INNOV 2020

The Ninth International Conference on Communications, Computation, Networks and Technologies

ISBN: 978-1-61208-833-4

October 18 -22, 2020

INNOV 2020 Editors

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster (WWU) / DIMF / Leibniz Universität Hannover, Germany

INNOV 2020

Forward

The Ninth International Conference on Communications, Computation, Networks and Technologies (INNOV 2020), held on October 18 - 22, 2020, aimed at addressing recent research results and forecasting challenges on selected topics related to communications, computation, networks and technologies.

Considering the importance of innovative topics in today's technology-driven society, there is a paradigm shift in classical-by-now approaches, such as networking, communications, resource sharing, collaboration and telecommunications. Recent achievements demand rethinking available technologies and considering the emerging ones.

The conference had the following tracks:

☐ Communications
☐ Networking
☐Computing
☐Web Semantic and Data Processing
☐Security, Trust, and Privacy

We take here the opportunity to warmly thank all the members of the INNOV 2020 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to INNOV 2020. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the INNOV 2020 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that INNOV 2020 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the areas of communication, computation, networks and technologies..

INNOV 2020 Publicity Chair

Lorena Parra, Universitat Politecnica de Valencia, Spain

INNOV 2020

Committee

INNOV 2020 Publicity Chair

Lorena Parra, Universitat Politecnica de Valencia, Spain

INNOV 2020 Technical Program Committee

Kishwar Ahmed, University of South Carolina Beaufort, USA
Amjad Ali,University of Swat, Pakistan
Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
YK Chang, National Cheng Kung University, Taiwan
DeJiu Chen, KTH Royal Institute of Technology, Sweden
Yung-Yao Chen, National Taiwan University of Science and Technology (NTUST), Taiwan
Albert M. K. Cheng, University of Houston, USA
Enrique Chirivella-Perez, University of the West of Scotland, UK
Panagiotis Fouliras, University of Macedonia, Thessaloniki, Greece
Marco Furini, University of Modena and Reggio Emilia, Italy
Nikolaos Gorgolis, University of Patras, Greece
Victor Govindaswamy, Concordia University Chicago, USA
Qiang He, Swinburne University of Technology, Australia
Shih-Chang Huang, National Formosa University, Taiwan
Wen-Jyi Hwang, National Taiwan Normal University, Taipei, Taiwan
Sergio Ilarri, University of Zaragoza, Spain
Brigitte Jaumard, Concordia University, Canada
Alexey Kashevnik, SPIIRAS, Russia
Khaled Khankan, Taibah University, Saudi Arabia
Hyunbum Kim, University of North Carolina at Wilmington, USA
Igor Kotenko, SPIIRAS, Russia
Boris Kovalerchuk, Central Washington University, USA
Yiu-Wing Leung, Hong Kong Baptist University, Hong Kong
Chanjuan Liu, Dalian University of Technology, China
Jaime Lloret, Universitat Politècnica de València, Spain
René Meier, Lucerne University of Applied Sciences and Arts, Switzerland
Alfredo Milani, University of Perugia, Italy
Amalia Miliou, Aristotle University of Thessaloniki, Greece
Vincenzo Moscato, University of Naples "Federico II", Italy
Stylianos Mystakidis, School of Natural Sciences | University of Patras, Greece
Sara Nayer, Iowa StateUniversity, USA
Shin-ichi Ohnishi, Hokkai-Gakuen University, Japan
Yash Vardhan Pant, University of California Berkeley, USA
Xingchao Peng, Boston University, USA
Ounsa Roudies, Ecole Mohammadia d'Ingénieurs - Mohammed-V University in Rabat, Morocco

Copyright Information

Table of Contents

SDN-based Dynamic Traffic Shaping Method

Suzuki Syunsuke
Faculty of Science and Technology, Seikei University
Tokyo, Japan
e-mail: us172092@cc.seikei.ac.jp

Kuribayashi Shin-ichi
Faculty of Science and Technology, Seikei University
Tokyo, Japan
e-mail: kuribayashi@st.seikei.ac.jp

***Abstract*- Traffic shaping controls communication traffic flow to prevent a specified communication rate from being exceeded. In conventional networks, the traffic shaping device is implemented at a predetermined location and only a communication flow passing through the device is targeted, which is implemented without network collaboration. This paper proposes a dynamic shaping method using Software Defined Networking (SDN) and Network Functions Virtualization (NFV) paradigms, which selects the optimal communication flows to be shaped and the optimal shaping points dynamically. This paper also presents system configuration and functions for the proposed dynamic shaping, and the method to simplify the process of collecting the traffic data of each communication flow by SDN controller.**

Keywords*- *Traffic shaping; SDN; NFV.

I. INTRODUCTION

Traffic shaping (hereafter "shaping") is a form of bandwidth control. It controls communication traffic flow to prevent a specified communication rate from being exceeded. Any data that exceed the specified rate are stored in the communication device concerned and sent when the link concerned has spare capacity. It smooths a burst packet flow to produce a packet flow that is within the specified rate. The most common conventional way of shaping is to place shaping devices in advance at predetermined points. These devices smooth out traffic according to a control policy specified for each application type or traffic type [1]-[3]. In most cases, traffic shaping is implemented in each network and only at points where there is regularly traffic congestion, and only a communication flow passing through the device is targeted. Therefore, it has been difficult to apply shaping dynamically when and where it is necessary. If traffic can be shaped dynamically on any selected communication flows at optimal points only when necessary, it will be possible to use network bandwidths and packet relay processing capacity more efficiently.

This paper proposes a dynamic shaping method using Software Defined Networking (SDN) [11][12] and Network Functions Virtualization (NFV) [13]-[16] paradigms, which selects the optimal communication flows to be shaped and the optimal shaping points dynamically.

The rest of this paper is organized as follows. Section II explains the overview of the proposed dynamic shaping method using SDN and NFV paradigms. Section III presents a system configuration for automating the proposed dynamic shaping. Section IV confirms the operation of the proposed dynamic shaping with the evaluation system. Section V proposes the method to simplify the process of collecting the traffic data of each communication flow by SDN controller, which is the key function for the proposed shaping method. Finally, Section VI presents the conclusions.

II. PROPOSED DYNAMIC TRAFFIC SHAPING METHOD

A. Issues of conventional traffic shaping methods [4]-[11]

In most cases, traffic shaping is implemented in each network and only at points where there is regularly traffic congestion and the pre-deployed traffic shaper device, and only a communication flow passing through the device is targeted. Therefore, it has been difficult to apply shaping dynamically when and where it is necessary. For example, if a receiving link between network C and the terminal at receiving side in Figure 1 is congested, the shaping has been implemented at the receiving side of Network C as Figure 1<1>. The bandwidth and the packet relay processing capacity of the networks A, B and C related to the communication flow will be wasted.

B. Overview of dynamic traffic shaping method

The proposed dynamic shaping method is applied to SDN- and NFV-based networks. It does not need to place shaping functions at predetermined points in advance, as is the case conventionally. Instead, it detects the link congestion and dynamically selects communication flows to be shaped and the shaping points that are both optimal for resolving the congestion. It also places virtual shaping functions at the selected points, and makes the selected communication flows pass through these points.

In the example of Figure 1, the shaping is performed at the transmission side as in Figure 1 <2>. This can avoid wasteful use of network bandwidth and packet relay processing capacity, and consequently, can reduce the network cost.

Conventionally, it has been common to shape traffic not only for each link, but also for each application type and for each traffic type. If the shaping is to be applied to each application type in the conventional method, for example, the one that results in the greatest reduction in wasteful use of network bandwidth and packet relay processing capacity is selected. This paper discusses cases where traffic is shaped for each link, but the same discussion applies to cases where traffic is shaped for each application type or for each traffic type.

 ISBN: 978-1-61208-833-4

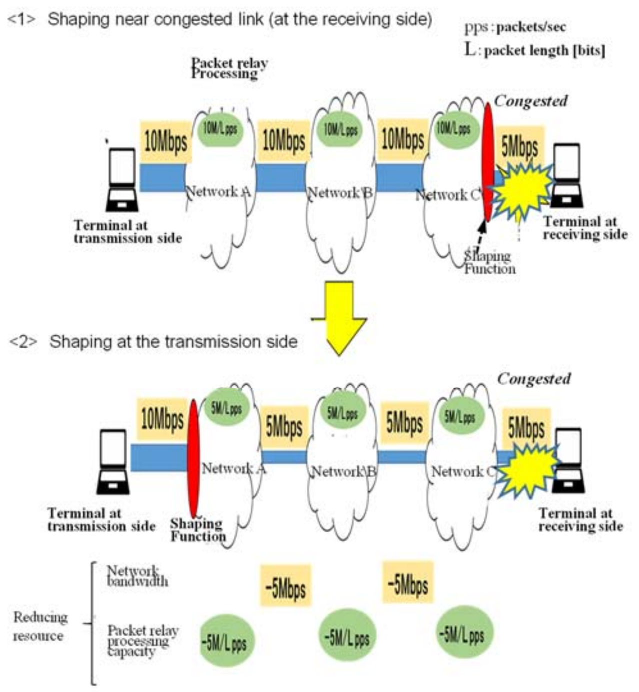

Figure 1. Example of network cost reduction effect by shaping location

If the dynamic traffic shaping is to be implemented in a conventional network, it would be necessary to introduce a system dedicated to measuring the traffic of each communication flow. If the traffic is shaped at optimal points, it would be necessary to place shaping functions economically at all the points where communication flows pass through. However, the implementation of the above-mentioned functions is not economical and can be made easier if we take advantage of the features of the SDN and NFV, as shown in Figure 2. Specifically, SDN features make it possible as a basic function to measure the traffic data of each communication flow. Since the SDN controller keeps track of the route of each communication flow as a basic function, it is also possible to specify the route of each communication flow and perform shaping at an appropriate point. In addition, NFV features make it possible to place a shaping function of any capacity (even with small capacity) at any point more economically than before.

C. Issues in implementing the proposed dynamic shaping method

1) Trigger for shaping

It is assumed in this paper that a link is congested and shaping is executed if the link's average usage rate exceeds P_{max} (e.g., 0.7). This also could apply to cases where shaping is executed for each application type or for each traffic type, as mentioned in Section 2.A.

2) Selection of shaping points

Shaping at a point near the transmission side can reduce more network bandwidth and packet relay processing capacity

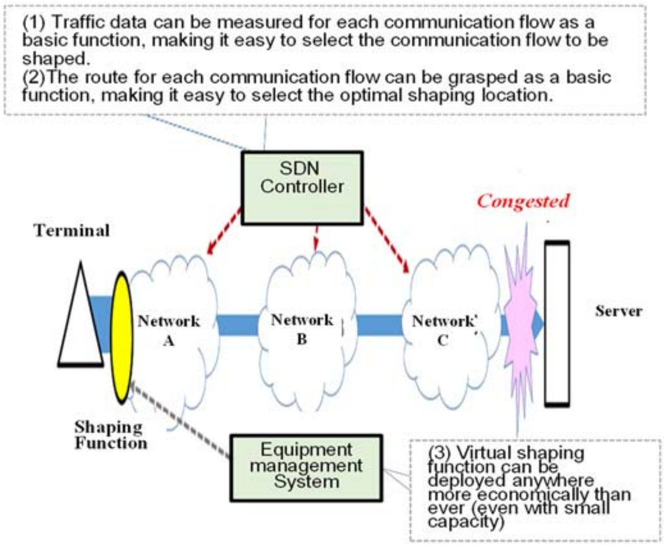

Figure 2. Features of SDN and NFV suitable for the proposed dynamic shaping

than shaping at the congested point. Therefore, it is proposed to create a virtual shaping function with NFV features dynamically at a point near the transmission side when necessary, and the SDN controller changes the route so that the communications flow to be shaped will pass through that point.

3) Selection of the communication flow to be shaped

It is not efficient to shape all the communication flows that pass through the congested link. It is proposed to select the communication flow to be shaped as follows:

<Step 1> Among all the communication flows that pass through a link that is congested and requires traffic shaping, up to N (e.g., 10) fastest communication flows are selected as candidates for shaping.

<Step 2> As stated in 2), shaping at the transmission side can reduce the network bandwidth by L*V, where L is the link length between the transmission side and the congested link, and V is communication speed. As it is not easy to estimate the link length, it is proposed to use the number of hops (H), instead. The term x1 calculated by (1) is the reduced network bandwidth:

$$x1 = \text{communication speed (V)} \times \text{number of hops (H)} \quad (1)$$

Similarly, x2 calculated by (2) is the reduced number of packets to be relayed:

$$x2 = \text{communication speed (V)} \times \text{number of hops (H)} / \text{packet length (P)} \quad (2)$$

Here, the packet length P is considered for the following reason. That is, even at the same communication speed, the shorter the packet length, the larger the number of packets to be relayed.

Finally, the communication flow with the largest x3 value calculated by (3) is selected to be shaped:

$$\mathbf{x3} = x1 * Cb + x2 * Cp \quad (3)$$

where Cb and Cp are cost-coefficients which are used to calculate the cost of two different units at the same level. Flow b in Figure 3 is subject to shaping as it has the largest x3

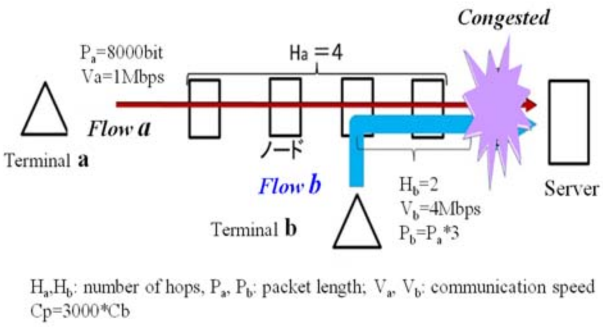

Figure 3. Example of selection of communication flow to be shaped

value.

4) Determination of shaping rate

It is desirable that the shaping rate that will resolve the congestion is selected. However, to guarantee a certain degree of quality of service, it is reasonable to limit shaping rate so that the communication rate after shaping does not go below half of the original communication rate of each flow. If the congestion cannot be resolved by shaping the first flow with the largest x3 value, the flow with the next largest x3 value will be subject for shaping. This is continued until the congestion is resolved.

III. SYSYTEM CONFIGURATION AND FUNCTIONS FOR THE PROPOSED DYNAMIC SHAPING

To execute the proposed dynamic shaping automatically, 'the management orchestration' is introduced in addition to the existing SDN controller and the equipment management system, which tries to minimize further additions to the existing system. The configuration of the system is illustrated in Figure 4. Terminals a, b and c communicate with the server individually. Their respective communication flows are called Flows a, b and c. In this example, the link from the OpenFlow switch to the server is congested.

The management orchestration manages and executes the control scenario for the dynamic shaping. Specifically, it observes the entire situation and executes the necessary processes. When it detects a congested link, the management orchestration instructs the SDN controller to collect data on the communication rate, the number of hops and packet length for communication flows that pass through the congested link. The SDN controller gets these items of data from the OpenFlow Switch. However, if data on a large number of communication flows are to be collected, the performance of both the OpenFlow Switch and the SDN controller degrades dramatically. To avoid this, the OpenFlow Switch monitors the traffic counter of each communication flow and notifies the SDN controller, in advance, of the communication flows whose traffic counters exceed a certain threshold. The SDN controller requests the OpenFlow Switch to send traffic data of only these communication flows.

Since the SDN controller knows the route in the network of each flow in advance, it sends the numbers of hops of these communication flows to the management orchestration. Based on the collected data, the management orchestration instructs the equipment management system to set the shaping control parameter and to provide information about the positions of shaping functions. It also instructs the SDN controller to change the route (route switching) so that the communication flows subject to shaping will pass through the specified shaping functions. The main functions that the management orchestration should have, as stated above, are summarized in Table 1.

The cooperation method from selection of shaping function to route switching is illustrated in Figure 5, in which the number enclosed in squares indicates the process number.

- Process 1: The management orchestration instructs the equipment management system to select the optimal shaping points and to set shaping control parameters. It specifies the identifier of the network to which shaping is applied and information about the area, (α), where the terminal concerned is located, information about the matching rule [2], (β), of Flow a, which is subject to shaping, and the shaping rate (γ).
- Process 2: The equipment management system selects the optimal shaping point (Shap1 in this example) based on α, β, and the usage status of the shaping points.
- Process 3: After setting the shaping control parameters, the shaping function of the shaping point notifies the equipment management system of the completion of the parameter setting.
- Process 4: The equipment management system transfers the switch number (δ) of the switch to which the selected shaping point is connected and the switch port number (ε) to the management orchestration.
- Process 5: The management orchestration notifies the SDN controller of γ, δ and ε, and instructs it to change the route so that Flow a will pass through the selected shaping point.
- Process 6: The SDN controller rewrites the flow entry of the OpenFlow switch concerned based on δ and ε to change the route so that Flow a will pass through the shaping point. When shaping becomes no longer necessary, an instruction of the reverse operations is issued.

IV. CONFIRMATION OF THE OPERATION OF THE PROPOSED DYNAMIC SHAPING

A. Design and construction of an evaluation system

Since the main aim of the evaluation is to confirm the operation of the management orchestration function, we implemented the SDN controller, the OpenFlow switch, and the equipment management system using substitute devices (Software router VyOS [17]). The evaluation tool (C#-based software program) was designed and constructed to implement the management orchestration functions proposed in Section III. An example of the evaluation system is shown in Figure 6. It consisted of four VyOS nodes, three terminals, one control terminal and one server. Terminals a, b and c individually communicated with the server. It is assumed that the link from Node 4 to the server is congested. The shaping function was implemented using VyOS function, not dedicated functions.

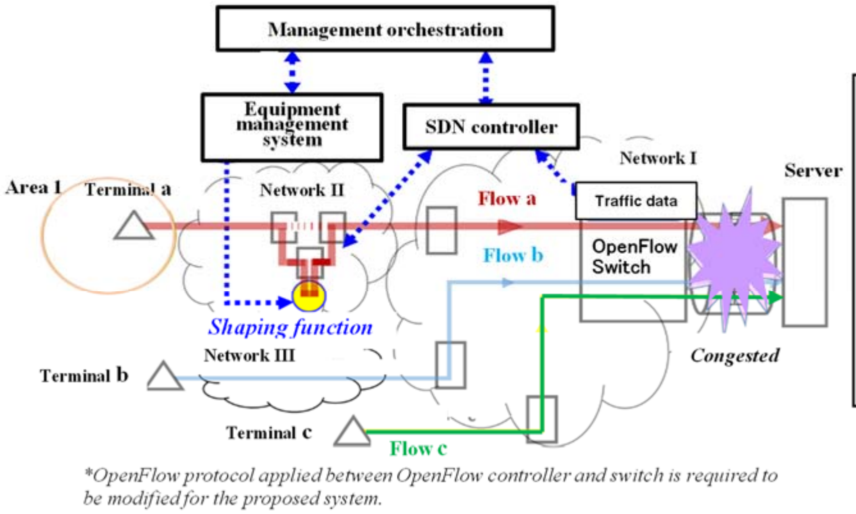

Figure 4. Proposed system configuration that automates dynamic shaping

TABLE 1. MAIN FUNCTIONS OF MANAGEMENT ORCHESTRATION

・Summarize the flow of processing and controls other functions. Check the progress of each function and manage the processing order. ・Collect line usage rate ・Collect communication speed for each flow ・Collect hop count for each flow ・Select shaping target flow *Select the largest x3 value ・Determine shaping data for shaping target flow ・Decide shaping function (or create by NFV) ・Set shaping function parameter ・Route switching ・Link with SDN controller ・Link with equipment management system

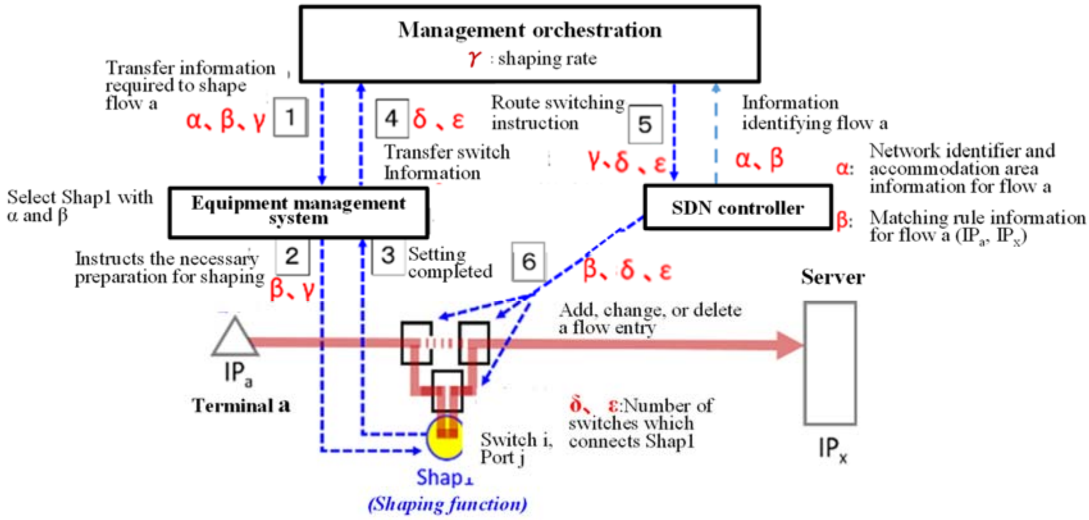

Figure 5. Cooperation method from selection of shaping function to route switching

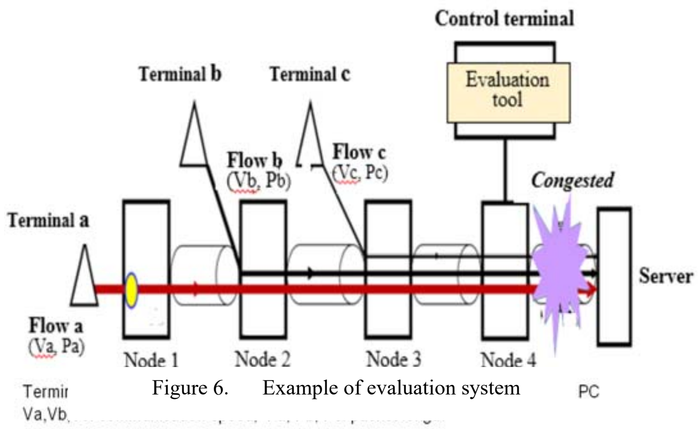

Figure 6. Example of evaluation system

B. Results and evaluation

The operation of the dynamic shaping was confirmed under the following conditions:

- Maximum bandwidth between node 4 and server: 20Mbps
- P_{max}: 0.7; Va: 2Mbps, Vb: 9Mbps, Vc: 6Mbps
- Pa: 5000 bytes, Pb: 3000 bytes, Pc: 500 bytes;
  Cp = 3000 * Cbs

Flow c with the largest x3 value among three flows is selected as the communication flow to be shaped in this example. In order to eliminate congestion, it is necessary to reduce the total communication speed from 17 Mbps to 14 Mbps (=20Mbps*P_{max}), and therefore the shaping rate of flow c will be 3Mbps. Figure 7 shows changes in the measured speed from Node 4 to the server. As had been expected, the speed of Flow c was reduced to 3 Mbps.

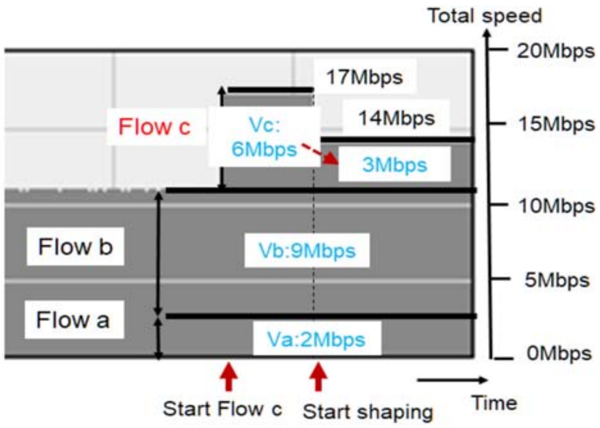

Figure 7. Measured speed from Node 4 to server in Fig.6

V. METHOD TO SIMPLIFY THE PROCESS OF COLLECTING THE TRAFFIC DATA OF EACH COMMUNICATION FLOW BY SDN CONTROLLER

A. Basic concept and overview

The dynamic shaping method proposed in Section II requires the SDN controller to continuously collect traffic data at regular intervals for each communication flow, which is the key function for the proposed dynamic shaping method. As a result, the load on the SDN controller becomes very large and the performance of the SDN controller degrades dramatically. To avoid this performance degradation, the following policies can be considered:

<Policy 1> Reduce the number of communication flows for speed measurement.

For example, one method is to measure the speed continuously only for the communication flow with a higher speed measured after the start of communication. Another method is that OpenFlow Switch notifies the SDN controller of the communication flow in which its traffic counter value is greater than a certain value in advance, and the SDN controller queries the OpenFlow switch for only the traffic data of the notified communication flows.

<Policy 2> Increase the speed measurement interval for each communication flow.

<Policy 3> Stop the traffic data inquiry processing itself at the SDN controller. The switch side periodically collects traffic data of communication flows to be monitored and estimates the communication speed (no inquiry from the SDN controller). Then, only when the speed has increased or decreased significantly compared to the previous cycle, the speed change is reported to the SDN controller. This is similar to the trap function of Simple Network Management Protocol (SNMP) [18], and the SDN controller instructs the switch in advance of the communication flow to be monitored.

In this paper, we propose a method based on the approach of policy 3, in order not to reduce measurement accuracy.

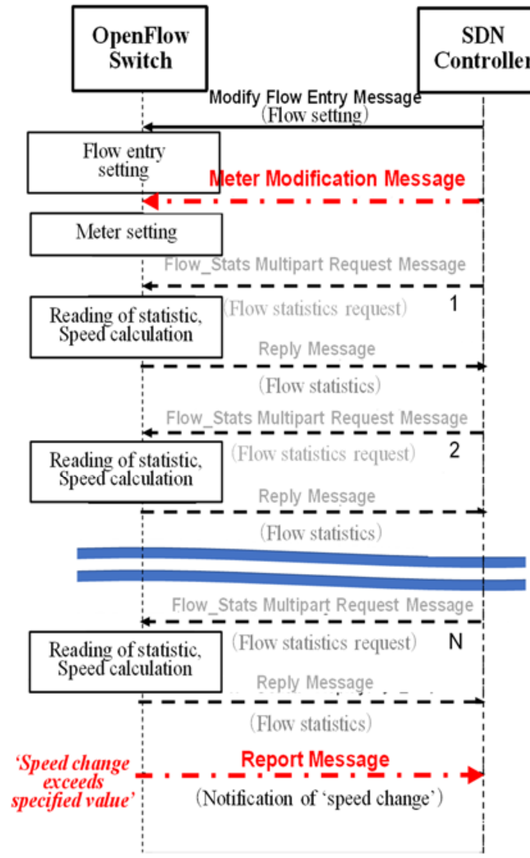

Figure 8. Message sequence of the proposed dynamic traffic shaping

B. Extension of OpenFlow specification

To implement the method proposed in Section 5.A, the following extensions to OpenFlow specification [11] are required:

(1) Add the following new fields to Meter Modification message.

-Communication flow ID to be monitored

-Communication speed measurement cycle

-Degree of changes of communication speed and packet length (Z_{0s}, Z_{0p})

* SDN controller is notified only when these are exceeded.

(2) In order to notify the SDN controller from the switch side, a "Report message" including the following fields is newly established.

-Communication flow ID

-Communication speed and packet length

C. Message Sequence of Proposed Method

Figure 8 shows an example of the message sequence of the proposed method based on Sections 5.A and 5.B. A broken line indicates a message which is used in the conventional method but becomes unnecessary this time, and a dashed line indicates a message which is added this time.

1) The SDN controller sends the switch a Meter Modification message containing the fields described in Section 5.A.

2) Unlike the conventional method, periodic exchange of the Flow_Stats multipart Request message, which is a request for statistical information for each communication flow, and the Reply message, which is a response to the request, is unnecessary. If the number of monitored flows is F for N times until the communication speed is notified, F*N messages processing could be eliminated in the entire SDN controller, and the processing load is greatly reduced.

VI. CONCLUSIONS

This paper has proposed a dynamic shaping method using SDN and NFV paradigms. This method can select the optimal communication flows to be shaped and the optimal shaping points dynamically. It has also presented a system configuration for automating the proposed dynamic shaping and the method to simplify the process of collecting the traffic data of each communication flow by SDN controller. The feasibility of the proposed method has been confirmed by an evaluation system.

It is required to evaluate the proposed dynamic shaping method experimentally in terms of resource utilization and stability comparing with the existing methods. It is also required to study how multiple SDN controllers and multiple equipment management systems collaborate.

REFERENCES

[1] Cisco, "Comparing Traffic Policing and Traffic Shaping for bandwidth Limiting". http://people.cs.pitt.edu/~znati/Courses/WANs/DirRel/RdngMtrl/Traffic_ShapingCISCO.pdf 2020.08.13

[2] T. Flach et al, "An Internet-Wide Analysis of Traffic Policing," SIGCOMM'16, 2016.

[3] M. Marcon, M. Dischinger, K. P. Gummadi, and A. Vahdat, "The local and global effects of traffic shaping in the internet," 2011 Third International Conference on Communication Systems and Networks (COMSNETS), pp.1-10, 2011.

[4] S. Bhaumik and S. Chakraborty, "Hierarchical Two Dimensional Queuing: A Scalable Approach for Traffic Shaping using Sontware Defined Networking," NetSoft 2018.

[5] Y. Zhao, B. Zhang, C. Li, and C. Chen, "ON/OFF Traffic Shaping in the Internet: Motivation, Challenges, and Solutions," IEEE Networks, Vol.31, Issue 2, pp.48-57, 2017.

[6] A. Grigorjew, F. Metzger, and T. Hobfeld, "Simulation of Asynchronous Traffic Shapers in Switched Ethernet Networks," NetSys2019.

[7] A. M. Aladwani, A. Gawanmeh, and S. Nicolas, "Traffic Shaping and Delay Optimization in Demand Side Management," 2013 UKsim 15th International Conference on Computer Modelling and Simulation.

[8] 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.

[9] D. D. Simion and G. D. Horia, " Traffic shaping and traffic policing impacts on aggregate traffic behavior in high speed networks," 6th IEEE International Symposium on Applied Computational Intelligence and Informatics, 2011.

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

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

[12] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg and S. Azodolmolky,"Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, Vo.103, Issue 1, pp.14-76, Jan.2015.

[13] "Network Functions Virtualization - An Introduction, Benefits, Enablers, Challenges and Call for Action," ETSI Introductory White Paper. https://portal.etsi.org/NFV/NFV_White_Paper.pdf 2020.08.13

[14] "Network Functions Virtualisation (NFV); Architectual Framework," ETSI GS NFV 002 v1.2.1, Dec. 2014.

[15] 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.

[16] 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.

[17] VyOS, https://www.vyos.io/ 2020.08.13

[18] IETF RFC3416 "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)" https://tools.ietf.org/html/rfc3416

 ISBN: 978-1-61208-833-4