「マルチメディア，分散，協調とモバイル
(DICOMO2018)シンポジウム」 平成30年7月

クライアントのHTTPリクエスト送信動作に着目したボット検知手法の検討

藤 竜成[1] 油田 健太郎[1] 山場 久昭[1] 朴 美娘[2] 岡崎 直宣[1]

概要：近年，人間によって引き起こされるフラッシュイベントと呼ばれる DDoS 攻撃に類似した現象による問題が表面化している．DDoS 攻撃は大量のトラフィックやクライアントの要因がボットであるが，フラッシュイベントはそれらの要因が人間であるため，検知後の対応が異なる．そのため，大量のトラフィックやクライアントが観測された際に，その要因が何であるかを特定しなければならない．本研究では，その要因を特定するため，クライアントの HTTP リクエストの送信動作に着目したボット検知手法を検討する．評価実験では，提案手法の検知精度を評価し，その有用性について議論する．

Investigation of Bot Detection Method Focused on HTTP Request Transmission Behavior of Clients

RYUSEI FUJI[1] KENTARO ABURADA[1] HISAAKI YAMABA[1] MIRANG PARK[2]
NAONOBU OKAZAKI[1]

1. はじめに

DDoS(Distributed Denial-of-Service) 攻撃の検知手法について多くの研究が行われている．DDoS 攻撃は Web サービスに対して大量の不正なトラフィックを送りつけることにより Web サービスのサービス提供を妨害する攻撃であり，深刻な被害を引き起こす．2000 年には EC サイト Amazon やテレビ局 CNN に対する DDoS 攻撃 [1] が，2016 年にはマルウェア「Mirai」による IoT 機器を踏み台にした，最大で 1.5Tbps の DDoS 攻撃 [2] が観測されている．また，DDoS 攻撃の規模や頻度が著しく増加している [3] ことから，DDoS 攻撃の検知システムの重要性が増している．一般に，DDoS 攻撃はボットネットと呼ばれるボットに感染したコンピュータによって構成されるネットワークを通じて実行される．ボットネット管理者は C&C(Command and Control) サーバと呼ばれるサーバを介してボットに攻撃命令を送信することにより，Web サービスに対して DDoS 攻撃を実行する．

しかし近年，DDoS 攻撃検知において，フラッシュイベントについての考慮が重要となってきている．フラッシュイベントとは短時間に大量の人間が，ある特定の行動を目的としてサーバへアクセスする現象である．フラッシュイベントの実例としては，2016 年 2 月に発生したインドの Ringing Bells 社のサーバクラッシュがある．格安スマートフォン「Freedom 251」の予約受付が Web サイト上で開始されたが，その格安スマートフォンを求め何百万の人々が同時に Web サイトにアクセスしたため，サーバのクラッシュを引き起こした [4][5]．日本における，フラッシュイベントと考えられる実例としては，2017 年 11 月に即時買取サービス「メルカリ NOW」のサービス開始直後の大量のアクセスにより引き起こされたサーバのクラッシュがある [6]．これらの実例から，フラッシュイベントは非常に身近な現象であることが分かる．

フラッシュイベントは大量のトラフィックやクライアントが観測される点で DDoS 攻撃と類似しており，フラッシュイベントを DDoS 攻撃と誤検知しやすい．典型的な DDoS 攻撃とフラッシュイベントの概略図を図 1 に示す．典型的な DDoS 攻撃は，大量のトラフィックやクライアントの要因がボットであるのに対し，フラッシュイベントは，大量のトラフィックやクライアントの要因が人間である．

[1] 宮崎大学 工学部
[2] 神奈川工科大学 情報学部

そこで DDoS 攻撃とフラッシュイベントの双方が想定される状況での DDoS 攻撃の検知では，アクセス元がボットであるか人間であるかの識別が重要である．なぜならば，DDoS 攻撃とフラッシュイベントの検知後において，それぞれ別の対応を行う必要があるからである．DDoS 攻撃を引き起こす要因はボットであるため，ボットの Web サービスへのアクセスを禁止し，サーバが過負荷状態となるのを防ぐよう措置を講じる必要がある．それに対して，フラッシュイベントを引き起こす要因は人間であるため，Web サービスへのアクセスを禁止するのではなくサーバのスケールアウトやスケールアップ等を実施し，可用性の低下を防ぐよう措置を講じる必要がある．もし，フラッシュイベントを DDoS 攻撃であると誤検知してしまうと，Web サービスを必要とする人々にサービスを提供できず様々な問題が引き起こされる．例えば [5] において，フラッシュイベントのトラフィックが誤検知により遮断された場合，Web サイト上で予約を行うことが不可能となるため Ringing Bells 社は甚大な損失を被ることが考えられる．また，DDoS 攻撃とフラッシュイベントの識別を行わず，高可用性を優先するとコストの増加に繋がる．なぜならば，本来，サービスの提供が不要な DDoS 攻撃に対してもサービスを提供するからである．

本研究では，ボットと人間の Web サーバへアクセスする際の特徴の違いを利用し，サーバにアクセスしているクライアントがボットであるか否かを判定する手法を提案する．一般に DDoS 攻撃時においてボットは，サーバやネットワーク等のリソースの枯渇を目的として大量のトラフィックを継続的に送りつける．それに対して，フラッシュイベント時において人間は，特定の情報の収集など，ある特定の行動を目的として Web サービスへアクセスするため，多くの場合，大量のトラフィックを送りつけることはない．また，同じボットネットに属するボットは，攻撃を実行するためのプログラムが予めインストールされている．このプログラムにより，ボットネット管理者は C&C サーバを介してボットを一括で操作することが可能となる．ボットネット内では，ボットにインストールされるプログラムは同一であると考えられる [7]．そのため，ボット同士の挙動が類似することが考えられる [8]．そこで，本手法は以下の 3 つの特徴によってボットの検知を行う．

(1) クライアント毎の単位時間当たりのリクエスト量
(2) Web ページを要求するリクエスト送信間隔の類似性
(3) 大量のリクエスト送信の継続性

本手法によりクライアント毎に人間或いはボットかの識別が可能となる．これにより，大量のトラフィックやクライアントが観測された際に，それらの要因が特定できるため，DDoS 攻撃とフラッシュイベントの識別が可能となる．

以下，本論文の構成を述べる．2 章では関連研究を紹介し，問題点を指摘する．3 章では提案手法，提案手法で利用する 3 つの特徴としきい値決定法について説明する．4 章では利用したデータセットと提案手法の検知精度について評価実験を行った結果を示す．5 章ではまとめと今後の課題について述べる．

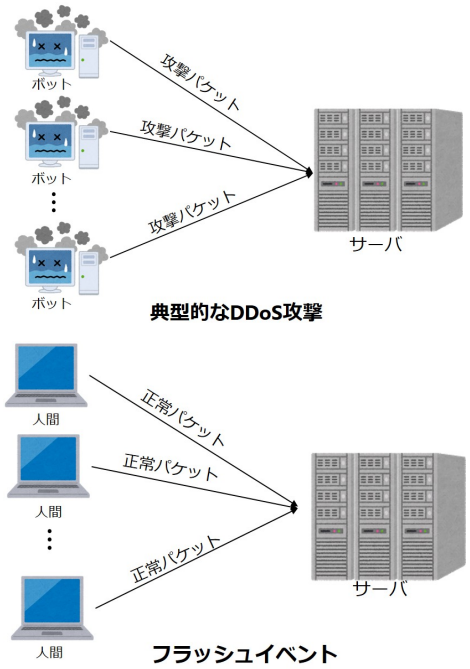

図 1　典型的な DDoS 攻撃とフラッシュイベント

2. 関連研究

本章では関連研究として異常検知に関する研究，DDoS 攻撃とフラッシュイベントの検知に関する研究，flash crowd 攻撃に関する研究を紹介し，それぞれの研究における問題点を指摘する．

2.1 異常検知に関する研究

文献 [9] で小島らは，多次元マハラノビス距離法 (Entropy based Multi dimensional Mahalanobis distance Method: EMMM) によりトラフィックが異常 (攻撃) であるか否かを判定する手法を提案している．パケットヘッダが持つ送信元/宛先 IP アドレス，送信元/宛先ポート番号，パケットバイト数などの 9 つの情報源シンボルを利用してエントロピー系列を求め，そのエントロピー系列を利用してトラフィックの異常検知を行っている．DDoS 攻撃時においては，送信元 IP アドレスのエントロピー値が増加，宛先 IP アドレスのエントロピー値が減少することを利用して攻撃検知が可能であることを示している．しかし提案された手法はフラッシュイベントの発生を考慮していない．フラッシュイベント発生時においては DDoS 攻撃時と同様に，エントロピー値の増減が DDoS 攻撃と一致する．なぜならば，大量のクライアントやトラフィックが観測されるためである．そのため，フラッシュイベントを異常 (攻撃) と見

なし，誤った判断を招く可能性がある．

2.2 DDoS 攻撃とフラッシュイベント検知に関する研究

文献 [10] で Bhatia は，単位時間当たりのパケット数や単位時間当たりの送信元 IP アドレス数などネットワークトラフィックに関する 4 つの特徴と CPU 使用率やメモリ使用率などサーバに関する 4 つの特徴，計 8 つの特徴と Exponentially Weighted Moving Average(EWMA) を利用したこれらの変化を検知する技術を用いて，到達しているトラフィックが DDoS 攻撃或いはフラッシュイベントかを判定する検知システムの提案をしている．提案された検知システムにより，ICMP Flood 攻撃，HTTP Flood 攻撃，SSL Flood 攻撃，そしてフラッシュイベントが検知可能である．しかし提案された検知システムは，到達している全体のトラフィックの判定を行っているためクライアント単位での判定が出来ず，DDoS 攻撃時に人間のアクセスまでも攻撃とみなしてしまう可能性がある．

2.3 flash crowd 攻撃に関する研究

文献 [11] で Saravanan らは，flash crowd 攻撃と呼ばれるフラッシュイベント間に実行される DDoS 攻撃に対応するため，クライアント毎にボットまたは人間の判定が可能な検知システムを提案している．DDoS 攻撃時におけるボットの行動とフラッシュイベント時における人間の行動の差異が，フローの類似性，アクセスしたページ，クライアントの正当性などに表れることを利用して，クライアントがボットか否かを判定している．この論文で提案された手法はクライアント毎に人間かボットかを識別することが可能であるが，DDoS 攻撃が発生した際に，フローの類似度の計算量が爆発的に増加する問題やフローの類似性によるボットの検知をかい潜るような攻撃が行われた際に，ボットを正しく検知することが出来ない問題がある．さらに，パラメータの値やクライアントの分類方法など，提案された検知システムにおいて不明確である点が存在する．

3. 提案手法

本手法では，ボットと人間の Web サーバへアクセスする際の特徴の違いを利用し，サーバにアクセスしているクライアントがボットであるか否かを判定する．本手法で利用する特徴については 3.1 で説明し，しきい値については 3.2 で説明する．また本手法では，サーバに到達するトラフィックを単位時間 U_T 毎に解析し，クライアントがボットであるか否かを判定する．提案手法のボット検知処理の流れを図 2 に示す．

3.1 ボット検知に利用する特徴

ボット検知に利用する，クライント毎の単位時間当たりのリクエスト量，Web ページを要求するリクエスト送信間隔の類似性，大量のリクエスト送信の継続性の 3 つの特徴について説明する．

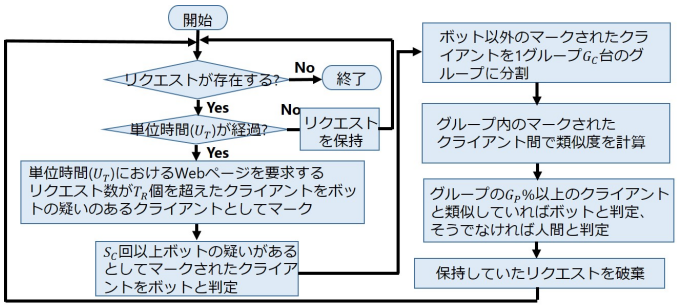

図 2 提案手法のボット検知処理の流れ

3.1.1 クライント毎の単位時間当たりのリクエスト量

一般に DDoS 攻撃時においてボットは，サーバやネットワーク等のリソースの枯渇を目的として大量のリクエストをサーバへ送信する．それに対して，フラッシュイベント時において人間は，特定の情報の収集など，ある特定の行動を目的として Web サービスへアクセスするため，多くの場合，大量のリクエストを送りつけることはない．すなわち，短時間に大量のリクエストをサーバへ送信するクライアントはボットの疑いがある．そこで，あるクライアントが単位時間 U_T 内に C_{RN} 回 Web ページを要求するリクエストを送信したとすると，式 (1) を満たす時，このクライアントをボットの疑いのあるクライアントとしてマークする．ここで，T_R はクライント毎の単位時間当たりのリクエスト量に関するしきい値と定義する．しきい値 T_R の決定法については，3.2.1 で説明する．

$$C_{RN} \geq T_R \tag{1}$$

3.1.2 Web ページを要求するリクエスト送信間隔の類似性

一般に DDoS 攻撃はボットネットを通じて実行され，同じボットネットに属するボットは攻撃を実行するためのプログラムが予めインストールされている．このプログラムはボットネット内では同一であるため，ボット同士の挙動が類似することが考えられる．そこで，マークされたボットの疑いのある全てのクライアントにおいて，Web ページを要求するリクエストの送信間隔の確率分布を求め，ボットの疑いのあるクライアント間で確率分布の類似度を計算する．本研究では，クライアント同士が類似しているか否かを，計算した確率分布間の類似度を用いて判断する．

本論文では，Web ページのファイルの拡張子を htm ファイルと html ファイルとし，それらのファイルを要求するリクエストの送信間隔を用いる．これは htm ファイルや html ファイルが，一般に Web ページを表現するためのファイルとして用いられており，Web ページを要求するリクエストはクライアントが意図しないと送信されないた

めである．また，CSS ファイルや JS ファイルなどの Web ページ以外のリソースを要求するリクエストを送信間隔の計算に含めた場合，リクエスト送信間隔の類似性を適切に表現できないと考えたため，本論文では Web ページ以外のリソースを要求するリクエストは類似度の計算には使用しないことにした．リクエスト送信間隔の単位は秒で，秒以下の値は切り捨てる．

例えば，あるクライアントの Web ページを要求するリクエストの送信間隔として $x_1, x_2, ..., x_n$ が観測されたとすると，式 (2) により Web ページを要求するリクエスト送信間隔の確率分布 $p(x)$ を求める．

$$p(x) = \frac{cnt(x)}{n} \tag{2}$$

ここで $cnt(x)$ は送信間隔 x が観測された回数である．

ボットの疑いのある全てのクライアントにおいて，Web ページを要求するリクエスト送信間隔の確率分布を求めた後，確率分布間の類似度の計算を行う．確率分布間の類似度の計算には Hellinger 距離を用いる．Hellinger 距離は式 (3) で定義される．

$$D_H(p(x), q(x)) = \frac{\sqrt{\sum_x (\sqrt{p(x)} - \sqrt{q(x)})^2}}{\sqrt{2}} \tag{3}$$

ここで $D_H(p(x), q(x))$ が取りうる値の範囲は式 (4) の通りである．

$$0 \leq D_H(p(x), q(x)) \leq 1 \tag{4}$$

確率分布 $p(x)$,$q(x)$ が非常に近い確率分布である時，$D_H(p(x), q(x))$ は 0 に近づき，確率分布 $p(x)$,$q(x)$ が全く異なる確率分布である時，$D_H(p(x), q(x))$ は 1 に近づく．本研究では，式 (5) を満たした際にこのクライアント同士は類似していると判定する．ここで，T_H は Web ページを要求するリクエスト送信間隔の確率分布間の類似度に関するしきい値と定義する．しきい値 T_H の決定法については，3.2.2 で説明する．Hellinger 距離を用いたのは，文献 [11] において，他の確率分布間の類似度を測る指標を用いた場合よりも検知精度が高かったからである．

$$D_H(p(x), q(x)) \leq T_H \tag{5}$$

Saravanan らの手法 [11] では，DDoS 攻撃が発生した際に，ボットの疑いがあるとしてマークされたクライアント間の全ての組み合わせでフローの類似度の計算を行ってしまう．そのため，DDoS 攻撃を実行するボットの台数が増加すると爆発的に計算量が増加するという問題が存在する．計算量が増加した結果，ボットのアクセスの遮断が遅延し，サーバやネットワークが甚大な被害を被ることが予測される．そこで本手法では，マークされたボットの疑いのあるクライアント群を 1 グループ当たり G_C 台に分割し，そのグループ内の全ての組み合わせで，確率分布間の類似度の計算を実行するようにした (図 3)．マークされたボットの疑いのあるクライアント群を複数のグループに分割することにより，計算量の爆発的な増加を防止することが可能となる．

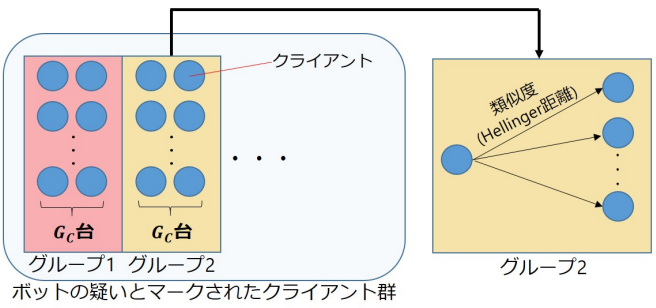

図 3 グループ内での類似度の計算

グループ内のクライアント台数 G_C のうち、G_P% 以上のクライアントと類似していると判定されれば，多くのクライアントと類似していると判断する．すなわち，このクライアントをボットと判定し，以降のアクセスを禁止する．

3.1.3 大量のリクエスト送信の継続性

基本的なボット検知は，クライアント毎の単位時間当たりのリクエスト量と Web ページを要求するリクエスト送信間隔の類似性を用いて行う．しかし攻撃者が Web ページを要求するリクエストの送信間隔をランダムにするなど，Web ページを要求するリクエスト送信間隔の類似性によるボット検知をかい潜るような攻撃を意図的に行った場合，クライント毎の単位時間当たりのリクエスト量と Web ページを要求するリクエスト送信間隔の類似性だけではボットを適切に検知することが出来ないという問題がある．また，Saravanan らの手法 [11] でも同様に，攻撃者がボットのフローを意図的に変更するなど，フローの類似性によるボットの検知をかい潜るような攻撃が行われた際に，ボットを適切に検知することが出来ないという問題がある．

そこで本手法では，大量のリクエストを継続して送信するクライアントをボットと判定することにした．例えば，あるクライアントが C_{SC} 回ボットの疑いのあるクライアントとしてマークされたとすると，式 (6) を満たしたとき，そのクライアントをボットと判定し，以降のアクセスを禁止する．これにより，攻撃者が Web ページを要求するリクエスト送信間隔の類似性による検知をかい潜るような攻撃を意図的に行った場合においても，適切にボットを検知することが可能となる．

$$S_C \leq C_{SC} \tag{6}$$

3.2 しきい値 T_R，T_H の決定法

クライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R と Web ページを要求するリクエスト送信間隔

の確率分布間の類似度に関するしきい値 T_H の決定方法について説明する．

3.2.1 しきい値 T_R の決定法

クライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R は，通常時のサーバのログを用いて決定する．通常時のサーバのログとは，DDoS 攻撃やフラッシュイベントが観測されていない期間におけるサーバのログと定義する．最初に各単位時間毎にユーザ 1 人あたりの Web ページを要求するリクエスト数の平均を求め，それらの平均を μ_R とする．同様に各単位時間毎にユーザ 1 人あたりの Web ページを要求するリクエスト数の標準偏差を求め，それらの平均を σ_R とする．次に式 (7) によりクライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R を決定する．

$$T_R = \mu_R + \alpha_R * \sigma_R \tag{7}$$

ここで，α_R が取り得る値の範囲は $0 \leq \alpha_R$ であり，α_R を大きくするにしたがい，クライアントがボットの疑いのあるクライアントとしてマークされにくくなる．3.1.1 において説明したように，本研究では，単位時間 U_T 当たりの Web ページを要求するリクエスト量が，しきい値 T_R 以上のクライアントは，ボットの疑いのあるクライアントとしてマークされる．

3.2.2 しきい値 T_H の決定法

Web ページを要求するリクエスト送信間隔の確率分布間の類似度に関するしきい値 T_H は，DDoS 攻撃時のトラフィックを用いて式 (8) により決定する．

$$T_H = \mu_H + \alpha_H * \sigma_H \tag{8}$$

ここで μ_H は DDoS 攻撃時のボット間における，Web ページを要求するリクエスト送信間隔の確率分布の類似度の平均を表し，σ_H は DDoS 攻撃時のボット間における，Web ページを要求するリクエスト送信間隔の確率分布の類似度の標準偏差を表す．ここで，α_H が取り得る値の範囲は $0 \leq \alpha_H$ であり，α_H を大きくするにしたがい，クライアント同士が類似していると判定され易くなる．3.1.2 において説明したように，本研究では，クライアント間の Web ページを要求するリクエスト送信間隔の確率分布の類似度が，しきい値 T_H 以下である時，このクライアント同士は類似していると判定する．α_R と α_H を式 (7) と式 (8) にそれぞれ設けたのは，ネットワークやサーバ等の状態を考慮してしきい値 T_R と T_H を柔軟に決定出来るようにするためである．

4. 評価

提案手法の検知精度を実験によって評価し，本手法の有用性について議論する．最初に評価実験に用いるデータセットについて説明し，次に評価実験時における各パラメータの値について説明する．その後，提案手法の検知精度を示す．

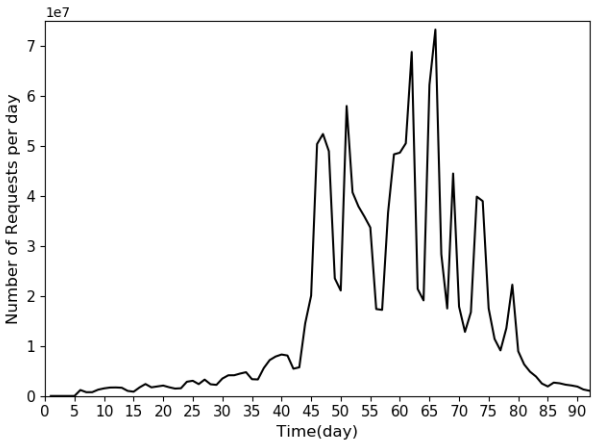

図 4 1998 FIFA World Cup のリクエスト量の推移

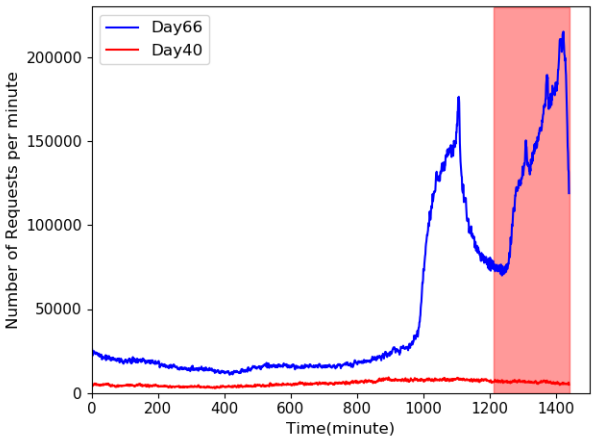

図 5 40 日目と 66 日目におけるリクエスト量の推移

4.1 評価実験に用いるデータセット

評価実験に用いるデータセットとして，フラッシュイベント時のデータセットと DDoS 攻撃時のデータセットのデータセットを用いる．これら 2 つのデータセットについて説明する．

4.1.1 フラッシュイベント時のデータセット

フラッシュイベント時の実験用のデータセットとして，1998 FIFA World Cup[12] を用いる．1998 FIFA World Cup は，文献 [10]，[11]，[13] など，フラッシュイベントに関する研究において，フラッシュイベントを示すデータセットとして広く活用されている．このデータセットは 92 日間のデータセットであり，各地に設置されたサーバが受信した 1,352,804,107 個の HTTP リクエストを含んでいる．このデータセットの 92 日間におけるリクエスト数の推移を図 4 に示す．横軸が時間 (日) で，縦軸がサーバが受信したリクエスト数を表す．また，図 4 の 40 日目と 66 日目に着目したリクエスト数の推移を図 5 に示す．横軸が時

間 (分) で，縦軸がサーバが受信したリクエスト数を表す．

評価実験にはフラッシュイベントが発生していると考えられる，1998 FIFA World Cup 66 日目の約 230 分間を用いた (図 5 の赤い領域)．この約 230 分を用いたのは，図 4 から読み取れるように 66 日目に最大のリクエスト数が観測されており，うち実験で利用した約 230 分間のデータに，1 分間あたりの最大のリクエスト数と急激なリクエスト数の増加が観測されていたからである．この約 230 分間に 63,337 台のクライアントがサーバにアクセスしており，約 200 万個の Web ページを要求するリクエストが含まれていた．

4.1.2 DDoS 攻撃時のデータセット

DDoS 攻撃時の実験用のデータセットとして Bonesi[14] と呼ばれるボットネットシミュレータを利用して，Web サーバに対して DDoS 攻撃を行った際に取得したトラフィックを用いる．DDoS 攻撃時のトラフィックを取得した際の環境の概略図を図 6 に示す．Windows 10 が稼働する PC 上に仮想化ソフトウェアを用い，ゲスト OS として Ubuntu 14.04 LTS が動作する仮想環境を構築した．その際に，ゲスト OS にメモリを 4G，2 つの CPU コアを割り当てた．その後，Ubuntu 14.04 LTS が動作するゲスト OS 内で，Bonesi を用いて Apache2 がインストールされている Web サーバに対して DDoS 攻撃を行い，tshark を用いてパケットキャプチャを行った．

Bonesi は DDoS 攻撃実行時にアクセスするボットの台数を指定できる．文献 [15] によると，ボットネットに属するボットの台数の平均は，約 20,000 台に減少していることが報告されているが，文献 [16] では，観測対象のボットネットに属するボットの台数は約 180,000 台と報告されており，ボットの台数はボットネット毎に大幅に異なる．本論文の実験では，DDoS 攻撃を行うボットの台数は，文献 [15] で報告されている平均のボットの台数，約 20,000 台より 10,000 台多い 30,000 台とし，DDoS 攻撃のトラフィックを 60 秒間取得した．Bonesi を用いた DDoS 攻撃実行時の Web ページを要求するリクエスト数の推移を図 7 に示す．横軸が時間 (秒) で，縦軸がサーバが受信した，Web ページを要求するリクエストの数を表す．取得したトラフィックの中には Web ページを要求するリクエストが約 90 万個含まれていた．さらに，Bonesi の実行時にボットがアクセスする URL 群を指定することができる．本論文では，1998 FIFA World Cup に含まれている htm ファイルや html ファイルをボットがアクセスする Web ページとして設定した．

4.2 評価実験時におけるパラメータ

本実験において各パラメータの値は $U_T = 60$，$S_C = 3$，$G_C = 10$，$G_P = 60$，$T_R = 4$，$T_H = 0.3$，とした．ここで U_T，S_C，G_C，G_P は経験的に定めた．

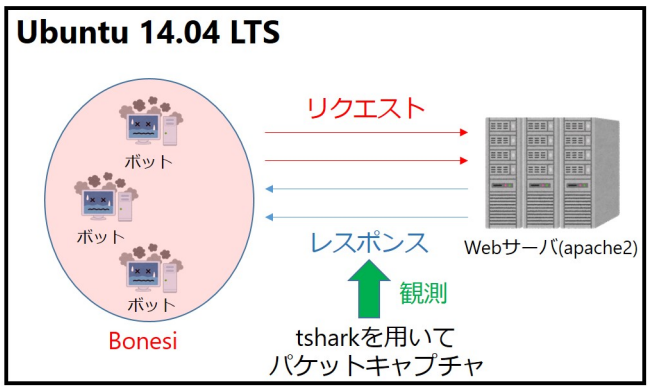

図 6　DDoS 攻撃のデータセット取得時の環境

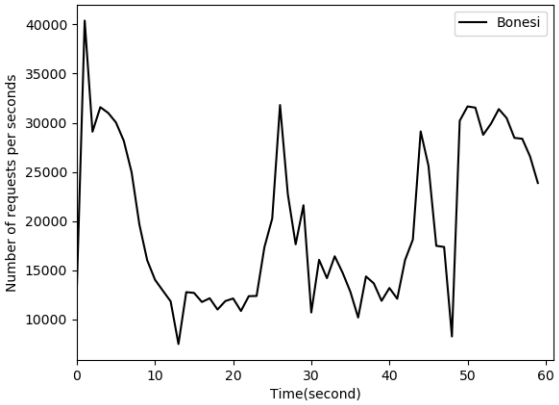

図 7　Bonesi を用いた DDoS 攻撃実行時のリクエスト数の推移

クライント毎の単位時間当たりのリクエスト量に関するしきい値である，T_R の値は通常時のサーバのログとして 1998 FIFA World Cup の 40 日目を用いて計算した．これは，40 日目におけるリクエスト数の推移 (図 5) からフラッシュイベントや DDoS 攻撃が発生しておらず，通常時のサーバのログとして利用できると考えたからである．40 日目を用いて計算を行った結果，$\mu_R = 2.13$，$\sigma_R = 2.01$ であった．また，$\alpha_R = 1$ と定め，小数第一位の四捨五入を行った結果，$T_R = 4$ と決定した．

Web ページを要求するリクエスト送信間隔の類似性に関するしきい値である，T_H の値は Bonesi によって DDoS 攻撃を行った際に取得したトラフィックからランダムに選択した 10 台のボットのトラフィックを用いて計算した．具体的には，ランダムに選択した 10 台のボットの全ての組み合わせにおいて Web ページを要求するリクエスト送信間隔の類似度を求め，それらの平均 μ_H と標準偏差 σ_H を計算した．ランダムに選択した 10 台のボットのトラフィックを用いて計算した結果，$\mu_H = 0.25$，$\sigma_H = 0.05$ であった．また，$\alpha_H = 1$ と定めた結果，$T_H = 0.3$ と決定した．

4.3 提案手法の検知精度

提案手法の検知精度を示し，有用性を確認する．最初に提案手法の検知精度の評価指標として利用する，Detection Rate (DR) と False Positive Rate (FPR) の定義を示す．次に，フラッシュイベント時のデータセットと DDoS 攻撃時のデータセットを用いた実験結果を示し，提案手法の有用性を確認する．

4.3.1 評価指標

提案手法の検知精度の評価指標として Detection Rate (DR) と False Positive Rate (FPR) を用いる．DR はボットをボットとして正しく検知した割合を示し，式 (9) で定義される．FPR は人間をボットとして誤って検知した割合を示し，式 (10) で定義される．DR や FPR は検知システムの検知精度を測る指標として，文献 [11] や文献 [17] などで利用されている．DR の評価には 4.1.2 において説明した，Bonesi を用いて作成した DDoS 攻撃時のデータセットを用い，FPR の評価には 4.1.1 において説明した，フラッシュイベント時のデータセットである 1998 FIFA World Cup 66 日目の約 230 分間を用いた．

$$DR = \frac{TP}{TP + FN} \tag{9}$$

$$FPR = \frac{FP}{TN + FP} \tag{10}$$

式 (9)，式 (10) における各記号の定義を表 1 に示す．

表 1　各記号の定義

記号	定義
True Positive (TP)	ボットをボットとして検知した数
True Negative (TN)	人間を人間として判定した数
False Positive (FP)	人間をボットとして検知した数
False Negative (FN)	ボットを人間として判定した数

4.3.2 実験結果

提案手法の検知精度を表 2 に示す．表 2 から，提案手法の DR は 0.93 で FPR は 0.04 となっており，十分な検知精度を持っていると考えられる．この結果から，DDoS 攻撃時にはボットを適切に検知でき，かつフラッシュイベント時には人間を発見することが可能であることが分かる．これより，大量のトラフィックやクライアントの要因が特定できるため，DDoS 攻撃とフラッシュイベントの識別が可能となる．また，提案手法ではクライアント毎に人間或いはボットかの識別を行うため，DDoS 攻撃時においても人間のアクセスを遮断することは無い．

本論文の実験では 60 秒間の DDoS 攻撃のデータセットを用いたため，DR の値は，ボットの 93%がクライアントの Web ページに対するリクエスト送信間隔の類似性の特徴によってボットと検知されたことを意味する．

また，本論文の実験では $S_C = 3$ と設定しているため，180 秒以上の DDoS 攻撃のデータセットを用いると，提案手法の DR は 1 に近づいていくことが考えられる．なぜならば，DDoS 攻撃を行うボットが $S_C = 3$ 回ボットの疑いのあるクライアントとしてマークされ，その結果，ボットと判定されるからである．

提案手法の FPR は 0.04 であるが，これは人間がボットの疑いとマークされた回数が $S_C = 3$ となったため，ボットと判定されたからである．

表 2　提案手法の検知精度

DR	FPR
0.93	0.04

5. まとめ

本論文では，クライアントの HTTP リクエストの送信動作に着目したボット検知手法の提案を行った．提案手法は，クライアント毎の単位時間当たりのリクエスト量，Web ページを要求するリクエスト送信間隔の類似性，大量のリクエスト送信の継続性の 3 つの特徴を用いて，クライアント毎に人間或いはボットかの識別を行う．提案手法により，大量のトラフィックやクライアントが観測された際に，それらの要因が特定できるため，DDoS 攻撃とフラッシュイベントの識別が可能となる．また，クライアント毎に人間或いはボットかの識別を行うため，DDoS 攻撃時においても人間のアクセスを遮断することはない．

また，フラッシュイベント時のデータセットと DDoS 攻撃時のデータセットを用いて，提案手法の検知精度について評価実験を行った結果を示した．評価実験の結果，提案手法の DR は 0.93，FPR は 0.04 であり，十分な検知精度を持っていると考えられる．

今後の課題としては，本論文の実験ではパラメータ U_T，S_C，G_C，G_P を経験的に定めたが，これらのパラメータを適切に設定する仕組みが必要である．また，本論文では Bonesi と呼ばれるボットネットシミュレータを用いて作成したトラフィックを評価実験に用いており，実際に観測された DDoS 攻撃のトラフィックを用いていない．さらに現在の検知システムではクライント毎の単位時間当たりのリクエスト量に関するしきい値 T_R をぎりぎり下回るように DDoS 攻撃を実行された場合，ボットを正しく検知することが出来ない．このような DDoS 攻撃におけるボットを適切に検知するための特徴の選択や仕組みが必要である．

謝辞　本研究は JSPS 科研費 JP17H01736, JP17K00139, JP18K11268 の助成を受けたものです．

参考文献

[1] Deshmukha, R. V. and Devadkarb, K. K.: Understanding DDoS Attack & Its Effect In Cloud Environment, Procedia Computer Science 49 pp.202-210 (2015).
[2] “マルウェア「Mirai」による DDoS 攻撃が多発”, トレン

ドマイクロ is702 available at:https://www.is702.jp/news/2050/ (2016) (accessed 2018/05/03).
[3] “DDoS 攻撃規模は 5 年で 12 倍に増加”, ZDNet Japan available at:https://japan.zdnet.com/article/35096332/ (2017) (accessed 2018/05/03).
[4] Behal, S., Kumar, K. and Sachdeva, M.: Characterizing DDoS attacks and flash events: Review, research gaps and future directions, Computer Science Review 25 pp.101-114 (2017).
[5] “Freedom 251 website down for second day”, The Hindu available at:http://www.thehindu.com/sci-tech/technology/gadgets/freedom-251-website-down-for-second-day/article8257501.ece (2016) (accessed 2018/05/03).
[6] “即時買い取りサービス「メルカリＮＯＷ」開始　サーバーわずか 1 7 分でダウン”, 産経 ニュース available at:http://www.sankei.com/economy/news/171127/ecn1711270029-n1.html (2017) (accessed 2018/05/03).
[7] Thing, V.L.L., Sloman, M. and Dulay, N.: A Survey of Bots Used for Distributed Denial of Service Attacks, 22nd IFIP International Information Security Conference pp.229-240 (2007).
[8] Acarali, D., Rajarajan, M., Komninos, N. and Herwono, I.: Survey of approaches and features for the identification of HTTP-based botnet traffic, Journal of Network and Computer Applications 76 pp.1-15 (2016).
[9] 小島 俊輔, 中嶋 卓雄, 末吉 敏則: エントロピーベースのマハラノビス距離による高速な異常検知手法, 情報処理学会論文誌 Vol.52 No.2 pp.656-668 (2011).
[10] Sajal Bhatia: Ensemble-based model for DDoS attack detection and flash event separation, Future Technologies Conference pp.958-967 (2016).
[11] Saravanan, R., Shanmuganathan, Y. and Planichamy, Y.: Behavior-based detection of application layer distributed denial of service attacks during flash events, Turkish Journal of Electrical Engineering & Computer Sciences 24 pp.510-523 (2016).
[12] 1998 World Cup Web Site Access Logs available at:http://ita.ee.lbl.gov/html/contrib/WorldCup.html(accessed 2018/05/03).
[13] Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y. and Tang, F.: Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, No.6, pp.1073-1079 (2012).
[14] GitHub - Markus-Go/bonesi: BoNeSi - the DDoS Botnet Simulator available at:https://github.com/Markus-Go/bonesi(accessed 2018/05/03).
[15] “Bots slim down to get tough”, CNET News available at:https://www.cnet.com/au/news/bots-slim-down-to-get-tough/ (2005) (accessed 2018/05/03).
[16] Stone-Gross, B., Cova, M., Cavallaro, L., Gilbert, B., Szydlowski, M., Kemmerer, R., Kruegel, C. and Vigna, G.: Your Botnet is My Botnet: Analysis of a Botnet Takeover, Proceedings of the 16th ACM Conference on Computer and Communications Security (2009).
[17] Sachdeva, M., Kumar, K. and Singh, G.: A comprehensive approach to discriminate DDoS attacks from flash events, JOURNAL OF INFORMATION SECRITY AND APPLICATIONS 26 pp.8-22 (2016).