Vol.2020-MBL-97 No.4
Vol.2020-ITS-83 No.4
2020/11/24

特性が未知のボトルネックリンクに対して有効な Low-rate DDoS 攻撃戦略の検討

髙橋 佑太[1]　稲村 浩[2]　中村 嘉隆[2]

概要：Low-rate DDoS 攻撃は短いオンオフパルスを用いて，TCP/IP プロトコルやルーティングキュー管理機構の脆弱性を悪用することによって，低い平均通信量で TCP リンクの品質を低下させる．最も古典的な RTO ベースの LDDoS 攻撃（別名：Shrew 攻撃）における攻撃パルスの送信レートの理想値は標的ボトルネックリンクの帯域幅以上の値であり，攻撃者が適切な大きさの送信レートを設定できることが前提となっている．しかし，現実において攻撃者が常に標的ボトルネックリンクの帯域幅を把握しているとは限らない．本稿では，限定的な攻撃シナリオのもとで，帯域幅とバッファサイズが未知のボトルネックリンクに対する Low-rate Shrew DDoS 攻撃の実行戦略を提案する．提案する攻撃戦略は，標的ネットワーク内にボットノードを構築することで攻撃効果を推定しながら，探索的にパルスレートを増加することによって，攻撃トラフィックをステルスに保ちながら，標的ネットワークに対する特定サービスからの下りトラフィックを目的の品質まで妨害することを目的としている．提案戦略を ns-3 を用いたシミュレーションにより評価した結果，提案戦略を実行することにより，ボトルネックリンクの特性が未知の場合においても，攻撃パルスの送信レートを探索的に決定可能なことを示した．加えて，ボトルネックリンクのバッファサイズを増加することによって，提案戦略はステルス性を損ない，既存のフラッド型 DDoS 攻撃の検知スキームに発見されるリスクが高まることを示した．

キーワード：ネットワークセキュリティ，Low-rate DDoS 攻撃，TCP 輻輳制御

1. はじめに

分散型サービス妨害（DDoS: Distributed Denial of Service）攻撃は，インターネットを代表する脅威のひとつである．近年では，クラウドサービスの可用性を低下させることや，DDoS 攻撃代行サービス [1] によってサイバー犯罪者が金銭を得ることを目的に DDoS 攻撃が実行される．DDoS 攻撃の対策が進むにともない，DDoS 攻撃は攻撃強度とステルス性の 2 つの側面で進化を遂げている．攻撃強度は年々増加の一途をたどり，最大攻撃レートは 2020 年 2 月 17 日に Amazon 社によって，同社のマネージド脅威防御サービス「AWS Shield」が観測した 2.3Tbps にまで達している [2]．ステルス性の側面では，高レート DDoS 攻撃の検知スキームを回避することが可能な低レート DDoS（LDDoS: Low-rate DDoS）攻撃へと進化している．

LDDoS とは，インターネット上で利用されているプロトコルの脆弱性を利用することで，必要な攻撃トラフィックの平均通信量を低く抑えることを可能にした DDoS 攻撃の一種である．2003 年に Kuzmanivic と Nightly[3] によって Shrew 攻撃と呼ばれる LDDoS 攻撃が示されて以降，様々なプロトコルを標的とした LDDoS 攻撃手法が示されている [4]．LDDoS 攻撃の最大の強みは高レート DDoS 攻撃に対する検知手法を回避するステルス性であり，その検知や防御について様々な手法が検討されている [5]．

本稿では，LDDoS 攻撃の手法として Shrew 攻撃 [3][6] を扱う．Shrew 攻撃は，TCP の再送信タイムアウト機構の脆弱性を利用する LDDoS 攻撃の一種である．経路上のボトルネックリンクに対して，短く高レートのパルス状の攻撃トラフィックを周期的に送信することで，標的の TCP 通信を継続した再送タイムアウト状態に陥らせて妨害する．

Shrew 攻撃を成功させるためには，送信する攻撃パルスの合計ピークレートが標的ネットワークのボトルネックリンク帯域幅以上の値に設定される必要がある [6][7]．これを言い換えると，現実の攻撃シナリオにおいて，攻撃者が標的 TCP フローの経路上のボトルネックリンク帯域幅をあらかじめ知っていることが必要であることを意味する．しかし，現実の攻撃シナリオにおいて，攻撃者が常に標的ボトルネックリンクのパラメータを熟知しているとは限ら

[1] 公立はこだて未来大学大学院 システム情報科学研究科
[2] 公立はこだて未来大学 システム情報科学部

ない．

そこで，本稿では，攻撃者が攻撃パルスの送信に必要なネットワークパラメータを熟知しているという前提を取り払い，攻撃パルスのパラメータの中で最も重要なパルスレートに関連するボトルネックリンクの帯域幅とバッファサイズが未知の場合を想定し，攻撃強度の自動最適化のための LDDoS 攻撃戦略（以下，提案戦略）について検討する．提案戦略は，ボットネットによって自動化されることを想定しており，ネットワークパラメータが未知のボトルネックリンクに対して有効な攻撃パルスレートの最適化を，ステルス性を維持しながら行うことを目的とする．

本稿の貢献を以下に示す．

- 標的 TCP フローと競合する観測 TCP フローから攻撃効果を推定することで，特性が未知のボトルネックリンクに対して，自動でパルスレートの最適化を行い，攻撃者が指定した値まで標的 TCP の品質を低下させることが可能であることを示した．
- 標的ネットワークに送信されたパルスレートを観測することで，ボトルネックリンク帯域幅を推定し，ボトルネックリンクの特性が未知の場合においてもステルス性の高い LDDoS 攻撃が実行可能であることを示した．
- 提案戦略の対策として，ボトルネックリンクのバッファサイズを大きく設定することを提案し，バッファサイズを対象 TCP フローの帯域遅延積以上に設定することで，攻撃パルスが過剰に送信されステルス性を失うことを示した．

本検討の主な動機は，LDDoS 攻撃のユースケースを明らかにすることである．LDDoS 攻撃は実行難易度が高く，攻撃の実例の報告がほとんど存在しない [5] ため，現実における有効性が不明である．本稿で提案する攻撃戦略は，家庭ネットワークへ向けて送信される TCP を標的とすることを想定しており，LDDoS 攻撃代行サービス [1] などによって，実行される可能性があると考えている．

本稿は以下のように構成される．2 章では LDDoS 攻撃の攻撃原理を説明する．3 章では特性が未知のボトルネックリンクに対する LDDoS 攻撃の前提条件と要件を整理し，攻撃のシナリオをモデル化する．4 章では提案戦略の概要とアルゴリズムの詳細を説明する．5 章では提案戦略をシミュレータ上に実装し，有効性と性能を評価する．6 章では提案戦略に対して有効な対策手法について議論する．7 章では関連研究を紹介する．最後に 8 章で本稿のまとめと今後の課題について述べる．

2. 背景

2.1 再送タイムアウト機構を標的とした Shrew 攻撃

TCP 通信においてパケットが送信されると，再送信タイマーがスタートする．再送信タイマーの最大待ち時間を再送信タイムアウト（RTO:Retransmission Time Out）と呼び，送信したセグメントに対する送達確認応答が RTO 時間以内に返ってこない場合，TCP は当該パケットが廃棄されたと判断し再送信する．多くの場合で，RTO の初期値は送信者固有の値である $minRTO$ が設定される．$minRTO$ は RFC6298[8] により，1 秒に設定することが推奨されている．TCP は 2 回以上連続して同じパケットがタイムアウトした場合，当該パケットが再送なく正常に応答を返すまでタイムアウトごとに RTO の値を 2 倍ずつ増加させていく指数バックオフが組込まれている．

Shrew 攻撃は TCP の RTO メカニズムの指数バックオフの弱点を利用する低レート攻撃である [3][5][6]．Shrew 攻撃は，TCP 送信者がタイムアウトから回復してパケットを再送する間隔にあわせて，ボトルネックリンクを瞬間的に輻輳させるパルス状の攻撃トラフィックを送信することで，送信者が新しいデータを送信できないタイムアウト状態を繰り返し誘発する．これによって，正当な TCP フローのスループットは非常に低いか，ほぼゼロになる [7]．攻撃原理のさらなる詳細は文献 [5] に記載されている．

Shrew 攻撃の典型的なモデルを図 1 に示す．モデルは 3 つのパラメータ $< R, L, T >$ で表すことができる．ここで，R は攻撃パルスのレート，L は攻撃パルスの持続時間，T は隣接する攻撃パルス間の間隔である．攻撃が成功するためにはそれぞれの攻撃パラメータが，R がボトルネックリンク帯域幅 C 以上のレート，L が 2RTT∼3RTT または，ボトルネックリンクのバッファを満たすために十分な長さ，T が標的 TCP の $minRTO$ 以上の間隔となる必要がある [5][7]．攻撃パルスの平均レートは $R \cdot L/T$ で算出される．この大きさは，ボトルネックリンク帯域幅のおよそ 10% ∼ 20% 程度 [5] と言われており，フラッド型 DDoS 攻撃の検知手法では攻撃パルスを検出することができない [5]．例えば，帯域幅 $C = 10Mbps$ のボトルネックリンクに対して，$R = 10Mbps$，$L = 200ms$，$T = 1000ms$ の攻撃パルスを送信した場合，平均レートは $2Mbps$ であるため，この攻撃パルスがボトルネックリンクを占める割合は 20%となる．

本稿では以降，複数の攻撃ノードから実行される Shrew 攻撃を LDDoS 攻撃と呼ぶ．

2.2 LDDoS 攻撃による攻撃パルスの分散

LDDoS 攻撃は，ボットネットによって構築した複数の攻撃ノードから攻撃パルスを送信する．LDDoS 攻撃を N 台のノードで構成した攻撃モデルを図 2 に示す．最も基本的な LDDoS 攻撃は，N 台の攻撃ノードから，レート R/N，間隔 T，幅 L，の攻撃パルスを送信する．攻撃ノード N 台分の攻撃パルスをボトルネックリンクで適切に集約できた

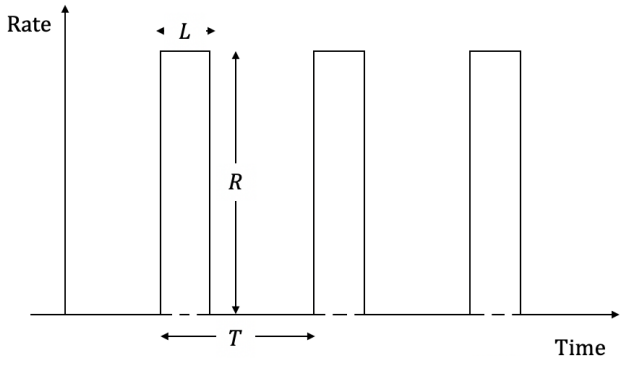

図 1　Shrew 攻撃の一般的な攻撃モデル

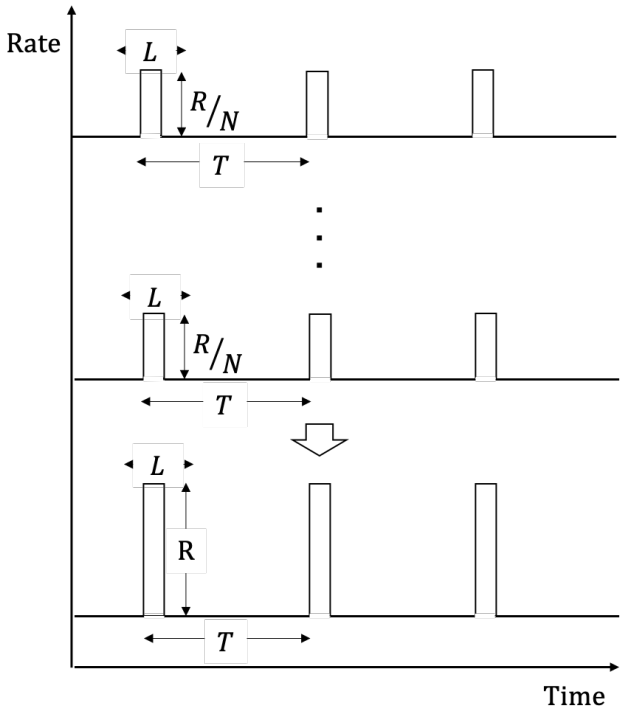

図 2　LDDoS 攻撃モデル N 台の攻撃ノードによるバーストレート R の集約

場合，集約されたトラフィックのパルスレートは R となる．攻撃パルスを分割することにより，個々の攻撃パルスの平均通信量がさらに低くなるため，検知が困難になる．その他の分散手法は文献 [9] に詳細に記載されている．

3. 特性が未知のボトルネックリンクに対する LDDoS 攻撃

本章では，提案戦略を検討する動機を述べた後，前提条件と要件に加えて，実行を想定した攻撃シナリオの概略と現実における攻撃シナリオの適用例を説明する．

インターネット上で攻撃者が任意のボトルネックリンクを標的として LDDoS 攻撃を実行する場合，パルスレート R をボトルネックリンク帯域幅 C 以上のレートに設定する必要があることを 2 章で説明した．これまでの LDDoS 攻撃の研究 [3][5][6][7] では，攻撃者にボトルネックリンク

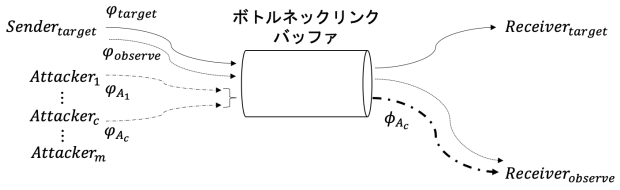

図 3　攻撃シナリオ

帯域幅 C が与えられていることが前提となっている．このような前提のもとでは，パルスレート R を容易に決定することが可能であるが，攻撃者が標的のネットワークについて熟知している必要がある．したがって，理想的な R が設定できることを前提とした評価は，現実における LDDoS 攻撃の有効性を過大評価している可能性がある．

現実的な攻撃シナリオでは，攻撃者に C が与えられていない可能性も十分に考えられる．この場合に攻撃者が最初から理想的な強度のパルスレートを設定することは困難である．攻撃者が適当にパルスレート R を決定した場合，次の問題で攻撃に失敗する可能性がある．

- ボトルネックリンク帯域幅に対して攻撃パルスが小さく，十分な攻撃効果を達成できない．
- ボトルネックリンク帯域幅に対して攻撃パルスを過剰に送信し，攻撃フローの平均がフラッド型 DDoS の攻撃平均ビットレートに近づくことでステルス性を失い，既存の検知手法で検知されてしまう．

したがって，本稿では，特性が未知のボトルネックリンクに対する LDDoS 攻撃の有効性を評価するために，以下の前提条件と要件，および攻撃シナリオを満たした提案戦略を検討する．

3.1 前提条件と要件

提案戦略では，ボトルネックリンクの特性（帯域幅とバッファサイズ）が攻撃者に与えられないことを前提とする．この前提を攻撃者，並びに提案戦略の観点から言い換えると，理想的な攻撃強度（攻撃パルスのピークレート R）を事前に決定することができないことを意味する．

提案戦略には，上記の前提のもとで効果的な LDDoS を実行するために，以下に示す要件が求められる．

(1) 標的の TCP フローのスループットを目的値以下に低下させるために必要なパルスレート R を自動で決定できること
(2) 高レート DDoS 攻撃の検知を回避するために，攻撃フローの平均レートがボトルネックリンクを占める割合を可能な限り小さくすること

3.2 攻撃シナリオ

上記の要件を満たすために必要な攻撃シナリオを図 3 に示す．標的 TCP 送信ノード $Sender_{target}$ は，標的 TCP 受

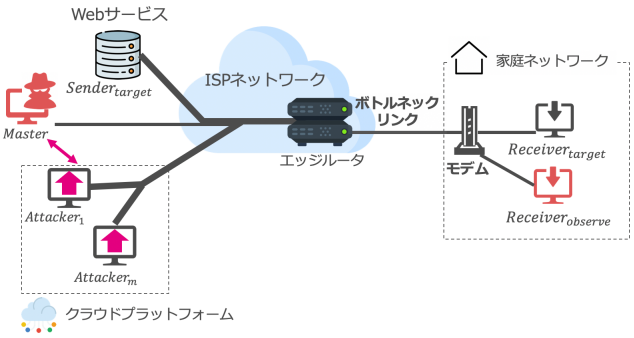

図 4　現実における攻撃シナリオの適用例

信ノード $Receiver_{target}$ からのリクエストに対して，バルク転送の TCP フロー ϕ_{target} を送信する．

攻撃者は，観測ノード $Observer$ を $Receiver_{target}$ が接続されている LAN に構築する．$Observer$ の役割は，攻撃効果の観測と，攻撃パルスの受信と観測である．$Observer$ は $Receiver_{target}$ と同様に，正規のリクエストによって，$Sender_{target}$ からバルク転送の観測 TCP フロー $\phi_{observe}$ を受信する．$\phi_{observe}$ は，4 章で説明する手法によって，目的の攻撃効果を達成しているかを判別するために使用する．ここで $\phi_{observe}$ は，正規のリクエストによって通信する正常なトラフィックであるため，攻撃トラフィックに含まれない．さらに，m 台の攻撃ノード $Attacker_{1\cdots m}$ のうち，アクティブな $c(\forall c \in [1..m])$ 台の攻撃ノード $Attacker_{1\cdots c}$は，$Observer$ に対して LDDoS 攻撃フロー $\Phi_{A_c} = (\phi_{A_1} \cdots \phi_{A_c})$ を送信することで，同じボトルネックリンクを共有する ϕ_{target} の平均スループット δ_{target} を低下させる．アクティブ攻撃ノード数 c の制御については 4 章で説明する．

3.3　現実における攻撃シナリオの適用例

図 4 に攻撃シナリオを現実に適用した一例を示す．$Sender_{target}$ は，TCP バルク転送で大容量データを送信するオンラインストレージサービスや動画共有サービスなどの Web サービスである．$Receiver_{target}$ はインターネットサービスプロバイダ（ISP）のブロードバンドリンクを通して接続された家庭ネットワーク内部の端末である．すなわち，標的となる TCP フロー ϕ_{target} は，インターネット上のサービスが家庭ネットワークの一般ユーザに向けて送信するトラフィックである．

$Sender_{target}$ と $Receiver_{target}$ 間のボトルネックリンクとなるのは，家庭ネットワークと ISP ネットワークのラストホップである．これは，文献 [10] のブロードバンド測定研究の結果から裏付けることができる．したがって，攻撃フロー Φ_{A_c} は，ISP ネットワークの外側から，家庭ネットワークに向けて送信される．LDDoS 攻撃はクラウドコンピューティングに対して脅威であることが指摘されている [5] が，2013 年に発生した NTP DDoS 攻撃は，その多くが個人（エンドホスト）を対象に開始されていることが明らかになっている [11] ことから，LDDoS 攻撃についても，代行サービスによって個人が標的となる可能性は十分に考えられる．

$Observer$ は，$Receiver_{target}$ が接続されている LAN 内に存在する脆弱な端末を Mirai Botnet[12] のようなマルウェアを用いて，ボット化することで構築する．ここで，$Sender_{target}$ の通信を直接観測する手法も考えられるが，その場合，イーサネットであれば標的ネットワークのルータに接続し，プロミスキャス・モードで通信を観測する必要があることや，Wi-Fi の場合，通信の暗号化を解くことが必要となることを考えると、制御を奪取した端末を $Observer$ として利用し攻撃トラフィックや標的フローを単体で観測させるほうが容易に構成可能であると考えられる．$Attacker_{1\cdots m}$ は，クラウドコンピューティングの特定リージョンの Infrastructure as a Service を利用して構築した仮想マシン上で動作する．これによって，すべての攻撃ノードからボトルネックリンクまでの遅延を統一し，攻撃フローを正確に集約できる．

4.　攻撃強度の自動最適化のための LDDoS 攻撃戦略

4.1　提案戦略の概要

提案戦略では，図 5 に示すフィードバック制御によって，攻撃強度を最適化する．攻撃制御ノード $Master$ が攻撃全体を制御する役割を持つ．$Master$ は目的攻撃効果 $throughput_{objective}$ を入力として受け取り，フィードバックされた観測 TCP フロー $\phi_{observe}$ のスループットのサンプル値 $\theta = (\vartheta_1 \cdots \vartheta_t)$ をもとに，アクティブ攻撃ノード数 c を決定する．ここで，ϑ_t は，$\phi_{observe}$ の通信開始後の $t-1$ 秒から t 秒までの間に $Observer$ で計測したスループットである（t は自然数）．そして，アクティブな c 台の各攻撃ノードは，$< \Delta R, L, T >$ のパラメータで攻撃パルスを送信し，ボトルネックリンクで合成される攻撃パルスのパラメータは $< R = \Delta R \cdot c, L, T >$ となる．1 回のフィードバック動作につき，観測窓幅 W 秒の長さだけ θ を計測する（W は自然数）．例えば，$W = 3$ のとき，1 回目のフィードバックでは，$\theta = (\vartheta_1, \cdots, \vartheta_3)$，2 回目のフィードバックでは，$\theta = (\vartheta_1, \cdots, \vartheta_3, \cdots, \vartheta_6)$ を得る．もう 1 つのフィードバック値 R_{max} は，目的攻撃効果の達成後にボトルネックリンク帯域幅 C を推定するために使用される．$Master$ はこの値を使用して，攻撃のステルス性を低下させる．

アルゴリズム 1 に，上記のフィードバック制御を $Master$ に実装したより詳細な手順を示す．関数 $ControlAttack$ は，$Master$ によって観測窓幅 W 秒ごとに繰り返し実行されフィードバック制御を実現する．処理の内容を要約すると，各攻撃ノードが送信するパルスパラメータを初期

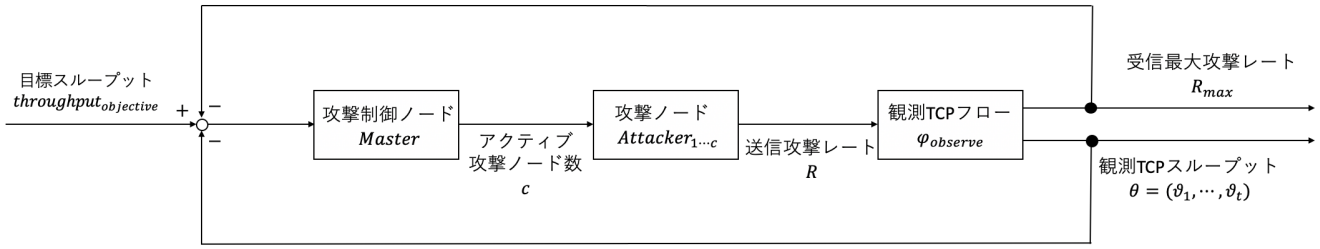

図 5　提案戦略のフィードバック制御

化（4.2 節で説明）した後，攻撃パルス送信毎に現在の攻撃効果の推定値（4.3 節で説明）をもとに，アクティブ攻撃ノード数 c の増加量 $incrementC$ を決定（4.4 節で説明）し，$SendAttackPulse$ を実行して W 秒間 LDDoS 攻撃を実行する命令を $Attacker_{1\cdots c}$ に送信する．さらに，攻撃のステルス性を維持するために，目的攻撃効果の達成後にボトルネックリンクを推定し，その大きさまで合成後の攻撃パルス R の大きさを下げる機能をもつ（4.5 節で説明）．

4.2　攻撃パルスのパラメータ設定

アクティブな各攻撃ノード $Attacker1\cdots c$ は，パラメータ $<\Delta R, L, T>$ で定義される攻撃パルスを送信する．L, T は固定値設定し，ΔR のみ，ボトルネックリンクの帯域幅に応じた値を使用する．パルス間隔 T は，標的 TCP 送信ノード $Sender_{target}$ の初期 RTO である $minRTO_{target}$ 以上の値を設定する必要がある．パルス幅 L は，T に対して十分に小さくなければならない [7]．本稿では，$\phi_{target,observe}$ の RTT が $300ms$ 以下である前提をおき，$L = 300ms$ に固定する．各攻撃ノードが送信するパルスレート ΔR は，ボトルネック帯域幅 C を超えない値に設定する必要がある．そこで，提案戦略では攻撃パルスの送信（$ControlAttack$ の実行）前の安定したスループットの平均値を ΔR に設定する．具体的には，Algorithm1 の 2 行目において，$Observer$ からフィードバックされた θ の，直近 W 秒分のスループットサンプル $(\vartheta_{t-W+1}, \cdots \vartheta_t)$ から平均値 $\delta_{observe}$ を算出する．そのため，攻撃を開始する数十秒前から $\phi_{observe}$ の通信を開始し，輻輳ウィンドウ値がある程度安定した後に，攻撃パルスの送信（$ControlAttack$ の実行）を開始する必要がある．$\delta_{observe}$ は必ず C 以下になるため，C を超えない ΔR が設定され，安全に攻撃を開始できる．

4.3　攻撃効果の推定

攻撃時には，現在の合計パルスレート R が目的の攻撃効果を達成しているかどうかを判断するために，観測 TCP フロー $\phi_{observe}$ を用いて攻撃効果を推定する．図 3 より，ϕ_{target} と $\phi_{observe}$ は，同じボトルネックリンクを共有する同種フローであるため，TCP 親和性のモデル式 [13] により $\delta_{target} \simeq \delta_{observe}$ が成り立つ．この特性を利用して，$\delta_{observe}$ から δ_{target} への攻撃効果を推定する．

4.4　アクティブ攻撃ノード数 c の決定

フィードバックごとに，条件式 2a と条件式 3 を満たしている場合，前回のフィードバック時に送信した攻撃パルスのレート R では目的の攻撃効果を達成できないと判断し，アクティブ攻撃ノード数 c を増加する処理を実行する．アクティブ攻撃ノード数 c の増加量は，Algorithm1 の 11 行目の $computeIncrementC(\Delta R, c, Tp_{obj})$ によって計算される．計算される値は，以下の式 (1) によって，攻撃ノード 1 台あたりの TCP 損失量に比例して攻撃ノードを増加する．

$$incrementC = |round\left(\frac{\theta_{average} - Tp_{obj}}{\frac{\theta_{max}-\theta_{min}}{\theta_{a}verage} \cdot \Delta R \cdot c}\right)| \qquad (1)$$

ここで，θ_{max} は，前回 c を増加した時刻から最後のサンプリング時刻 t までの最大値，θ_{min} は，θ_{max} と同範囲における最小値，$\theta_{average}$ は，時刻 $t - W + 1$ から t までの平均値（$\delta_{observe}$ と同様）である．

算出した $incrementC$ は単純に c に加えるのではなく，合計攻撃ノード数を上回らない範囲で加える（条件式 4a, 5a, 5b）．条件式 4b は，c を増加するべきであると条件式 2a と条件式 3 によって判断されたが，$incrementC$ の値が 0 になってしまったときに 1 度だけ実行される調整処理である．このケースは主に，目的攻撃効果の達成まで c がわずかに足りない場合，すなわち式 (1) の $round$ 内の項の分子が 0 に近い場合を想定している．

4.5　ステルス性の優先制御

提案戦略は，攻撃のステルス性を優先制御する機能をもつ．ステルス性優先制御は，目的攻撃効果が達成された場合に，ボトルネックリンク帯域幅 C を推定して合成後のパルスレート R を C の推定値まで低下させる．攻撃者は，攻撃開始前にステルス性優先制御を有効にするか選択できる．有効化しなかった場合，ステルス性優先制御は実行されない（条件式 2c）．

ボトルネックリンク帯域幅 C の推定は，$Observer$ が，100 ミリ秒間隔で計測した Φ_{A_c} のスループットの最大値

Algorithm 1 Core Algorithm of Master Node

Require: ΔR ▷ 攻撃ノード 1 台あたりのパルスレート（Kbps）
Require: $L \Leftarrow L_{initial}$ ▷ 攻撃ノード 1 台あたりのパルス幅（ms）
Require: $T \Leftarrow T_{initial}$ ▷ 攻撃ノード 1 台あたりのパルス間隔（ms）
Require: $W \Leftarrow W_{initial}$ ▷ 観測窓幅
Require: $c \Leftarrow 0$ ▷ アクティブ攻撃ノード数
Require: $m \Leftarrow m_{initial}$ ▷ 攻撃ノードの総数
Require: $prev\_\delta_{observe} \Leftarrow \infty$ ▷ 前回のフィードバックにおける $\delta_{observe}$
Require: $Tp_{obj} \Leftarrow throughput_{objective}$ ▷ 目的攻撃効果
Require: $initDeltaR \Leftarrow false$ ▷ ΔR 初期化フラグ
Require: $ajust \Leftarrow false$ ▷ c の調整フラグ
Require: $loweredRToBW \Leftarrow false$ ▷ パルスレート R をボトルネック帯域幅の推定値まで低下させたかを表すフラグ
Require: $prioStealthy \Leftarrow true$ **or** $false$ ▷ ステルス性優先制御のオンオフ

function *ControlAttack*
  $\delta_{observe} \Leftarrow computeLatestAverageObserveTp()$
  $noMoreIncrease = loweredRToBW$ **and** $prioStealthy$
  if !($initDeltaR$) **then** ▷ 条件式 1
    $\Delta R \Leftarrow \delta_{observe}$
    $initDeltaR \Leftarrow true$
    $c \Leftarrow 1$
  end if
  if $\delta_{observe} > Tp_{obj}$ **and** !($noMoreIncrease$) **then** ▷ 条件式 2a
    if $\delta_{observe} > prev\_\delta_{observe}$ **then** ▷ 条件式 3
      $incrementC \Leftarrow computeIncrementC(\Delta R, c, Tp_{obj})$
      if $incrementC > 0$ **then** ▷ 条件式 4a
        if $(c + incrementC) \leq m$ **then** ▷ 条件式 5a
          $c \Leftarrow c + incrementC$
        else if $c! = m$ **then** ▷ 条件式 5b
          $c \Leftarrow m$
        end if
      else if $c < m$ **and** !($ajust$) **then** ▷ 条件式 4b
        $c \Leftarrow c + 1$
        $ajust \Leftarrow true$
      end if
    end if
  else if $noMoreIncrease$ **then** ▷ 条件式 2b
    print 'ステルス性優先制御によって設定されたパルスレートを維持する'
  else if $prioStealthy$ **then** ▷ 条件式 2c
    $BW_{probe} \Leftarrow probeBW()$
    $R_{current} \Leftarrow \Delta R \cdot c$
    if $R_{current} > BW_{probe}$ **then** ▷ 条件式 6
      $\Delta R \Leftarrow ceil\,(BW_{probe}/c)$
      $loweredRToBW \Leftarrow true$
    end if
  end if
  $prev\_\delta_{observe} \Leftarrow \delta_{observe}$
  $SendAttackPulse(\Delta R, L, T, c, W)$
end function

R_{max} を利用する．ここで計測する Φ_{A_c} のスループットは，例えば $R = 10Mbps, L = 300ms, T = 1000ms$ の場合，$[10, 10, 10, 0, \cdots, 0]\,(Mbps)$ のように得られる．ただし，攻撃パルスと TCP が競合しており，ボトルネックリンクの帯域利用率が 100%の場合，競合 TCP によって攻撃パケットの一部がボトルネックリンクバッファで損失してしまうため，$Observer$ で計測される R_{max} は，実際に送信した R よりも低下する．そこで，R_{max} の上位一桁の概数を切り上げで求めることによって，C の値を推定する．上位 1 桁の概数とは，上位 2 桁目を切り上げて，上位 3 桁以下を切り捨てた数である．Algorithm 1 の 26 行目で実行している $probeBW()$ は R_{max} の概数からボトルネックリンク帯域幅を推定している．現在送信しているピークレート R が，ここで得られたボトルネックリンクの推定帯域幅 BW_{probe} よりも大きい場合，R が高すぎると判断し，ΔR に BW_{probe} を現在の攻撃ノード数 c で割った値（小数点未満切り上げ）を設定する（条件式 6）．

5. 実験と評価

本章では，3 章と 4 章で説明した提案戦略が正常に機能することを検証し，攻撃性能を評価する．

5.1 実験環境

実験は離散イベントネットワークシミュレータ ns-3[14] を用いたシミュレーション環境で行う．図 6 にシミュレーションに用いたネットワークトポロジと関連する構成を示す．$Sender_{target}$ の $minRTO$ は，推奨値の 1 秒 [3][6][8] に設定した．$Sender_{target}$ が送信する TCP パケットのサイズは 590Byte である．TCP 輻輳制御アルゴリズムは NewReno を使用する．ボトルネックリンクの帯域幅 C とバッファサイズ B は，シミュレーションごとに後述の複数の値を使用する．すべてのノードは DropTail 方式でバッファに蓄積されたパケットを処理する．$Sender_{target}$ と $Receiver_{target,observe}$ の TCP ソケットの送受信バッファサイズはそれぞれ 512KByte に設定した．各ノードが送信するトラフィックは図 3 で示した攻撃シナリオと同様である．$Attacker_{1\cdots c}$ が送信する攻撃トラフィックのプロトコルは UDP である．攻撃パケットのサイズは 80Byte である．攻撃制御ノード $Master$ はトポロジ上に配置せず，ns-3 のイベント動作として実行される．したがって，今回のシミュレーションでは，フィードバック制御に本来発生する $Master$ が関与する通信の遅延は無視される．

5.2 実験内容

提案戦略を評価するために 6 通りのボトルネックリンクの特性に対して実験する．シミュレーション 1-a ～ 3-a は，ステルス性優先制御を無効にし，目的攻撃効果の達成を優先するシミュレーションである．シミュレーション 1-b ～ 3-b は，ステルス性優先制御を有効にしたシミュレーションである．

シミュレーションごとに変動するパラメータを表 1 に示

Vol.2020-MBL-97 No.4
Vol.2020-ITS-83 No.4
2020/11/24

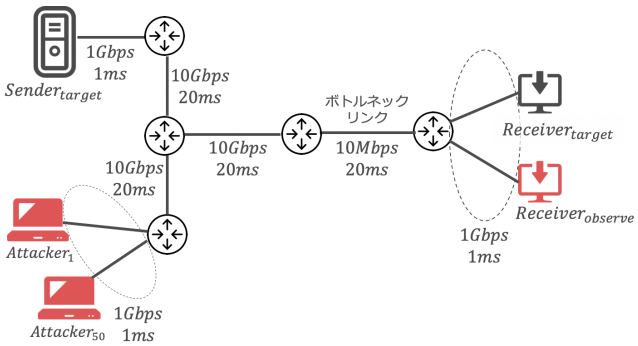

図 6　ns-3 シミュレーションで使用したネットワークトポロジ

表 1　シミュレーションごとに設定したパラメータ

シミュレーション	C	B	ステルス性優先制御 $prioStealthy$
1-a	10Mbps	38,750Bytes	false
2-a	30Mbps	116,250Bytes	false
3-a	50Mbps	193,750Bytes	false
1-b	10Mbps	38,750Bytes	true
2-b	30Mbps	116,250Bytes	true
3-b	50Mbps	193,750Bytes	true

す．ボトルネックリンクのバッファサイズ B は，文献 [15] で提案されているバッファサイズの計算式 $B = C \cdot \overline{RTT}/\sqrt{n}$（$n$ はフロー数）に $n = 16$ を代入した値を計算して設定した．すべてのシミュレーションで共通のパラメータは，$L = 300ms$, $T = 1000ms$, $throughput_{objective} = 500kbps$, $m = 50nodes$，$W = 3s$ である．パルス幅 L は，通常ネットワークの RTT が $10ms$ から $100ms$ であることを踏まえ，タイムアウトを起こすために十分な $300ms$ を設定した．目的攻撃効果 $throughput_{objective}$ は，動画配信サービスの Netflix が公表している必要最低帯域幅 [16] の $500kbps$ に設定した．各シミュレーションは 80 秒間ずつ実行する．シミュレーションにおける各フローの通信開始時刻は，標的 TCP フロー ϕ_{target} が 0 秒，観測 TCP フロー $\phi_{observe}$ が 5 秒，1 ノード目の攻撃フロー ϕ_{A_1} が 25 秒である．ϕ_{target} と $\phi_{observe}$ は通信を開始してからシミュレーションが終わるまでデータを送信し続ける．

5.3　実験結果の評価と考察

5.3.1　攻撃効果

目的攻撃効果を達成することを優先したシミュレーション 1-a ～ 3-a の結果を表 2 に示す．目的攻撃効果達成時刻 t_{suc} 後の ϕ_{target} と $\phi_{observe}$ の平均スループットを比較すると，1-a ～ 3-a のすべてのケースで目的攻撃効果の 500kbps を下回っており，攻撃効果を達成できた．t_{suc} は，帯域幅が増加するごとに遅くなることがわかった．最終ピークレート R は，1-a ～ 3-a のすべてのケースで，ボトルネックリンク帯域幅 C を超える結果となった．これは，B に対して $throughput_{objective}$ が低すぎたことが原因だと考えら

表 2　シミュレーション 1-a ～ 3-a の攻撃効果

評価項目/シミュレーション	1-a	2-a	3-a
目的攻撃効果の達成	✓	✓	✓
目的攻撃効果達成時刻 (t_{suc})	40s	49s	58s
ΔR	2.88Mbps	15.12Mbps	25.0Mbps
c	5	3	3
最終パルスピークレート (R)	14.41Mbps	45.38Mbps	75.01Mbps
t_{suc} 以降の平均攻撃レート (R_{avg})	4.32Mbps	13.61Mbps	22.50Mbps
R_{avg} のボトルネックリンク使用率 u	43%	45%	45%
t_{suc} 以降の ϕ_{target} の平均スループット	392.23kbps	51.31kbps	36.04kbps
t_{suc} 以降の $\phi_{observe}$ の平均スループット	435.30kbps	64.86kbps	33.90kbps

れる．攻撃フローの帯域使用率 u は 43% ～ 45%となった．文献 [5] では，LDDoS フローの帯域使用率は，ボトルネックリンクの 10% ～ 20%程度であると述べられており，この主張と結果を比較すると，ステルス性の観点では改善の余地がある．以上の結果から，提案戦略はボトルネックリンクの特性に影響されず，目的の攻撃効果までパルスレートを調整できることがわかった．一方で，ステルス性優先制御を無効化した状態で，$throughput_{objective}$ に低い値を設定すると，R が C より大きくなってしまい，ステルス性が低下することがわかった．

5.3.2　攻撃のステルス性

ステルス性を優先したシミュレーション 1-b ～ 3-b の R の推移結果を図 7 に示す．1-b ～ 3-b のすべてのケースで，最終パルスピークレート R が，ステルス性を優先しなかったシミュレーション 1-a ～ 3-a の値と等しい値まで増加しているが，その後 C の推定が有効に機能して，R が C まで低下していることが確認できた．すなわち，1-b ～ 3-b のすべてのケースにおいて攻撃フローの帯域使用率 u は 30%となり，シミュレーション 1-a ～ 3-a の結果と比較して 13 ～ 15%減少した．したがって，提案戦略のステルス性優先制御が正確に機能していることが示された．今回のように正確に機能している要因として，すべての攻撃ノードとボトルネックリンク間の遅延が一定であることと，外乱トラフィックがないためジッタがほぼ発生しないことが挙げられる．今後の検討として，外乱トラフィックが発生している状況下や，現実的なパルス同期における評価を行う必要があると考える．攻撃効果の観点では，ステルス性を優先したことにより，1-b ～ 3-b のすべてのケースで目的攻撃効果の 500kbps 以下を達成できない結果となった．この結果は，LDDoS 攻撃に失敗しているのではなく，R が C と等しい場合に得られる最大攻撃効果が 500kbps 以上であるということを意味する．以上の結果と考察から，攻撃効果とステルス性にはトレード・オフの関係があると言える．

6.　提案戦略に対する対策の議論

本章では，ボトルネックリンクルータのバッファサイズを増加することで，提案戦略を失敗に追い込むという単純

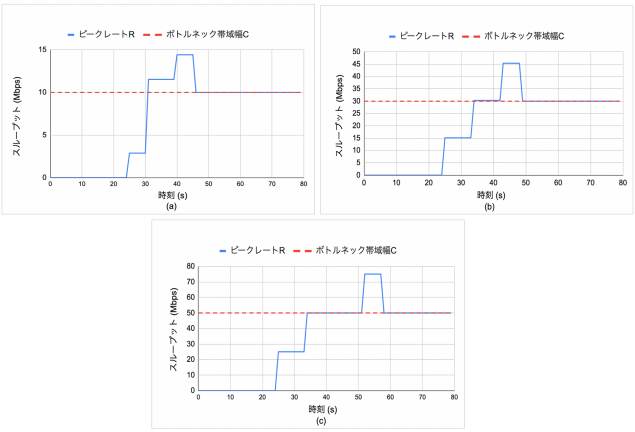

図 7　ステルス性優先制御を有効にしたシミュレーションのピークレート R の推移 (a) 1-b (b) 2-b (c) 3-b

表 3　バッファサイズを 2BDP に増加したボトルネックリンクに対して提案手法で攻撃を実行した結果

目的攻撃効果の達成	×
ΔR	1.68Mbps
c	14
最終パルスピークレート (R)	23.55Mbps
t_{suc} 以降の平均攻撃レート (R_{avg})	7.1Mbps
R_{avg} のボトルネックリンク使用率 u	71%

な防御策について述べる．LDDoS 攻撃は，バッファサイズが大きいほどバースト幅 L の必要最小値が大きくなり，攻撃効果が低下することが明らかになっている [7]．このことから，バッファサイズを増やすことで，目的攻撃効果の達成に必要な R を増加し，攻撃を高レート化する防御策の有効性を評価する．

上記の防御策の有効性を確認するために，5 章で行った実験と同様の内容で，$C = 10Mbps$, $B = 2BDP(RTT \cdot C \cdot 2)$, ステルス性優先制御を無効化したシミュレーションを実行した．結果を表 3 に示す．攻撃を開始してから終了までの間，ϕ_{target} のスループットが目的攻撃効果の 500kbps を達成することはなかった．最終パルスピークレートは 23.55Mbps，R_{avg} は 7.1Mbps となり，u が，5 章攻撃のステルス性が大きく低下する結果となった．文献 [17] では，攻撃パルスの送信を検知して，一時的に B を増加させる防御策が提案されている．上記の結果から，提案戦略についても，一時的に B を増加させる防御策を適用することが有効であると考える．

7. 関連研究

2015 年，Massimo と Massimiliano[18] は，Slowly-Increasing-Polymorphic DDoS Attack Strategy(SIPDAS) を提案した．SIPDAS の目的は，目標とする攻撃効果が得られるまで，攻撃期間ごとに攻撃速度を 1 単位ずつ増加させることで攻撃による被害額と攻撃コストの比率を最大化することである．SIPDAS は，クラウドアプリケーションの計算資源を枯渇させるために Deeply-Nested XML 攻撃を用いており，検知を回避するためにできるだけ長い時間を書けてゆっくりと攻撃強度を増加させる．SIPDAS と本稿の提案戦略のアプローチは，目的の攻撃効果を設定して攻撃強度を最適化するという点で共通しているが，提案戦略は純粋な Shrew 攻撃によってネットワークの品質を劣化させる点，標的が家庭ネットワークである点，できるだけ早く目的の攻撃効果を達成するために，攻撃レートの増分を攻撃効果に比例して変動している点で SIPDAS と異なる．

2020 年，Park ら [19] は，現実的なネットワーク同期を前提とした Very Short Intermittent DDoS 攻撃 (VSI-DDoS) の有効性の評価を行った．VSI-DDoS は，数ミリ秒単位で攻撃トラフィックがサーバに集中することを前提としている攻撃であり，現実における有効性を過大評価されている側面があった．検証の結果，VSI-DDoS は約 90ms の小さな同期のずれが発生しただけで 85.7%の効果が失われ，有効性が低下することを実証した．本稿で扱った Shrew 攻撃についても，現実的なパルス同期の前提で評価することが，現実的な有効性を明らかにするために必要であると考える．

8. おわりに

本稿では，標的ボトルネックリンクの帯域幅とバッファサイズが未知であるという仮定に基づいた LDDoS 攻撃のパルスレート最適化戦略について提案した．提案戦略は限定された攻撃シナリオと環境において，標的の TCP 通信の品質を目的の攻撃効果まで低下させるために必要とされるパルスレートを探索的に決定することができる．加えて，目的攻撃効果達成後のパルスレートがボトルネックリンク帯域幅より高い場合，ステルス性を優先してパルスレートを帯域幅まで抑えることができる．ns-3 を利用した提案戦略のシミュレーションにより，ボトルネックリンク帯域幅の特性に影響されず，目的攻撃効果の達成と攻撃のステルス性の優先が可能であることを確認した．ただし，本稿で評価した攻撃シナリオは最も単純な環境であるため，現実のインターネットでは VSI-DDoS[19] のように，十分な攻撃効果が得られない可能性がある．したがって今後は，外乱トラフィックが競合する環境やトポロジを複雑化した環境並びに，さらに複数のネットワークパラメータを設定した環境における戦略の評価を行う予定である．

謝辞　本研究は JSPS 科研費 JP20K11772 の助成を受けたものです．

参考文献

[1] Booters, Stressers and DDoSers, Imperva (online), available from ⟨https://www.imperva.com/learn/ddos/booters-

stressers-ddosers/⟩ (accessed 2020-10-28).
[2] AWS Shield Threat Landscape Report – Q1 2020, AWS Shield (online), available from ⟨https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf⟩ (accessed 2020-10-19).
[3] Kuzmanovic, A. and Knightly, E. W.: Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants, *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 75–86 (2003).
[4] Agrawal, N. and Tapaswi, S.: Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges, *IEEE Communications Surveys & Tutorials*, Vol. 21, No. 4, pp. 3769–3795 (2019).
[5] Zhijun, W., Wenjing, L., Liang, L. and Meng, Y.: Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey, *IEEE Access*, Vol. 8, pp. 43920–43943 (2020).
[6] Kuzmanovic, A. and Knightly, E. W.: Low-rate TCP-targeted denial of service attacks and counter strategies, *IEEE/acm transactions on networking*, Vol. 14, No. 4, pp. 683–696 (2006).
[7] Luo, J., Yang, X., Wang, J., Xu, J., Sun, J. and Long, K.: On a mathematical model for low-rate shrew DDoS, *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 7, pp. 1069–1083 (2014).
[8] Paxson, V., Allman, M. and Sargent, M.: Computing TCP's Retransmission Timer, Internet RFC 6298 (online), available from ⟨https://tools.ietf.org/html/rfc6298⟩ (accessed 2020-02-18).
[9] Zhang, C., Cai, Z., Chen, W., Luo, X. and Yin, J.: Flow level detection and filtering of low-rate DDoS, *Computer Networks*, Vol. 56, No. 15, pp. 3417–3431 (2012).
[10] Dischinger, M., Haeberlen, A., Gummadi, K. P. and Saroiu, S.: Characterizing residential broadband networks, *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 43–56 (2007).
[11] Czyz, J., Kallitsis, M., Gharaibeh, M., Papadopoulos, C., Bailey, M. and Karir, M.: Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks, *Proceedings of the 2014 Conference on Internet Measurement Conference*, pp. 435–448 (2014).
[12] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M. et al.: Understanding the mirai botnet, *26th USENIX security symposium (USENIX Security 17)*, pp. 1093–1110 (2017).
[13] Padhye, J., Firoiu, V., Towsley, D. and Kurose, J.: Modeling TCP throughput: A simple model and its empirical validation, *Proceedings of the ACM SIGCOMM'98 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 303–314 (1998).
[14] ns-3 — a discrete-event network simulator for internet systems, nsnam.org (online), available from ⟨https://www.nsnam.org/⟩ (accessed 2020-02-18).
[15] Appenzeller, G., Keslassy, I. and McKeown, N.: Sizing router buffers, *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 4, pp. 281–292 (2004).
[16] Internet Connection Speed Recommendations, Netflix (online), available from ⟨https://help.netflix.com/en/node/306⟩ (accessed 2020-10-27).
[17] Maciá-Fernández, G., Díaz-Verdejo, J. E. and García-Teodoro, P.: Mathematical model for low-rate DoS attacks against application servers, *IEEE Transactions on Information Forensics and Security*, Vol. 4, No. 3, pp. 519–529 (2009).
[18] Ficco, M. and Rak, M.: Stealthy Denial of Service Strategy in Cloud Computing, *IEEE Transactions on Cloud Computing*, Vol. 3, No. 1, pp. 80–94 (2015).
[19] Park, J., Mohaisen, M., Nyang, D. and Mohaisen, A.: Assessing the effectiveness of pulsing denial of service attacks under realistic network synchronization assumptions, *Computer Networks*, p. 107146 (2020).