Complex Event Processing to Detect Congestions in Mobile Network

Tatsuya TAKAHASHI[*], Hiroshi YAMAMOTO[*], Norihiro FUKUMOTO[**], Shigehiro ANO[**], Katsuyuki YAMAZAKI[*]

[*]*Nagaoka University of Technology, 1603-1 Kamitomioka, Nagaoka, Niigata 940-2188 Japan*

[**]*KDDI R&D Labs. Inc., 2-1-15 Ohara, FujiMino, Saitama, 356-8502 Japan*

Contact: taka_tatsu@stn.nagaokaut.ac.jp

Abstract— **With a wide spread of smartphones and tablets, a mobile network becomes frequently congested when many users concentrate to the same place. Especially when a large-scale event is held, a heavy network congestion interferes with the communication of the participants as well as local residents. In order to detect the network congestion, a large amount of traffic log should be analyzed in real time. In this paper, the proposed system attempts to detect a sign of the congestion by using a CEP (Complex Event Processing). First, by analyzing network status when the large-scale event, Nagaoka Fireworks festival, is held, it is observed that the network congestion can be effectively detected from the combination of (1) the RTT, (2) the specific type of TCP session termination (FIN-No-ACK) and (3) the number of retransmission packets. Next, we develop our proposed congestion detection system by using a CEP for detecting these metric in real-time. Through the experimental evaluation, it is concluded that the proposed system can scalably analyze a large amount of traffic log in real-time.**

Keywords— **Network Congestion, Mobile Network, Congestion Detection, Stream Data Processing, Complex Event Processing**

I. Introduction

With a wide spread of smartphones and tablets, a mobile network becomes frequently congested when many users concentrate to the same place. The network congestion in a specific and limited place may cause serious problem. In particular, when a large-scale event is held, the total amount of network traffic becomes much larger than usual. And, a heavy network congestion interferes with the communication of participants as well as local residents. In order to detect this network congestion, a large amount of traffic should be analyzed in real-time.

On the other hand, a stream data processing is widely studied in order to handle a large amount of data in real-time. In particular, we focus on a Complex Event Processing (CEP) [1] which is one of the stream data processing, because the network traffic is the typical streaming data consisting of multiple events.

Therefore, this paper proposes a new real-time network congestion detection system which attempts to detect a sign of the congestion by using a CEP. We first analyze a traffic captured on a mobile network when a large-scale event, Nagaoka Fireworks festival [2], is held in order to decide functional structure of the system [3]. As a result, it is concluded that a RTT (Round-Trip Time), a specific type of TCP session termination and that a retransmission packet are extracted from the log, and the combination of the three metrics is useful for detecting the congestion. Then, our proposed distributed system is implemented using CEP so as to detect these metrics in real-time. Through the experimental evaluation, we evaluate the processing performance of our proposed CEP-based system which is developed on a cloud computing environment.

II. Related Works and Objectives of This Study

The detection method of network failure has been proposed in order to provide high-quality and high-reliability services [4], [5]. The network failure expresses a condition where a communication is stopped due to failure of network equipment or configuration errors. However, even if a network failure does not exist, a network congestion occurs when a large amount of traffic concentrates to a specific place. Therefore, the network congestion detection method should be considered in a different manner. In addition, the network failure is sometimes caused by heavy network congestion [6], [7]. Hence, the congestion detection is useful for avoiding the network failure.

On the other hand, a stream data processing becomes widely studied [8], [9], and is mainly utilized to analyze web access [10], sensor data [11] and automotive driving [12]. The processing technique can also be applied to the detection of the network congestion, because a traffic log is the time series data generated continuously from the measurement equipment (e.g., routers, switches).

In particular, we focus on a CEP which is a suitable technique for quickly analyzing a large amount of data [13], [14]. By analyzing a combination of multiple data streams or events, the CEP can extract more effective event from them. The network traffic is the typical streaming data consisting of multiple events. Furthermore, analysis of a large amount of traffic log should be completed in a short time for detecting a sign of the network congestion. Therefore, by using the CEP, the network management system can rapidly and precisely detect the occurrence of the congestion. In this research, we utilize an Oracle CEP [15] for building our proposed system. Oracle CEP is free for prototyping, and has been applied to various commercial systems.

Furthermore, in recent years, a web service that provides a virtual cloud computing environment such as Amazon EC2 (Amazon Elastic Compute Cloud) [16] provided by AWS (Amazon Web Services) [17] is attracting attention. By utilizing Amazon EC2, computing resources can be prepared on-demand for constructing high performance distributed processing environment [18].

Considering these situations, we propose to use the Oracle CEP on the virtual machine of an Amazon EC2 so as to detect a sign of the network congestion in real-time.

III. Proposed Congestion Detection System

A. *Overview of Proposed System*

Figure 1 shows an overview of the proposed CEP-based congestion detection system. This system is composed of three functions. First, this system captures data traffic on a communication path between user terminals and a server on the Internet. The capture point sends the traffic log to the distributed processing server in a cloud computing environment, and a large amount of data is analyzed by utilizing the CEP. Second, the CEP server analyzes metrics for congestion detection (i.e., RTT, a specific type of termination of each TCP session and retransmission packet). Finally, the CEP notifies a network administrator of a sign of network congestion. The proposed system utilizes Amazon EC2 because the virtual server is useful for increasing the number of distributed servers in the future.

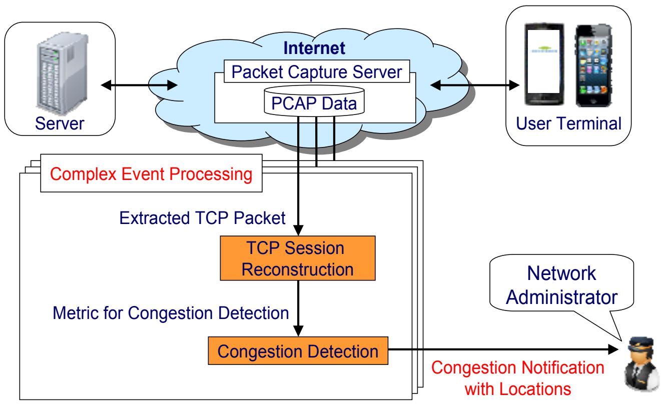


Figure 1. Overview of the congestion detection system

B. *Functional Detail of Proposed System*

Figure 2 represents a functional structure and a flow of the proposed system. First, the TCP session is reconstructed from the original captured traffic in order to detect the metric for network congestion. Next, these network metrics (RTT, a specific type of TCP session termination (FIN-No-ACK), and the number of retransmission packets) are derived from the extracted data. The reason why these metrics are selected for detecting network congestion is explained in sections IV and V. Finally, the proposed system detects the network congestion by analyzing time variation of these network metrics.

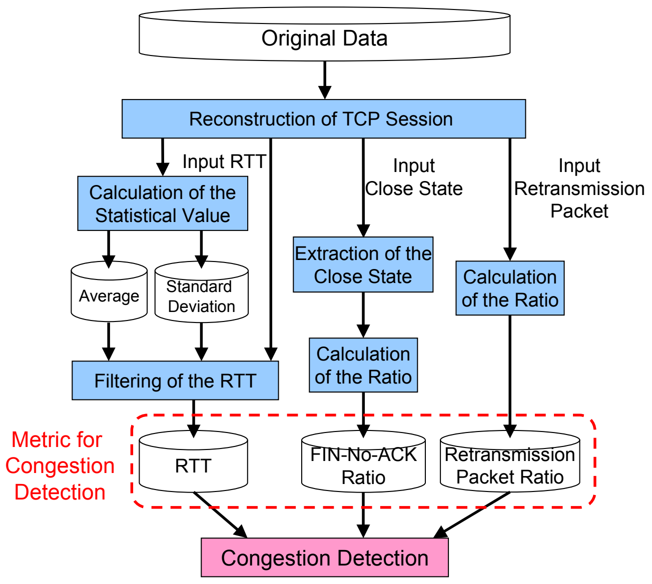

Original Data
Reconstruction of TCP Session
Input RTT
Input Close State
Input Retransmission Packet
Calculation of the Statistical Value
Average
Standard Deviation
Extraction of the Close State
Calculation of the Ratio
Calculation of the Ratio
Filtering of the RTT
Metric for Congestion Detection
RTT
FIN-No-ACK Ratio
Retransmission Packet Ratio
Congestion Detection

Figure 2. Functional structure

IV. Analysis for Evaluation Metrics of Congestion Detection

A. *Format of the Original Captured Data from PCAP*

In this study, the data traffic during the Nagaoka Fireworks festival is compared with that during the usual day. The Nagaoka Fireworks festival is ranked as one of the three biggest Japanese firework festivals. The site of this festival is Shinano riverside, Nagaoka-city, Niigata-prefecture, Japan, and it has been held on August 2 (Thu.) and 3 (Fri.), 2012 and 2013. In this festival, more than 400,000 people participate per one day. Therefore, during the festival, serious congestion is expected to occur in Nagaoka-city. By analyzing the captured data traffic, we attempt to extract the sign of the network congestion.

In the evaluation, the original data is captured on August 1 (normal day) and 2 (event day), 2012 and 2013. Table 1 shows a format of the obtained data. The important elements of the traffic log are summarized below;

- begin, end: Start time and end time of the TCP session.
- close state: Reason why the TCP session is closed. The detail is explained in section IV-C.
- client_rtt: RTT of the communication path between user terminals and a capture point. The RTT is measured when three-way-handshake of TCP session is performed.

TABLE 1. DATA FORMAT OF THE CAPTURED TRAFFIC

begin	end	close state	src ipaddr	dst ipaddr	src port	dst port	upload size	download size	upload n_packets	download n_packets	client rtt
8-1_16:59:29.35	8-1_16:59:36.80	clt_fin/srv_fin	x.x.x.x	a.a.a.a	37238	80	1106	516	8	7	79682
8-1_16:59:45.02	8-1_16:59.49.72	timeout	y.y.y.y	b.b.b.b	57972	443	839	2348	11	8	77283
8-1_16:59:46.02	8-1_17:01.02.98	srv_fin/clt_fin	x.x.x.x	c.c.c.c	36028	443	869	34228	26	29	99712
8-1_17:00:52.02	8-1_17:01.30.10	clt_fin/srv_fin	z.z.z.z	a.a.a.a	58326	80	1141	3069	14	8	186060
8-1_17:01:54.46	8-1_17:02.02.10	clt_fin/timeout	z.z.z.z	d.d.d.d	57977	443	0	0	2	1	99450

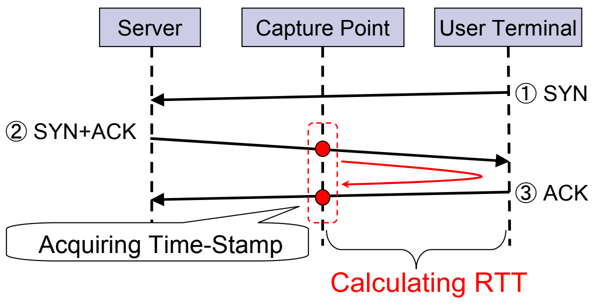


Figure 3. Metric-1: RTT

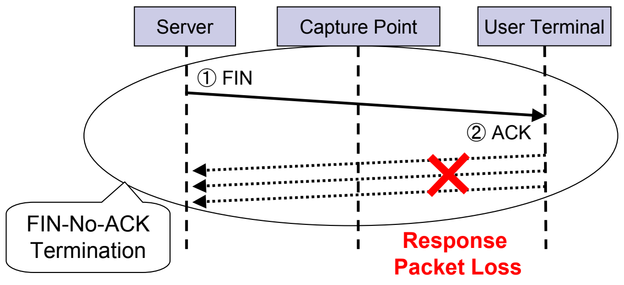


Figure 4. Metric-2: FIN-No-ACK termination

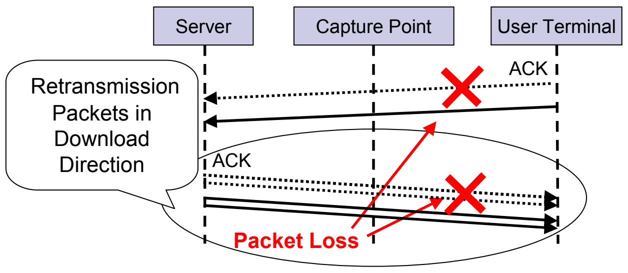


Figure 5. Metric-3: Retransmission packet

B. *Metric-1:RTT*

In this study, the RTT is round-trip time of the communication path between user terminals and a capture point as shown in Figure 3. As illustrated in this figure, the capture point obtains a time-stamp when capturing "SYN+ACK" packet from the server and "ACK" packet from the user terminal. The RTT between the capture point and the user terminal (i.e., mobile access network) is a difference between these time-stamps. The proposed system measures RTT of mobile access networks where the network resources are shared by many users. Therefore, the RTT can represent the congestion condition of the access network.

C. *Metric-2:FIN-No-ACK Termination*

We attempt to evaluate whether or not the type of TCP termination is also useful to detect the congestion. The "close state" of Table 1 represents a type of the TCP session termination. Some examples of the close state are as follows:

- srv,clt/fin: the server or client has requested the session termination.
- srv,clt/rst: the TCP session has been reset by the server or client
- timeout: Any packet has not been transmitted for a given time interval.

In this study, we focus on the case of "timeout" after "srv_fin" event is occurred, because the congestion of the access network may cause the loss of the response packets as represented in Figure 4. In this study, the "FIN-No-ACK" termination is referred to as a case where the timeout is occurred after "FIN" packet is sent from the server.

D. *Metric-3: Retransmission Packets*

In recent years, SACK (Selective ACKnowledgement information) option is set to almost all the TCP sessions. Therefore, the number of retransmission packets is equivalent to the number of packet losses. Main reason of the packet loss is network congestion. The number of packet losses in download direction is large compared with that in upload direction because the number of packets in download direction is larger than that in upload direction. Accordingly, we focus on the retransmission packet in download direction to detect the network congestion of the access network.

E. *Procedures of each Metric for Congestion Detection*

Procedures of analyzing the RTT are described below.

1) Data including the RTT is extracted from the original captured traffic.
2) The average and the standard deviation of RTT during each one minute are calculated. Here, a distribution of the RTT is assumed as a normal distribution for simplicity, and an upper threshold of the RTT is decided so that a confidence level becomes 1%.
3) The average of the RTT values that are smaller than the threshold are calculated.

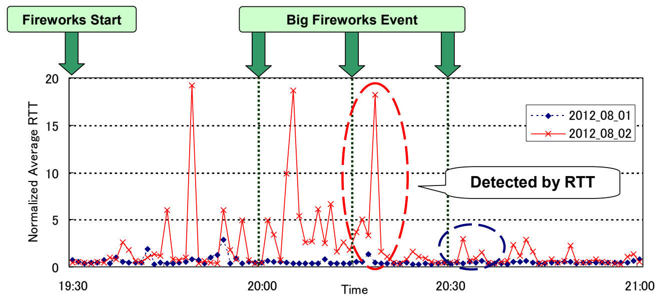
Next, procedures of analyzing FIN-No-ACK termination and the number of retransmission packets are described below.

1) The FIN-No-ACK termination and retransmission packets are extracted from the captured data.
2) The ratio of the FIN-No-ACK termination and the number of retransmission packets are calculated per one minute. The calculated value is used to estimate the degree of the network congestion.

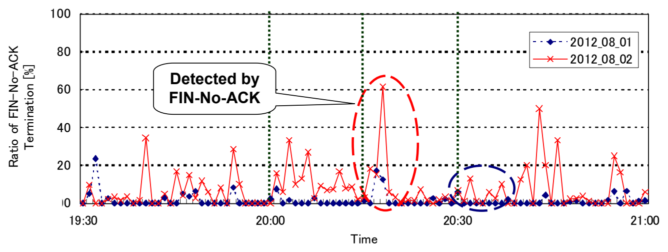
V. Evaluation of Metric for Congestion Detection

A. Time Valuation of the Metrics for Congestion Detection

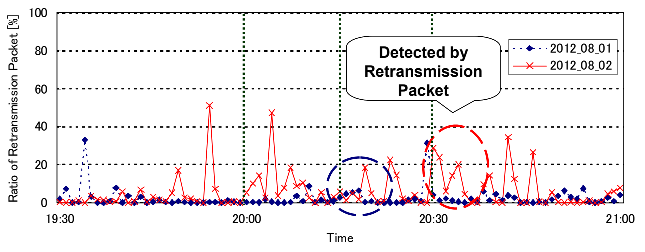
We evaluated whether the network congestion can be detected by analyzing the average RTT, the ratio of the FIN-No-ACK termination and the number of retransmission packets. Figure 6 shows the time valuation of (a) the average RTT, (b) the ratio of the FIN-No-ACK termination and (c) the ratio of the retransmission packet in each minute during 19:30 to 21:00 on August 1 (normal) and 2 (event day), 2012, in Nagaoka-city. In this figure, the RTT is normalized by the average value for the two days. People had been the most concentrating to the venue of fireworks because the fireworks had been displaying on this period of time.



(a) Metric-1: RTT



(b) Metric-2: FIN-No-ACK termination



(c) Metric-3: Retransmission packet

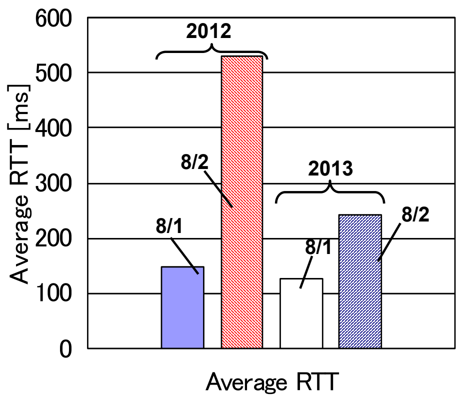
Figure 6. Time Valuation of the Metrics for Congestion Detection

The network congestion was not detected on August 1 because Nagaoka Fireworks Festival was not held on this day. On the other hand, these metrics have increased when big fireworks event are held on August 2. The average RTT has increased approximately 20 times, and the ratio of FIN-No-ACK termination and retransmission packet have increased 50% compared with the usual day. This result indicates that the communication traffic increased because many participants may send e-mails and/or use SNS for expressing impression of the fireworks to their friends or family.

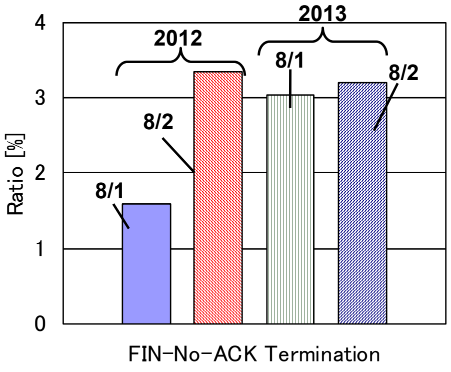
In Figure 6, it should be emphasized that there is the time when only either the RTT, the FIN-No-ACK termination or the retransmission packet increases. Therefore, it should be concluded that the network congestion can be effectively detected by analyzing the combination of the RTT, the FIN-No-ACK termination of TCP session and the retransmission packet.

B. Analysis of each Metric on the Event Date

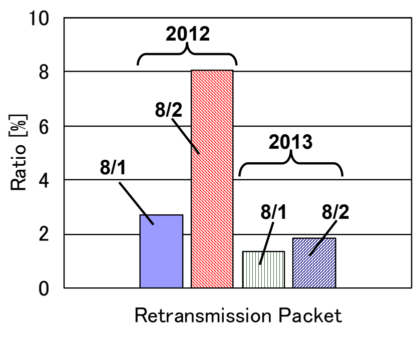
Figure 7 shows each metric during 19:30 to 21:00 on August 1 and 2 in 2012 and 2013. As a result, all metrics on August 2 became larger than that of August 1 in both years. Therefore, it is concluded that the selected metrics indicate a sign of the network congestion.



(a) Metric-1: RTT



(b) Metric-2: FIN-No-ACK termination



(c) Metric-3: Retransmission packet

Figure 7. Each metric measured on August 1(normal day) and 2(event day)

VI. Evaluation of Proposed CEP for Congestion Detection

A. Experimental Environment

In this section, effectiveness of our proposed system is clarified by evaluating whether or not the network congestion can be detected in real-time. As a performance measure, we adopt the processing time required for processing one-minute traffic log. Table 2 shows the evaluation environment. Here, in order to evaluate feasibility of the cloud-based system, the evaluation server was deployed on a local host, and virtual servers of an Amazon EC2 provided by AWS. The virtual server on Amazon EC2 provides the CPU capacity equivalent of a 2.0-2.4 GHz 2007 Opteron® or 2007 Xeon® processor [19].

TABLE 2. Evaluation Environment.

	Local host	Amazon EC2 (Micro Instance)
OS	Ubuntu 12.04 LTS	
CPU Clock	2.27GHz * 4	Equivalent of 2GHz
Memory	3.7GB	615MB

B. Block Diagram of CEP

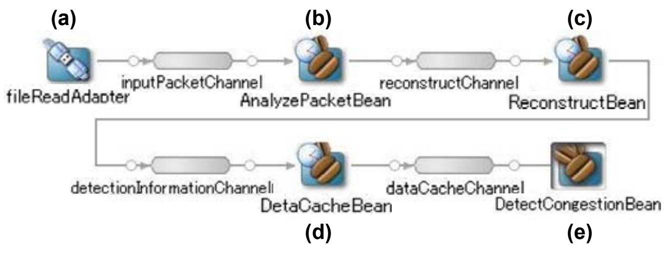


Figure 8. Overview of Block Diagram of CEP Application

Figure 8 represents an overview of block diagram of our developed CEP application. These application components are invoked when receiving events from others. Since the analyzed data traffic was already extracted in Nagaoka-city, a filtering process is omitted from this application. Procedures for detecting the network congestion are described below.

(a) The "fileReadAdapter" component reads PCAP files including traffic logs and sends each packet to the next component as an event.

(b) The "AnalyzePacketBean" extracts the necessary information for reconstructing TCP session from received packets and send that to the next component.

(c) The TCP session is reconstructed from the received events (packets) in the "ReconstructBean" component. When the RTT is calculated and a type of TCP session termination is detected, these metrics are sent to next component with the packet captured time.

(d) In "DataCacheBean", the received metric is cached for one minute in order to detect network congestion.

(e) The "DetectCongestionBean" component executes congestion detection as described in Section IV-E.

We have completed to implement the application which can detect only two metrics, the RTT and the ratio of FIN-No-ACK termination. In the future study, we will implement a detection method of the number of retransmission packets.

C. Performance Evaluation Results

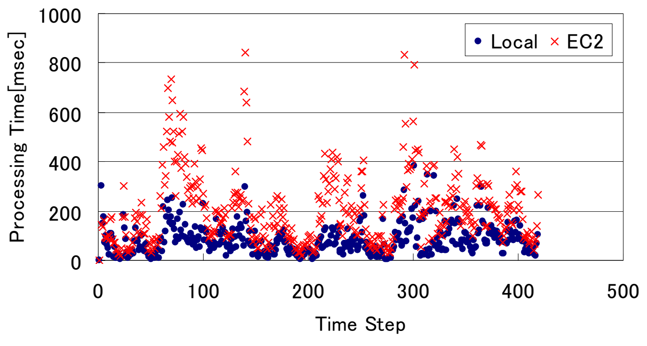


Figure 9. Time step of the processing time

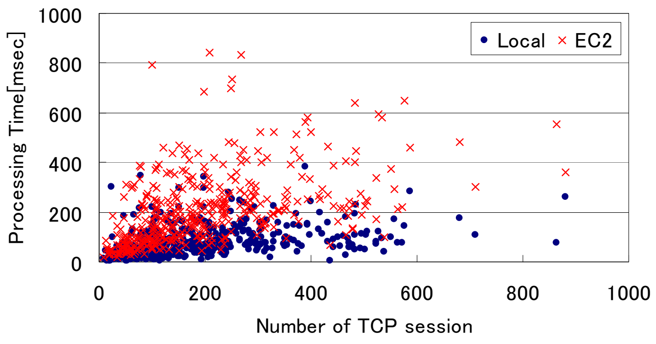


Figure 10. Relationship between number of TCP session and processing time

Figure 9 shows the time valuation of the processing time. As shown in this figure, the processing time on the Amazon EC2 is slower than that on the local host. Next, Figure 10 represents relationship between the number of TCP sessions extracted from the traffic log and the processing time. As shown in this figure, the processing time becomes large with the increase in the number of TCP sessions. However, it is observed that both systems can detect the network congestion in real-time because the processing time is kept within only 1 [sec].

Finally, we consider the processing power required to detect congestion of all areas in Japan. Here, the processing time does not exceed 1 [sec] when the system analyzed a one-minute traffic log captured in Nagaoka-city. The processing load of all areas in Japan may become 300 times larger than Nagaoka-city because the population of Japan is approximately 300 times as the population of Nagaoka Fireworks festival. Thereby, this system needs 5 minutes in order to detect the network congestion of all areas in Japan. Furthermore, it is envisaged that detection of retransmission packets needs further processing power. As a further study, we will distribute the processing load to multiple servers in order to improve the performance in the future.

VII. Conclusions and Future Work

In this study, a new network congestion detection system has been proposed, which analyzes a large amount of network traffic in real-time by using a CEP (Complex Event Processing) on the virtual server of Amazon EC2.

We have first evaluated whether the analysis of the RTT, the type of TCP session termination (FIN-No-ACK) and the number of retransmission packets are useful for detecting the congestion. As the analytical result, compared with the usual day, the average RTT has increased about 20 times, and the FIN-No-ACK termination and the download retransmission packet has increased 50% on the event day. Therefore, it has been concluded that the network congestion can be effectively detected by analyzing the combination of (1) the RTT, (2) the FIN-No-ACK termination of TCP session and (3) the number of retransmission packets.

Next, we have developed the proposed system by using a CEP on a local host and virtual servers of Amazon EC2. As a result of evaluation, the process of one-minute traffic log can be completed no later than 1 [sec], so it has been concluded that the proposed system can detect congestion in real-time.

In the future work, we will implement a distributed processing method of our proposed system in order to improve the performance.

References

[1] D. Luckham, "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems," May 2002.

[2] Nagaoka Fireworks Festival, http://nagaokamatsuri.com/index.html

[3] T. Takahashi, H. Yamamoto, N. Fukumoto, S. Ano, K. Yamazaki, "Congestion Detection in Mobile Network towards Complex Event Processing," Proceedings IEEE Annual International Computers, Software & Applications Conference 2013, pp.459-462, July 2013

[4] M. Thottan and C. Ji, "Anomaly detection in IP networks," IEEE Trans. Signal Processing, vol.51, no.8, pp. 2191– 2204 (2003).

[5] T. Kihara, N. Tateishi and S. Seto, "A Study on a Fault Detection Method with Relation Analysis of Network Data," The Institute of Electronics, Information and Communication Engineers Technical Report. Technical Committee on Information and Communication Management, vol.110, no.466, pp.17-22, March 2011.

[6] News Releases from NTT docomo: http://www.nttdocomo.co.jp/info/news_release/2011/06/14_00.html

[7] News Releases from KDDI: http://www.kddi.com/corporate/news_release/2013/0610a/

[8] Apache S4, http://incubator.apache.org/s4/

[9] Strom, http://storm-project.net/

[10] T. Imai, T. Ebiyama, K. Kida and K. Fujiyama and N. Nakamura, "A Web Access Analysis System Using Data-Stream Processing Technique," Forum on Information Technology, vol.8, no.2, pp.207-208, August 2009.

[11] K. Kida, K. Fujiyama, T. Imai and N. Nakamura, "Development and Evaluation of High Performance Floating Car Data System Based on Data-stream Processing," The Institute of Electronics, Information and Communication Engineers Technical Report. Intelligent Transport Systems, vol.108, no.200, pp.1-8, September 2008.

[12] K. Sato, "Sensor Data Processing System for Automotive Driving Environment Recognition," The Institute of Electronics, Information and Communication Engineers Technical Report. Data Engineering vol.110, no.107, pp.51-56, October, 2010.

[13] S. Kuwata, Y. Inaba, M. Yokogawa, T. Namatame and K. Nakagawa, "Stream Data Analysis Application for Customer Behavior with Complex Event Processing," The Institute of Electronics, Information and Communication Engineers Technical Report. Data Engineering, vol.110, no.107, pp.13-18, June 2010.

[14] T. Tanaka, "Evidence-Based Execution Realized by Algorithmic Trading," Journal of the Japanese Society for Artificial Intelligence, vol.24, no.3, pp. 376-384, May 2009.

[15] Oracle CEP, http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html

[16] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/

[17] Amazon Web Service , http;//aws.amazon.com/

[18] S. Toyoshima, S. Yamaguchi and M. Oguchi, "Development of middleware for data processing load distribution using cloud computing resource," The Institute of Electronics, Information and Communication Engineers Technical Report. Computer Systems, vol.110, no.167, pp.91-96 July 2010

[19] AWS documentation - EC2 User Guide, "Micro Instances," http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts_micro_instances.html

Tatsuya Takahashi received B.E. degree from Nagaoka University of Technology in '12. He is currently a graduate school student in Nagaoka University of Technology. His research interests include computer networks and network managemant.

Hiroshi Yamamoto received M.E. and D.E. degrees from Kyushu Institute of Technology, Iizuka, Japan in '03 and '06, respectively. From April '06 to March '10, he worked at FUJITSU LABORATORIES LTD., Kawasaki, Japan. Since April '10, he has been an Assistant Professor in the Department of Electrical Engineering, Nagaoka University of Technology. His research interests include computer networks, distributed applications, and networked services. He is a member of the IEEE.

Norihiro Fukumoto received B.E. and M.E. degrees in Information and Computer Science from Waseda University, Tokyo in '99 and '01, respectively. He joined KDDI R&D Laboratories, Inc. in '01, and has been engaged in research and development on speech application services and voice packetization system over IP networks. He is currently a research engineer of the Speech Processing Laboratory of KDDI R&D Laboratories Inc.

Shigehiro Ano received B.E. and M.E. degrees electronics and communication engineering from Waseda University, Japan in '87 and '89, respectively. Since joining KDD in '89, he has been engaged in the field of ATM switching system and ATM networking, His current research interests are traffic routing, control and management schemes over the next generation IP networks. He is currently the Senior Manager if Communications Network Planning Laboratory in KDDI R&D Laboratories Inc.

Katsuyuki Yamazaki received B.E. and D.E degrees from the University of Electro-communications and Kyushu Institute of Technology in '80 and '01, respectively. At KDD Co. Ltd., he had been engaged in R&D and international standardization of ISDN, S.S. No.7, ATM networks, L2 networks, IP networks, mobile and ubiquitous networks, etc., and was responsible for R&D strategy of KDDI R&D Labs. He is currently a Professor of Nagaoka University of Technology.