Vol.2020-DPS-182 No.62
Vol.2020-CSEC-88 No.62
2020/3/13

eBPFによるMAC層ループ対策

野呂 正明[1,a)]　高野 陽介[1]　小口 直樹[1]　阿部 俊二[2]

概要：ネットワークの仮想化技術の進展・普及に伴い，複数の組織間で仮想化されたMAC層のネットワークを相互接続するケースが増加している．MAC層のループ対策として，Spanning Tree Protocol(STP)を用いることが望ましいが，運用体制やネットワークの運用ポリシーが異なる組織間の接続では，正しく設定・運用することを期待できない場合も増加している．
STPを用いないMAC層のループ対策としては，LAN機器にループやブロードキャストストームを検出するセンサを搭載し，問題が検出されたリンクを切断する機能が搭載されているものの，この切断機能の対象が物理ポートに限られており，特定の仮想MAC層のリンクだけを切断することができない．そのため，MAC層のループが発生する度に，人手での解析と対策を行っている広域ネットワークが存在する．このような問題を解決するため，仮想化されたMAC層のネットワーク上に接続した計算機にループを検出するセンサとLAN機器や仮想ネットワークを構成している計算機の設定を自動で制御する機能を配備することで，従来の手法が適用できなかったケースでも対応を可能とした．

キーワード：extended Berkeley Packet Filter, Ansible, MAC層ループ

Detection and measures of MAC layer loop using eBPF

Abstract: It is getting easy to use virtual layer 2 network not only 802.1q but also 802.1ae or VxLAN. This tend leads to use virtal layer 2 network between different organization over wide area netowks. The operator of this kind of layer2 network sometimes causes layer 2 loop network. Usually, the operator use spanning tree protocol to prevent layer 2 netowrk loop. But, it is sometimes diffucult using stp on network between difficult organization, becase of operation policy. In such situation, it is common to use network loop detection sensor in network equipment for a long time.
However, the network loop detection sensor in network equipment is not enough progress to catch up recent virtual network technology. Because of above reason, the network operator of some organization watch and analyze network trouble caused by layer 2 loop. This manual operation is increasing becase spreading use of virtual layer 2 network.
Therefore, we propose software virtual layer 2 loop detection sensor to decrease operation cost of wide virtual layer 2 network. The our software system use extented berkeley packet filter for highspeed packet processing. And, our system also use network operation open source software named ansible for easy to extend new network equipment or new virtual network technique.

Keywords: extended Berkeley Packet Filter, Ansible, MAC layer loop

1. はじめに

ネットワークにおけるMAC層のループ対策は非常に古くから行われてきており，LAN機器間でループを検出するプロトコルの通信を行う方法 (STP：Spanning Tree Protocol)[1]の他に，ループによって発生する現象を検出した場合に，検出したネットワークの物理的なポートを切断する方法が多くのLAN機器に採用されており，一部のベンダの機器では，MAC層のマルチキャストやブロードキャストのパケットを廃棄する方法が利用されている．

一方，仮想化されたMAC層を組織間で接続する例も珍

[1] 富士通研究所 (Fujitsu Labratories)
211–8588 神奈川県川崎市上小田中 4–1–1
4–1–1 Kamikodanaka, Kawasaki city, Kanagawa 211–8588, Japan
[2] 国立情報学研究所 (National Institute of Informatics)
101–8430 東京都千代田区一ツ橋 2–1–2
2–1–2 HItotsubashi, Chiyoda–ku, Tokyo 101–8430, Japan
[a)] noro@fujitsu.com

Vol.2020-DPS-182 No.62
Vol.2020-CSEC-88 No.62
2020/3/13

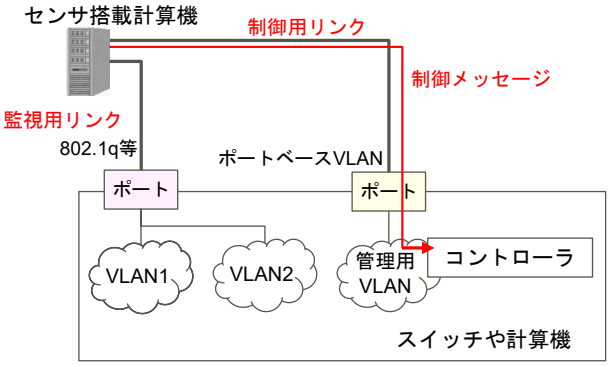

図 1　LAN 機器を制御する場合

Fig. 1　Loop prevention system on phisical system.

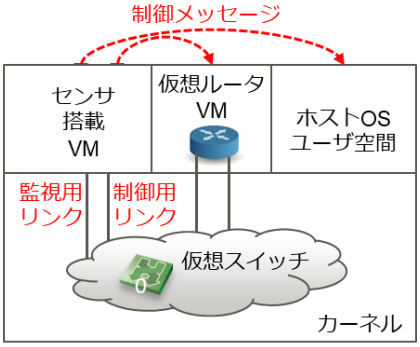

図 2　計算機内の仮想ネットワークを制御する場合

Fig. 2　Loop prevention system on virtual environment.

しくない状況であり，古くは 802.1Q(TAG VLAN)[2] を用いて複数の組織のネットワークを束ねて，上流の広域ネットワークに接続する使い方が多く用いられてきたが，近年は 802.1Q のタグを多重に付与する方法 [3] や，VxLAN[4] を用いて，異なるホスト・複数拠点間の MAC 層のネットワークを接続する例も増えている．

このような組織間接続において，組織間の運用ポリシーの違いなどの問題から STP の利用ができない場合も増えており，ループを検出して自動で切断するような対策の仕組みを仮想ネットワークに搭載することが求められている．

これに対して，計算機内 (もしくは，計算機間) で仮想化されたネットワークにおいて適用可能なループ対策は存在しないため，ユーザ自らがそれを開発する必要がある．また，LAN 機器におけるループ対策も，切断対象が種類が増えている仮想化技術に追いついていないため，多くの LAN 機器で利用可能な物理的なポートを切断する方法を用いた場合，複数の組織が LAN 機器の 1 つの物理ポートに同居している環境では巻き添えとなる組織が出る．

以上のような問題を解決するため，extended Berkeley Packet Filter(eBPF)[5] を用いて，MAC 層のネットワークがループしている場合に発生する現象を検出するセンサを実装した Linux を制御対象に接続し (図 1, 2)，ループを検出した場合に LAN 機器や他の計算機の仮想ネットワークの構成を制御するために広く利用されているミドルウェアである Ansible[6] を用いて制御する．これにより，仮想ネットワークを構成する機器のベンダや利用している仮想ネットワークの技術などに依存しない仕組みを実現した．

2. 従来手法

本研究で想定するネットワークは，さまざまな仮想ネットワーク技術を活用したネットワーク (図 3) であり，このネットワークを構成する装置としては従来のように LAN 機器を組み合わせたものだけでなく，計算機内に仮想ネットワークを作成し，その計算機間を接続するようなものも対象とする．

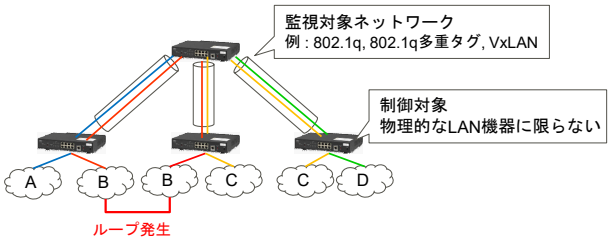

図 3　想定するネットワーク環境

Fig. 3　Out target network based on network virtualzation technologies.

2.1 従来のループ検出方法

正常なネットワークにおける MAC 層は Tree 状になっているため，ブロードキャストパケットがある装置から発信されると，Tree の枝を経由して発信装置以外の全ての末端の装置に配送される．これに対してループした環境では，発信装置がつながっているあるブロードキャストパケットを最初に受信したリンクとはとは異なるリンクから元のブロードキャストパケットが戻ってくるため，再度，全てのリンクに転送してしまう．このことから，1 つのブロードキャストのパケットがネットワークの転送能力の速度でループを周り続けることとなる．これをブロードキャストストームと呼ぶ．

この現象は，ブロードキャストパケットにかぎらず，全ての装置に転送されるパケットであれば発生するため，マルチキャストパケットや経路上の全ての装置 (イーサネットスイッチ等) にアドレスが学習されていないユニキャストアドレス宛のパケットでも類似の現象が発生する．

このような状況で該当ネットワークに接続している装置は，単位時間あたり大量のブロードキャストパケットを受信するため，この単位時間あたりのブロードキャストパケットの受信数を計測することで問題ありと判断する方法 (図 4) であり，多くの LAN 機器ベンダに採用されている．

もう一種類の現象は，パケットがループする場合に起きる現象を利用する．ループしている状態では，自分が送信 (もしくは転送) したブロードキャストやマルチキャストが再度自分に届く．この性質を利用してループを検出する方

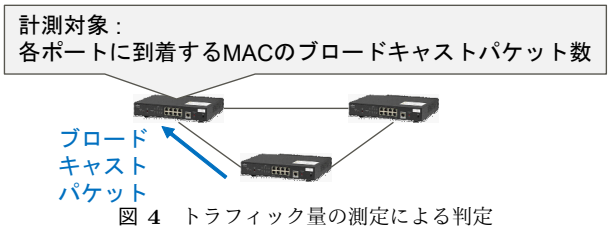

図 4　トラフィック量の測定による判定
Fig. 4　Loop detection by measuring packet number.

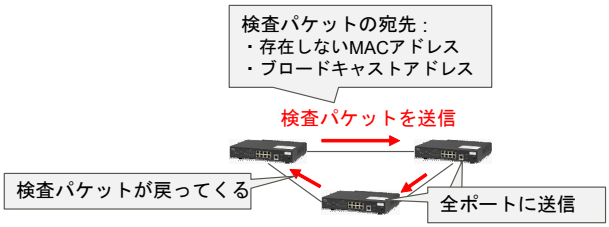

図 5　受信パケットのソースアドレスを確認する方法
Fig. 5　Source address checking for Loop detection.

法として [7] や [8] がある．[7] はパケットのハッシュを取得し，同じハッシュ値を持つパケットが通過するとループと判断する．[8] はループ検出用の特殊なフォーマットのブロードキャストパケットを送信し，受信した全パケットのソースアドレスを確認することで，自分が発信したパケットが戻ってきているか否かを判定する (図 5)．

仮想化されたネットワークで動作するループ判定手法は，同じ回線に多重化されたネットワークのうちのどれがループを起こしているか判定するために，各種の仮想化技術の全てに対応する必要があるが，多くのベンダの LAN 機器が対応している仮想化技術は非常に限られている．

さらに，計算機内で構成された仮想ネットワークを相互接続している環境では，計算機内で動作するループ判定のセンサが必要となるが，このようなループ検出のセンサは今のところ存在していないため，ユーザが自分の環境に合わせて開発する必要がある．

2.2　ループ発生時の対応

上記のループ検出手法のうち，いずれかを用いてループを検出した場合に，ループを切断する等の対策を取る必要がある．

従来の LAN 機器 (イーサネットスイッチ等) ではループを検出した場合に，物理ポートでループ検出のセンサを動作させ，ループが検出された物理ポートを OFF にすることでループを切断するものの他に，ブロードキャスト，マルチキャストに絞って廃棄することで，影響を少なくするものある．

1 章で述べたように，本研究では MAC 層を 1 つの物理接続に 802.1Q や VxLAN を用いて複数の回線を束ねた環境を想定しているが，制御対象の仮想化技術はごく限られており，仮想化されたネットワークの切断に対応しているベンダも非常に限られている．さらに，センサと同じく計算機内の回線には対応していないため，仮想化されたループ回線の切断を行う場合は，ループの検出と同じくユーザが自分で対応する必要がある．

一方，広域網を経由して仮想化された複数組織の回線を束ねて運用している組織では，ネットワーク下流の組織に所属する回線を切断することができないため，検出したループに関するアラームを上げるだけの運用をしている場合もある．

2.3　パケットキャプチャ技術

ループを検出するセンサを開発するためには，パケットをキャプチャする必要があるが，現在利用可能なキャプチャ用のインターフェイスは，「packet capture(pcap)[9]」，「Data Plane Development Kit(DPDK)[10]」，「eBPF」の 3 種類がある．

pcap は非常に古くから存在するパケットキャプチャ用のインターフェイスであるが，ネットワークインターフェイスを通過するパケットをユーザ空間のプログラムで拾う構造であるため，大量のパケットが流れる環境では，取りこぼしが多く発生する．

それに対して DPDK はパケットキャプチャの能力が高く，一時期採用が増えたものの，キャプチャを行うネットワークインターフェイスのハードやデバイスドライバに対応機能が必要なため，利用可能な環境が限られるという問題がある．そのため，この制限が問題となり，利用できない場合が多い．

近年利用が増加している eBPF は，Linux カーネル内の様々なレイヤ (NIC のドライバの出口やカーネルのプロトコルスタックのさまざまな部分) にカーネル内で動作する小型の VM を挿入し，パケット到着時に該当の VM 内のプログラムが実行され，そのプログラムでパケットのデータを検査/加工することができる．実際のカーネル内のパケット処理は，VM 内のプログラムの実行が終わるまで待たされるため，カーネルのプロトコルスタックが受け取ったパケットは必ず処理することができる．さらに，DPDK とは異なり利用するハードや VM の環境への依存が低い．以上のような理由から，本研究ではパケットキャプチャインターフェイスとして eBPF を利用する．

3.　提案システム

本研究では，eBPF を用いたループ検出のセンサを実現し，このセンサと様々なネットワーク機器や計算機の設定を変更可能なミドルウェアである Ansible を接続して，ループの切断を行う (図 6)．

センサが動作する計算機と制御対象の LAN 機器 (もしくは別の計算機) 間のネットワークは，監視対象のネットワークとは別のものを用いる．これにより，監視対象のネット

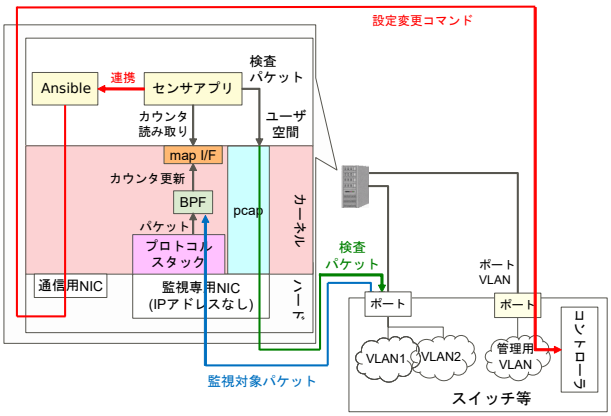

図 6　提案アーキテクチャ
Fig. 6　Proposed loop sensor architecture.

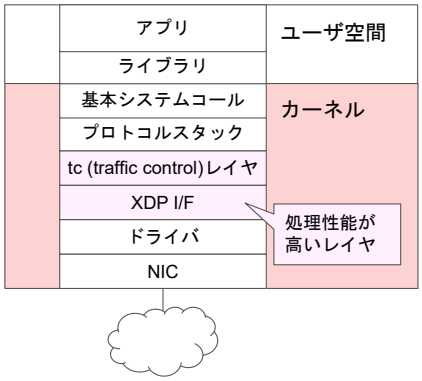

図 7　プロトコルスタックと eBPF の関係
Fig. 7　eBPF interface and linux protocol stack.

ワークで問題が発生した場合でも，制御対象との通信が阻害されない．

また，今回開発したセンサは実際のユーザネットワークに適用するため，ユーザ環境に合わせて検査対象の仮想ネットワークは 802.1Q の TAG VLAN となっている．

今回開発した MAC のループ対策機構の特徴は以下のようなものである．

- eBPF を用いることにより，DPDK と異なり，様々なネットワークインターフェイスが利用でき，pcap と比較してパケット取りこぼしを削減できる．
- 検査対象のパケットがブロードキャストのみであるため，検査対象の LAN 機器や仮想ネットワークにパケットミラーリング等の機能が不要な上，通常の計算機やハイパーバイザを用いた仮想環境中の VM でも動作させることができ，センサを動作させる環境を選ばない．
- Ansible を制御プログラムに用いるため，さまざまな制限対象に対応する場合に必要なコストが小さい．

3.1　eBPF を用いたループ検出センサ

先に述べたように，パケット検査可能な eBPF 用のインターフェイスは，ネットワークインターフェイスのドライバのすぐ上 (XDP[11][12][13]) に加えて，プロトコルスタックの様々な場所に存在し，XDP のドキュメントによると，XDP がカーネルにかかる負荷が一番小さい．

そのため，パケットを検査するためには XDP を用いることが望ましいが，現在のバージョンの XDP では，802.1Q のタグが eBPF のプログラム内から読み取ることができないという制限があるため，ブロードキャストストーム検出のセンサでは，XDP の代わりに，Linux の tc 部分のインターフェイスを用い，自分が投げたパケットを検出するためのインターフェイスに XDP を用いている (図 7)．

3.1.1　ブロードキャストストームの検出

ブロードキャストストームの検出は，tc のインターフェイスを用いているため，センサが動作する Linux ホストが受信した全てのパケットが eBPF のプログラムにかけられる．

そのため，監視対象のインターフェイスで受信したパケットか，制御用インターフェイスで受信したパケットかを区別するため，TAG VLAN の情報がパケットに付加されているか否かを最初に判断し，TAG VLAN の情報が付加されていないパケットに関する詳細な検査は行わず，そのままカーネルのプロトコルスタックにパケットを処理させる．

パケットに TAG VLAN の情報が付加されていた場合は，ブロードキャストパケットか否かを検査し，ブロードキャストパケットであった場合は，ユーザ空間のプログラムにデータを引き渡すためのインターフェイスである MAP のうち，ブロードキャストパケットに含まれている VLAN の ID に対応するカウンタをインクリメントする．

最後に，TAG VLAN の付与されていたパケットは監視専用のインターフェイスで受信したものであり，OS の上位層では不要なものであるため，全て廃棄する．これにより，ブロードキャストストームが発生した場合に，センサが動作する計算機環境のカーネルが無駄なプロトコルスタックでの処理で負荷があがることを防止する．

ユーザ空間のセンサプログラム本体では，周期的にカーネル内の eBPF プログラムからデータを読み出すインターフェイスである MAP を読み出し，各 VLAN 毎の単位時間あたりのブロードキャストパケット数を計算し，この値がしきい値を超えていた場合に，該当の VLAN に障害 (ループ) が発生していると判定し，制御プログラムに VLAN ID を引数として，切断の処理を呼び出す．

3.1.2　自分が発信したパケットの検出

このセンサでは，Tzeng[8] と類似の方法を用いるが，Tzeng [8] とは異なり，センサが動作する計算機環境から，周期的に ARP を送信する．さらに，監視対象のネットワークが接続されているインターフェイスだけを XDP のインターフェイスで監視し，eBPF のプログラムで受信したパケットが自分が送信したものか否かをソース MAC アドレ

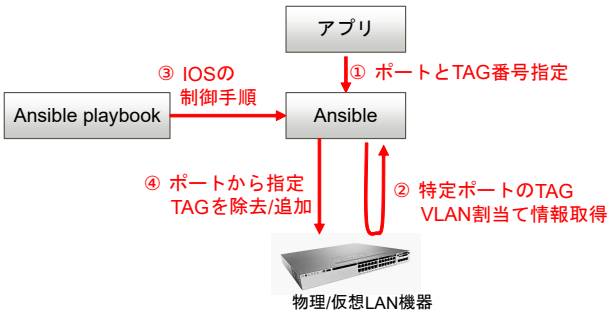

図 8　Ansible による TAG VLAN の切断
Fig. 8　Disconnect TAG VLAN link by Ansible

スで判定する．

ここにおいて，XDP のインターフェイスでは，TAG VLAN のデータを読み取ることができないため，検査対象のネットワーク毎に，異なる送信 MAC アドレスを用いてパケットを送信し，受信したパケットの MAC アドレスを検査することで，どの VLAN でループが発生しているかを判断する．

また，検査用のパケットは ARP(MAC のブロードキャスト) を python のライブラリである scapy を用いて送信する．この際，個別の TAG VLAN のインターフェイスではなく，全ての TAG を受信する trunk インターフェイスから，scapy で合成した TAG 付きパケットを送信する．また，この際，解決すると回答先の IP アドレス，MAC のソースアドレスはネットワーク上に存在しないアドレスを用いることで，無駄な回答パケットの発生やアドレスの衝突を避ける．

3.2　Ansible によるネットワークの制御

ネットワーク制御用のプログラムはセンサから起動されるが，引数としてループが存在すると判断された TAG VLAN の ID が渡される．

制御プログラムは最初に LAN 機器もしくは，仮想ネットワークを構成する計算機にログインし，ループが検出された VLAN を除去するコマンドを生成し，制御対象の機器 (LAN 機器もしくは計算機) に Ansible のモジュールを用いて投入する (図 8)．

4.　評価

本研究の仕組みでは，制御対象の仮想ネットワークの技術や対象の機器に縛られることがなく，新しい仮想ネットワーク技術や異なるベンダの LAN 機器等に対応するためのコストが低いなどの定性的な特徴は除き，パケットキャプチャの成功率，LAN 機器の制御に必要な時間といった性能面の評価を行った．

4.1　パケット受信時の性能

図 9 が評価の環境であり，Windows PC の中に Virtual Box を用いて 2 台の仮想マシンを用意し，一方からセンサ搭載の仮想マシンに対して MAC のブロードキャストパケット (パケットサイズは 100 バイト) を連続送信し，センサ搭載仮想マシンでどの程度拾うことができたかの成功率を評価した．なお，ループ検出用のセンサを pcap に用いた場合を比較対象とした．

図 9 の評価用パケット送信プログラムのパケット送信処理の間の待機時間をパラメータとして，単位時間あたりのパケット量を変化させた場合の測定した結果が図 10 である．この図からわかるように，検査対象のパケット量が増えた場合でも，eBPF を利用したセンサの方が，パケットの受信成功率は高い．

4.2　Ansible による機器制御の処理時間

次に，ループを検出した場合に LAN 機器を制御するために必要な所要時間を評価した．図 11 は評価環境で，Ansible を搭載した仮想マシンを KVM 上に作成し，同一のサブネット上に接続した Catalyst3850 および，NFV 環境のための仮想 Juniper スイッチ (vQFX) を制御した．

評価環境では，スイッチの 1 つのポートに 802.1Q のタグを複数割当て，Ansible を用いて特定のタグを設定から取り除くが，Catalyst では現在割当てられている TAG のリストを取得し，そこから特定の TAG の定義を取り除いたリストを作成し，新しいリストでタグ VLAN の定義を投入する．それに対して vQFX の場合は，削除対象のタグの番号を設定変更のコマンド (delete) に投入するだけで良いため，余分な処理が発生しない．

この制御処理の処理時間の 100 回実施した平均値は，vQFX の場合で 9.18 秒であった．Catalyst の場合は，実行中のコンフィグ (running-config) だけを変更する場合で 8.72 秒，再起動時に読み込まれるコンフィグ (startup-config) まで変更する場合は 15.47 秒であった．

図 12 は，市販機器でのループ検出とポート切断の評価環境である．ループ検出の条件として，ブロードキャストパケットの毎秒あたりのパケット受信数 (pps) で指定し，ループを検出した場合にポートを切断 (link down) させる処理を指定している．

この場合，PC によるブロードキャストパケットの連続送信から，ポートの切断まで約 1 秒で行われており，ループの検出後の処理時間はごく僅かな時間で行われている．

以上の測定結果からわかるように，Ansible による機器の制御は比較的実行時間が長いという問題があるが，ループが発生している期間の影響は，ユーザの利用可能な帯域が減るのみであること，TCP のコネクションタイムアウトが多くの OS で 30 秒であることを考慮すると，ループ切断の所要時間が測定結果のレベルであれば大きな問題とならないと考えている．

また，複数組織のネットワークを仮想化して，広域網で

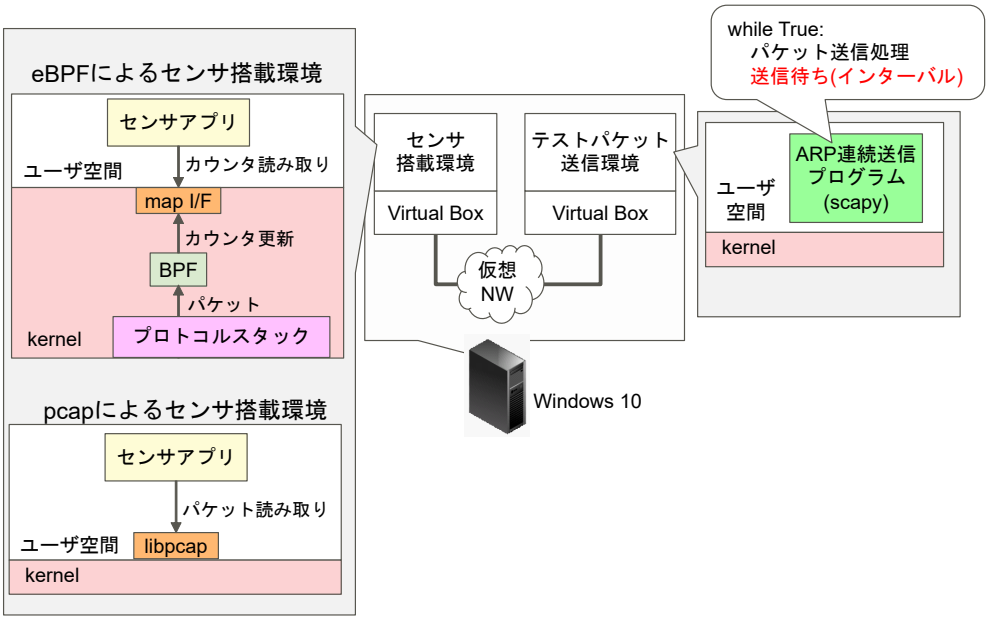

図 9　評価環境
Fig. 9　Evaluation environment

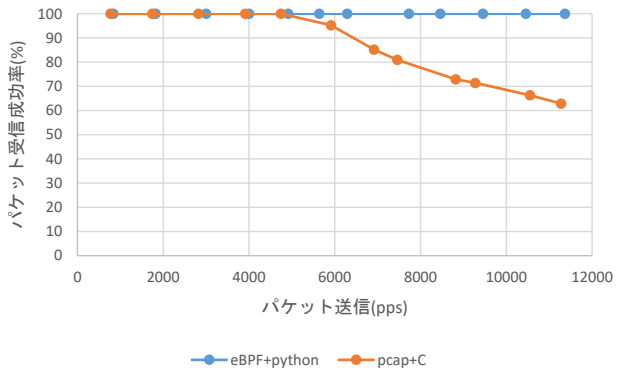

図 10　パケットキャプチャ成功率
Fig. 10　packet capture rate.

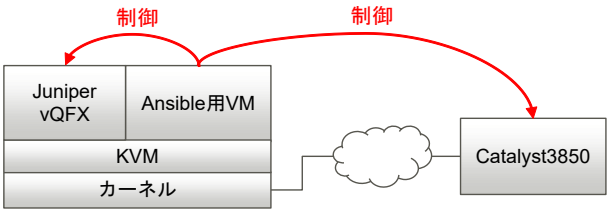

図 11　LAN 機器制御の評価環境
Fig. 11　System for evaluation of control performance.

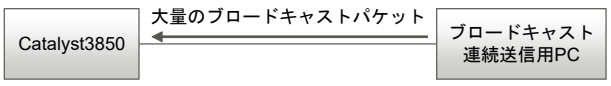

図 12　LAN 機器のループ切断評価環境
Fig. 12　Evaluation environment of ethernet device.

相互接続するような運用において，広域網の管理者がループ対応機能を下流組織に提供することで，サービスレベルの向上を図ることが可能となる．

以上の結果から，制御時間が長くても，設定ファイルを置き換えで様々なネットワークを制御可能な Ansible を利用するループ対策は有用性が高いと考えている．

5.　まとめ

近年利用が広がっている広域網をまたいで構成する仮想ネットワークで MAC 層のループが発生した場合，仮想ネットワークを構成する組織間の運用体制の問題や，仮想ネットワークを構成する要素 (LAN 機器や計算機の仮想ネットワーク機能) がループを起こした仮想ネットワークだけを対象に対策を取る機能を提供していないなど，古くて新しい問題に対応するため，本研究では計算機ベースで様々な仮想ネットワークの技術，新たなベンダの LAN 機器や計算機の仮想ネットワーク設定を制御する仕組みを実現するために，eBPF を用いたパケットの検査を行うセンサと，LAN 機器や計算機の設定変更に用いるオープンソースのミドルウェアである Ansible を組み合わせたシステムを開発した．

本研究での試作を対象とした評価の結果から，eBPF を用いることで，ループを検出するためのセンサの性能が従来用いられてきたパケットキャプチャ技術より有利であること，ループを検出した場合に LAN 機器 (物理と仮想の両方) 数秒から十数秒で可能であることを確認することができた．

実際のネットワーク環境に適用して一定期間運用することと，実環境での性能評価は今後の課題である．

参考文献

[1]　IEEE: IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges, *IEEE*

Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998), pp. 1–281 (online), DOI: 10.1109/IEEESTD.2004.94569 (2004).

[2] IEEE: IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks, *IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014)*, pp. 1–1993 (online), DOI: 10.1109/IEEESTD.2018.8403927 (2018).

[3] IEEE: IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Amendment 4: Provider Bridges, *IEEE Std 802.1ad-2005 (Amendment to IEEE Std 802.1Q-2005)*, pp. 1–74 (online), DOI: 10.1109/IEEESTD.2006.6044678 (2006).

[4] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M. and Wright, C.: Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, RFC 7348 (2014).

[5] Gregg, B.: Performance Superpowers with Enhanced BPF, *USENIX*, Santa Clara, CA, USENIX Association (2017).

[6] RedHat: RedHat Ansible. `https://www.ansible.com/` (参照 2020 年 01 月 14 日参照).

[7] 武藤亮一，杉谷樹一：ループフレーム検知装置およびループフレーム検知方法 (2006). japanese.

[8] Tzeng, S.: LOOP DETECTION FOR A NETWORK DEVICE (2006). United States.

[9] Tcpdump/Libpcap: TCPDUMP and LiBPCAP. `https://www.tcpdump.org/` (参照 2020 年 01 月 14 日参照).

[10] IO-Visor: XDP eXpress Data Path. `https://www.dpdk.org/` (参照 2020 年 01 月 14 日).

[11] Høiland-Jørgensen, T., Brouer, J. D., Borkmann, D., Fastabend, J., Herbert, T., Ahern, D. and Miller, D.: The EXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel, *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies*, Vol. CoNEXT, No. 18, New York, NY, USA, Association for Computing Machinery, pp. 54–66 (online), DOI: 10.1145/3281411.3281443 (2018).

[12] Choudhury, D. G.: XDP-Programmable Data Path in the Linux Kernel., *; login:*, Vol. 43, No. 1 (2018).

[13] DPDK: Data Plane Development Kit. `https://www.iovisor.org/technology/xdp` (参照 2020 年 01 月 14 日).