情報処理学会論文誌 Vol.59 No.9 1656–1665 (Sep. 2018)

拡散型フロー制御を用いる自律分散的なDDoS攻撃緩和システム

平 空也[1] 高野 知佐[1,a)] 前田 香織[1]

受付日 2017年12月11日, 採録日 2018年6月8日

概要：DDoS（Distributed Denial of Service）攻撃の中でもインフラストラクチャレイヤへの攻撃は規模が大きく，機器の処理やネットワークの帯域に大きな負荷を与える．また，常時通信が必要なサービスが増加する超スマート社会では，DDoS 攻撃によるサービス中断の影響はより大きくなる．そこで，本研究では DDoS 攻撃検知から対策適用までの期間の DDoS 攻撃の被害を小さくするため，この期間もサービスが継続できるように正常な通信のパケット損失を防ぐ DDoS 攻撃緩和システムを提案する．提案システムはネットワーク基盤上で自律分散制御である拡散型フロー制御を応用したトラヒック制御を行うことによって，DDoS 攻撃の複数の攻撃元から攻撃対象となるサーバ向きの大量のトラヒックを緩和し，正常通信を継続できるようにする．本稿では，トラヒックの緩和のためのアルゴリズムと提案システムの設計を述べ，シミュレーション実験により提案システムの緩和性能の評価を示す．

キーワード：セキュリティ，DDoS 攻撃緩和，拡散型フロー制御，自律分散制御

Autonomous Decentralized DDoS Mitigation System Based on Diffusion Flow Control

Kuya Taira[1] Chisa Takano[1,a)] Kaori Maeda[1]

Received: December 11, 2017, Accepted: June 8, 2018

Abstract: Infrastructure layer attacks of DDoS (Distributed Denial of Service) damage on networks and servers that support essential communication over the Internet on a large-scale. These attacks use a lot of bandwidth of networks and serious load on the servers. Also, always-connected services required in a super smart society receives more serious impact by service interruption due to DDoS. In this paper, we propose a DDoS mitigation system to avoid packet losses of regular communications during the period from detection of a DDoS attack to measures against it in order to reduce the damage to the continuous service. The proposal system can mitigate a large amount of traffic from multiple attack sources of DDoS to a target server by applying the diffusion-type autonomous decentralized flow control. In this paper, we describe the mitigation algorithm of a large amount of traffic flows and the design of the autonomous decentralized mitigation system using this algorithm. Also, we show some evaluations on mitigation effect of the proposed system by simulation experiments.

Keywords: Security, DDoS mitigation, Diffusion Flow Control, Autonomous Decentralized Control

1. はじめに

コンテキストアウェアなサービスの提供や消費者生活の利便性向上のため，IoT 機器でのデータ収集や消費者の要求に対してきめ細かな対応を行うクラウドサービスなど常時通信を行うサービスが創出され，周りのあらゆるモノが高機能化（スマート化）した超スマート社会 [1] が今後到来するといわれている．インターネットにつながる家電やセンサなど IoT 機器の増加とともに，すでに IoT 機器をターゲットにした Mirai[*1]のようなマルウェアの出現が影響して DDoS（Distributed Denial of Service）攻撃が問題となっている．今後，IoT 機器の爆発的な増加にともなって DDoS 攻撃は業務停止にもつながる深刻さを増す一方，消費者が安心安全にスマートな生活をおくるためには，こ

[1] 広島市立大学大学院情報科学研究科
Graduate School of Information Science, Hiroshima City University, Hiroshima 731–3149, Japan
[a)] takano@hiroshima-cu.ac.jp

[*1] http://jvn.jp/ta/JVNTA95530271/

のような脅威に対する早急な対策の検討が必要である．

DDoS 攻撃は主にアプリケーションレイヤへの攻撃とインフラストラクチャレイヤへの攻撃に分類される．前者は DDoS 攻撃の中では比較的規模が小さく，WAF（Web Application Firewall）や IPS（Intrusion Prevention System），DDoS 攻撃緩和装置（以下，緩和装置）などで緩和が可能である．一方，後者はリフレクション攻撃のような規模が大きな攻撃（以下，量的攻撃）であり，300 Gbps を超える攻撃も報告されている [2]．文献 [3] には DDoS 攻撃の約 65%が大量にパケットを送信する量的攻撃で，特にリフレクション攻撃が増加傾向にあることが述べられている．また文献 [4] には 2017 年度第 1 四半期に DDoS 攻撃の 99%がインフラへの攻撃であったこと，その DDoS 攻撃の 7 割以上がリフレクション攻撃であったことが述べられている．量的攻撃は機器の処理やネットワークの帯域に大きな負荷を与えるため，個人や企業が持つセキュリティ機器による LAN 内での対策は困難である．そのため，BGP Flowspec [5] や AS 間で連携して複数 AS の緩和装置を利用する手法 [6] など事業者間で連携することで早期に対策する手法も提案されている．

DDoS 攻撃の被害を抑えるためには，DDoS 攻撃の検知と対策適用の早さが重要になる．DDoS 攻撃の検知手法として実際に用いられているものには人の手によるエッジルータのトラヒック監視 [7] や SAMURAI [8], [9] を用いたトラヒックおよびフロー監視による検知がある．また，DDoS 攻撃の対策手法として実際に用いられている方法にはパケットの宛先 IP アドレスでフィルタリングするブラックホールルーティングや攻撃パケットのみをフィルタリングする緩和装置による緩和 [2] がある．これらの DDoS 攻撃対策はユーザの正常の通信で発生するパケット（以下，正規パケット）の損失を防ぎ，また DDoS 攻撃検知が早いほど対策の適用も早くなるため正規パケットの損失が減少する．しかし，DDoS 攻撃検知後から対策の適用までの期間は DDoS 攻撃の被害を被るため正規パケットを損失し，現状これを防ぐ方法はない．

本研究では，サービスのパケット損失を防ぐために，DDoS 攻撃検知後からの正規パケットの損失を防ぐシステムを提案 [10], [11] し，その有効性を評価する．超スマート社会では様々な IoT デバイスおよびサービスから創出される多種多様なデータがネットワーク上を流れることになるが，提案システムではネットワーク全体の状態情報を収集することなく局所情報を用いた自律分散制御によって正規パケットをなるべく損失させず，悪意ある DDoS 攻撃を緩和することができる点が既存のアプローチと異なる．

本稿の構成は以下のとおりである．2 章で関連研究について，3 章で提案する DDoS 攻撃緩和システムについて，4 章で提案システムの評価について述べる．最後に 5 章で本稿のまとめと今後の課題について述べる．

2. 関連研究

2.1 DDoS 攻撃の検知・対策

DDoS 攻撃の検知方法として実際に用いられているものの多くは人による判断か閾値による検知である．文献 [12] は Snort[*2] という IDS（Intrusion Detection System）を使用し，通過するパケットがシグネチャと呼ばれるルールファイルに定義されるものであればログを作成して，収集したログを解析することで DDoS 攻撃の検知を行うものである．文献 [13] は OpenFlow を用いてパケットの到着レートを調べ，到着レートの平均二乗誤差の上昇率と閾値を比較することで検知を行うものである．しかし，これらの手法では DDoS 攻撃を検知するまでに時間がかかり被害が大きくなるため，被害を抑えるべく早期検知を目的とし，複数のネットワークサービスプロバイダ（ISP）間で対策を講じる提案 [14] もなされている．文献 [14] では，任意の ISP が閾値を超えるトラヒックや平常時と異なるトラヒックを観測した場合，近隣の ISP に通知を出すことで DDoS 攻撃の早期検知を可能としている．

また DDoS 攻撃の対策方法として実際に用いられているものには，ブラックホールルーティングやアクセスフィルタによる遮断 [2] がある．しかし，これらの遮断は攻撃パケットとともに正規パケットも落としてしまう可能性がある．そのほか，緩和装置による攻撃トラヒックのみの遮断 [15] があるが，単一の緩和装置では量的攻撃が来た場合に過負荷で処理できないという問題がある．この問題を解決するために，文献 [6] では，AS 間で緩和装置リソースを融通し合うことで量的攻撃による負荷を分散させ，攻撃パケットのみを遮断している．しかし，送信元ごとに対処するような閾値フィルタリングや緩和装置は IP スプーフィングと呼ばれる送信元を偽装する攻撃の遮断が困難である．そのため文献 [15] では，TTL の値の減少数が偽装できないことを利用し，攻撃元から標的サーバまでの中間ルータで送信元 IP アドレスごとの TTL の値が正しいかどうかを確認してフィルタリングを行う．

このように既存研究では DDoS 攻撃の検知や対策に関する有効な提案がなされているが，検知から対策までの正常な通信への被害は回避できない．提案するシステムはネットワーク監視機器による DDoS 攻撃の検知や，緩和装置による攻撃トラヒックの遮断といった既存の検知・対策方法と提案するトラヒック制御を組み合わせることで正常な通信が中断されないようにする．

2.2 拡散型フロー制御

拡散型フロー制御 [16] とは，物理学における拡散現象を指導原理とし，ネットワークの輻輳回避を目的とした自

[*2] https://www.snort.org/

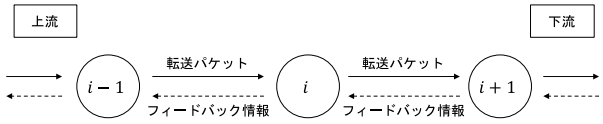


図 1 ノードの動作モデル
Fig. 1 Node behavior model.

律分散型フロー制御である．エンドホスト間で経由されるネットワーク機器（以下，ノード）が隣接するノードとの相互作用のみで自律分散的に転送レートの制御を行い，バッファ使用量の平滑化を実現する．

ノードの動作モデルを**図 1** に示す．送信元エンドホスト側を上流，宛先エンドホスト側を下流とすると拡散型フロー制御におけるノードの動作は以下の 4 つである．

- 上流ノードへ自ノードのバッファ使用量とパケットの転送レートを含んだフィードバック情報を送信
- 下流ノードから送られてきたフィードバック情報を受け取り，その情報と自ノードのバッファ使用量をもとにパケットの転送レートを計算
- 上流ノードから送られてきた宛先宛のパケットを受け取りバッファリング
- 下流ノードへ自ノードの転送レートに従って宛先宛パケットを転送

拡散型フロー制御方式による転送レートの算出方法について説明する．拡散型フロー制御方式では，ノード i のバッファ使用量 $n_i(t)$ と下流ノードのバッファ使用量 $n_{i+1}(t-d_i)$ の差に応じて，フィードバック情報から求める転送レート $\tilde{J}_i(t)$ は式 (1) となる．

$$\tilde{J}_i(t) = r_i(t-d_i) - D_i(n_{i+1}(t-d_i) - n_i(t)) \qquad (1)$$

ただし，下流ノードから送られてきた転送レート $r_i(t-d_i)$ をフィードバック情報として用い，d_i はノード i とノード $i+1$ 間の伝搬遅延時間，D_i は拡散係数である．つまり，ノード i よりも下流ノード $i+1$ の方が混雑している（$n_{i+1}(t-d_i) > n_i(t)$）場合，転送レートを抑える方向に，一方下流ノードの方が空いている（$n_{i+1}(t-d_i) < n_i(t)$）場合は転送レートを上げるといったブレーキ/アクセルの役割を果たす．ただし，転送レートは非負の値であるということ，また使用できる帯域以下でなければならないことから，実際に利用される転送レート $J_i(t)$ は式 (2) のようになる．

$$J_i(t) = \max(0, \min(\tilde{J}_i(t), L_i(t))) \qquad (2)$$

ここで，$L_i(t)$ はノード i とノード $i+1$ 間の帯域を制御中のフロー数で割ったものであり，フロー単位で使用できる帯域を示している．本稿ではこれを使用可能帯域と呼ぶことにする．実装上，各ノードがフロー数をどのように取得するのかという問題については文献 [17] に示されている．本文献の技術によって各ノードは制御中のフロー数の情報を自律的に取得することが可能である．また，拡散係数 D_i は式 (3) で表される．上流ノード数とは上流側の隣接ノード数であり，以降も同様とする．

$$D_i = \frac{1}{\text{下流ノード } i+1 \text{ が持つ上流ノード数} + 1} \qquad (3)$$

既存の拡散型フロー制御方式は，送信元から宛先までの経路が確定したフローを対象に制御を行う．つまり既存の拡散型フロー制御方式は，一次元空間における拡散を目指したもので，二次元トポロジ空間を拡散する効果はない．この場合，拡散の効果を得るためには，拡散係数は $0 \le D_i \le 1/2$ の範囲でなければならない．物理現象としての拡散は連続媒体中，連続時間で起こるが，実際のネットワークは離散的（ノードの空間的配置や制御動作のタイミングが離散的）であり，上記に示した D_i の範囲は離散空間，離散時間の差分計算で求める際に必要な制約条件である．また，D_i が大きいほど拡散する速度が高くなるため，制約条件を考慮すると $D_i = 1/2$ のとき最も早く拡散する．本稿では式 (3) のように D_i を定義しているが，エンドホスト間で経由するルータをノードとした一次元トポロジでは各ノードの上流ノード数は 1 であるため $D_i = 1/2$ となる．

図 1 の一次元トポロジ（$D_i = 1/2$）における転送レートの計算例を示す．時刻 $t-d_i$ でのノード $i+1$ の転送レートが 4 pps，バッファ使用量が 10 packet であり，時刻 t でノード i のバッファ使用量が 6 packet，使用可能帯域が 10 pps である場合，フィードバック情報から算出されるノード i の転送レート $\tilde{J}_i(t)$ は式 (1) より

$$\tilde{J}_i(t) = 4 - 0.5 * (10 - 6) = 2$$

となる．また，式 (2) の制限によって実際に利用される転送レート $J_i(t)$ は 2 pps となる（この場合，$J_i(t) = \tilde{J}_i(t)$）．一方，ネットワークの輻輳が原因でノード i の使用可能帯域が小さく 1 pps の場合，2 pps でパケットを送信したくても帯域が足りないため，使用可能帯域の上限 1 pps が転送レートとなる．このように拡散型フロー制御方式によって算出される転送レートは，自ノードと下流ノードとのバッファ使用量の差を縮め，ネットワーク全体でバッファ使用量を平滑化する効果がある．

本研究では，提案する DDoS 攻撃緩和手法に拡散型フロー制御 [16] を応用し，ネットワーク全体の状況を知ることなく，ノードの局所情報に基づく自律動作により，ネットワーク全体の性能を制御する．ただし，インターネットの DDoS 攻撃の緩和に使えるように，3 章で既存拡散型フロー制御方式を二次元トポロジに拡張する．

3. DDoS 攻撃緩和システム

3.1 システムの概要

提案する DDoS 攻撃緩和システム [10], [11] を用いた DDoS 攻撃緩和では，DDoS 攻撃検知後からの正規パケッ

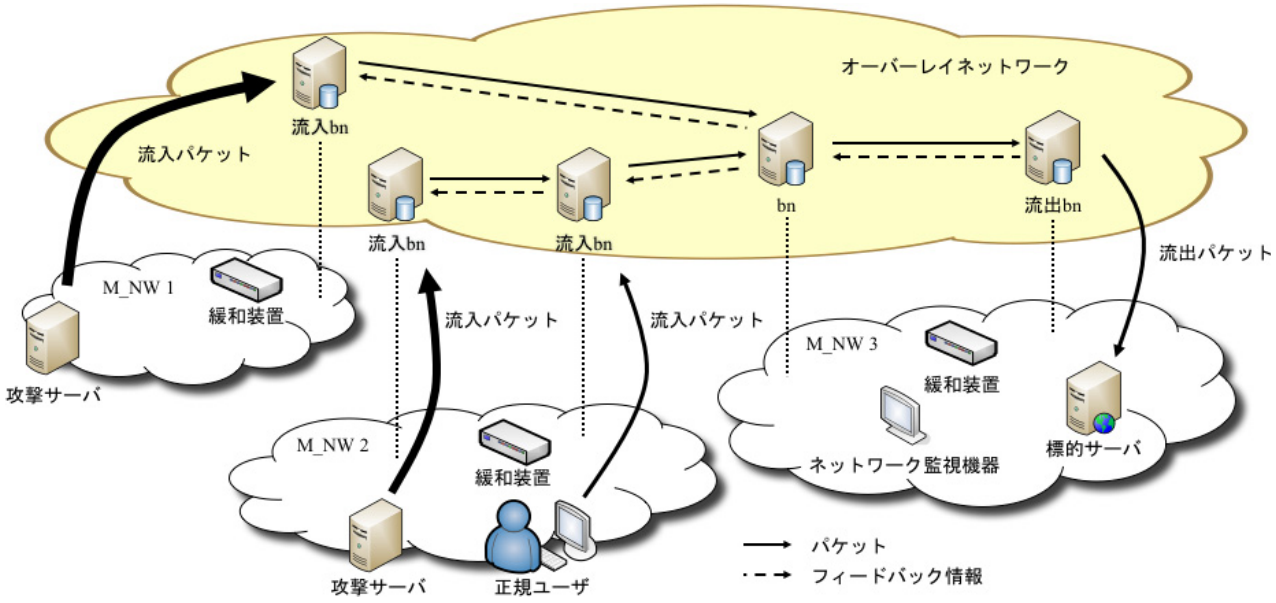


図 2 提案システムの全体構成
Fig. 2 Overall configuration of the proposed system.

トの損失を防ぐことを目的とする．そのため，適用する対策は攻撃パケットのみを遮断するものでなければならない．提案システム中の対策には文献 [18] のような振舞い検知やトラヒックの使用帯域などから攻撃トラヒックを特定する既存の緩和装置を用いてフィルタリングルールを生成し，攻撃パケットのみの遮断を行うものとする．また，DDoS 攻撃トラヒックは標的サーバ側である下流で合流し，輻輳やネットワーク機器の過負荷，緩和装置のリソース不足が生じる可能性が高いので，対策は文献 [6] のように DDoS 攻撃の上流（攻撃元側）にある緩和装置を用いて緩和を行うことで DDoS 攻撃トラヒックが下流（標的サーバ側）に流れることを防ぐものとする．

このような対策を講じることを前提とし，提案システムは下流での輻輳を回避するために，DDoS 攻撃緩和の対象となるネットワーク（以降，緩和ネットワーク）が連携して標的サーバ宛のトラヒックの転送レートを制御し，正規パケットの損失を防ぐものである．緩和ネットワークで DDoS 攻撃トラヒックの転送レートを制御するために，提案システムでは大きなバッファ容量を持つ専用サーバを設置し，専用サーバに DDoS 攻撃トラヒックを経由させる．このとき，専用サーバの転送レートを超えた DDoS 攻撃トラヒックのパケットはバッファリングされるが，任意の専用サーバにバッファが偏った場合バッファ溢れを引き起こす．そのため，専用サーバのバッファの偏りを防ぐために，転送レートを制御してバッファ使用率の平滑化を行い，バッファを効率的に使用する．このトラヒックの転送レート制御を以降ミチゲーションと呼び，提案するシステムではこれに拡散型フロー制御を応用する．

本システムでは複数の ISP のネットワークで連携することで，攻撃元により近いところでバッファリングを開始したりバッファリング箇所を増やしたりすることができるので，よりミチゲーションの効果を高めることが期待できる．

3.2 システムの構成

提案システムの全体構成を図 **2** に示す．提案システムは各緩和ネットワーク（以下，M_NW）でミチゲーションを行うバッファリングノード（以下，bn），常時ネットワークを監視し，DDoS 攻撃を検知するネットワーク監視機器，bn と連携してトラヒックを学習し，フィルタリングルールを生成する緩和装置から構成される．各 M_NW の bn どうしはオーバレイネットワークを構成しており，隣接する bn 間で連携して下流に転送するパケットの転送レートを制御する．このオーバレイネットワークの bn のうち特に，送信元からのパケットを受信する bn を流入 bn，最下流に位置する標的サーバにパケットを転送する bn を流出 bn と呼ぶことにする．

bn は拡散型フロー制御の 4 つの動作のほかに，「上流の bn へミチゲーションの要求」，「ルータへ経路変更の要求」を行う．ネットワーク監視機器もしくは下流の bn からミチゲーションの要求が来た場合にこれらの 2 つの動作を行うことで，DDoS 攻撃検知後に下流から bn を起動していくことができる．また，bn は M_NW の境界に配置することを想定しており，攻撃元から標的サーバ間のすべてのルータ間に必ず bn を設置する必要はない．標的サーバ以外へのトラヒックの影響を最小限にするために，図 **3** のように bn を配置して DDoS 攻撃の検知後に標的サーバ宛のパケットのみを bn に誘導する．

ここで，提案システムが対象としている DDoS 攻撃緩

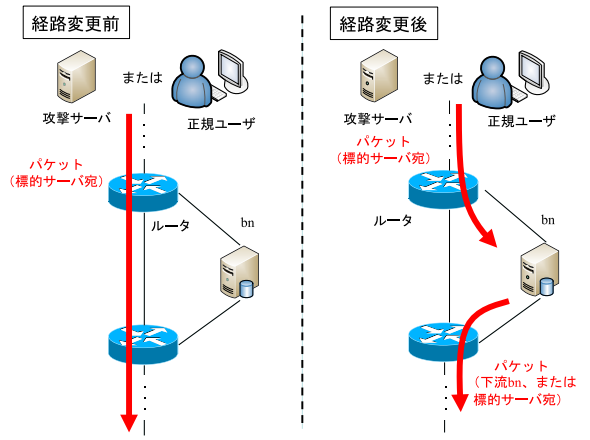


図 3 経路変更前後のパケットの流れ
Fig. 3 Packet flow before and after change of routing.

和が可能なネットワークの規模について簡単に触れておく．文献 [19] では，DDoS 攻撃を 1 秒で検知，30 秒以内に対策を開始できることが示されている．目安として標的サーバ宛の総トラヒックが 100 Gbps の DDoS 攻撃の場合に DDoS 攻撃の検知後 30 秒間のパケット損失を防ぐためには，バッファ使用率に偏りがない理想的なミチゲーションが行えた場合，4 GByte のバッファ容量を持つ bn が 100 台 ($\cong 30\,[\mathrm{sec}] \times 100\,[\mathrm{Gbps}]/(4 \times 8\,[\mathrm{Gbit}])$) 程度必要となる．

3.3 システムの動作

提案システムの動作は「DDoS 攻撃検知段階」,「ミチゲーション段階」,「フィルタリング段階」の 3 段階に分けることができる．各段階の動作の詳細について以下で述べる．

「DDoS 攻撃検知段階」では，ネットワーク監視機器が異常トラヒックを検知した場合，流出 bn にミチゲーションを要求する通知（以下，ミチゲーション通知）を送信し，ミチゲーション通知を受け取った bn は，ミチゲーションを開始するとともに，上流の bn へミチゲーション通知の送信とルータへ経路変更の要求を行う．このとき，各 bn は標的サーバ宛のパケットの送信元アドレスをもとに，上流の bn へミチゲーション通知を送るか自身が最上流の bn であるかを判断できる情報を持っているものとする．

「ミチゲーション段階」では，各 bn が隣接する bn と連携して転送レートを制御し，下流にパケットを転送する．また，緩和装置と連携し，フィルタリングルールの生成も同時に行う．

「フィルタリング段階」では，流入 bn と連携した緩和装置が生成したフィルタリングルールを用いて bn が攻撃パケットのみを遮断する．また，フィルタリングルールを適用した bn は下流の bn にフィルタリングルールを通知することで下流の bn にバッファリングされた攻撃パケットもフィルタリングできる．

以上の 3 段階より，より攻撃元の近くから DDoS 攻撃の緩和を行うことでインターネット規模での DDoS 攻撃緩和が可能となる．「ミチゲーション段階」で，“正規あるいは攻撃か，区別のつかないパケット” のバッファ溢れを回避し，サーバに到達させることにより，3.1 節で述べた目的の 1 つ「正規パケットの損失を防ぐこと」を達成する．このとき，バッファ容量や bn 台数の増加によって設計コストがかかるが，提案システムによるバッファ使用率の平滑化は，通信路上のどこかのバッファで起こる輻輳を回避し，結果，設計コストの低減につながると考えている．一方，「フィルタリング段階」により 3.1 節で述べたもう 1 つの目的「攻撃パケットのみを遮断する」を達成する．ただし，本提案システムでは「フィルタリング段階」について，攻撃パケットのみをフィルタリングすることができる既存の緩和装置を用いることを想定しているため，フィルタリングルール作成などの具体的な対策については本研究の対象外である．

3.4 転送レート算出式の拡張

式 (1) はノードが直列につながった一次元のトポロジ（フローごとの制御）を想定し，かつバッファ使用量の平滑化を目的とした転送レート算出式である．しかし，DDoS 攻撃の送信元は複数あり，また各 bn のバッファ容量が異なることが想定されるため，提案システムでは二次元トポロジに対応し，かつバッファ使用率の平滑化を行うものとする．この 2 つの条件を満たすように式 (1) の転送レート算出式を拡張したものが式 (4) である．また，その要素を式 (5)，式 (6)，式 (7) 示す．

$$\begin{aligned}\tilde{J}_{f,i}(t) = {} & D'_{f,i} * r_{f,i}(t - d_{f,i}) \\ & - D_{f,i} * S_{f,i}(u_{f,i+1}(t - d_{f,i}) - u_{f,i}(t))\end{aligned} \tag{4}$$

$$J_{f,i}(t) = \max(0, \min(\tilde{J}_{f,i}(t), L_{f,i})) \tag{5}$$

$$D'_{f,i} = \frac{1}{\text{下流ノード } i+1 \text{ が持つ上流ノード数}} \tag{6}$$

$$D_{f,i} = \frac{1}{\text{下流ノード } i+1 \text{ が持つ上流ノード数} + 1} \tag{7}$$

ただし，f をフロー番号，i をノード番号，分散係数を $D'_{f,i}$，拡散係数を $D_{f,i}$，スケール係数を $S_{f,i}$ とすると，$J_{f,i}(t)$ は，下流ノードから送られてきた転送レート $r_{f,i}(t - d_{f,i})$，バッファ使用率 $u_{f,i}(t)$，下流ノードから送られてきたバッファ使用率 $u_{f,i+1}(t - d_{f,i})$，使用可能帯域 $L_{i,j}$ より算出される．ただし，スケール係数 $S_{f,i}$ はフロー f，ノード i のバッファ容量とする．提案システムでは，この式 (4) から式 (7) を用いて算出した転送レートを採用する．また，転送レート算出式の拡張にともない，フィードバック情報も「転送レート」,「バッファ使用率」,「上流ノード数」とする．

4. シミュレーションによる評価

4.1 提案システム評価のためのシミュレータ

提案システムを評価するために，標的サーバ宛の DDoS

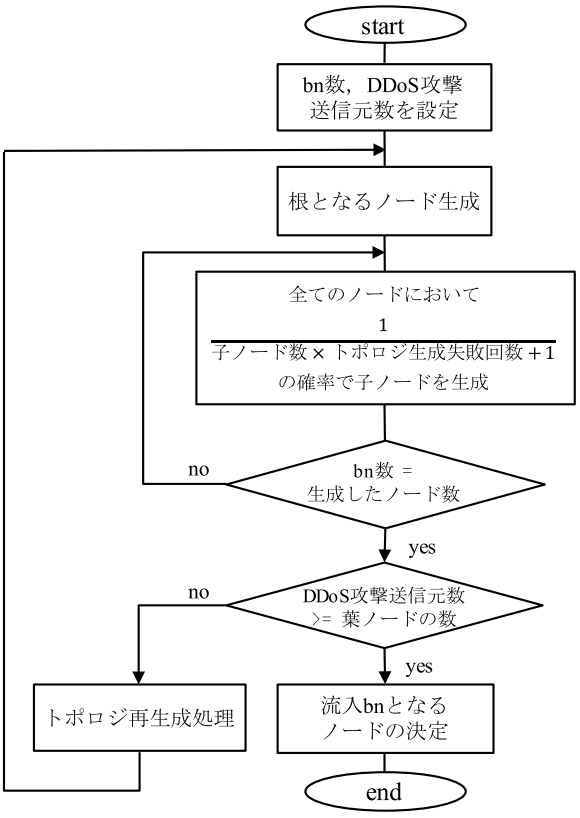


図 4 トポロジの生成アルゴリズム
Fig. 4 Topology generation algorithm.

攻撃を開始した後，提案システムのミチゲーションを行うようなシミュレータを 1 台の物理マシン上に作成した．シミュレータのうち，送信元と標的サーバ間のトポロジを生成するプログラムは ruby 言語で，DDoS 攻撃とその緩和をシミュレーションするプログラムは C 言語で作成した．シミュレーションの流れは以下のとおりである．

(1) 複数の送信元が標的サーバ宛に DDoS 攻撃を開始
(2) 全送信元からのパケットが標的サーバに到達したことをトリガに，流出 bn にミチゲーション通知を送信
(3) ミチゲーション通知を受け取った bn はミチゲーションを開始するとともに，上流 bn にミチゲーション通知を送信

ミチゲーションに使用される bn の数および DDoS 攻撃の送信元の数は後述するシミュレータの入力パラメータで指定する．DDoS 攻撃の送信元の数と流入 bn 数は等しいものとする．

複数の送信元から標的サーバまでのトポロジは**図 4** のアルゴリズムに従ってランダムに生成した木構造である．このとき，トポロジ生成には以下の条件を設けている．

- bn 数をトポロジ中のノード数とする
- 流出 bn をルートとする
- DDoS 攻撃送信元数は木の葉ノード数以上
- 流入 bn となるノードをランダムに選択する（ただし，葉ノードを優先して流入 bn とする）

表 1 シミュレーションのパラメータ
Table 1 Simulation parameters.

パラメータの種類	値		
	測定(1)	測定(2)	測定(3)
bn 数	60	60	60
DDoS 攻撃送信元数	20	20	20
攻撃トラヒック送信レート[pps]	5.0×10^2	1.0×10^5	1.0×10^5
流出 bn 送信レート[pps]	1.0×10^4	1.0×10^4	1.0×10^4
bn のバッファ容量[packet]	2.0×10^6~6.0×10^6	1.0×10^5~16.0×10^6	2.0×10^6
bn 間使用可能帯域[pps]	1.0×10^6	1.0×10^6	1.0×10^6
bn 間の伝搬遅延時間[ms]	10	10	10
フィードバック情報送信間隔[ms]	10	10	10

DDoS 攻撃送信元数が葉ノード数より少ない場合，ミチゲーションに使用されない bn ができる．シミュレーションでは全 bn がミチゲーションするように bn 数を固定して測定・比較を行うため，図 4 中の条件式である「DDoS 攻撃送信元数 >= 葉ノードの数」を満たさない場合はトポロジ生成を失敗とし，再生成を行う．また，子ノードの生成確率を単純な乱数で決めると bn 数を増やしてトポロジを生成した際に木の幅が広くなる確率が高くなり，この条件を満たさずにトポロジ生成が失敗する．そのため，子ノードの生成確率を 1/(子ノード数 × トポロジ生成失敗数 + 1) としている．これにより，トポロジ生成が失敗するごとに木の幅が狭く，深さが大きいトポロジが生成されやすい．このようにして生成した木構造を bn のオーバレイネットワークとし，シミュレーションではミチゲーション開始前の bn はトラヒック制御を行わずパケットの転送のみを行うものとした．

また本稿で想定しているトポロジでは標的サーバは 1 台であるが，標的サーバが複数存在する場合，流出 bn を根とするオーバレイネットワークを複数構成することになる．このように bn が複数の標的サーバ宛のトラヒックをミチゲーションする場合は，オーバレイごとに使用可能なバッファ容量を分割し，それぞれのオーバレイで転送レートを算出する必要がある．具体的なバッファ容量の分割方法についてはそれぞれの DDoS 攻撃の影響の強さを考慮する必要があり，今後の課題とする．

シミュレータの入力パラメータは**表 1** のとおりである．「攻撃トラヒック送信レート」は DDoS 攻撃の送信元からの攻撃パケットの転送レートである．「流出 bn 送信レート」は流出 bn が標的サーバに転送するパケットの送信レートで，「bn 間使用可能帯域」は標的サーバ宛のトラヒックが使用可能な帯域である．各 bn は 3.4 節で示した転送レート算出式により実際に使用する転送レートを決定する．「bn

間の伝搬遅延時間」はbn間の転送パケットとフィードバック情報の転送にかかる時間で，「フィードバック情報送信間隔」は各bnのフィードバック情報を送信する時間間隔である．提案システムの基本性能評価のため，bn数はバッファ使用率の拡散を観測しやすい台数（60台）に設定しているが実際の運用においてbn数の適正値を検討する必要がある．攻撃サーバから標的サーバまでに経由する緩和システムの数や各緩和システムに存在するbn数によってその適正値は変化すると考えられる．またDDoS攻撃送信元数は今回使用したパラメータの値20に比べ実際は大きくなる可能性もある．提案システムの基本性能を測るうえでこの値にしているが，攻撃送信元が増え，DDoS攻撃緩和システムを構築するネットワークが巨大化しても，拡散効果によりbn間でのバッファ使用率の平滑化は実現する．bn数や攻撃送信元数に対する提案システムの性能評価については今後の課題とする．

また，一般的なエンドホストが属するネットワークの帯域を100 Mbpsとし，インフラへのDDoS攻撃は1 packetあたりのサイズが大きくなる傾向にあることを考慮し1250 Byteを1 packetとして換算したため，流出bn送信レートを1.0×10^4 ppsに固定している．また本稿のシミュレーションでは提案システムの基本特性を調査するために「bn間使用可能帯域」はインターネットで用いられる回線帯域やルータの処理性能を考慮したうえで，算出される転送レートよりも十分大きな値としている．bn間使用可能帯域は固定値であり，シミュレーション中bn数およびフロー数は変化しないため，使用可能帯域$L_{f,i}$の時間変動はない．bn間の伝搬遅延時間はネットワークの異なるエンドホスト間の伝搬遅延時間よりも短い10 msとした．フィードバック情報送信間隔は短い間隔であるほど迅速にバッファ使用率の平滑化が行える．フィードバック情報は，バッファ使用量とパケットの転送レート情報だけで構成されているためフィードバック情報のパケットサイズは大きくない．またフィードバック情報は上流方向にのみ送信されるため，下流方向のネットワーク負荷を増長することはなくフィードバック情報のオーバヘッドによる大きな影響はないと考えている．

4.2 評価方法

提案システムは既存の緩和装置を用いてフィルタリングすることを想定しているため，「ミチゲーション段階」までの時間に焦点を当てて性能評価する．また，DDoS攻撃の総トラヒックに対してbn全体のバッファ容量が大きいほど対策開始までのバッファ溢れを遅らせることが可能となるが，バッファ全体を効率的に使用できなければbnの台数やバッファ容量の増設にかかるコストが増加する．そのため本研究ではバッファの使用効率に着目して評価を行う．流入bnがミチゲーションを開始後，送信元に近いbnからミチゲーションを行うため，下流M_NWでの輻輳を回避することが可能である．しかし，ミチゲーションによってすべてのbnでバッファ溢れを起こさないようにしなければ，パケット損失が発生する．そのため，作成したシミュレータでミチゲーションを行い，bnのバッファ容量の違いやトポロジの違いによる全bnのバッファの使用効率を調べる．

4.2.1 バッファ容量のばらつきに関する測定

シミュレーションでは，まず拡張した転送レート算出式がbnごとのバッファ容量にばらつきがある場合でも，各bnのバッファ使用率を平滑化できるかをバッファ使用率の分散を用いて確認し，バッファ溢れによるパケットの損失を遅延させることが可能かを調べる．このとき，表1の測定(1)のパラメータの値を用いた．

各bnのバッファ容量が

(a) 4.0×10^6 packetで固定

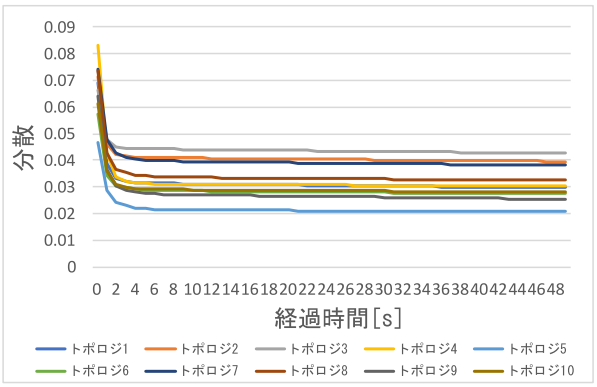
(b) 2.0×10^6 packetから6.0×10^6 packetの間でランダム

の場合の2通りについて評価する．また初期状態での各bnのバッファ使用率は$[0, 0.9]$の範囲の乱数とする．シミュレーションは10パターンのトポロジを用いて，バッファ容量2通り(a)，(b)それぞれの場合で10回行う．経過時間の開始時刻を流出bnのミチゲーション開始（4.1節(3)）時刻とする．上記の条件を用いて，全bnのバッファ使用率の総量は増やさずにバッファ使用率が平滑化する様子をシミュレーションにより確認する．

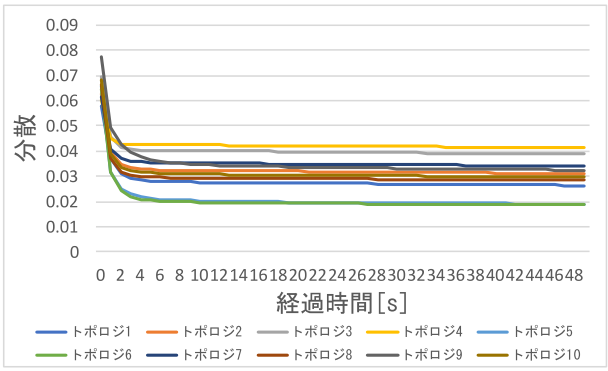
上記評価により，ネットワーク全体のバッファ使用率のばらつきは分かるが，どこに位置するbnのバッファ使用率が高いか/低いかといった空間上の分布は分からない．そこでトポロジ空間における上流側と下流側のbnのバッファ使用率において差異が認められるかを確認した．使用するパラメータは表1の測定(1)，各bnのバッファ容量は4.0×10^6 packetで固定とする．

4.2.2 バッファ容量の総量の違いに関する測定

bnの総バッファ容量を変更してシミュレーションを行い，任意のbnのバッファ溢れによるパケット損失時の全bnの平均バッファ使用率を測定することで，bnのバッファ容量の大小が平滑化に与える影響を調べる．表1の測定(2)のパラメータの値を用いた．実際のDDoS攻撃を想定して，流出bn送信レートよりも全DDoS攻撃送信元からの攻撃トラヒック送信レートの合計を大きく（200倍）設定し，bnのバッファ容量が1.0×10^5 packet，2.0×10^5 packet，5.0×10^5 packet，1.0×10^6 packet，2.0×10^6 packet，4.0×10^6 packet，8.0×10^6 packet，16.0×10^6 packetの場合の8通り，かつそれぞれのバッファ容量において10パターンのトポロジを用いてシミュレーションを行う．4.2.1項での評価と同様，流出bnのミチゲーション開始時刻を0とする．

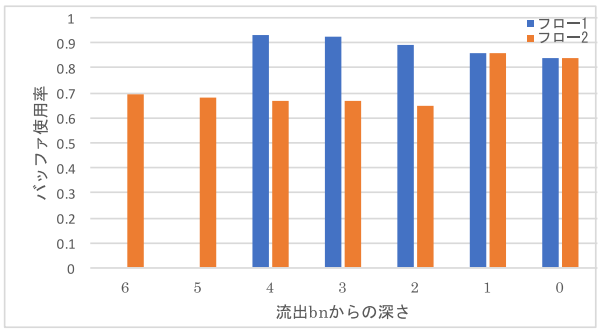


(a) 全ノード(bn)のバッファ容量が同じ場合
(a) The case of same buffer size of all nodes (bn)

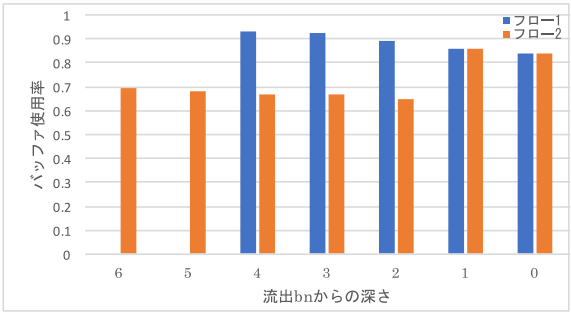


(b) 各ノード(bn)のバッファ容量が異なる場合
(b) The case of different buffer size of each node (bn)

図 5 バッファ使用率の分散の経時変化
Fig. 5 Variance of buffer usage rate.



(a) 全攻撃トラヒック送信レートの合計が流出 bn 送信レートと等しい場合
(a) In the case when the rate of all attack traffic is equal to transmission rate from outgoing bn



(b) 全攻撃トラヒック送信レートの合計が流出 bn 送信レートと異なる場合
(b) In the case when the rate of all attack traffic is different from the transmission rate from outgoing bn

図 6 フローごとの bn のバッファ使用率
Fig. 6 Buffer usage rate of bn for each flow.

4.2.3 トポロジの違いに関する測定

送信元から標的サーバまでのトポロジをランダムに 100 パターン生成し，各トポロジでのパケット損失発生時刻とそのときの全 bn の平均バッファ使用率を計測して，トポロジがミチゲーションに与える影響を考察する．このとき，表 1 の測定 (3) のパラメータの値を用いた．4.2.1 項同様，流出 bn のミチゲーション開始時刻を 0 とする．

4.3 結果と考察

4.3.1 バッファ使用率の効率化

4.2.1 項の測定結果として 10 パターンのトポロジでのバッファ使用率の分散の経時変化を**図 5** に示す．図 5 の (a) と (b) のどちらも 10 パターンのすべての分散がシミュレーション開始後数秒で下降し，その後は横這い，もしくは緩やかに減少している．このことから，バッファ容量のばらつきの大きさにかかわらず，拡張した転送レート算出式を用いることで各 bn のバッファ使用率の平滑化を数秒で行うことができることが分かった．一方で，バッファ使用率の初期値やトポロジの違いにかかわらず，バッファ使用率の分散がある一定値から下降しないことも判明した．その原因は以下のとおりである．

連続空間，連続時間における拡散現象では，物理量は時間とともに拡散し一定になる．つまり，理論上，分散は 0 に近づくが，本提案方式のように離散的な制御をしている限り，分散の収束には限界がある．連続空間・連続時間上で考えると一瞬でフィードバック情報が届くということを示唆しているが，実際には，フィードバック情報は伝搬遅延時間だけ遅れて届き，伝搬遅延時間分だけ「隣接ノードの古い情報」を用いて転送レート算出することになる．また，想定しているトポロジは二次元の離散空間上の点（均一に配置されているわけではない点）をリンクしたネットワーク構造になるため，トポロジ構造によって平滑化の振舞いが異なる．文献 [19] では 30 秒以内に対策を開始できることが示されているが提案システムは数秒でバッファを平滑化できており，ミチゲーションによりパケット損失の発生を遅らせることが可能なため，通信セッションの維持や提供サービス継続に有効である．

次に上流側と下流側の bn のバッファ使用率について調査した．平滑化後のバッファ使用率の分散値が最も大きかったトポロジにおいて特徴的な 2 つのフロー（フロー 1，

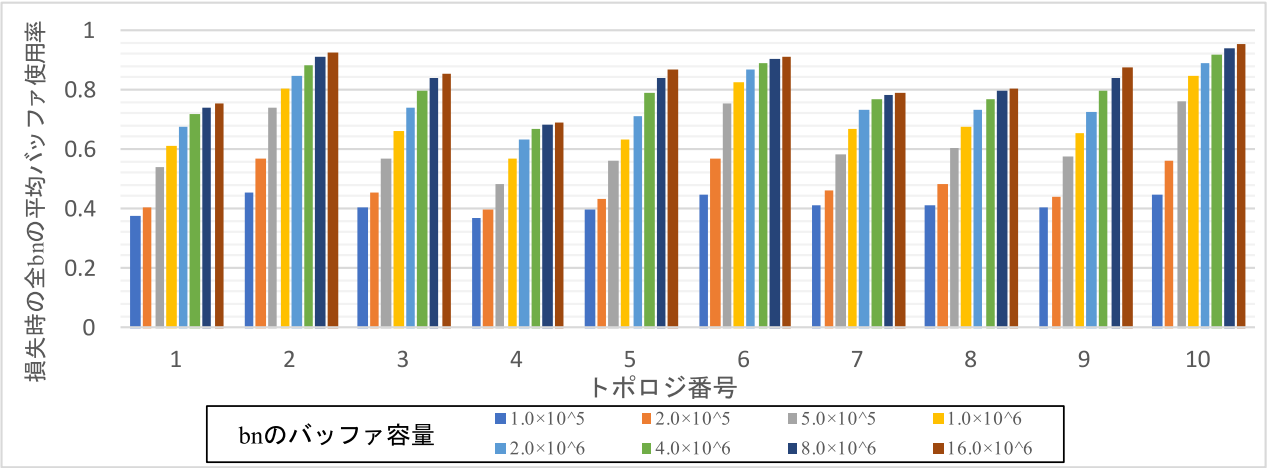


図 7 パケット損失発生時のノードの平均バッファ使用率
Fig. 7 Average of buffer usage rate of all nodes at the time of packet loss occurrence.

フロー 2）を選択し，30 秒経過時の各 bn のバッファ使用率を**図 6** の (a) に示す．横軸は流出 bn（最下流 bn）からの深さ（ホップ数）を表し，フロー 1 およびフロー 2 の流入 bn（最上流 bn）はそれぞれ深さ 4 および 6 に位置する．

フロー 1 は上流の bn ほどバッファ使用率が高くなっており最上流 bn と最下流 bn のバッファ使用率の差は 1%程度である．一方，フロー 2 は上流の bn に比べ下流の bn のバッファ使用率が高くなっている．これは，フロー 2 と合流する他のフローの bn のバッファ使用率が高いことに起因している．また，DDoS 攻撃を想定して測定 (1) の攻撃トラヒック送信レートを 200 倍にし，全攻撃トラヒック送信レートの合計が流出 bn 送信レートに比べて高い状況で同様の測定を行った．結果は図 6 (b) のとおりである．図 6 のフロー 1 より，最上流の bn と最下流の bn のバッファ使用率の差が 10%程度あり，全攻撃トラヒック送信レートの合計が大きいほど上流と下流の bn のバッファ使用率に差が出ることが分かった．他の 9 つのトポロジで同様の測定を行ったところ，バッファ使用率の差はトポロジごとに異なるが同じ傾向が見られた．フロー 2 に関しては図 6 (a) のフロー 2 の結果と同様バッファ使用率の高いフローと合流する bn のバッファ使用率が高くなっていることが確認できる．これらの結果より，上流と下流の bn のバッファ使用率の差は全攻撃トラヒック送信レートの合計とトポロジに依存することが分かった．

4.3.2 パケット損失時の全 bn のバッファ使用率の平均

4.2.2 項の測定結果として，パケット損失時の全 bn の平均バッファ使用率を**図 7** に示す．図 7 より 10 パターンのトポロジすべてにおいて，bn のバッファ容量が大きいほどパケット損失時の全 bn の平均バッファ使用率が高いという結果が得られた．バッファ容量が小さな場合にパケット損失時の全 bn のバッファ使用率が低い理由は，攻撃トラヒックの転送レートを制御する以前に任意の bn でバッファ溢れを引き起こしているためである．またバッファ容量が大きいと転送レートを制御するまでの時間ができるため全 bn のバッファを効率良く使用できる．攻撃トラヒックに対して大きなバッファ容量を用意することで，早期のバッファ溢れを防ぐことができ，正規パケットの損失を小さくすることができる．

表 2 トポロジの違いによる結果
Table 2 Results by difference of topologies.

	平均値	最小値	最大値
パケット損失発生時刻[s]	47.25	36.47	54.72
全 bn の平均バッファ使用率	0.78	0.61	0.91

4.3.3 トポロジの違いによる影響

トポロジの違いによるパケット損失発生時刻とそのときの全 bn の平均バッファ使用率の平均値，最小値，最大値を**表 2** に示す．トポロジによりパケット損失発生時刻が最大 20 秒異なり，トポロジによる影響が大きいことが読み取れる．これは，トポロジによって送信元から標的サーバまでに経由する bn 数が異なること，また流出 bn からの深さが浅い位置に DDoS 攻撃の送信元がいる場合，その送信元からのトラヒックを緩和する bn が少ないため，1 台あたりの bn のバッファ使用率が高くなることが起因する．解決には，転送レート算出式の改良や bn のオーバレイネットワークトポロジ構築方法の変更が必要である．

5. まとめと今後の課題

本稿では，ISP 間で連携し DDoS 攻撃トラヒックを制御することで正規パケットの損失を防ぐ DDoS 攻撃緩和システムを提案と評価について述べた．実験的評価により，提案の緩和システムは拡散型フロー制御を用いたことで，ネットワーク全体の状況を一元管理することなく，自律的に DDoS 攻撃の緩和をすることができることを示した．また，ネットワーク内に設置する緩和用のバッファリング

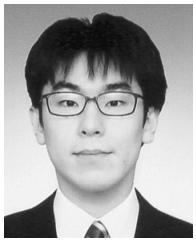
ノードはバッファ容量が異なる場合も数十秒以内に平滑化できることから，ネットワークに設置するバッファリングノードの制約も小さい．ただし，バッファリングノードのバッファ容量が小さい場合やトポロジの違いによってバッファ溢れが早期に発生する可能性があるため，バッファ容量の設定やトポロジの構築方法は今後の検討課題である．

今後は，バッファリングノードのオーバレイネットワークのトポロジ構築方法の検討と検知，緩和装置を含めたプロトタイプシステムの設計と実装によって評価を行う．

謝辞 本研究の一部は，日本学術振興会科学研究費助成金 15K00130，15K00431，16H02808，17H0173 の支援を受けて実施しました．

参考文献

[1] 文部科学省：平成 28 年版科学技術白書 (2016).
[2] NIST: Advanced DDoS Mitigation Techniques, 入手先 ⟨https://www.nist.gov/programs-projects/advanced-ddos-mitigation-techniques⟩ (参照 2017-07-12).
[3] 西塚 要：増え続ける DDoS 攻撃に対抗するために事業者間で協してできること，JANOG38（オンライン），入手先 ⟨https://www.janog.gr.jp/meeting/janog38/download_file/bof_ddos_pub.pdf⟩（参照 2017-11-22）.
[4] Akamai: akamai's [state of the internet]/security Q1 2017 report, pp.1–26 (2017).
[5] 土屋師子生：BGP Flowspec（RFC5575）, JANOG35（オンライン），入手先 ⟨https://www.janog.gr.jp/meeting/janog35/download_file/106/index.pdf⟩ (参照2017-11-22).
[6] 前田浩明，小島久史，相原正夫：大規模 DDoS 攻撃対処を想定した AS 間連携対処方式の一検討，電子情報通信学会信学技報，Vol.116, No.251, pp.55–60 (2016).
[7] Akamai：DDoS 防御に Akamai のクラウドセキュリティを使用する理由，入手先 ⟨https://www.akamai.com/jp/ja/products/cloud-security/ddos-protection-service.jsp⟩（参照 2017-11-22）.
[8] 西塚 要：ISP における DoS/DDoS 攻撃の検知・対策技術，JPNIC（オンライン），入手先 ⟨https://www.nic.ad.jp/ja/materials/iw/2013/proceedings/s2/s2-nishizuka.pdf⟩（参照 2017-11-22）.
[9] 水口孝則，神谷和憲，吉田友哉，桑原 大，グエンホウバッ，牛久裕輔，宮川 晋，市川弘幸：トラフィック解析システム SAMURAI とサービス展開，NTT（オンライン），入手先 ⟨http://www.ntt.co.jp/journal/0807/files/jn200807016.pdf⟩（参照 2017-11-22）.
[10] 平 空也，高野知佐，前田香織：拡散型フロー制御を用いる DDoS 攻撃緩和システムの提案，電子情報通信学会信学技報，Vol.117, No.173, CCS2017-29, pp.51–56 (2017).
[11] 平 空也，高野知佐，前田香織：拡散型フロー制御を用いる DDoS 攻撃緩和システムの評価，電子情報通信学会信学技報，Vol.117, No.294, IA2017-23, pp.1–6 (2017).
[12] 山田洋之，久保田光一：IDS を用いた DDoS 攻撃の検知，情報処理学会第 78 回全国大会，pp.555–556 (2016).
[13] 太田 悟，田島伸一，佐藤 信，長野純一，篠宮紀彦，勅使河原可海：OpenFlow を用いた DDoS 攻撃検知システムの検討，情報処理学会第 75 回全国大会，pp.543–544 (2013).
[14] Chen, Y., Hwang, K. and Ku, W.S.: Collaborative Detection of DDoS Attacks over Multiple Network Domains, *IEEE Trans. Parallel and Distributed Systems*, Vol.18, No.12, pp.1649–1662 (2007).
[15] Wang, X., Li, M. and Li, M.: A Scheme of Distributed Hop-count Filtering of Traffic, *IET International Communication Conference on Wireless Mobile and Computing* (*CCWMC*), pp.516–521 (2009).
[16] 住 達郎，高野知佐，会田雅樹，石田賢治：拡散方程式に基づく自律分散的輻輳制御技術の実証実験，電子情報通信学会論文誌 D，Vol.J95-D, No.12, pp.2048–2058 (2012).
[17] 高野知佐，山内正志，会田雅樹：拡散現象を指導原理とする自律分散フロー制御の実装に向けたアクティブフロー数計測技術の検討，電子情報通信学会論文誌 B，Vol.J91-B, No.10, pp.1254–1266 (2008).
[18] radware：DoS/DDoS 攻撃からアプリケーションレベル攻撃までカバーする究極のセキュリティ対策，入手先 ⟨http://www.radware.co.jp/product/dp/index.html⟩（参照 2018-03-19）.
[19] ARBOR NETWORKS: Arbor Networks Peakflow® 7.0 が，DDoS 攻撃検知とミティゲーションの時間を大幅に短縮，入手先 ⟨http://jp.arbornetworks.com/lorem-post-3/⟩（参照 2017-09-26）.



平 空也

平成 28 年広島市立大学情報科学部情報工学科卒業．平成 30 年同大学大学院情報科学研究科博士前期課程修了．修士（情報工学）．ネットワークセキュリティに関する研究を行う．

高野 知佐 （正会員）

平成 12 年大阪大学工学部電子通信工学科卒業．平成 20 年首都大学東京大学院博士後期課程修了．平成 12 年 NTT アドバンステクノロジ（株）入社．平成 20 年広島市立大学大学院情報科学研究科准教授．博士（工学）．通信トラヒック制御，自律分散制御技術，社会ネットワーク分析の研究に従事．IEEE，電子情報通信学会各会員．

前田 香織 （正会員）

昭和 57 年広島大学総合科学部卒業．同大学工学部助手，（財）放射線影響研究所技術員，広島市立大学情報科学部助手，同大学情報処理センター助教授を経て，現在，同大学大学院情報科学研究科教授．博士（情報工学）．コンピュータネットワーク，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．