成蹊大学理工学研究報告
J. Fac. Sci.Tech., Seikei Univ.
Vol.56 No.1 (2019)pp.29-34
(研究速報)

SDN ネットワークを前提とした動的トラヒックシェーピングの提案

影山 瞭翔[*1]，栗林 伸一[*2]

Dynamic Traffic Shaping Method for SDN-based Networks

Ryoka KAGEYAMA[*1], and Shin-ichi KURIBAYASHI[*2]

(Received March 19, 2019)

1．はじめに

トラヒックシェーピング（以後、シェーピング）とは、通信量を規定速度に抑える帯域制御の方式の一つで、規定速度を超えるデータを通信機器内部に保存し、容量に空きができたときに送信する方式である。バースト的に到着したパケットを規定速度に平滑化して転送する方式である。従来は想定した場所にシェーピング装置を事前設置し、アプリケーション種別毎またはトラヒック種別ごとに制御ポリシーに従って実施されることが多い[1]-[4]。また、ネットワーク毎で、かつ混雑している部分のみを考慮して実施されることが多く、必要時に任意の位置で動的にシェーピングを実施することは困難な状況であった。シェーピングが必要な時に、通信フローを選定しかつ最適な場所で動的にシェーピングを実施することができれば、ネットワークの回線帯域や中継処理能力などの資源をより効率的に使用できる可能性がある。

本論文では、ソフトウェア定義型ネットワークSDN (Software Defined Network)[5]およびネットワーク機能仮想化技術NFV(Network Functions Virtualization)[6]-[9]ベースネットワークを前提に、状況に応じて最適なシェーピング対象通信フローとそのシェーピング実施位置を動的に選定する**動的シェーピング方式**を提案する。また、動的シェーピング方式を自動化するためのシステム構成と機能を明らかにする。なお、シェーピングアルゴリズムはリーキーバケツ方式やトークンバケツ方式[10]などを想定する。

本論文は文献[11]-[13]を発展させ、とりまとめたものである。

*1：情報科学科4年生（現在：大崎電気工業）
*2：情報科学科教授(kuribayashi@st.seikei.ac.jp)

2．提案する動的シェーピング方式[11]-[13]

2．1　動的シェーピング方式の概要

例えば、図1のように受信側回線が混雑した場合、従来(図1<1>)は受信側でシェーピングを実施することが多い。これを図1<2>のように、シェーピングを送信側近くで実施できれば、無駄な回線帯域および中継処理能力を削減し、ネットワークコストを削減できる可能性がある。シェーピングを実施しない場合、バースト的に発生するトラヒックに対応できるように回線帯域と中継処理能力を準備しておく必要があるためである。

提案する動的シェーピング方式は、SDNおよびNFVベースのネットワークを前提とし、従来のようにシェーピング機能を特定の場所に事前配備せず、ネットワーク回線混雑を動的に検出し、その混雑を解消するために最適なシェーピング対象通信フローと最適なシェーピング実施位置を動的に選定し、仮想シェーピング機能を生成してそこを経由させる。従来のシェーピングは、回線毎だけでなく、アプリケーション種別やトラヒック種別毎に実施されることも多い。そのため、例えばアプリケーション種別ごとにシェーピングを実施する場合、シェーピ

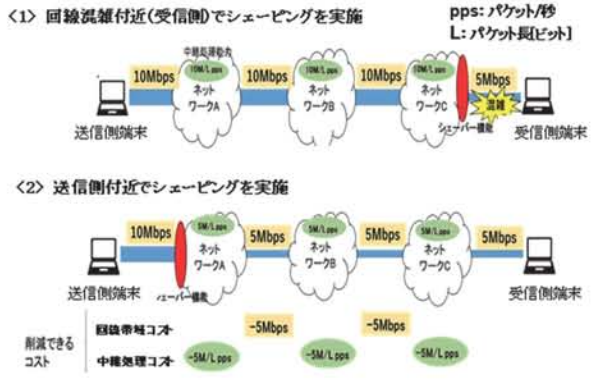

図1．シェーピング実施位置によるネットワークコスト削減効果の比較例

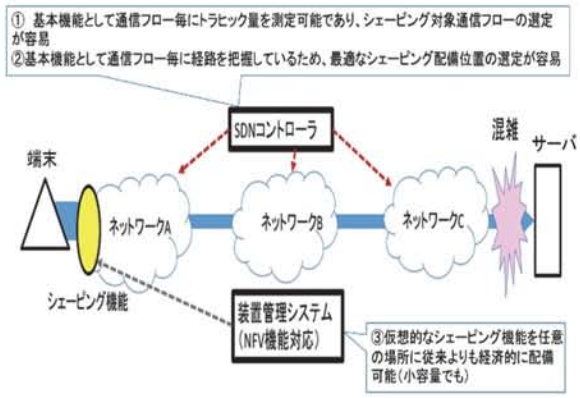

図２．動的シェーピング方式の実現を容易とするSDNとNFVの特徴

ング対象となったアプリケーション種別に属する通信フローの中から最も多く無駄な回線帯域を削減できる通信フローとそのシェーピング実施位置を選定する。これ以降は回線毎を前提に説明するが、同じようにアプリケーション種別毎、トラヒック種別毎、にも適用できる。

従来のネットワークで同様のことをやる場合、通信フロー毎のトラヒック量を測定するために専用のシステムを別途導入しなければならないためコストが高くなってしまう。また、シェーピングを最適な位置で実施するためには、通信フローが経由する可能性がある全ての位置にシェーピング機能を配備する必要があり、膨大なコストがかかる。図２に示すように、SDNとNFVの特徴を使用することで実現が容易になる。具体的には、SDNの特徴を利用することにより、基本機能として通信フロー毎のトラヒック量を測定可能となる。また、SDNコントローラが通信フローの経路を把握しているため、通信フローの経路を絞り込んで適切な位置でシェーピングを実施可能となる。さらに、NFVの特徴を利用することにより、任意の容量のシェーピング機能を任意の場所に従来よりも経済的に配備することが可能となる。

2．2　動的シェーピング方式実現上の課題と解決策

(1)　シェーピング実施契機

各回線の平均使用率がPmax（例えば、0.7）を超えた時点で‘回線混雑’と判定し、シェーピングを実施することを想定する。2.1 節で述べたように、アプリケーション種別毎、トラヒック種別毎、に実施する場合も同様である。

(2)　シェーピング実施位置

混雑位置でシェーピングを行うよりも、送信元近くでシェーピングを行うことでより多くの回線帯域コストを削減することができる可能性がある。そのため、必要時に動的に仮想シェーピング機能を送信元近くに作成し、SDNコントローラがそこを経由するように経路変更する。

(3)　シェーピング対象通信フローの選定

混雑回線を経由する全ての通信フローを対象にシェーピングを実施することは効率的ではなく、以下のようにシェーピング対象通信フローを選定する。

<ステップ 1>　混雑しシェーピングが必要な回線を経由するすべての通信フローの中から、最も通信速度の速い通信フローを最大N本（例えば 10）選択し、それらをシェーピング対象通信フローの候補とする。

<ステップ 2>　上記(2)で提案したシェーピング実施位置を前提とすると、混雑位置でシェーピングを実施する場

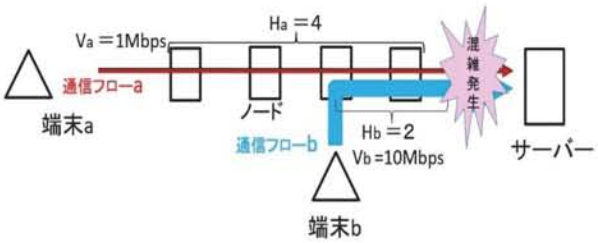

図３．シェーピング対象通信フローの選定比較

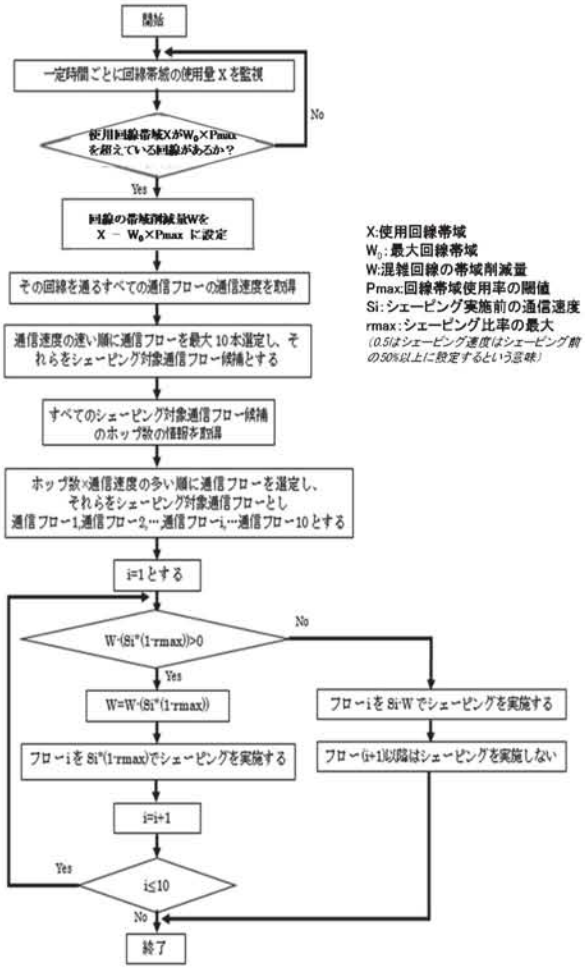

図４．回線混雑判定からシェーピング対象通信フローとシェーピング速度決定までの処理フロー

合に比べ、「送信元と混雑位置までの総回線距離」×通信速度　だけ回線帯域を削減することができる。しかし、総回線距離を簡単に求めることは困難であり、総回線距離をホップ数で置き換え、「**ホップ数×通信速度**」の値が最も大きい通信フロー候補をシェーピング対象通信フローに選定する。図 3 の例では、ホップ数(Hb)×通信速度(Vb)が大きい通信フローbをシェーピング対象通信フローとして選定する。

(4)　シェーピング速度の決定

上記(3)で選定したシェーピング対象通信フローのシェーピング速度は回線の混雑が解消するように設定する。但し、サービス品質を考慮し、通信フロー毎のシェーピング速度はもともとの通信速度の半分以下には設定しないものとする。もし、通信速度の半分にシェーピング速度を設定しても回線混雑が解消できない場合は、上記(3)で算出した削減量の大きい順に回線混雑が解消できるまでシェーピング対象通信フローを追加する。

(5)　処理フロー

上記(1)から(4)までの処理の流れをまとめたものを図4に示す。

2. 3　動的シェーピング方式を実現するシステム構成と機能

動的シェーピングを自動的に実施するため、既存のSDNコントローラと装置管理システム(NFV機能対応)に加え、**管理オーケストレーション**を新たに追加する。なお、極力既存システムに変更を加えないことを前提とする。

管理オーケストレーション、SDNコントローラ、装置管理システム（NFV機能対応）を含む動的シェーピングを自動化するためのシステム構成を図 5 に示す。端末a、b、cがサーバーとそれぞれ個別に通信を行い、それら通信フローをそれぞれ通信フローa、通信フローb、通信フローcと呼ぶ。OpenFlowスイッチからサーバー向け回線が混雑しているケースである。動的シェーピングの制御シナリオは管理オーケストレーションが管理し、実施する。具体的には、全体の状況を把握し、必要な処理を行う。‘回線混雑’と判定したら、管理オーケストレーションはSDNコントローラに対し通信フロー毎の通信速度とホップ数の収集を指示する。SDNコントローラはOpenFlowスイッチから通信フロー毎の通信データ量を収集する。ただし、収集する通信フロー数が多いとOpenFlowスイッチならびにSDNコントローラの大幅な性能劣化につながる。そのため、OpenFlowスイッチは通信フロー毎のトラヒックカウンタ値が一定値以上の通信フローを事前にSDNコントローラに通知し、SDNコントローラはその通知された通信フローの通信データ量だけをOpenFlowスイッチに問い合わせることとする。

また、SDNコントローラはネットワークの接続構成を事前に把握しているため、対象となる通信フローのホップ数を管理オーケストレーションに転送する。管理オーケストレーションは、収集したデータをもとに、装置管理システムに対してシェーピング制御パラメータの設定、シェーピング機能の位置情報提供などを指示する。SDNコントローラに対しては、シェーピング対象通信フローが指定されたシェーピング機能を経由するように経路切り替えを指示する。上記で説明した管理オーケストレーションが具備すべき主な機能を表 1 に整理する。

シェーピング機能の決定から経路切り替えまでを対象に、管理オーケストレーション、SDNコントローラ、装置管理システム間で必要となる連携処理の概要を図 6 に示す。破線矢印で示す数字は処理フローの順番、ギリシャ文字はやり取りするパラメータをそれぞれ示す。以下、図 6 に示す処理を順番に説明する。

・処理 1 ：管理オーケストレーションが装置管理システムに対して、シェーピングを実施するネットワーク識

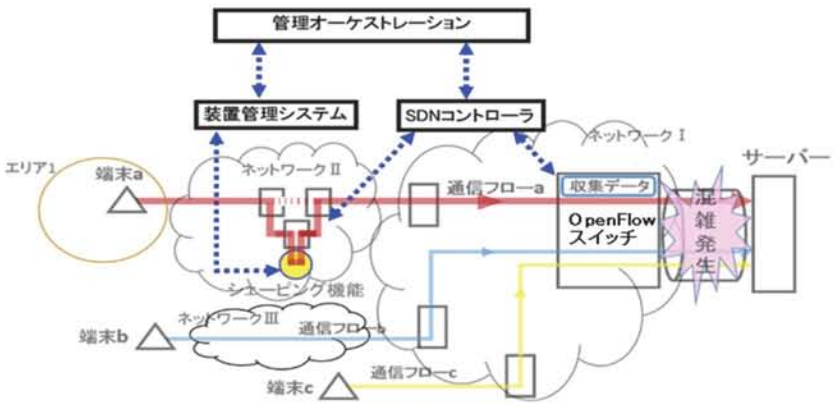

図５．動的シェーピングを自動化するためのシステム構成

表１．管理オーケストレーションが具備すべき主な機能

⓪管理オーケストレーションが処理をする流れをまとめ、他機能を制御する機能 各機能ごとに進行状況を確認し処理する順番を管理する。
①回線の使用状況の収集機能
②通信フロー毎の通信速度の収集機能
③通信フロー毎のホップ数の収集機能
④シェーピング対象通信フローの選定機能 ホップ数と通信速度の積を計算して一番大きいものから順に選択
⑤シェーピング対象通信フローのシェーピング速度決定機能 通信フロー毎の通信速度からシェーピング速度を決定する
⑥シェーピング機能の決定機能
⑦シェーピング機能へのパラメータ設定機能 シェーピング対象通信フローのパラメータを設定する
⑧シェーピング実施時にシェーピング機能を経由するように経路切り替え機能 シェーピング解除時にシェーピング機能を経由しないように経路切り替え機能
⑨SDNコントローラとの連携機能
⑩装置管理システムとの連携機能

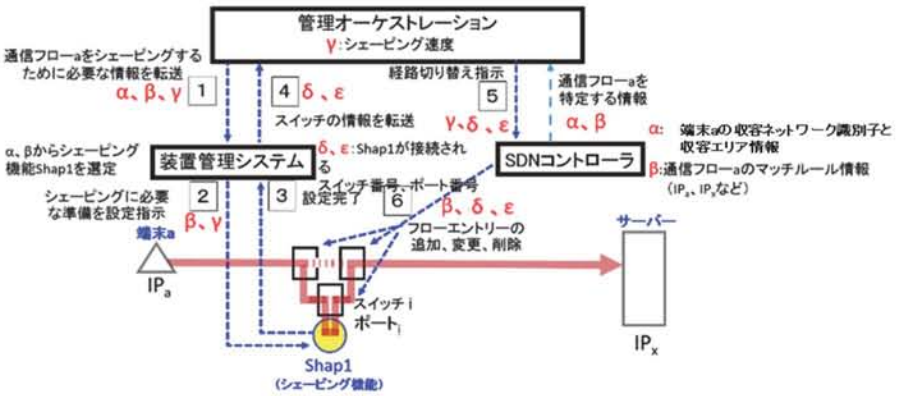

図６．シェーピング機能の選定から経路切り替えまでの連携法

別子と端末の収容エリア情報（α）、シェーピング対象通信フローaのマッチルール情報（β）、シェーピング速度（γ）を指定し、シェーピング機能の選定とシェーピング制御パラメータ設定を指示する。

・処理 2：装置管理システムはαとβ、ならびにシェーピング機能の使用状況などをもとに最適なシェーピング機能を選定する（この例では、Shap1）。

・処理 3：シェーピング機能はシェーピング制御パラメータ設定後、設定完了を装置管理システムに通知する。

・処理 4：装置管理システムは、選定したシェーピング機能が接続されているスイッチの番号(δ)とスイッチのポート番号（ε）を管理オーケストレーションに送付する。

・処理 5：管理オーケストレーションは、SDNコントローラに対してγ、δ、εを通知し、選定したシェーピング機能を経由するように通信フローaの経路切り替えを指示する。

・処理 6： SDNコントローラはδ、εをベースに対応するOpenFlowスイッチのフローエントリーの設定、書き換えを行い、通信フローaがシェーピング機能を経由するように経路を切り替える。なお、シェーピングが不要になった場合には逆の操作指示を行う。

3．　動的シェーピング方式の動作確認

3．1　評価システムの設計・構築

今回は、管理オーケストレーション機能の確認を主目的とし、SDNコントローラ、OpenFlowスイッチ、装置管理システムはそのものではなく代替え装置（ソフトウェアルータVyOS[14]など）で実現する。また、2 章で提案した管理オーケストレーションの主な機能を実現する「評価ツール（C#ベースのソフトウェア）」を設計・作成した。評価ツールは、新たに設計したツールと既存のツール（plinkなど）で構成されている。動作確認のために使用した評価システムの概要を図 7 に示す。VyOSノード数 4 台、端末数 3 台、制御用端末 1 台、サーバー 1 台から構成される。端末a、b、cはそれぞれサーバーと個別に通信を行い、ノード 4 からサーバー方向の回線が混雑する状況を想定する。シェーピング機能はVyOSが具備する機能[14]を流用する。

3．2　動作確認結果

動作確認として、今回はノード 4 とサーバーの間の最大帯域を 20Mbps、Pmaxを 0.6 とした動作確認を行った。通信フローa、b、 c のホップ数×通信速度の値はそれぞれ 39Mbps、6Mbps、1Mbpsとなり、一番値の大きい通信フローaがシェーピング対象通信フローとして選定される。混雑を解消するためにはトータル 17Mbpsから 6Mbps下げて 11Mbpsにする必要があり、通信フローaのシェーピング速度は 7Mbpsと設定される。図 8 はノード 4 からサーバー方向へのトラヒック量変化を実測したものであり、想定通り通信フローaのトラヒック量が 7Mbpsに変化していることが確認できる。

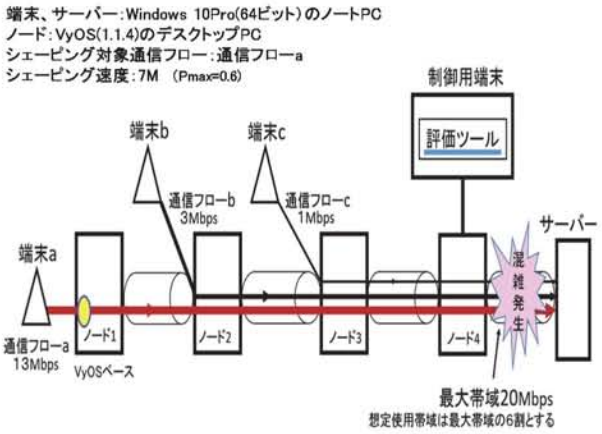

図 7．評価システム構成（一例）

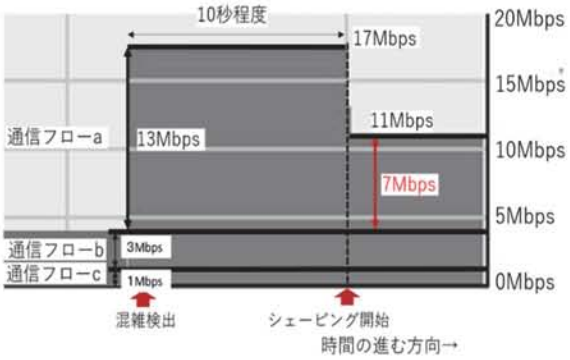

図 8.　ノード 4 とサーバー間のトラヒック測定

4．まとめ

SDNおよびNFVベースのネットワークを前提に、状況に応じて最適なシェーピング対象通信フローとそのシェーピング実施位置を動的に選定する動的シェーピング方式を提案した。また、動的シェーピング方式を自動化するためのシステム構成を明らかにし、評価システムを構築して提案方式が想定通り動作することを確認した。

提案方式は対象となる回線や通信フローの数が多くなると処理量も大幅に増大する可能性があり、その削減法が今後の課題である。また、複数のネットワークにまたがった場合の複数SDNコントローラ間の連携法、複数の装置管理システム間連携法も今後検討する必要がある。

参考文献

[1] M.M.Marco et al.,"The local and global effects of traffic shaping in the internet," 2011 Third International Conference on Communication Systems and Networks (COMSNETS), pp.1-10, Jan. 2011

[2] X.Liu and A.Men, "QoE-aware Traffic Shaping for HTTP Adaptive Steaming," International Journal of Multimedia and Ubiquitous Engineering Vol.9,No.2,pp.33-44,2014.

[3] R.Houdaille and S.Gouache, "Shaping HTTP adaptive streams for a better user experience," 3rd Multimedia Systems Conference, pp. 1–9. ACM (2012).

[3] A.Elwalid and D.Mitra, "Traffic shaping at a network node: theory, optimum design, admission control," Proceedins of INFOCOM'97, April 1997.

[4] Blue Coat PacketShaper
http://www.edgeblue.com/

[5] "OpenFlow Switch Specification Ver.1.5.1", ONF (March 2015)

[6] M. Chiosi et al., "Network Functions Virtualization - An Introduction, Benefits, Enablers, Challenges and Call for Action," ETSI NFV, Oct. 2012.

[7] "Network Functions Virtualisation (NFV); Architectual Framework," ETSI GS NFV 002 v1.2.1, Dec. 2014.
file:///C:/Users/Kuribayashi/Downloads/gs_NFV002v010201p.pdf

[8] R.Mijumbi, J.Serrat, J.Gorricho, N.Bouten, F.D.Trurck and R.Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges," IEEE Communications Surveys & Tutorials, Vol. 18, Issue 1, pp.236-262, 2016.

[9] M.Bouet, J.Leguay, and V.Conan, "Cost-based placement of virtualized Deep Packet Inspection functions in SDN," 2013 IEEE Military Communications Conference, pp.992-997, Nov. 2013.

[10] 銭飛:"NS2 によるネットワークシミュレーション"、森北出版.

[11] 寺島:"資源効率を高める動的シェーピング実施法の提案", 2017 年度　成蹊大学理工学部　卒業論文.

[12] 影山:"動的シェーピング方式を実現するための管理オーケストレーション機能の検討", 2018 年度　成蹊大学理工学部　卒業論文

[13] 影山、栗林:"資源効率を高める動的シェーピング方式の提案"、2019.3 電子情報通信学会・東京支部学生研究発表会　講演番号 78.

[14] VyOS
https://vyos-users.jp

-以上-