Application of TCP Multi-Pathization Method with SDN by IoT Devices to Web Service

Yuki Yoshida and Yoshihiro Ito
Nagoya Institute of Technology
Nagoya, 466-8555
Japan
Email: {yuki@en, yoshi@}.nitech.ac.jp

***Abstract*—IoT (Internet of Things) that everything is connected to the Internet and communicates is beginning to spread, and network control is required to respond to the rapid increase in traffic volume by IoT. However, it is difficult to realize such control with only existing protocols, and it takes time and cost to reform the protocol. Therefore, reference [1] suggests using IoT devices as SDN equipment and proposes TCP multi-pathization method with SDN which selects paths based on packet length and traffic volume of each path.**

On the other hand, due to the upgraded performance of IoT devices, it is expected to use Web services even on IoT devices, and it is conceivable to apply the [1] method to Web services. However, reference [1] does not validate the effectiveness of using Web services. In this paper, we aim to investigate the applicability of [1] method to Web services, and measure and compare the HTTP throughput and the response time by experiment. Experimental results show that this method is suitable for using Web services.

I. Introduction

In recent years, IoT (Internet of Things) devices, which are not only legacy communication equipments but also everything connected to the Internet, such as sensors, have become widespread. Consequently, as the number of users and that of services using IoT increases, the amount of traffic of IoT devices also increases. In order to deal with such a rapid increase in traffic of IoT devices, some traffic control in the network is mandatory and must meet two requirements: *flexibility* and *fault tolerance*. The flexibility allows the dynamic management of network. The fault tolerance prevents communication from being interrupted when one path of the network is lost.

Since services using IoT did not exist at the time when the major protocol suite of the Internet was standardized, it is difficult to realize the above-mentioned network control for IoT devices only by the existing protocols. For example, now we can use multiple access networks at a low cost. Therefore IoT devices can be expected to also use multiple access networks in the near future. However, TCP, which is the main transport protocol for the Internet, cannot use multiple paths at the same time and cannot utilize multiple access networks. To treat multiple paths, MPTCP (MultiPath TCP[2]) is standardized as a next generation transport layer protocol. Reference [3] shows the effectiveness of MPTCP in the use of Web services. However, in order to replace from TCP to MPTCP, it is necessary to exchange all clients and servers. This will take much time and cost.

To solve the above-mentioned problem, we consider a network control with SDN (Software Defined Network). By using SDN, we can realize multipath forwarding of TCP without utilizing any protocols that can use multiple paths such as MPTCP. If TCP can cope with multipath forwarding, clients and servers can suppress network congestion by load balancing to multiple paths without changing any existing environment. Furthermore, even if one of the paths is disconnected, the connection can be maintained on other paths, so that connectivity between the end terminals can be expected to be improved.

Reference [1] considers IoT devices as SDN devices and proposes a TCP multi-pathization method with SDN. It is expected that IoT devices can be operated as SDN devices by improving computer performance in the near future. Operating IoT devices as SDN devices can realize more flexible network control for the service using IoT devices.

Meanwhile, the standardization regarding current IoT devices and networks for IoT is not sufficient. Thus Web of Things (WoT) which uses web technologies such as HTML and JavaScript is standardized to develop IoT services and applications. A project named Physical Web by Google which gives URLs to all items and makes it accessible only on the Web is in progress. In this way, the movement to use the Web technology on the IoT device is advancing. However, the effectiveness of the proposed method of [1] at the time of using the Web service has not been verified. It is important to verify the effectiveness of the proposed method of [1], which can respond to the rapid increase of IoT traffic when using Web services.

This paper studies the effectiveness of TCP multi-pathization method with SDN by IoT devices proposed in [1] for Web services. The structure of this paper is shown below. Section II describes the TCP multi-pathization method adopted in this research. Section III describes the evaluation experiment of TCP multi-pathization method. Section IV describes the conclusion of this paper and future tasks.

II. Multi-Pathization of TCP

Reference [1] proposes a method of constructing multipath on SDN by branching packets according to their length and traffic volume of each path. This method of [1] has the following four advantages.

Improvement of throughput in congested networks
In the case where congestion occurs in the network, it is possible to use other than congested paths, so throughput of packet transfer can be expected to be higher in multipath than in single path.

Control of network congestion
By using multiple paths, it is possible to suppress severe congestion on the network.

Improve connectivity between end terminals
By multi-pathization, end-to-end connectivity can be improved by using other paths even when there is a failure in one route.

Use of existing protocol
By using SDN, it is possible to realize multi-pathization using TCP without using a new protocol such as MPTCP.

III. Evaluation experiment

A. Purpose

In order to evaluate the effectiveness of TCP multi-pathization method with SDN on the IoT network, we implement this method and perform QoS evaluation by experiment. In this experiment, QoS of multi-pathization method is compared with that of conventional TCP.

B. Experimental environment

1) Experimental configuration diagram: Our experimental environment is shown in Fig.1. This environment consists of an OpenFlow controller, four OpenFlow switches, an Web client, an Web server, a load client, a load server, and a network emulator. We use version 0.4.6 of Trema [4] as an OpenFlow controller. Trema is one of the popular frameworks of OpenFlow controller. On the other hand, as an OpenFlow switch, we use version 2.8.90 of Open vSwitch [5], which is virtual OpenFlow switch software. The IDs of the four openflow switches are 0x1, 0x2, 0x3, and 0x4. As shown in [1], the threshold value of the packet length used for judging the branching of the packet is 300 bytes.

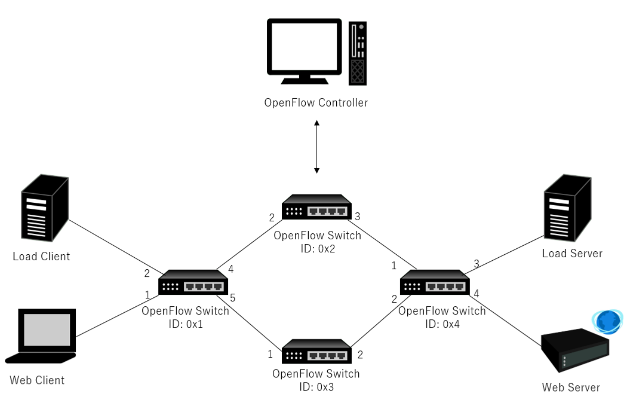

Fig. 1. Experimental configuration diagram

C. Evaluation scale

This experiment considers the HTTP throughput and the response time are as QoS parameters. The HTTP throughput is defined as the mean of the amount of transfered HTTP messages (Kbyte/sec) per unit time. Also, the response time is defined as the mean completion time (ms) of one request. ApacheBench is used to measure the HTTP throughput and the response time. ApacheBench is a Web server benchmarking tool attached to Apache HTTP Server [6].

D. Experiment contents

In the evaluation of QoS, we compare the throughput using normal TCP in a single path with that of our method. In this experiment, only the path via the OpenFlow switch with ID 0x2 is used for the conventional method. To cause congestion, the load traffic is generated by iPerf with respect to a path via an OpenFlow switch with ID 0x2 in the case of a single path, and a path via an OpenFlow switch with ID 0x3 in this method. Our experiments are performed in six types of experimental environments in which the number of TCP connections of load traffic is changed to change the amount of traffic on the path over each of single path and multipath. Each communication environment is shown in the table I.

TABLE I
Communication environment in experiment

environment	method	Number of TCP connections for load traffic
1	single-path	5
2		6
3		7
4	multipath	5
5		6
6		7

Under each communication environment, delay is added by the network emulator, thereby delaying the path between the OpenFlow switch with ID 0x1 and the OpenFlow switch with ID 0x2. There were five kinds of delay values of 0 ms, 10 ms, 25 ms, 50 ms, and 100 ms. With the load traffic being generated, the Web client generates a request to the Web server on the condition that the number of concurrent connections is 50 and the request is 10 requests per connection. We measure the HTTP throughput and the response time and calculates the maen value . In the experimental result, the mean values obtained from the measurements made five times for the HTTP throughput and the response time to be evaluated are used.

E. Experimental Results and Discussion

The experimental results of the HTTP throughput are shown in Fig.2 as the regression lines with the delay difference as independent variable and the HTTP throughput as dependent variable.

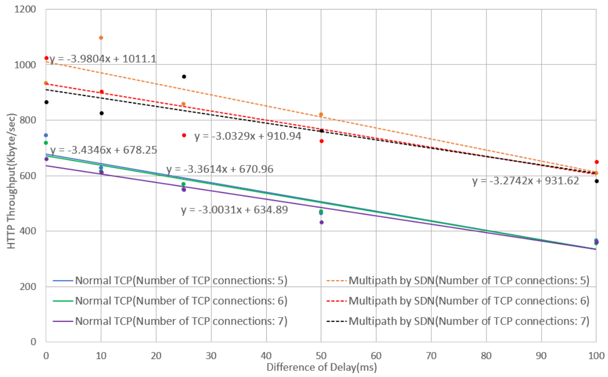

Fig. 2. HTTP throughput

In Fig.2, we see that the HTTP throughput of the proposed method is higher than that of normal TCP under any experimental environment. From the regression lines, the throughput decreases as the delay increases in both methods, but the degree of the decrease is almost the same in both methods even when the delay is increased, and then the throughput of the proposed method can be expected to increase. Therefore, from the viewpoint of QoS, we confirmed the effectiveness of the proposed method when using Web service.

Reference [1] measures the TCP throughput using the same method, and shows that the TCP throughput of the method is higher than that of TCP when the delay difference among paths is small. However, when the delay difference among paths becomes large, the TCP throughput of this method becomes smaller than that of TCP. This is due to the fact that when the delay difference between the two paths becomes large, the order of arrival packet changes frequently, and then TCP timeout retransmission occurs frequently. However, when this method is applied to the use of the Web service, even though the delay difference between the two paths increases, the HTTP throughput of this method remaines larger than that of TCP. The reason is that the number of connections is large in the case of the Web service and the generation time of one connection is short, so that the occurrence of retransmissions is small.

On the other hand, the experimental results of the response time are shown in Fig.3 as the regression lines with the delay difference as independent variable and the response time as dependent variable.

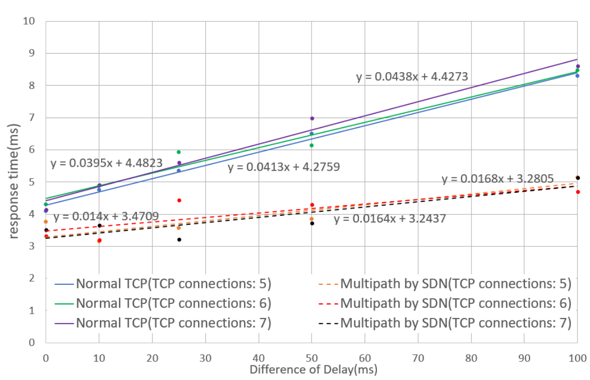

Fig. 3. response time

In Fig.3, we see that the response time of the proposed method is higher than that of normal TCP under any experimental environment. Fig.3 also shows that the inclination of the normal TCP regression line is larger than that of the proposed method, and the way the response time increases when delay increases is large. In other words, the proposed method increases the number of routes that can be used for transfer, so that the time to receive one request can be reduced as a result, and additionally, suppress the speed of increasing the time to receive one request when the delay increases more than normal TCP. Therefore, even when the delay on the path assumed in the real environment increases, QoE at the time of using the Web service is expected to be improved by the proposed method.

Reference [1] measured TCP throughput and confirmed the effectiveness of this method, but in this research we measured the HTTP throughput and the response time of HTTP request. From the above results, it is shown that this method is suitable for using Web services.

IV. Conclusions

In this paper, we implemented TCP multi-pathization method with SDN by IoT devices proposed by [1], and evaluated its effectiveness in Web service from the viewpoint of QoS by evaluation experiment.

As a result of the evaluation experiment, the HTTP throughput of the proposed method is higher than that of TCP, and the effectiveness at the time of using the Web service is confirmed. In addition, the degree of decrease in HTTP throughput when the delay becomes large is almost the same in both cases of TCP and this method, and even when the delay is further increased, the HTTP throughput of this method will be higher than that of TCP. As for the response time, the response time of this method becomes smaller than that of TCP, and the effectiveness at the time of using the Web service is confirmed. Furthermore, the proposed method suppresses the increase rate of the response time with respect to the increase of the delay compared to TCP. From the above, we can expect improvement of QoE when using Web services by this method.

Acknowledgment

This work was supported by MEXT KAKENHI Grant Number JP 16K00122.

The authors would like to thank the people of the National Institute of Information and Communications Technology and related people. In addition, they thank the member of their laboratory for assistance in research.

References

[1] K. Iwata and Y. Ito. “Proposal and Development of TCP Multi-Pathization Method with SDN by IoT Devices”. *Proc. ITNAC*, 2017.
[2] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. “TCP Extensions for Multipath Operation with Multiple Addresses”. RFC 6824 (Experimental), January 2013.
[3] Daisuke Yamauchi and Yoshihiro Ito. “A Study of Effect of MPTCP on Web Usability”. *GCCE*, 2014.
[4] “Trema”. https://trema.github.io/trema/.
[5] “Open vSwitch”. http://openvswitch.org/.
[6] “Apache HTTP Server”. https://httpd.apache.org/.