Computer Security Symposium 2020
26 - 29 October 2020

DDoS攻撃の自動水際防御を目的とした疎なISP間連携フレームワークの提案

小西 崇之[1,a)]　西村 俊和[1]　瀧本 栄二[1]

概要：DDoS 攻撃の効果的な抑制には，攻撃を攻撃元に近い ISP で自動的に防御する自動水際防御が重要である．しかし，トラフィック制御ポリシが異なる ISP 同士が連携して攻撃を防御するためには解決すべき課題がある．そこで本論文では，このような課題を解決し自動水際防御を実現するためのフレームワークを提案する．本フレームワークは，DDoS 攻撃対策の自動化のためのシグナリングプロトコルである DOTS を使用しているが，各 ISP が各々のポリシに基づいて連携できるように防御要請パラメータを改変している．また，本フレームワークのプロトタイプを実装した仮想ネットワークで実験を行い，本フレームワークによって DDoS 攻撃を自動水際防御可能であることを示す．

キーワード：セキュリティ，DDoS 攻撃，セキュリティオートメーション，DOTS，ISP 間連携

Loose inter-ISP collaboration framework for DDoS attack defense automation close to attackers

Takayuki Konishi[1,a)]　Toshikazu Nishimura[1]　Eiji Takimoto[1]

Abstract: The automatic defense at ISPs that are close to attackers is important to suppress DDoS attacks effectively. Although the collaboration among ISPs is required in order to do that, the difference in traffic control policies makes the collaboration harder. In this paper, we propose a new framework that enables to suppress DDoS attacks at ISPs close to attackers automatically avoiding the influence of the difference in traffic control policies. The proposed framework presupposes some servers and they communicate each other via DOTS. DOTS is a signaling protocol aiming to protect a victim from DDoS attacks. The data model exchanged via DOTS is designed as a minimum data set considering the difference in policies. We also experimented with the framework in the virtual environment. From the results, we certificated the functionality that suppresses DDoS attacks at ISPs close to attackers automatically.

Keywords: security, DDoS attack, security automation, DOTS, inter-ISP collaboration

1. はじめに

インターネットが重要なインフラとなった現代において，標的サービスを提供するサーバ（以下，標的）を攻撃して当該サービスを停止させることを目的とした攻撃である Denial of Service (DoS) 攻撃を，高度に攻撃元を分散させた Distributed DoS (DDoS) 攻撃は非常に深刻な問題となっている．NETSCOUT Blog [1] によると，2018 年 3 月に観測史上最大ビットレートである 1.7Tbit/s の DDoS 攻撃が観測されており，攻撃の大規模化の傾向がみられることが報告されている．

従来の DDoS 攻撃対策を**図 1** に示す．従来の DDoS 攻撃対策は，標的が検知して標的自身が防御する手法と，標的が検知して上流 Internet Service Provider (ISP) に防御を要請し上流 ISP で防御する手法を組み合わせた対策が主流である．しかし，攻撃元から上流 ISP までに存在するネットワークには攻撃パケットが流入しているため，通

[1] 立命館大学
Ritsumeikan University
[a)] is0306he@ed.ritsumei.ac.jp

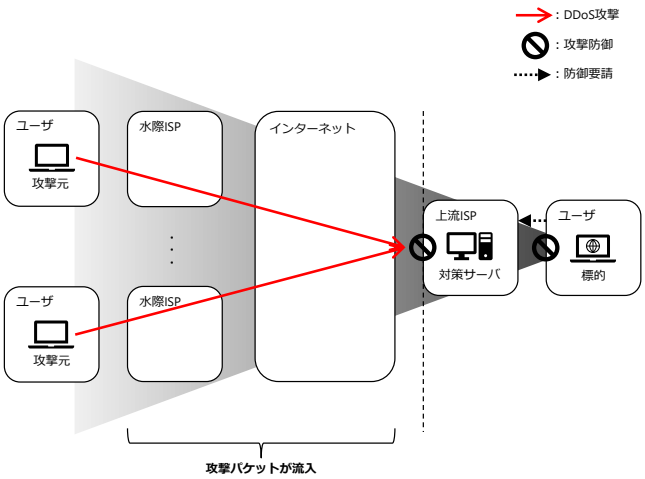

図 1　従来の DDoS 攻撃対策
Fig. 1　Conventional DDoS attack Protection.

信量増加などの影響は防げない．そのため，攻撃元に近い ISP における防御が必要である．また，攻撃検知から攻撃元に近い ISP における防御までの自動化による高速化も併せて求められる．文献 [2] でも，可能な限り攻撃元に近い場所で即時に防御する機能の必要性が報告されている．

しかし，DDoS 攻撃を攻撃元に近い ISP で自動的に防御すること（以下，自動水際防御）を目的とした既存研究 [3], [4] は，ISP 間連携だけでなく ISP 内のトラフィック制御ポリシなども厳密に定義している．そのため，ISP 間でトラフィック制御ポリシを統一しなければならない．しかし，ISP ごとにトラフィック制御ポリシはそれぞれ異なり，ISP 間で連携して自動水際防御をすることに協力的な ISP もいれば，ポリシに抵触してしまうため防御を実施しない非協力的な ISP，攻撃パケットの完全遮断はできないが帯域制限であれば応じる ISP など，様々存在すると考えられる．そもそも，競合他社である ISP 同士が積極的に協力するとは考えにくい．よって，ポリシ統一は容易ではなく，これらの既存研究 [3], [4] は実現可能性が非常に低くなっている．また，対策の自動化のためのシグナリングプロトコルである DDoS Open Threat Signaling (DOTS) [5] は ISP 間連携の詳細なユースケースを提示していない．

そこで本論文では，DDoS 攻撃の自動水際防御を目的とした疎な ISP 間連携フレームワーク（以下，FW）を提案する．これは，FW 管理組織に設置されたグローバルマネージャ（以下，GM）と，各 ISP に設置されたローカルマネージャ（以下，LM）で構成される．FW は，ISP ごとに既に行われている DDoS 攻撃対策を可能な限り流用し，ISP ごとのトラフィック制御ポリシを尊重し防御実施の有無を要請先の ISP の判断にゆだねる．また，GM と LM 間の通信は DOTS を各 ISP が各々のポリシに基づいて連携できるように防御要請パラメータを改変した ISP 間 DOTS を使用する．最後に，インターネットを模した簡易的な仮想ネットワークを構築し，そこに FW のプロトタイプを実装した環境で，DDoS 攻撃を発生させプロトタイプを使用し防御する 4 つの実験を行う．そして，実験結果から FW によって DDoS 攻撃を自動水際防御可能であることを示す．

2. 関連研究

本章では，自動水際防御を目的とした既存研究について説明する．

2.1 Antidose

Antidose [3] は，標的ノードが所属するネットワークに設置されたターゲットエージェントが，正規クライアントのホワイトリストに基づいたフィルタを定義し，それを隣接する Autonomous System (AS) に伝搬することで水際防御を実現している．フィルタは，攻撃を受けている最中にも新たな正規クライアントが標的ノードに接続できるような設計となっている．しかし，実装が複雑であることと Antidose の通信によるオーバヘッドが課題となっている．

2.2 Cochain-SC

Cochain-SC [4] は，ドメイン内 DDoS 攻撃対策とドメイン間 DDoS 攻撃対策を組み合わせた DDoS 攻撃対策スキームである．攻撃情報を単一ノードが集中管理するのではなくブロックチェーンを活用して分散管理することでスキーム自体の DDoS 攻撃に対する脆弱性を回避している．しかし，Cochain-SC に参加するドメインは Software Defined Networking (SDN) による DDoS 攻撃対策が可能でなければならない．

3. DOTS

DOTS は，DOTS サーバと DOTS クライアントと呼ばれる 2 つのコンポーネント間の通信を定義している．DOTS が想定している一般的なユースケースは，標的ネットワークに設置されている DOTS クライアントが，ISP などの上流ネットワークに設置されている DOTS サーバに対して防御を要請するケースである．DOTS において DOTS クライアントと DOTS サーバは，シグナルチャネル [6] とデータチャネル [7] という 2 種類のチャネルを確立する．シグナルチャネルは，防御を要請するためのチャネルである．Constrained Application Protocol (CoAP) [8] over Datagram Transport Layer Security (DTLS) [9] または CoAP over Transport Layer Security (TLS) [10] が用いられている．データフォーマットは，JavaScript Object Notation (JSON) [11] を Concise Binary Object Representation (CBOR) [12] でシリアライズしたものである．DOTS の防御要請の例を**図 2** に示す．また，防御要請パラメータを以下に示す．

cdid　クライアントドメイン識別子を記述する．DOTS サーバ側ネットワークに DOTS ゲートウェイと呼ば

れるコンポーネントがあるシナリオにおいて DOTS サーバのみが記述できる．DOTS ゲートウェイが存在するシステム構成の場合は必須項目である．

cuid DOTS クライアント識別子を記述する．必須項目である．

mid 防御要請識別子を記述する．必須項目である．

target-prefix Classless Inter-Domain Routing (CIDR) 表記で標的 IP アドレスを記述する．複数記述できる．target-fqdn と target-uri と alias-name がない場合は必須項目である．

target-port-range 標的ポート番号の範囲を記述する．lower-port に下限ポート番号を，upper-port に上限ポート番号を記述する．単一ポートの場合は lower-port のみ記述する．複数記述できる．推奨項目である．

target-protocol 標的プロトコル番号を記述する．指定しない場合は全てを意味する．複数記述できる．推奨項目である．

target-fqdn 標的ドメイン名を記述する．複数記述できる．target-prefix と target-uri と alias-name がない場合は必須項目である．

target-uri 標的 Uniform Resource Identifier (URI) を記述する．複数記述できる．target-prefix と target-fqdn と alias-name がない場合は必須項目である．

alias-name 防御要請パラメータの組み合わせに名前を付けたものを記述する．あらかじめデータチャネルで登録しておく．複数記述できる．target-prefix と target-fqdn と target-uri がない場合は必須項目である．

lifetime 防御の継続時間を記述する．推奨値は 3600 秒で-1 は無期限を意味する．必須項目である．

trigger-mitigation 即座に防御を開始するかの有無を true もしくは false で記述する．デフォルトは true で，false の場合はハートビートと呼ばれる定期的な通信を DOTS クライアントから受信しなくなったタイミングで防御を開始する．これをデッドマントリガという．推奨項目である．

データチャネルは，シグナルチャネルを用いて防御を要請する前に，共有する設定情報などを交換するためのチャネルである．REpresentational State Transfer CONFiguration protocol (RESTCONF) [13], [14] が用いられている．データフォーマットは，JSON である．

4. FW

FW が想定している環境は，各 ISP ネットワークの配下に複数のユーザネットワークが存在し，ISP 間はインターネットを介して通信を行う環境である．また，FW 管理組織ネットワークも ISP 同様にインターネットに接続されている．FW の適用イメージを**図 3** に示す．

FW を構成するコンポーネントは，各ユーザネットワークに設置された，標的となり得るサーバに内包されている DOTS クライアント，各 ISP ネットワークに設置された LM，FW 管理組織ネットワークに設置された GM である．DOTS クライアントと LM 間の通信には，第 3 章で述べた DOTS の使用を想定している．ただし，DOTS の使用は必須ではなく，例えば ISP が提供する DDoS 攻撃対策サービスで使用されているプロトコルを使用してもよい．DOTS を採用する理由は，DDoS 攻撃を検知した際に即効的な対策を行うために使われるシグナルチャネルと，平常時の通信に用いるデータチャネルが備わっていることによる．したがって，FW ではこれらに相当する通信チャネルがあれ

```
Header: PUT (Code=0.03)
Uri-Path: ".well-known"
Uri-Path: "dots"
Uri-Path: "mitigate"
Uri-Path: "cdid=7eeaf349529eb55ed50113"
Uri-Path: "cuid=dz6pHjaADkaFTbjr0JGBpw"
Uri-Path: "mid=123"
Content-Format: "application/dots+cbor"

{
  "ietf-dots-signal-channel:mitigation-scope": {
    "scope": [
      {
        "target-prefix": [
          "2001:db8:6401::1/128",
          "2001:db8:6401::2/128"
        ],
        "target-port-range": [
          {
            "lower-port": 80
          },
          {
            "lower-port": 443
          },
          {
            "lower-port": 8080
          }
        ],
        "target-protocol": [
          6
        ],
        "lifetime": 3600
      }
    ]
  }
}
```

図 2 DOTS の防御要請の例
Fig. 2 Examples of DOTS Mitigation Request.

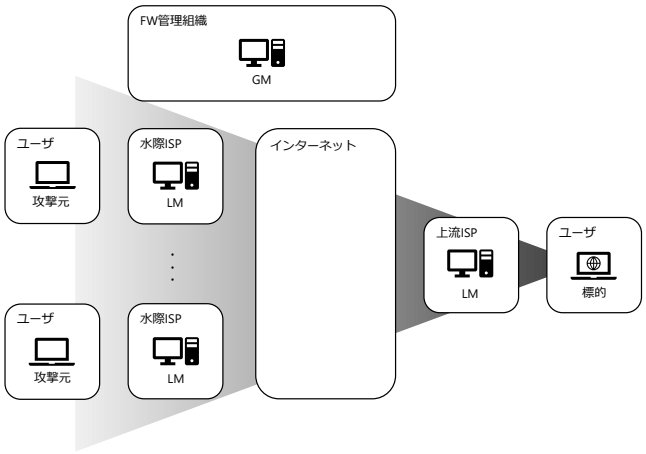

図 3 FW の適用イメージ
Fig. 3 Example of FW Application.

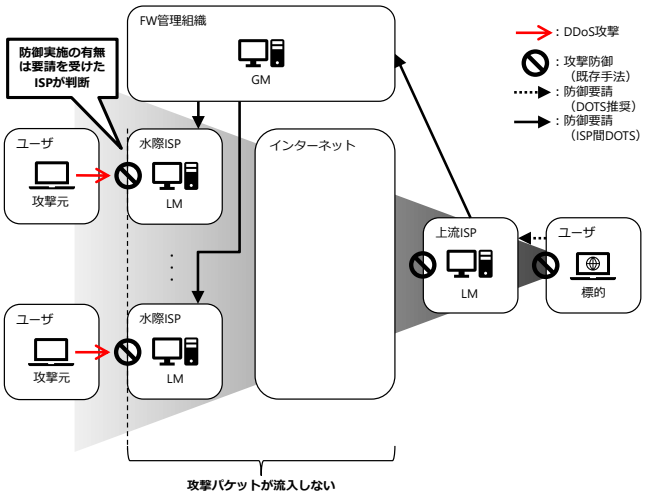

図 4　攻撃防御機能の動作フロー

Fig. 4　Operation Flow of DDoS attack Defense Function.

ば DOTS である必要はない．LM と GM 間の通信は共通プロトコルとして，DOTS を改変した ISP 間 DOTS を使用する．ISP 間 DOTS は，FW の機能に合わせて DOTS の通信内容を変更したプロトコルである．通信方式やプロトコルスタックの変更といった大きな変更を加えていないため，本論文では詳細な説明は省略する．

FW が実現する機能は 2 種類ある．1 つは，DDoS 攻撃を検知した際に即座に対応するための機能である．これは，ISP 間 DOTS シグナルチャネルを介して行われる．もう 1 つは，ネットワーク上で行われている DDoS 攻撃およびその予兆と考えられる通信，例えば，DDoS 攻撃で発生するバックスキャッタなどの情報の共有である．これは ISP 間 DOTS データチャネルを介して行われる．本論文では，前者の攻撃防御機能について述べる．

攻撃防御機能の動作フローを**図 4** に示す．攻撃防御機能は，DDoS 攻撃を検知した標的ユーザが既存の通信方式もしくは DOTS クライアントのシグナルチャネルで上流 ISP の LM に防御を要請することで起動する．防御を要請された LM は要請に応じた防御を防御機器に指示し，当該要請を ISP 間 DOTS シグナルチャネル の防御要請に書き換え，ISP 間 DOTS シグナルチャネルで GM に送信する．防御を要請された GM は他の LM に当該要請を ISP 間 DOTS シグナルチャネルで転送し，要請された他 ISP の LM は ISP ごとのトラフィック制御ポリシに従い防御機器に防御を指示する．ISP 内の防御は，新たな防御機器設置のコストを削減するため，既存のものを流用する．また，全ての LM が防御を指示する必要はない．これは，ISP ごとにトラフィック制御ポリシが異なり統一することが困難なためである．これにより FW は，トラフィック制御ポリシを統一しなければならない既存研究 [3], [4] と比較し，疎な ISP 間連携を実現することで実現可能性を高めている．

GM の構成を**図 5** に示す．GM は ISP 間 DOTS クライアントと ISP 間 DOTS サーバと防御要請配信部で構成される．防御要請配信部は ISP 間 DOTS サーバのシグナルチャネルから受け取った JSON 形式の防御要請を，GM が保持している LM の IP アドレスに対して ISP 間 DOTS クライアントのシグナルチャネルで送信する．

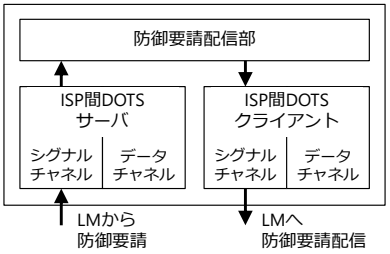

図 5　GM の構成

Fig. 5　GM Structure.

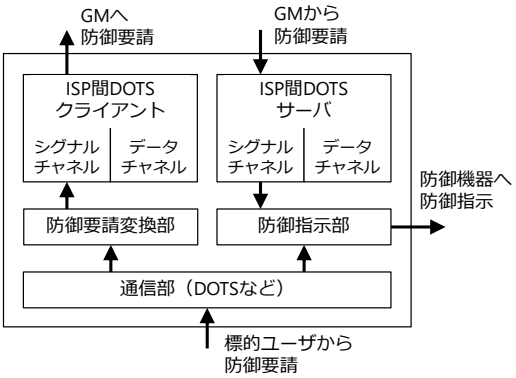

図 6　LM の構成

Fig. 6　LM Structure.

LM の構成を**図 6** に示す．LM は ISP 間 DOTS クライアント，ISP 間 DOTS サーバ，防御要請変換部，防御指示部，通信部で構成される．防御要請変換部は通信部から受け取ったユーザからの防御要請を ISP 間 DOTS の防御要請に変換する．防御指示部はユーザもしくは GM からの防御要請に基づき ISP 内の防御機器に防御を指示する．通信部はユーザとの通信を行う．通信プロトコルは DOTS を推奨しているが必須ではない．

ISP 間 DOTS シグナルチャネルは，DOTS シグナルチャネルの防御要請パラメータを各 ISP が各々のポリシに基づいて連携できるように改変したプロトコルである．パラメータの選択基準は，DDoS 攻撃を防御するために最低限必要なパラメータであることである．その結果，偽装された防御要請でないことを証明するための一意な識別子である cuid，mid，標的 IP アドレスが示されている target-prefix，防御の継続時間が示されている lifetime が最低限必要であると判断した．target-port-range と target-protocol は，記述されていなくても，DDoS 攻撃を防御可能であるため，推奨項目となっている．記述すると，正規パケットも防御してしまう可能性が低くなる．ISP 間 DOTS シグナルチャネルの防御要請パラメータを以下に示す．

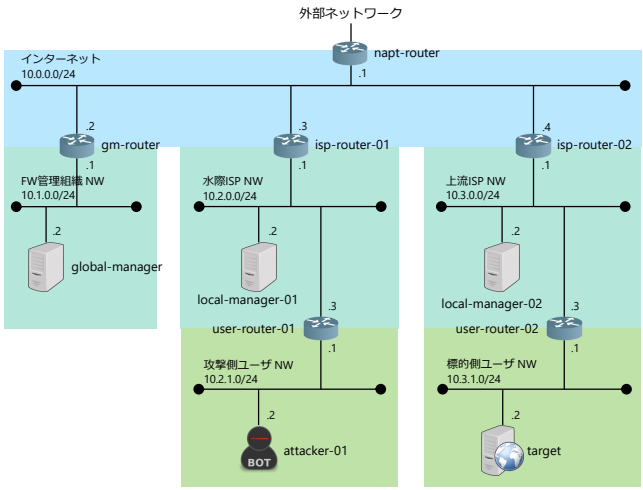

図 7　実験環境

Fig. 7　Experimental Environment.

```
{
  "ietf-dots-signal-channel:mitigation-scope": {
    "scope": [
      {
        "target-prefix": [
          "10.3.1.2/24"
        ],
        "target-port-range": [
          {
            "lower-port": 53
          }
        ],
        "target-protocol": [
          17
        ],
        "lifetime": 3600
      }
    ]
  }
}
```

図 8　防御要請の JSON ファイル

Fig. 8　JSON File of Mitigation Request.

cuid　LM 識別子を記述する．必須項目である．

mid　防御要請識別子を記述する．必須項目である．

target-prefix　CIDR 表記で標的 IP アドレスを記述する．複数記述できる．必須項目である．

target-port-range　標的ポート番号の範囲を記述する．lower-port に下限ポート番号を，upper-port に上限ポート番号を記述する．単一ポートの場合は lower-port のみ記述する．複数記述できる．推奨項目である．

target-protocol　標的プロトコル番号を記述する．指定しない場合は全てを意味する．複数記述できる．推奨項目である．

lifetime　防御の継続時間を記述する．推奨値は 3600 秒で-1 は無期限を意味する．必須項目である．

5. 評価

本章では，インターネットを模した簡易的な仮想ネットワークを構築し，そこに FW のプロトタイプを実装した環境で，DDoS 攻撃を発生させプロトタイプを使用し防御する 4 つの実験を行った．そして，実験結果から FW によって DDoS 攻撃を自動水際防御可能であることを示す．仮想ネットワークは．仮想化ソフトウェアのネットワーク機能と複数ゲスト Operating System (OS) 起動機能を用いて実現した．ここでは，仮想化ソフトウェアとして Oracle VM VirtualBox [15] を用いている．実験環境を**図 7** に示す．

実験環境は，FW 管理組織ネットワークと 2 つの ISP ネットワークがインターネットを介して接続され，各 ISP がそれぞれ 1 つのユーザネットワークを配下に持つ環境を再現した．

実験 1 では，従来手法である上流 ISP における防御を再現するため，isp-router-02 のみで攻撃パケット廃棄による防御を行った．実験 2 では，プロトタイプを使用して isp-router-02 と user-router-01 で攻撃パケット廃棄による防御を行った．実験 3 では，攻撃パケットの完全遮断はできないが帯域制限であれば応じる ISP を再現するため，プロトタイプを使用して isp-router-02 で攻撃パケット廃棄，user-router-01 で攻撃パケットシェーピングによる防御を行った．攻撃パケットシェーピングは上限レートを 1 Mbit/s，下限レートを 1 kbit/s に設定した．実験 4 では，非協力的な ISP を再現するため，プロトタイプを使用して isp-router-02 で攻撃パケット廃棄したが user-router-01 では何もしなかった．

次に，実験シナリオを説明する．あらかじめ，global-manager で GM，local-manager-01 と local-manager-02 で LM を実行する．そして，attacker-01 が hping3 を用いて UDP フラッド攻撃を開始する．攻撃パケットはパケットサイズが 1500byte のパケットを標的の 53 番ポートに 100,000 パケット送信する設定となっている．攻撃開始から 20 秒後に標的が攻撃を検知したと仮定し，target が上流 ISP に防御を要請する．その後，全ての攻撃パケットが送信されるのを待ち実験を終了する．

防御要請の JSON ファイルを**図 8** に示す．防御要請は，IP アドレスが 10.3.1.2 のノードの 53 番ポートへの UDP パケットを 3600 秒間防御してほしいことを意味している．

5.1 実験結果

isp-router-01 と isp-router-02 と user-router-01 と user-router-02 それぞれの target 側のインタフェースと target のインタフェースにおいてパケットをキャプチャした．そして，isp-router-01 と isp-router-02 と user-router-01 と user-router-02 は転送した攻撃パケットのみ，target は受信した攻撃パケットのみを，縦軸がビットレートで横軸が時間の折れ線グラフとして描画した．

実験 1，実験 2，実験 3，実験 4 の結果をそれぞれ**図 9**，**図 10**，**図 11**，**図 12** に示す．

実験 1 では，target が防御を要請してから数秒後に，

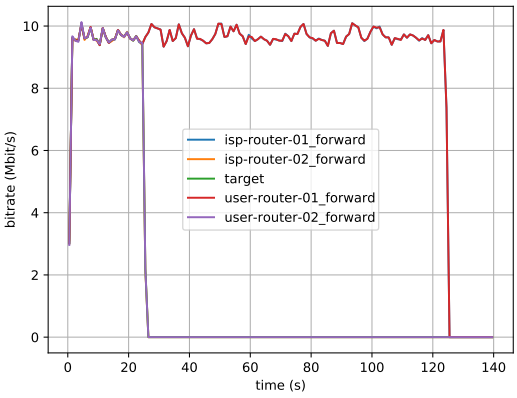

図 9　実験 1 の結果
Fig. 9　Result of Experiment 1.

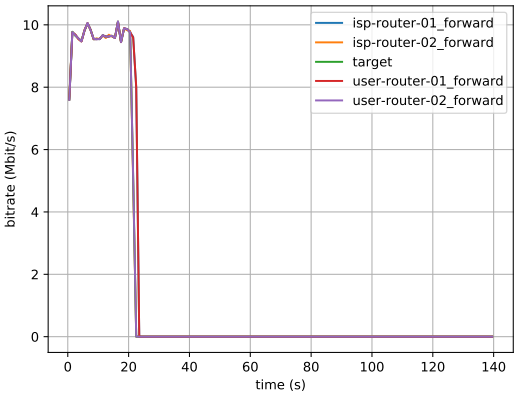

図 10　実験 2 の結果
Fig. 10　Result of Experiment 2.

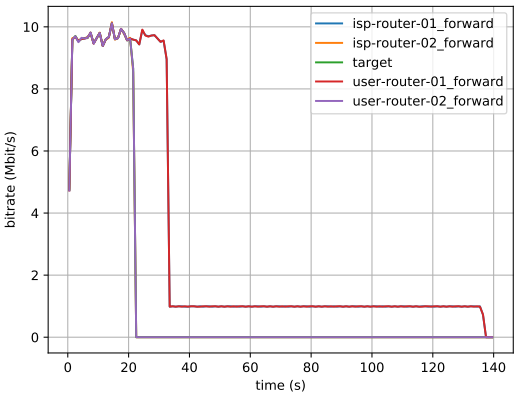

図 11　実験 3 の結果
Fig. 11　Result of Experiment 3.

target と user-router-02 と isp-router-02 のビットレートが 0 になっている．しかし，isp-router-01 と user-router-01 のビットレートは変化していない．よって，従来手法では，上流 ISP ネットワークには攻撃パケットが流入していないが，水際 ISP ネットワークには攻撃パケットが流入しているため，自動水際防御が実施されていないといえる．

実験 2 では，target が防御を要請してから数秒後に，target と user-router-02 と isp-router-02 のビットレートが 0 になっている．さらに数秒後に，isp-router-01 と user-router-01 のビットレートが 0 になっている．よって，本プロトタイプを使用して上流 ISP と水際 ISP において攻撃パケット廃棄をした場合は上流 ISP ネットワークと水際 ISP ネットワークの両方ともに攻撃パケットが流入していないため，自動水際防御が実施されたといえる．

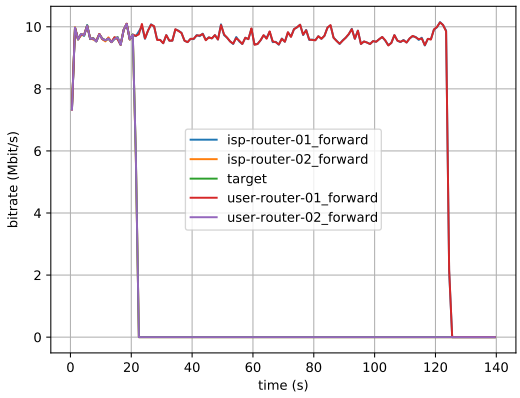

図 12　実験 4 の結果
Fig. 12　Result of Experiment 4.

実験 3 では，target が防御を要請してから数秒後に，target と user-router-02 と isp-router-02 のビットレートが 0 になっている．さらに十数秒後に，isp-router-01 と user-router-01 のビットレートが 1 Mbit/s になっている．よって，本プロトタイプを使用して上流 ISP において攻撃パケット廃棄，水際 ISP において攻撃パケットシェーピングをした場合は上流 ISP ネットワークには攻撃パケットが流入せず，水際 ISP ネットワークには攻撃パケットが流入しているが上限レートを 1 Mbit/s に制限されているため，自動水際防御が実施されたといえる．

実験 4 では，target が防御を要請してから数秒後に，target と user-router-02 と isp-router-02 のビットレートが 0 になっている．しかし，isp-router-01 と user-router-01 のビットレートは変化していない．よって，本プロトタイプを使用して上流 ISP において攻撃パケット廃棄，水際 ISP において何もしなかった場合は上流 ISP ネットワークには攻撃パケットが流入していないが，水際 ISP ネットワークには攻撃パケットが流入しているため，自動水際防御が実施されていないといえる．

5.2　考察

実験結果から，FW によって DDoS 攻撃を自動水際防御可能であることが確認できた．また，防御を要請されたにもかかわらず防御しない，非協力的な ISP のみしか存在しなかったとしても，少なくとも上流 ISP では防御が行われるため，従来手法と同程度の効果を確認できた．

実験開始から 20 秒経過時点で target が防御を要請してから，上流 ISP が防御し target，user-router-02，isp-router-02 のビットレートが 0 になるまでに，実験 1 で 7 秒，実験 2，実験 3，実験 4 で 3 秒のオーバヘッドが発生してい

る．これは，target が防御を要請するまでの処理，防御要請が target から local-manager-02 に届くまでの通信遅延，local-manager-02 の LM が防御を指示するまでの処理によるものである．したがって，オーバヘッドはプロトタイプの実装方法やネットワーク構成に依存する．また，上流 ISP が防御してから水際 ISP が防御するまでに実験 2 では 1 秒，実験 3 では 11 秒のオーバヘッドが発生している．これは，実験 2 においては，local-manager-02 の LM が防御してから local-manager-01 の LM が防御を指示するまでの各ノードの処理や通信遅延によるものである．そのため，これも同様にオーバヘッドはプロトタイプの実装方法やネットワーク構成に依存する．実験 3 においては，実験 2 と同様の処理や通信遅延によるものに加えて，シェービングを実行してから実際にシェービングされたトラフィックが送信されるまでの遅延によるものと考えられる．そのため，オーバヘッドはプロトタイプの実装方法やネットワーク構成に加え，user-router-01 の target 側インタフェースの送信ソケットバッファサイズや user-router-01 と isp-router-01 間の帯域幅に依存すると考えられる．

上流 ISP ネットワークと標的ユーザネットワーク間の回線が輻輳するような大規模な DDoS 攻撃の場合は，回線輻輳によるパケットロス率増加により target から local-manager-02 への防御要請が届かない可能性がある．FW は，LM とユーザ間の通信に DOTS を推奨しており，DOTS は防御要請のために別リンクやモバイル回線を用意することを検討しているため，FW もこれに従う．

一般的な UDP フラッド攻撃は，攻撃パケットの送信元 IP アドレスの偽装，いわゆる IP スプーフィングがされている場合が多いが，本論文の実験では攻撃パケットの IP スプーフィングをしていない．これは，送信元 IP アドレスベースの防御は攻撃パケットの IP スプーフィングによってほぼ無意味であり，DOTS は宛先 IP アドレスベースの防御を想定しているためである，よって，攻撃パケットの IP スプーフィングの有無によって実験結果に変化はないと考えられる．

GM 自体が攻撃の標的となってしまうことが考えられるが，本論文では最も簡単な構成を提案しており，Cochain-SC [4] のような分散配置など現実的な実装方法を今後検討していく余地があると考えている．

6. おわりに

本論文では，DDoS 攻撃の自動水際防御を目的とした疎な ISP 間連携フレームワークを提案した．FW は，DDoS 攻撃の標的となったユーザが上流 ISP の LM へ防御を要請すると起動し，当該攻撃の自動水際防御を実現する．また，本フレームワークのプロトタイプを実装した仮想ネットワークで実験を行い，本フレームワークによって DDoS 攻撃を自動水際防御可能であることを示した．今後の課題は，協力的な ISP や非協力的な ISP が多数混在する大規模な実験環境で同様の実験を行い FW の有用性を示すこと，GM の現実的な実装方法を検討すること，DDoS 攻撃関連情報を ISP 間で共有する機能の詳細な設計と評価である．

参考文献

[1] Bienkowski, T.: 1.7tbps DDoS Attack Makes History — NETSCOUT, NetScout Systems, Inc. (online), available from ⟨https://www.netscout.com/blog/security-17tbps-ddos-attack-makes-history⟩ (accessed 2020-08-14).

[2] Zargar, S. T., Joshi, J. and Tipper, D.: A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks, *IEEE Communications Surveys Tutorials*, Vol. 15, No. 4, pp. 2046–2069 (2013).

[3] Simpson, S., Shirazi, S. N., Marnerides, A., Jouet, S., Pezaros, D. and Hutchison, D.: An Inter-Domain Collaboration Scheme to Remedy DDoS Attacks in Computer Networks, *IEEE Transactions on Network and Service Management*, Vol. 15, No. 3, pp. 879–893 (online), DOI: 10.1109/TNSM.2018.2828938 (2018).

[4] Abou El Houda, Z., Hafid, A. S. and Khoukhi, L.: Cochain-SC: An Intra- and Inter-Domain Ddos Mitigation Scheme Based on Blockchain Using SDN and Smart Contract, *IEEE Access*, Vol. 7, pp. 98893–98907 (2019).

[5] Mortensen, A., Reddy.K, T. and Moskowitz, R.: DDoS Open Threat Signaling (DOTS) Requirements, IETF (online), DOI: 10.17487/RFC8612 (accessed 2020-08-14).

[6] Reddy.K, T., Boucadair, M., Patil, P., Mortensen, A. and Teague, N.: Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification, IETF (online), DOI: 10.17487/RFC8782 (accessed 2020-08-14).

[7] Boucadair, M. and Reddy.K, T.: Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification, IETF (online), DOI: 10.17487/RFC8783 (accessed 2020-08-14).

[8] Shelby, Z., Hartke, K. and Bormann, C.: The Constrained Application Protocol (CoAP), IETF (online), DOI: 10.17487/RFC7252 (accessed 2020-08-14).

[9] Rescorla, E. and Modadugu, N.: Datagram Transport Layer Security Version 1.2, IETF (online), DOI: 10.17487/RFC6347 (accessed 2020-08-14).

[10] Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3, IETF (online), DOI: 10.17487/RFC8446 (accessed 2020-08-14).

[11] Bray, T.: The JavaScript Object Notation (JSON) Data Interchange Format, IETF (online), DOI: 10.17487/RFC8259 (accessed 2020-08-14).

[12] Bormann, C. and Hoffman, P. E.: Concise Binary Object Representation (CBOR), IETF (online), DOI: 10.17487/RFC7049 (accessed 2020-08-14).

[13] Bierman, A., Björklund, M. and Watsen, K.: RESTCONF Protocol, IETF (online), DOI: 10.17487/RFC8040 (accessed 2020-08-14).

[14] Björklund, M., Schönwälder, J., Shafer, P. A., Watsen, K. and Wilton, R.: RESTCONF Extensions to Support the Network Management Datastore Architecture, IETF (online), DOI: 10.17487/RFC8527 (accessed 2020-08-14).

[15] Oracle Corporation: Oracle VM VirtualBox, Oracle Corporation (online), available from ⟨https://www.virtualbox.org/⟩ (accessed 2020-08-14).