Computer Security Symposium 2016
11 - 13 October 2016

# 通信環境とSlow Read DoS攻撃効果の関係に関する一考察

田山 俊介[1]　田中 秀磨[1]

**概要**：Slow Read DoS 攻撃は，攻撃者が Web サーバーへ送信したリクエストに対するレスポンスを遅くすることで，Web サーバーの正常なサービスを妨害する攻撃手法である．現状のシステムでは攻撃の検知が困難等の理由から，有効な防御対策は今のところ存在しない．一方で，TCP 高速化のように通信速度を通信環境に最適化させる方法がある．このような方法は強制的に通信速度を向上させるため，Slow Read DoS 攻撃に対する有効な対策となる可能性がある．本論文では，高速化を想定した通信環境と Slow Read DoS 攻撃効果の間における関係を分析した．その結果，通信環境によってサービス不能状態継続時間に変化はないが，その開始時間が変化することが判明した．

**キーワード**：Slow Read DoS 攻撃，TCP 高速化，DoS 攻撃，帯域幅，RTT

# A Study on the relationship between communication environment and effectiveness of Slow Read DoS attack

Shunsuke Tayama[1]　Hidema Tanaka[1]

***Abstract:*** Slow Read DoS attack is the technique that prolongs time to read the response from the Web server. We do not have effective countermeasures to defend from this attack nowadays. On the other hand, there are some methods which optimize communication speed such as TCP acceleration. Such methods will make communication speed high, it may influence the effectiveness of Slow Read DoS attack. In this paper, we analyzed the relationship between accelerated communication and effectiveness of Slow Read DoS attack. As the result, we find that communication environment influence start time of the attack.

***Keywords:*** Slow Read DoS Attack，TCP acceleration，DoS Attack，bandwidth，RTT

## 1. 序　論

サイバー攻撃は大きく分けて標的型攻撃，Web 改ざん，DoS 攻撃の 3 つの形態がある．標的型攻撃及び Web 改ざんはそれぞれの手法は異なるが，目的が攻撃対象の情報の入手にあるのに対し，DoS 攻撃はサービスの妨害が目的であり，情報の窃盗や破壊は目的ではない．このように目的だけ見ると DoS 攻撃の脅威は少ないように感じるが，その脅威は他のサイバー攻撃と遜色ない．2015 年以降，全世界で DoS 攻撃の検知数は 1.2 万件/週を超え，その手法もますます多様化，巧妙化している．攻撃対象になるのは主にオンラインバンク，インターネット上のショッピングサイトであり，1 回の DoS 攻撃により 100 万ドルの被害が生じた報告もあり，その脅威は計りしれない [1].

本研究では、DoS 攻撃の一つである Slow Read DoS 攻撃に着目する．この攻撃は，2012 年 1 月に米国セキュリティ会社の QUALYS に所属する Sergey Shekyan によって考案された攻撃手法である [2]．攻撃者は攻撃対象の Web サーバへ複数のリクエストを送信するとともに，Web サーバからのレスポンスを読み取る速度を故意に遅くすることで，多くのコネクションを接続した状態で維持する．その

[1] 防衛大学校理工学研究科
Graduate School of Science and Engineering, National Defense Academy

結果，Web サーバが有するリソースを使い切ることによって，他の正当なクライアントからの接続ができなくなる．また，攻撃者は見た目上正常なリクエストを送信するため，検知するためにはトランスポートレイヤをモニタリングする必要があり，防御にコストがかかる．

一方で，より大規模な通信を想定した商用ネットワークではこれまでの 1GbE/10GbE ネットワークに代わる 100GbE ネットワークの導入が本格化しつつある．しかし，そういった理論上の通信速度を実現することはネットワーク間の遅延により困難である．遅延の問題を軽減する手段の一つとしてアクセラレータがある．これはネットワーク間を最適化して遅延を最小限に抑えることで，通信速度を向上させる装置である．

先行研究として朴らは，Web サーバの Timeout，最大同時接続数に関して攻撃効果の解析を行った [3]．この解析により，Timeout の短縮または最大同時接続数の拡大によって攻撃に対して一定の防御効果を得られることが判明している．しかし，Timeout の大幅な短縮や最大同時接続数の過度な拡大は，正当な他のクライアントの接続を阻んだりサーバの QoS を大幅に低下させることも同時に指摘している．本研究では通信環境に着目し，Slow Read DoS 攻撃効果と回線の通信環境との関係の解析を目的とした．そのため，アクセラレータによる通信速度の向上を得たと想定した仮想環境を構築し，実際に攻撃ツールを用いて Slow Read DoS 攻撃を行った．その結果をもとに Slow Read DoS 攻撃効果と通信環境の関係について考察する．

## 2. Slow Read DoS 攻撃

Slow Read DoS 攻撃は，攻撃クライアントが Web サーバの最大同時接続数を占有することで正当なクライアントからの接続を妨害する攻撃である．攻撃が行われる際の攻撃者と Web サーバ間のパケットフロー例を図 **1** に示す．攻撃者は TCP の 3way-handshake によってコネクションが確立された後，正当なリクエストを送信し Web サーバからデータを受信する．しかし，攻撃者は ACK パケットを返す際に window size を小さく設定することで，Web サーバから送信されるデータを小さくさせてレスポンスの速度を低下させる．極端に window size を 0 にした場合，Web サーバはデータの送信を止めてコネクションが維持された状態で待機することになる．攻撃者はこのようなコネクションを Web サーバの最大同時接続数まで接続することで，Web サーバは正当なクライアントからの接続を処理できなくなり，サービス不能状態となる．

Slow Read DoS 攻撃は Flood 攻撃のように大容量かつ大量のトラフィックを送信して帯域幅を溢れさせる従来の DoS 攻撃手法と異なるため，新規に対策を講ずる必要がある．しかし，攻撃手法が比較的新しく，研究段階ということから，現在のところセキュリティベンダーによるトラ

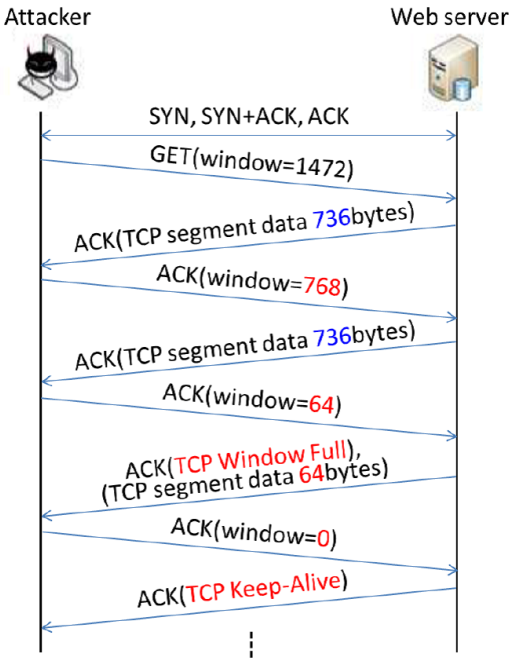


図 **1** Slow Read DoS 攻撃手法
**Fig. 1** Outline of Slow Read DoS attack

ンスポートレイヤのモニタリングが確実な防御手法であるが，これには高いコストがかかる．

文献 [3] では，Slow Read DoS 攻撃の効果は window size が 0 の時，Timeout 値に比例している．また，同報告では複数人で Slow Read DoS 攻撃を実行する Slow Read DDoS 攻撃について提案されている．これはトランスポートレイヤのモニタリング等確実なセキュリティ対策を講じていなければ理論上永遠にサービス不能状態を継続できる．Slow Read DoS 攻撃への対策で一般的に使用されている ModSecurity では文献 [3] の提案手法に対して効果的でないことが示されており，低コストかつ効果的な対策が求められている．

## 3. TCP 高速化

通信回線の帯域幅はその回線における最大のスループットを示している．帯域幅は回線によって決まっており，一般的なネットワークに使われる 100Mbps，1Gbps の回線もあれば，大手検索サイト等の商用ネットワークでは 10Gbps 以上の帯域を持つ回線が導入されている．回線のスループットは 1 秒間に伝送できるデータ量を表し，1Gbps の帯域幅を持つ回線であれば理論上で 1 秒間に最大 1Gbit のデータトラフィックを伝送することができる．しかしながら，実際の通信において通信速度が最大値に達することはない．これにはいくつかの要因が示されている．

まず，クライアント側とサーバ側の回線が異なる場合である．仮にサーバ側の回線のスループットが 10Gbps でクライアント側の回線が 100Mbps の場合，最大スループットは低い方の 100Mbps に制限される．

次に，回線がベストエフォート型の場合，回線の混み具合でスループットが低下することがある．一つの回線に対して，接続者数が一人ならば最高のパフォーマンスで利用できる．しかしながら，複数人が一斉に使用すれば最大値の数%のパフォーマンスしか発揮できない場合もある．個人や小企業が所有するネットワークはベストエフォート型が多いため，同じ回線で同時に使用しているクライアントの数によってスループットは大きく左右される．一方で，帯域保証型の回線のように常時一定の通信速度が保証されている回線もある．しかし，これはベストエフォート型に比べて高価なため，一般に企業等の法人の専用回線として導入されている．

最後に，通信時に発生する RTT（Round Trip Time）によってもスループットは低下する．RTT はネットワークにおける遅延のことであり，サーバとクライアント間のネットワークを往復するのにかかった時間を指す．RTT は，主に通信時にネットワーク上のルータやスイッチでの転送処理にかかる時間やサーバとクライアント間の通信距離で決定される．特に大きい要素を占めるのが通信距離であり，日本国内の通信で約 10ms，日本と米国間で約 100ms の RTT が発生する [4]．

以上のことから帯域幅と同じ通信速度を達成するのが困難な一方，ネットワークに高速化を実装してより理論値に近いスループットを実現させる技術が存在する．ここではそれらの技術をアクセラレータと呼ぶ．アクセラレータには様々な手法が存在するが，本研究では特に RTT を改善するアクセラレータに注目する．RTT は主に通信距離に影響され，3way-handshake によって通信の往復が多いほど RTT も大きくなる．これに対し，クライアントの情報を記録した cookie 等を用いて 3way-handshake を簡略化し，通信の往復回数を減らして RTT を短縮するようなアクセラレータを考える．そのようなアクセラレータに，KLab 社が 2014 年に発表した AccelTCP がある [5]．これは RTT の大きな回線における TCP 通信を高速化するためのプロキシサーバ型のソフトウェアである．クライアント側とサーバ側にそれぞれ AccelTCP のプロキシを立て，RTT の大きな長距離ネットワーク上での通信を代理で行う．このソフトのアクセラレータとしての機能は大きく二点ある．一点目は，コネクションプーリングによる TCP 接続のオーバーヘッド削減である．これにより，コネクションの確立時に発生する 3way-handshake のオーバーヘッドを削減し，通信の RTT を短縮することができる．二点目は，プロキシサーバの設置により通信区間が分割され，各通信区画の RTT が減少することである．これは，パケット消失時の再送時間の短縮につながり，通信全体の高速化を望める．

**表 1**　httpd.conf パラメータ
**Table 1**　httpd.conf parameter.

| | パラメータ | 値 |
|---|---|---|
| Directive | Timeout | 60 |
| | KeepAlive | Off |
| prefork MPM | StartSevers | 8 |
| | MinSpareServers | 5 |
| | MaxSpareServers | 20 |
| | ServerLimit | 300 |
| | MaxClients | 300 |
| | MaxRequestChild | 4000 |

## 4. 実験準備

### 4.1 実験環境

実験環境として，実マシン 1 台の上に仮想環境を構築した．ホスト OS として Windows10 Home，攻撃者及び攻撃対象の Web サーバは VMware Workstation 12.1.0 Player for Windows 上にゲスト OS として CentOS6.7 を構築した．なお，攻撃には slowhttptest-1.6[6]，Web サーバには Apache(httpd-2.2.15) で作成した 100KB の Web ページをそれぞれ用いた．文献 [3] では，ホスト OS に Windows 7，ゲスト OS に VMware（R） Player 6.0.0 上に CentOS 6.5 で実験環境を構築している．本研究における環境とはバージョンの差はあるが，実験に関係する機能に差異はない．また，攻撃ツールの slowhttptest 及び Web サーバの Apache は同バージョンであり，攻撃効果を検証する実験環境は同等といえる．

本実験において，Web サーバの設定である httpd.conf の中で，Directive 及び prefork MPM(Multi-Processsing Module) のパラメータを**表 1** に示す．これらの内，Directive はクライアントとのコネクションを制御し，prefork MPM はプロセスの動作を制御している．また，攻撃で使用する slowhttptest ツールのパラメータを**表 2** に示す．文献 [3] では，表 1 のパラメータの内，Timeout，ServerLimit 及び MaxClients を変化させながら攻撃を行った．本実験においては通信環境に注目したため，それらのパラメータを表 1 に示すように一定の値とした．その他のパラメータは文献 [3] と同値である．なお，特に記載のないパラメータ値についてはデフォルト設定である．

### 4.2 通信環境の設定

本実験では，**図 2** のように，攻撃者と Web サーバ間を一つの仮想スイッチで接続して通信環境を構築した．通信環境について，攻撃者と Web サーバ間の回線はベストエフォート型で接続者は攻撃者のみとする．本実験で用いる回線は，帯域制限をしていない状態で通信速度を計測したところ概ね 1.5Gbps であった．この状態は文献 [3] での通

表 2　slowhttptest パラメータ
**Table 2**　slowhttptest parameter.

| パラメータ | 値 |
|---|---|
| Number of attack connections | 500 |
| Connection rate | 50（connections/sec） |
| Window size | 0 |
| Pipeline factor | 1 |
| Read rate from receive buffer | 5（byte/sec） |
| Timeout for probe connection | 10 |

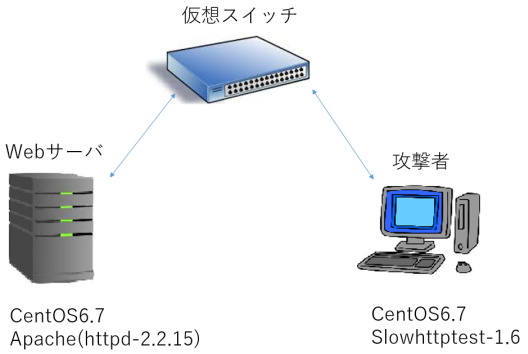


図 2　通信環境
**Fig. 2**　Communication environment

信環境設定と同じである．したがって文献 [3] では，理想的な通信環境における Slow Read DoS 攻撃の効果を検証しているのみで，第 3 節で示したように実際の通信環境とは大きく異なる．

本研究では，帯域制限及び RTT について図 2 における仮想スイッチを用いてシミュレートし，実際の利用環境に近い通信環境で Slow Read DoS 攻撃の効果を解析する．帯域制限は第 3 節で述べたようにベストエフォート型回線を想定し，1.5Gbps の回線の内，一部のみ使用可能という条件を与えた．RTT に関しては利用環境ごとに大きく異なるので，第 3 節で示したように AccelTCP 等が実装されていると仮定する．すなわち，通信距離以外で発生しうる RTT を無視し，安定して RTT の値が保たれるという条件を与えた．なお，帯域制限及び RTT は仮想スイッチと攻撃者間でのみ行う．したがって，Web サーバと仮想スイッチ間は理想状態とし，1.5Gbps の通信速度を達成している．帯域制限及び RTT の具体的なパラメータは第 4.4 節に示す．

### 4.3　用語の定義

Slow Read DoS 攻撃の計算機実験を行う際，攻撃に成功したか否かの判定の基準及び結果の図中に表記されているパラメータについて以下のように定義する．

#### 4.3.1　攻撃成功

攻撃成功とは，Web サーバが他のクライアントに対して正常なサービスを提供できない状態を示す．これを満たす要件は，攻撃者によって接続されたコネクションに対して，Web サーバにより生成された子プロセス（child process）の値がサーバの最大同時接続数（value of MC）と同値以上になることである．Slow Read DoS 攻撃において一度接続されたコネクションはその接続を維持するため，Timeout によって強制的にコネクションが切断されるまで攻撃成功の状態が継続することになる．

#### 4.3.2　攻撃失敗

攻撃失敗とは，Web サーバが他のクライアントに対して正常なサービスを提供できる状態を示す．これを満たす要件は，攻撃者によって Web サーバに接続されたコネクション数がサーバの最大接続数より小さいことである．

#### 4.3.3　図中の表記

攻撃効果の解析を行う上で使用する図中の表記は以下の通りである．

・ established connections

TCP の 3way-handshake によって接続が確立されたコネクションであり，通信が開始されると子プロセスを発生させる．

・ child processes

接続が確立されたコネクションとデータを通信するために Web サーバが生成する子プロセスを指す．この値がサーバに設定された最大同時接続数と同じになる状況が維持されるとサービス不能状態となる．

・ value of MC

httpd.conf 内で設定されたプロセスの最大同時接続数を指し，この値を最大値として子プロセスが生成される

### 4.4　実験手法

本実験では Slow Read DoS 攻撃の効果を実通信環境を考慮して解析を行う．第 4.2 節で述べたように，帯域制限を付加した場合と RTT を想定した場合である．帯域制限は 1Gbps から 50Kbps まで変化させる．予備実験の結果等から Slow Read DoS 攻撃の効果に顕著な特徴が発見できる 1Gbps，10Mbps，100Kbps，50Kbps の場合について示す．これらは以下では実験 1〜4 として示す．これらの結果から，攻撃に用いている回線のスループットが攻撃効果に与える影響について解析できる．次に，RTT については文献 [4], [7] を参考に，10ms，100ms，200ms とした．これらはそれぞれ日本国内，日米間，日欧間で生じる RTT の値である．したがって，国内外での攻撃効果の違いについて解析ができる．これらはそれぞれ実験 5〜7 として示す．以上の各実験のパラメータを**表 3** にまとめる．

## 5.　攻撃効果の解析

前節で示した各実験結果を図 3〜9 に示す．実験結果のグラフについて，青い実線は Web サーバで生成された子プロセス数，赤い実線は Web サーバに接続された TCP の攻撃コネクション数をそれぞれ表す．黒い点線は Web サー

表 3　実験パラメータ
**Table 3**　experiment parameter.

| 実験 | 帯域制限（bps） | RTT（ms） |
|---|---|---|
| 1 | 1G | 0.5 |
| 2 | 10M | 0.5 |
| 3 | 100K | 0.5 |
| 4 | 50K | 0.5 |
| 5 | 1.5G | 10 |
| 6 | 1.5G | 100 |
| 7 | 1.5G | 200 |

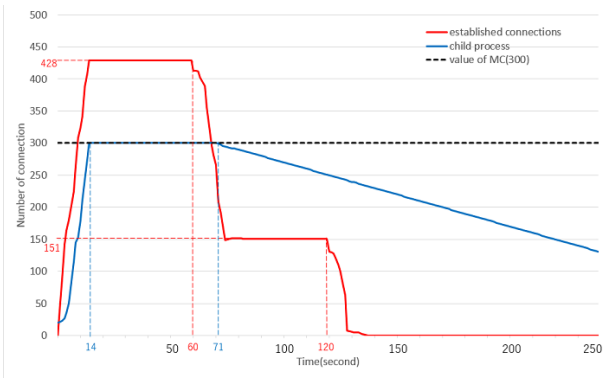


図 3　実験 1（帯域制限 1Gbps）
**Fig. 3**　experiment 1 (bandlimited 1Gbps)

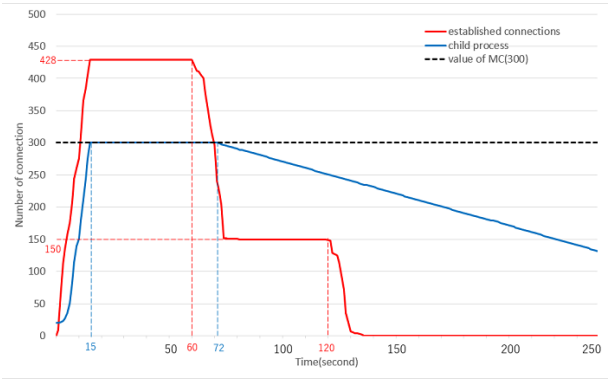


図 4　実験 2（帯域制限 10Mbps）
**Fig. 4**　experiment 2 (bandlimited 10Mbps)

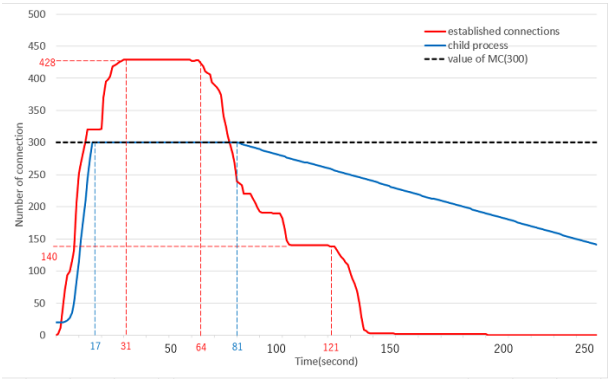


図 5　実験 3（帯域制限 100Kbps）
**Fig. 5**　experiment 3 (bandlimited 100Kbps)

バの MC を示す．したがって，子プロセス数が MC 値に達している期間で攻撃が成功しているとみなす．

### 5.1　帯域制限による影響

実験 1 の結果を**図 3** に示す．図から，1Gbps の帯域制限をかけた時，攻撃開始から 14 秒で子プロセスが MC の値である 300 に到達し，Web サーバはサービス不能状態に陥り攻撃成功となった．その後は Timeout の設定により 71 秒を過ぎると子プロセスの強制切断が始まり，サービス可能状態へ回復し，攻撃失敗となった．よって，実験 1 において攻撃者は 57 秒間の攻撃に成功したといえる．コネクション確立数に注目すると，攻撃を開始してから最大で 428 まで上昇していることが分かる．これは TCP の 3way-handshake により接続を確立済みのコネクション数 128 が含まれているためである．コネクション確立数も子プロセスと同様に Timeout の設定によって，60 秒後から強制的に切断され始め，Web サーバの処理待ちであった確立済みのコネクションを処理できるようになった．この接続も 120 秒から切断され始め，概ね 130 秒後にはすべてのコネクションが切断された．

実験 2 では，10Mbps の帯域制限をかけて計算機実験を行った．その結果を**図 4** に示す．図から，攻撃開始から 15 秒でサービス不能状態となり，71 秒後に強制的切断されるまでの 56 秒の間，攻撃に成功した．実験 2 では実験 1 と同様の傾向を示しており，一定以上の最大スループットを確保した場合，攻撃に影響を与えないことが確認できた．

実験 3 では，100Kbps の帯域制限をかけて計算機実験を行った．その結果を**図 5** に示す．ここでは，攻撃に成功した時間が 17 秒から 81 秒までの 64 秒間だった．実験 1 及び 2 と比較して，生成された子プロセスが切断され始めるのが概ね 9 秒遅く，同時に攻撃成功時間が 8 秒程度長くなった．また，コネクション確立数が最大数に到達するのに 31 秒かかり，実験 1 及び 2 より 15 秒程度遅かった．

実験 4 では，50Kbps の帯域制限をかけて計算機実験を行った．その結果を**図 6** に示す．図から，攻撃に成功した時間は 26 秒から 81 秒までの 55 秒間であり，サービス不能状態になるのが他の実験と比較して 10 秒程度遅かった．また，コネクション確立数が 350 までしか上昇せず，確立できる最大値の 482 まで到達する前に Timeout の設定で切断されていることがわかる．実験 3 及び 4 から，スループットの変化が子プロセス数の生成及びコネクションの確立するのにかかる時間に影響を与えることが確認できた．

### 5.2　RTT による影響

実験 5，6 及び 7 では，RTT が攻撃に与える影響を解析するために，攻撃者と Web サーバの間に 10ms，100ms，200ms の RTT を発生させて攻撃を行った．これにより，スループットの値を決定する要素である RTT の変化が攻

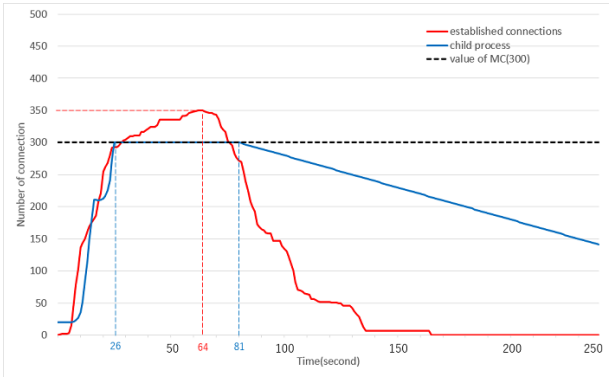


図 6　実験 4（帯域制限 50Kbps）
**Fig. 6**　experiment 4 (bandlimited 50Kbps)

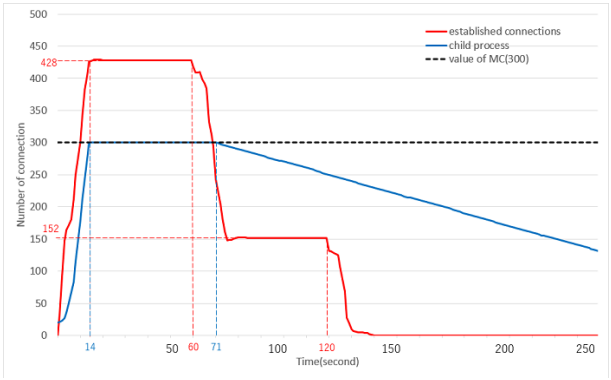


図 7　実験 5（RTT 10ms）
**Fig. 7**　experiment 5 (RTT 10ms)

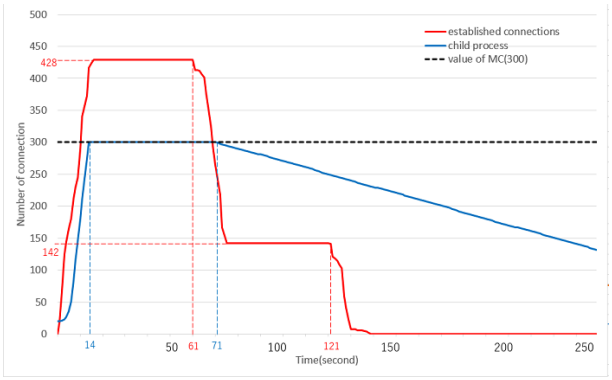


図 8　実験 6（RTT 100ms）
**Fig. 8**　experiment 6 (RTT 100ms)

撃に与える影響を解析できる．

実験 5 では RTT が 10ms の時の計算機実験を行った．その結果を**図 7** に示す．図から，14 秒から 71 秒までの 57 秒間サービス不能状態に陥ったことが確認できる．コネクション確立数についても 14 秒で最大値の 428 に到達し，60 秒後に減少を開始して 130 秒後には全てのコネクションが切断された．

実験 6 では RTT が 100ms の時の計算機実験を行った．その結果を**図 8** に示す．ここでは，実験 5 と同様に 57 秒間のサービス不能状態を維持した．

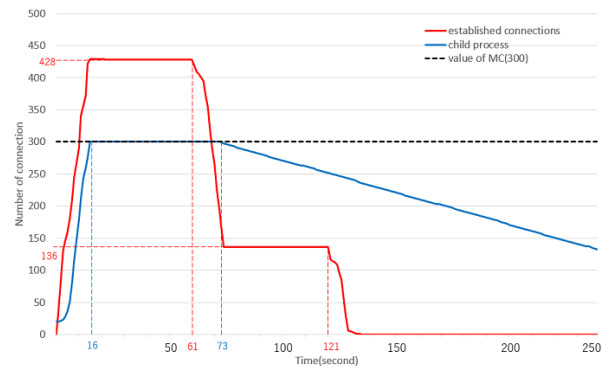


図 9　実験 7（RTT 200ms）
**Fig. 9**　experiment 7 (RTT 200ms)

実験 7 では RTT が 200ms の時の計算機実験を行った．その結果を**図 9** に示す．ここでは，実験 5 と同様に 57 秒間のサービス不能状態を維持した．実験 5，6 及び 7 より，RTT の変化は Slow Read DoS 攻撃に対して特に影響を与えないことが確認できた．

## 6. 考　察

本実験では，仮想環境上で計算機実験を行い，実験 1 から 4 でスループット値の変化による攻撃への影響を解析した．その結果，一定値を下回るスループットの変化は，Web サーバの子プロセスの生成及びコネクションの確立にかかる時間に影響を与えることが確認できた．Web サーバが子プロセスを生成する平均速度は，実験 1 及び 2 で 20.0（processes/sec），実験 3 で 17.6（processes/sec），実験 4 で 11.5（processes/sec）であった．また，TCP の 3way-handshake によってコネクションが確立される平均速度は，実験 1 及び 2 で 28.5（connections/sec），実験 3 で 13.8（connections/sec），実験 4 で 5.6（connections/sec）だった．この結果より，スループットを低下させることで TCP とのセッションを確立する速度及び Web サーバが子プロセスを生成する速度が遅くなることが確認できた．また，本実験において攻撃成功の継続時間は各実験を通して大きな差はなかった．しかしながら，攻撃成功の開始時間が実験 1，2，3 でそれぞれ 14 秒，15 秒，17 秒とほとんど差はないのに対し，実験 4 では 26 秒と大幅に遅れる結果となった．これは，スループットが低下することによってコネクションを確立させる速度及び子プロセスの生成速度が遅くなった影響と考えられる．この結果より，スループットの低下によっては攻撃成功の開始時間が大幅に遅れることが確認できた．さらに，実験 1，2 において established connections 及び child processes は同時に最大値に到達しているのに対し，実験 3 では child processes が最大値に到達した 14 秒後に established connections は最大値に到達している．実験 4 については established connection が最大値に到達しておらず，スループットの低下による 2 つ

の速度の低下度合いに差があると考えられる．本実験で確認できた事項について，スループット値と established connections 及び child processes との関係性の詳細な解析は今後の課題としたい．

一方，実験 5，6 及び 7 では RTT の変化による攻撃への影響を解析したが，3 つの実験を経て攻撃効果に大きな差異は確認できなかった．ネットワークで発生する RTT は日本国内と日本-海外間では大きな差があるが，本実験の結果より，RTT のみの変化は攻撃効果に影響を与えないと考えられる．しかしながら，RTT がスループットの大小に大きく関係することも事実なため，これらの関連の解析も今後の課題としたい．

本実験では，Slow Read DoS 攻撃効果と通信回線の最大スループット値，RTT との関係について考察した．その結果，ネットワークの遅延を解決する種類のアクセラレータでは Slow Read DoS 攻撃に対する効果的な対策にならないが，攻撃者と Web サーバの間で一定以上の最大スループット値を保証する回線を使用しなければ期待通りの攻撃ができないと現状では考えられる．今後も両者の関係についてさらに明らかにすることが課題である．また，Timeout 及び MC の設定によっては子プロセス数が最大数に到達する前に切断が開始する可能性があるため，今後検討していきたい．

TCP のスループットの理論値は以下の計算式で求められる．

$$\text{スループット（bps）} = \text{window size（bit）} / \text{RTT（sec）}$$

Slow Read DoS 攻撃はコネクションを確立させると，最終的に window size を 0 にして通信を停止させるため，RTT の変化は攻撃の可否についてそれほど影響しないと考えられる．TCP の高速化のために実装している他の機能について検討し，Slow Read DoS 攻撃に対する TCP 高速化の有効性の評価を今後の課題としたい．

また，通信環境に応じて window size を最適化させて高速化を実現する技術もある．その一例として，パケット到着時間をリアルタイムで測定した結果に基づき window size を決める方法がある [8]．Slow Read DoS 攻撃は window size を極端に小さくすることで成立する攻撃であるため，そのようなアクセラレータを実装した際の攻撃効果への影響の解析を今後の課題にしたい．

## 7. 結　論

本研究では，Slow Read DoS 攻撃効果と TCP 高速化で向上が見込まれるスループット及び RTT の関係について明らかにするため，仮想環境上で計算機実験を行った．その結果，スループット及び RTT の向上は Slow Read DoS 攻撃を防ぐ対策にならないことがわかった．この結果から，データ伝送間の RTT を解消することでスループットを向上する機能を持つアクセラレータは，回線の通信速度改善には有効だが Slow Read DoS 攻撃の対策には効果はないといえる．また，最大スループット値が低い回線は，攻撃のためのコネクションを確立させる速度や Web サーバの子プロセスの生成速度を著しく低下させるため，攻撃者にとって期待通りの攻撃効果を得られない可能性があるといえる．回線の最大スループット値と攻撃コネクション確立速度及び子プロセスの生成速度との関係は今後の課題である．

本実験では，解析を容易にするためアクセラレータを実装した際に改善が予想されるパラメータのみを変化させながら仮想環境上において計算機実験を行った．しかしながら，実環境においてアクセラレータを実装した場合，RTT 等のパラメータ以外の要素も改善してスループットを向上させていると考えられる．今後は実環境上でのアクセラレータ実装をより想定した計算機実験を課題とする．

### 参考文献


[1] ARBOR Networks : Worldwide Infrastructure Security Report 2011，ARBOR Networks(2011).

[2] Sergey Shekyan : “Are you ready for slow reading?”，2012 年 1 月 5 日，https://blog.qualys.com/securitylabs/2012/01/05/slow-read，2016 年 8 月 2 日参照

[3] 朴　駿漢 : Slow Read DoS 攻撃の解析と対策効果に関する考察，防衛大学校理工学研究科 (2015).

[4] IT-EXchange : “仮想アプライアンス-WAN 高速化”，https://www.it-ex.com/distribution/vappliance/products/category/wan.html，2016 年 8 月 3 日参照

[5] KLab : “ACCELerate TCP proxy”，2014 年 9 月 26 日，https://github.com/KLab/AccelTCP，2016 年 8 月 3 日参照

[6] Sergey Shekyan : “Application Layer DoS attack simulator”，2016 年 7 月 29 日，https://github.com/shekyan/slowhttptest，2016 年 8 月 3 日参照

[7] Think IT : “ネットワーク遅延対策技術”，2011 年 8 月 23 日，https://thinkit.co.jp/story/2011/08/23/2239，2016 年 8 月 3 日参照

[8] Takeshi Isobe，Naoki Tanida，Yuji Oishi and Ken-ichi Yoshida : TCP acceleration technology for cloud computing: Algorithm，performance evaluation in real network，2014 International Conference on Advanced Technologies for Communications (2014)