広域仮想環境向けMAC層ループ対策手法の評価

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

概要：クラウド環境の普及に伴い，複数組織で仮想化されたMAC層のネットワークを相互接続するケースが増加しており，ループが起きたネットワークを検出した上で，対策を行う必要がある．
一方，運用ポリシーの違い等の理由から，ネットワークのループ対応にSpanning Tree Protocolが利用できない場合が多く，ブロードキャストパケットの監視や検査パケットを送信する等の方法で異常な回線の有無を確認する必要があるが，仮想ネットワーク方式の増加にネットワークループ対策の機能追加が追いついていない．
これに対応するため，仮想計算機ネットワークや，インテリジェントスイッチで構成されたネットワークに接続し，検査パケットによりブロードキャストパケットのループを起こし，ストームの検出と仮想ネットワークの制御を行う仕組みを研究してきた．
現在，実際のユーザが利用している環境での実証実験を計画しており，広域ネットワーク上に故意にネットワークループを作成し，提案方式を適用する計画を進めている．本提案方式では，故意にブロードキャストストームを起こすため，ネットワークループの検査パケットのストームによりネットワークにかかる負荷，ループ検出に必要な所要時間について評価を行った．

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専用プロトコルの通信 (STP : Spanning Tree Protocol) を行う方法 [1] の他に，ループによって発生する現象を監視し，検出したネットワークの物理ポートを切断する方法が多くのLAN機器に採用されている．また，一部ベンダの機器では，マルチキャストやブロードキャストパケットを廃棄する方法が採用されている．

現在，仮想化されたMAC層を組織間で接続することも珍しくなく，古くは802.1Q(TAG VLAN)[2] に始まり，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．また，このネットワークを構成する装置としてはLAN機器だけでなく，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 (図1) 多く用いられている．

このような組織間接続において，ネットワーク運用ポリシーの違いなどからSTPが利用できない場合も多く，ループ検出と自動切断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，計算機内の仮想化ネットワークで利用可能なループ検出の仕組みは存在せず，ユーザ自らが開発する必要がある．また，LAN機器におけるループ対策も，切断対象となる仮想ネットワーク技術の増加に追いついていない．それため，多くのLAN機器で利用可能なスイッチの物理ポートを切断した場合，複数組織のVLAN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2, 3)，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提案 [7] してきた．ただし，以前の報告 [7] では実験室内の小規模な環境でループ検出機能のパケット処理性能や，商用のLAN機器を制御するために必要な所要時間の測定しか行ってい

[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

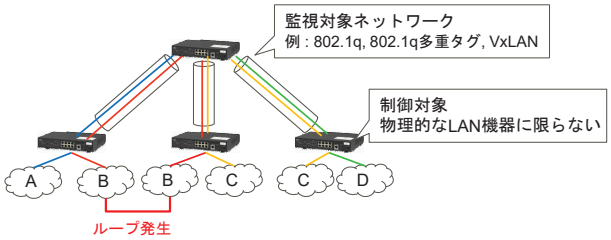

図 1　想定するネットワーク環境

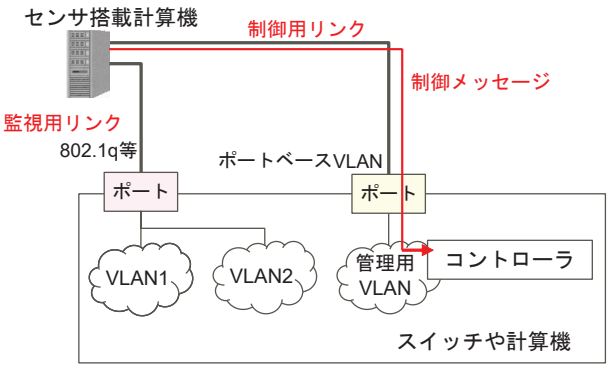

図 2　LAN 機器を制御する場合

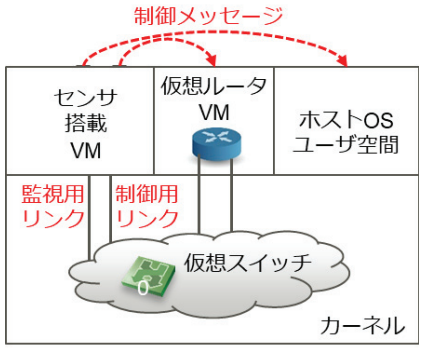

図 3　計算機内の仮想ネットワークを制御する場合

ない．

実際のユーザに使ってもらうためには，実際のユーザ環境での有効性，使い勝手等の評価が必要であることから，多数のユーザが利用している広域 LAN サービスに故意にループを作成した実証実験を予定している．しかし，本提案方式ではループ検出用の検査パケットがブロードキャストストームを起こすため，他のユーザやネットワーク事業者に対する影響を事前に評価する必要があり，数値シミュレーションを行った．

2. 従来手法

2.1 ブロードキャストストーム

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 (ブロードキャストパケットを

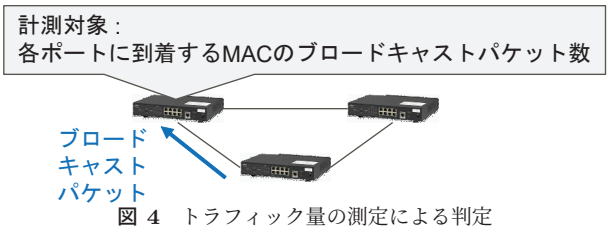

図 4　トラフィック量の測定による判定

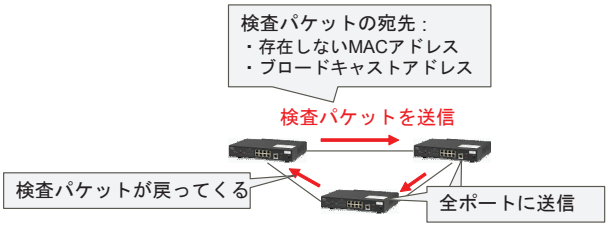

図 5　受信パケットのソースアドレスを確認する方法

最初に受信した) リンク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．

2.2 ループの判定

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．

もう一種類の手法は，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法 (武藤 [8],Tzeng[9]) がある．武藤 [8] の方法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．Tzeng[9] の方法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．

2.3 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．

従来の LAN 機器 (イーサネットスイッチ等) ではループ検出時に，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，ほとんどのベンダの機器は物理ポートの切断しか対応しておらず，Linux 等の仮想ネットワークではループの切断等の機能は実装されていない．そのため，現状で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．

2.4 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)[10]」，「Data Plane Development Kit(DPDK)[11]」，「eBPF」の 3 種類がある．

pcap はネットワークインターフェイスを通過するパケットをカーネルの補助なしにユーザ空間のプログラムで処理するため，トラフィック量が多いと，取りこぼしが多くなる．

DPDK はデバイスドライバに特別なインターフェースを実装し，そのインターフェースを利用することで，取りこぼす確率を非常に小さくすることができるものの，利用可能なハードウェアが制限されるため，本研究では利用しない．

上の 2 つに対して，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 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．カーネル内のパケット処理は，VM 内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ため，本研究ではパケットキャプチャに eBPF を採用した．

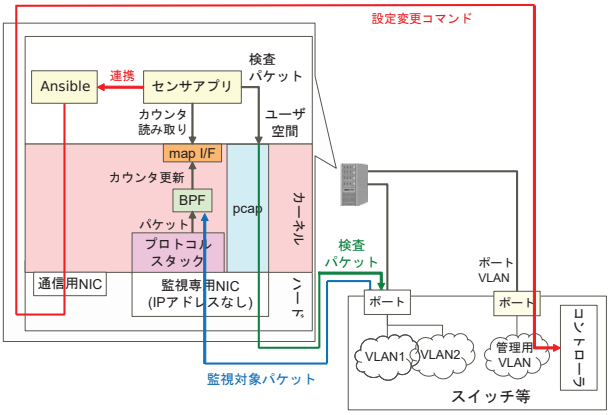

図 6 提案アーキテクチャ

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制御可能な 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[12][13][14]) 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，自分が投げたパケットを検出するためのイン

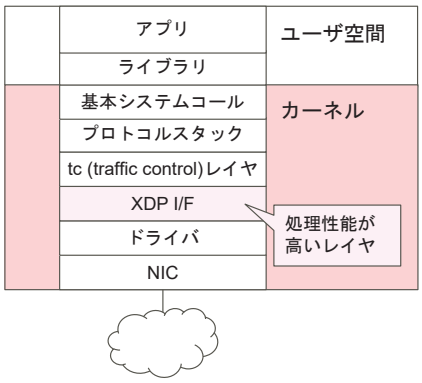

図 7 プロトコルスタックと eBPF の関係

ターフェイスに XDP を用いている (図 7).

このセンサでは，Tzeng[9] と類似の方法を用いるが，Tzeng [9] の専用プロトコルのパケットを送信するのではなく，そのネットワーク上に存在しないアドレス宛の ARP を周期的に送信する．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).

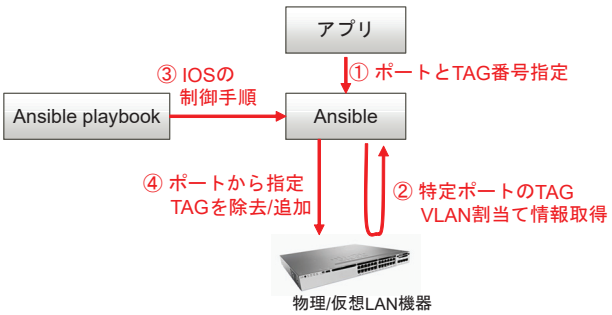

図 8 Ansible による TAG VLAN の切断

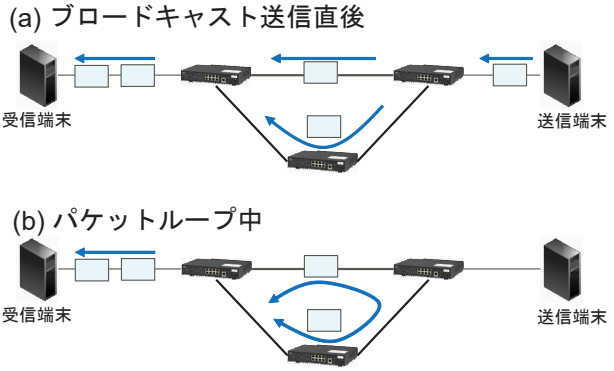

図 9 ループ発生時のパケットの配送

4. 評価

本方式の提案 [7] では，実験室内の小規模な環境でブロードキャストストーム検出機能のパケット処理性能や，商用の LAN 機器を制御するために必要な所要時間の測定しか行っていない．一方，実際のユーザに使ってもらうためには，実際のユーザ環境での有効性，使い勝手等の評価が必要である．

ただし，本方式はブロードキャストストームを故意に起こすため，ブロードキャストストームでネットワークにかかる負荷 (帯域の消費やパケットの量) がどの程度であるかを示すことが，ネットワーク管理者に採用してもらうために必要なことである．

実際に，実証実験を計画しているネットワーク管理者に実験の計画を説明したところ，この点について心配されたことから，想定するネットワークでどの程度の影響が出るかを理論的に評価した．

4.1 評価モデル

図 9 はループの有無を確認するための検査パケットが実際にループしたネットワーク上で流れる様子を示している．実際にループを形成しているネットワークに検査パケットが送信されると，パケットは 2 分岐し，時計回り・反時計回りの 2 つのループが発生するため，ループ 1 周ごとに 2 個のパケットが受信される．

本評価における各種パラメータを表 1 に示す．なお，本

表 1 評価パラメータ

内容	式
パケットがループを一周する所要時間 (s)	T
ループ検査パケットのサイズ (bit)	S
ループ検査センサのパケット損失率	x
ループ検査センサのパケット観測時間 (s)	t

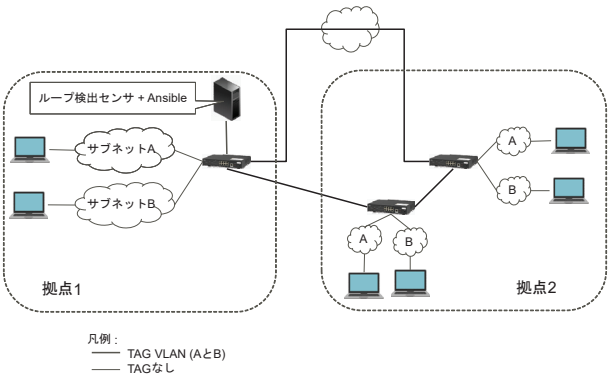

図 10 複数キャンパス接続

方式の試作システムでは S は 100 バイト (=800bit) となっている．表 1 のパラメータから，1 秒間に検査パケットがネットワークをループする回数は $1/T$，センサが 1 秒間に検査パケットを受信する数 $P(T)$ は上り下りの 2 つを受信するため $2/T$ となる．すると，ループしている検査パケットが消費する帯域 $B(T,S)$ と t 秒間センサがループしている検査パケットの受信に失敗し続ける確率 $f(T,x,t)$ は以下の式となる．

$$B(T,S) = S/T \tag{1}$$

$$f(T,x,t) = x^{(2t/T)} \tag{2}$$

実際のユーザ環境で利用する場合，$f(T,x,t)$ が十分小さくなるのに必要な所要時間 x の時間と LAN 機器の制御に必要な時間の和が検査パケットがループし続ける期間となり，その期間上り下り両方の回線を $B(T,S)$(bps) 検査パケットが消費し続ける．そのため，この 2 つの値が想定環境にとって十分小さい値となれば良い．

4.2 想定環境

図 10 と図 11 は想定している環境である．図 10 は異なる拠点間を通信事業者の広域 LAN サービスを結ぶ環境を模擬するものであり，国内で閉じている環境 (WAN 回線の RTT が数ミリ秒以上) を想定している．それに対して，図 11 はデータセンサが国際回線経由で接続するような場合も含めた環境となっている．これらの環境では経路の RTT が異なるため，パケットがループを一周する所要時間 T を 5ms, 100ms, 500ms(=0.5s) の 3 種類を用いる．

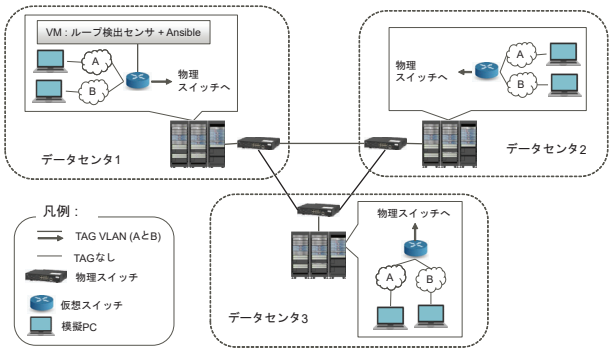

図 11 複数データセンタ接続

表 2 評価データ

T の値 (s)	$P(T)$ (pps)	$B(T,S)$ (Kbps)	$f(T,x,t)$
0.005 (=5ms)	400	160.0	x^{400t}
0.1 (=100ms)	20	8.0	x^{20t}
0.5(=500ms)	4	1.6	x^{4t}

4.3 ループ検出センサがループを見逃す確率

表 2 に評価に必要なデータをまとめている．また，ループ検出センサの性能に問題があり，パケット損失率が非常に高い ($x = 0.5$) と想定して，1 秒間センサが検査パケットを全く検出できない確率は，T が 100ms の場合 9.5×10^{-7} であり，T が 5ms の場合はさらに小さい値となり，発生する確率は非常に低い．これに対して，T が 0.5 秒の場合，6.25% と発生する確率は高くなる．ただし，観測期間 t を 5 秒まで延ばすと，10^{-5}% 未満となり，十分小さくすることができる．

4.4 検査パケットのループによる影響

先の表 2 から，最大でも 160.0Kbps,400pps しかネットワークに負担がかからないため，ほとんどの環境では問題にならない．もし，回線容量が逼迫していて検査パケットのような非常にサイズの小さいパケットを多数送信されると影響が及ぶような環境であっても，想定している範囲 (T が最大で 1 秒) の環境では，最大でも 10 秒もしくはそれ以下の時間でセンサはループを捉えることができる．

一方，以前報告した結果 [7] では，LAN 機器の制御が終了するまでに最大 16 秒程度必要であったことから，ループの検出に 5 秒必要となる場合でも 30 秒未満でループが消滅する．

ループしている回線を使っている端末を想定すると，検査パケットのループが回線を圧迫していても，30 秒未満で解消するのであれば，広域網を介して使われる TCP は利用可能な帯域が減ることはあっても，コネクションタイムアウトが発生するような重大な影響は受けない．

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に対応す

るための手法を実際の環境に適用した場合の検査パケットによるネットワークへの影響をモデルによって評価した．

日本国内に閉じた環境はもちろん，国際回線で複数のデータセンタをつないだ場合でも，ループを検査するパケットによる帯域の消費やパケットの発生量は非常に小さく，衛星回線のような遅延の大きい環境であっても，多くのユーザが利用する TCP への影響はほとんどないと考えられる．

今回の評価で用いたモデルが想定したものと同等のネットワークで本システムを動作させ，モデルによる理論的な計算と同じ結果が得られるかどうかを確認することが今後の課題である．

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 07 月 22 日参照).

[7] 野呂正明，高野陽介，小口直樹，阿部　俊：eBPF による MAC 層ループ対策，情報処理学会研究報告. マルチメディア通信と分散処理研究会報告， Vol. 2020, No. 62, pp. 1–7 (2020).

[8] 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, 特開 2006-33275 (2006).

[9] Tzeng, S.: LOOP DETECTION FOR A NETWORK DEVICE, US Patent 0285.499A1 (2006).

[10] Tcpdump/Libpcap: TCPDUMP and LiBPCAP. `https://www.tcpdump.org/` (参照 2020 年 07 月 22 日参照).

[11] IO-Visor: XDP eXpress Data Path. `https://www.dpdk.org/` (参照 2020 年 07 月 22 日).

[12] 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).

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

[14] DPDK: Data Plane Development Kit. `https://www.iovisor.org/technology/xdp` (参照 2020 年 07 月 22 日).