

A Study on QoE Improvement of Online Games with UDP Multipathization by SDN

Sou Takabayashi¹, Yoshihiro Ito¹

¹Nagoya Institute of Technology, Japan

Abstract. This paper studies QoE improvement of Online games by applying the UDP/TCP multi-pathization method by SDN. The method can distribute packets for one stream to multiple paths by controlling the network with software with-out any new transport layer protocol. The author evaluates QoE for two actual online games using the multi-pathization method by experiments with subjects. The experimental results show the following. The authors show that multi-pathization by SDN is effective for online games. However, to further improve QoE, it is necessary to investigate and examine the packet length threshold that is appropriate for online games when distributing packets to each path.

1 INTRODUCTION

In recent years, the market for online games has been expanding in the world. The market size has exceeded 1000 million dollars and it is expected to reach 1700 million dollars by 2022[2]. Especially attention has been focused on 'e-Sports', which refers to competitions and sports played against computer games and video games. E-Sports competitions are held all over the world, and the competition population including the audience is said to be 350 million.

As online games develop, a more comfortable environment for users is desired. In order to consist of such environments, a network with low communication delay and packet loss is required. However, the number of online game services becomes large, the amount of traffic generated by them also increases. This causes congestion over the network and makes high communication delay and many packet losses.

To solve the above problem, we consider network control by SDN. SDN is an abbreviation of Software Dened Networking and is a generic term for technologies that can control network devices centrally with a single software and it can change network configurations and settings flexibly and dynamically. For example, UDP cannot use multiple paths simultaneously. However, by SDN, it is possible to realize UDP data transmission over multiple paths without any new protocol that supports multiple paths, such as MPUDP [3]. If the transmission over multiple paths can be performed by SDN, network congestion will be suppressed by distributing the load to multiple paths without changing any existing configurations on terminals. Furthermore, even if a part of a network is disconnected, the UDP data transmission can be compensated by the other paths. That is, the connectivity between end terminals can be kept.

As such a control, [1] proposes TCP/UDP multi-pathization method by SDN. In this method, OpenFlow switches are controlled by the OpenFlow controller, and packets are distributed to multiple paths. At that time, network congestion can be suppressed by checking congestion for each route and controlling switches. Reference [1] studies the application of the multi-pathization method to Web services. We think that the method is also effective for online games, and we investigate the effectiveness of it by evaluating the user's quality of experience (QoE) through subject experiments.

This paper is organized as follows. Chapter 2 describes the multi-pathization method by SDN. Chapter 3 and Chapter 4 show our experiments and the results, respectively. Chapter 5 concludes our paper.

The use of sections to divide the text of the paper is optional and left as a decision for the author. Where the author wishes to divide the paper into sections the formatting shown in Table 2 should be used.

2 Multi-Pathization Method by SDN

In the conventional communication method using UDP, only a single path can be used. Therefore, if the amount of traffic on this path increases, congestion occurs, and the service provided by UDP deteriorates. Therefore, there is a need for a network that can distribute packets to multiple paths and reduce congestion. In [1], the multi-pathization method by SDN is adopted as such network control.

As described in Chapter 1, SDN refers to dynamically controlling a network with software and its architecture. In the conventional network, when sending data from the server to the client, the transmission path is determined by each network device individually, and it is difficult to change to the optimum path according to the congestion. Besides, changing the network requires a lot of work because it requires setting up a large number of devices. However, SDN separates the path control and data transfer processing that has been performed by each network device one by one, and dynamically controls the device that performs data transfer processing by software. This enables flexible network control.

In the multi-pathization method used in the [1], when congestion occurs in the network, SDN control the network to send packets using a path without congestion. As a result, congestion on the network can be suppressed and the throughput of packet transfer can be improved. Furthermore, by using this method, end-to-end connectivity can be improved by using other paths even if there is a failure in one path. Also, SDN can be used to communicate with conventional protocols such as TCP and UDP without using new protocols such as MPTCP and MPUDP.

[1] uses OpenFlow as a technology for implementing SDN. OpenFlow performs path control with the OpenFlow controller and performs data transfer processing with the OpenFlow switch. In this experiment, We use OpenFlow to implement SDN.

3 Experiment

3.1 Purpose

We apply multi-pathization method by SDN to online games and assess QoE. For comparison, We assess QoE of online games even in an environment using a conventional UDP communication method. The purpose is to investigate the effectiveness of this multi-pathization method for online games.

3.2 Experimental environment

Our experimental network is shown in Fig. 1. This network consists of two game terminals (game terminal 1 and game terminal 2), a load client, a load server, an OpenFlow controller, four OpenFlow switches (OpenFlow switch 1 through OpenFlow switch 4) and a network emulator. We use Trema as an OpenFlow controller. Trema [5] is one of the frameworks of OpenFlow controller. We adopt Open vSwitch as openflow switches. The threshold value of the packet length used to assign packets to paths is set to 300 bytes as the same with [1]. Due to the circumstances of the experimental environment, subjects play the game on game terminal 1. Since game wins and losses may affect QoE, I always play as an opponent on game terminal 2 to maintain fairness.

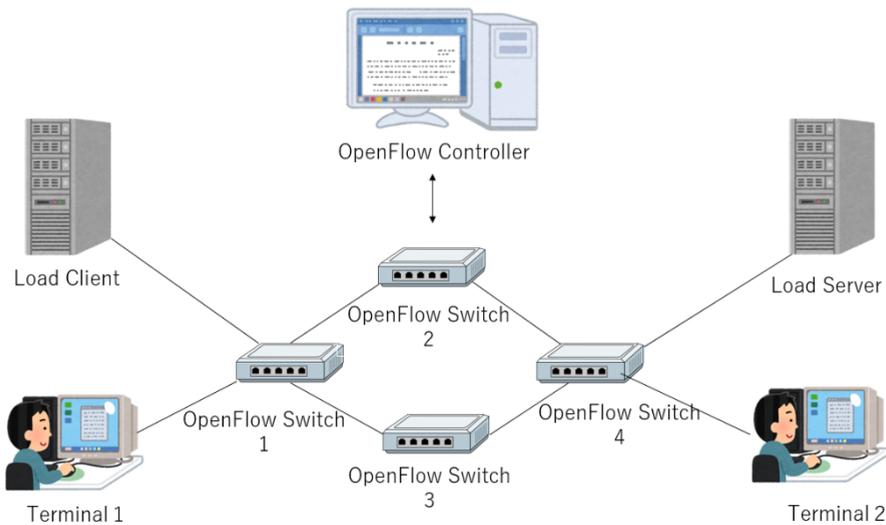


Fig. 1. Experimental configuration diagram

For comparison, we assess not only QoE with the multi-pathization method but also that without the method. Here we prepare two paths (Path1 and Path2) over the experimental network. Path1 and Path2 are constructed via OpenFlow switch2 and OpenFlow switch 3, respectively.

When we assess QoE without the multi-pathization method, the traffic between the game terminals passes through Path1. On the other hand, for the assessment of QoE with the multi-pathization method, we use both of Path1 and Path2. In either case, we generate load traffic, which makes a path congested over only Path1. In order to generate the load traffic, we use iPerf[7]. Moreover, we add delay by a network emulator which is between OpenFlow switches. When we assess QoE without the multi-pathization, we add delay between OpenFlow switch1 and OpenFlow switch2. For assessment QoE with the multi-pathization, we add delay between OpenFlow switch1 and OpenFlow switch2, and between OpenFlow switch1 and OpenFlow switch 3.

We consider the satisfaction, which is one of the usability scales standardized in ISO 9241-11, as a QoE parameter. We use the SD method [6] to assess satisfaction. The SD method is a psychological method that evaluates an individual's impression of a problem using conflicting adjective pairs. Each adjective pair is evaluated on a 5-point scale, and 20 sets of questions are used for the evaluation. The satisfaction in this paper is the mean of the voted score for the 20 pairs of questions. We used items related to video, related to sound, related

to operability, and related to the quality of experience such as 'interesting' and 'satisfied'. Fig. 2 shows the adjective pairs used in this experiment.

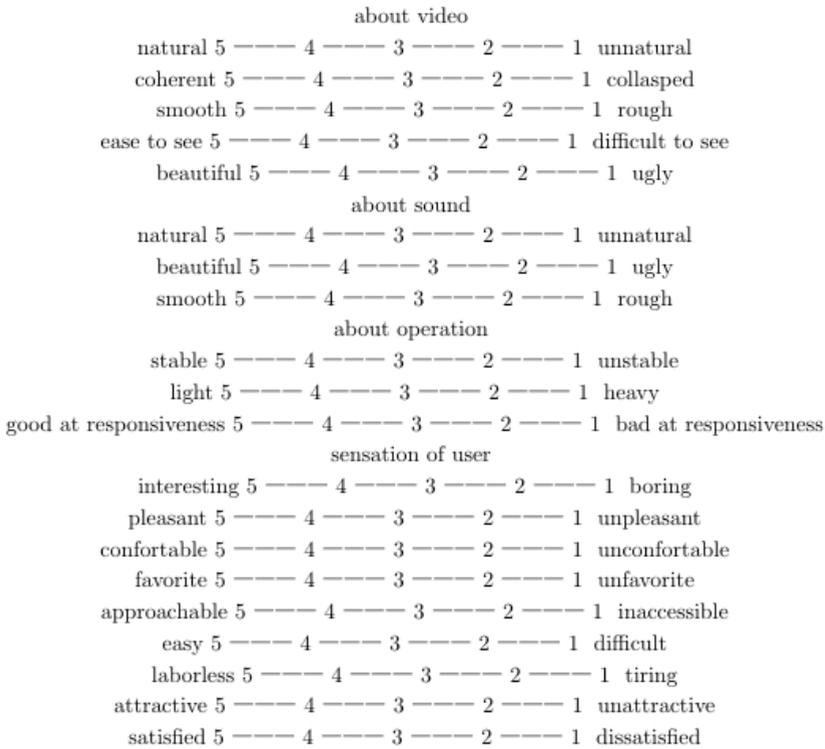


Fig. 2. Question items for experiments

In this study, we target two online games: Paintball Party 2[8] and SuperTuxKart[9]. Paintball Party 2 is a game that multiple players shoot guns and defeat each other. Players move on the 2D field using dash or jump and use items that are falling on the field. The winner is the player who defeated more opponents in time. Another game, SuperTuxKart is a car racing game. Players aim at the goal faster than other players by using not only the accelerator but also brake and drift. It is also possible to pick up items falling on the course and attack or accelerate using them. Paintball Party 2 can be played with up to 16 players, and SuperTuxKart can play with up to 8 players in a local environment, but in this experiment, two players play one-on-one in each online game. Fig. 3 shows the Paintball Party 2 game screen, and Fig. 4 shows the SuperTuxKart game screen.



Fig. 3. Game screen of Paintball Party 2



Fig. 4. Game screen of SuperTuxKart

In this paper, we treat 12 environments. Fig. 6 shows environment for Paintball Party 2 and Fig. 5 shows environment for SupertuxKart. Environments 5 to 12 use the multi-pathization method and environments 1 to 4 without it. Among them, the experiment is divided into environments with no load or delay, environments with the only load, environments with the only delay, and environments with both load and delay. For example, about Paintball Party 2, the number of connections when applying load was set to 374, and the delay was set to 200ms. If the load and delay are too small, users can play the game without complaints and if it is too large, they cannot be able to play the game. In order to avoid this, the load and delay values were set by the author in a preliminary experiment and found to be just right.

environment	path	number of connections	delay of path1(ms)	delay of path2(ms)
Env1	normal	0	0	X
Env2			200	
Env3		374	0	
Env4			200	
Env5	Multi-pathization	0	0	0
Env6			200	200
Env7			200	0
Env8			200	200
Env9		374	0	0
Env10			200	200
Env11			200	0
Env12			200	200

Fig. 5. Communication environment of Paintball Party 2

environment	path	number of connections	delay of path1(ms)	delay of path2(ms)
Env1	normal	0	0	X
Env2			500	
Env3		374	0	
Env4			500	
Env5	Multi-pathization	0	0	0
Env6			500	500
Env7			500	0
Env8			500	500
Env9		374	0	0
Env10			500	500
Env11			500	0
Env12			500	500

Fig. 6. Communication environment of SuperTuxKart

4 Results

We investigate the results for PaintParty2. First, we consider the case with no load. To compare with QoE of Env1(SINGLE), which is no delay and no load, Env 5(MULTI) is close to Env1 but Env6(MULTI) is lower than Env1. This reason is that in Env6 send packets using Path2 but the delay of Path2 is 200ms, so the assignment of packets to Path2 causes the deterioration of QoE. On the other hand, QoE of Env2(SINGLE), which is no load and with delay, is lower than Env7(MULTI) but higher than Env8(MULTI). This is probably because most of the packets used in this game use path 2. From the result, the multi-pathization with heterogeneous paths may degrade QoE of PaintParty2 with no load environment.

Next, let us consider environments with load. QoE of Env3(SINGLE), which is with load and no delay, is slightly but lower than that of Env9(MULTI). But QoE of Env4(SINGLE), which with load and with delay, is less than both of Env10(MULTI) and Env11(MULTI).

This might be because the influence of the load could be reduced by using the multi-pathization method. Therefore, for PaintParty2, under congested environment with long delay, the multi-pathization method is effective.

Next, we consider the cases of SuperTuxKart. Under environments of Env2, Env 4, Env6, Env8, Env10 and Env12, QoE is degraded. That is, for SINGLE, when with no delay, QoE becomes high regardless of the load. On the other hand, for MULTI, QoE is high when no delay for Path2. Also, in SuperTuxCart, the multi-pathization method assigns almost packets to Path2. Therefore, if we modify the multi-pathization method to select a path according to RTT of paths, QoE of these games will be higher than that of this time's experiments. So, we need to investigate the packet lengths in these games and consider the optimal threshold for assigning them.

From the results of experiments on two games, we found that the multi-pathization method used in this experiment is effective for online games because it can avoid congestion and delay. However, for future works, we examine the thresholds for distributing packets not only for the two games used in this experiment but also for various games.

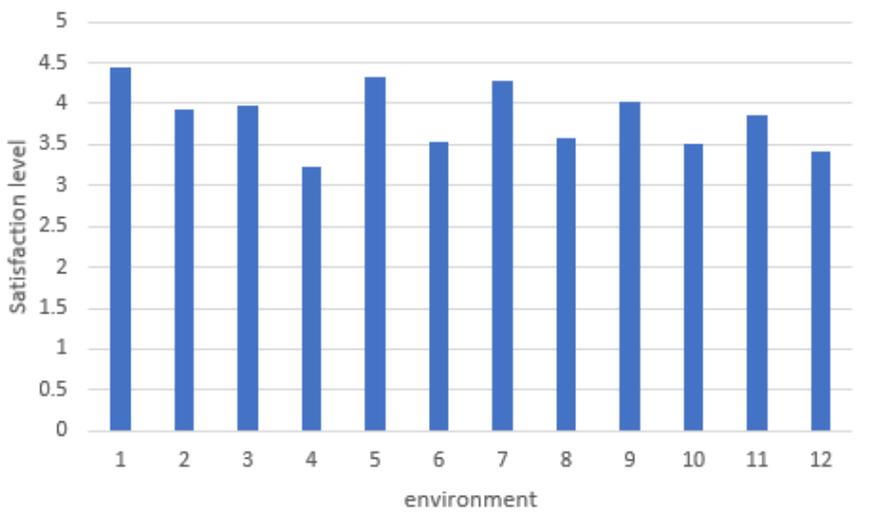
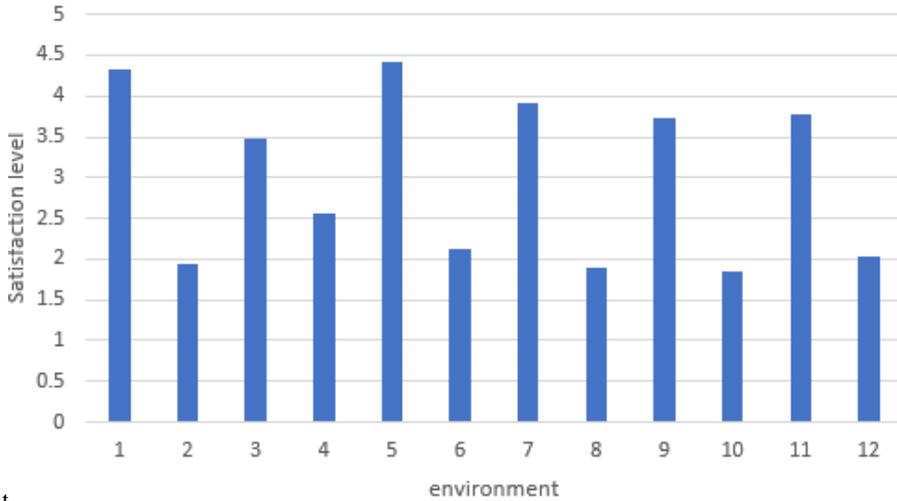


Fig. 7. Result of Paintball Party 2



t
Result of SuperTuxKart

Fig. 8

5 Conclusions

This paper studied the application of the multi-pathization method by SDN to online games by experiments with subjects. Experimental results show that this method is effective for online games.

However, it is not clear that the packet length threshold used to distribute used in this experiment is most effective for online games. Therefore, as future work, we will investigate the most effective threshold for online games.

6 Acknowledgments

The authors would like to thank Ms.Kaori Iwata and the member of our laboratory for their assistance in our research. In addition, we also thank the subjects who cooperated in our experiment.

References

- [1] K. Iwata and Y. Ito. " Proposal and Development of TCP Multi-Pathization Method with SDN by IoT Devices ". Proceedings of ITNAC, pp.1-16, November 2017.
- [2] <https://www.statista.com/statistics/490522/global-esports-market-revenue/>
- [3] S Liu, W Lei, W Zhang, H Li." MPUDP: Multipath Multimedia Transport Protocol over Overlay Network. " 5th International Conference on Machinery, Materials and Computing Technology (ICMMCT 2017), vol.126,pp731-734, April 2017
- [4] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental), January 2013.
- [5] _Trema_. <https://trema.github.io/trema/>.
- [6] _Semantic Differential_Scale<https://businessjargons.com/semantic-differential-scale.html>
- [7] _What is iPerf_. <https://iperf.fr/>
- [8] _Paintball Party 2_. http://www.t3-i.com/pages/project.php?id=paintball_party_2

[9] _SuperTuxKart_.https://supertuxkart.net/Main_Page