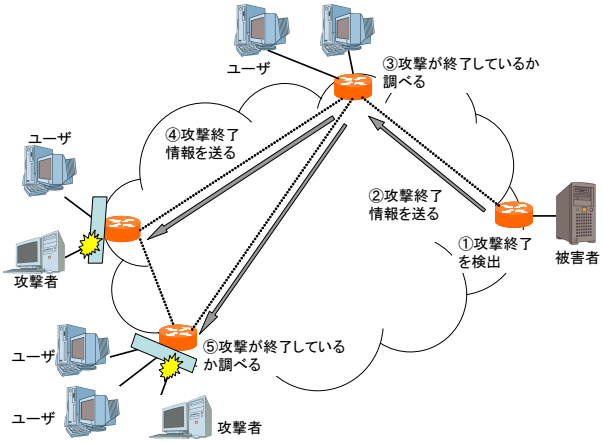


図 7　攻撃終了検出の手順

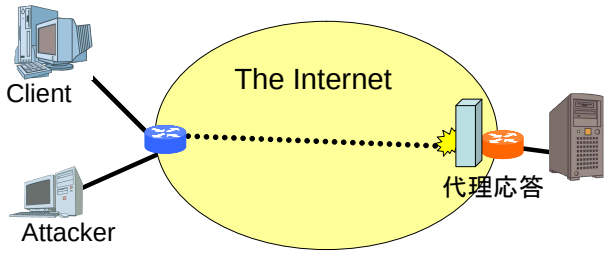
次に，その攻撃終了検出情報を受け取った防御ノードは，オーバーレイネットワークでのルーティング情報を元に，攻撃終了検出情報の送信元が，被害者宛パケットの転送先であるかを調べる．被害者宛パケットの転送先であった場合，つまり，被害者の方向から攻撃終了情報が転送された場合，攻撃検出情報を伝播された防御ノードでも，3-way handshake を終了していないコネクション数を調べる．そして，ここでも攻撃終了を検出されると，さらに隣接する防御ノードに攻撃終了情報を伝播する．これを繰り返すことにより攻撃の終了情報を被害者側から攻撃者・クライアント側に伝えることができる．

3.　提案手法の評価

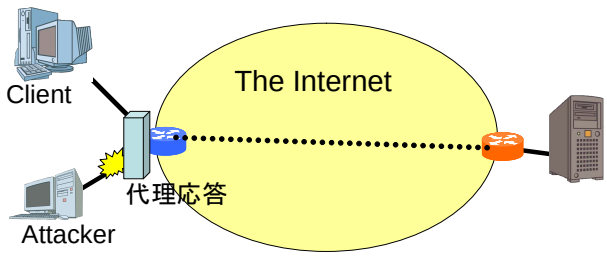
3.1　攻撃レートに対する接続確率

まず，攻撃者の近くで代理応答を行うことの有効性を示すために，分散箇所で攻撃を代理応答した場合 (図 8(b)) と，サーバの近くで代理応答を行った場合 (図 8(a)) のクライアントの接続確率をシミュレーションを行い，調べた．なお，代理応答を行う防御ノードまでの RTT は平均 20 ms のパレート分布に従うとし，また，クライアントからサーバまでの RTT は平均 200 ms のパレート分布に従うとした．攻撃者，正常なユーザ，いずれも同一の防御ノードを経由しているものとする．通常のクライアントからの SYN パケットの到着間隔は正規分布に従うとし，その到着レートは平均 100 SYNs/sec であるとした．代理応答を行う機構の SYN Cache のパラメータは FreeBSD の実装のデフォルト値とした．

図 9 は，対策機構をまったく導入していないサーバ，被害者



(a) サーバの近くで対策



(b) 攻撃者の近くで対策

図 8　シミュレーションの環境

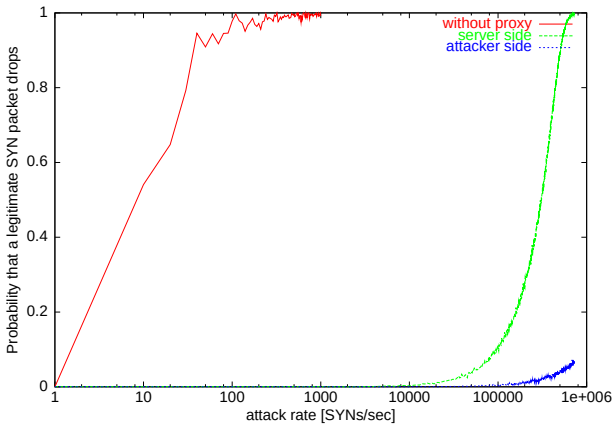


図 9　攻撃レートと接続確率

の近くで代理応答，あるいは攻撃者の近くで代理応答を行った場合の SYN パケットの廃棄率である．同じレートの攻撃であれば，攻撃者の近くで防御した場合， RTT が小さいためより高いレートに耐えることが可能である．[3] によると，60 万パケット毎秒の DDoS 攻撃も観測されている．そのような高いレートの攻撃においては，サーバの近くで代理応答を行った場合，正常な SYN パケットの廃棄率はほぼ 1 となり，ほぼすべての SYN パケットが廃棄されてしまうため，クライアントはサーバとの接続を行うことができないが，攻撃者の近くで代理応答を行った場合であれば，SYN パケットの廃棄率を 0.1 以下と低く抑えることができる．その上，大規模な DDoS 攻撃の場合，攻撃者は分散しているため，防御ノードでの攻撃レートは低くなり，分散箇所での対策はより効果的である．

3.2　攻撃開始後からの接続確率の時間変化

前節と同様に図 8(b) の環境を想定し，通常パケットを一定

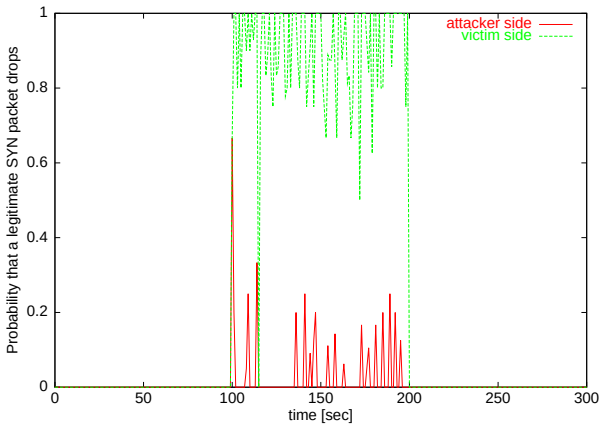


図 10　正常な SYN パケットの廃棄率

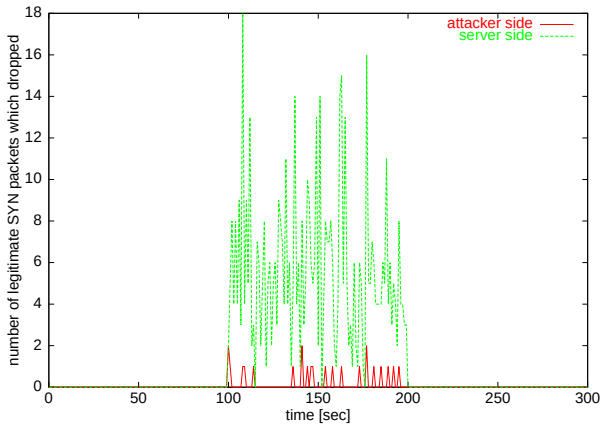


図 11　廃棄された正常な SYN パケットの数

時間送信した後，攻撃パケットを混ぜる．そして，サーバの近くで攻撃を検出し，検出情報を防御ノードに伝播し，防御ノードで代理応答を行うという一連の動作をシミュレートすることにより，提案手法の対策効果を調べる．ここでは，通常 SYN パケット到着のタイミングは，大阪大学の学内ネットワークとインターネットをつなぐゲートウェイで観測されたデータを用い，攻撃レートは一定であるとした．

図 10 は 50 万 SYNs/sec の攻撃を 100 秒から 200 秒の間加えた際に，1 秒ごとに測定した正常な SYN パケットの廃棄率，図 11 は廃棄された正常な SYN パケットの数である．この図より，サーバの近くのみで対策を行った場合，攻撃中の正常な SYN パケットの廃棄率は 0.8 から 1 の間で変動しているが，提案手法では，0 から 0.2 の間で変動している．さらに，廃棄された正常な SYN パケットの数はサーバの近くのみでの対策の場合，毎秒 15 パケット前後であるが，提案手法では毎秒 2 パケット以下である．よって，サーバの近くのみで対策を行った場合よりも大幅に SYN パケットの廃棄率が改善されていることがわかる．ここで，攻撃直後のパケット廃棄率が高くなっている原因は攻撃開始から攻撃検出情報が伝播されるまでの間は防御ノードではまだ代理応答が行われていないためである．しかしながら，代理応答を開始するまでの時間は十分短く，その間に廃棄された SYN パケットが再送された時には防御ノードの代理応答が開始されており，クライアントはコネクションを確立することが可能である．また，ここでもすべての攻撃パケットが同一の防御ノードを経由すると仮定しているが，大規模な DDoS 攻撃においては攻撃ノードは分散しているため，防御ノードでの攻撃レートは低くなり，より一層対策の効果が得られると考えられる．

4. まとめと今後の課題

本稿では，分散 SYN Flood 攻撃に対してオーバーレイネットワークを用いた分散防御手法を提案した．シミュレーションにより，攻撃者の近くで対策を行うことの有効性，提案手法の有効性を示した．今後の課題としては，フロー内のパケットが同一の経路を通ると保証できない場合において代理応答を行うための手法の検討等があげられる．

文　献

[1] “CERT advisory CA-1996-21 TCP SYN flooding and IP spoofing attacks,” available at `http://www.cert.org/advisories/CA-1996-21.html`, September 1996.

[2] “CERT advisory CA-1998-01 smurf IP Denial-of-Service attacks,” available at `http://www.cert.org/advisories/CA-1998-01.html`, January 1998.

[3] 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.

[4] P. Ferguson, and D. Senie, “Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing,” RFC 2267, January 1998.

[5] J. Lemon, “Resisting SYN flooding DoS attacks with a SYN cache,” Proceedings of USENIX BSDCon’2002, pp.89–98, February 2002.

[6] A. Zuquete, “Improving the functionality of SYN cookies,” Proceedings of 6th IFIP Communications and Multimedia Security Conference, pp.57–77, September 2002.

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

[8] S. Floyd, S.M. Bellovin, J. Ioannidis, K. Kompella, R. Manajan, and V. Paxson, “Pushback messages for controlling aggregates in the network,” draft-floyd-pushback-messages-00.txt, internet-draft, work in progress, July 2001.

[9] J. Mirkovic, M. Robinson, P. Reiher, and G. Kuenning, “Alliance formation for DDoS defense,” Proceedings of the New Security Paradigms Workshop, ACM SIGSAC, pp.11–18, August 2003.

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

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

[12] T. Peng, C. Leckie, and K. Ramamohanarao, “Detecting distributed denial of service attacks using source IP address monitoring,” available at `http://www.ee.mu.oz.au/pgrad/taop/research/detection.pdf`, November 2002.

[13] Y. Ohsita, S. Ata, and M. Murata, “Detecting distributed Denial-of-Service attacks by analyzing TCP SYN packets statistically,”Proceedings of IEEE Globecom 2004, November 2004.

[14] I. Maki, G. hasegawa, M. Murata, and T. Murase, “Throughput analysis of TCP proxy mechanism,”to be presented at Australian Telecommunication Networks and Applications Conference (ATNAC) 2004, December 2004.