情報処理学会論文誌 Vol.58 No.9 1432–1449 (Sep. 2017)

マルチホーミングにおける IPアドレスホッピングを用いたDDoS攻撃防御方式

岩永 崇裕[1,†1,a)] 木村 成伴[1,b)]

受付日 2016年11月29日, 採録日 2017年6月6日

概要：インターネットセキュリティを脅かす強力で防御が難しい攻撃の 1 つに DDoS 攻撃があげられる．この DDoS 攻撃では，送信元の IP アドレスを偽装する “IP スプーフィング” を用いることで，攻撃の発生源を容易に特定することが妨げられるため，結果的に，攻撃を防ぐことが困難になる．この問題を解決するため，Khor らは，Overfort を提案している．この方式では，サーバはマルチホーミングしており，通常は，一方の IP アドレスからのみ接続を受け付ける．攻撃時には，この IP アドレスでの接続は放棄し，秘匿していた方の IP アドレスでのトンネリング接続のみ受け付けるが，攻撃者がサーバの名前解決に利用する LDNS（Local DNS）サーバに，トンネルの入り口の IP アドレスを通知しないペナルティを与えることで，攻撃パケットを送付できなくしている．しかし，この方式では，ISP が Overfort に対応するゲートウェイを導入する必要があるなどの課題があった．そこで本論文では，攻撃者が用いる LDNS に対して，変更後のアドレスを通知しない，というアイデアに基づき，ISP との連携を必要とせず，サーバサイドのみの変更で実現が可能な，DDoS 攻撃防御方法を提案する．最後に，シミュレーション実験を行い，サーバに与えられる十分な IP アドレスがある場合，提案方式は，本論文で定義する様々な攻撃が，防御可能であることを示す．

キーワード：IP アドレスホッピング，マルチホーミング，DDoS 攻撃防御方式

A DDoS Attack Defense Method Using IP Address-hopping on Multihoming

Takahiro Iwanaga[1,†1,a)] Shigetomo Kimura[1,b)]

Received: November 29, 2016, Accepted: June 6, 2017

Abstract: A DDoS attack is one of the most important threats that are hard to protect for internet security. The attack uses IP spoofing to fake the source address so that the attacker's source point is not easy to identify. In order to solve the problem, Khor et al. proposed Overfort. In this method, a server is multi-homing, where it usually accepts connections only from the one of the IP addresses and hides the rest of them. When the server is attacked, it ignores all the connections to the IP address, and accepts only tunneling connections to the one of hidden IP addresses. It also gives a penalty not to notify the IP address of the entrance of the tunnel for the LDNS (Local DNS) server that an attacker uses to resolve the server's address. However, the method has problems that ISPs need to introduce a gateway to support Overfort, and so on. This paper proposes a new DDoS attack defense method based on the idea not to notify the changed address to the attacker's LDNS. The proposed method can be realized by only the efforts of the server side without cooperation of the ISPs. Finally, the simulation experiments show that the proposed method can protect against DDoS attacks defined in the paper, if the server has enough IP addresses.

Keywords: IP address-hopping, multihoming, DDoS attack defense method

[1] 筑波大学
University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[†1] 現在，東京大学
Presently with University of Tokyo

[a)] iwanaga-takahiro650@g.ecc.u-tokyo.ac.jp
[b)] kimura@cs.tsukuba.ac.jp

1. はじめに

インターネットが多くのビジネスによって利用され，重要なインフラとなった現代において，DDoS（Distributed Denial of Service）攻撃によってもたらされる損害は，とても深刻ものとなっており，いかに迅速に，かつ正確にDDoS攻撃を防ぐかが，ネットワークセキュリティ上での最も重要な課題の1つになっている．しかし，DDoS攻撃は，送信元のIPアドレスを偽装する“IPスプーフィング”を用いることで，攻撃の発生源を容易に特定することが妨げられることから，その防御は困難であった．各ISPが流入パケットの送信元IPアドレスが偽装されていないもののみを通すSource Address Validationなどを導入すれば改善することが考えられるが，導入のモチベーションが不足していることから普及していない現状にある[1]．これに対し，既存研究としてオーバプロビジョニングやIPトレースバックなどの防御方式があるが，対処に時間がかかる，利用範囲に制約があるなど，根本的な解決にはならなかった．

この問題を解決するため，Khorらは，Overfort [2]を提案している．この方式では，サーバはマルチホーミングしており，通常は，このうちの一方のIPアドレスからのみ接続を受け付ける．攻撃時には，このIPアドレスでの接続は放棄し，秘匿していた方のIPアドレスでのトンネリング接続のみ受け付ける．さらに，攻撃者がサーバの名前解決に利用するLDNS（Local DNS）サーバに，トンネルの入り口のIPアドレスを通知しないペナルティを与えることで，攻撃パケットを送付できなくしている．しかし，この方式では，ISP（Internet Service Provider）が対応するゲートウェイを導入する必要があるなどの問題があった．

そこで本論文では，攻撃者が用いるLDNSに対して，変更後のアドレスを通知しない，というアイデアに基づき，ISPとの連携を必要とせず，サーバサイドのみの変更で実現が可能な，DDoS攻撃防御方式を提案する．最後に，IPスプーフィングされたDDoS攻撃のシミュレーション実験を行い，サーバに与えられる十分なIPアドレスがある場合は，提案方式により，本論文で定義する様々なDDoS攻撃から，防御可能であることを示す[3]．

2. 既存研究・方式

IPスプーフィングによるDDoS攻撃を防ぐための，既存の防御方式について説明し，その問題点を指摘する．

2.1 オーバプロビジョニング

オーバプロビジョニングとは，DDoS攻撃時であっても，正規ユーザの利用に支障がない，十分な，ネットワークリソースを確保しておく方式である．しかし，この方式は，非攻撃時にはリソースを持て余すうえに，その実現にはコストがかかりすぎるという問題がある．また，オーバプロビジョニングしたリソースを上回るDDoS攻撃を攻撃者が行うことも考えられ，これらのことから，オーバプロビジョニングはDDoS攻撃への有効な策とはなりえない．

2.2 IPトレースバックを用いた方式

IPトレースバックは，攻撃のターゲットから攻撃送信元までの通信経路をさかのぼり，攻撃送信元，またはこれに近い中継機器を特定する方式[4]で，DDoS対策手法においては，特定した後に送信量を制限するメッセージを送るなどして攻撃トラフィックの送信量を攻撃者に近い場所で減らす．

しかし，このIPトレースバックには，主に3つの問題がある．1つ目は，通信のオーバヘッドが発生する点である．パケットに対して，どのルータを通過したかを記録する処理がルータごとに行われるために，宛先へのホップ数が増えるほど，中継に要する遅延は大きくなる．2つ目の問題は，DDoS攻撃との相性が悪いことである．DDoS攻撃では，攻撃ノードは数千にのぼることがあり，すべての攻撃ノードを特定するには時間がかかる．3つ目の問題は，トレースバック環境の普及が難しい点である．通過したルータを記録するルータを設置する必要があるが，既存のネットワークすべてに本機能を実装するには，難しいと思われる．

2.3 パケットフィルタリングを用いた方式

ファイアウォールや，サーバが接続するネットワークの入口ルータおよび出口ルータでのACL（Access Control List）による，IPアドレスやプロトコルに基づいたフィルタリングは，送信元IPアドレスを偽装していない，サービスを提供していないプロトコルを使うなど，明らかにそれと分かる攻撃に対しては効力があるが，IPスプーフィングを行い，正規ユーザを装うパケットに対しては効力が薄い．そこで，送信元IPアドレスなどの情報に頼らないパケットフィルタリング手法がいくつか提案されている．

まず1つ目はHCF（Hop Count Filtering）を利用した手法[5], [6]である．これは，経由するルータごとに減らされるTTLフィールドから，送信元IPアドレスごとの正しいホップカウントを推測し，これをサーバサイド側でテーブルとして保持する．そして，送信されてきたパケットのホップカウントが，テーブル上のホップカウントと一致するか（もしくは近しい値か）どうかを基準にフィルタリングを行う．攻撃者はTTLの値の減少数を偽装できないことを利用したフィルタリング手法であるが，サーバサイドでのフィルタリングになるので，サーバサイドに至るまでの帯域を保護できない．これに対して，中間ルータでHCFを実装して帯域を保護する手法も考案されている[7]が，これらのルータに本来の役割から逸れた処理・負荷が

増えてしまう．また，それぞれの送信元から複数の経路を取りうる場合も考慮していくと，テーブルのサイズが膨大になり，フィルタリングの処理に支障がでるといった欠点もある．

2つ目の手法として，ルータが，自身を経由したパケットに対して，ルータ自身の識別情報を付加（マーキング）し，宛先では，このマーキング値を元にフィルタリングを行うものがある．たとえば，Pi（A Path Identification Mechanism）方式 [8] では，ネットワーク上の各ルータが，通過したパケットに対して自身のIPアドレスの一部をマーキングする．パケットを受信したサーバは，到着するまでの過程で格納されてきたマーキング値，すなわち経由した経路ごとに固有の値を用いて，DDoS攻撃時に遮断するパケットを選択する．この方式は，他の提案手法に比べて，高い効果が得られることが示されている [9]．また，フィルタリングの際に用いる情報として，Pi方式のパス識別子だけでなく，プロトコルの種類を用いて精度を上げる方式 [10] や，Pi方式と Pushback 方式 [11], [12] を組み合わせて，攻撃元特定の精度向上を図りつつ，攻撃送信元に近いトラフィック上流に対してトラフィック制限要求を行う防御方式 [13] も提案されている．しかし，この方式には，IPトレースバックと同様に，マーキングのためのオーバヘッドが発生する，マーキングをするルータの普及が難しい，という2つの問題がある．また，1つ目の問題点を補うために，送信元のノードが属するサブネットのWAN側ルータでのみ，パケットに送信元情報（MACアドレスとTTL）をマーキングすることで，送信元を識別する方式が提案されている [14]．この方式では，一度マーキングするだけで済むので，通信上のオーバヘッドは比較的小さいが，攻撃者らがWAN側ルータを設置し，偽の送信元情報を送られると，防御が難しくなる可能性がある．また，MACアドレスをインターネット上に流すことは，プライバシ保護の観点から，好ましくない．

2.4 新しいアーキテクチャの考案・利用

簡単に送信元アドレスが偽装できてしまうなどの欠陥を抱える現在のネットワークアーキテクチャの根本的見直しを行うことでDDoS攻撃に対処する提案もされている．

たとえば，新しいルーティングアーキテクチャであるThe Locator/ID Separation Protocol（LISP）をさらに拡張し，送信元の特定を容易に行えるようにしたうえで，攻撃者の所属するネットワークに近いネットワークに攻撃を引き受けるダミーサーバを置くことによって，コアネットワークに流れる攻撃トラフィックを減らす手法が提案されている [15]．この手法では，正規サーバと全く同じ振舞いをするダミーサーバが攻撃トラフィックを引き受けるため，攻撃者が防御されていることに気づきにくい．ただし，攻撃者に近い位置にあることが望ましいダミーサーバの設置コストは大きな課題となってくる．また，ダミーサーバをオリジナルのサーバと同じ振舞いをさせるためには，オリジナルサーバの変更内容などを適宜ダミーサーバに同期させる必要があり，その点で手間がかかってしまう．

ほかにも，既存のDDoS攻撃防御手法の限界がインターネットアーキテクチャにあるとして，DDoS攻撃に対して効果的に防御できるアーキテクチャプリミティブがあるかどうかを検討するために考案されたSTRIDEなどがある [16] が，これらの新しいアーキテクチャによる手法では，既存の設備に対する変更コストが比較的大きく，今までのノウハウが流用できない環境での安定した運用が求められる点から，現場での導入が避けられがちなため，普及させるための実世界でのプロモーションが大きな課題となってくる．

2.5 ブラックホールルータを用いるもの

ISPなどで実用されているDDoS対策の1つにブラックホールルーティングがある．これは，攻撃発生時に被害者サーバへ向かうトラフィックを，受け取ったパケットをすべて廃棄するブラックホールルータへ転送する手法である [17]．

そのような手法の1つとして，攻撃発生時に，ネットワークの入口となるエッジルータに対して被害者サーバ宛のトラフィックをすべて落とすブラックホールになるよう要請するDestination-Based Remotely Triggered Black Hole Filtering [18] がある．この手法は宛先IPアドレスに基づいた防御なので，IPスプーフィングにかかわらず，被害者サーバへの負荷を減らすことができるが，正規トラフィックや攻撃トラフィックの区別をすることなく廃棄する無差別な防御法なので，結果的に攻撃者のサービスを妨害する目的が達成されてしまう危険性があり，サーバのサービス可用性よりもネットワークの帯域を保護することを優先する方式であると考えられる．

また，ルータが，パケットを受信したとき，受信したインタフェースから，送信元への最適なリバースパスがあるかどうかを確認することで，偽装パケットを識別するUnicast Reverse Path Forwarding（URPF）を用いる方式として，Source-Based Remotely Triggered Black Hole Filtering [18] があるが，URPFを用いるためには，リンクが双方向通信可能であり，かつ送信者からルータへのフォワードパスとルータから送信者へのリバースパスが対称的で，ユニキャストのルーティングテーブルが通信中はつねに固定している必要がある．特に，大規模なネットワークでは，フォワードパスとリバースパスが非対称なケースは珍しくないので，バックボーンネットワークなどでは利用できないという問題がある．

2.6 代理応答

代理応答は，ISP などで実用されている IP スプーフィングされた攻撃へのもう 1 つの対策方式である．対象サーバへリクエストを中継する前に，ISP のマシンがいったん送信元に対して代理で応答を行う．リクエスト送信元が正規ユーザであれば，正当な手順をふんでこの代理応答に応えるはずであるが，送信元 IP が偽装されていると，代理応答は “跳ね返りパケット” となり，リクエストを送っていない無関係のマシンに届いて代理応答に対して期待した応答は得られない．したがって，代理応答に正当に応えたユーザのパケットのみを ISP から対象サーバへ転送することで，IP スプーフィングされていないパケットの選別ができる．しかし，DDoS 攻撃では代理応答マシンが高負荷となり，このマシンがボトルネックとなることで正規パケットが落とされる危険性がある．さらに，TCP Connection Flood 攻撃のような IP スプーフィングしない正当な手順を踏む DDoS 攻撃は，素通ししてしまうという課題もある．

2.7 クラウド型防御

近年 DDoS 防御手法として実用されている方式の 1 つにクラウドを利用したスクラビングサービスがあり，たとえば F5 Networks，Arbor Networks などの DDoS ソリューションを提供する企業ら [19], [20] や，Akamai Technologies が運用している [21]．帯域型（ボリューム型）DDoS 攻撃は通常，サーバサイドで遮断しても意味は薄く，サーバサイドに至るまでの帯域が浪費されて攻撃は成功してしまう．クラウド型防御では，サーバサイドで DDoS トラフィックを検出した場合，トラフィックを高帯域・大容量データセンタ（クラウド）へリダイレクトする．そして，サーバサイドへ至る前のクラウド上で攻撃トラフィックを遮断し，正規パケットのみをサーバサイドへ届けることにより，帯域などのリソースの枯渇を回避する．この攻撃トラフィックを落とすスクラビングデータセンタは，あらゆる攻撃者，正規ユーザに近しい場所にあることが望ましいが，世界展開は容易ではない．2016 年現在，Akamai は全世界に 6 つしかこのデータセンタを持っておらず，Arbor Networks に関しても米国東海岸（ヴァージニア州），米国西海岸（カリフォルニア州），中欧（オランダ），アジア（シンガポール）にしか拠点を持っていない．DDoS 攻撃が発生した場合，リダイレクトされたトラフィックはこれらの場所をいったん経由しなければならないわけであるから，たとえば，日本から日本宛の正規トラフィックもシンガポールなどを経由しなければならないことになり，防御時のトラフィックにはフィルタリング以外にも迂回分のオーバヘッドが生じることになる．

2.8 Overfort

オーバプロビジョニングや，IP トレースバックを用いない，新たな DDoS 攻撃防御方式として，Khor らは Overfort を提案した [2]．この方式では，サーバはマルチホーミングしており，このうちの一方の IP アドレスを Public IP，他方を Secret IP アドレスと呼ぶ．通常は，Public IP への接続のみ受け付けるが，攻撃時には Public IP での接続は放棄し，あらかじめ正規ユーザの ISP などに設置したゲートウェイと Secret IP アドレス間のトンネリング接続のみ受け付ける．さらに，攻撃者がサーバの名前解決に利用する LDNS（Local DNS）サーバに，トンネルの入り口にあるゲートウェイの IP アドレスを通知しないペナルティを与えることで，攻撃パケットを Secret IP アドレスに送付できなくしている．

この Overfort は，IP スプーフィングなどのパケット偽装に強く，攻撃トラフィックを送信元になるべく近い場所で食い止められ，計画的な DDoS 攻撃を難しくできるという利点がある．特に，攻撃後はトンネルを通したアクセスのみ受け付けるため，サーバの名前をあらかじめ解決しておき，そこで得られた Public IP アドレスを継続して使用するような DDoS 攻撃はすべて空振りに終わる．しかし，ISP にこの方式に対応するゲートウェイを設置する必要があるなどサーバサイド以外のネットワークに変更を要する点や，なんらかのきっかけで Secret IP が漏えいすると，防御策が破綻する点が問題となる．

3. 提案方式

本章では，2.8 節で記した Overfort の問題点を解決しつつ，IP スプーフィングされた DDoS 攻撃を防御可能な方式の提案を行う．

3.1 設計方針

提案方式は，以下に示す方針に基づいて設計している．

I. **対攻撃用のオーバプロビジョニングを必要としない**
　ある程度のフラッシュクラウドに耐えうるプロビジョニングはしても，攻撃を防ぐ目的での大規模なプロビジョニングを必要としない方式を目指す．

II. **サーバサイドのみの実装で実現可能である**
　既存インフラに多くの変更点を必要とする方式は，普及の観点から考えて，あまり現実的ではない．したがって，本提案方式では，サーバサイドの実装のみで実現できる防御方式とする．

III. **サーバの IP アドレスが 1 つ漏えいするだけで防御策が破綻しない**
　Overfort では，サーバに Public IP と Secret IP の 2 種類の IP アドレスしか用意しておらず，Secret IP が攻撃者に漏えいすると，防御策が破綻してしまう恐れがあった．本提案方式では，サーバの IP アドレスがいくつか漏えいしても，破綻しない防御方式を目指す．

IV. 通信のオーバヘッドを最小限にする

提案方式では，IP トレースバックなどのように，パケットに防御のための付加情報を与える処理を行わず，Overfort のようにトンネル処理も行わない，オーバヘッドの小さな防御方式を目指す．

V. 帯域を消費させる攻撃・システム資源を消費させる攻撃の両方を防御可能とする

サーバの CPU やメモリといったシステム資源を対象とする攻撃だけでなく，帯域を浪費させる Flood 攻撃も防げる方式とする．

VI. フラッシュクラウド時にもある程度のユーザにサービスを継続し続けることを可能とする

予期せず正規ユーザが大量に訪れるフラッシュクラウドに直面した場合，何も対処を行っていないと，すべてのユーザがサーバにアクセスしにくくなるような事態を招くことになる．本方式では，フラッシュクラウド時にはあえて一時的に一部のユーザからのアクセスを封じることで，サーバへのアクセスを時間的に分散させ，すべてのユーザがサービスを利用できなくなる最悪の事態を回避することとする．

3.2 アーキテクチャ

提案方式を実現するためのサーバサイド（サーバが接続するサブネット）の構成を，以下に示す．

① マルチホーミング環境

提案方式では，Overfort と同様に，DDoS 攻撃を空振りに終わらせるため，また，サーバへ至るまでのリンクにおける帯域枯渇による，正規パケット損失に対処する目的で，サーバサイドは複数の ISP と契約するなどし，異なる複数のサブネットに接続する，マルチホーミング環境を構築している．ここで，サーバサイドが接続するサブネットの数を，本論文では外部接点数と呼び，この数は最低でも 2 とする．そして，サーバサイドにおいて，いずれか 1 つのサブネットのみからパケットを受け取る設定（以下アクティブ設定）にしておき，そのほかのサブネットから受け取ったパケットは捨てる設定（以下スタンバイ設定）にしておく．

② サーバの IP アドレス

提案方式では，サーバがそれぞれのサブネットに接続するネットワークインタフェースに，2 つ以上の同数のグローバル IP アドレスを設定する．

③ 権威 DNS サーバ

Overfort と同様に，各クライアントが利用するフルサービスリゾルバであるキャッシュ DNS（以降 Local DNS もしくは LDNS と呼ぶ）の IP アドレスに基づいて，サーバの権威 DNS（以下 ADNS）は通知する IP アドレスを変更する．これを実現するために，ADNS は LDNS の IP アドレス範囲と通知する IP アドレスの対応表である “IP アドレス通知テーブル” を保持することとする．LDNS によって通知するサーバの IP アドレスを変更することで，攻撃に利用されている IP アドレスから，攻撃に利用された LDNS（以下 Bad LDNS）を推定でき，この Bad LDNS に対して名前解決要求時にペナルティを与えることによって攻撃トラフィックの宛先を操作する．なお，このペナルティの内容については，後述する．

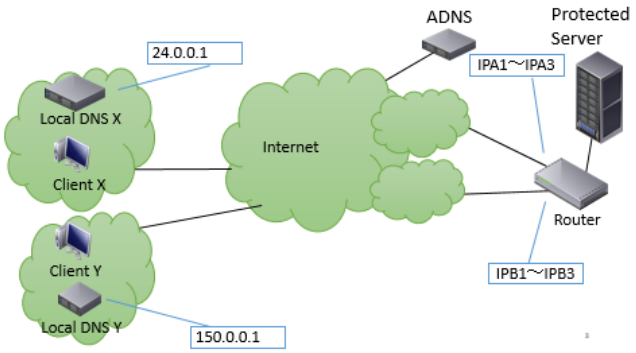

図 1 提案方式のネットワーク構成

Fig. 1 Network construction in the proposed method.

④ 到着トラフィック量の観測機構

提案方式では，Bad LDNS を推定するため，サーバサイドに到着した宛先 IP アドレスごとのスループットを観測し，トラフィック量が比較的多い宛先 IP アドレスを判別しなければならない．サーバサイドに FW（FireWall）を設置する場合，ここで破棄したパケットも含めたスループットを観測し，その情報をサーバに伝える機構が必要である．たとえば，FW に，宛先 IP アドレスごとのフィルタリングのルールを設け，そのルールを適用したバイト数を，SNMP（Simple Network Management Protocol）などを利用して，サーバが定期的に取得することによって，宛先 IP アドレスごとの，おおよそのスループットを算出することができる．

3.3 防御メカニズム

提案方式の防御メカニズムを，**図 1** を用いて説明する．この図において，サーバサイドは図右のルータとサーバであり，マルチホーミングによる外部接点数は 2 つである．それぞれのサブネットを図の上から A，B と呼ぶ．そして，それぞれのサブネットにつき，サーバの IP アドレスは 3 つであり，サブネット A におけるサーバの IP アドレスを IPA1，IPA2，IPA3 とし，サブネット B におけるサーバの IP アドレスを IPB1，IPB2，IPB3 とする．

サーバサイドでは，初期設定として，サブネット A をアクティブ設定にし，サブネット B はスタンバイ設定にする．また，サーバにアクセスするクライアント Client X と Client Y が存在し，それぞれは，自身が接続するサブネットのキャッシュ DNS の Local DNS X（LDNS X）と

表 1 アドレス通知テーブルの例
Table 1 An example of address notification table.

LDNS のアドレス	0.0.0.0〜 73.255.255.255	74.0.0.0〜 146.255.255.255	147.0.0.0〜 223.255.255.255
通知する IP アドレス	IPA1	IPA2	IPA3

Local DNS Y（LDNS Y）を利用する．ここで，LDNS X と LDNS Y の IP アドレスをそれぞれ 24.0.0.1 と 150.0.0.1 とする．

ここで，サーバの ADNS が持つ IP アドレス通知テーブルが，**表 1** のとおりだったとする．この LDNS のアドレス範囲は，IP アドレスの空間を，1 つのサブネットに与えられたアドレス数で分割している．つまり，この例では，おおよそ 3 分割するように LDNS の範囲を決定している．なお，アドレス通知テーブルにセットするサーバの IP アドレスは，アクティブなサブネット A のものである．

3.3.1 正常時の挙動

サーバに対して DDoS 攻撃が行われていない正常時の場合，Client X はサーバの名前解決を行うため，LDNS X に問合せを行う．これに応じて，LDNS X は ADNS に対して再帰問合せを行うが，LDNS X の IP アドレスは 24.0.0.1 なので，ADNS はアドレス通知テーブルから IPA1 を選択し，これを応答する．同様に Client Y の利用する LDNS Y は IP アドレスが 150.0.0.1 なので，ADNS はサーバのアドレスとして IPA3 を通知する．これにより Client X と Client Y は，それぞれ IPA1 と IPA3 という異なる IP アドレスを用いてサーバにアクセスする．

3.3.2 攻撃時の挙動

図 1 において，Client Y が攻撃者（Attacker）だった場合を考える．Client Y が DDoS 攻撃を始めた後で，サーバサイドは，まず DDoS 攻撃を検出する．ただし，本提案方式では，DDoS 攻撃検出方式は提案せず，他の方式，たとえば，古典的な帯域に関する閾値ベースの攻撃検出方式やパケットのエントロピに基づく攻撃検出方式 [22] などを用いることとする．そして，現在，アクティブ設定になっているサブネットをスタンバイ設定にするとともに，攻撃を受けていない，スタンバイ設定のサブネットを 1 つ選び，これをアクティブ設定にする．ここでは，サブネット A がスタンバイ設定になり，サブネット B がアクティブ設定になる．また，これにともない，ADNS のアドレス通知テーブルの内容を変更する．このように，攻撃が行われてサーバサイドの IP アドレスを変更する一連の処理を，本論文では，IP アドレスホッピングと呼称する．そして，IP アドレスホッピング後の IP アドレス通知テーブルが，**表 2** であったとする．

この表において，通知する IP アドレスは，アクティブにしたサブネットのもの（ここでは IPB1〜IPB3）にし，LDNS のアドレス範囲は，上位 8 bit を 10 ずつずらしたものになっている．LDNS の範囲をずらす理由は，主に 2 つある．1 つはすべてのアドレスに対して平均的に攻撃を仕掛けるなどの，計画的な攻撃をしにくくするためであり，もう 1 つは，変更後の IP アドレスに追従して攻撃が行われる場合，IP アドレスホッピングを何度か繰り返すことで，Bad LDNS の範囲を絞り込んでいくためである．

表 2 新たな IP アドレス通知テーブル
Table 2 New IP address notification table.

LDNS のアドレス	10.0.0.0〜 83.255.255.255	84.0.0.0〜 156.255.255.255	157.0.0.0〜 223.255.255.255 0.0.0.0〜 9.255.255.255
通知する IP アドレス	IPB1	IPB2	IPB3

前述したとおり，Client X と Client Y は，それぞれ IPA1 と IPA3 という異なる IP アドレスを用いてサーバにアクセスする．したがって攻撃パケットの宛先 IP アドレスはすべて IPA3 宛となっているはずである．サーバサイドは DDoS 攻撃を検出すると，宛先ごとのトラフィック量から，後述の判定方法によって，IPA3 が攻撃に利用されていることを知る．そして，IPA3 が攻撃に利用されていることと表 1 のアドレス通知テーブルから，Bad LDNS は 147.0.0.0〜223.255.255.255 の範囲にあると推測できるので，攻撃が続くと思われる時間だけ，この範囲の LDNS に対して，ペナルティを与える．攻撃が続く時間の見積もりは難しいが，たとえば，近年の DDoS 攻撃の傾向 [23] を考慮し，DDoS 攻撃の平均継続時間 30 分の間，ペナルティを与えることが考えられる．ペナルティの内容もいくつか考えられ，IP アドレスホッピング後の新規 IP アドレスの通知をいっさい行わない方法や，IP アドレスホッピング前の IP アドレスを通知し続ける方法が考えられる．前者を用いた場合，DDoS 攻撃の首謀者が，名前解決結果が得られないことを不審に思い，防御機構が働いていることや，防御機構に DNS の仕組みを利用していることに気づいてしまう危険性がある．後者の場合，名前解決結果が得られるため，攻撃者が DNS 関連で不審に思う可能性は低い．また，攻撃されてスタンバイ設定になったサブネットに攻撃トラフィックが届き続けることになるが，ここに届いたパケットは FW によってすべて破棄することができる．この結果，攻撃者からすると，輻輳やシステムリソースの枯渇により DDoS 攻撃が成功したかのように見えるため，防御機構が働いていることに感づかれにくくなる．したがって，本論文では，Bad LDNS へのペナルティとして，IP アドレスホッピング前の IP アドレスを通知し続ける方法を採用する．

こうして，IP アドレスホッピング後は，正規ユーザ（Client X）には新しい IP アドレス通知テーブルに基づいたサーバの新しい IP アドレス（IPB1）が通知され，攻撃者（Client Y）には古い IP アドレス（IPA3）が通知される．このようにして，パケットを破棄する設定にした古いサブネットに攻撃トラフィックを置き去りにし，正規トラ

フィックのみを，輻輳が起こっていないサブネットで受け取るのが本提案方式の防御の挙動である．なお，攻撃者を見逃してしまい，IP アドレスホッピング後も，DDoS 攻撃が続く可能性がある．このため，本提案方式では，DDoS 攻撃が検出されなくなるまで，可能な限り，IP アドレスホッピングを繰り返す．

3.3.3 攻撃に用いられている IP アドレスの判別方法

DDoS 攻撃を検出した時点で，直近の宛先 IP アドレスごとのスループットを算出する．ちなみに DDoS 攻撃に利用されている宛先 IP アドレス（Bad IP）を正確に判別するためには，FW から宛先 IP アドレスごとのトラフィック量を得る時間間隔は短いほうが好ましい．その後，この情報に基づき，どの IP アドレスを Bad IP と判断するかが問題になる．次章の実験では，正規ユーザに用いられている宛先 IP アドレス（Good IP）の誤認をなるべく減らしつつ，一気に攻撃トラフィックを分離するために，スループットが全体の平均＋標準偏差よりも大きいものを Bad IP として判別する方法をとっている．

3.3.4 IP アドレスホッピングの限界数

提案方式によって，攻撃トラフィックを置き去りにしたサブネットは，攻撃が継続する間は IP アドレスホッピング先として再利用できない．したがって，1 度の攻撃中に IP アドレスホッピングできる回数（IP アドレスホッピング限界数）はマルチホーミングによる外部接点数 M に依存して限界数が決まる．すなわち，IP アドレスホッピング限界数 C_{AH} は以下の式で表すことができる．

$$C_{AH} = \mathrm{M} - 1$$

提案方式は，この C_{AH} 回以内に正規トラフィックに差し支えがなくなるよう，DDoS 攻撃のトラフィック量を減らす必要がある．

3.3.5 Bad LDNS の絞り込み

提案方式では，Bad LDNS が潜伏すると推定される範囲の LDNS に対して，いっせいにペナルティを与える．そのため，その範囲に属する正規ユーザが利用する LDNS（Good LDNS）も巻き添えを受ける可能性がある．このような Good LDNS を減らすため，提案方式では，攻撃を検出する度に Bad LDNS の潜伏先を絞り込む．たとえば，3.3.2 項の 1 度目の攻撃において，攻撃者は IPA1 に対して攻撃したことから，表 1 より Bad LDNS は 147.0.0.0〜223.255.255.255 だと推定できた．この範囲をさらに絞り込むため，サーバサイドはあえてペナルティを解除し，IP アドレスホッピング先のアドレスを攻撃者に通知することを選択できる．これにより，攻撃者が，新しい IP アドレスホッピング先に追従して攻撃すると，サーバサイドは攻撃者の用いる LDNS の範囲を再び得ることがでる．本節の例であれば，IP アドレスホッピング後，表 2 より，Client Y は IPB2 に対して攻撃することになる．その結果，サーバサイドは Bad LDNS の範囲を 84.0.0.0〜156.255.255.255 と推定し，この範囲と，1 度目の Bad LDNS 推定範囲との重複範囲をとると，147.0.0.0〜156.255.255.255 と絞り込むことができる．このように，IP アドレスホッピング前後で通知する LDNS のアドレス範囲をずらすことによって Bad LDNS を絞り込むことができ，次回以降この範囲にペナルティを与えることによって，巻き添えを受ける Good LDNS を減らすことができると考えられる．

3.4 利点と新規性

通常，ネットワーク型の DDoS 攻撃を防ごうとすると，スクラビングクラウドなどとの連携やサーバサイドのネットワーク以外への変更が不可欠であった．本章で提案した手法では，マルチホーム化という下準備以外はすべてサーバサイドのネットワークへの変更だけで済み，防御機構の実装についても既知の技術で既存の機器（DNS および FW）に行うだけで十分である．防御実行時に関しても，サーバサイド以外のネットワークへの要求や変更を要さず，マルチホーミングさえしていれば他の既存手法よりも導入への敷居が低い．したがって，導入の手間およびコスト，既存技術との大きな差異からくる運用ノウハウの再構築がネックだった既存手法のデメリットを持ち合わせていない点で，提案方式は新規性があるといえる．

また，他の防御手法と異なり，本提案手法は防御と同時に攻撃者が利用する Bad LDNS の特定処理が行われる．大規模な DDoS 攻撃では，サービス化された攻撃のための botnet を使用していることが多く，コンピュータウィルスに感染された IoT デバイスが攻撃源となっている事例も知られている [24]．しかし，これらの機器が bot 化されたことを，ネットワーク管理者は気づかないことが多いことが，DDoS 攻撃の問題をより解決し難いものにしている．提案手法により，Bad LDNS が特定され，この情報をセキュリティ関連組織で共有することで，botnet の特定・警告へ活かすことが期待できる．

3.5 防御対象とする攻撃

なお，提案手法はあくまで防御手法であり，攻撃の検出手法ではない．TCP SYN Flood 攻撃のように被害者サーバのリソースを枯渇させることを目的としたホスト型 DDoS 攻撃でも，UDP Flood 攻撃のようにネットワークリンクの帯域を浪費させることを目的としたボリューム型 DDoS 攻撃でも，攻撃されたことが検出されてしまえば，そのトラフィックを，利用しないサブネットへ置き去りにすることで遮断できる．したがって，防御可能な攻撃の種類は，提案方式と併用する検出手法に依存する．次章の実験では，現在最も利用されている DDoS 攻撃の 1 つである UDP Flood 攻撃が検出できることを前提として，提案方式により，この攻撃が防御可能であることを示す．

3.6 サーバのアドレス秘匿に関する懸念と考察

3.2 節の③で述べたように，提案方式は攻撃者が自分のLDNS を用いることを前提としているが，所属するネットワーク外からの名前解決問い合わせにも応じる DNS キャッシュサーバ（以下オープンリゾルバ）を用いることが考えられる．たとえば，Google などがサービスとして提供する Public DNS を利用した場合，提案方式がこれらのオープンリゾルバにペナルティを与えてしまうと，巻き添えを受ける User の規模が大きなものになることが予想される．その対策例として，このようなオープンリゾルバにも提案方式のような IP アドレス通知テーブルを実装する方法が考えられる．すなわち，オープンリゾルバは ADNS からサーバの複数のアドレスを取得しておき，アクセスしてくるノードの IP アドレスに応じて通知するサーバの IP アドレスを変更する．これによって，オープンリゾルバを利用するユーザ全体にペナルティが与えられる事態を回避できる．オープンリゾルバを利用するノードは攻撃ノードであっても名前解決結果を得るために IP スプーフィングを行わない可能性が高いので，攻撃ノードそのものの特定にも貢献できると考えられる．

その一方で，所有者が意図せずに，オープンリゾルバとなっている DNS サーバも多数存在しており，これらをすべて把握することは困難である．標的サーバの名前解決時に，これらのオープンリゾルバを用いることで，攻撃者がサーバサイドのアクティブな全 IP アドレスを容易に取得可能となってしまう懸念がある．そこで，提案方式に以下の仕組みを導入することが考えられる．

A) ホワイトリストを用意する
   Public DNS など，この仕組みの対象外としたいオープンリゾルバを，ホワイトリストに登録する．
B) オープンリゾルバか調査
   ADNS に，ホワイトリストに含まれていないリゾルバからの問合せがあると，これに回答する前に，そのリゾルバがオープンリゾルバかどうかを調べる．たとえば，保護したいサーバサイドから名前解決を要求し，これにリゾルバが返答すれば，オープンリゾルバと判定する方法が考えられる．
C) オープンリゾルバからのアクセスであれば遮断
   ホワイトリストに含まれないオープンリゾルバと分かった場合，全 IP アドレスを取得させない対策を行う．たとえば，サーバの IP アドレスを通知しない，Bad LDNS と見なして，IP アドレスホッピング後の IP アドレスは通知しないなどが考えられる．

なお，B) の調査を，ADNS そのものが行ってもよいが，攻撃者が自前でオープンリゾルバを構築し，ADNS からの問い合わせのみにエラーを返すように設定することも考えられる．これを避けるため，調査用のサーバは，ADNS とは別に用意する方が有効である．たとえば，ホップ先で使用するサブネットに調査用のサーバを設置することなどが考えられる．

ただし，この調査により，ADNS からの応答時間が増大することになる．これを短縮するため，調査結果をキャッシュすることが可能である．もしくは，ADNS が一時的に応答すると並行して，オープンリゾルバの調査も行ってしまう方法も考えられる．この場合，オープンリゾルバの IP アドレスを有効期限付きでブラックリストに入れておき，IP アドレスホッピング後からブラックリストからの名前解決要求を遮断などする方法も考えられる．

このほかにも，攻撃者が DNS に頼ることなく，事前に，主要な ISP のアドレス範囲をしらみつぶしに探索し，サーバのすべての IP アドレスを特定することが懸念される．しかし，このような無差別なスキャン行為は ISP 側の警戒を招くことにつながる．また，3.2 節の①で述べたように，提案方式では IP アドレスホッピングを行う前のサブネットからのパケットは捨てる設定（スタンバイ設定）になっているので，アドレスホッピング前のサーバの IP アドレスを特定することは困難である．

4. 実験

本章では，提案方式が，IP スプーフィング処理がなされた DDoS 攻撃に対して，防御が可能であることを示すためのシミュレーション実験を行う．数種類の攻撃方法を定義し，これらに対して防御する実験結果を通して，提案方式の長所や短所を確認する．

実験で用いる DDoS 攻撃を決定するにあたっては，Verisign のセキュリティレポートを参考にした [25]．このレポートによると，DDoS 攻撃はマルチベクトル化の気配があるものの，いまだに手軽な単一種類の DDoS 攻撃が最も多いこと，そして UDP に基づく攻撃が多いことが示されている．これをふまえて，輻輳を発生させることを目的としたネットワーク型攻撃の 1 つである UDP Flood 攻撃による，ネットワーク型の帯域攻撃を採用することとした．特に，UDP は輻輳制御機構を持たないために，輻輳環境下でも比較的高スループットとなるが，TCP は輻輳制御によりスループットが落ちてしまうことから，UDP Flood 攻撃は，TCP サービスに対して有効であることが示されている [26]．そこで，実験で用いるサーバでは，TCP によるファイル転送サービスを提供するものとする．ただし，サーバサイドのファイアウォールで，不要な UDP パケットを破棄することができるため，UDP Flood 攻撃により，サーバの処理が圧迫されることはない．

4.1 実験概要

本実験は Linux 上で，ネットワークシミュレータ ns-3 を用いて行った．本実験で用いた実験トポロジを**図 2** に示す．

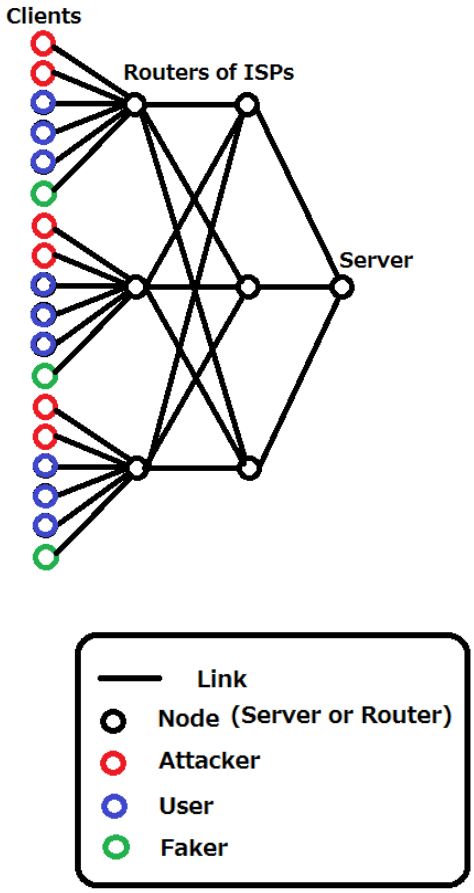

図 2 実験トポロジ
Fig. 2 Experiment topology.

表 3 送信ノードの送信トラフィック
Table 3 Transmission traffics from source nodes.

ノード	データレート	プロトコル	データサイズ
Attacker	20Mbps	UDP	—
User	4Mbps	TCP	5MB
Faker	4Mbps	TCP	5MB

◆ 送信ノードについて

図 2 左側には，サーバ（Server）にアクセスする送信ノード（Clients）が 18 台ある．各ノードは，便宜上，上から順に，node1，node2，node3，…，node18 と呼称する．そのうち，攻撃トラフィックを送信する攻撃ノード（以降 Attacker）が 6 台，サーバの提供するサービスを正しい手順で利用する正規ノード（以降 User）が 9 台，User を装ってアクセスする攻撃者側のノード（以降 Faker）が 3 台ある．

Attacker，User，Faker はネットワーク上に分散して配置されており，ISP のルータを通して，サーバにアクセスできるものとする．それぞれの送信ノードの送信トラフィックの設定を，**表 3** に示す．この表に示すとおり，Attacker は攻撃時間中ずっと UDP Flood 攻撃を行う．また，User と Faker はサーバに対して，TCP の輻輳制御が許す限り，最大 4 Mbps で 5 MB のデータを送信した後に，送信を停止する．

◆ User と Attacker/Faker の割合について

本実験では，YouTube など，User が Server に短時間の動画ファイルなどをアップロードすることを想定している．Statista によると，YouTube のアップロードサーバには 2015 年時点で 1 分あたり 400 時間分の動画がアップロードされており [27]，動画の長さが 1 本あたり平均 4 分と仮定すると，10 秒ほどで 1,000 本の動画のアップロードが行われていることになる．また，G Data Software によるブラックマーケットの調査報告書「アンダーグラウンドエコノミー」[28] によると，bot 感染サービスや botnet は，bot を 1,000 台から提供している実態が指摘されている．これらのことから，本実験では，botnet から bot を購入した攻撃者が 10 秒あたり 1,000 台規模での攻撃/偵察を，10 秒あたり 1,000 台規模の正規アクセスをうける動画サーバに対して行うことを想定し，User と Attacker/Faker の割合を 1 : 1 とした．攻撃側の Attacker と Faker の内訳については，参考となる資料が見あたらなかったため，試験的に Attacker : Faker = 2 : 1 とした．

また，それぞれの送信ノードは ISP の LDNS ではなく，内包する LDNS（自身が所属するネットワーク内にある LDNS）を用いて名前解決を行うものとする．すなわち，すべてのノードが異なる LDNS を用いる．さらに，それぞれの送信ノードが利用する LDNS の IP アドレスは，ランダムに決定した．その結果を表 3 に示す．ここに示したとおり，各々のサブネットに重複はない．また，攻撃指令を出す Zombie コンピュータが利用する LDNS の IP アドレスもランダムに決定しており，そのサブネットも，いずれの LDNS のサブネットとも重複していない．本実験で実際に用いた LDNS の IP アドレスを**表 4** に示す．

◆ サーバについて

図 2 右の Server には，サーバやそれに付随する ADNS，FW，到着トラフィック量の観測機構を含む，サーバサイドのサブネットがあるとする．このマルチホーミングにおける外部接点数は 3 とする．また，マルチホーミング先の ISP のサブネットからサーバに割り当てられるアドレスの個数の違いによる，防御性能の変化を見るために，これらのアドレスが 4 個，8 個，16 個の場合について実験する．また，このサーバは，短時間の動画などのファイルをアップロードする TCP サービスを提供しているものとする．なお，表 3 における User と Faker の設定も，それを想定した値になっている．

◆ ネットワークの各種設定

上述していない各種パラメータを**表 5** に示す．

なお，今回の実験では，ネットワークの輻輳による DDoS 攻撃を観測するため，ルータや Server の処理能力は十分に高いことを仮定し，トポロジのボトルネックがリンクになるように設定している．実際に，実験によって，User と

表 4 各 LDNS の IP アドレス
Table 4 IP address of each LDNS.

LDNS を利用するノード	*LDNS の IP アドレス*
node1	183. 54. 239. 104
node2	121. 74. 4. 24
node3	8. 61. 172. 5
node4	164. 213. 250. 85
node5	222. 203. 52. 39
node6	34. 129. 148. 94
node7	26. 20. 35. 95
node8	89. 131. 235. 57
node9	135. 136. 108. 16
node10	151. 178. 161. 4
node11	79. 127. 135. 127
node12	101. 22. 175. 18
node13	82. 181. 239. 14
node14	23. 54. 156. 68
node15	176. 191. 235. 122
node16	46. 167. 19. 43
node17	67. 181. 239. 14
node18	199. 248. 192. 13
Zombie マスタ	12. 61. 92. 34

表 5 実験パラメータ
Table 5 Experiment parameters.

設定項目	設定した内容
リンク帯域	100Mbps
リンクの伝搬遅延	10ms
キューのドロップ設定	テイルドロップ
キューサイズ	40Packets
最大パケットサイズ	1500Bytes

Attacker がいっせいに Server にアクセスすると，Server 手前のリンクがボトルネックとなって，User が Server にアクセスできなくなる．

4.1.1 攻撃方法

DDoS 攻撃の種類としては UDP Flood 攻撃を用いるが，その攻撃パケットをどのように送るかによって，様々な攻撃方法が考えられる．提案方式が LDNS に基づいた防御を行うことから，それに合わせて，攻撃方式も様々な名前解決方法に基づいたものを想定した．実験で行う攻撃方法 3 種類を以下に記す．

I. Attacker が各々行った名前解決結果を各々で用いて攻撃を行う場合

攻撃法 I では，主に，悪意を持った個人らが結託して，同じ日時にターゲットのサーバに各々 DoS 攻撃ツールを用いて攻撃を行う場合などを想定している．すなわち，botnet における名前解決結果の共有のような高度な連携を行わない攻撃とする．この攻撃法では，悪意を持った個人らを Attacker と置く．攻撃側は Faker を利用しないものとし，実験中に，Faker はいっさいのパケットを送信しない．

II. Attacker が Faker の名前解決結果を用いない場合と用いる場合

攻撃法 II の実験目的は，Faker が Attacker へ名前解決結果を共有することによる影響の観測である．Faker による名前解決結果を用いる場合について以下に記述する．Attacker が攻撃を行った後，防御機構が働いているのかを調査するために，Attacker とは異なるネットワークに接続した Faker が，User を装ってサーバへアクセスを行う．もし，アクセスできなければ，攻撃が成功していると判断できるが，Attacker がアクセスできないにもかかわらず，Faker はアクセスできたとすると，攻撃トラフィックを分離する防御機構が働いていると判断できる．そこで，この攻撃法では，Faker によってアクセスできた（User だけが知らされているであろう）IP アドレスに対して，Attacker が攻撃する．

この IP アドレスの共有方法として，Faker が，あらかじめ決められた掲示板などに，アクセス可能だった IP アドレスを書き込み，Attacker が，定期的にこれを調査する方法などが考えられる．ただし，Attacker や Faker らによる掲示板へのアクセスが集中してしまうと，掲示板への DDoS 攻撃になりかねない．したがって，Attacker らの掲示板へのアクセスは，時間的に分散させて行わせる必要がある．これは，攻撃側がターゲットの防御機構が働いていることに気づいても，すぐに変更後の IP アドレスへの攻撃ができないことを意味している．つまり，攻撃側は提案方式の IP アドレスホッピングに追従するのに時間を要し，その間は，User が Server へアクセスできることになる．

また，攻撃アルゴリズムとして，サーバにアクセスできた Faker らが掲示板に順次アップロードしていく IP アドレスの中から，Attacker は最も新しくアップロードされた IP アドレスを攻撃対象として用いるものとする．

III. Zombie マスタなどの単一ノードが名前解決した結果を Attacker らが用いる場合

攻撃法 III は，2001 年に猛威をふるった CodeRed のように，あらかじめ単一ノードが名前解決を行っておき，その IP アドレスに対して，同じ日時に攻撃を行うよう仕組んだワームを，インターネット上の多数のマシンに感染させるといった手段を取る攻撃法である．

本実験では，Zombie マスタが名前解決を行って得たターゲットの IP アドレスや攻撃時刻・攻撃命令を，

あらかじめ決めておいた掲示板に対して書き込み，Zombie コンピュータである Attacker らに掲示板の攻撃命令を，定期的に見に行かせて，攻撃を行うモデルを想定している．なお，本攻撃法では Faker を使用しないので，攻撃法 I と同様に，Faker はいっさいのパケットを送信しない．

さらに攻撃法 I，II，III について，

A) 攻撃中は最初に名前解決した結果の IP アドレスを使い続ける場合
B) 攻撃中も再度の名前解決により IP アドレスホッピングに追従する場合

の 2 通りを考える．つまり，本実験では 6 通りの攻撃方法について実験を行う．便宜上，以下では，I における A の条件での攻撃法を攻撃法 I-A，などと記すことにする．ただし，攻撃法 II-A は，Attacker も Faker も攻撃中は最初に自身が解決した名前解決結果を用いて Server にアクセスを行うことを意味する．攻撃法 II-B は，当初は攻撃法 II-A と同じだが，IP アドレスホッピング後に，Faker が再度名前解決を行い，Attacker がその Faker の名前解決結果を用いて攻撃を行うところが異なる．

4.1.2 観測区間

実験中，サーバサイドは受け取るトラフィック量の明確な変化に基づいたいくつかの状態を遷移することになる．ここではサーバサイドのおかれる状態を，“正常時”，“攻撃時”，“防御直後”，“防御後非攻撃時”，“防御後再攻撃時”の 5 つとして，それぞれを以下のように定義する．

正常時とは，User が Server に対してアクセスしている状態である．このとき，Server に対して攻撃トラフィックは流れていない．しかし，Faker が User を装い，情報を収集するなどの水面下での動きはあるものとする．したがって，攻撃法 II において，実験中，Faker は User のように名前解決を行い，Server に対して，User と同様のパケットを送信する．また，比較のため，実験では，開始直後に，この区間を 10 秒間行う．

攻撃時とは，User が Server に対してアクセスするだけでなく，Attacker が Server に対して攻撃トラフィックを送り付ける状態である．Server は，User と Attacker からの大量のトラフィックにより，User に対して，正常なサービスの提供を行えなくなる．また，Server が DDoS 攻撃検出・防御機構を備えている場合，Server がこの状態にある時間は，攻撃検出時間により決定される．実験では，Attacker が攻撃開始してから，攻撃を検出するまでの時間を 10 秒とする．

なお，厳密には攻撃時と次の防御直後の間には，防御移行状態が含まれる．この防御移行状態は，提案方式では IP アドレスホッピングをしている状態であり，この間はいかなるパケットも Server に届かない．しかし，Server のネットワークインタフェースのアクティブ状態とスタンバイ状態の変更や，ADNS のアドレス通知テーブルの更新を行うのみであるため，その処理にはあまり時間はかからない．このため，本実験ではこの状態を無視する．

防御直後は，Server に対する名前解決情報の DNS キャッシュの有効期限が満了した User が，順次，Server の IP アドレスホッピング先の IP アドレスに，アクセスする状態である．キャッシュがあるために，防御直後はどのノードもすぐにサーバへアクセスできない場合がある．防御直後に User に早くアクセスしてほしい場合は，ADNS のコンテンツの TTL を短く設定しておく．本実験では，この TTL を 10 秒に設定した．一方，ペナルティを課せられた LDNS には，ペナルティが終わる時間まで，ADNS から変更後の IP アドレスが通知されない．このため，これに巻き込まれた User や，LDNS 以外の名前解決手段を持たない Attacker は，Server にパケットを送ることはできない．本実験では，このペナルティが終わるまでの時間を 30 分に設定した．この理由は，近年の DDoS 攻撃の動向に関するレポートにおいて，ほとんどの攻撃が 30 分かそれ未満で攻撃を終了していたからである [23]．このペナルティを与える時間はサーバサイドの DNS から適宜変更できるため，実環境で運用する際は，その時々の攻撃動向に合わせた時間を設定することが望ましい．

防御後非攻撃時は，防御機構が働いた後，ADNS の TTL が満了し，ペナルティに巻き込まれていない，すべての User が Server にアクセスできるようになってから，防御機構によって攻撃が空振りに終わっていることを Attacker が気づいていない状態，もしくは気づいても変更後の IP アドレスに追従して攻撃パケットをサーバに送れない状態である．この時間帯は User（と Faker）だけが Server にアクセスできる．この状態が長ければ長いほど良く，提案方式でも，攻撃を防御していることを Attacker が感づきにくいように工夫を行っている．

防御後再攻撃時は，防御機構によって攻撃が空振りに終わっていることに Attacker が気づき，変更後の IP アドレスに対して攻撃パケットを送信する状態である．この状態では，再び User と Attacker からのトラフィックが入り混じった状態で，Server に到達する．ここで，防御機構によって十分に攻撃トラフィックを分離できなかった場合，再び IP アドレスホッピングが行われる．

実験では，以上の状態における User と Attacker のスループットを観測し，それぞれを比較することでその有用性や性質を確認する．今回の実験では，それぞれの状態が 10 秒ずつになるように設定しているので，それぞれの区間を観測し，その結果をグラフにまとめる．

4.1.3 攻撃に関する判定方法

◆ 攻撃の検出方法について

提案方式では，攻撃の検出方法については対象外なので，実験では Attacker が攻撃してから 10 秒後に，攻撃が

100%検出できると仮定している．

◆ 攻撃に利用されている IP アドレスの判定方法

ISP の各サブネットにおける Server のアドレス数を N とおく．さらに，アクティブ設定のサブネットにおける Server の各 IP アドレスと，各 IP アドレスの合計スループットを IP_i，T_i（ただし i = 1, 2, 3, . . . , N）とおく．さらにそれらの平均スループットを μ，標準偏差を σ とおく．以下の条件式を満たす IP_i をすべて攻撃者に利用されているものと判断する．

$$T_i \geqq \mu + \sigma$$

平均より大きいスループットを記録している IP アドレスではなく，平均＋標準偏差より大きいスループットを記録している IP アドレスを悪用されていると判断することで，送信者が偏っている場合において，悪用されていない IP アドレスが悪用されていると誤認されることを防ぐ．

また，各宛先 IP アドレスに対するスループットがすべて等しい場合は，Attacker がこれらのアドレスへ均等に攻撃している可能性があるので，全アドレスのうち（アドレス数 $\times\frac{1}{4}$）個のアドレスをランダムに選択して，ペナルティを与えることとする．なお，この比率は，いくつかの値で実験したときの，最も性能が良かったものを採用している．

4.2 実験結果

各攻撃法に対して，提案方式による防御を行ったときのスループットの変化と，IP アドレスホッピング後のサブネットにアクセスできている Attacker と User の割合を示す．防御後のスループットが正常時のスループットに近いほど好ましい結果といえる．また，提案方式によって IP アドレスホッピングした後に，これに追従して，ホッピング後のサブネットにアクセスしているノードのことを，生存している（通信できる）と表現する．すなわち，User の生存率が高く，Attacker の生存率が低い結果が好ましいといえる．

なお，すべてのグラフについて，それぞれのサブネットに対してサーバが用いるアドレス数が 4・8・16 のときの結果を同時に記載している．また，スループットに関するグラフは，各サーバの状態ごとの攻撃スループットの合計と正規スループットの合計の積み上げグラフになっている．ただし，これらのスループットには，名前解決によるスループットは含まれない．

4.2.1 攻撃法 I-A について

攻撃法 I-A に対して提案方式を用いて防御を行ったときの，合計スループットを**図 3** に示す．なお，本章の実験におけるサーバは，ファイルのアップロードするサービスを提供するので，User の合計スループットを減らすことができたかどうかによって，DDoS 攻撃の成否が決まると考えられる．

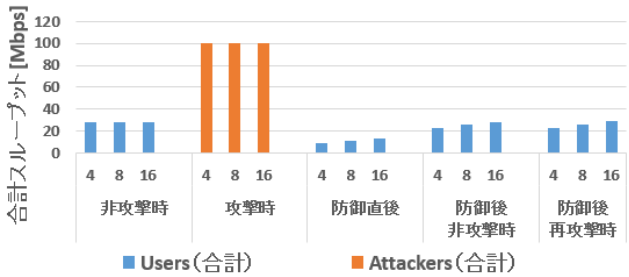

図 3 攻撃法 I-A の場合の合計スループットの変化

Fig. 3 Changes in total throughput under attack method I-A.

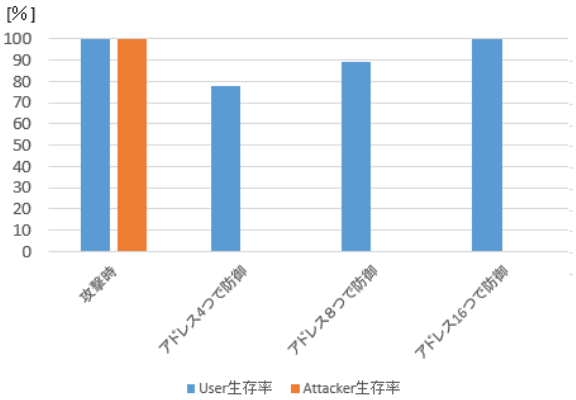

図 4 攻撃 I-A 防御後の最終的なノード生存率

Fig. 4 Final node survival rate after defended from attack method I-A.

非攻撃時は，User のトラフィックの合計スループットは 28.58 Mbps となった．攻撃時のデータは，激しい UDP Flood の影響が顕著に表れている．Attacker による，輻輳制御を持たない UDP パケットが帯域をほぼすべて使い切るなか，User のトラフィックは，TCP の輻輳制御とボトルネックでのパケットロスのため，スループットが 0.16 Mbps にまで低迷している．防御直後は，LDNS のキャッシュの影響で，ホップ後のアドレスにアクセスできる User は少ないが，その後は，時間の経過とともにスループットが改善されていき，攻撃トラフィックをすべて分離できていることが分かる．防御後の合計スループットはアドレス数によって異なり，アドレス数 4，8 のときの合計スループットは 16 のときよりも低くなってしまっている．この理由は，アドレス数が少ないと，Good LDNS を Bad LDNS と誤認してペナルティを与えてしまう割合が増えてしまうからだと考えられる．つまり，アドレス数が少ないと**図 4** に示すように，正規ノードの生存率が下がってしまうことに起因している．

攻撃時までは各ノードは通信できる状態だが，IP アドレスホッピング後はアドレス数に応じて User もいくつか通信できない状態に追いやられている．新しい IP アドレスに追従できない DDoS 攻撃であるため，IP アドレスホッピングの効果は大きく，防御後は攻撃ノードをすべて分離できている（アドレスホッピング前のサブネットにすべて

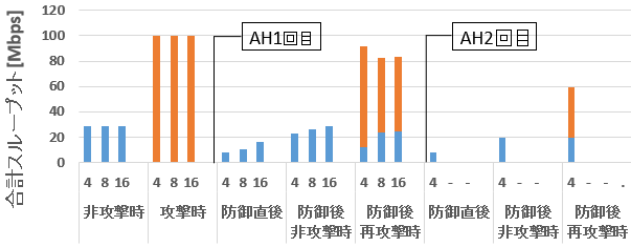

図 5 攻撃法 I-B の場合の合計スループットの変化
Fig. 5 Changes in total throughput under attack method I-B.

置き去りにできている）．

アドレス数 16 のとき，防御後のスループットは正常時並になった．この攻撃法に関しては，提案方式により，アドレス数によらずすべて防御可能であった．ただしアドレス数が少ないと防御後の正規ユーザのトラフィック量も減ってしまうことが分かった．なお，3.3.2 項で述べたとおり，攻撃ノードが利用した Bad LDNS へのペナルティは “攻撃が続くと思われる時間” だけ続くので，ここで減少した正規ユーザのトラフィックへのペナルティも，その時間経過後に解除される．

4.2.2 攻撃法 I-B について

攻撃法 I-B に対して，提案方式を用いて防御を行ったときの合計スループットの推移を**図 5** に示す．なお，グラフ中に記載されている AH は IP アドレスホッピングを示す．

図 5 の結果は，防御後非攻撃時までは攻撃法 I-A とあまり変わらない．際立った変化があるのは，防御後の再攻撃時である．攻撃法 I-B は IP アドレスホッピングが行われていると Attacker が気づいた時点で再度名前解決を行い，攻撃を仕掛ける．1 回目の攻撃時は，帯域をほぼ埋め尽くすほどの攻撃スループットが観測されたものの，防御後は，攻撃トラフィックの多くをアドレスホッピング前のサブネットに置き去りにできており，アドレス数が 8，16 のときは防御後の再攻撃時も，ボトルネックの帯域 100 Mbps に収まるように，User のトラフィックをなるべく減らさず Attacker のトラフィックを減らすことに成功している（2 回目のアドレスホッピングはしていないので，図 5 には記載していない）．ただし，アドレス数が 4 つの場合に関しては，1 回目の IP アドレスホッピングで十分に Attacker を分離できなかったため，User のスループットが芳しくない．したがって 2 回目の IP アドレスホッピングを行っており，それによりやっと十分な数の Attacker のトラフィックを移動前のアドレスに置き去りにし，その分，User のトラフィックを送る余裕ができている．アドレス数 4，8，16 における防御後再攻撃が行われた状況下での最終的な合計スループットは，19.67 Mbps，23.31 Mbps，24.25 Mbps である．非攻撃時の合計スループットが 28.58 Mbps あるから，アドレス数 8，16 の場合は少ない IP アドレスホッピング回数で非攻撃時並のスループットに回復できていると評価できる．

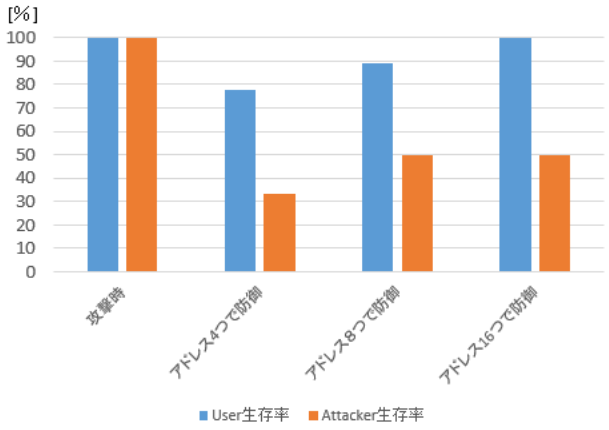

図 6 攻撃 I-B 防御後の最終的なノード生存率
Fig. 6 Final node survival rate after defended from attack method I-B.

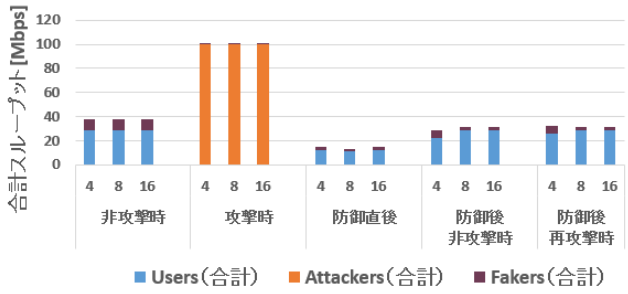

図 7 攻撃法 II-A の場合の合計スループットの変化
Fig. 7 Changes in total throughput under attack method II-A.

最終的なノードの生存率を**図 6** に示す．この図から，アドレス数 16 のとき，User はいっさい分離されておらず，Attacker を半数分離できていることが分かる．アドレス数 4 の場合，アドレスホッピングを 2 回行っているため，アドレス数 8，16 のときよりも Attacker を減らせているが，同時に巻き添えを受けた User の割合も多く，User の生存率は 8 割未満となっている．

4.2.3 攻撃法 II-A について

攻撃法 II-A に対して提案方式を用いて防御を行ったときの合計スループットを**図 7** に示す．

非攻撃時には User の合計スループットは 28.47 Mbps あり，Faker の合計スループットは 9.60 Mbps ある．攻撃時にはこれらが 0.098 Mbps，0.029 Mbps まで低下する．これに対して，Attacker のスループットはおよそ 100 Mbps と，帯域を埋め尽くしている．しかし防御機構が働いて IP アドレスホッピングして以降は徐々に User，Faker のスループットは改善している．防御後非攻撃時の User と Faker の合計スループットはアドレス数 4 のとき 22.53 Mbps，5.93 Mbps，アドレス数 8 および 16 のときは 28.41 Mbps，3.12 Mbps まで回復している．またこの後，再攻撃を行ってきても古いアドレスを用いているのでスループットは防御後正常時から変わっていない．アドレス数 8，16 のと

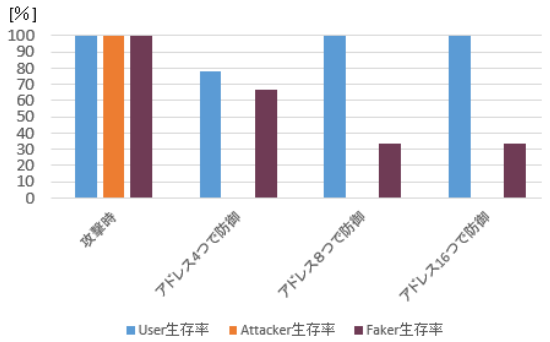

図 8 攻撃 II-A 防御後の最終的なノード生存率

Fig. 8 Final node survival rate after defended from attack method II-A.

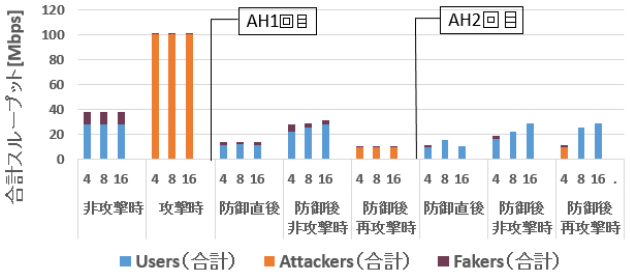

図 9 攻撃法 II-B の場合の合計スループットの変化

Fig. 9 Changes in total throughput under attack method II-B.

きの最終的な User のスループットは，非攻撃時と変わらないくらいにまで回復できている．この理由は，Attacker が Faker の LDNS の名前解決結果を用いたので，Faker の LDNS が Bad LDNS となり，Faker と User の区別ができたためである．

最終的なノードの生存率を**図 8** に示す．この図よりアドレス数にかかわらず，Attacker をすべて分離することに成功していることが分かる．また，アドレス数が少ない場合は User の生存率が低く，Faker の生存率が高くなってしまっているのに対し，アドレス数が 8，16 の場合には User の生存率が高く，Faker の生存率を低くすることに成功している．

4.2.4 攻撃法 II-B について

攻撃法 II-B に対して提案方式を用いて防御を行ったときの合計スループットを**図 9** に示す．非攻撃時，攻撃時の合計スループットに関しては攻撃法 II-A と同値である．一回目の IP アドレスホッピングの後，徐々にスループットの回復が見られ，十分に時間が経過した防御後正常時には User のスループットが大幅に回復する．このときの User の合計スループットはアドレス数 4 ならば 22.53 Mbps，アドレス数 8 ならば 25.57 Mbps，アドレス数 16 ならば 28.41 Mbps である．アドレス数が多いほど非攻撃時並のスループットに近い値まで回復している．一方，Faker の合計スループットはアドレス数 4 ならば 5.92 Mbps，アドレス数 8 ならば 3.01 Mbps，アドレス数 16 ならば 3.12 Mbps である．しかし，この後再攻撃が行われると，User のスループット

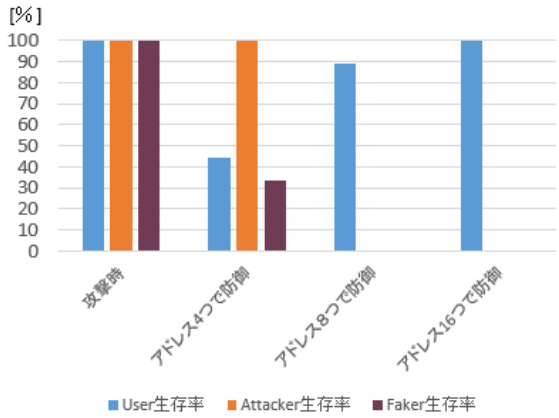

図 10 攻撃 II-B 防御後の最終的なノード生存率

Fig. 10 Final node survival rate after defended from attack method II-B.

は，アドレス数 4 のとき 0.15 Mbps，アドレス数 8 のとき 0.11 Mbps，アドレス数 16 のとき 10.16 Mbps と激減する．これは Faker が分離されずに残っているために Attacker が Faker の名前解決結果を用いることができており，攻撃の勢いが 1 度目から下がっていないためだと考えられる．すなわち，アドレス数が 4，8，16 いずれの場合も，Faker の LDNS を Bad LDNS として認識してペナルティを与えるためには，IP アドレスホッピング数は 1 回では足りず，もう一度 IP アドレスホッピングを行うこととなった．これにより，2 回目も防御直後から徐々に合計スループットは回復していく様子がうかがえる．この後の防御後再攻撃時，アドレス数が 4 の場合のみまだ Faker がペナルティを与えられず残っていたために再度の激しい攻撃にあい，User の合計スループットはわずか 0.16 Mbps しか出ていない．今回の実験における IP アドレスホッピングの限界数は 2 回であるため，アドレス数 4 つの場合に攻撃法 II-B を受けた場合，アドレスホッピング限界数内に防御できないという結果になった．しかしアドレス数 8，16 においては，**図 10** より，最終的に IP アドレスホッピングを 2 回行い，Faker ノードをすべて分離できているので Attacker の新 IP アドレスへの追従手段がなくなり，攻撃トラフィックを止めることに成功している．このときの User の合計スループットは，アドレス数 4 のときは 25.72 Mbps，アドレス数 8 のときは 28.58 Mbps であり，非攻撃時並のスループットにまで回復できていることが分かる．

4.2.5 攻撃法 III-A および III-B について

攻撃法 III-A および III-B に対して提案方式を用いて防御を行ったときの合計スループットを**図 11** に示す．いずれの攻撃法も，得られた結果が同じであったため，両方の結果をまとめて載せている．また，防御を行った後のノード生存率を**図 12** に示す．非攻撃時と攻撃時のスループットに関しては攻撃法 I-A などと同値である．IP アドレスホッピング 1 回で名前解決を行う単一ノード（ここでは

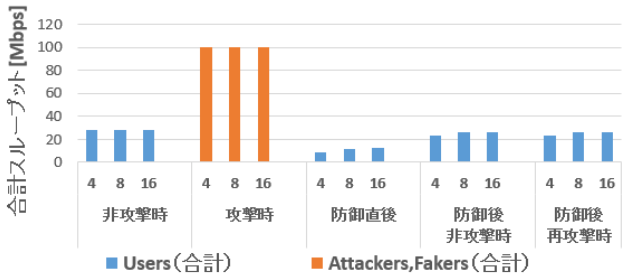

図 11 攻撃法 III-A および III-B の場合の合計スループットの変化
Fig. 11 Changes in total throughput under attack methods III-A and III-B.

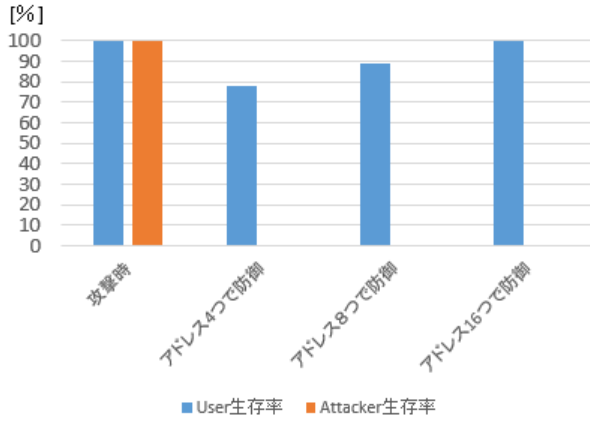

図 12 攻撃 III-A および III-B 防御後の最終的なノード生存率
Fig. 12 Final node survival rate after defended from attack methods III-A and III-B.

Zombie マスタ）の LDNS を分離できるので，アドレス数にかかわらず攻撃トラフィックをすべて分離することに成功している．

防御後非攻撃時・再攻撃時のスループットはアドレス数 4 のときで 23.01 Mbps，8 のときで 25.74 Mbps，16 のときで 25.74 Mbps であった．これまで同様，アドレス数が少ないとアドレス数が多いときに比べて User の生存率が下がるため，防御後の User のスループットも低くなってしまっている．

4.3 評価

アドレス数に関係なく，攻撃中に再度の名前解決を行わない攻撃はすべて IP アドレスホッピング 1 回で防御可能であった．

今回の攻撃法で唯一防御ができなかった条件は，攻撃法 II-A に対して各サブネットにアドレス 4 つずつを用いた場合であった．アドレス数 4 のときの提案方式では，User に誤ってペナルティを与えてしまう割合が多く，最終的に攻撃トラフィックを輻輳が起こらないレベルにまで抑え，DDoS 攻撃への防御ができたとしても，User の被害が大きくなる傾向があった．対照的に，アドレス数 8，16 の場合の提案手法はすべての攻撃を IP アドレスホッピング制限回数（今回の場合 2 回）以内に防御することができた．アドレス数 8，16 の場合で IP アドレスホッピングが 2 回必要だったのは，攻撃法 II-B だけであり，攻撃者らが User を装う Faker を用い IP アドレスを巧みに共有しつつ，攻撃中も再度の名前解決を行うような攻撃を行われると，提案手法において IP アドレスホッピング数が比較的多くなってしまうことが分かった．アドレス数 8，16 の場合は，防御後に再攻撃が行われたとしても，User に非攻撃時並のスループットを確保させることができた．

実験を通して，Server に十分なアドレス数を持たせることによって，IP スプーフィングされた Flood 攻撃を防御可能であることを示した．外部接点数が 3 以上で，アドレス数が十分であればここで示した攻撃法については防御が可能であった．

4.4 考察

最後に，実験を通して得られた，提案方式において検討すべき項目として，IP アドレスをずらす範囲，IP アドレスとサブネット数の関係について，そして IP アドレスホッピング回数の制約，Overfort との比較について考察する．

4.4.1 IP アドレスをずらす範囲について

本実験では，3.3.2 項で述べたように，IP アドレス通知テーブルの，LDNS の IP アドレス範囲の上位 8 ビットを，IP アドレスホッピングにともない，10 ずつずらしている．ずらす範囲は大きすぎても小さすぎても，Bad LDNS を特定するまでの IP アドレスホッピングの回数が増加する懸念があったが，4.2 節の結果では，ランダムに選択された LDNS の IP アドレスから，Bad LDNS を IP アドレスホッピング 1 回で特定できていることから，このずらす量は適切であったと考えられる．

このずらす量の影響は，IP アドレス数との兼ね合いによるところが大きいと考えられる．すなわち，IP アドレス数が多いと，IP アドレス通知テーブルの持つ 1 要素ごとのアドレス範囲もより狭くなるため，ずらす範囲を小さくして小刻みに Bad LDNS を特定する方が，効率が良いと考えられる．たとえば，IP アドレスが 16 個のときに，IP アドレス通知テーブルの LDNS のアドレスの範囲を，表 1 のように，クラス A から C までの範囲を 16 等分したとする．このとき，LDNS のアドレスの 1 番目の範囲（上位 8 ビットが 0〜13 のもの，以下同様）と 2 番目の範囲（14〜27）は，IP アドレスホッピングを 1 回すると，ずらす範囲を 10 とした場合は 10〜23 と 24〜37 となり，これにより，0〜27 の範囲は，0〜9，10〜13，14〜23，24〜27 に分離され，ペナルティを与える対象範囲は，上位 8 ビットが最小で 4 個（10〜13 や 24〜27），最大で 10 個（0〜9 や 14〜23）に集約されることとなり，表 4 の node7 と node14 は分離される．さらに IP アドレスホッピングをもう 1 回すると，0〜5，6〜9，10〜13，14〜19，20〜23，24〜27 となり，最大で 6 個に集約される．なお，LDNS のアドレスの範囲の幅

が 14 であるので，ずらす範囲 n が 14 以下ならば，$14-n$（$n=10$ のときは 4）ずらすのと同様な結果が得られる．

一方，ずらす範囲を 16 とすると，0〜13 と 14〜27 の範囲は，IP アドレスホッピングを 1 回した後に 16〜29，30〜43 となり，LDNS のアドレスの範囲の 15 番目（196〜209）と 16 番目（210〜223）が 212〜223 と 0〜1，および 2〜15 になることに注意すると，0〜27 の範囲は，0〜1，2〜13，14〜15，16〜27，30〜31 に分離され，ペナルティを与える対象範囲は，上位 8 ビットが最小で 2 個，最大で 12 個に集約されることになる．LDNS のアドレスの範囲の幅が 14 であるので，ずらす範囲 n は，n mod 14（$n=16$ のときは 2）ずらすのと同様であり，小さく絞れる範囲と大きい範囲が交互に生じるため効率が悪い．表 4 の node7 と node14 は 16〜27 に含まれるため，これらが別な範囲の Bad LDNS であることを認識するためには，IP アドレスホッピングをもう 4 回する必要がある．

逆に，IP アドレス数が少ないときは，ずらす範囲が小さいと IP アドレスホッピングの前後で重複する範囲が広く残る部分があり，効率が良くない．たとえば，IP アドレス 4 個のとき，LDNS のアドレスの範囲は上位 8 ビットが 56 個ずつとなり，表 4 の node3，node6，node7，node14，node16 は 0〜55 の範囲に含まれる．ずらす範囲が 10 の場合は，IP アドレスホッピングが 1 回で 10 個（0〜9）と 46 個（10〜55）に分かれ，node3 はほかとは区別できるのに対し，ずらす範囲を 4 とした場合は，4 個（0〜3）と 52 個（4〜55）に分かれるため，区別できる node はない．

以上のように，ずらす範囲が 10 の場合，IP アドレス数が 4，8，16 のいずれの場合にも，1 度の IP アドレスホッピングによって，適度な重複範囲が得られたため，良好な結果が得られたと考えられる．今後，IP アドレス数に基づいて，ずらす量を定量的に決定する方法を検討する必要がある．その際，2 分探索的に Bad LDNS を特定していくような仕組みが好ましいと思われ，上位 8 bit をずらす以外の手法についても検討の余地がある．

4.4.2 IP アドレスの数とサブネット数（外部接点数）

4.4.1 項で述べたように，IP アドレスの数が少ないと一度に多くのノードがペナルティを与えられるが，これにともない正規のユーザが巻き添えを受けてペナルティを受ける可能性も多くなる．逆に IP アドレスの数が多いと，攻撃者を中心に細かくペナルティを与えられるので，正規ユーザの巻き添えを受ける可能性を低くし，攻撃者へペナルティを与える可能性を高くすることができるが，うまく攻撃者を分離できなければ，再度 IP アドレスホッピングを行う必要がある．

ここで，1 つのサブネットで利用できる IP アドレス数を 2^n，利用できるサブネット数（外部接点数）を m とすると，分割可能な IP アドレスの範囲は $m \cdot 2^n$ 個となる．この範囲のいずれかの中に可能な限り多くの Bad LDNS を集約し，Bad LDNS を含まない範囲にすべての Good LDNS を集約することが理想である．たとえば，IP アドレスを 16 個用いた場合，上位 8 ビットが異なる IP アドレスを互いに異なる範囲に入れるためには，最低でも $224/16=14$ のサブネットが必要になる．このように，IP アドレスの範囲を均等に分割するには多くのサブネットが必要となり，新規 IPv4 アドレスが枯渇している現在，1 つのサブネットから潤沢に IP アドレスを取得するのは高コストになることが懸念される．このことから，攻撃を受けていない範囲は IP アドレスホッピングをしても分割しないなど，IP アドレスの効率の良い利用方法が求められる．

4.4.3 IP アドレスホッピング回数の制約について

実験中でアドレス 4 つの場合に攻撃法 II-B が，IP アドレスホッピング限界回数以内に防御できなかったが，Attacker の数が多い場合，アドレス数が 16 の場合でも，IP アドレスホッピング限界回数の制約が問題になると考えられる．限界回数がある理由は，攻撃トラフィックを IP アドレスホッピング前のサブネットに置き去りにするため，同じ攻撃中にこのサブネットを IP アドレスホッピング先として再利用できなくなるからである．その対策として，攻撃を受けたサブネットを管理する ISP と連携して本提案方式を実行することが考えられる．すなわち，IP アドレスホッピング前に DDoS 攻撃がある旨を ISP に報告し，ISP 内でこれらのトラフィックをブラックホールルーティングしてもらうことで，ISP は無駄なトラフィックを減らすことができるとともに，サーバサイドは攻撃中に利用した IP アドレスホッピング前のサブネットを，再利用することが可能になる．また，Bad LDNS に対するペナルティとして，通知する IP アドレスを ISP などが保有するブラックホールルータのものにすることも考えられる．これにより，ISP 側のインフラに対する変更点も少なくて済む．

4.4.4 Overfort との比較

2.8 節で述べた Overfort と比較するため，本章で行った実験を Overfort に適用した場合を考える．実験では，Server はサブネット 3 つにそれぞれ最大で 16 個の IP アドレスを用いていることから，公平性のため，Overfort でも 48 個のアドレスを用いて防御すると想定する．この内，サーバサイドに Public IP と Secret IP のための 2 つの IP アドレスを割り当て，残りの 46 個の IP アドレスを用いて ISP にゲートウェイを設置するものとする．文献 [2] では，これらは異なるサブネットに設置することが望ましいとしており，本考察でもそのように設置すると仮定する．

さて，攻撃時には，Public IP は放棄され，ゲートウェイのいくつかがアクティブになる．LDNS は ADNS からこれらのいずれかのゲートウェイの IP アドレスが通知され，LDNS は自分の近隣（k 個の AS 以内で，k は 1 などなるべく小さい正整数）のクライアントのみに，このゲートウェイの IP アドレスを通知する．これによって，User から送

られたパケットは，ゲートウェイを介して，Secret IP にトンネリングされる．ただし，LDNS の通知範囲の制約から，3.6 節で述べたオープンリゾルバの対策は可能であっても，Public DNS の利用は難しいと考えられる．

この状況下で本章の攻撃を継続した場合，Attacker が自らの LDNS を経て知ったゲートウェイが攻撃されるものの，このゲートウェイを停止すれば，Secret IP が漏えいしない限り，Server に攻撃が届かなくなることから，本章の実験の攻撃は防げると思われる．また，攻撃を受けたゲートウェイの IP アドレスを通知した LDNS サーバには，提案方式と同様に，新たなゲートウェイの IP アドレスが通知されないペナルティが与えられるため，巻き添えを受ける User がいることも提案方式と同様である．さらに，faker を用いることで，正規の手順で得たゲートウェイの IP アドレスを Attacker らと共有することができる．これにより，各 faker が知ったゲートウェイも攻撃対象となり，いくつかのゲートウェイに分散したトラフィックを，合計して Server に到達させることができるため，ゲートウェイにペナルティを受けない程度の負荷をかけつつ，Server に高負荷をかけることで，攻撃が成功してしまう懸念がある．

また，Overfort では，トンネリングにかかるオーバヘッドがあるほか，2.8 節で述べたとおり，46 もの異なる ISP にゲートウェイを設置するのはコストが高くなることが予想され，また，Secure IP が漏えいすることにより防御が破綻することから，提案方式は Overfort と同等以上の防御能力を持ちつつ，より少ないコストで実現が可能であると考えられる．

5. まとめ

本論文では，IP トレースバックなどのような，既存のインフラに多くの変更点を必要としない，IP スプーフィングに強い DDoS 攻撃防御方式を提案した．そして，実験を通して，アドレス数・外部接点数が十分である場合に，ネットワーク型 Flood 攻撃に対して防御が可能であることを示した．また，アドレス数・外部接点数が十分であれば，防御後の再攻撃時も正規トラフィックは非攻撃時並のスループットでサーバにアクセスできることも示した．

今後の課題として，UDP Flood 以外の DDoS 攻撃を防げることを示すことや，ISP との連携による防御策の有用性の検証などがあげられる．

参考文献

[1] 高田美紀：ISP における DoS/DDoS 攻撃の検知・対策技術 (2013)，入手先 ⟨https://www.nic.ad.jp/ja/materials/iw/2013/proceedings/s2/s2-takata.pdf⟩.

[2] Khor, S.H. and Nakao, A.: Overfort: Combating DDoS with Peer-to-peer DDoS Puzzle, *Proc. IEEE International Symposium on Parallel and Distributed Processing*, pp.1–8 (2008).

[3] 岩永崇裕，木村成伴：マルチホーミングにおける IP アドレスホッピングを用いた DDoS 攻撃防御方式，情報処理学会第 78 回全国大会講演論文集，5R-02 (2016).

[4] Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T. and Strayer, W.T.: Hash-based IP Traceback, *Proc. ACM SIGCOMM Conference '01*, pp.3–14 (2001).

[5] Mopari, I.B., Pukale, S.G. and Dhore, M.L.: Detection of DDoS Attack and Defense-Against IP Spoofing, *Proc. International Conference on Advances in Computing, Communication and Control*, pp.489–493 (2009).

[6] Mukaddam, A., Elhajj, I., Kayssi, A. and Chehab, A.: IP Spoofing Detection Using Modified Hop Count, *Proc. 2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pp.512–516 (2014).

[7] Wang, X., Li, M. and Li, M.: A Scheme of Distributed Hop-count Filtering of Traffic, *Proc. IET International Communication Conference on Wireless Mobile and Computing* (*CCWMC*), pp.516–521 (2009).

[8] Yaar, A., Perrig, A. and Song, D.: Pi: A Path Identification Mechanism to Defend against DDoS Attacks, *Proc. 2003 IEEE Symposium on Security and Privacy*, pp.1–15 (2003).

[9] Collins, M. and Reiter, M.K.: An Empirical Analysis of Target-Resident DoS Filters, *Proc. 2004 IEEE Symposium on Security and Privacy*, pp.103–114 (2004).

[10] 志田雄哉，木村成伴，海老原義彦：DDoS 攻撃のためのパス識別子メカニズムにおけるプロトコル単位でのフィルタリング方式の提案，情報処理学会全国大会講演論文集，Vol.68, No.3, pp.673–674 (2006).

[11] Ioannidis, J. and Bellovin, S.M.: Implementing Pushback: Router-Based Defense against DDoS Attacks, *Proc. Network and Distributed System Security Symposium*, pp.1–12 (2002).

[12] 寺田剛陽，双紙正和，宮地充子：Pushback 機構の一提案とそのモデル化に向けて，情報処理学会論文誌，Vol.45, No.8, pp.1948–1953 (2004).

[13] 金子陽一，木村成伴，海老原義彦：Pushback 方式を導入した Path Identification 方式による DDoS 攻撃防御対策の提案，信学技報，Vol.106, No.420, IN2006-132, pp.109–114 (2006).

[14] 井熊一博，木村成伴，海老原義彦：DDoS 攻撃の為の送信元識別子を用いたフィルタリング方式，情報処理学会全国大会講演論文集，Vol.73, No.3, pp.493–494 (2011).

[15] Okada, K., Hazeyama, H. and Kadobayashi, Y.: Oblivious DDoS Mitigation with Locator/ID Separation Protocol, *Proc. 9th International Conference on Future Internet Technologies*, No.8 (2014).

[16] Hsiao, H.-C., Kim, T.H.-J., Yoo, S., Zhang, X., Lee, S.B., Gligor, V. and Perrig, A.: STRIDE: Sanctuary Trail – Refuge from Internet DDoS Entrapment, *Proc. 8th ACM SIGSAC Symposium on Information*, Computer and Communications Security (2013).

[17] 西塚　要：キャリアにおける DoS/DDoS 対策の取り組み (2013)，入手先 ⟨https://www.nic.ad.jp/ja/materials/iw/2013/proceedings/s2/s2-nishizuka.pdf⟩.

[18] Cisco: Remotely Triggered Black Hole Filtering—Destination Based and Source Based (2005), available from ⟨http://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf⟩.

[19] F5 Networks：F5 DDoS 防御リファレンスアーキテクチャ (2014)，入手先 ⟨https://f5.com/Portals/1/PDF/JAPAN/SOLUTIONS/RA%20DDoS%20Protection%20Technical%20White%20Paper-jp.pdf⟩.

[20] Arbor Networks：Arbor Networks の DDoS 攻撃防御ソリューション (2016)，入手先 ⟨http://jp.arbornetworks.com/wp-content/uploads/2016/08/sb_ddosattackprotection_jp-110816-FINAL.pdf⟩.
[21] Akamai：DDoS 防御に Akamai のクラウドセキュリティを使用する理由 (2017)，入手先 ⟨https://www.akamai.com/jp/ja/solutions/products/cloud-security/ddos-protection-service.jsp⟩.
[22] No, G. and Ra, I.: An Efficient and Reliable DDoS Attack Detection Using a Fast Entropy Computation Method, *Proc. 9th International Symposium on Communications and Information Technology*, pp.1223–1228 (2009).
[23] Corero Network Security: DDoS Trends and Analysis Report (2015), available from ⟨http://info.corero.com/2015-Mid-Year-DDoS-Report.html⟩.
[24] Arbor Networks: DDoS Attacks from IoT Botnets Don't Have to Mean Game Over (2016), available from ⟨https://www.arbornetworks.com/blog/asert/rio-olympics-take-gold-540gbsec-sustained-ddos-attacks/⟩.
[25] Verisign: Verisign Distributed Denial of Service Trends Report (2016), available from ⟨https://www.verisign.com/assets/report-ddos-trends-Q12016.pdf⟩.
[26] Li, M., Li, J. and Zhao, W.: Simulation Study of Flood Attacking of DDOS, *Internet Computing in Science and Engineering*, pp.286–293 (2008).
[27] Statista: Hours of Video Uploaded to YouTube Every Minute as of July 2015 (2015), available from ⟨https://www.statista.com/statistics/259477/hours-of-video-uploaded-to-youtube-every-minute/⟩.
[28] Ester, M.-A. and Benzmüller, R.: Underground Economy, G Data Whitepaper (2009), 岸本眞輔，瀧本往人（訳）(2009)，入手先 ⟨https://sv20.wadax.ne.jp/~gdata-co-jp/securitylabs/whitepaper/images/whitepapers/WP_UndergroundEconomy.pdf⟩.

岩永 崇裕 （学生会員）

2014 年佐世保工業高等専門学校電子制御工学科卒業．2016 年筑波大学情報学群情報メディア創成学類卒業．現在，東京大学大学院学際情報学府学際情報学専攻修士課程在籍中．サイバー攻撃に対する防御方式，サイバー攻撃発生のプロセスおよびメカニズムに関する研究に興味を持つ．

木村 成伴 （正会員）

1967 年生．1990 年東北大学工学部情報科学科卒業．1992 年同大学大学院博士前期課程修了．1995 年同大学院博士後期課程修了．博士（情報科学）．同年筑波大学講師．2001 年同大学助教授．2007 年同大学准教授．プロセス代数，ネットワークプロトコル，通信システムの効率評価等に関する研究に従事．電子情報通信学会，ソフトウェア科学会，IEEE，IEEE-CS，IEEE-ComSoc，ACM 各会員．