

# ネットワークスイッチのMACアドレス認証と パケットダンプ機能を活用したループ検知と対応の支援

大森 幹之<sup>1,a)</sup>

**概要:** ネットワークでループが形成されると輻輳が発生し、通信障害を招くことがある。そのため、ループを迅速に検知し、解消することが重要である。そこで、本稿では、ループ検知パケット及びスイッチのCPUの高い使用率(CPU高騰)、MACアドレス認証ログの急増、エッジスイッチでのコアスイッチのMACアドレスの観測によるループ検知手法を提案する。また、ループ発生源の手掛かりとして、VLAN IDやパケットダンプなども提示する。そして、提案手法を実ネットワークで評価し、ループ検知パケットとCPU高騰によるループ検知手法が感度が高いことがわかった。一方、コアスイッチのMACアドレスの観測により、ループの発生源となっているエッジポートを迅速かつ断定的に特定できた。

**キーワード:** ループ検知, MACアドレス認証, パケットダンプ, ネットワーク障害対応

## A Loop Detection and Trouble-Shooting Support Utilizing MAC address authentication and Packet Dump Function of a Network Switch

MOTOYUKI OHMORI<sup>1,a)</sup>

**Abstract:** A loop in a network can cause congestion and communication failures. It is, therefore, important to detect and eliminate loops as soon as possible. In this paper, we propose a loop detection method that utilizes loop detection packets, CPU spikes, a rapid increase in MAC address authentication logs, and observations of a core switch MAC address at an edge switch. In addition, the proposed method provides a clue of the cause of the loop, such as a VLAN ID, a dump of the packets, and the source MAC address. We evaluated the proposed method on a real network, and found that the loop detection methods using loop detection packets and CPU spikes were highly sensitive. On the other hand, by observing the MAC address of the core switch, we could quickly and decisively identify the edge port that was the source of the loop.

**Keywords:** loop detection, MAC address authentication, packet dump, network trouble-shooting support

### 1. はじめに

ネットワークにおけるループは迅速に発生箇所と原因を特定し解消することが重要である。ループ検知手法としては、STP (Spanning Tree Protocol) [1] やベンダ独自のループ検知パケットを用いるもの、閾値を超えた大量のパ

ケット受信 (ストーム) を検知するものなどがある。これらの手法では、ループの発生を管理者に通知できなかったり、ループの原因となっている機器やケーブルを収容しているスイッチやポートを正確には示せないことがある。そのため、ループ発生自体を認知できなかったり、ループ検知後に修正すべき端末やケーブルの配線などが不明である場合がある。

そこで、本稿では、ループの原因となっているスイッチとポートをより正確に自動的に導出することを提案する。

<sup>1</sup> 鳥取大学 情報基盤機構  
Organization for Information and Communication Technology, Tottori University

<sup>a)</sup> ohmori@tottori-u.ac.jp

ループ検知に関しては、上述のループ検知パケットに加え、スイッチの CPU の高い使用率 (CPU 高騰) の検知と MAC アドレス認証も活用する。ループに起因したブロードキャストパケットなどのストームによる CPU 高騰を検知することにより、高確率でループを検知できる。また、MAC アドレス認証により、ループ発生時にコアスイッチの MAC アドレスが下流側、つまり、エッジ側のポートで検知されることを利用し、ループの発生源であるポートをより正確に検知できる。ループ検知後は、原因となるポートや端末を同定するため、ループ検知されたスイッチとその隣接スイッチでパケットダンプを自動的に取得する。そして、パケットダンプを解析し、MAC アドレスや IP アドレスなどを提示することで、管理者や利用者がループ発生原因をより容易に特定できる様に支援する。

本稿の構成は以下のとおりである。2 節では、上述したループ検知における課題をより詳細に述べる。3 節では、ネットワーク機器のパケットダンプ機能を活用したループ検知と解決支援を提案する。4 節では、提案手法の実装について述べる。5 節では、提案手法を評価する。6 節では、提案手法などについて考察する。7 節では、関連研究に言及する。最後に、8 節で本論文をまとめる。

## 2. ループの検知と解消における課題

### 2.1 ループ検知パケットを透過しないスイッチ

スイッチにとって未知なイーサタイプである IPv6 や EAP, STP, ベンダ独自のループ検知パケットを透過的に転送しないスイッチも存在する。その様なスイッチを利用者が独自に組織内ネットワークに接続すると、STP やベンダ独自パケットを用いたループ検知・対策では、当該スイッチ以降でのループ発生を組織内ネットワークからは検知できない。そのため、ループ検知パケットを利用する手法だけではなく、他のループ検知手法が必要である。

### 2.2 管理者への通知漏れ

ループ発生時にはブロードキャストやマルチキャストに加え、未学習のユニキャストにより大量のパケットが複製されることが多い。それに伴い、輻輳が発生し、パケット損失が発生することがある。その結果、syslog や SNMP での通知のパケットが失われ、スイッチでループを検知できても管理者が気付けないことがある。その場合、利用者が障害を申し出るまで、障害の継続時間が長くなり得る。障害による被害を最小限に留めるため、管理者へのループ発生の通知漏れは可能な範囲で防ぐべきである。その一方で、数秒から数十秒以下といった短時間のループの発生は許容でき、管理者が気付けなくても良いと考えられる。

### 2.3 発生源ではないポートでのループ検知

ループ発生時には、発生源ではないスイッチのポートで

ループが検知されることがあり、混乱を招くことがある。これは、例えば、ループ検知パケットがループの発生源で複製された後、送信元のスイッチで再び受信された場合に発生する。そのため、何らかの手法で、発生源ではないポートでのループ検知を可能な限り排除できる必要がある。また、ループが発生したポートを特定できることが望ましい。

## 2.4 困難なループ発生源の特定

ループが発生した事実は既存手法などである程度検知できる。また、ループ検知されたポート、もしくは、その下流ポートの通信を手動もしくは自動で遮断し、組織内ネットワークから切り離すことで、組織内ネットワーク内でループを一時的に解決できる。これにより、被害を最小限に留めることはできるが、当該ポート配下のループの発生源でない端末の通信も遮断してしまう。そのため、根本的に解決するためには、ループの発生源を特定し、ループを解消することが必要となる。組織内のネットワークに利用者が独自のスイッチを接続している場合は、組織内ネットワークの管理者がループ箇所を発見することが難しい。また、利用者自身もループ発生の実事だけを管理者から伝えられても、ループ箇所の特定が難しいことがある。そのため、ループ解消に資する付加情報を明らかにし、管理者や利用者に伝えるべきである。少なくとも、ループ発生源の VLAN ID は明らかにすべきである。また、ループ発生源周辺の端末の MAC アドレスを明らかにできればなお良い。

## 3. ループ検知と解消支援

ここでは、より正確なループ検知手法を提案する。また、ループの根本原因の解消を支援するため情報を管理者に提示する手法を提案する。

まず、対象とするネットワークの前提条件を挙げる。次に、提案するループ検知と解消支援の概要を述べる。次に、各種ログからループを検知する手法について述べる。最後に、ループ解消を支援する情報の提示について述べる。

### 3.1 前提条件

以下の組織内ネットワークを前提とする。

- (1) L3 スイッチであるコアスイッチがある。
- (2) 全てのコアスイッチの MAC アドレスが既知である。
- (3) コアスイッチはスター状もしくはリング状にディストリビューションスイッチを収容している。
- (4) ディストリビューションスイッチはエッジスイッチを収容している。
- (5) コアスイッチ、ディストリビューションスイッチ、エッジスイッチは情報系センターなどの管理者が管理、運用している。
- (6) エッジスイッチは利用者の機器 (端末やスイッチなど) を収容している。

- (7) エッジスイッチの利用者の機器を収容しているポート (エッジポート) 全てで MAC アドレス認証を設定している。ただし、MAC アドレスの登録は必須ではなく、未登録の MAC アドレスも常に認証することとする。また、セキュリティ上必要なエッジポートでは、IEEE802.1x 認証もしくは Web 認証のネットワーク認証を設定していても良いものとする。
- (8) 全てスイッチの CPU 高騰とループ検知のログ、ネットワーク認証のログを syslog サーバが保持している。
- (9) エッジスイッチの一部はパケットダンプ機能を有している。

### 3.2 ループ検知と解消支援の概要

提案手法は以下の様な流れで、ループ検知とループの解消を支援する。

#### (1) ログの取得

MAC アドレス認証などのログとスイッチのログを syslog サーバに蓄積しておく。

#### (2) ログの解析によるループ検知

syslog サーバに蓄積されたログを解析し、ループを検知する。

#### (3) ループ解消を支援する情報の提示

ループ検知時に原因と考えられるエッジポート、VLAN ID、ループの発生源近辺の端末の MAC アドレス、ループしているパケットのダンプなどを提示する。提示にあたっては、ログやエッジスイッチの CLI (Command Line Interface) 経由で必要な情報を得る。

### 3.3 ループ検知

#### 3.3.1 ベンダ独自のループ検知パケットによるループ検知

syslog サーバに蓄積されるスイッチのログから検知する。図 1 にアラクスアラネットワークス社 (以下 AlaxalA という) のスイッチのループ検知の syslog 出力を示す。

#### 3.3.2 CPU 高騰によるループ検知

ループ発生時にはブロードキャストストームなどによりスイッチの CPU 高騰が発生することが多い。そこで、CPU 高騰を検知することよりループを検知する。これにより、ループ検知パケット受信前に発生したループをより早期に検知できる可能性がある。そして、輻輳でループを通知するパケットが失なわれてもループ発生を管理者が認知できるかもしれない。さらに、ループ発生源以外のスイッチでも、CPU 高騰が観測されてループを検知できる可能性がある。これは、発生源のスイッチが、ループによりブロードキャストパケットを複製し、ブロードキャストストームを発生させるため、近隣のスイッチでも CPU 高騰が発生するからである。

CPU 高騰は、syslog サーバに蓄積される図 2 に示す様なスイッチのログから検知する。

#### 3.3.3 MAC アドレス認証ログの急増によるループ検知

ループは、エッジスイッチが収容している利用者のスイッチなどで発生することが多い。ループ発生時には、発生源以外のポートに接続している MAC アドレスがループ発生源のエッジポートでも観測され得る。それに伴い、MAC アドレス認証が急増すると考えられる。

そこで、図 3 に示す MAC アドレス認証ログの急増により、ループを検知する。これにより、ループによる輻輳が発生しても、ループを検知できることが期待される。これは、ループ発生時に MAC アドレスがループ発生源で観測されると、MAC アドレス認証により認証されないパケットは転送されず、転送されるパケットは必ず MAC アドレス認証が成功しており、ログに残るはずだからである。つまり認証されない限り当該 MAC アドレスを送信元アドレスとして持つパケットは転送されないため、ループが発生し輻輳が発生するならば、必ず MAC アドレス認証に成功するはずである。

#### 3.3.4 コアスイッチの MAC アドレスによるループ検知

3.3.3 節に前述のとおり、ループ発生源のエッジポートでは、実際には収容していない機器の MAC アドレスも観測される。もし、コアスイッチの MAC アドレスが観測されたエッジポートが存在すれば、当該エッジポートがループ発生源と断定できる。

そこで、MAC アドレス認証ログで、コアスイッチの MAC アドレスが観測されれば、当該エッジポートでループが発生したとみなし、ループを検知する。これにより、より正確なループ発生源の特定が期待される。

### 3.4 ループ解消を支援する情報の提示

#### 3.4.1 ループ発生 VLAN ID の提示

図 1 にも示した様に、ループ検知で発生源であるエッジポートがログなどに記録されることは多い。その一方で、VLAN ID を含めて記録されないことがある。そこで、ループが発生した VLAN を自動的に算出し、管理者に提示する。これにより、ループが発生している VLAN ID が判明することで、その VLAN の管理者へ連絡などの対応を迅速に進めることが期待される。

VLAN ID の導出にあたっては、以下の様にループ検知手法毎に異なる手法により実現する。

- (1) MAC アドレス認証ログの急増: MAC アドレス認証のログ
- (2) ループ検知パケット: ループ検知ポートでのループ検知履歴

AlaxalA の場合は、show loop-detection logging コマンドによる図 4 に示す出力から、ループが発生している VLAN を取得できる。この様なコマンドが実装されていないエッジスイッチでは、ループが検知されたポートのスイッチの設定を確認することで取得

```
EVT E4 VLAN L2LD : ChGr(2) loop detection. from ChGr(2).  
EVT E4 VLAN L2LD : Port(0/49) loop detection. from port (0/49).  
EVT E4 VLAN L2LD : Port(0/38) inactivated because of loop detection. from port(0/33).  
EVT E4 VLAN L2LD : Port(0/33) activate by automatic restoration of the L2loop detection function.
```

図 1 ベンダ独自のループ検知パケットによるループ検知のログの例

```
EVT 01S E3 SOFTWARE Received many packets and loaded into the queue to CPU.  
EVT 01S E3 SOFTWARE Processed the packets in the queue to CPU.  
EVT 11/01 13:00:11 01S E3 SOFTWARE 00003303 1000:000000000006 Received many packets and loaded into the queue to CPU.  
EVT 11/01 13:00:16 01S E3 SOFTWARE 00003304 1000:020709150c11 Processed the packets in the queue to CPU.
```

図 2 CPU 高騰のログの例

できる。

(3) CPU 高騰: パケットダンプ中の IEEE 802.1Q ヘッダ

#### 3.4.2 パケットダンプの提示

3.3.2 節に前述のとおり、ループ発生時には CPU 高騰が発生することが多い。

そこで、ループ発生時に CPU 高騰が発生したスイッチにおいて、CPU 高騰の原因となっているパケットダンプを取得・解析し、ループの原因となり得るパケットの内容を管理者に提示する。これにより、ループの発生源の特定に資することが期待される。

例えば、AlaxalA の `show receive alarm dump` コマンドでは、CPU 高騰の原因となったパケットが図 5 の様に出力される。この出力は、イーサネットのフレームの宛先 MAC アドレスの先頭から 64 オクテットまでの 16 進ダンプである。この 16 進ダンプを人が確認するには時間を要する。そこで、出力を解析し、図 6 の様に、時刻、ポート番号、VLAN ID、送信元 MAC アドレス、宛先 MAC アドレス、イーサ種別、IP アドレスなどを提示する。また、ARP, RARP, Buffalo 社のループ検知, IPv4, IPv6, STP のパケットの解析に対応することとする。

## 4. 実装

### 4.1 MAC アドレス認証

鳥取大学 (以降本学という) の情報基盤機構が管理している AlaxalA とシスコシステムズ合同会社 (以下 Cisco という) のエッジスイッチの全てのエッジポートで MAC アドレス認証を実装した。3.1 節で前述したとおり、任意の MAC アドレスを認証 (つまりアクセス許可) した。講義室やパブリックスペースなどの情報コンセントではアクセス制限をかけるため、登録された MAC アドレスもしくは IEEE802.1x 認証, Web 認証によりアクセスを制限した。認証サーバとしては FreeRADIUS 3.0 を採用した。

### 4.2 ログ収集

AWS (Amazon Web Services, Inc.) の仮想マシン (以降 syslog サーバ) で syslog で各種ログを収集し保存した。ストレージとしては、比較的安価な Amazon S3 を利用し、

FUSE (Filesystem in Userspace) でマウントした。受信したスイッチのログ, MAC アドレス認証のログなどをそれぞれ別ファイルとして保存した。各ログファイルは 1 日 1 回ローテーションしており、2019 年 8 月 27 日から現在まで削除はしていない。

スイッチではログの送信先を上記 syslog サーバに設定した。そして、認証ログは FreeRADIUS の `linelog` モジュールで図 3 に示すログを生成し送信する設定をした。

### 4.3 ログの解析

syslog サーバで受信されるログをループ検知サーバで解析した。ループ検知サーバでログを受信する実装も考えられたが、以下の懸念から断念した。

- (1) 全スイッチなどでの syslog の送信先の追加は運用・管理コストや本学内トラフィックなどの増加となる。
- (2) syslog サーバからループ検知サーバへのログメッセージへの転送はトラフィック増により AWS 課金の増加となる。

そして、ループ検知サーバ上で、ループ検知スクリプトを Ruby で実装した。ループ検知スクリプトは、syslog サーバに ssh でログインし、syslog サーバ上で `grep` によりループ検知に関連するログのみを抽出する。そして、抽出されたログからループ検知サーバ上でループを検知することとした。この様に、syslog サーバ上では抽出のみを行うことで、syslog サーバとループ検知サーバとのトラフィック量を削減した。また、他の処理をループ検知サーバ上で実施することにより、syslog サーバの負荷を軽減し、ログの喪失の可能性を低下させた。さらに、ループ検知スクリプトでは AlaxalA の機種により異なる syslog 中のホスト名の表現の差異 (ホスト名か IP アドレス) を吸収した。加えて、AlaxalA のログではループ検知パケットの受信と CPU 高騰のメッセージに、一貫性が無く機種に依存して異なるため、それらの差異も吸収することとした。

### 4.4 パケットダンプの取得

パケットダンプをスイッチから取得するためには、各スイッチの IP アドレスが必要となる。そこで、A.1 節のス

```
Accept: [de:ad:be:ef:de:ad] MAC: de:ad:be:ef:de:ad, NAS: 10.xx.xx.xx, Port: Port 0/1, VLAN: xxxx from 10.xx.xx.xx
```

図 3 FreeRADIUS の linelog モジュールを活用した認証ログ

```
2021/xx/xx 16:50:55 0/51 Source: 0/51 Vlan: 4321
```

図 4 ループ検知の履歴

スイッチ制御ツールを実装し、各スイッチのホスト名と IP アドレスを事前に自動的に収集しておくこととした。そして、スイッチ制御ツールで得られたホスト名と IP アドレスを保存しておくこととした。syslog 中のホスト名から IP アドレスの導出が必要な場合には、スイッチ制御ツールが事前に生成したスイッチのホスト名と IP アドレスのデータベースから IP アドレスを導出することとした。機能に乏しい上に実装が複雑、運用が困難であることが予想される API は一切利用しなかった。また、スイッチなどに高負荷を招く SNMP も利用しなかった。

## 5. 評価

### 5.1 評価環境

本学の実ネットワークのキャンパスネットワークで評価した。キャンパスネットワークは 322 台のスイッチで構成されていた。エッジスイッチのエッジポートでは、ループ検知時にポートの通信遮断と復旧を自動化する設定であった。一方、上流ポート、並びに、下流ポート（下流のスイッチを収容するポート）では、ループ検知パケットの送信とループ検知だけを行い、ポートの通信遮断は実施しない設定であった。スイッチのログと認証ログは 4.2 節で述べたとおり、AWS 上の syslog サーバで受信し、保存していた。

そして、2021 年 4 月から 8 月までの評価期間中に発生したループに対して評価した。評価期間中に検知できたループの発生は A, B, C の 3 回であり、いずれも異なる日に観測された。継続もしくは断続して同じ原因で発生していたループは 1 つのループとみなした。

なお、CPU 高騰は 38 回と頻発していたが、2 回はループと検知できた。それ以外は、3 回がループに依らないものであり、33 回がループとは断定できないものであった。

### 5.2 ループ検知

表 1 に各ループ検知手法と検知の可否、ポート特定の可否を示す。全てのループを検知できた手法は存在しなかった。しかし、唯一 MAC アドレス認証によって検知できたループ C は、ストームの発生形跡も見受けられず、通信障害に至らなかった軽微なループであったと考えられる。そのため、通信障害を検知するという観点からは、ループ C を検知する必要性はそれほど高くはないと考えられる。

ループ C を除けば、ループ検知パケット、CPU 高騰によるループ検知が感度が高いと言える。ループ検知パケッ

トによる検知に関しては、ループ時のログの欠損も懸念されたが、ループ発生時に複数スイッチでループが検知されたため、ログの欠損があったとしても、ループそのものの検知は可能であったと考えられる。

一方、認証ログの増加による検知では、ループ A を検知できなかった。これは、ループ B ではループ発生時に単位時間当たりの認証数に 10 倍以上の増加が見られた一方、ループ A では変化が見られなかったためであった。このことから、認証ログの増加だけでは検知しきれないループがあることが明かとなった。

MAC アドレス認証による検知でもループ A は検知できなかった。これは、コアスイッチからのパケットの転送が少なかったことや MAC アドレス認証のパケットが損失されたことなどが考えられる。その一方で、MAC アドレス認証により検知できたループでは、発生源となるポートが容易に断定できた。

### 5.3 ループと断定できなかった CPU 高騰

上述のとおり、CPU 高騰の内、3 回は明らかにループではなもので、33 回はループとは断定できないものであった。

前者は、特定の端末からの異常な頻度の ARP リクエストの送信であった。ループ検知には関係ないが、通信障害や異常通信という観点では有用な検知であった。

後者に関しては、数秒間しか発生していたものであった。瞬間的にループが発生していた可能性も考えられるが、その様なループは許容できると考えられる。

### 5.4 VLAN ID とパケットダンプの提示による対応支援

5.2 節に前述のとおり、MAC アドレス認証によりコストの MAC アドレスをエッジポートで観測された場合は、ループの発生源が容易かつ断定的に特定できた。一方、ループ検知パケットや CPU 高騰によるループ検知時には、複数のスイッチやポートでループが検知され、ループの発生源であるエッジポートを断定的に特定することが難しかった。

そこで、ループ検知パケットで検知できた場合に VLAN ID が提示されることで、管理者がループ発生源のエッジポートを特定するのに 1 つの判断材料とすることができた。

また、ループ発生時に CPU 高騰状態になっているスイッチでパケットダンプを提示することで、送信元 MAC アドレスも判明し、ループの発生源となっているネットワークの特定の補助となった。しかし、CPU 高騰時にパケットダンプに提示されるパケットの 1000 個中 988 個は、大量に複製されたループしている元は 1 つのパケットであり、得ら

Date 2021/xx/xx xx:xx:xx JST				
No.	Date/Time	Size	Port	Data
1	2021/xx/xx xx:xx:xx.xxx	68	0/48	ffffffff ffffdead beefdead 8100801c 08060001 08000604 0001dead beefdead acxxxxxx 00000001 0000acxx xxxxa510 923d7090 71f259b8 5dabfd07 00000000

図 5 AlaxalA のパケットダンプ

```
2021-xx-xx xx:xx:xx +0900 0/48 28 dead.beef.dead ffff.ffff.ffff 806 ARP request (1) 172.xxx.xxx.xxx 172.xxx.xxx.xxx
```

図 6 AlaxalA のパケットダンプの解析

れる情報量は必ずしも多いとは言えなかった。AlaxalA のループ検知パケット自体がパケットダンプ中に大量に観測されることもあり、決定的な情報が得られないこともあった。しかし、いずれの場合でも少なくとも VLAN ID は明らかとなった。

## 6. 考察

### 6.1 最適なループ検知手法

5.2 節に前述のとおり、全てのループを検知できる手法は存在しない。また、感度が高い検知手法であっても、ループの発生源となっているエッジポートの特定が難しいこともある。これらのことから、最適なループ検知手法は無く、複数の手法を併用し、それぞれの長所を生かしながら、ループを検知し対応していくことが重要と考えられる。

### 6.2 上流ポートでの誤検知の抑制

AlaxalA のスイッチに実装されているループ検知機能である loop-detection uplink-port [4], Cisco の Catalyst 9300 シリーズの loop-detect source-port [5] などにより、上流ポートでの誤検知を抑制できる。しかし、uplink-port では上流ポートにループ検知パケットを送出しないため、上流ポートでのループを検知できない。上流ポートでループが発生するのは頻度が少なく、無視して良いとも考えられるかもしれない。しかし、鳥取大学では過去に実際に発生しており、大規模障害を招いたため、無視するのが難しい。加えて、uplink-port を適用する場合、下流スイッチを接続する下流ポートへもループ検知パケットを送信しない設計としなければならない。これは、エッジ側ポートでループ検知パケットを送信する場合に、ループ発生時に広範囲に影響を及ぼすポートで通信遮断する可能性があるためである。さらに、管理者の理解不足により設定漏れや設定ミスも発生する可能性もあり、uplink-port の適用により大きな障害を招く可能性も否定できない。この様に uplink-port の適用は制限されるため、本提案などの何らかの別の手法で、発生源ではないポートでのループの誤検知を可能な限り排除することが望ましい。

## 7. 関連研究

ループ検知手法としては、STP [1] や Cisco や AlaxalA などのベンダ独自のループ検知パケットを用いるものがある。しかし、ループ検知パケットを透過的に転送しないスイッチも存在するため、エッジポートでのループ発生を完全には防止できない。

ブロードキャストやマルチキャスト、未学習のユニキャストにより、ブロードキャストドメイン全体に送信されるパケットが、閾値を超えて受信されたことをストームとして検知する手法がある。ループ発生時にはストームにより通信障害が発生するため、ストームを検知してエッジポートで通信遮断することで障害を防止できる。しかし、閾値による検知であり、誤検知もあり得るため、ネットワークの状況を調査して適切な閾値を算出する必要があり、運用が難しい。

永井らは、トラフィック量が事前に設定された閾値を超える場合にループを検出し、ループの発生箇所と影響を受ける経路を示す手法を提案している [6]。この手法もストーム検知と同様閾値による検知であるため、適切な閾値の算出のための運用が難しいと考えられる。

野呂らは、eBPF を用いて STP が適用不可な状況で VXLAN や VLAN 上などでのループ検知手法を提案している [7]。しかし、eBPF でネットワークのトラフィックを監視できる必要があり、キャンパスネットワークといった物理的に広範囲にエッジスイッチが点在するネットワークに適用するのは難しいと考えられる。

## 8. おわりに

本稿では、ループ検知パケット及び CPU 高騰、MAC アドレス認証ログの急増、エッジスイッチのエッジポートでのコアスイッチの MAC アドレス観測によるループ検知を提案した。また、ループの解消に向けて VLAN ID やループ発生源となっているエッジポートの提示手法を提案した。そして、本学の実ネットワークで評価し、ループ検知パケットと CPU 高騰によるループの検知が感度が高いことが明らかとなった。また、ループの発生源の特定にはよ

表 1 ループ検知とポート特定

検知手法	A		B		C		備考
	検知	ポート特定	検知	ポート特定	検知	ポート特定	
ループ検知パケット	○	△	○	△	×	×	ポート特定に別途上流ポートの確認を要した。 複数スイッチで検知しポート特定が難しかった。 断定的なポート特定はできなかった。
CPU 高騰	○	×	○	×	×	×	
認証ログ増加	×	×	○	△	×	×	
MAC アドレス認証	×	×	○	○	○	○	

り調査を要し、断定することが難しいことも明らかとなった。その一方で、エッジポートでのコアスイッチの MAC アドレスの観測では、発生源を容易に断定的に特定できた。

今後は、ログの設定がなされていないために評価できなかったストームによるループの検知といった他の手法も評価し、複数手法を併用してループ検知の精度を高めたい。

#### 参考文献

- [1] IEEE Std. 802.1w-2001: *Local and Metropolitan Area Networks. Rapid Reconfiguration of Spanning Tree*, IEEE (2002).
- [2] 802.1ab 2004, I. S.: *Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks: Station and Media Access Control Connectivity Discovery*, The IEEE Standards Association (2004).
- [3] Red Hat, Inc.: Ansible, <https://www.ansible.com/> (2021). Accessed: 2021/09/08.
- [4] アラクサラネットワークス株式会社: AX2300S ソフトウェアマニュアル Ver.1.0 コンフィグレーションガイド Vol.2, <https://www.alaxala.com/jp/techinfo/manual/#AX2300S> (2021). Accessed: 2021/09/08.
- [5] Cisco Systems, Inc.: Layer 2 Configuration Guide, Cisco IOS XE Amsterdam 17.2.x (Catalyst 9300 Switches): Configuring Loop Detection Guard (2021). Accessed: 2021/09/08.
- [6] 永井隆広, 新井敏正: ネットワーク監視システム (2001).
- [7] 正明野呂, 陽介高野, 直樹小口, 俊二阿部: eBPF による MAC 層ループ対策, 技術報告 62 (2020).

## 付 録

### A.1 スイッチ制御ツール

スイッチ制御ツールは、CLI を通じてスイッチに接続し、情報取得や設定を投入する Ruby で記述されたライブラリである。コアスイッチの IP アドレスを与えれば、LLDP (Link Layer Discovery Protocol) [2] と CDP (Cisco Discovery Protocol) により全てのスイッチを自動的に発見することもできる。Ansible [3] の活用も考えられたが、CLI 出力の文字列操作に関して Python よりも優れていると考えられる Ruby を採用した。

#### A.1.1 CLI へのログイン

telnet と ssh による CLI を通じたスイッチへのログインを実装した。対応にあたっては、Ruby gem の net-telnet

0.2.0, net-ssh 5.0.2, net-ssh-telnet 0.2.1 を用いた。認証方式としてはパスワード認証のみに対応した。本学ではスイッチなどの導入ベンダなどに依りて異なるパスワードを用いている。スイッチなど毎にアカウント情報を事前に定義するのは運用コストが増加するため、複数のアカウント情報を定義可能とした。そして、複数のアカウント情報を自動的に試行し、ログイン可能とした。連続して複数回ログイン失敗すると TCP のコネクションが切断されることがあるが、その場合には再接続する様に実装した。

#### A.1.2 接続直後のエスケープシーケンス処理

接続直後には、スイッチが送信するプロンプト文字列を受信後に、スイッチ制御ツールがコマンドの文字列をスイッチへ送信することでコマンドを実行できる。送信したコマンドが破棄される可能性を無くするため、プロンプト文字が受信されるのを待つ必要がある。しかし、本学内にあるネットワーク機器 (パロアルトネットワークス社, アルバネットワークス社, Cisco, NEC, Alaxala) の内, Alaxala のスイッチだけは接続直後にプロンプトを表示する前にエスケープシーケンスを送信する。具体的には, CSI (Control Sequence Introducer) の DSR (Device Status Reports) で現在のカーソル位置を取得する `0x1b[6n` がスイッチから送信される。Alaxala の場合, DSR に対して現在のカーソル位置をスイッチ側に伝えなければ、続くプロンプトなどがスイッチから送信されず、コマンドの実行ができない。また、今回用いた Ruby gem の net-ssh-telnet はエスケープシーケンスを解釈や処理はしない。そこで、Alaxala のスイッチに対応するため、スイッチ制御ツールで上述の DSR を受信した場合に、カーソル位置を (1,1) として、`^1;1R` をスイッチ側へ送信することとした。

また、CLI でログイン後にスイッチのメーカーや機種を検出するため、Alaxala 以外のスイッチへの接続も考慮しなければならない。そこで、接続直後のプロンプト文字列の受信を待つ正規表現として、`/(?:\e\[6n|[$%#>] ?\Z)/` を net-ssh-telnet に与えることとした。これによって、Alaxala を含めた学内のスイッチ全てでプロンプト文字列を正しく受信して処理できるようになった。

#### A.1.3 制御文字やエスケープシーケンス処理

Alaxala のスイッチでは、接続直後だけでなく通常の出

力でも、制御文字やエスケープシーケンスにより表示を制御していた。例えば、\b (バックスペース) や\E[K (行末までの文字を削除) が送信されていた。また、理由と契機が不明なヌル終端文字の送信が観測された。そこで、接続後にスイッチから送信されるバイト列は1バイトずつ処理し、バックスペースなどによる文字列の削除にも対応した。また、ASCII 文字以外の文字列は無視することとした。

#### A.1.4 メーカーと機種検出

パケットダンプを実装しているスイッチは本学においては AlaxalA のみであった。また、AlaxalA であっても機種が限られており、本学で確認できたのは AX2500 と AX260 シリーズのみであった。そのため、スイッチのメーカーと機種を検出する機能を実装した。メーカーと機種を1つの手法で検出することは難しかったため、メーカーの検出と機種の検出それぞれ段階的に実装することとした。

まず、メーカーを以下で検出した。

##### (1) ログイン直後にスイッチから送信される文字列

ssh で接続直後に ALAXALA を含む文字列を送信する AlaxalA のスイッチのみを検出した。なお、接続直後に受信されるエスケープシーケンスで AlaxalA とメーカーを断定しても良かったかもしれないが、今回検証したメーカー以外でもエスケープシーケンスが利用されている可能性も考慮し、エスケープシーケンスによりメーカーを特定することは避けた。

##### (2) ページャの無効化コマンドの成功

システムの情報を出力するコマンド (show system など) の出力時に機種に依ってはページャが有効になってしまい、機種の検出が複雑になった。例えば、AlaxalA の AX8600 では出力する情報が多くページャが有効になってしまう。そこで、ページャの無効化のコマンドが各社で異なる点に着目し (表 A.1)、当該コマンドの成功でメーカーを検知することとした。各メーカーのエラーメッセージの全てのパターンにマッチしないことを以って、コマンドが成功したとみなした。なお、ページャの無効化コマンドの失敗時のエラーメッセージによるメーカーの検出も検討したが、Cisco と AlaxalA, Aruba などのエラーメッセージが共通で区別ができなかった場合があったため、断念した。

そして、メーカーを検出した後、show system コマンドで機種を判定した。

#### A.1.5 コンソールログの抑制と削除

CLI 経由でコマンドを実行しその出力を得る場合、コマンドの実行とは無関係のエラーや警告メッセージといった帯域外メッセージが出力されることがある。コマンドの出力結果を解析する場合、帯域外メッセージは除外しなければならない。そのため、帯域外メッセージを抑制、もしくは

表 A.1 ページャの無効化コマンド

メーカー	機種	コマンド
Cisco	catalyst	terminal length 0
Cisco	WLC/WiSM	config paging disable
AlaxalA		set terminal pager disable
Palo Alto		set cli pager off
Aruba	WLC	enable no paging
NEC	IX2215	configure terminal terminal length 0

は、削除することとした。まず、メーカー検出後に表 A.2 に示すコマンドを投入し、帯域外メッセージを抑制した。

しかし、AlaxalA のスイッチでは、これらのコマンドだけでは全ての帯域外メッセージを抑制できなかった。そのため、AlaxalA の帯域外メッセージのみ削除することとした。AlaxalA の帯域外メッセージは CLI のプロンプト、改行、帯域外メッセージ、改行、CLI のプロンプトで構成されることに着目し、CLI のプロンプトで挟まれた1行のメッセージを帯域外メッセージと検出し、削除することとした。削除にあたっては、コマンドの出力の終端との区別も考慮に入れた。一般的にコマンドの出力の終端は CLI のプロンプト (改行に続くホスト名など) の文字列にマッチすることで検出することが多い。しかし、AlaxalA の帯域外メッセージにはプロンプトも含まれているため、単純にプロンプトを出力の終端として検出してしまうと正しくコマンドの出力を得られない。そこで、帯域外メッセージを検出して削除した場合は、再度スイッチの出力を読み込み、コマンド実行後に出力されるプロンプトまで正しく出力を読み込む様に実装した。

表 A.2 コンソールへのログ出力の無効化コマンド

メーカー	機種	コマンド
Cisco		no login console
AlaxalA	AX8600	username default_user logging-console event-level 0
AlaxalA	AX3800 AX3600 AX2500 AX260	set logging console disable E{3-9}
AlaxalA	AX2130 AX2230	無し