Vol.2018-MBL-89 No.8
Vol.2018-ITS-75 No.8
2018/11/15

実ネットワーク環境下におけるLDDoS攻撃の検証

髙橋 佑太[1]　稲村 浩[1]　中村 嘉隆[1]

概要：インターネット通信において広く使われている TCP は，低量分散型サービス妨害（LDDoS: Low-rate Distributed Denial of Service) 攻撃によって継続的な通信妨害が可能であることが既存研究により明らかにされつつある．しかし，これまで実際に LDDoS 攻撃が行われた事例は確認されておらず，既存研究の実験もシミュレータ上のみで行われており，実ネットワーク環境下において，LDDoS 攻撃を行った際の効果は明らかではない．そこで我々は，実ネットワーク環境下において LDDoS 攻撃を構成・評価し，その知見を用いて LDDoS 攻撃の効果的な検知・抑止の手段を確立することを目指している．実際に攻撃ノードを IoT 機器である Raspberry Pi を用いて，標的のキューサイズが既存研究に示された 50 については実験で，現実的と思われる 1000 については外挿にて必要な攻撃ノード数の見積りを行い LDDoS 攻撃が実現可能であることを示した．LDDoS を成立させる条件の一つである既知の minRTO については一般には成り立たないが，意図的にサーバから受信するパケットを廃棄することで検知されることなく minRTO を検出する手法を提案し簡易な実現例を示した．

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

Yuta Takahashi[1]　Hiroshi Inamura[1]　Yoshitaka Nakamura[1]

1. はじめに

サービス妨害（DoS:Denial of Service）攻撃や分散型 DoS（DDoS:Distributed DoS）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，セキュリティ対策が不十分な IoT（Internet of Things）機器を踏み台とした大規模な DDoS 攻撃が引き起こされていることが問題となっている [1]．インターネットに接続する IoT 機器の台数は年々増加しており，2020 年には 200 億台を超えると予測されている [2] ことから，今後も IoT 機器が DDoS 攻撃の踏み台とされる可能性が十分に考えられる．しかし，このような一般的な DDoS 攻撃は，大量であるという点で攻撃トラフィックの特徴は捉えやすく，単純な手法の特性はよく知られており，検知や対策に利用されている．

一方で，低量 DoS(LDoS:Low-rate DoS) 攻撃 [3] と呼ばれる手法により，低い平均攻撃通信量で TCP 通信の妨害が可能であることが既存研究により明らかにされている．LDoS 攻撃を複数の攻撃ノードから分散して行う LDDoS 攻撃も可能であることが明らかになりつつあり [4][5][6][7]，一般的な DDoS 攻撃の防御手段を回避するよう企図されていることから脅威となり得ることが指摘されている．しかし，これまでに LDDoS 攻撃による被害は確認されていないことから，複雑な実ネットワーク環境下において LDDoS 攻撃が効果的に形成されるのかは明らかではない．これまでの既存研究によって，LDDoS 攻撃の検知手法はいくつか提案されているが，単純なネットワークトポロジやネットワークシミュレータ上のみの実験に留まっているものが多いため，実ネットワーク環境下における動きについて知見が求められている．

本研究は，実ネットワーク環境下にて LDDoS 攻撃を構成・評価し，その知見を用いて LDDoS 攻撃の効果的な検知・抑止の手段を確立することを目的としている．

本稿は以下の通りに構成する．2 章で関連技術，3 章で関連研究における LDDoS 攻撃実験の課題について，4 章で課題解決に向けての研究目的，5 章で目的達成のために今回取り組んだ 3 つのアプローチについて，6 章から 8 章にかけてアプローチの詳細を述べ 9 章でまとめる．

[1] 公立はこだて未来大学　システム情報科学部
School of Systems Information Science, Future University Hakodate

2. 関連技術

2.1 TCP 再送信タイムアウト

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:Retransmisson Time Out）と呼び，RTO 以内に送信したパケットの応答が返ってこない場合，当該パケットは廃棄されたと判断し再送信する．RTO の初期値は RFC6298[10] により，次の式で設定される．

$$RTO = max\{minRTO, SRTT + max(G, 4 \times \ RTTAVR)\} \quad (1)$$

ここで $minRTO$ は RTO の最小値，$SRTT$ は smoothed RTT，G は clock granularity，$RTTAVR$ は RTT vairation である．$minRTO$ は RFC6298[10] により，1 秒に設定することが推奨されている．多くの場合で（1）式の右辺では $minRTO > SRTT + max(G, 4 \times \ RTTAVR)$ が成り立つため，これ以降 RTO の初期値は $minRTO$ に設定されるものとして議論を進める．

$$RTO_1 = minRTO \quad (2)$$

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．ただし，RTO の値は 60 秒以上の上限値を持つように制限されている．i 回連続でタイムアウトした際の RTO の値は次の式で再設定される．

$$RTO_i = 2RTO_{i-1} \quad (3)$$

当該パケットの送信と応答が成功した場合，（2）式により RTO は $minRTO$ に再設定される．

このアルゴリズムは Karn のアルゴリズムと呼ばれ，ほとんどの TCP で実装されているが，RTO_i が $minRTO$ に依存して一意に決定されるという単純な仕様が LDoS/LDDoS 攻撃に利用されている．

2.2 LDoS 攻撃

LDoS 攻撃は TCP 再送信タイムアウトの脆弱性を利用し，標的サーバの送信トラフィックを妨害しクライアントとの通信を抑止する攻撃である [3]．図 1 のような短いバースト通信と無通信が一定の周期で繰り返される矩形波状の攻撃トラフィックを連続して送信することで攻撃を行う．攻撃トラフィックはバースト間隔 T，バースト長 L，バーストレート R の 3 つのパラメータにより形成される．LDoS 攻撃は，T を minRTO と等しい長さ，L を RTT 程度の長さ，R をボトルネックリンクのバッファを十分に満たす大きさに設定した場合に最も大きな被害が得られる．

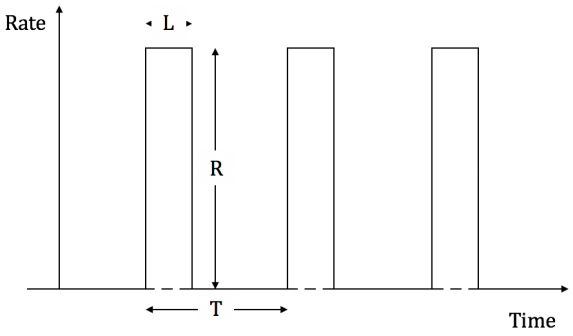

図 1　LDoS 攻撃のバーストトラフィック

攻撃者による 1 回目のバーストトラフィックにより，ボトルネックリンクのバッファが枯渇し正規の送信トラフィックにパケット喪失が発生することで送信側 TCP は再送信タイムアウトを待つ．（2）式により，minRTO だけ待ったあと通信に失敗したパケットの再送信を行う．しかし，このときも次のバーストトラフィックによって TCP 通信が再び失敗する．以後（3）式により，RTO の値が minRTO の倍数の値を取り続けながら再送信されるため，攻撃が続く間はバーストトラフィックと再送信のタイミングが重なり，通信が抑止された状態が継続される．

2.3 LDDoS 攻撃

LDDoS 攻撃は，LDoS 攻撃の攻撃トラフィックを複数の攻撃ノードから分散または増幅して行う攻撃手法である．攻撃トラフィックの分散・増幅の例を図 2 に示す．代表的な例として，バースト間隔，バースト長，バーストレートそれぞれについて分散・増幅する手法とそれらを組み合わせた手法が存在する．文献 [4] においてさらに詳細にモデル化がされている．バーストトラフィックの分散・増幅を行うことで，通常のトラフィックとの判別を困難にしたり，攻撃効果を高めることができる．

このように LDDoS 攻撃はインターネットの新たな脅威となる可能性があるが，今までに LDDoS 攻撃が実際に行われた事例は確認されていないため，実ネットワーク環境下でどの程度の被害を及ぼすのかは明らかになっていない．

3. 関連研究

LDDoS 攻撃は平均通信量の低いトラフィックの集合で形成されるため，既存の防御手段では対策するこが困難である．そのため，トラフィックの輻輳参加率をもとに LDDoS 攻撃を検出する手法 [4][5] や様々なエントロピーベースの検知手法 [6][7] が提案されている．これらの既存研究の検知手法の共通の課題として，実験に使用しているバーストトラフィックの分散手法が少ないことや，ネットワークトポロジが単純であることから，さらに複数の複雑

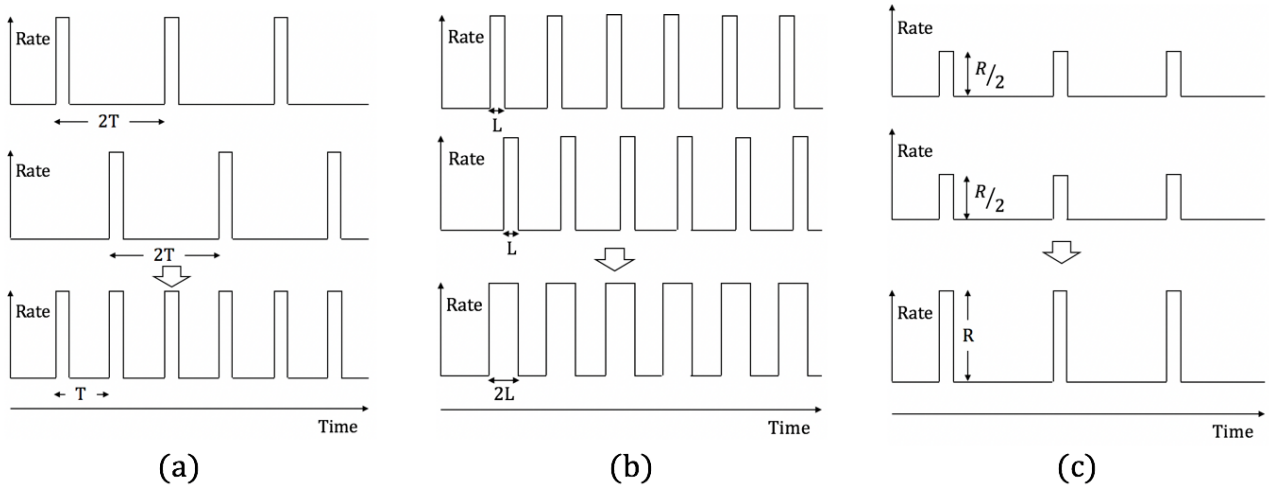

図 2 LDDoS 攻撃のバーストトラフィック（a）バースト間隔の分散，（b）バースト長の増幅，(c) バーストレートの分散　文献 [4] 図 3 を参考に作成

な条件下で評価を行い有効性を確かめる必要がある．これは LDDoS 攻撃がこれまで実際に行われたことがないため，検知の指標となる有効なデータセットが存在しないことが原因である．

4. 研究の目的

関連研究の課題から，本研究では実ネットワーク環境下にて効果的な LDDoS 攻撃を構成する際に必要な条件を明らかにし，その条件を用いて LDDoS 攻撃の効果的な検知・抑止の手段を確立することを目的とする．

5. アプローチ

関連研究 [4][5] における LDDoS 攻撃の評価は，以下のような条件がすべて成立しているという前提の下に行われている．

- ネットワークの接続状況によって決定する条件
  - (1) サーバとクライアントを含むネットワークトポロジは単純なダンベル型である
  - (2) サーバと同一ネットワーク内に攻撃ノードが接続されている（図 3）
- ネットワークの動作状況によって変化する条件
  - (3) 攻撃トラフィックによって溢れさせるべきボトルネックリンクの帯域幅が事前に判明している
  - (4) サーバの TCP 制御パラメータである minRTO が事前に判明している
  - (5) 攻撃ノード間でトラフィックを送出するタイミングを同期することができる

これらの条件が満たされることで攻撃者は効果的な LDDoS 攻撃を見込める．この中でネットワークの接続状況によって決定する条件を満たすことは，ダンベル型のネットワークトポロジが存在し，標的サーバのネットワーク内に

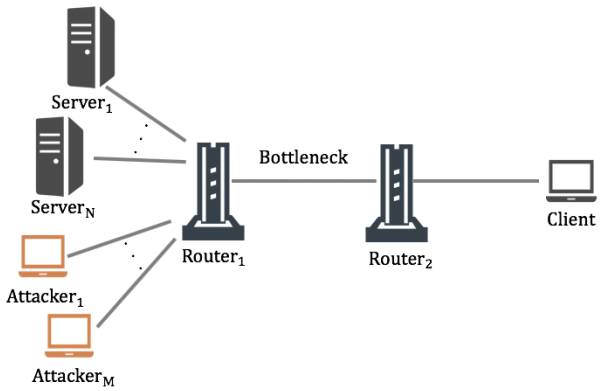

図 3 ダンベル型ネットワークトポロジ

接続されている IoT 機器などにマルウェアを感染させボット化することにより可能である．しかし，ネットワークの動作状況によって変化する条件はネットワークの利用状態や標的サーバの設定によって変化するため，攻撃者が状況に応じて個々の標的サーバごとに特定する必要がある．この前提条件を崩すことが可能であれば，既存研究に示されているようなトラフィックの分析に依らずに LDDoS 攻撃を抑止できる可能性があると考える．

そこで，本稿では以下の 3 つのアプローチで実ネットワーク環境下における LDDoS 攻撃の特性を明らかにしていく．

1 つ目は，IoT 機器を用いた LDDoS 攻撃の検証である．2020 年の IoT 機器の普及拡大 [2] に伴って事例 [1] のように大規模な IoT ボットネットが構築され，LDDoS 攻撃に利用される可能性が考えられる．しかし，関連研究で IoT 機器を攻撃ノードとして LDDoS 攻撃を構成した事例がないため，IoT 機器のようなリソースの限られているコンピュータでネットワークシミュレータや PC と同様にバーストトラフィックを生成できるのかは不明であることから，攻撃ノードに IoT 機器用いた際でも適切にバーストト

ラフィックを送信し集約することが可能であるかを示す．

2 つ目は，ルータの送信キュー容量が LDDoS 攻撃の効果へ及ぼす影響の検証である．LDDoS 攻撃は標的のボトルネックのバッファを溢れさせることで通信を妨害する手法であることから，本 IoT 機器を用いた LDDoS 攻撃の検証では，標的ルータの送信キューの大きさを既存研究 [4][5] を参考に 50 パケットに設定して検証を行う．しかし，Linux の送信キューの大きさの初期設定が 1000 パケットであることや，文献 [8] の図 5 から家庭用ルータの送信キューの大きさがおおよそ 1000 パケットであると読み取れることから，ルータの送信キューの大きさをより実ネットワーク環境下に近い環境に設定して検証を行う．

3 つ目は，標的サーバの minRTO を特定する手法の提案である．正規の通信を装って標的の minRTO を取得することが可能であることを検出手法の構成を以って指摘し，LDDoS 攻撃の実現性が高いことを示し今後の対策へ繋げる．関連研究や本稿で行った実験では，標的サーバの minRTO が RFC6298[10] で推奨されている 1 秒に設定されているものとして，集約後のバーストトラフィックのバースト間隔が 1 秒になるように LDDoS 攻撃を構成している．しかし，実際に minRTO が 1 秒に設定されているという保証はなく，Linux の初期設定でも minRTO が 0.8 秒に設定されている．このことから，そのままでは攻撃者が適切なバースト間隔で攻撃を行うことは難しいことが期待されるが，minRTO の取得が容易であればその限りではない．

6. IoT 機器を用いた LDDoS 攻撃の検証

6.1 実験環境

図 4 に使用するトポロジを示す．2 台のルータ（Router1，Router2）と 1 台のサーバ（Server）と 3 台の攻撃ノード（Attacker1 · · · 3）と 1 台のクライアント（Client）からなるダンベル型トポロジである．2 つのルータをつなぐリンクは Linux のトラフィックコントロールの Token Bucket Filter[11] を利用して，帯域幅が 10Mbps，RTT が 20ms のボトルネックリンクとして設定した．その他のリンクは帯域幅を 100Mbps，RTT を 1ms と設定した．Router1 の送信キューのサイズは [4] と [5] の実験を参考に 50 パケットに設定した．サーバの minRTO は RFC6298[10] で推奨値とされている 1 秒に設定し，輻輳制御アルゴリズムは CUBIC を用いた．1 パケットの大きさは 1514Byte である．攻撃ノード 3 台は Raspberry Pi 3 Model B，その他は PC を用いて環境を構築した．

実験は時刻 0 から 60 秒間サーバがクライアントに対し TCP パケットを帯域幅の速度で送り続け，攻撃ノードも時刻 0 から同時に 60 秒間バーストトラフィックを送り続けた．Server は Client に対して iPerf[12] によって TCP 通信で帯域幅の速度でデータを送信し，Attacker1 · · · 3 も Client に対して UDP でバーストトラフィックを送信してサーバとクライアント間の通信妨害を行った．

表 1　実験に使用したマシン

Entity	OS	CPU
Client	Ubuntu 18.04	Intel(R) Core(TM) i5 @3.60GHz
Server	Ubuntu 16.04	Intel(R) Core(TM) i5 @3.60GHz
Router1 & 2	Ubuntu 16.04	Intel(R) Core(TM) i5 @3.60GHz
Attacker1 · · · 3	Ubuntu Mate 16.04	ARMv7 Processor rev4

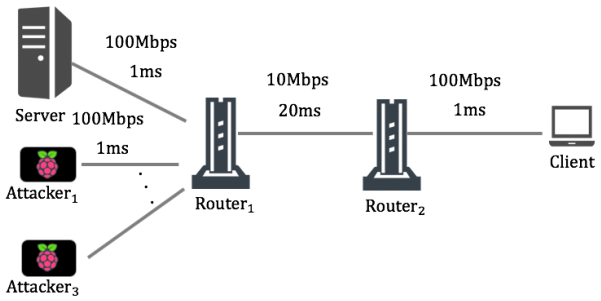

図 4　実験環境のトポロジ

バーストトラフィックは大きさが 50Byte の UDP パケットから構成した．Attacker1 · · · 3 によるバーストトラフィックは，集約後のバーストトラフィックのパラメータが R=10Mbps，L=300ms，T=1000ms となるように R と L について 2 通りの手法によって分散を行った．バースト間隔 L を 3 秒間隔で分散した手法（手法 1）とバーストレート R を 3.5Mbps ずつ分散した手法（手法 2）の 2 つを用いて検証を行った．手法 1 のバーストトラフィックのパラメータは，すべての攻撃ノードで R=10Mbps，L=300ms，T=3000ms と設定し，Attacker2 の攻撃開始時間を 1 秒，Attacker3 の攻撃開始時刻を 2 秒遅らせた．手法 2 のバーストトラフィックのパラメータはすべての攻撃ノードで R=3.4Mbps，L=300ms，T=1000ms と設定し，同時に攻撃を開始した．

6.2 結果と考察

図 7 に Router1 で取得したパケットキャプチャデータを図示してバーストトラフィックを示す．この図から手法 1 と手法 2 のどちらを用いても意図したとおりのバーストトラフィックを生成し，適切な時間間隔で集約できていることがわかる．攻撃中におけるサーバの平均正規化スループットは，手法 1 が 165Kbps，手法 2 で 483Kbps と両方の手法で 9 割以上の通信を妨害する結果となった．手法 1 を用いた攻撃中におけるサーバの正規化スループットの遷移を図 5，手法 2 を用いた攻撃中におけるサーバの正規化スループットの遷移を図 6 に示す．この結果から，PC に比べてリソースの少ない IoT 機器からでも LDDoS 攻撃が可能なことがわかった．また，手法 1 の方が手法 2 に比べてわずかではあるが攻撃効果が高いことから，手法 1 のよ

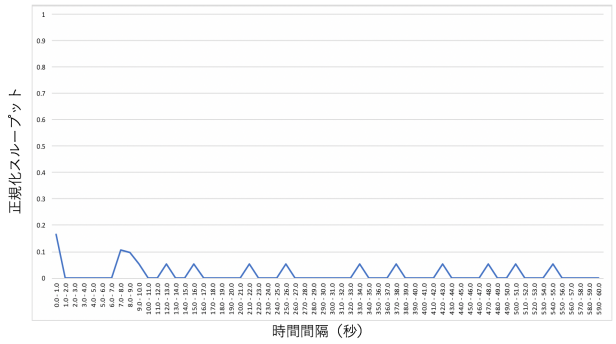

図 5 手法 1 を用いた攻撃中におけるサーバの正規化スループットの遷移

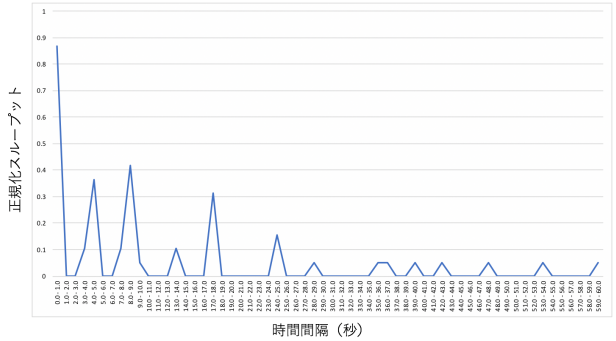

図 6 手法 2 を用いた攻撃中におけるサーバの正規化スループットの遷移

うに攻撃ノード 1 台あたりのバーストレートが高い方がボトルネックリンクのバッファを早く満たすことができる可能性があると考える．

今回は各攻撃ノードのネットワークの条件が同じだったため，バーストトラフィックの正確な集約が行えた．しかし，実際にこのようなネットワークトポロジの攻撃シナリオを考えると，標的サーバと同一ネットワーク内に踏み台を構築しなければならないという制限があるため，今後の課題として遅延や帯域幅の異なる別々のネットワークに攻撃ノードを分散させた状態でも任意のボトルネックリンクで正確にバーストトラフィックの集約を行えるのかを検証する必要がある．

7. ルータの送信キュー容量が LDDoS 攻撃の効果へ及ぼす影響の検証

7.1 実験環境

トポロジは基礎実験で使用したものと同様のものを使用した．バーストトラフィックは 6 章の実験において，手法 2 と比べてわずかに攻撃効果の高かった手法 1 と同じものを使用し，Router1 の送信キューの大きさを 50 パケットから 1000 パケットまで 50 パケットずつ増加させて，サーバが 60 秒間通信した際の平均正規化スループットの変化を観察した．

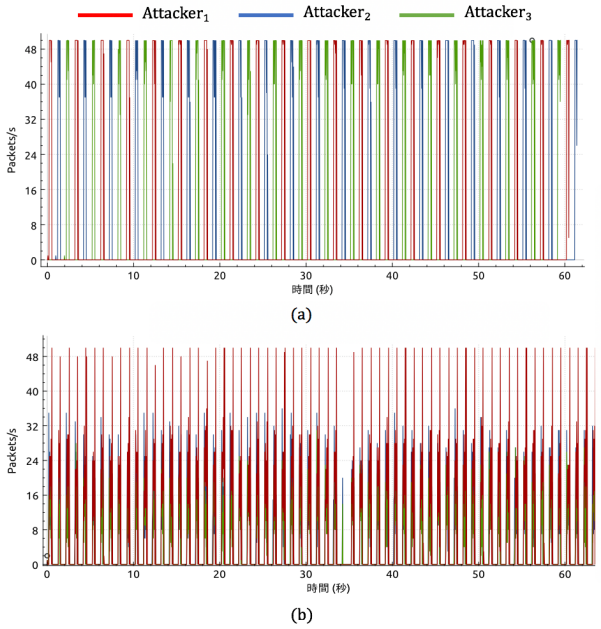

図 7 Router1 でキャプチャしたバーストトラフィック (a) 手法 1 (b) 手法 2

7.2 結果と考察

図 8 に結果を示す．500 パケットまでは平均正規化スループットが 0.6 まで大きく上昇しているが，500 パケットから 1000 パケットの間はほとんど上昇しないという結果が得られた．1000 パケットでの平均正規化スループットは約 7 割出ており，この状態ではパケットロスは発生するものの，問題なく通信はできていると窺える．このことから，一般的な家庭用ルータ程度のスペックのルータを LDDoS 攻撃の対象にする際に送信キューの容量が 50 パケット程度のルータと同様の攻撃効果を発揮するためには，さらに複数の攻撃ノードを増やして，集約後のバーストトラフィックの強さを増幅する必要がある．

そこで，送信キュー容量が 1000 パケットのルータに対して効果的な LDDoS 攻撃の構成に必要な攻撃ノード数の見積もりを行った．見積もりには 6 章の手法 1 のバーストトラフィックをバースト長について増幅させた場合を再現するために 1 台の攻撃ノードで R=10Mbps，T=1000ms を固定し，バースト長を L=600ms，700ms，800ms，900ms，1000ms と 100ms ずつ増加させた DoS バーストトラフィックを送信し，6 章と同様の方法でサーバとクライアントの平均正規化スループットを計測した．結果を図 9 に示す．この結果から L=900ms であれば平均正規化スループットを 1 割以下に落とせることがわかった．R=10Mbps，T=3000ms，L=300ms と設定し，集約後のバーストトラフィックが R=10Mbps，T=1000ms，L=300ms となるように設定した手法 1 を用いて，集約後のバーストトラフィックが R=10Mbps，T=1000ms，L=900ms となるように分散を行うためには L を 3 倍増幅させる必要がある．よって必要な攻撃ノード数も 3 倍必要となるため，この手法で

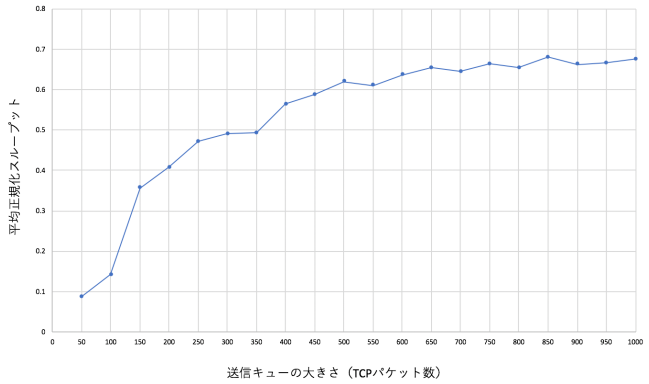

図 8　Router1 の送信キューの大きさとサーバの平均正規化スループット

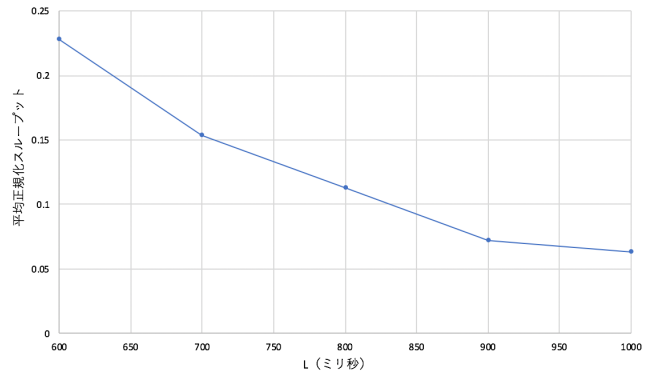

図 9　バースト長 L の増幅によるサーバの平均正規化スループットの変化

バーストトラフィックの増幅を行った場合，最低でも 9 台の攻撃ノードが必要になると推測できる．今回の実験では十分な攻撃ノードの台数を確保できなかったことから実際に 9 台の攻撃ノードで増幅した実験を行えなかったが，6 章の実験結果からバーストトラフィックを適切なタイミングで集約できることは示したため，実際に 9 台で増幅させた際にも同等の効果が期待できると考えられる．

以上の結果から，既存研究 [4][5] ではボトルネックリンクのバッファの大きさに対して最小限の大きさのバーストトラフィックを検知する試みが行われているが，今後は深刻な被害をもたらす強さの LDDoS 攻撃に着目して検知を行うアプローチも検討できれば良いと考える．

8. 標的サーバの minRTO を特定する手法の提案

TCP ではサーバが送信したパケットに対する ACK パケットがクライアントから返ってこない場合，2.1 節で解説した再送信タイムアウトだけ待ってから同じパケットを再送信するが，この仕組みを利用することで任意のサーバの minRTO を取得することが可能である．つまり正規の利用者を装いサーバと通信を行いつつ，意図的にサーバから送信された TCP セグメントに対する ACK を送信しないことで RTO を引き起し，その挙動を観察することで minRTO を推定する．

今回の実験で用いた機材を流用して上記手法をアドホックに構成する例を以下に示す．minRTO の特定は攻撃者が図 10 のような任意のネットワークに正規ユーザを装って通信するクライアント（Decoy Client）とルータ（Decoy Router）と攻撃用の端末（Attacker）を用意することで可能である．まず，Decoy Client が標的の TCP サーバに対して正規のリクエストで TCP パケットの受信を開始する．この間，Decoy Client は Victim Server に対して ACK パケットを送信し続ける．ここで，Attacker が Decoy Router の送信キューを意図的に溢れさせることで，Victim Server から Decoy Client に送信されたパケットまたは，Decoy Client から Victim Server に向けて送信した ACK パケットのどちらかを意図的に廃棄させる．これにより，Victim Server は該当するパケットの再送信を行うため，Decoy ネットワーク内でパケットキャプチャを行うことにより，意図的に廃棄したパケットとそのパケットが再送信された時間を特定し，2 つのパケットの時間差から minRTO を計算することができる．このとき Decoy Client と Victim Server 間の RTT によって正確な minRTO が計算できない可能性があるが，複数回特定を重ねて算出した平均値と平均 RTT の差を求めることでより正確な値を計算できると予想されるため，今後検討を行う．

この手法は Decoy ネットワーク内のみで攻撃を行うため，外部のネットワークからは通常のパケット廃棄が発生したようにしか見えず悪意を持った挙動の観察が行われていることを隠蔽できる．よって攻撃者は標的に検知されることなく minRTO の値を特定することが可能であり，効果的なパラメータで LDDoS 攻撃を構成することが可能になる．

具体的な対策は検討中であるが，この手法を利用して効果的なパラメータでバーストトラフィックが送信される可能性があることを考慮しておく必要がある．

関連研究 [9] において，連続した RTO の値の再計算の際に (3) 式の代わりに $RTO_i = (1+u)^{(i-1)} minRTO for (0 < u < 1)$ とすることにより，RTO_i の値が minRTO の整数倍にならないことから minRTO が特定されていても攻撃を緩和する手法が提案されている．このような手法と本提案手法の位置付けについても今後検討していく．

9. おわりに

本稿では，実ネットワーク環境下において LDDoS 攻撃を構成・評価するために 5 章で述べた LDDoS 攻撃の構成を可能にする前提条件リストに示した条件下で検証を行い，実際に攻撃ノードを IoT 機器である Raspberry Pi を用いて，標的のキューサイズが既存研究に示された 50 に

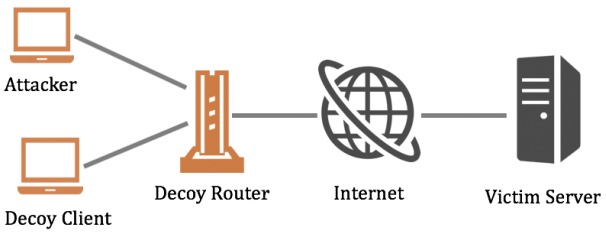

図 10　minRTO を特定するためのネットワークトポロジ

ついては実験で，現実的と思われる 1000 については外挿にて必要な攻撃ノード数の見積りを行い LDDoS 攻撃が実現可能であることを示した．今回列挙した LDDoS を成立させる条件の一つである既知の minRTO については現実的には成り立たないたが，意図的にサーバから受信するパケットを廃棄することで検知されることなく minRTO を検出する手法を提案し簡易な実現例も示した．

今後の課題として，既存研究では検証されていない遅延や帯域幅の異なる別々のネットワークに攻撃ノードを分散させたトポロジ上においても効果的な LDDoS 攻撃の構成が可能であるかを検証した上で，攻撃に必要な条件を明らかにし，効果的な LDDoS 攻撃の抑止手段を確立していく．また，今回提案した標的サーバの minRTO を特定手法によって minRTO を特定されないための対策手法も検討していく．

参考文献

[1] JPCERT コーディネーションセンターほか:Mirai 等のマルウェアで構築されたボットネットによる DDoS 攻撃の脅威, Japan Vulnerability Notes（オンライン）, 入手先〈https://jvn.jp/ta/JVNTA95530271/index.html〉(参照 2018-07-30).

[2] Rob van der Meulen:Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016, Gartner(オンライン), 入手先〈https://www.gartner.com/newsroom/id/3598917〉(参照 2018-07-30).

[3] A. Kuzmanovic et al:Low-rate TCP-targeted Denial of Service Attacks and Counter Strategies, IEEE/ACM Transactions on Networking, Vol.14, No.4，pp.683-696，2006.

[4] Zhang et al:Flow level detection and filtering of low-rate DDoS, Computer Networks, Vol.56, No.15, pp.3417-3431, 2012.

[5] Kieu et al:Using CPR Metric to Detect and Filter Low-Rate DDoS Flows, Proceedings of the Eighth International Symposium on Information and Communication Technology. ACM, pp.325-332, 2017.

[6] P. N.Jadhav and B. M. Patil: Low-rate DDOS Attack Detection using Optimal Objective Entropy Method, International Journal of Computer Applications，Vol.78, No.3, pp.33–38, 2013.

[7] Y. Xiang, K. Li, and W. Zhou:Low-rate DDoS attacks detection and traceback by using new information metrics，IEEE Transactions on Information Forensics and Security，Vol.6，No.2，pp.426–437，2011.

[8] 山本ら:深層学習を用いた無線 LAN パケット解析に基づく輻輳の予測，マルチメディア，分散，協調とモバイル (DICOMO2018) シンポジウム，pp.1772-1769，2018.

[9] 細井 琢朗，松浦 幹太: TCP 再送信タイマ管理の変更による低量 DoS 攻撃被害の緩和効果，コンピュータセキュリティシンポジウム 2013 論文集，Vol.2013，No.4，pp.957-964，2013.

[10] V. Paxson et al:Computing TCP's Retransmission Timer, Internet RFC 6298(オンライン), 入手先〈https://tools.ietf.org/html/rfc6298〉(参照 2018-07-30).

[11] Alexey N. Kuznetsov: tc-tbf (8), Linux Man Pages(オンライン), 入手先〈https://www.systutorials.com/docs/linux/man/8-tc-tbf/〉(参照 2018-10-18).

[12] iPerf - The ultimate speed test tool for TCP, UDP and SCTP, 入手先〈https://iperf.fr/〉(参照 2018-10-18).