Packet Out-of-order and Retransmission in Statistics-based Traffic Analysis

Su-Kang Lee, Hyun-Min Ahn, and Myung-Sup Kim
Dept. of Computer and Information Science
Korea University
Sejong, Korea
{sukanglee, queen26, tmskim}@korea.ac.kr

***Abstract*—With the rapid growth of the Internet, the importance of application traffic analysis increases for efficient network management. The statistical information in traffic flows, can be efficiently utilized for application traffic identification. However, the packet out-of-order and retransmission generated at the traffic collection point reduce the performance of the statistics-based traffic analysis. In this paper, we propose a novel method to detect and resolve the packet out-of-order and retransmission problem in order to improve completeness and accuracy of the traffic identification. To prove the feasibility of the proposed method, we applied our method to a real traffic analysis system using statistical flow information, and compared the performance of the system with the selected 9 popular applications. The experiment showed maximum 4.9% of completeness growth in traffic bytes, which shows that the proposed method contributes to the analysis of heavy flow.**

Keywords—retransmission; out-of-order; statistic signature; network management; traffic analysis;

I. INTRODUCTION

With the high speed Internet growth, Internet-based services become more diverse, and the importance of network management is more emphasized. In this situation, the network manager must be able to Figure out the specific application of traffic for effective network management. Classification method with statistic signature [1] is one of the methods that detects and classifies specific applications.

Classification method with statistic signature uses statistical information in traffic flows such as packet size, transfer direction, collected time, and so on. However, collected packet's features have two limitations, Packet Out-of-order and Packet duplication caused by Retransmission. Such problems make it difficult for managers to classify applications.

Packet Out-of-order problem occurs when packets are transmitted through multiple paths, because the packet transmission speed of a path could be faster or slower than other paths. Therefore, packet sequence that is received by a recipient can differ from the original sequence sent by the sender. Similarly, the sequence of captured packets at a traffic collection point can differ from the original sequence at the sender. Out-of-order problem at a traffic collection point is the difference of packet sequence at a collection point from that of the sender.

The packet duplication problem at a collection point is caused by packet retransmission. If an error is detected in a received packet, a recipient requests a retransmission of the packet. If a sender did not receive a response within a specified time, the sender retransmits the packet. In this situation, a traffic collector which is located in the middle the path might store both of the original and retransmitted packets. Likewise, packets which have same sequence number are duplicated and stored at the traffic collection point. Such problem is referred to packet duplication problem caused by retransmission.

In this paper, we propose a novel algorithm to resolve the packet out-of-order and packet duplication problem at the traffic collect point. If such problems are resolved at the collection point, the same statistical flow information can be collected. Consequently, we can generate constant statistical signature of applications which is not affected by packet out-of order and duplication. The signature of the application enables us to detect traffic data of application correctly.

The remainder of this paper is organized as follows: we survey the existing research which classifies applications using statistical information in Section II. Then in Section III we define the problems that are packet out-of-order and packet duplication caused by retransmission at collection point. In Section IV, we suggest our proposed algorithm which resolves the packet out-of-order and packet duplication. In Section V, the experimental results and their analysis are explained. Finally, we give the conclusion and future work in Section VI.

II. RELATED WORK

Recently over the years, the application traffic classification using statistical characteristic of flow has been researched, most of which use Machine Learning (ML) algorithms. These methods apply flow characteristics such as port number, flow duration, inter-arrival time distribution, and packet size distribution to ML algorithm. These methods are not only good to analyze encrypted traffic data, but also free to the no privacy issue because they do not investigate packet payload. For this reason, these methods have additional advantage that they can classify applications faster with higher accuracy than other methods.

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2010-0020728), Brain Korea 21 Plus (BK21+), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2007483).

However, some of the past research [2,3] did not adapt to traffic analysis system working in real time because they had to use the statistical attributes extracted from an entire flow. To solve this problem, some researches [4,5] that extract attributes from the first N packets have been studied before, but they could not be used in a real time analysis on high-speed network link owing to calculation overhead and high computational complexity of machine learning algorithm. Furthermore, the most of the previous research could not provide detailed results because unit of traffic classifying is defined by protocol of applications. Therefore, it is difficult to apply network management and management policy which require classification of individual application units.

In previous works, the problem about change of statistics information caused by packet out-of-order generated in traffic acquisition point and packet duplication by retransmission did not be treated so far. According to this problem, we propose a novel algorithm to resolve the packet out-of-order packet duplication by retransmission.

III. PROBLEM DEFINITION

In order to define packet out-of-order problem generated at acquisition point and packet duplication by retransmission, we first describe the difference between packet out-of-order and packet duplication generated at end host and them generated at traffic acquisition point.

TCP provides reliable data transmission to applications by providing channel of stream between two end hosts. To ensure reliable data transmission between end hosts, TCP utilizes several fields such as sequence, acknowledge, and checksum in TCP header. It uses methods such as retransmission and flow control. ACK (Acknowledgement) is used to check whether data goes to TCP of destination. If a user who receives data from sender gets data without error, he sends ACK to sender again. Checksum is utilized to check any error in packet data. Sequence number is also used to check the orderly transmission of packets.

Packets transmitting between two end hosts send through various paths depending on situation of lines. In case of sending packets through various paths, receivers can get different order of packets when packet from sender transmits to receiver. This problem is out-of-order generated in end host. In order to solve this problem, receiving TCP has to check the sequence of packet and rearrange packet's order.

In Figure 1, host A sends a number of packets to host B, packets can be transmitted in different path with various reason. Also, packet receiving order at host A can be different from that at the sender. When generating this problem, TCP transport layer of host B conducts rearrangement to the original order. However, TCP transport layer of system storing collected traffic in traffic collection point does not conduct that kind of rearrangement. Hence, traffic stored in TCS (Traffic Collection Point) is different from packet order transmitted from host A. This problem is called packet out-of-order at collection point.

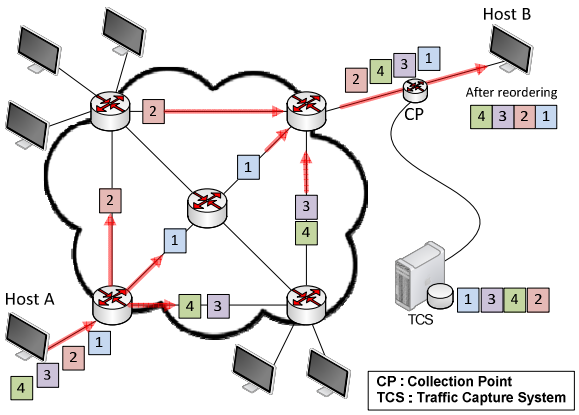


Fig. 1. The Packet out-of-order problem

Packet retransmission is generated in case of that packet from a sender does not be transmitted by any reasons or error is detected in received packet.

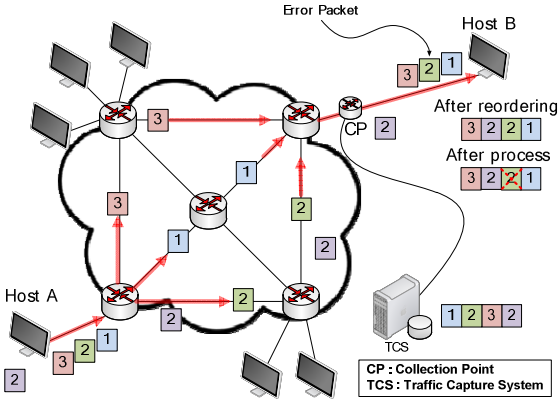


Fig. 2. The Packet retransmission problem

In that case the sender retransmits packet having same sequence number to the packet data not transmitted by error. Figure 2 describes the situation of retransmission packet when sending a number of packets from host A to host B. Host B receives 3 packets from host A. After TCP of host B detects error in second packet, it requests retransmission of second packet from host A. Host A retransmits second packet by retransmission request. Also, host B receives retransmitted packet and delete packet detecting error.

However, there is no such process in TCP transport layer of traffic capture system of collection point. Moreover, it collects original packet with error and retransmitted packet. This problem is packet duplication generating in collection point. Thus, packet having same sequence can be collected by packet retransmission in collection point. Normally, retransmitted packet is equal data and size to original packet.

Sometimes, TCP transmits the reassembled packets in the maximum MTU size in order to improve performance when TCP has additional data which must be sent to recipient. It is called packet repacketization. In case of retransmitted packet of repacketiztion, the problem will be generated such as Figure 3.

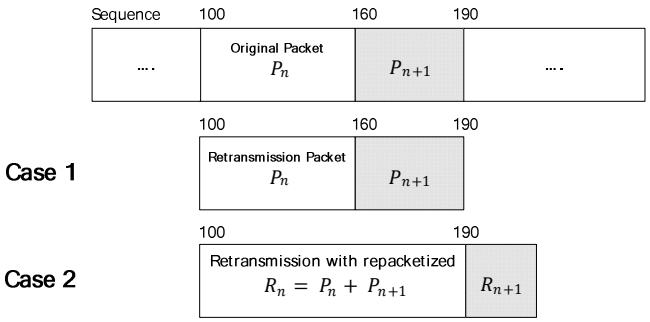


Fig. 3. Retransmission problem with repacketization

In case of transmitting repacketized packet, Figure 3 describe problem occurred in collection point. Original packet is delivered to receiver after getting through collection point. A receiver detects packet error and request retransmission of packet to sender. After receiving request in sender, packet will be sent again.

Case 1 is case sending retransmission packet having same data. Case 2 is case sending larger packet than original packet by repacketization. In case of case 1, it is same size between original packet and retransmission packet. Thus, sequence number of packet 1 and packet 2 is equal to 160. Accordingly, problem is solved if deleting retransmitted packet in collection point. However, Case 2 is the case sending larger packet than original packet by repacketization. Thus, sequence number of packet 1 and packet 3 is not equal to each other. Therefore, not only retransmission packet (P2) which is repacketized but also the next packet (P3) of repacketization packet must be removed in order to solve the retransmission problem completely.

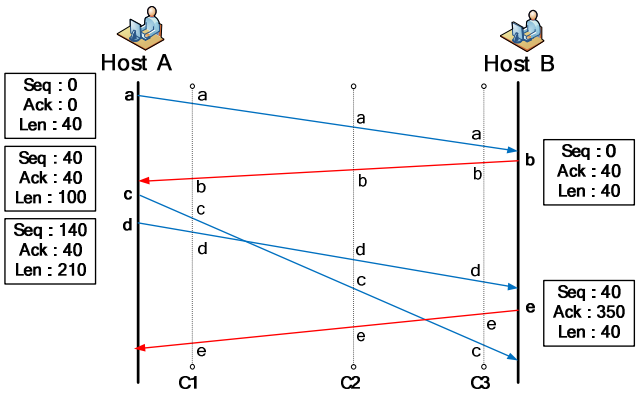
IV. SOLUTION

In this section, we propose a novel algorithm which detects the packet out-of-order and packet duplication problem.

A. Solution to the out-of-order problem

In previous research [5], we compared packets in the same direction only when occurring to out-of-order problem. Afterwards we have moved the position of the packet that has out-of-order problem. In this situation the sequence of packets may not be accurate. In order to find original sequence of packets that was generated by application we have to consider sequence of packets not only same direction but also reverse direction. Then original sequence of packets can be restored correctly.

Figure 4 indicates several collection points (C1, C2, C3) where occurred packet out-of-order problem. Host A and host B exchanged the packets such as sequence: a-b-c-d-e. On the other hand, the packets are passed through a different sequence such as: C1 is a-b-c-d-e, C2 is a-b-d-c-e, and C3 is a-b-d-e-c. This problem occurs because the collection point stores packets by passed order when passing through collection points. If the collection point does not have process to solve out-of-order problem, then the packets sequence being stored at collection point may differ from original sequence of packets.



Out-of-order problem at traffic collection pointsTable 1 is pseudo-code of algorithm which solve packet out-of-order problem) which consists of 5 steps. Input of the algorithm is Packet P(n) being collected in real-time process.

TABLE I. ALGORITHM FOR OUT-OF-ORDER PROBLEM

```
    Remove all non-payload packets from the packet sequence
    P(n) : n-th packet in a TCP flow
    P(n).seq : n-th packet's sequence number
    P(n).ack : n-th packet's acknowledge number
    P(n).dir : n-th packet's direction
    P(n).len : n-th packet's payload length
1   module Solution for the Out-of-order problem
2:  Input : P(n) in a TCP Flow
3:  find P(k) which P(k).dir == P(n).dir && biggest k in 0 <= k < n
4:  if(P(k).seq > P(n).seq)  // out-of-order detect
5:  {
6:     find P(i) which P(i).dir == P(n) && P(i).seq < P(n).seq
7:     find P(j) which P(j).dir == P(i) && smallest j in 0 <= j < k
8:     for each P(m) from P(j-1) to P(i+1)
9:     {
10:       if(P(m).dir != P(n).dir && P(m).ack == P(n).seq + P(n).len)
11:          put P(n) before P(m);    end module;
12:       if(P(m).dir != P(n).dir && P(m).ack == P(n).seq
13:          put P(n) after P(m);     end module;
14:    }
15:    put P(n) after P(i);
16: }
17: end module;
```

Step1: (Line3) Find P(k) which is same direction of P(n), and P(k) is the closest packet with P(n).

Step2: (Line4) Compare sequence value of P(n) with sequence value of P(k. If sequence value of P(k) is larger than sequence value of P(n) then the module detects out-of-order problem.

Step3: (Line6) In order to find correct location of P(n), the module find P(i) and P(j). P(i)'s direction is the same as P(n)'s and the biggest packet which is smaller than P(n)'s sequence value. Afterwards Find P(j) being located between P(i) with

P(k). P(j)'s direction is the same as P(n)'s and the smallest packet which is bigger than P(n)'s sequence value.

Step4-1: (Line10) If the module finds the packet P(m) where located between P(i) and P(j) that satisfies 'line 10' condition, P(n) is located before P(m).

Step4-2: (Line12) If the module finds the packet P(m) where located between P(i) and P(j) that satisfied 'line 12' condition, P(n) is located after P(m).

Step5: (Line15) If the module does not find the packet P(m) where located between P(i) and P(j) that satisfied 'line 12' condition, P(n) is located after P(i).

B. Solution to the retransmission problem

Packet duplication problem caused by retransmission is detected by comparing to sequence and directions. In previous research, we deleted the packet which is the original packet when occurring to packet retransmission, and we saved the packet which is retransmission packet. But our purpose is finding original feature of packets that was generated by application. Therefore, we have to store the packet which is original packet. Also we consider repacketized packet. The repacketized packet has features that are same direction, same value of sequence, and different length from original packet.

1) Common Retransmission Problem

TCP requests retransmission packet to opponent when TCP finds error from transmitted packet or does not receive a response packet within defined time. The retransmitted packet and original packet have the same packet sequence number and same packet direction.

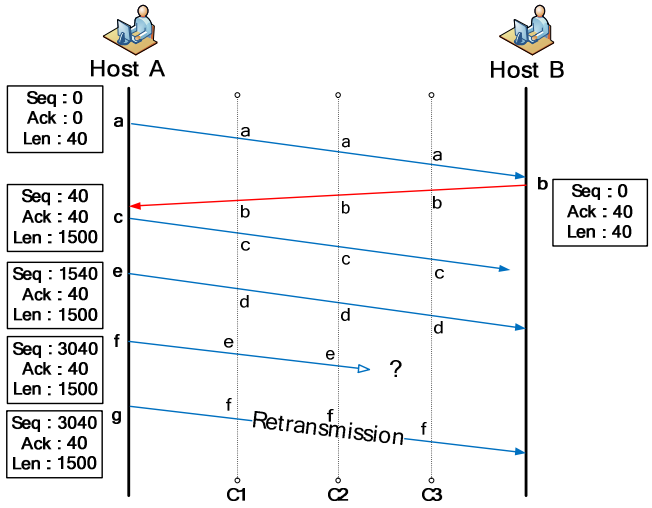


Fig. 4. Common Retransmission Problem at Traffic Collection Points

Figure 5 indicates common retransmission problem. Host A and host B exchanged the packets such as sequence: a-b-c-d-e-f-g-h, and packets are stored at collection point such as: C1, C2 are a-b-c-d-e-f, and C3 is a-b-c-d-f. We can find the 'packet f' which is retransmitted.

2) Repacketized Retransmission Problem

TCP requests retransmission packet to opponent when TCP finds error from transmitted packet or does not receive a response packet within defined time. At this time, opponent TCP can transmit the reassembled packets to the maximum MTU size in order to improve performance when TCP has data which must be sent to recipient. It is called packet of repacketized retransmission problem. In case of repacketized packet, next packet sequence number differs from originals. Therefore, we must delete not only retransmitted packet which is repacketized but also the next packet of repacketized packet.

Figure 6 indicates repacketized retransmission problem. Host A and host B exchanged the packets such as sequence: a-b-c-d-e-f-g-h-i. 'Packet d' and 'packet h' are same sequence value, same direction, and different length. Therefore, 'Packet h' is repacketized retransmission packets.

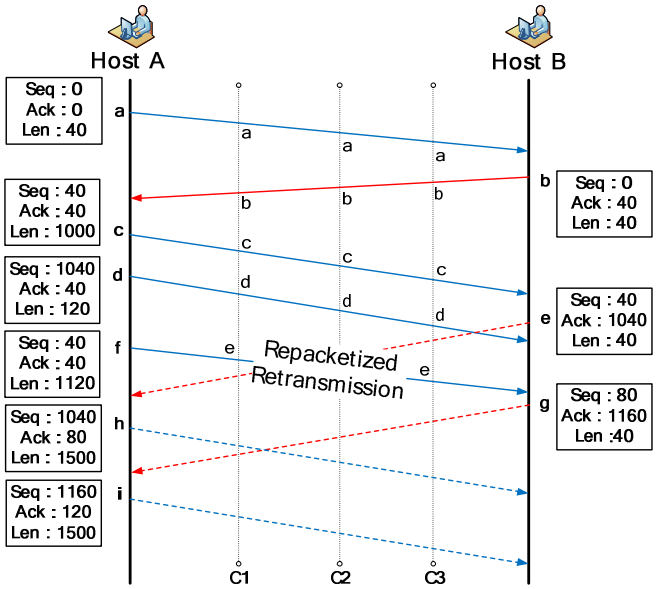


Fig. 5. Repacketized Retransmission Problem at Traffic Collection Points

Table 2 is pseudo-code of algorithm which solve packet duplication problem caused by retransmission, which consists of 2 steps. Input of the algorithm is Packet P(n) collected in real-time.

TABLE II. ALGORITHM FOR RETRANSMISSION PROBLEM

```
Remove all non-payload packets from the packet sequence
P(n) : n-th packet in a TCP flow
P(n).seq : n-th packet's sequence number
P(n).ack : n-th packet's acknowledge number
P(n).dir : n-th packet's direction
P(n).len : n-th packet's payload length
1:  module Solution for the Retransmission problem
2:  Input : P(n) in a TCP Flow
3:  find P(k) which P(k).dir == P(n).dir && biggest k in 0<= k < n;
4:  if( P(k).seq == P(n).seq ) // Retransmission
5:      Delete P(n);
6:  else if( P(k).seq + P(k).len !=P(n).seq )
7:      Delete P(n);
8:  end module;
```

Step1: (Line3) Find P(k) which is same direction of P(n), and P(k) is the closest packet with P(n).

Step2-1: (Line4) Compare sequence value of P(n) with sequence of P(k). If sequence value of P(k) is same sequence of P(n), then the module detects retransmission problem, deletes P(n).

Step2-2: (Line6) If the packet is satisfied 'line 6' condition, the module deletes P(n).

V. EXPERIMENT

This section explains about the result of the analysis with the experimental conditions for the test. We tested two kinds of experiments. First, we have performed experiments the Out-of-order and Retransmission problem in order to verify the performance of the algorithm. Second, we applied the algorithm to the traffic analysis system.

A. Signature Extraction System Based on Statistical Features

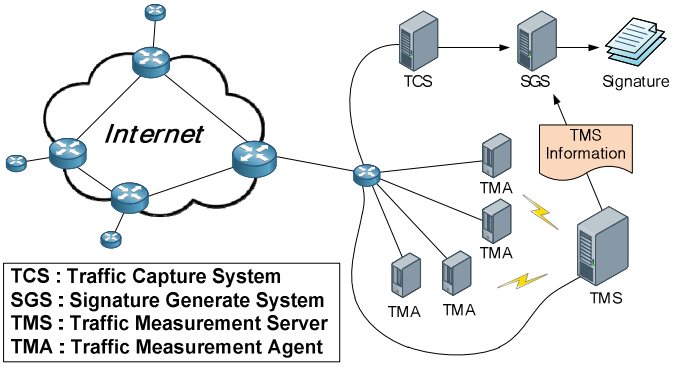


Fig. 6. Statistical Characteristic Signature Extraction System

The environment for the signature extraction system is organized like Figure 7 that collects the packets from the top router linked between the networks: campus and outside Internet. The packets are stored at TCS (Traffic Capture System). TMS (Traffic Measurement Server) generates Ground Truth[7] which use TMA log (Traffic Measurement Agent). Finally, SGS (Signature Generation System) extract statistical signature which use. The method of generating the Ground Truth provides reliability higher than other methods [8]. Packet out-of-order, duplication problem must be solved in TCS before extracting statistic signature.

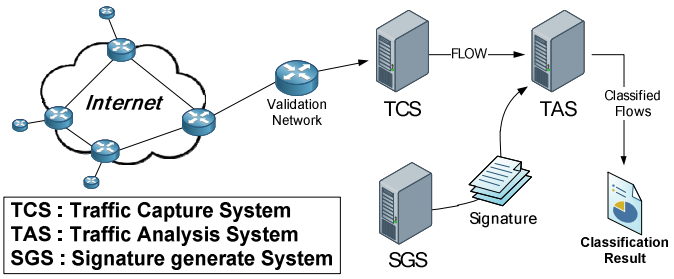


Fig. 7. Traffic Classification System Configuration

The environment for the traffic classification system is organized like Figure 8 that collects the packets from the border router linked between the network in campus and outside Internet, and the traffic data is stored at TCS (Traffic Capture System). TAS (Traffic Analysis System) is classified the traffic data by each application unit using signatures that were generated at SGS. The traffic data that has been stored in TCS must be solved out-of-order and retransmission problem by method that had been proposed in this paper before being used as input TAS.

B. Experimental result

In this section, we discuss two kinds of experiments. The first experiment has been shown performance of algorithm which resolves the packet out-of-order problem and the packet duplication problem caused by retransmission. The second experiment applied the algorithm to traffic analysis system in order to ensure that the performance is improved, and we compare the result of experiment after the problem resolved with the result of experiment before the problem resolved.

Traffic classification system [6] uses statistical features of flows. The features are the first five packets in a flow such as transmission direction of packets, sequence of packets and length of payload. These features are represented by the 5-dimensional flow vector, and the vectors are grouped according to the same application. The traffic classification system which uses the flow vector as a statistical signature classifies specific application.

TABLE III. OVERALL ACCURACY AND COMPLETENESS

	Accuracy			Completeness		
	Flow	**Packet**	**Byte**	**Flow**	**Packet**	**Byte**
Before application	99.80%	99.11%	99.40%	52.76%	40.04%	38.98%
After application	99.82%	99.20%	99.40%	52.01%	40.30%	39.29%

Table 3, 4 is result of experiment, completeness and accuracy applying proposed algorithm. Packet out-of-order problem occupies 0.03% in TCP flow, 0.02% in packet, and 0.02% in bytes. Also, packet retransmission problem occupies 23.27% in flow of TCP, 25.52% in packet, and 26.33% in bytes. Packet repacketization problem occupies 2.1% in flow, 3.47% in packet and 2.91% in bytes.

TABLE IV. OUT-OF-ORDER, RETRANSMISSION, REPACKETIZATION RATE

state	Flow		Packet		Byte	
	#	%	#	%	GB	%
Normal	118,657	74.68	64,781K	70.97	59.35	70.72
Out-of-order	50	0.03	28K	0.02	0.02	0.02
Retransmission	36,974	23.27	23,295K	25.52	22.1	26.33
Repacketization	3,186	2.1	3,169K	3.47	2.45	2.91
TCP Total	158,867		91,275K		83.92	

Table 5 shows traffic analysis rate before and after application of proposed method. Most of applications increase analysis rate in flow, packet and bytes respectively. Analysis rate of Outlook, XShell decreases in flow, packet and bytes

respectively. As the result of investigation into reason of decreasing flow analysis rate, signature is created through statistics information of retransmission packets although much retransmission packet is generated. After processing these retransmission flows by utilizing proposed algorithm to solve packet duplication problem, we identify affecting to created signature and decrease of analysis result.

We investigated applications which showed that analysis rate decreased. In Result of investigation, most flows of application have consisted of retransmitted packets only. The retransmitted packets are removed by the module after resolving the problem of retransmission, so there is only one packet. Because of this, these flows are not analyzed by TAS.

Despite of the reduction in the amount of flow analyzed, amount of packet, byte analysis was significantly increased. This result shows that the proposed method contributes to the analysis of heavy flow.

Consequently, by applying algorithm proposed in this paper, TAS (Traffic Analysis System) was able to efficiently analyze applications that generate heavy traffic flows.

VI. CONCLUSION

In this paper, we have addressed the problems occurred at the traffic collection point by TCP out-of-order and retransmission. We have proposed a novel algorithm to detect these problems and reorder the sequence of packets in a flow. The experimental result showed a significant improvement in detection and analysis of application traffic about 4 % by applying the proposed algorithm to a statistical signature based analysis system. While the amount of uTorrent traffic in flow has been decreased by 8000, total amount of analysis in byte has been increased by 1Gbytes. This shows that the amount of heavy flow analysis, which affects network resources, has been increased by the proposed algorithm.

In the future, the research in this paper is expected to improve the analysis rate and figure out the reason why the analysis rate has gone down.

TABLE V. RESULT OF EXPERIMENT BEFORE RESOLVE PROBLEMS / AFTER RESOLVE PROBLEMS

Application	Total			Analyzed					
	Flow	Packet	Byte	Flow		Packet		Byte	
				before application	after application	before application	after application	before application	after application
skype	1,141	22K	8,593K	751	794	15K	16K	6,978K	7,400K
naverlive	1,218	22,421K	18,159,291K	1,015	1,026	20,049K	20,395K	16,246,821K	16,537,135K
gomTV	981	826K	810,302K	701	702	745K	746K	732,854K	733,415K
xshell	403	64K	8,333K	389	388	63K	62K	8,130K	8,107K
teamviewer	485	231K	74,359K	426	431	196K	197K	62,835K	62,875K
nateon	299	103K	12,119K	272	273	102K	102K	11,682K	11,684K
dropbox	4,642	124K	66,631K	4,599	4,612	122K	123K	65,505K	65,774K
putty	266	27K	4,115K	261	262	27K	27K	4,080K	4,096K
outlook	4,872	496K	261,193K	3,515	3,420	324K	321K	154,314K	154,029K
uTorrent	417,724	422,910K	374,298,568K	216,016	208,475	157,435K	158,246K	136,173,646K	137,105,776K
Total	432,031	447,224K	393,703,504K	227,945	220,383	179,078K	180,235K	153,466,845K	154,690,291K

REFERENCES

[1] Young-Tae Han and Hong-Shik Park, "GameTraffic Classification Using Statistical Characteristics at the Transport Layer," ETRIJournal, Vol.32, No.1, Feb., 2010, pp.22-32.

[2] Y. Jin, N. Duffield, J. Erman, P. Haffner, S.Sen, and Z. L. Zhang, "A Modular MachineLearning System for Flow- Level TrafficClassification in Large Networks," ACMTransactions on Knowledge Discovery from Data,vol. 6, pp. 1- 34, 2012.

[3] J. Tan, X. Chen, and M. Du, "An InternetTraffic IdentificationApproach Based on GA andPSO-SVM," Journal of Computers, vol. 7, pp.19-29, 2012.

[4] C. Yin, S. Li, and Q. Li, "Network traffic classification via HMM under the guidance of syntactic structure," Computer Networks, vol. 56, pp. 1814-1825, April 2012.

[5] H. M. An, J. H. Choi, J. H. Ham, M. S. Kim. "A Method to resolve the Limit of Traffic Classification caused by Abnormal TCP Session", KNOM Review Vol. 15, No.1, Dec. 2012, pp. 31-39

[6] J. W. Park, M. S. Kim "Performance Improvement of the Statistic Signature based Traffic Identification System", KIPS Journal, Vol. 18-C, no. 4, Aug 2011.

[7] B. C. Park, Y. J. Won, M. S. Kim, and J. W. Hong, "Towards automated application signature generation for traffic identification," in Proc. Of IEEE Network Operations and Management Symposium (NOMS), pp. 160-167, 2008.

[8] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso, and K. C. Claffy, "GT: picking up the truth from the ground for internet traffic" ACM SIGCOMM Computer Communication Review, vol. 39, pp. 12-18, 2009.