情報処理学会論文誌 Vol.61 No.3 581–590 (Mar. 2020)

TCPコネクション数と継続時間に基づくSlow HTTP DoS攻撃に対する防御手法

平川 哲也[1,a)] 小倉 加奈代[1,b)] ベッド バハドゥール ビスタ[1,c)] 高田 豊雄[1,d)]

受付日 2019年6月16日, 採録日 2019年11月29日

概要：ネットワークやコンピュータの正常な利用を妨げる Denial of Service（DoS）攻撃の発生件数，規模が年々増加している．DoS 攻撃の一手法である Slow HTTP DoS 攻撃は比較的少ないトラフィックで実行可能であり，トラフィック量の監視といった単純な方法では検出が困難である．そこで，タイムアウトの設定，クライアントごとの同時リクエスト数制限，機械学習による検出といった対策手法が提案されているが，正当ユーザへの悪影響や，多数の攻撃元を用いる分散型 DoS 攻撃への耐性などの問題を残している．本稿では，Slow HTTP DoS 攻撃に対し，クライアントごとの TCP コネクションの数と継続時間に基づき，攻撃コネクションを選択的に切断することによる防御手法を提案した．50 の攻撃者を模擬したシミュレーション実験の結果，正当ユーザのトラフィックに基づき適切なパラメータを設定すれば，正当ユーザが誤って切断される確率は 1%未満と，許容できる程度であることが分かった．

キーワード：DoS 攻撃，Slow DoS，LBR DoS，Slow HTTP DoS，防御手法

A Defence Method against Slow HTTP DoS Attacks Based on the Duration and the Number of TCP Connections

Tetsuya Hirakawa[1,a)] Kanayo Ogura[1,b)] Bhed Bahadur Bista[1,c)] Toyoo Takata[1,d)]

Received: June 16, 2019, Accepted: November 29, 2019

Abstract: Both the number and the volume of Denial of Service (DoS) attacks, that attempt to make a machine or network resource unavailable are getting larger in recent years. A Slow HTTP DoS attack is a method of DoS attacks that attempts to saturate the requests in process of a HTTP server. This attack is known to be effective for a small scale attack. It is difficult to detect it using traditional methods such as monitoring traffic volume. There are some known countermeasures, such as setting timeout, limiting the number of simultaneous requests, and detecting it using machine learning techniques, but they have some drawbacks such as denying services to legitimate users and resistance against attacks from multiple attackers. In this paper, we propose a defense method against Distributed Slow HTTP DoS attack based on the number of connections for each client and the duration of connections. We performed experiments simulating 50 attackers and we found that the probability that legitimate users got erroneously disconnected is less than 1% with appropriate parameters based on the normal traffic.

Keywords: DoS attack, Slow DoS, LBR DoS, Slow HTTP DoS, defence method

1. はじめに

ネットワークやコンピュータの正常な利用を妨げる攻撃を，Denial of Service（DoS）攻撃と呼ぶ [1]．DoS 攻撃のうち，複数のコンピュータから攻撃を実行するものを分散型 DoS 攻撃または Distributed Denial of Service（DDoS）攻撃と呼ぶ．

[1] 岩手県立大学大学院ソフトウェア情報学研究科
Graduate School of Software and Information Science, Iwate Prefectural University, Takizawa, Iwate 020–0693, Japan
[a)] g236q003@s.iwate-pu.ac.jp
[b)] ogura_k@iwate-pu.ac.jp
[c)] bbb@iwate-pu.ac.jp
[d)] takata@iwate-pu.ac.jp

© 2020 Information Processing Society of Japan

DoS 攻撃の脅威は年々増加している．Neuster Inc. の報告 [2] によると，2019 年第 1 四半期の DoS 攻撃発生件数は，2018 年同四半期に対して 200%増加し，最大の攻撃規模（トラフィック量）は同期間で 70%増加した．同報告では，小さな規模の攻撃で大きな効果を得られる DoS 攻撃手法として，Slowloris が紹介されている．

Slowloris は，HTTP サーバを目標とした DoS 攻撃で，Slow HTTP DoS 攻撃と呼ばれる攻撃手法の 1 つである．Slow HTTP DoS 攻撃は，大量のリクエストを長時間維持することで，HTTP サーバの処理中リクエストを飽和させ，サービスを妨害する手法である．Slowloris は，特に HTTP サーバに対してリクエストヘッダを逐次的に送信することで，リクエストを長時間維持する．

Slow HTTP DoS 攻撃は，少ないトラフィックで効果的な攻撃が行えることから，トラフィック量の監視といった単純な手法では検出，防御が困難である．そこで，リクエストが長時間維持されることに着目してタイムアウトを設定する，クライアント IP アドレスごとの同時リクエスト数を制限する，機械学習により攻撃を検出する，などといった防御手法が提案されている．しかしこれらは，通信環境が劣悪な正当ユーザが誤って切断される，複数のコンピュータを用いる分散型 DoS 攻撃に対して効果が低い，攻撃をただちに遮断できない，といった問題をかかえている．前述のように，DoS 攻撃の規模が大きくなりつつあり，Slow HTTP DoS 攻撃の脅威が高まっている現状にあっては，分散型 Slow HTTP DoS 攻撃への効果的な防御手法が求められている．

本稿では，クライアント IP アドレスごとのコネクション数と，コネクションごとの継続時間を基に攻撃クライアントを判別，コネクションを切断することでサーバがサービス不能状態に陥ることを防ぐことで，Slow HTTP DoS 攻撃を防御する手法を提案し，その有効性を評価する．

2. 背景

2.1 DoS 攻撃

DoS 攻撃を実施する動機としては，大きく以下の 3 つがあげられる [3]．

個人的動機 怨恨やサービスに対する不満などを晴らす．

政治的動機 政治的主張を宣伝する．

経済的動機 被害者を脅迫したり，競合相手に損害を与えたりすることで利益を得る．

組織的に実行される DoS 攻撃の多くは，銀行や株取引といった財務に関わるウェブサイトを標的とする．DoS 攻撃の被害者は，機会損失や対策・復旧のために経済的損害を被る．

攻撃の実施に複数のコンピュータを必要とする DDoS 攻撃は，インターネット上のコミュニティで攻撃への参加を呼びかける [4]，ハッキングツールの提供と引き換えに参加を募る [5]，マルウェアによって支配下においたコンピュータに攻撃の指示を与える，といった方法で実施される．

攻撃者の支配下におかれたコンピュータはゾンビと呼ばれる．Command and Control（C&C）サーバにより複数のゾンビを統括したボットネットが構成され，新たなゾンビ獲得や DDoS 攻撃，スパムメールの送信などに利用される [6]．構築済みのボットネットを有料で貸し出すサービス [7] の存在は，ボットネットの構築を収益化・助長すること，DDoS 攻撃の実施が容易になるという問題をもたらしている．

古典的な DoS 攻撃は，標的に向けて大量のトラフィックを発生させることによって実行される．これはフラッディング攻撃と呼ばれ，標的に至るネットワークを輻輳させることにより，正常な通信を妨げるものである．フラッディング攻撃には，大量のトラフィックを発生させるため，多くのコンピュータを用いて DDoS 攻撃として実行する必要があること，トラフィック量の観測により検出・遮断されやすいこと，といった欠点がある．

そこで，少ないコンピュータ，少ないトラフィックで DoS 攻撃を行うための手法が複数知られている．Cambiaso ら [8] は，相対的に少ない帯域幅（Low Bandwidth Rate, LBR）で目的を達成する DoS 攻撃を Slow DoS 攻撃または LBR DoS 攻撃と呼んだ．本稿ではこの攻撃を Slow DoS 攻撃と呼ぶ．以下に Slow DoS 攻撃に属する攻撃の例を示す．

Algorithmic Complexity 攻撃 [9]

サーバ上でハッシュテーブルや正規表現が処理されるとき，高いコスト（CPU 時間，記憶領域など）を要するよう工夫されたデータを送り込む攻撃．

Shrew 攻撃 [10]

TCP パケットが再送されるタイミングに合わせて，瞬間的に大量のトラフィックを送信してネットワークを輻輳させ，再送を妨害し続けることで平均トラフィック量を抑えつつ TCP 通信を妨害する攻撃．

Slow HTTP DoS 攻撃

HTTP サーバの処理中リクエストを飽和させる攻撃．

次節で，本稿で提案する手法が防御の対象とする Slow HTTP DoS 攻撃について述べる．

2.2 Slow HTTP DoS 攻撃

Slow HTTP DoS 攻撃は Slow DoS 攻撃の一種で，HTTP サーバに対し大量のリクエストを処理中の状態に維持することで，処理中リクエストを飽和させる攻撃手法である．処理中リクエスト数がサーバソフトウェアに設定された上限値に達すると，新たなリクエストを受け付けることができなくなり，サービス不能状態に陥る．リクエスト数の上限値の設定は主記憶容量により制約を受ける一方で，クライアントは比較的少ない計算資源とトラフィック量でリクエストを維持することができる．この性質により，フラッ

© 2020 Information Processing Society of Japan

ディング攻撃に比べて少ないコストで効果的な攻撃が可能となる．

リクエストを維持する手段によって，以下の 3 種の Slow HTTP DoS 攻撃が知られている．

Slowloris リクエストヘッダを逐次的に時間を開けて送信する．Slow Header とも呼ばれる．

Slow HTTP POST リクエストボディを逐次的に時間を開けて送信する．Slow Message Body，R.U.D.Y とも呼ばれる．

Slow Read サーバに対し，小さな TCP ウィンドウサイズを通知することにより，レスポンスを長時間かけて送信させる．

Slowloris，Slow HTTP POST は，タイムアウトによる切断を防ぐため，リクエストを逐次的に一定の時間間隔を開けて送信する．この時間間隔はタイムアウトより短くなければならないが，長くするほど少ないトラフィックで攻撃できる．

Slow Read は，TCP が持つ輻輳制御のためのウィンドウ制御の仕組みを利用する．ウィンドウサイズが小さいほどスループットは低下し，少ないトラフィックで攻撃できる．ただし OS に依存する SYN パケットのウィンドウサイズより過度に小さい値を設定すると，攻撃として検出されやすくなる [11]．また，レスポンスの大きさがサーバの送信バッファより小さい場合，バッファに格納された時点でリクエストの処理が完了するため，攻撃は失敗する．よって，十分な大きさのレスポンスを返す URL を攻撃先として指定する必要がある．

2.2.1 影響を受けるサーバ

Tripathi ら [12] は，Apache httpd, Microsoft IIS, Nginx, Lighttpd の 4 つのサーバソフトウェアについて，Slowloris 攻撃，Slow HTTP POST 攻撃への脆弱性を実験により調査した．結果を**表 1** に引用する．

Tripathi らは同稿で，インターネット上のウェブサイト 4 カテゴリ（各 25 サイト）について，Slowloris 攻撃への脆弱性を実験により調査した．これによると，16%（カテゴリ 4 のうち 8 サイト）ないし 56%（カテゴリ 1 のうち 14 サイト）が脆弱であることが明らかとなった．

表 1 各ソフトウェアの Slow HTTP DoS 攻撃への脆弱性（○：脆弱，×：脆弱でない）文献 [12] より引用

Table 1 The vulnerability of popular HTTP servers against Slow HTTP DoS attacks (○: Vulnerable, ×: Not vulnerable). Cited from Ref. [12].

ソフトウェア	バージョン	**Slowloris**	**Slow HTTP POST**
Apache	2.4.18	○	○
IIS	7.5	×	○
Nginx	1.4.6	×	×
Lighttpd	1.4.33	○	○

3. 関連研究

Slow HTTP DoS 攻撃は，正当ユーザと同程度のトラフィック量で，サーバの処理中リクエストを飽和させることができる．よって検出，防御のためには，トラフィック量以外の特徴に着目する必要がある．

Tripathi ら [13] は，Slow HTTP DoS を HTTP/2 において実行する 5 通りの手法を提案した．以下に詳細を示す．

- SETTING_INITIAL_WINDOW_SIZE フィールドが 0 に設定された SETTINGS フレームを送る．Slow Read 攻撃に相当する．
- END_HEADERS がセット，END_STREAM がリセットされた HEADERS フレームを送る．Slow HTTP POST 攻撃に相当する．
- コネクション直後に送るべき Connection Preface のみを送り，以降何も送らない．
- END_HEADERS，END_STREAM ともにリセットされた HEADERS フレームを送る．Slowloris 攻撃に相当する．
- サーバから受け取った SETTINGS フレームに対し Acknowledgement を返さない．

同稿では，カイ二乗検定によってこれらの攻撃を検出する手法があわせて提案されている．しかしこの手法は，特定の期間の通信に攻撃が含まれているか否かは判断できるものの，攻撃と正当な通信を区別し，攻撃を遮断することができない．

Park ら [14] は，Apache の `Timeout` ディレクティブの値と Slow Read 攻撃への耐性を実験により検証した．実験では，`Timeout` を 200 秒，100 秒，10 秒に設定したサーバに Slow Read 攻撃を模したコネクションを 500 個生成し，処理中リクエスト数の推移を観察した．結果，`Timeout` を小さくすることで，処理中リクエスト数が上限に達する，すなわちサービス不能状態に陥る時間が短縮されることが分かった．一方で，小さすぎる値は正当な利用をも妨げてしまうため，一般に 10 秒以上の設定が望ましいとしている．しかし，現実の攻撃においては攻撃者のコネクション総数の制限はないため，再接続により処理中リクエスト数を維持することができる．ただし，再接続を強いることにより攻撃に必要なトラフィックが増えるため，攻撃コストの増加が期待できる．

Cambiaso ら [15] は，Slowloris 攻撃，Slow HTTP POST 攻撃を改良した SlowDroid 攻撃を提案した．同 2 手法を実装した既存ツールがリクエストを 1 行ずつ送信するのに対して，SlowDroid 攻撃では 1 文字ずつ送る．この手法ではより少ないトラフィックで攻撃が可能であり，一般に通信環境が貧弱な携帯端末に適するとしている．実験の結果，送信間隔をサーバのタイムアウトより短く設定したときに，より少ないトラフィック量で Slowloris 攻撃と同等か

© 2020 Information Processing Society of Japan

つ Slow Read 攻撃より大きな攻撃効果が確認された．

Siracusano ら [16] は，特定の TCP コネクションが Slow HTTP DoS 攻撃であるか否かを，ロジスティック回帰，k 近傍法，サポートベクターマシン，決定木，ランダムフォレスト，深層学習の 6 種の機械学習手法を用いて判定する手法を提案した．同手法では，平均のウィンドウサイズ，スループット，合計送信バイト数などコネクションが持つ 14 の特徴量を使用している．しかし，これらすべての特徴量はコネクション終了時に初めて決定するため，攻撃をただちに遮断することができない．

Apache HTTP Server で利用できる拡張機能として，Slow HTTP DoS 攻撃への耐性を高めることができる mod_security [17] および mod_reqtimeout [18] がある．mod_security は Web Application Firewall の 1 つであり，アプリケーション層での攻撃を防ぐための様々な機能を持つ．その中に，Slow HTTP DoS 対策としてクライアント IP アドレスごとに並列で実行できるリクエスト数を制限する機能がある．しかし，複数の IP アドレスから攻撃される DDoS 攻撃に対しては効果が低い [14]．一方，mod_reqtimeout では，リクエストヘッダおよびリクエストボディを受け取る際のタイムアウトと最小レートをそれぞれ設定できる．しかし前述の `Timeout` ディレクティブによる対策と同じく，小さすぎる値を設定すると正当ユーザのリクエストを誤って拒否してしまう可能性がある．

4. 提案手法

本稿では，クライアント IP アドレスごとのコネクション数と，コネクションごとの継続時間に着目して，分散型 Slow HTTP DoS 攻撃を防御する手法を提案する．

提案手法は，サービス不能状態に至らないよう，サーバのコネクション数の上限に達する前に，Slow HTTP DoS 攻撃の特徴である長時間維持される大量のコネクションを選択的に切断することで効果的な防御を実現する．

手法は大きく以下の 3 つのステップからなる．手法中の各パラメータの意味は**表 2** のとおりで，$l \leq u < m$ とする．

(1) 新たなコネクションの確立後，確立済みコネクション数が u より大きければ Slow HTTP DoS 攻撃を受け，同時処理リクエスト数の余裕が少なくなっていると判断し，(2) へ移行する．
(2) 継続時間 t 以上のコネクションを接続元 IP アドレスごとに分類し，最も多くのコネクションを保持している IP アドレスからのコネクションをすべて切断する．ただし同数であれば，コネクション数の多い IP アドレスからのコネクションをすべて切断する．
(3) コネクション数が l 未満であれば処理完了とし (1) へ移行する．そうでなければ切断処理を続行するため (2) へ戻る．

(2) から (3) の手続き，すなわちコネクション数が u を超えてから l を下回るまでの手続きを以下「切断行程」と呼ぶ．

本手法は，確立済みコネクション数を u 以下に維持しようとする．言い換えれば，サーバが同時に処理できるリクエスト数が m ではなく u によって制約されることになる．攻撃を受けていないときにコネクション数が u を超えると，正当なコネクションが切断され，サービスの質が低下する．

攻撃者がこの手法を適用したサーバに対し攻撃を実施する場合，攻撃元 IP アドレスあたりの t 以上継続するコネクションの数を減らすことで，コネクションの切断を防ぐことができる．しかしこの場合，多くの攻撃元から攻撃する，またはコネクションの継続時間を短くする必要があり，少ない攻撃元，少ないトラフィックで攻撃が可能である Slow HTTP DoS 攻撃の利点が損なわれる．

表 2 提案手法で用いるパラメータ
Table 2 The parameters used in the proposed method.

記号	説明
m	サーバの同時処理リクエスト数の上限
u	切断を開始するコネクション数のしきい値
l	処理終了とするコネクション数のしきい値
t	コネクション継続時間のしきい値

4.1 実装方法

本手法は，サーバ上，またはサーバに至るトラフィックの通過するネットワーク上で実行される．ネットワーク上で実行する場合，サーバに TCP RST パケットを送出することでコネクションを切断することができる．

本手法では，1 つの確立済み TCP コネクションを 1 つのリクエストと見なして，同時処理リクエスト数のひっ迫を判断している．HTTP リクエストを追跡するには，アプリケーション層での解析が必要となるが，1 つのコネクションを 1 つのリクエストと見なすことで，トランスポート層で実行できる．これにより単純な実装とすることができ，また Transport Layer Security（TLS）による暗号化がなされたまま処理することもできる．ただし，1 回のコネクションで複数のリクエストを処理する Keep-Alive 機能をサーバにおいて有効にしている場合は，リクエストの処理完了後もコネクションが一定時間保持されるため，正当ユーザのコネクションが誤って切断される確率が高まる．しかし，コネクションが切断された時点でリクエストが正常に完了していればユーザへの悪影響はない．

4.2 誤切断が起こる確率の推定

提案手法は正当ユーザのコネクションを誤って切断し，サービスの質を低下させる可能性がある．提案手法は，1 つ以上のクライアントとのコネクションをいっせいに切断

© 2020 Information Processing Society of Japan

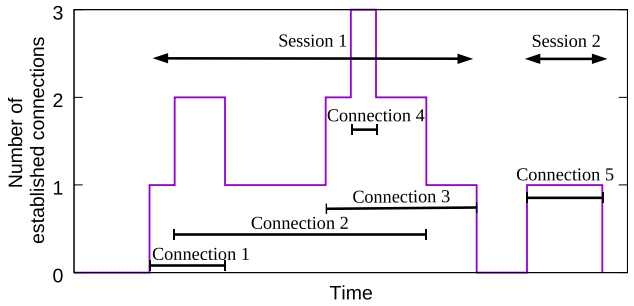

図 1 5 コネクションからなる 2 つのセッションの例
Fig. 1 An example of 2 sessions consist of 5 connections.

表 3 式中で用いる記号
Table 3 The symbols used in the formula.

記号	意味
a	攻撃クライアントの数
r	各攻撃クライアントのコネクションレート（毎秒）
u, l, t	提案手法のしきい値

する．よって，特定のクライアント・サーバの組の間で連続して確立しているコネクションがまとめて切断される．このようなコネクションの集まりを以下，セッションと呼ぶ．提案手法の切断行程において正当ユーザのセッションが誤って切断される事象を以下，誤切断と呼ぶ．

図 1 にセッションの例を示す．縦軸は，あるクライアント・サーバの組の間で確立済みのコネクションの数，横軸は時間を表している．Connection 1，2，3，4 は連続して確立しているため，これらは 1 つのセッションに属する．Connection 5 はこれらとは連続していないため，独立したセッションである．

筆者らは，誤切断が発生する確率を推定する式を提案した [19]．この式を利用して，ある攻撃規模の想定に対し，誤切断率を一定以下に抑えるための提案手法のしきい値を求めることができる．本節では，この式を紹介する．式中で用いる記号を**表 3** に示す．また，誤切断率を推定するにあたり，以下の事柄を仮定する．

仮定 1：コネクションレート r は，各攻撃クライアントで一様である．一部のクライアントだけレートを高くした場合，そのようなクライアントは切断行程開始時に多くのコネクションを保持していることとなり，優先的に切断されることとなる．これは攻撃者にとって不利である．

仮定 2：正当コネクションの数は u，l に比べて十分に少ない．すなわち，正当コネクションの数は考慮せず，u，l を攻撃コネクション群がすべて占めているものとする．正当コネクション数を考慮する場合，u，l からその数を差し引く必要がある．

正当セッションのうち，時間 d 以上継続するコネクションを n 個以上含むセッションの割合を $f(d,n)$（$d \in \mathbb{R}$, $d \geq 0$，$n \in \mathbb{N}$）と書く．$f(d,n)$ は d，n の両方に対して単調減少である．$f(d,n)$ は平時のトラフィックから求めることができる．$f(d,n)$ の求め方は文献 [19] を参照されたい．

攻撃クライアントの数を a，コネクション数の順に並べた攻撃クライアントを $A_1, A_2, \ldots, A_a$，攻撃クライアント A_n のコネクション数を A_n^c と書く（i.e., $A_n^c < A_{n+1}^c$）．仮定 1 より，切断行程における攻撃クライアントどうしの比較では，t の値にかかわらずコネクション数が最も多い攻撃クライアントが切断される．よって，クライアント A_n が最後に切断されてからの経過時間は，コネクション数の順位 n に比例する．r が一定であるから，A_n のコネクション数は A_n が最後に切断されてからの経過時間に比例する．よって，攻撃者のコネクション数はコネクション数の順位に比例する．

切断開始時のコネクション数は $u+1$ である．仮定 2 より，攻撃クライアントのコネクション数の平均は $\frac{u+1}{a}$ であるから，仮定 1 から以下が成り立つ．

$$A_a^c = 2\frac{a}{a+1} \times \frac{u+1}{a} = \frac{u+1}{a+1}$$
$$A_n^c = 2\frac{u+1}{a+1} \times \frac{n}{a} = \frac{2(u+1)}{a(a+1)}n. \quad (1)$$

切断行程の最後に切断される攻撃クライアントを A_m（$1 \leq m \leq a$）とする．切断開始から終了までの間に切断されるコネクション数は，$A_m^c, \ldots, A_a^c$ の合計であるから，

$$\begin{aligned} u - l + 2 &= \sum_{n=m}^{a} \frac{2(u+1)}{a(a+1)}n \\ &= \frac{2(u+1)}{a(a+1)} \times \frac{(a+m)(a-m+1)}{2} \\ &= m(m-1) \times \frac{l-1}{u+1} \times a(a+1) \end{aligned}$$

$a \geq 1$，$m \geq 1$ より，

$$m = \frac{1 + \sqrt{4a(a+1)\frac{l-1}{u+1} + 1}}{2}$$

式 (1) より，

$$A_m^c = \frac{2(u+1)}{a(a+1)} \times \frac{1 + \sqrt{4a(a+1)\frac{l-1}{u+1} + 1}}{2}.$$

攻撃クライアントの秒間コネクションレート r を用いて，各攻撃者が提案手法のしきい値 t 秒の間に確立するコネクション数は $t \times r$ と書ける．よって，攻撃クライアント A_m が保持するコネクションのうち t 秒以上継続しているコネクション数 A_m^l は，以下の式で表される．

$$A_m^l = \max(A_m^c - t \times r, 0).$$

切断が発生したとき，攻撃クライアント A_m よりも t 以上継続するコネクションを多く含む正当セッションは誤切断される．そのような正当セッションの割合，すなわち誤切断率は以下のとおり求められる．

© 2020 Information Processing Society of Japan

$$(t, \max(A_m^c - t \times r, 0) + 1). \quad (2)$$

ここで，

$$A_m^c = \frac{2(u+1)}{a(a+1)} \times \frac{1 + \sqrt{4a(a+1)\frac{l-1}{u+1} + 1}}{2}.$$

$f(d,n)$ は d，n に対して単調減少であるから，提案手法のしきい値 u，l が大きいほど，また a，r が小さいほど，誤切断率は小さくなる．これはサーバの主記憶容量に依存する同時処理リクエスト数を引き上げ，u，l を大きくするほど誤切断率は低下し，攻撃の規模が大きくなり，攻撃クライアント数およびコネクションレートが大きくなるほど，誤切断率が上昇することを意味している．

4.3 しきい値の決定方法

本節では，提案手法中のしきい値 u，l，t の決定方法について述べる．

4.3.1 u，l の決定方法

前述のとおり，u，l はサーバの同時処理リクエスト数の上限を m として，$l \leq u < m$ を満たす必要がある．

$u = m - 1$ とした場合，コネクション数が同時処理リクエスト数の上限に達してから切断を開始することになる．この場合，サーバの性能や攻撃トラフィック量によっては，切断処理中に一時的なサービス不能状態に陥ることがある．ただし，u が小さすぎる場合，正当コネクションを攻撃と誤認して切断する確率が高まる．m と u の適切な差は，サーバの性能，提案手法の実装方法を考慮して実験的に求める必要がある．

u と l の差を大きくした場合，切断行程の実行される間隔が大きくなる．この間隔中に完了する正当コネクションは，提案手法により切断されることはない．ただし，1 回の切断行程でより多くのコネクションを切断するため，継続時間の長い正当コネクションは切断されやすくなる．切断行程の実行される間隔は，以下のとおり求められる．ただし，a を攻撃者の数，r を攻撃クライアントあたりのコネクションレートとする．また，正当コネクションは攻撃コネクションに比べて十分に少ないものとする．

$$\frac{u-l}{ar}.$$

4.3.2 攻撃者のコネクションレートの推定

正当コネクションの数が攻撃コネクションに比べて十分に少ないと仮定すると，攻撃開始時刻からコネクション数がしきい値 u を超えるまでの間の全クライアントの平均レートを，1 攻撃クライアントあたりのコネクションレートと見なすことができる．

サーバのコネクション数が u を超える直前に，しきい値 n ($n \in \mathbb{N}$) に達した時刻を攻撃開始時刻とする．n は，正当クライアントの同時接続数が n より大きい確率が p 以下となる最小の値とする．n の値は，正当コネクションのトラフィックの特徴から事前に決定することができる．HTTP サーバを待ち行列モデルの 1 つである M/M/c/c システムと見なすと，以下の式が成り立つ．ただし λ を正当コネクションの到着率，μ をコネクションあたりのサービス率，m（$m > n$）を同時処理リクエスト数の上限とする．また，正当コネクションの到着間隔はポアソン分布に従い，コネクションの継続時間は指数分布に従うものとする．

$$p = \prod_{i=n+1}^{m} \frac{\lambda}{i\mu} = \frac{m!}{n!} \cdot \left(\frac{\lambda}{\mu}\right)^{m-n}.$$

攻撃者の推定コネクションレート r' は，以下のとおり求められる．ただし，コネクション数が最後に n に達した時刻を T_n，u に達した時刻を T_u（$T_u > T_n$），T_n から T_u の間にコネクションを確立したクライアント群のユニーク IP アドレス数を C（$C > 0$）とする．

$$r' = \frac{u-n}{C(T_u - T_n)}.$$

p の値が大きすぎると，正当コネクションのみによってコネクション数が n に達し，攻撃開始時刻を過早に判断してしまう確率が高まる．一方，p の値が小さすぎると，同時処理リクエスト数の上限 m を大きくする必要が生じ，サーバに求められるリソースが増加する．

4.3.3 t の決定方法

t の適切な値は，攻撃実行中に攻撃者の推定コネクションレート r' を基に求めることができる．

式 (2) より，t が $1/r$ 大きくなるごとに，切断される正当セッションの最大同時接続数は 1 小さくなる．$f(d,n)$ は d，n に対して単調減少であるから，t が $1/r$ 未満であれば t を大きくするほど誤切断率は小さくなる．しかし，t を $1/r$ 以上に大きくすると，誤切断率は大幅に増加する．この現象を回避するため，正当コネクションの存在により r' を過小に推定することを考慮して，r' が実際の攻撃レートの 95%であると想定し，t の値を以下の式で求める．

$$t = \frac{0.95}{r'}$$

4.4 計算量

本手法は，コネクションを IP アドレスごとに分類するなど，比較的多くの計算を必要とする．本節では，提案手法の計算量について述べる．

クライアント数を n，1 クライアントあたりのコネクション数を m とすると，t 以上継続しているコネクションが最も多いクライアントを求めるための計算量は，nm である．1 回の切断行程，すなわちコネクション数が u から l に下がるまでの間に，$\frac{u-l}{m}$ 個のクライアントが切断される．

以上より，1 回の切断行程では以下の計算が必要となる．

$$nm \times \frac{u-l}{m} = (u-l)n$$

© 2020 Information Processing Society of Japan

ある一定の単位時間 T あたりに実行される切断行程の回数は，T あたりに確立されるコネクションの数 r を使って，$\frac{r}{u-l}$ と書ける．よって，T あたりの計算量は，rn である．

以上より，本手法は多項式時間で実行可能であり，計算量の観点からは実用に適する．

5. 評価

前章で述べたとおり，提案手法は正当ユーザのセッションを誤って切断し，サービスの質を低下させることがある．本章では，提案手法がスループット，計算量の観点から実機上で実現可能であること，および誤切断率が実用上許容できる程度であることを確認するため，実機による実験ならびにシミュレーションを実施し，その結果について論ずる．

5.1 実機による実験

筆者らは，提案手法がスループット，計算量の観点から実機上で実現可能であることを確かめるための実験を行った [20]．実験では，1 および 30 の攻撃クライアントから，提案手法を実装したサーバに対し，Slowloris，Slow HTTP POST，Slow Read を実装したツール slowhttptest [21] を使用して攻撃を行った．正当クライアントは模擬していない．サーバの最大同時処理リクエスト数（`MaxRequestWorkers`）は 300，しきい値は $u = l = 250$，$t = 1$ 秒とした．

提案手法を適用していない場合，攻撃クライアントの数，攻撃手法にかかわらず，攻撃開始後ただちにコネクション数が `MaxRequestWorkers` の値に達し，サービス不能状態に陥った．一方，提案手法を適用した場合，Slowloris，Slow Read ではコネクション数が 250 程度に抑えられ，最大同時処理リクエスト数に達することはなかった．

しかしながら，30 の攻撃クライアントからの Slow HTTP POST に対しては，約 10 秒間サービス不能状態に陥った（**図 2**）．これは新たに確立されるコネクションに切断が追いついていないことが原因と考え，$l = 200$ として再度実験を行った．結果，コネクション数は最大でも 300 未満に抑えられ，サービス不能状態に陥ることはなかった．

この結果から，攻撃コネクションが大量に確立されたとき，一時的にコネクションの切断が追いつかずサービス不能状態に陥る可能性があるが，しきい値を適切に設定することで対処できることが分かった．

5.2 シミュレーションによる誤切断率の評価

誤切断率を評価するためには，正当ユーザを模擬する必要がある．正当ユーザのトラフィックデータセットは複数公開されているが，これを実機上で正確に再現することは困難である．そこで，誤切断率の評価は実機ではなくシミュレーションにより実施することとした．

筆者は，Slow HTTP DoS 攻撃，正当コネクション，提案手法を模擬するシミュレータを作成した．正当コネクションをセッションごとに仕分けし，そのうち提案手法によって切断されたセッションの割合を誤切断率とする．シミュレータは決定的で，乱数は使用していない．

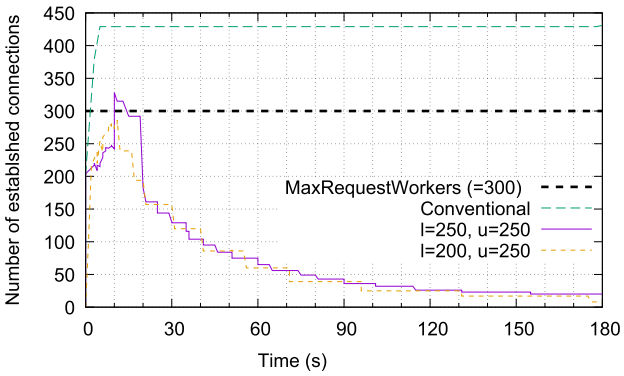

図 2 30 の攻撃クライアントから Slow HTTP POST を実行したときの確立済みコネクション数の推移（文献 [20] の実験結果より作成）

Fig. 2 The transition of the number of established connections against Slow HTTP POST from 30 clients (created from the experimental result in Ref. [20]).

攻撃コネクションは機械的に生成される．シミュレーション開始時から，a 個の攻撃クライアントからそれぞれ一定のレート r で，それぞれ $1/ar$ の間隔を開けて確立するコネクション群を生成する．すなわち，i（$0 \leq i \leq a-1$）番目の攻撃者の j（$j \geq 0$）個目のコネクションの開始時刻 T_{ij} は，以下の式で表される．

$$T_{ij} = \frac{aj + i}{ar}.$$

各攻撃コネクションは，提案手法によって切断されるまで継続するものとする．

Slow HTTP DoS 攻撃を実装したツールである slowhttptest，slowloris では，クライアントごとのコネクション総数の上限を設定できるが，本シミュレーションにおいては上限を設けず，シミュレーション中はつねにコネクションを確立させるものとする．これは，攻撃コネクションが提案手法によって切断されるため，ただちにコネクション総数の上限に達してしまうこと，攻撃者が過去に確立したコネクションについて提案手法が関知しないことから，上限を設けないことが攻撃者にとって不利にならないためである．

正当コネクションは，Wide Project の MAWI Working Group [22] が公開しているデータセットを基に模擬する．詳細を以下に示す．

(1) データセット中にコネクション開始（3 ウェイハンドシェイク），終了（FIN または RST パケット）の両方が含まれていて，かつどちらか一端のポート番号が 80 または 443 である TCP コネクションを抽出する．

(2) 各コネクションを，ポート番号が 80 または 443 である一端の IP アドレス（サーバアドレス），ポート番号

© 2020 Information Processing Society of Japan

表 4 シミュレーションに使用したデータセットの詳細
Table 4 The detail of datasets used in the simulation.

日付	7 月 3 日	8 月 11 日	9 月 25 日
サーバ数	7,682	3,584	8,862
クライアント数	17,897	9,771	19,296
セッション数	42,126	26,107	53,483
コネクション数	68,502	38,580	86,693

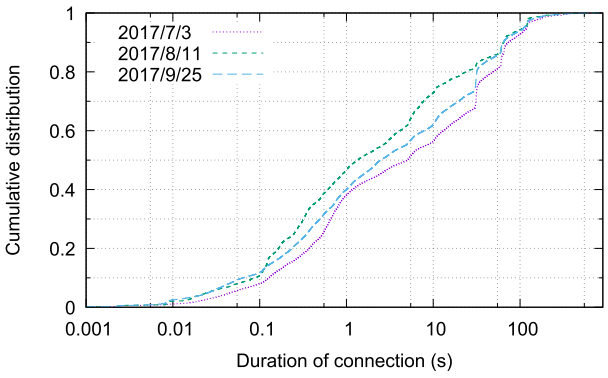

図 3 シミュレーションに使用したデータセットに含まれるコネクションの継続時間の分布
Fig. 3 The distribution of the dulation of connections included in the datasets used in the simulation.

（サーバポート番号）の組ごとに分類し，これらを 1 つのサーバに確立していたコネクション群と見なす．

(3) シミュレーション 1 回ごとに 1 つのサーバに確立していたコネクション群の開始・終了時刻，クライアント IP アドレスを模擬してシミュレーションを実施する．

1 回のシミュレーション時間は，データセットの観測期間に合わせて 900 秒とした．

MAWI Working Group は，テストベッドのネットワーク上のトラフィックを 1 日あたり 15 分間観測，公開している．シミュレーションでは，2017 年 7 月から同年 9 月の各月からそれぞれ 1 日分をランダムに選択して使用した．使用したデータセットの日付と，それに含まれる HTTP，HTTPS のサーバ数，クライアント数，セッション数，コネクション数を**表 4** に示す．また，HTTP，HTTPS コネクションの継続時間の累積度数分布を**図 3** に示す．観測期間が各日 15 分（900 秒）であるため，900 秒を超えるコネクションは含まれない．

5.3 シミュレーションの結果

シミュレーションの結果および式 (2) による誤切断率の推定値を**表 5** に示す．誤切断率は各データセット中に含まれる正当セッションのうち，提案手法によって切断されたものの割合を百分率で示した．データセットごとの正当セッション数（表 4）を重みとして加重平均を求めた．

実験 1 では，$a = 50$，$r = 50$，$t = 100\,\mathrm{ms}$，$u = l = 300, 500, 700$ として，提案手法のしきい値 u，l と誤切断率の関係を調べた．u，l が大きいほど，u，l を飽和させるために必要な攻撃クライアント 1 つあたりのコネクション数が増え，攻撃クライアントと正当クライアントの差異が大きくなり，誤切断率は減少した．ただしサーバにとっては，u，l が大きいほど同時に処理するべきリクエストの数が増え，より多くの計算資源が必要となる．

実験 2 では，$r = 50$，$u = l = 500$，$t = 100\,\mathrm{ms}$ として，攻撃クライアントの数 a と誤切断率の関係を調べた．攻撃クライアントが多いほど，攻撃クライアント 1 つあたりのコネクション数が少なくなり，誤切断率は上昇した．攻撃者にとっては，多くのクライアントを投入すれば誤切断を多く発生させ，サービスを妨害することができるが，攻撃コストは上昇する．

実験 3 では，$a = 50$，$u = l = 500$，$t = 100\,\mathrm{ms}$ として，攻撃クライアント 1 つあたりのコネクションレート r と誤切断率の関係を調べた．コネクションレートが高いほど，しきい値 t 以上継続している攻撃コネクションの割合が下がるため，相対的に正当セッションが切断されやすくなり，誤切断率は上昇した．攻撃者がコネクションレートを引き上げた場合，少ないトラフィックで検出されにくい攻撃ができる，という Slow HTTP DoS 攻撃の利点が損なわれる．

実験 4 では，$a = 50$，$r = 50$，$u = 500$，$t = 100, 500\,\mathrm{ms}$ として，提案手法のしきい値 l を u に対して小さく設定したときの誤切断率を調べた．$t = 100\,\mathrm{ms}$ では，l が小さいほど切断行程中により多くのセッションを切断するため，正当セッションが切断されやすくなり，誤切断率が上昇した．一方，$t = 500\,\mathrm{ms}$ では，50%以上の誤切断が発生し，$t = 100\,\mathrm{ms}$ の場合と対照的に l が小さいほど誤切断率が減少した．これは u と l の差を大きくすることで，切断行程の実行頻度が減り，正当セッションが切断される確率が低くなったためである．しきい値 t を過大に設定すると誤切断率が上昇するという現象に対しては，式 (2) を用いて誤切断率の小さくなる t の値を設定することで事前に防ぐことができる．適切な t の値は 4.2 節で述べたとおり，平時のトラフィックおよび攻撃規模の想定に依存する．

実験 5 では，$a = 50$，$r = 50$，$u = l = 500$ として，提案手法のしきい値 t と誤切断率の関係を調べた．$t = 100, 110, 120\,\mathrm{ms}$ では，シミュレーションでは t が大きいほど誤切断率が上昇する傾向が見られた．一方，推定値は $t = 110, 120\,\mathrm{ms}$ で誤切断率は等しいかわずかに低くなっている．t のわずかな変化は，式 (2) による推定値に必ずしも反映されないことが分かった．対照的に，$t = 500, 510, 520\,\mathrm{ms}$ では，t が大きいほど誤切断率が低下することが分かった．この傾向は推定値にも見られる．

以上の実験を通して，データセットごとの差異に着目すると，8 月 11 日のデータセットは誤切断率が最も低く，7 月 3 日は最も高い傾向が見られる．図 3 を見ると，8 月 11 日は長時間継続するコネクションの割合が少なく，7 月 3 日は多いことが分かる．このことは，長時間継続するコネ

© 2020 Information Processing Society of Japan

表 5 シミュレーションの結果
Table 5 The result of the simulations.

	パラメータ					誤切断率（%）							
						シミュレーション結果				式 (2) による推定値			
	攻撃者		提案手法のしきい値			データセット			加重平均	データセット			加重平均
	a	r	u	l	t(ms)	7/3	8/11	9/25		7/3	8/11	9/25	
実験 1	50	50	300	300	100	1.949	0.877	1.361	1.461	0.989	0.727	0.944	0.913
			500	500		0.598	0.058	0.303	0.352	0.170	0.028	0.112	0.114
			700	700		0.235	0.015	0.163	0.156	0.065	0.012	0.060	0.051
実験 2	10	50	500	500	100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	20					0.112	0.008	0.041	0.058	0.008	0.004	0.006	0.006
	30					0.204	0.008	0.131	0.130	0.041	0.008	0.040	0.033
	40					0.356	0.019	0.208	0.219	0.098	0.016	0.074	0.070
	50					0.598	0.058	0.303	0.352	0.170	0.028	0.112	0.114
実験 3	50	10	500	500	100	0.385	0.027	0.217	0.234	0.108	0.016	0.078	0.075
		20				0.413	0.031	0.226	0.249	0.121	0.016	0.084	0.082
		30				0.487	0.027	0.250	0.284	0.150	0.020	0.098	0.099
		40				0.520	0.042	0.273	0.309	0.170	0.028	0.112	0.114
		50				0.598	0.058	0.303	0.352	0.170	0.028	0.112	0.114
実験 4	50	50	500	300	100	1.403	0.617	0.892	1.010	0.341	0.192	0.313	0.297
				400		0.710	0.084	0.385	0.433	0.253	0.079	0.183	0.185
				500		0.598	0.058	0.303	0.352	0.170	0.028	0.112	0.114
				300	500	70.292	54.081	63.557	63.855	71.413	55.024	64.385	64.810
				400		71.298	54.686	64.222	64.626	71.413	55.024	64.385	64.810
				500		71.934	55.529	64.850	65.303	71.413	55.024	64.385	64.810
実験 5	50	50	500	500	100	0.598	0.058	0.303	0.352	0.170	0.028	0.112	0.114
					110	0.646	0.073	0.325	0.382	0.194	0.043	0.134	0.135
					120	0.674	0.073	0.329	0.394	0.194	0.043	0.132	0.134
					500	71.934	55.529	64.850	65.303	71.413	55.024	64.385	64.810
					510	71.550	55.050	64.544	64.932	71.090	54.596	64.084	64.474
					520	71.184	54.579	64.172	64.541	70.747	54.066	63.738	64.089

クションを多く含むセッションは誤切断される確率が高まることを表している．また，式 (2) による推定値を見ると，シミュレーション結果と比較して誤差が生じているが，各パラメータおよびデータセットと推定誤切断率の関係はシミュレーション結果とほぼ一致している．この結果から，式 (2) は，ある攻撃規模の想定に対して適切な提案手法のしきい値を求めるために有効であることが分かった．

6. まとめ

本稿では，HTTP サーバの利用を妨げる分散型 Slow HTTP DoS 攻撃に対し，クライアント IP アドレスごとのコネクション数と継続時間に着目した防御手法を提案した．実機に実装し，30 の攻撃者からの模擬攻撃実験を行った結果，攻撃コネクションが大量に確立されたとき，一時的にコネクションの切断が追いつかずサービス不能状態に陥る可能性があるが，提案手法のしきい値を適切に設定することで対処できることが分かった．

最大 50 の攻撃者を模擬したシミュレーション実験の結果，提案手法によって正当ユーザのセッションが誤って切断される事象（誤切断）の発生する確率はおおむね 1%未満に抑えられ，誤切断によるサービスの質への悪影響は許容できる程度であることが分かった．しきい値 t を適切に設定しない場合，誤切断率が 50%を超えることがあるが，このような現象は誤切断率を推定する式を用いて予測可能であり，しきい値を適切に設定することで防ぐことができる．

参考文献

[1] Mirkovic, J. and Reiher, P.: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms, *SIGCOMM Comput. Commun. Rev.*, Vol.34, No.2, pp.39–53 (2004).

[2] Neustar, Inc.: Q1, 2019 Cyber Threats & Trends Report (online), available from ⟨https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/neustar-cyber-threats-and-trends-report-q1-2019.pdf⟩ (accessed 2019-06-10).

[3] Aamir, M. and Zaidi, M.A.: A Survey on DDoS Attack and Defense Strategies: From Traditional Schemes to Current Techniques, *Interdisciplinary Information Sciences*, Vol.19, No.2, pp.173–200 (2013).

[4] 中央日報：韓日両国で「三一節サイバー戦争」の動き（オンライン），入手先 ⟨https://japanese.joins.com/article/716/126716.html⟩（参照 2019-05-21）.

[5] WIDE Project: Sledgehammer – Gamification of DDoS attacks (online), available from ⟨https://www.

© 2020 Information Processing Society of Japan

forcepoint.com/sites/default/files/resources/files/datasheet_sledgehammer_the_gamification_of_ddos_attacks_en.pdf⟩ (accessed 2019-05-21).
[6] Zhang, L., Yu, S., Wu, D. and Watters, P.: A Survey on Latest Botnet Attack and Defense, *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp.53–60 (2011).
[7] Symantec Corporation: Renting a Zombie Farm: Botnets and the Hacker Economy (online), available from ⟨https://www.symantec.com/connect/blogs/renting-zombie-farm-botnets-and-hacker-economy⟩ (accessed 2019-05-22).
[8] Cambiaso, E., Papaleo, G., Chiola, G. and Aiello, M.: Slow DoS Attacks: Definition and categorisation, *International Journal of Trust Management in Computing and Communications*, Vol.1, pp.300–319 (2013).
[9] Crosby, S.A. and Wallach, D.S.: Denial of Service via Algorithmic Complexity Attacks, *USENIX Security Symposium*, pp.1–16 (2003).
[10] Kuzmanovic, A. and Knightly, E.W.: Low-rate TCP-targeted Denial of Service Attacks: The Shrew vs. The Mice and Elephants, *Proc. 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, *SIGCOMM '03*, pp.75–86 (2003).
[11] 野岡弘幸, 神谷和憲, 倉上 弘：TCP の挙動に基づく Slow DoS 攻撃の検出（情報通信システムセキュリティ），電子情報通信学会技術研究報告 = IEICE Technical Report：信学技報，Vol.115, No.334, pp.13–18 (2015).
[12] Tripathi, N., Hubballi, N. and Singh, Y.: How Secure are Web Servers? An Empirical Study of Slow HTTP DoS Attacks and Detection, *Availability, Reliability and Security*, pp.454–463 (2016).
[13] Tripathi, N. and Hubballi, N.: Slow Rate Denial of Service Attacks Against HTTP/2 and Detection, *Computers & Security*, Vol.72, pp.1–18 (2017).
[14] Park, J., Iwai, K., Tanaka, H. and Kurokawa, T.: Analysis of Slow Read DoS Attack and Countermeasures on Web servers, *International Journal of Cyber-Security and Digital Forensics*, Vol.4, No.2, pp.339–353 (2015).
[15] Cambiaso, E., Papaleo, G., Chiola, G. and Aiello, M.: Mobile executions of Slow DoS Attacks, *Logic Journal of IGPL*, Vol.24, pp.1–14 (2015).
[16] Siracusano, M., Shiaeles, S. and Ghita, B.: Detection of LDDoS Attacks Based on TCP Connection Parameters, *2018 Global Information Infrastructure and Networking Symposium* (*GIIS*), pp.1–6 (2018).
[17] Trustwave: ModSecurity: Open Source Web Application Firewall, available from ⟨http://modsecurity.org/⟩ (accessed 2019-05-22).
[18] The Apache Software Foundation: mod_reqtimeout – Apache HTTP Server Version 2.4 (online), available from ⟨https://httpd.apache.org/docs/2.4/mod/mod_reqtimeout.html⟩ (accessed 2019-05-22).
[19] Hirakawa, T., Ogura, K., Bahadur Bista, B. and Takata, T.: An Analysis of a Defence Method against Slow HTTP DoS Attack, *2018 International Symposium on Information Theory and Its Applications*, pp.316–320 (2018).
[20] Hirakawa, T., Ogura, K., Bista, B.B. and Takata, T.: A Defense Method against Distributed Slow HTTP DoS Attack, *2016 19th International Conference on Network-Based Information Systems*, pp.152–158 (2016). NBiS-S4-5.
[21] GitHub: shekyan/slowhttptest Wiki (online), available from ⟨https://github.com/shekyan/slowhttptest/wiki⟩ (accessed 2019-06-14).
[22] WIDE Project: MAWI Working Group Traffic Archive (online), available from ⟨http://mawi.wide.ad.jp/mawi/⟩ (accessed 2019-04-12).

平川 哲也 （学生会員）

1993 年生．2016 年岩手県立大学ソフトウェア情報学部ソフトウェア情報学科卒業．2018 年同大学大学院ソフトウェア情報学研究科博士前期課程修了．現在，同大学院博士後期課程在学中．

小倉 加奈代 （正会員）

2006 年北陸先端科学技術大学院大学知識科学研究科博士後期課程修了．同年 4 月より北陸先端科学技術大学院大学知識科学研究科助教．2013 年 7 月より岩手県立大学大学院ソフトウェア情報学研究科講師．コミュニケーションおよびインタラクションデザイン研究に従事．岩手県立大学着任後より，人間の認知・心理特性に着目したセキュリティ技術・防止対策研究にも取り組んでいる．博士（知識科学）．

ベッド バハドゥール ビスタ
（正会員）

1991 年 York 大学電子工学科卒業．1997 年東北大学大学院情報科学研究科博士課程修了．1997～1998 年宮城大学勤務．現在，岩手県立大学大学院ソフトウェア情報学研究科准教授．博士（情報科学）．次世代ネットワーク，モバイル通信に関する研究に従事．電子情報通信学会，IEEE 各会員．

高田 豊雄 （正会員）

1962 年生．1989 年大阪大学大学院基礎工学研究科博士後期課程修了．現在，岩手県立大学大学院ソフトウェア情報学研究科教授．工学博士．セキュリティと誤り制御通信に関する研究に従事．電子情報通信学会，IEEE 各会員．

© 2020 Information Processing Society of Japan