Flow-aggregation Accelerating Strategy for TCP Traffic

Xiaoguo Zhang [1, 2] and Wei Ding [1]
1. School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
2. Information Engineering School, Henan University of Science and Technology, Luoyang 471023, China
Email: {xgzhang, wding}@njnet.edu.cn

***Abstract*—A great number of researches on network flow characteristics show a large proportion of the network flows are single-packet flows. However, almost all existing flow termination strategies have no optimization for single-packet flows, so the efficiency of flow-aggregation is lower. Based on in-depth study of flow characteristics and TCP protocol specifications, we find the packet status, packet arrival interval and SYN packet size can identify single-packet flows accurately, and then propose a flow-aggregation accelerating strategy for TCP traffic that aims to quickly identify single-packet flows. We build efficiency model and accuracy model to compare our strategy performance with others and make a lot of experiments on the traces collected from a main channel in the CERNET during the latest five years. The results prove our strategy can greatly improve the efficiency of flow-aggregation at the cost of very little loss of accuracy.**

***Index Terms*—Single-Packet Flow; Flow Timeout; Optimization; Flow Identification; Flow Characteristics; TCP; Accelerating**

I. INTRODUCTION

With the development of the Internet, network traffics grow rapidly and network behaviors become increasingly complicated. Network traffic analysis based on packet has not met the needs of network management. Instead, the traffic analysis based on flow is widely used in network security management, performance management, accounting management, traffic classification, topology analysis, and so on [1-5]. As network flow technology is widely applied, efficiency and accuracy of flow-aggregation become more and more important; under high accuracy to improve the efficiency of flow-aggregation is always one of the most important research issues of network flow technology.

Flow is an abstraction of network traffic. A flow is defined as a stream of packets that meet certain flow specification and termination constraints [6-14]. Flow specification only specifies the composition of a flow, while flow termination strategy, e.g. flow timeout, directly influences the efficiency and accuracy of flow-aggregation. For avoidance of ambiguity, in this paper flow-aggregation refers to a process that IP packets are matched into flow records. In a general way, accuracy is the opposite of efficiency; accuracy improves with the efficiency decreases. In essence, flow termination strategy is aimed to look for a balance between efficiency and accuracy of flow-aggregation. Although there are many excellent researches, all of them fail to achieve the best balance between efficiency and accuracy [6, 11, 12].

A lot of researches on flow characteristics indicate that a large proportion of network flows are single-packet flows in various networks. For the convenience of presentation, single-packet flow is denoted as SPF. Based on five-tuple specification and 64 seconds fixed timeout strategy, the SPF proportion is from 20% to 40% [6, 11, 13, 14], and the latest research on CERNET and CAIDA traffics shows the proportion is between 30% and 60% [15]. As we know, the duration of SPF is zero second, but it is not exported from memory until the end of timeout during flow-aggregation, so a lot of memory and computing resources are wasted on SPFs. If we can export SPFs in time, the efficiency of flow-aggregation will be improved greatly. However, flow timeout optimization strategy on SPF is very few up to now.

This paper focuses on flow termination strategy optimization of SPF, aims to increase efficiency as much as possible under the premise of high accuracy. For a long time, TCP traffic has occupied the dominant proportion of network traffic in terms of packets and bytes [15, 16-18]. Meanwhile, TCP is a connection-oriented protocol, has strict connection established, data exchange and connection release processes, and TCP packet can provide abundant transmission status information [19]. So, in this paper we only study flow termination strategy optimization for TCP traffic, while the optimization for UDP and other protocols will be researched in the future. Based on in-depth analysis of TCP SPF characteristics and the existing flow termination strategies, we propose a Flow-aggregation Accelerating Strategy for TCP traffic, denoted FAST. The main contributions of our work are as follows:

Firstly, we research TCP SPF characteristics deeply based on CERNET backbone network; find the packet status, packet arrival interval (denoted PAI) and the SYN packet size are high discerning features that can distinguish SPFs from the other flows. At the same time,

Manuscript received June 16, 2013; accepted November 4, 2013.
This work was supported by the National Key Technology R&D Program of China under Grant No. 2008BAH37B04 and the National Basic Research Program of China under Grant No. 2009CB320505.
Corresponding author email: xgzhang@njnet.edu.cn.

doi:10.4304/jnw.9.6.1416-1425

more than 95% of TCP SPFs only have SYN, SYN+ACK and RST+ACK three statuses.

Next, we propose a flow-aggregation accelerating strategy based on TCP protocol, which can fast identify and export SPFs. If it is used to optimize the existing flow termination strategies, under normal circumstances, about 50% of the memory space and computing resources will be saved.

The remainder of this paper is organized as follows. Section II deeply analyzes the existing flow termination strategies. Section III analyzes the characteristics of SPFs in detail. Section IV describes our flow-aggregation accelerating strategy, then build time cost model, space cost model and accuracy model to compare our strategy with the others in efficiency and accuracy. Finally, we conclude the paper in section V.

II. Related Work

The existing flow termination strategies can be classified as three categories. The first one is timeout strategies based on PAI, when PAI greater than timeout threshold the flow is terminated, such as fixed timeout strategy [6]. The second one is the termination strategies based on protocol flags, such as flag termination strategy for TCP traffic [6, 11, 20, 21]. The third one is the forced termination strategies based on resource consumption, such as time or space forced termination strategies [20, 21]. The third one is mainly used in routers, aims to ensure efficiency of flow-aggregation at the cost of loss of accuracy. The second one is often used with other termination strategies. Up to now, researches on flow termination strategy have focused on the first category. According to setting mode of timeout threshold, the first category is divided into fixed timeout and self-adaptive timeout strategies.

Fixed timeout strategy (denoted FT) is the earliest flow timeout strategy [6]. It judges the termination of a flow by PAI. It sets a global timeout threshold, when PAI is greater than the threshold a flow is terminated. For its simpleness, it is widely used. However, FT has its inherent defects: (1) It treats all flows equally without utilizing differences and correlation of flow characteristics to improve efficiency of flow-aggregation. For example, the terminated short flows, especially the SPFs, should be thrown in time, but because their waiting time does not exceed the timeout threshold, they are hold in memory. So, a lot of system resources are wasted and system efficiency decreases. (2) Small timeout threshold can throw the terminated flows in time, but the slow flows will be cut off frequently and result in thrashing; although large timeout threshold can reduce the thrashing, lead to a waste of system resources. So, it only can achieve a balance between efficiency and accuracy.

Ryu et al. [11] proposed a self-adaptive timeout strategy named measurement-based binary exponential timeout, denoted MBET. MBET maintains an independent timeout threshold for every flow and decreases flow timeout threshold by binary exponential based on flow rate during the initial timeout threshold. This strategy has much higher comprehensive performance than FT. However, it also has some disadvantages: (1) This strategy is based on the stability of PAI of a flow, it is suitable for steady and increasing flow rates, not applicable for fluctuant and decreasing rates, such as FTP flows [12]. (2) Timeout is not adjusted to a suitable value until receiving sufficient packets, so the strategy is mainly suitable for large flows, not applicable for short flows; especially the SPFs. So, MBET is not optimized for short flows and SPFs.

Wang et al. [12] proposed a self-adaptive timeout strategy named probability-guaranteed adaptive timeout, denoted PGAT. PGAT can adjust timeout threshold automatically based on application type, flow size and guarantee probability during any present timeout threshold. It has high flexibility and overall performance. However, it still has some shortcomings: (1) Application type is judged by transmission port number, accuracy is very low [22-24], so it is poor in generality. (2) It focuses on integrality of long flows, does not optimize timeout thresholds of short flows, especially of SPFs.

FT, MBET and PGAT are the most typical flow timeout strategies. However, they do not optimize the timeout thresholds of short flows, especially of SPFs. Besides, there are also other typical strategies, such as two-level self-adaptive timeout, denoted TSAT [13], multiclass support vector machines, denoted MSVM [14] and dynamical timeout strategy, denoted DToS [25], and so on. Among these timeout strategies, only TSAT noticed SPFs.

TSAT uses a small timeout threshold to filter SPFs. If the first PAI of a flow is less than this threshold, the flow is identified as SPF. TSAT optimized the timeout threshold of SPFs and improved the system efficiency. However, it also has some disadvantages: (1) The global unified filtering threshold ignores the differences of flow characteristics, lack of pertinence and flexibility. (2) The filtering mechanism is only based on a single feature and its accuracy is easily affected by network environment. (3) A large error of SPF identification is introduced by filtering mechanism and system overall accuracy is reduced.

It is thus clear that SPF timeout optimizing is only in its infancy at present, it is necessary to research new optimizing mechanisms so as to shorten the retention time of SPFs in memory and improve the efficiency of flow-aggregation system.

III. Single-packet Flow Characteristics

Traditional flow termination strategies usually ignore the differences of flow characteristics and employ the unified strategy for all flows; therefore, some chances to improve flow-aggregation efficiency are certainly lost. Because this paper aims to optimize SPF timeout, this section is about to study the characteristics of SPF deeply based on actual traffics so as to find applicable optimizing strategies. All traces used in this paper were collected from the main channel in the CERNET with 1/4 flow sampling. This main channel covers over 100 universities and high schools; its bandwidth is 10 Gbps. Table I lists the basic information of 17 traces that are

TABLE I. TRACES BASIC INFORMATIONS

Trace ID	Date	Start time	Duration	Size	# of TCP flows	# of total flows	# of total packets
1	01/21/2013	23:55:05	1 hour	27.5 GB	22,579,919	28,103,055	433,990,135
2	10/25/2012	23:54:37	1 hour	56.5 GB	41,111,872	53,130,374	891,719,826
3	09/20/2012	23:55:05	1 hour	32.2 GB	33,434,012	37,001,072	508,351,233
4	07/24/2012	23:55:05	1 hour	33.6 GB	57,246,177	61,419,050	530,934,446
5	04/25/2012	23:55:05	1 hour	28.4 GB	32,500,885	38,148,248	447,702,206
6	03/19/2012	23:55:05	1 hour	31.3 GB	44,742,398	55,469,794	494,010,718
7	11/16/2011	23:55:05	1 hour	35.6 GB	58,528,335	64,307,813	564,643,640
8	04/17/2011	23:55:04	1 hour	24.5 GB	19,177,374	26,589,458	387,309,462
9	03/12/2011	00:00:02	1 hour	33.2 GB	33,922,537	42,741,669	524,715,732
10	01/16/2011	09:55:17	1 hour	35.4 GB	23,188,424	29,888,032	558,212,507
11	11/14/2010	09:55:16	1 hour	48.2 GB	36,631,177	45,425,719	761,759,609
12	09/11/2010	09:55:17	1 hour	41.2 GB	17,478,313	27,382,109	650,733,917
13	05/18/2010	09:55:16	1 hour	57.8 GB	19,759,581	31,963,415	912,877,912
14	03/28/2010	13:55:16	1 hour	54.1 GB	14,926,245	29,239,129	854,281,019
15	02/23/2010	13:55:17	1 hour	18.7 GB	6,797,344	11,690,538	296,068,512
16	12/17/2009	13:55:16	1 hour	54.9 GB	14,745,839	28,372,727	866,617,207
I	12/21/2012	23:55:05	24 hours	1,139.4 GB	1,354,253,155	1,688,574,369	17,992,157,088

used in the paper, where the flow number was calculated by 64 seconds FT.

For presentation purposes, we define flow, TCP flow, TCP flow status and TCP bidirectional flow as follows.

Definition 1. A flow is defined as a unidirectional stream of packets subject to a specification that all the packets have same five-tuple (source IP address, destination IP address, source port number, destination port number, layer 3 protocol type) and termination constraints.

Definition 2. A TCP flow is defined as a unidirectional flow between source endpoint and destination endpoint of a TCP connection.

Definition 3. TCP flow status refers to the flag field value of a packet received by the TCP flow at some point. For the convenience of presentation, TCP Flow Status is denoted as TFS.

Definition 4. A TCP bidirectional flow consists of a forward TCP flow and a reverse TCP flow of a TCP connection.

In this section, we employ 64 seconds FT to generate flow records based on trace I that lasted for 24 hours. Then we analyze the distributions of source port, destination port, size, TOS, TTL and PAI of TCP flows. Unfortunately, port, size, TOS and TTL are very low discerning features to distinguish SPFs from the other flows. Although PAI is a high discerning feature, if we want to ensure high accuracy of SPF identification, the PAI threshold is still large. So, we need to seek some high discerning features to identify SPFs. Because this paper focuses on TCP traffic, we will deeply analyze the TCP SPF characteristics.

TCP SPF distribution based on trace I demonstrates that around 70% of SPFs are TCP flows, the distribution plots TCP SPF proportion by the hour, the details as shown in Fig. 1. And the measurements based on trace 1~16 also verify this finding; average 62% of SPFs utilize TCP protocol.

Obviously, most of the SPFs are normally TCP SPFs. Meanwhile, as a reliable transmission control protocol, TCP has rigid connection establishment, data exchange and connection release specifications; and it can provide abundant transmission statuses informations; these are important for TCP SPF identification. TCP transmission statuses refer to flag field value of a packet, i.e. TCP flow status. Although TCP flag field can indicate 63 statuses, distribution of TCP SPF statuses based on trace I shows 98% of TCP SPFs only have SYN (13%), SYN+ACK (75%) and RST+ACK (10%) three statuses, the details as shown in Fig. 2. And the measurements based on trace 1~16 also show average 95% of SPFs are these three statuses.

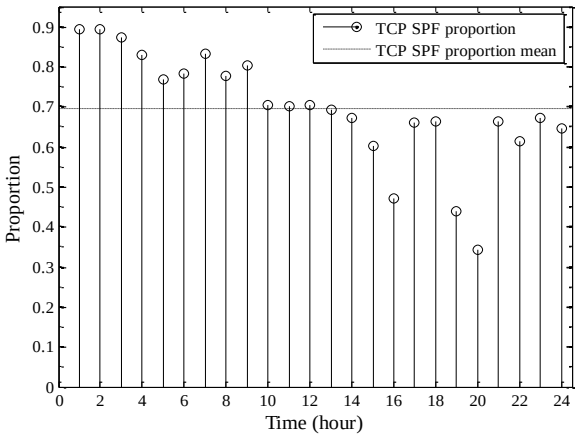


Figure 1. Distribution of TCP SPF proportion

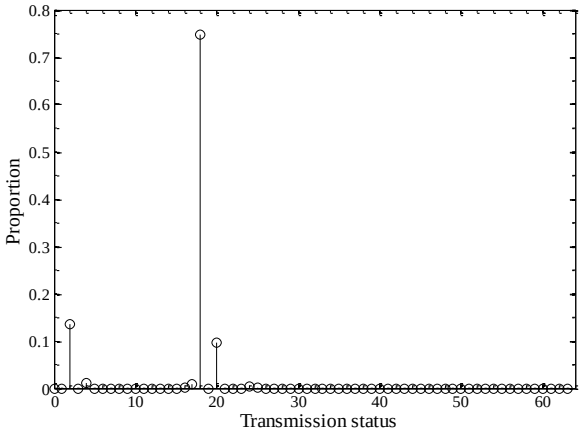


Figure 2. Distribution of TCP SPF transmission statuses

It is thus clear that, the vast majority of TCP SPFs only have SYN, SYN+ACK and RST+ACK three statuses. And the analysis based on entropy shows the TCP transmission status is a high discerning feature for TCP SPF identification. Therefore, this paper will research a quick identifying mechanism for TCP SPFs based on TCP transmission status to accelerate flow-aggregation process.

IV. FAST STRATEGY

As we know, TCP protocol specifies rigid connection establishment and termination processes. From the three-way handshake of TCP connection establishment and four-way handshake of connection termination can be concluded that an effective TCP connection contains at least 6 packets, and when timeout threshold is applicable, it contains two unidirectional flows (denoted uniflow) both packet number greater than 2. However, TCP SPF only has one packet; this illustrates that TCP SPF much more likely relates to an ineffective TCP connection or it is generated by an unreasonable timeout threshold that makes a TCP flow be shortened. Because we utilize the widely accepted 64 seconds timeout threshold, this eliminates the case that threshold is not reasonable. Therefore, we propose FAST based on effectiveness of TCP connection to identify TCP SPFs.

A. Principle

The basic starting point of FAST is the legality of flow status transition based on TCP protocol. When a new packet arrives, TCP transmission status of this packet is obtained and TCP flow status will change from previous status to this status, if the transition is legal, then flow status is changed as the new status, else the flow is terminated and exported. Because a SPF only has one packet, we only keep a watchful eye on the first two packets of a flow; once a flow receives the second packet legally, it is not a SPF and moved to TCP flow-aggregation module. This design not only can quickly identify SPFs, but also can save system resources, FAST working process as shown in Fig. 3.

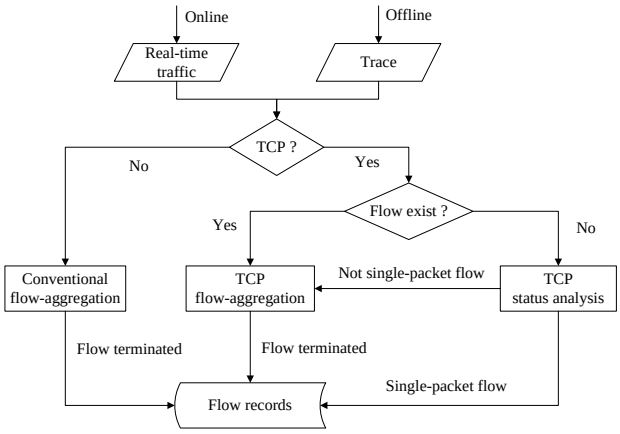


Figure 3. FAST working process

The vast majority of TCP SPFs only have SYN, SYN+ACK and RST+ACK three statuses; the other 60 statuses only occupy around 2% of TCP SPFs and they distribute evenly, the identifying cost of these 60 statuses SPFs is large but the improved system efficiency is very little. So, FAST only identifies the first three statuses SPFs to accelerate flow-aggregation process. In order to expound FAST more clearly, Table II describes the algorithm of FAST.

TABLE II. ALGORITHM OF FAST

FAST algorithm pseudo-code
1. PROCEDURE FAST(struct packet)
2. //FS stands for flow-aggregation space, //SAS stands for status analysis space.
3. IF(packet.protocol=6)
4. IF(packet belongs to a flow in FS)
5. update the flow record in FS
6. ELSE IF(packet belongs to a biflow in SAS)
7. BEGIN
8. update the biflow record in SAS;
9. divide the biflow record and move to FS
10. END
11. ELSE IF(packet belongs to a flow in SAS)
12. update and move the flow record to FS
13. ELSE CASE packet.tcpflag OF
14. 2:BEGIN
15. create a flow record for the packet in SAS;
16. IF(packet.pktLength=40)
17. choose time threshold in table VIII for the new flow
18. ELSE
19. choose time threshold in table III for the new flow
20. END

21. //match(): look for matched SYN for SYN+ACK.
//succeed in SAS, return 1; succeed in FS, return 2;
//otherwise, return 0.

22. 18: CASE match() OF
23. 0: create a flow record for the packet and export it to disk;
24. 1:BEGIN
25. create a biflow record for the packet in SAS;
26. copy matched SYN flow information to biflow in SAS;
27. delete the matched SYN flow in SAS
28. END
29. 2:create a flow record for the packet in FS
30. END
31. 20: create a flow record for the packet and export it to disk
32. ELSE create a flow record for the packet in FS
33. END
34. ELSE
35. Do flow-aggregation with usual timeout strategy in FS
36. END

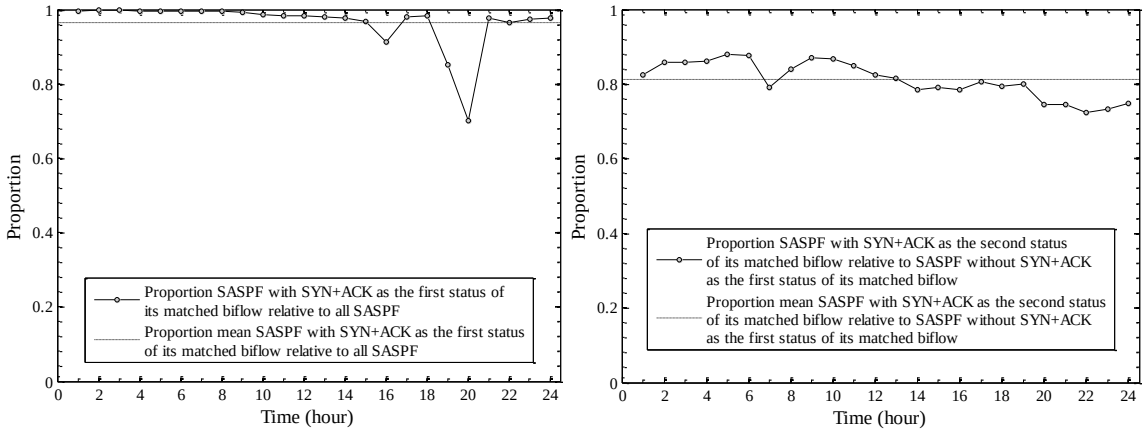
The core module of FAST is TCP status analysis module. For TCP status analysis, we need to focus on two issues:

Firstly, which status is the first legal TFS.

Secondly, what is the suitable holding time threshold of the first legal status, i.e., the longest waiting time of the first legal TFS transition.

If the first TFS of a flow is illegal, the flow is judged as a SPF and exported. If the first TFS of a flow is legal, the holding time threshold will be used to judge whether the flow is a SPF, if waiting time is greater than holding time threshold and the TFS is still not changed legally, then the flow is judged as SPF.

For an effective TCP uniflow, the first legal status only can be SYN or SYN+ACK status. However, for a TCP connection, SYN status is earlier than SYN+ACK status, this fact is not reflected by uniflow. In view of the bidirectional characteristic of TCP connection, we use



(a) SYN+ACK as the first status of its matched biflow (b) SYN+ACK as the second status of its matched biflow

Figure 4. Distribution of SYN+ACK SPF status

bidirectional flow to reflect the time sequence of TCP flow statuses. For the convenience of presentation, Bidirectional Flow is denoted as biflow. For an effective TCP biflow, the first legal status only can be SYN status.

For RST+ACK SPF, it can be identified directly; firstly RST+ACK is not the first legal flow status, furthermore RST is one of the flow termination flags in TCP flow termination strategies [6, 20, 21]. For SYN+ACK SPF, it can be identified by biflow; at first, SYN+ACK is not the first legal status of a biflow, secondly the researches based on trace I and trace 1~16 also show that the biflow with SYN+ACK status as the first status is a SPF. Fig. 4 describes the status distribution of SYN+ACK SPFs based on trace I. Fig. 4 (a) shows average 97% of SYN+ACK SPFs (denoted SASPF) can be identified by time sequence of statuses. Although, the remainder 3% cannot identified by time sequence of statuses, 81% of this 3% are SYN/SYN+ACK bidirectional SPFs that both forward and reverse flow are SPFs, the details as shown in Fig. 4 (b); they can be identified and exported in the course of SYN SPFs identification, so the final identifying ratio of SYN+ACK SPFs can exceed 99% and the identifying error is related to SYN SPF identifying error. Suppose that SYN SPF identifying error is 2%, SYN+ACK error is only 0.05% that can be ignored. The experiments based on trace 1~16 also verify the effectiveness of SYN+ACK SPF identification using time sequence of statuses.

For SYN status, because it is the legal first status of SPF or biflow, SYN SPF cannot be identified by time sequence of statuses. For SYN SPFs, this paper utilizes PAI and SYN packet size to identify them. Fig. 5 shows the distribution of SYN status holding time based on trace I, we will research the SYN status holding time threshold based on this distribution.

For the convenience of presentation, we define guaranteed probability, sample error and true error as follow.

Definition 5. For a certain time threshold, the probability that the status of a flow can transform successfully from SYN to another is called guaranteed probability that the flow is not a SPF.

Definition 6. For a certain time threshold, the probability that a flow of the available data sample is wrongly judged as SPF is called sample error and the error of all samples is called true error.

The sample error (denoted $error_S(h)$) of hypothesis h with respect to target function f and data sample S is:

$$error_S(h) = \frac{1}{n}\sum_{x\in S}\delta(f(x), h(x)) \tag{1}$$

where n is the number of examples in S, and the quantity $\delta(f(x),h(x))$ is 1 if $f(x)\neq h(x)$, and 0 otherwise.

The true error (denoted $error_D(h)$) of hypothesis h with respect to target function f and distribution D is:

$$error_D(h) = \Pr_{x\in D}[f(x) \neq h(x)] \tag{2}$$

Mitchell [26] proposed a general expression for approximate $N\%$ confidence intervals for $error_D(h)$ is:

$$error_D(h) = error_S(h) \pm Z_N\sqrt{\frac{error_S(h)(1-error_S(h))}{n}} \tag{3}$$

Here the constant Z_N is chosen depending on the desired confidence level.

Based on cumulative distribution in Fig. 5, we calculate the major thresholds of SYN holding time, the thresholds details as shown in Table III. Obviously, the vast majority of TCP flows have a very short SYN holding time. As we know, threshold is usually chosen depending on a requirement that is always the efficiency or accuracy. So, in practice, we should choose the most suitable threshold depending on the desired accuracy, so as to improve the system efficiency at the cost of loss of minimum accuracy.

TABLE III. TIME THRESHOLD AND ACCURACY

Time threshold	Guaranteed probability	Sample error	True error (99% confidence intervals)
1s	0.7896	0.182143	0.182143 ± 0.000052
2s	0.8161	0.165925	0.165925 ± 0.000050
4s	0.9408	0.042310	0.042310 ± 0.000027
8s	0.9567	0.032193	0.032193 ± 0.000024
16s	0.9787	0.015186	0.015186 ± 0.000016
32s	0.9960	0.002832	0.002832 ± 0.000007

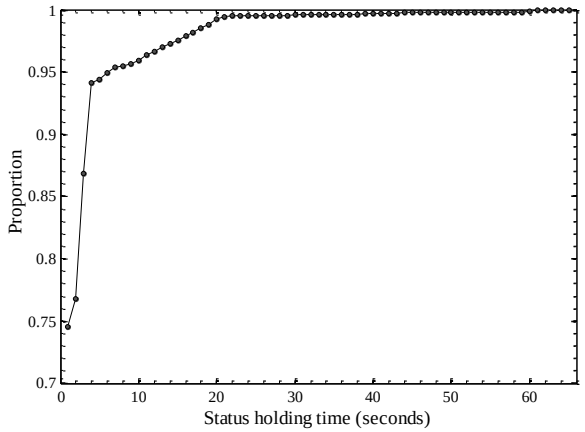


Figure 5. Cumulative distribution of SYN status holding time

However, if we identify SPF only according to SYN status holding time, the identifying accuracy is not ideal. In view of this, we deeply analyze the distributions of source port, destination port, size, TOS and TTL of SYN SPF based on trace I, where only the distribution of SYN SPF size has obvious regularity, the details as shown in Fig. 6. For the conveniences of description, a flow that its first packet is SYN packet and the packet size is 40 bytes is denoted as SYN40. If a SYN40 is a SPF, it is called SYN40 SPF, and otherwise it is called SYN40 MPF. SYN SPF size distribution shows that more than 65% of SYN SPFs are SYN40 SPFs, and the research of SYN40 based on trace I shows that the MPF proportion of SYN40s is less than 2%, this fact demonstrates that if we only use the first packet size to identify the SPFs, then the identifying accuracy of SYN40 SPFs can easily exceed 98%. The experiment results based on trace 1~16 also verify these facts, the details as shown in Table IV and Table V, where P_{S40SF} stands for the proportion SYN40 SPFs relative to SYN40s, P_{SSF40} stands for the proportion SYN40 SPFs relative to SYN SPFs.

In order to improve the identifying accuracy of SYN40 SPFs, we deeply study the PAI distribution of SYN40s based on trace I, the details as shown in Fig. 7. The distribution shows that the PAIs of more than 45% SYN40 MPFs are less than 2 seconds, and this fact is also verified by the experiment results based on trace 1~16, the details as shown in Table VI and Table VII, where P_{L2PAI} stands for the proportion the SYN40 MPFs which first PAIs are less than 2 seconds relative to all SYN40 MPFs. It is thus clear that, if the 2 seconds threshold is used to identify SYN40 SPFs, then the identifying error will be decreased 50%. Without loss of generality, we calculate the major holding time thresholds of SYN40 based on trace I, the details as shown in Table VIII.

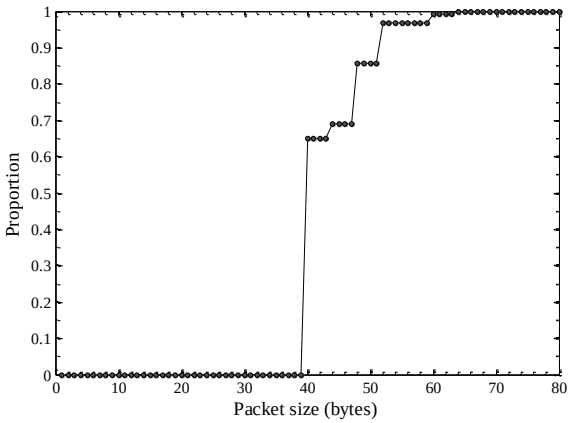


Figure 6. Cumulative distribution of SYN SPF size

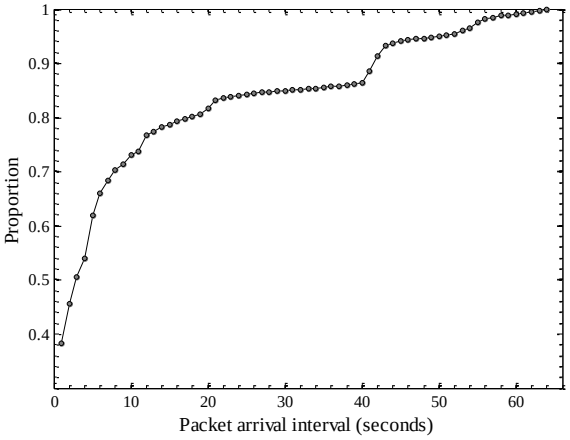


Figure 7. Cumulative distribution of SYN40 first PAI

TABLE IV. TRACE I AND TRACE 1~7 SYN40 FLOW CHARACTERISTICS

Trace ID	I	1	2	3	4	5	6	7
P_{S40SF}	0.9822	0.9936	0.9911	0.9936	0.9907	0.9917	0.9907	0.9790
P_{SSF40}	0.6503	0.6385	0.5294	0.7741	0.8278	0.6689	0.7735	0.8561

TABLE V. TRACE 8~16 SYN40 FLOW CHARACTERISTICS

Trace ID	8	9	10	11	12	13	14	15	16
P_{S40SF}	0.9832	0.9873	0.9736	0.9813	0.9900	0.9811	0.9901	0.9913	0.9901
P_{SSF40}	0.8751	0.8562	0.5517	0.5721	0.7075	0.4103	0.5842	0.6840	0.5076

TABLE VI. TRACE I AND TRACE 1~7 SYN40 FIRST PAI

Trace ID	I	1	2	3	4	5	6	7
P_{L2PAI}	0.4549	0.8624	0.8612	0.9749	0.7557	0.9288	0.9605	0.9547

TABLE VII. TRACE 8~16 SYN40 FIRST PAI

Trace ID	8	9	10	11	12	13	14	15	16
P_{L2PAI}	0.9823	0.8057	0.4582	0.8407	0.8724	0.9323	0.9208	0.8844	0.7621

TABLE VIII. SYN40 TIME THRESHOLD AND ACCURACY

Time threshold	Guaranteed probability	Sample error	True error (99% confidence intervals)
1s	0.3829	0.011007	0.011007 ±0.000032
2s	0.4549	0.009723	0.009723 ±0.000030
4s	0.5401	0.008203	0.008203 ±0.000028
8s	0.7033	0.005291	0.005291 ±0.000022
16s	0.7922	0.003706	0.003706 ±0.000019
32s	0.8509	0.002659	0.002659 ±0.000016

From the above analysis on SYN and SYN40 SPFs, this paper uses two sets of thresholds in the course of SYN SPF identification, if the first packet of a flow is SYN40, we choose the threshold in Table VIII according to the desired accuracy, and otherwise we choose the threshold in Table III.

TABLE IX. COST MODEL PARAMETERS

ID	Parameter	Parameter meaning
1	C_{SC}	Computing resources for creating a flow record in SAS
2	C_{FC}	Computing resources for creating a flow record in FS
3	C_{SS}	Computing resources for scanning a flow record in SAS
4	C_{FS}	Computing resources for scanning a flow record in FS
5	M_F	Memory resources for storing a flow record in FS
6	M_S	Memory resources for storing a flow record in SAS
7	α	Scanning frequency
8	μ	Proportion TCP SPFs relative to TCP flows
9	P_{SYN}	Proportion SYN SPFs relative to TCP SPFs
10	P_{SYN40}	Proportion SYN40 SPFs relative to SYN SPFs
11	P_{SSA}	Proportion SYN/SYN+ACK biflows relative to TCP SPFs
12	P_{S40SA}	Proportion SYN40/SYN+ACK biflows relative to SYN/SYN+ACK biflows
13	T_{SYN}	SYN status holding time threshold
14	T_{SYN40}	SYN status holding time threshold of SYN40
15	T_{SC}	Average waiting time of SYN status transition
16	T_F	Timeout threshold of FT
17	T_{DR}	Average duration of flows

B. Efficiency Evaluation

The cost of a strategy usually refers to the average computing resources and memory resources consumed by handling an input. For flow-aggregation, flow is the basic unit of a system, so the cost of FAST is the average computing resources and memory resources that are used to create and maintain a flow record. Because FAST only optimizes the flow timeout strategy for TCP traffic, the remainder of traffic is still aggregated by conventional methods; only the flow-aggregation cost of TCP traffic may be changed, and the costs of other traffics are not changed. So, we only need to build the cost models for TCP traffic. Table IX lists the main parameters and their meanings of our cost models. Although the typical adaptive timeout strategies, such as MBET, PGAT and so on, are more efficient than FT, they do not optimize the SPF termination strategy; for SPFs, these strategies have the same efficiency as FT. This means that for a same trace the cost saved by timeout optimizing of SPFs is equal for all strategies. As we know, the cost of FT is the largest among all strategies. So, cost reduction percent FAST relative to FT is less than FAST relative to other strategies. Therefore, if we want to compare efficiency of FAST with other strategies, we only need to compare with FT to get the minimum values.

For FAST, C_{FAST} and M_{FAST} are used to denote the average computing resources and memory resources that are used to handle a flow. Where C_{SYN} and M_{SYN} are used to denote the average computing resources and memory resources that are used to handle a SYN SPF, M_{SA} are used to denote the average memory resources that are used to handle a SYN+ACK SPF. Meanwhile, for FT, C_{FT} and M_{FT} are used to denote the average computing resources and memory resources that are used to handle a flow.

$$C_{SYN} = P_{SYN40}.(C_{SC} + \alpha.C_{SS}.T_{SYN40}) + (1-P_{SYN40})(C_{SC} + \alpha.C_{SS}.T_{SYN}) \quad (4)$$

$$C_{FAST} = \mu.(P_{SYN}.C_{SYN} + P_{SSA}.C_{SC}) + (1-\mu).[C_{SC} + \alpha.C_{SS}.T_{SC} + C_{FC} + \alpha.C_{FS}.(T_{DR} + T_F)] \quad (5)$$

$$M_{SA} = P_{S40SA}.M_S.T_{SYN40} + (1-P_{S40SA}).M_S.T_{SYN} \quad (6)$$

$$M_{SYN} = P_{SYN40}.M_S.T_{SYN40} + (1-P_{SYN40}).M_S.T_{SYN} \quad (7)$$

$$M_{FAST} = \mu.(P_{SYN}.M_{SYN} + P_{SSA}.M_{SA}) + (1-\mu).[M_S.T_{SC} + M_F.(T_{DR} + T_F)] \quad (8)$$

$$C_{FT} = \mu.(C_{FC} + \alpha.C_{FS}.T_F) + (1-\mu).[C_{FC} + \alpha.C_{FS}.(T_{DR} + T_F)] \quad (9)$$

$$M_{FT} = \mu.M_F.T_F + (1-\mu).M_F.(T_{DR} + T_F) \quad (10)$$

We use R_{CFF} and R_{MFF} to denote the cost reduction ratio FAST relative to FT of the average computing resources and memory resources that are used to handle a flow:

$$R_{CFF} = \frac{C_{FT} - C_{FAST}}{C_{FT}} \quad (11)$$

$$R_{MFF} = \frac{M_{FT} - M_{FAST}}{M_{FT}} \quad (12)$$

Because many parameters of R_{CFF} and R_{MFF} are closely related to network environment, this paper will calculate R_{CFF} and R_{MFF} based on trace 1~16. The flow-aggregation algorithm indicates M_S=M_F, C_{FC}=C_{SC}=3C_{FS}=3C_{SS}, If T_{SYN}=16s, T_{SYN40}=2s, T_F=64s, α=1Hz, we will get the values of R_{CFF} and R_{MFF} for trace 1~16, the details as shown in Table X and Table XI.

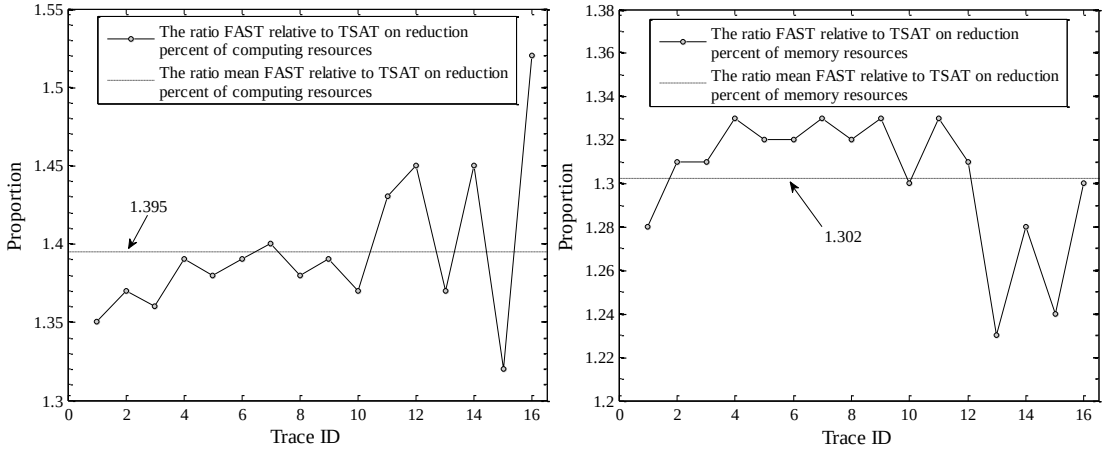
The experiment results show that FAST improves the efficiency of flow-aggregation greatly. Compared to MBET, PGAT, FT and other strategies, when SPF proportion is higher, average computing resources for

TABLE X. TRACE 1~8 COST REDUCTION PERCENT FAST RELATIVE TO FT

Trace ID	1	2	3	4	5	6	7	8
R_{CFF}	34.95%	55.64%	53.22%	77.18%	54.66%	57.33%	63.40%	44.49%
R_{MFF}	37.41%	57.17%	55.05%	78.02%	56.29%	58.96%	64.63%	46.78%

TABLE XI. TRACE 9~16 COST REDUCTION PERCENT FAST RELATIVE TO FT

Trace ID	9	10	11	12	13	14	15	16
R_{CFF}	61.79%	40.60%	34.21%	16.08%	8.93%	9.12%	14.37%	8.37%
R_{MFF}	63.14%	42.77%	36.39%	18.95%	12.02%	12.12%	17.24%	11.14%



(a) Reduction percent of computing resources (b) Reduction percent of memory resources

Figure 8. Efficiency comparison FAST relative to TSAT based on trace 1~16

handling a flow decrease around 52% and average memory resources decrease 54% approximately. Even if the SPF proportion is very low, the average cost reduction percent is also more than 11%. Meanwhile, when the number of SPFs increases rapidly, such as DDoS attack, the efficiency of MEBT, PGAT, FT and other strategies will decrease largely, and at worst, memory will be exhausted and system crashes. In this situation, FAST can ensure system works normally as much as possible.

In order to further evaluate the time and space efficiency of FAST, we compared FAST with TSAT that is the only strategy with filtering mechanism for SPFs up to now. TSAT uses 16 seconds timeout threshold to filter SPFs, if the first PAI of a flow is larger than 16 seconds, then the flow is identified as a SPF. For TSAT, C_{TSAT} and M_{TSAT} are used to denote the average computing resources and memory resources that are used to handle a flow. And the cost reduction percent TSAT relative to FT is denoted as R_{CTF} and R_{MTF}, the details as shown in (13)~(16). We calculate the cost reductions for TSAT based on trace 1~16 and compare with FAST. The results show the average computing resources reduction percent of FAST is around 1.40 times relative to TSAT and average memory reduction percent of FAST is about 1.30 times relative to TSAT, the details as shown in Fig. 8.

$$\begin{aligned} C_{TSAT} &= \mu.(C_{SC} + \alpha.C_{SS}.16) \\ &+(1-\mu).[C_{SC} + \alpha.C_{SS}.T_{SC} + C_{FC} + \alpha.C_{FS}.(T_{DR} + T_F)] \end{aligned} \quad (13)$$

$$M_{TSAT} = \mu.M_S.16 + (1-\mu).[M_S.T_{SC} + M_F.(T_{DR} + T_F)] \quad (14)$$

$$R_{CTF} = \frac{C_{FT} - C_{TSAT}}{C_{FT}} \quad (15)$$

$$R_{MTF} = \frac{M_{FT} - M_{TSAT}}{M_{FT}} \quad (16)$$

C. Accuracy Evaluation

The difference FAST relative to other strategies is the SPF timeout optimization for TCP traffic; the other traffic is still aggregated by conventional strategies and the flow-aggregation accuracy of that traffic is not changed. Therefore, when we evaluate the accuracy of FAST we only need to consider TCP traffic, and the accuracy of FAST is the TCP SPF identifying accuracy relative to benchmark. In many researches, the flow-aggregation result using FT with 64 seconds timeout is always viewed as a benchmark. So, we take the flow record set generated by FT with 64 seconds timeout as the true set, and we use thrashing and shortening to evaluate the accuracy of FAST.

Definition 7. For a trace, when a flow termination strategy is employed, the increased proportion the flow number generated by the strategy relative to true number contained in the trace is called thrashing, denoted R_{THRA}.

$$R_{THRA} = \frac{Num_{STRA} - Num_{TRUE}}{Num_{TRUE}} \quad (17)$$

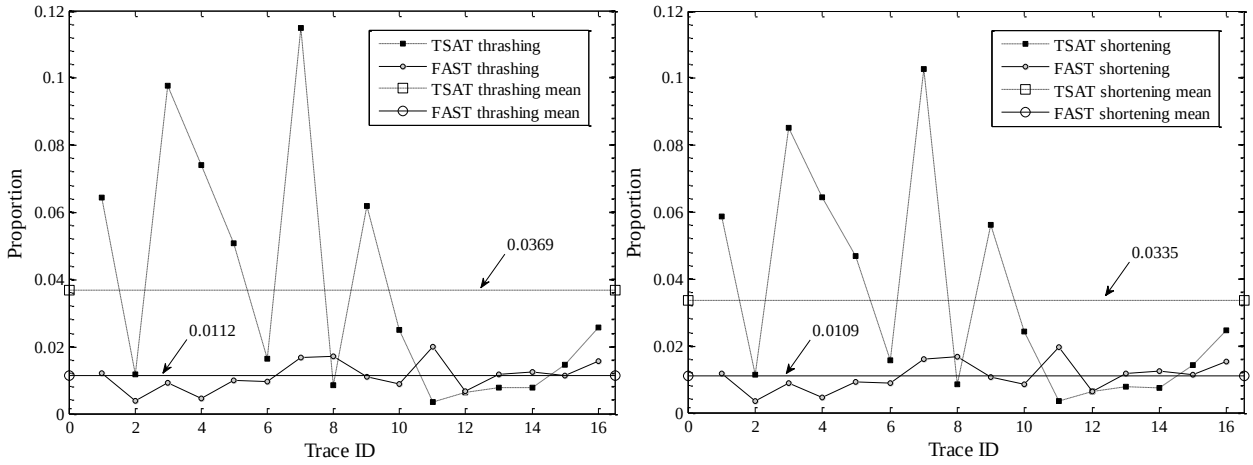
where Num_{STRA} denotes the number of generated flows by the strategy, and Num_{TRUE} denotes the true number of flows contained in the trace.

TABLE XII. ACCURACY VALUES OF TRACE 1~8

Trace ID	1	2	3	4	5	6	7	8
R_{THRA}	1.19%	0.36%	0.90%	0.45%	0.97%	0.96%	1.65%	1.72%
R_{SHOR}	1.15%	0.35%	0.87%	0.44%	0.92%	0.88%	1.59%	1.68%

TABLE XIII. ACCURACY VALUES OF TRACE 9~16

Trace ID	9	10	11	12	13	14	15	16
R_{THRA}	1.09%	0.86%	1.98%	0.67%	1.17%	1.24%	1.13%	1.57%
R_{SHOR}	1.06%	0.84%	1.96%	0.64%	1.15%	1.22%	1.12%	1.52%



(a) Thrashing (b) Shortening

Figure 9. Accuracy comparison FAST relative to TSAT of trace 1~16

Definition 8. For a trace, when a flow termination strategy is employed, the proportion the true flows contained in the trace are truncated is called shortening, denoted R_{SHOR}.

$$R_{SHOR} = \frac{Num_{TRUE} - Num_{SAME}}{Num_{TRUE}} \tag{18}$$

where Num_{SAME} denotes the number of the true flows contained in the trace that are not cut off when a flow termination strategy is employed.

We calculate the accuracy of FAST based on trace 1~16, and the values of R_{THRA} and R_{SHOR} are showed in Table XII and Table XIII. The experiment results show that both thrashing and shortening are only about 1%. Thus it can be seen that FAST only loses very tiny accuracy of flow-aggregation relative to benchmark.

In order to fully evaluate the accuracy of FAST, we calculate the thrashing and shortening of TSAT based on trace 1~16 and compare with FAST. The results show the average thrashing of TSAT is 3.29 times relative to FAST and its average fluctuation range is 8.96 times relative to FAST, the details as shown in Fig. 9 (a). Meanwhile, the average shortening of TSAT is 3.08 times relative to FAST and its average fluctuation range is 7.86 times relative to FAST, the details as shown in Fig. 9 (b). It is thus clear that both accuracy value and accuracy stability FAST is much better than TSAT.

V. CONCLUSIONS

Efficiency improvement of flow-aggregation is always an important research issue of flow-based network management. This paper aims to optimize SPF termination strategy. At first, we deeply research the characteristics of SPFs, find that packet statuses, PAI and SYN packet size are high discerning features that can identify SPFs accurately, and 95% of TCP SPFs only have SYN, SYN+ACK and RST+ACK three statuses in the measured network. Then, we propose FAST strategy and analyze its efficiency and accuracy based on 5 years traces from CERNET backbone. The results show FAST usually reduces more than 50% computing resources and memory resources relative to MEBT, PGAT, FT and other strategies, and the accuracy is only lost around 1%. Relative to the existing SPF filtering strategy (TSAT) FAST improves efficiency and accuracy of flow-aggregation much more. All these results demonstrate convincingly that FAST succeeds in optimizing SPF termination strategy. Meanwhile, in view of the advantage of SPF fast identification, FAST can ensure the system work normally as much as possible when network anomaly is raised, such as DDoS attack. And furthermore, FAST can be viewed as a filtering mechanism of SPFs; it can be easily transplanted to MEBT, PGAT, FT and other termination strategies, so as to optimize these strategies.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] B. Li, J. Springer, G. Bebis, and H. G. Mehmet, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 567-581, March 2013.

[2] C. Gates, M. Collins, M. Duggan, A. Kompanek, and M. Thomas, "More Netflow Tools: for Performance and Security," in *Proc. 18th USENIX conference on System administration*, Atlanta, 2004, pp. 121-132.

[3] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *Proc. ACM SIGCOMM 2002 Conference*, Pittsburgh, 2002, pp. 323-336.

[4] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Network*, vol. 26, no. 1, pp. 35-40, January-February 2012.

[5] X. Wu, K. Yu, and X. Wang, "On the growth of Internet application flows: A complex network perspective," in *Proc. IEEE INFOCOM 2011*, Shanghai, 2011, pp. 2096-2104.

[6] K. C. Claffy, H. W. Braun, and G C.Polyzosg, "A Parameterizable Methodology for Internet Traffic Flow Profiling," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1481-1494, Oct 1995.

[7] R. Jain and S. A. Routhier, "Packet Trains-Measurement and a New Model for Computer Network Traffic," *IEEE Journal on Selected Areas in Communications*, vol. 4, no. 6, pp. 986-995, Sep 1986.

[8] D. D. Clark, "The design philosophy of the DARPA Internet protocols," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 106-114, Jan 1995.

[9] B. Claise, G. Sadasivan, V. Valluri and M. Djernaes, "Cisco Systems NetFlow Services Export Version 9," IETF RFC 3954, 2004, http://www.rfc-editor.org/rfc/rfc3954.txt.

[10] P. Phaal, S. Panchen and N. Mckee, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks," IETF RFC 3176, 2001, http://www.rfc-editor.org/rfc/rfc3176.txt.

[11] B. Ryu, D. Cheney and H. W. Braun, "Internet Flow Characterization: Adaptive Timeout Strategy and Statistical Modeling," in *Proc. Passive and Active Measurement workshop*, Amsterdam, 2001, pp. 94-105.

[12] J. Wang, L. Li, F. Sun and M. Zhou, "A probability-guaranteed adaptive timeout algorithm for high-speed network flow detection," *Computer Networks*, vol. 48, no. 2, pp. 215-233, June 2005.

[13] M. Zhou, J. Gong and W. Ding, "Study of network flow timeout strategy," *Journal on Communications*, vol. 26, no. 4, pp. 88-93, April 2005.

[14] J. Cai, Z. Zhang, P. Zhang and X. Song, "An adaptive timeout strategy for UDP flows using SVMs," in *Proc. PDCAT 2010*, Wuhan, 2010, pp. 118-127.

[15] X. Zhang and W. Ding, "Comparative Research on Internet Flows Characteristics," In *Proc. ICNDC 2012*, Hangzhou, 2012, pp. 114-118.

[16] CAIDA, "Analyzing UDP usage in Internet traffic," http://www.caida.org/research/traffic-analysis/tcpudpratio, 2010-01-14/2013-03-21.

[17] D. Lee, B. E. Carpenter and N. Brownlee, "Observations of UDP to TCP ratio and port numbers," in *Proc. ICIMP 2010*, Barcelona, 2010, pp. 99-104.

[18] H. Zhu, W. Ding, L. Miao and J. Gong, "Effect of UDP traffic on TCP's round-trip delay," *Journal on Communications*, vol. 34, no. 1, pp. 19-29, January 2013.

[19] J. Postel, "Transmission Control Protocol," IETF RFC 793, 1981, http://www.rfc-editor.org/rfc/rfc793.txt.

[20] Cisco Systems, "NetFlow Services Solutions Guide," http://www.cisco.com/en/US/docs/ios/solutions_docs/netflow/nfwhite.pdf, 2007-01-22/2013-03-21.

[21] N. Hohn and D. Veitch, "Inverting Sampled Traffic," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 68-80, February 2006.

[22] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proc. PAM 2005*, Boston, 2005, pp. 41-54.

[23] S. Sen, O. Spatscheck and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proc. WWW2004*, New York, 2004, pp. 512-521.

[24] T. Karagiannis, A. Broido, M. Faloutsos and K. C. Claffy, "Transport layer identification of P2P traffic," in *Proc. IMC 2004*, Taormina, 2004, pp. 121-134.

[25] M. Zhou, J. Gong and W. Ding, "High-Speed Network Flows' Dynamical Timeout Strategy Based on Flow Rate Metrics," *Journal of Software*, vol. 17, no. 10, pp. 2141-2151,October 2006.

[26] T. M. Mitchell, *Machine Learning*, 1st ed. New York: McGraw-Hill, 1997, ch. 5, pp. 128-145.

Xiaoguo Zhang, male, was born in Henan Province, China on March 21, 1980. He received the M.S. degree in Computer Application Technology from Henan University of Science and Technology, Luoyang, China in 2008. From December 2008, he is a lecturer at the Electronic Information Engineering School, Henan University of Science and Technology. His current research interests include network measurement, network management and network behavior, etc.

Wei Ding, female, was born in Jiangsu Province, China in 1962. She received the Ph.D degree in Computer Application Technology from Southeast University, Nanjing, China in 1995. From April 2001, she is a professor at the School of Computer Science and Engineering, Southeast University. Her research interests include network measurement, network behavior, network management and network security, etc.