Application Specific Traffic Control using Network Virtualization Node in Large-Scale Disasters

Tsumugi Tairaku
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku,
Tokyo, JAPAN
Email: tsumugi
@ogl.is.ocha.ac.jp

Akihiro Nakao
Shu Yamamoto
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku,
Tokyo, JAPAN
Email: {nakao, shu}
@iii.u-tokyo.ac.jp

Saneyasu Yamaguchi
Kogakuin University
1-24-2 Nishi-Shinjuku,
Shinjuku-ku,
Tokyo, JAPAN
Email: sane
@cc.kogakuin.ac.jp

Masato Oguchi
Ochanomizu University
2-1-1 Otsuka, Bunkyo-ku,
Tokyo, JAPAN
Email: oguchi
@is.ocha.ac.jp

Abstract—When the Great East Japan Earthquake occurred in 2011, the network connectivity was significantly degraded in the wide area due to the multiple network failures as well as the traffic congestion. When the network failures occurred in multiple areas, it was difficult to quickly recognize the entire network situation only using the network traffic monitor system. In our prior works, we found that SNS messages contain the useful information to recognize the big picture of the network failures and proposed the network control system using SNS messages to improve the quickness of the network recovery. As the another critical issue in case of a large-scale disaster, users could not obtain the emergency information due to the network disturbance because the current IP network is operated not being aware of the applications. Thus we propose the application specific traffic control system with failure detection function based on SNS message to prioritize the important application traffic in the event of the large-scale disaster.

Based on a series of experiments, this paper shows the effectiveness of a system that detects connection failure based on social information and controls the network bandwidth for each application. Especially, we focus on application specific traffic control experiment. An automatic SDN control is performed with the network virtualization node FLARE having SDN extension capability as well as the network slicing capability. We perform the experiments to determine the type of application based on the traffic and perform bandwidth control for each application using real Internet applications.

Index Terms—SDN; OpenFlow; FLARE; traffic control;

I. INTRODUCTION

In the current environment of the Internet, the traffic from a wide variety of applications is mixed due to the high functionality of mobile terminals and the development of cloud computing. In such a network, the traffic patterns are significantly changed depending on the real world situation. During a disaster such as a large earthquake, the number of user accesses considerably increases. As a result, a large amount of traffic is generated in the network. In fact, when the Great East Japan Earthquake [1] occurred on 11th March 2011, the amount of traffic increased several times more than usual [2].

In addition, the network is usually operated by network traffic monitoring system. But as in such large-scale disaster, multiple failures occur in the wide area, it is difficult to obtain the big picture of the network failure only using traffic monitoring system.

Meanwhile, over one hour after the earthquake occurred, more than 1200 tweets per minute about the network disturbance of the damaged area were posted from Tokyo where the network access was available [3]. It can therefore be said that SNS messages such as Twitter [4] contain the useful information to obtain the situation of the network failures in such a large-scale disaster. Therefore, we propose the network control system using SNS information to automatically change the network configuration and to control the traffic by Software Defined Networking (SDN).

As the another important issue, people could not obtain the emergency information due to the network disturbance because the current IP network is operated not being aware of the applications. In addition to the information on disaster, users should be able to confirm the safety using the real time voice application. Thus we propose the application specific traffic control method to prioritize important applications such as the voice communication application of LINE [5], Skype [6], Messenger [7] and so on to communicate with people in damaged areas. As opposed to such applications, the entertainment applications such as YouTube and games might be less important in the event of a disaster. Therefore the mechanism to prioritize the voice application will be required in the network traffic control.

In the experiment, we use the FLARE switches. The FLARE switch is the network virtualization node which can create SDN enabled slices by instantiating OpenFlow datapath in their programmable data plane. In the network consisting of FLARE switches, we carried out the application specific traffic control experiment controlled by SDN using SNS messages of Tweets data when the Great East Japan Earthquake occurred. In addition, using the laboratory network simulating wide area...
test bed with FLARE switches, we perform the experiment using real applications on the application specific traffic management by identifying the application name of TCP packet flows and limiting the bandwidth of each application traffic using SDN controller.

The contributions of this paper are as follows.

i) By designing and implementing application specific SDN control system, we can restrict applications with low importance after network failure is detected.

ii) By using the slice function on FLARE switch, we can efficiently use the link bandwidth and prioritize important applications.

iii) By doing some experiments using real Internet applications, we can show the effectiveness of application specific SDN control system.

The rest of the paper is organized as follows. Section II describes related technology. Section III explains SDN and DPN. Section IV explains the procedure of the proposed SDN control system. Section V explains the experimental demonstration, and Section VI explains the experimental demonstration using two slices on FLARE switch. Section VII explains the experimental demonstration using real Internet applications. Finally, we conclude in Section VIII.

II. RELATED WORK

The automatic traffic control using SDN and OpenFlow [8] has already been implemented [9][10][11]. The SDN control system could dynamically reconfigure the network according to the network traffic change. The SDN control system monitoring the amount of the traffic can be deployed in the switch network as well. In addition, there were the studies of the traffic control systems using breaking news on earthquake sent by Twitter and the information posted in the earthquake category of Yahoo! News [12][13].

These network controls do not directly use the users’ messages on the network disturbance whereas our proposed system controls the network by SNS messages themselves.

In addition, our scheme control the traffic by the specific application. Namely, the traffic of emergency such as that of the voice application is prioritized rather than entertainment purposes application such as YouTube and games by classifying the type of the applications. Therefore, our approach is different from other studies of the network control without the application based traffic classification.

In our prior works, we perform the routing experiment using FLARE switches [14][15][16][17] that can switch the route by detecting network failure based on SNS messages with wide area network test bed. In this paper, we combine the application specific traffic control with the network reconfiguration using SNS information.

III. SDN AND DPN

A. SDN

SDN (Software Defined Nework) is a concept which makes it possible to dynamically set the network configuration and functions by software, or the architecture. OpenFlow, which is one of the SDN implementation technologies, separates the control plane that controls routing and the data plane that performs data transfer, which were conventionally included in one network device. These two functions are integrated into a conventional hardware switch. In contrast, in OpenFlow C-plane is implemented as software on the server outside of the switch, ordering the D-plane inside of switch to transmit data. OpenFlow is the standard interface to connect these two parts.

B. DPN

DPN (Deeply Programmable Network) is a concept of evolving SDN, and the difference with the conventional SDN is that it is possible to program not only the control plane that controls the network but also the data plane that transfers packets. There is a FLARE switch in the switches that implement this DPN concept. Fig.1 shows the architecture of FLARE. FLARE can install multiple virtual switches called slice on the physical switch and assigns packets to each slice with Slicer slice. This slice is independent of each other. FLARE can program and operate multiple network functions by network virtualization. Therefore it is appropriate for each application to be virtualized properly. In addition, FLARE supports the OpenFlow function, and FLARE’s OpenFlow is implemented by Click module router [18]. So using FLARE makes it possible to prioritize the traffic of a specific application in an emergency disaster. In this paper, we use FLARE switch to realize application specific traffic control.

IV. PROCEDURE OF THE APPLICATION SPECIFIC SDN CONTROL SYSTEM

The procedure of the automatic SDN control system for each application in the event of a disaster is outlined as follows. Fig.2 shows the procedure of this system. In the flow of (1)-(3), we control the network automatically.

1) Failure detection

Qiu et al. [19] reported that users posted messages on Twitter before they called a customer service center if
they experienced network failures. So, we use Twitter to detect network failures. Tweets are monitored in real time to detect where the connection failure has occurred. We collect the tweets including the keywords related to the network connection failure and summarize each tweet that contains the same place name noun. In addition, we judge whether the place appearing in the tweets is having connection trouble using machine learning and count the number of tweets related to communication failure for each prefecture [20].

2) Application identification

We use user terminal that installed the client software for judging the application. On the client PC user uses, the system identifies the application and attaches the trailer bytes for application identification. It observes the flowing packets and identifies the application by checking process table and destination packet number. In this study, it monitors only SYN packets, which are generated at TCP session establishment, because there are a large number of packets. In addition, the application information is added to the end of the packet. In this study, the system attaches the application name and the length of the application name.

3) SDN control for each application

Of the packets forwarding to the FLARE switch, only the SYN packets are sent to OpenFlow controller. The application name is identified by checking the trailer bytes added in the client PC; whether the application is important for users in the event of a disaster is also determined, and the system adds Table id for each application when we perform the SDN control for each application. Table id is the id of flow table to control for each flow. After it adds the Table id for each application, the application identification is removed from the end of the packets.

This system prepares shell scripts that restrict the bandwidth for each Table id using REST-API of OpenFlow. It resets the bandwidth automatically from the controller.

V. EXPERIMENT

In this section, we show the effectiveness of application specific SDN control system by the experiment.

A. Experiment scenario

Fig.3 shows the outline of the network control system for each application. This system attaches the application identifier on the client PC and performs the network control for each application by appending the table id necessary for the network control for each application and removing the identifier on the FLARE switch. In addition, in these experiments, packets are connected to external networks and servers via the FLARE switches and Proxy server.

Fig.4 shows the experimental network. Four FLARE switches are connected in a mesh shape, and each switch can be controlled by a controller. Client PCs are connected to FLARE switch 1, and a proxy server is connected to FLARE switch 2. The line between FLARE switch 1 and FLARE switch 2 and the line between PCs and the switches are 1G, the other lines are 10G. The FLARE switch 1 and 2 were assumed as Miyagi (Earthquake concurrence area) and Tokyo respectively. As SNS information, the Tweets of the Great East Japan, which occurred during 14:00 and 15:00, March 11, 2011 are used. The system detects network failure between Miyagi and Tokyo and limits the bandwidth of each application traffic.

In the experiment, we sent traffic that simulated YouTube and games considered to be low important at the time of a large-scale disaster, Skype considered to be high importance to the Proxy server, and operated this system which limits bandwidth for each application. YouTube and games are applications for amusement purposes that use a wide range of bandwidth and they are considered to be low important in large-scale disasters. But Skype is an application that can exchange information in real time, so it is important during large-scale disasters and considered to be be high important. When it is detected that a communication failure has occurred between FLARE switch 1 and FLARE switch 2, the system...
limits the bandwidth of traffic simulating YouTube to 300 Mbps as the operation after the network failure is detected, and the traffic simulating Skype is automatically restricted. We measure each throughput with iperf.

Fig. 4. Experimental network used in Section V

B. Experiment result

We measure the throughput of band-limited packets and the throughput of packets without bandwidth limitation.

Fig.5 shows the throughput of YouTube which was band limited to 300 Mbps by the system and the throughput of Skype. From the result, Skype has the throughput of about 500 Mbps, while YouTube shows that the throughput is about 280 Mbps. The reason that YouTube is suppressed to 280 Mbps is because the system limits the bandwidth to 300 Mbps. In other words, after the network connection failure is detected, the system can restrict only YouTube, which is considered to be low importance. And it proves that the bandwidth control for each application is actually useful based on experimental results.

VI. EXPERIMENT WITH TWO SLICES

In this section, we show the effectiveness of using two slices on FLARE to use link bandwidth effectively when bandwidth limiting is performed for each application.

A. Experiment scenario

With using the network in the laboratory environment used in Section V, we experiment with two slices on the FLARE switch in order to effectively use the link bandwidth while limiting the bandwidth for each application. Fig.6 shows an experimental network. Two slices are prepared for each switch, and we pass YouTube on Slice 1, and Skype on Slice 2. Traffic is allocated to each slice by Slicer slice, and the bandwidth limiting is performed on each slice. As in Section V, when it is detected that a communication failure has occurred between FLARE switch 1 and FLARE switch 2 based on the Tweets when the Great East Japan occurred, the system limits the bandwidth of YouTube to 300 Mbps as the operation after detecting the network failure and does not limit the bandwidth of Skype. The throughput of each traffic is measured.

B. Experiment result

Fig.7 shows the throughput of Skype and the throughput of YouTube when using one slice and two slices. Looking at the middle and the right side of the graph, we can see that the throughput of Skype is about 610 Mbps when using two slices, although the throughput of Skype is about 470 Mbps with one slice. In other words, by using two slices, it proved that link bandwidth can be used effectively when bandwidth limiting is performed for each application.
VII. CONGESTION CONTROL EXPERIMENT USING REAL APPLICATIONS

In this section, we demonstrate the effectiveness of application specific SDN control system by experiments using real Internet applications.

A. Experiment scenario

Fig. 8 shows the experimental network. The experimental network used is the same as in Section V. and VI. In the local environment, we narrow the link bandwidth between FLARE switches because there is a margin in link bandwidth. In the experiment, we narrow the link bandwidth to 1Mbps, 500Kbps, 300Kbps, and 100Kbps. The FLARE switch 1 and 2 were assumed as Miyagi (Earthquake concurrence area) and Tokyo respectively. As SNS information, the Tweets of the Great East Japan, which occurred during 14:00 and 15:00, March 11, 2011 are used as with Section V and VI. The system detects network failure between Miyagi and Tokyo and limits the bandwidth of each application traffic.

Experiments are carried out using real Internet applications. We use Skype as a high important application and YouTube as a low important application in large-scale disasters and operate the application specific SDN control system which identifies the application name of TCP packet flows and limits bandwidth for each application. YouTube is an application for amusement purpose that uses a wide range of bandwidth and is considered to be of low importance in large-scale disasters. But Skype is an application that can exchange information in real time, so it is considered to be of high importance during large-scale disasters.

At the same time we do videophone call with Skype, we stream YouTube in the experiment. When it is detected that a communication failure has occurred between FLARE switch 1 and FLARE switch 2, the system limits the bandwidth of YouTube to 0 Mbps as the operation after the failure detection. We measure the disturbance of Skype voice and image before and after SDN control.

B. Experiment result

Fig. 9 shows Skype voice and image disturbance before and after bandwidth control. Before the control, videophone of Skype was impossible to use in the link band restricted to 400 Kbps and 300 Kbps, but after the control there was a disturbance in voice and image, but it became possible to use a videophone call.

From the experiment, it became possible to use Skype videophone which is thought to be of high importance at the time of a large scale disaster by limiting the bandwidth of YouTube. It was proved that the proposed system was useful by performing experiments using real Internet applications.

VIII. CONCLUSION

In the case of a large scale disaster such as the Great East Japan Earthquake, a method of quickly recognizing the entire network situation by detecting sudden changes in real world from social data and a method of judging traffic type and prioritize the important application traffic by controlling for each application are required. As a previous study, we implemented and evaluated a routing control system based on social information such as Twitter, which is necessary for detecting...
network failure in the event of a large scale disaster such as the Great East Japan Earthquake, etc. In this research, we implemented a system that identifies the application name of TCP packet flows and controls the bandwidth of each application, and did some experiments using network virtualization node. In the experiments, it is proved that the proposed SDN control for each application is actually useful. Flow corresponding to the application was generated by iperf and the experiment was carried out. It was confirmed that the bandwidth can be individually controlled for the flow, and the throughputs of the band-limited traffic and the traffic without bandwidth limiting are measured. We experimentally confirmed that this system is operating. In addition, we used multiple slices and bandwidth control. The results showed that by using multiple slices, the system can use the bandwidth effectively compared to using only one slice. Also, we used real Internet applications and operated the application specific SDN control system and conducted an experiment. In the experiment, the system can actually control traffic by application identification.

As a future task, we will reproduce the state close to the congestion state where the traffic at the time of disaster and perform the experiment of this system. And we will perform experiments with the system combining the bandwidth limitation and the route switching.

ACKNOWLEDGMENT

This work is partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) and JST CREST Grant Number JPMJCR1503.

REFERENCES

[18] UCLA.edu: "The Click Module Router Project" http://read.cs.ucla.edu/click/click