An Adaptive Timeout Strategy for Profiling UDP Flows

[123]Jing Cai [13]Zhibin Zhang [13]Peng Zhang [13]Xinbo Song
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]Graduate University of Chinese Academy of Sciences, Beijing, China
[3]National Engineering Laboratory for Information Security, Beijing, China
caijing@software.ict.ac.cn

***Abstract*—With the increase of network bandwidth, more and more new applications such as audio, video and online games have become the main body in network traffic. Based on real-time considerations, these new applications mostly use UDP as transport layer protocol, which directly increase UDP traffic. However, traditional studies believe that TCP dominates the Internet traffic and previous traffic measurements were generally based on it while UDP was ignored.**

In view of this, we mainly discuss the adaptive timeout strategy of UDP flows in this paper. Firstly, due to its dynamism of packets inter-arrival times, we expound and prove that the existing adaptive timeout strategies are not appropriate for UDP flows. Secondly, we present our adaptive strategy using Support Vector Machine techniques. We build six classifiers to accurately predict its corresponding maximum packet inter-arrival time and adapt its timeout value within the flow duration. Limited to its accurate rating, we present another concept of adjust accuracy rating which can probability-guaranteed(90%,95%,98%) to avoid long flow to be cut into short flows. The experiment result reveals that our adaptive strategy has the potential to achieve significant performance advantages over other widely used fixed and other adaptive timeout schemes.

I. Introduction

The main purpose of the network traffic measurement is to enhance people's awareness about traffic characteristics. The traffic measurement that works based on the network layer started from the 1980s. Earlier studies took the packet as the Building Block. But due to its small granularity, it could not meet the needs in many ways. Claffy *et al.*[1][2] firstly proposed a parameters flow model. The network measurement based on flows can make up the lack of the study based on packets. And in this paper, we formally define a UDP flow to be bidirectional. It is consist of a set of packets with the same 5-tuple {source address, destination address, source port, destination port, transport layer protocol}, and its packet inter-arrival time does not exceed the fixed timeout 64s.

The traffic measurement based on flows have always been a hot issue. However, in the past, during the process of network traffic measurement, people generally believed that TCP traffic occupied the main body of the network traffic, and UDP traffic is negligible, and therefore ignored the measurements of the UDP flows. However, the situation has undergone tremendous changes at present. With the increase of network bandwidth, the traditional networking services based on images and text could no longer satisfy people's needs. More and more audio, video, and online games, have gradually become the main body of the network traffic. These applications mostly use UDP as their transport layer protocol[3], which directly results in the increase of UDP traffic. The organization of CAIDA[4] analyzed the trace collected in the period 2002-2009 on several backbone links located in the US and Sweden and found the ratio between the UDP and TCP in packets, bytes, and flows have increased greatly.

Since the increase of UDP traffic, more and more people have started to pay attention to the traffic of UDP. However, compared with the TCP, we find there at least exist two big differences. Firstly, TCP is a connection-oriented protocol, it has controlling flags such as FIN and RST to explicitly identify the end of flow. But for UDP, it is a connectionless protocol. The main methodology to terminate udp flow is the timeout strategy. The second, compared with TCP, the composition of UDP is more complicated. The characteristics of different applications often demonstrate significant differences. Therefore, the situation is more complex for UDP.

Due to these two great differences, earlier network measurement based on flows mostly focused the TCP flows, while UDP flows was ignored. The study on the UDP flows is nearly in the blank stage. In view of this, we mainly discuss the adaptive timeout strategy of UDP flows in this paper. To the best of our knowledge, we are the first to do so. There are two main contributions in our paper.

- Firstly, through the indication of COV, we find the flow rate of UDP flows are more unsteadily. In common sense, if the flow rate is steadily, it can be used to forecast the adaptive timeout value. However, for UDP flows, due to its dynamism of packets inter-arrival times, we can not use the known information to forecast the adaptive timeout value. Therefore, the existing adaptive timeout strategies are not appropriate for UDP flows.
- Next, we present our adaptive timeout strategy named MSVM. The key notion behind our strategy is that we use the maximum packet inter-arrival as its timeout value. We divided the whole UDP flows into six classes according to its maximum packet inter-arrival. To accurately predict its corresponding class-id, we used the Support Vector Machine techniques. In our strategy, we train six classifiers and use these classifiers to dynamic adapt its timeout value. Limited to its low accurate rating,

978-0-7695-4277-5/10 $26.00 © 2010 IEEE
DOI 10.1109/IC-NC.2010.15

we present another concept of adjust accuracy rating. It is a probability-guaranteed(90%,95%,98%) strategy to avoid long flow to be cut into short flows. We prove our scheme has the potential to achieve significant performance advantages over widely used fixed and other adaptive timeout strategies. And the performance of our strategy increase with the increase of the probability-guaranteed.

The remainder of this paper is organized as follows: Section 2 presents some related work on timeout strategies of flows. In section 3, we compare the differences between the TCP flows and UDP flows in flow characteristic. In section 4, we present our new adaptive timeout strategy named MSVM which uses the Support Vector Machines techniques. In section 5, we compare the performance between our new strategy and other fixed or adaptive timeout schemes. We conclude the paper and give some suggestions in Section 6.

II. Related Work

For traffic measurement based on flows, how to decide the flow termination is significant. At present, there exist two main strategies.

Firstly, a simple way to mark the beginning and ending of a flow is to utilize the protocol label in the packet header fields. The protocol label provides explicit indicator such as SYN/FIN/RST mechanism of TCP-based flows. Though the flag-based explicit flow beginning and termination determination is straightforward, the drawbacks of which are still obvious. For UDP flows, there are no explicit indicators in protocol header that can be tracked for flow termination.

Secondly, timeout is also an important indication for identifying flows. The basic idea behind the timeout-based flow termination decision is that if a flow became inactive beyond a given time duration, it is deemed to be end and removed from memory. The timeout-based method does not rely on the explicit protocol labels in packet header, thus it can deal with the TCP-based flows and UDP-based flows as well. According to the timeout threshold selection algorithms, it could be classified further into two kinds, i.e., the fixed timeout scheme and the adaptive timeout scheme.

A. Fixed timeout scheme

Claffy[1][2] presented a fixed timeout scheme to mark the termination for all flows, and evaluates the flow compression performance with the timeout value from 2s to 2048s. Iannaccone's work[5] also shows that the timeout value in the range of 60s-120s would provide a reasonable estimation for flow numbers. However, the disadvantages of this fixed timeout scheme are also serious. If a larger timeout value is chosen, it may result in the storage space occupied by end flows staying in system memory larger, thus leading to related observation or scheduling system overload. Conversely, a shorter timeout value will cause long flows to be cut into multiple short flows, leading to continuously frequently termination and recreation of flows and resulting in inefficient system resource utilization.

TABLE I
The basic information of the trace

Id	Begin time	End time	Bytes	Packets
I	2009,5.5,14:57	2009,5.6,00:30	275G	2805(million)
II	2010,1.29,13:40	2010,1.29,20:01	274G	492(million)

B. Adaptive timeout strategy

Rye *et al.*[6] developed an adaptive timeout algorithm-Measurement-based Binary Exponential Timeout algorithm(MBET) for flow termination decision. The MBET algorithm is based on the statistical correlation analysis between a flow throughput and its coefficient of variation(COV) of packet inter-arrival times. It preserves a independent timeout value for each flow and dynamic adapt its timeout value according to the observation signals such as the packets inter-arrival and the flow throughput. The initial timeout value of a flow is set to maximum, and the value decreases when the flow throughput exceeds a given threshold during the flow observation period.But there are also some inherent problems in this strategy. Firstly, once the timeout value is decreased, it is never increased again. Secondly, the selection of parameters affects the measurement accuracy. Setting a inappropriate parameters may leads to the unreasonable measurement results and has a great difference with the actual result.

Wang,Li *et al.*[7] presented a probability-guaranteed adaptive timeout algorithm(PGAT). Through the statistical investigation of the correlations between flow size and the maximum packet inter-arrival time, it can obtain the empirical conditional distribution functions for some popular TCP protocol-based application flows. By these functions, this scheme can supply a probability-guaranteed adaptive timeout algorithm for flow termination decision. However, the scheme needs to analyze and judges the flow type, and it is mainly used in the situation of long flows and does not make any optimization for short flows. Besides that, this scheme is more complex, and it is difficult to implement.

III. The characteristic of UDP flows

A. Date set

We collected the traces from a backbone router in China. The basic information of these traces is in Table I. Among these two traces, trace I only contains UDP packets while trace II is a clone of the entire network environment in which there contains not only TCP packets, but also UDP and other protocol packets. For the continence of our process, trace I is partitioned into subtraces with 2.0h-2.5h in length, i.e, the trace I is splited into 4 subtraces. Table II shows the segmentation information and classified statistic metrics for the generated 4 traces.

B. Coefficient of Variation

The COV(Coefficient of Variation)[6] of packet inter-arrival times is defined as the ratio of their standard deviation to its mean, and it indicates how much variation is exhibited

TABLE II
THE BASIC INFORMATION OF THE SUBTRACES

Id	Begin time	End time	Bytes	Packets
I-1	2009,5.5,14:57	2009,5.5,17:11	60G	617(million)
I-2	2009,5.5,17:11	2009,5.5,19:24	60G	617(million)
I-3	2009,5.5,19:24	2009,5.5,21:37	66G	684(million)
I-4	2009,5.5,21:37	2009,5.6,00:30	86G	885(million)

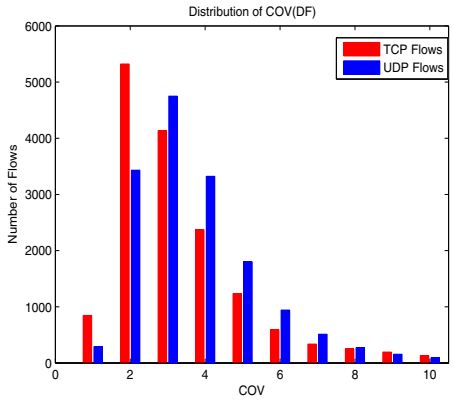

Fig. 1. Distribution of COV(coefficient of variation)

compared to their mean. As a benchmark, independent and exponentially distributed inter-arrival times yield the COV of unity(1).

In intuitive sense, there might exist a fair amount of persistency in flow packet inter-arrival times. The persistency means that when a large(small) inter-arrival time is detected, it is more likely that future inter-arrival times are large(small) as well. If such persistency does exist in the majority of flows, it can serve as quite an effective means for designing adaptive timeout. For example, if a flow appears to exhibit high throughput(pkts/sec) in the beginning, a smaller timeout is likely to be sufficient to determine its end.

Fig.1 illustrates the distribution of COV with the form of DF graph from the same traffic. To better analysis the persistency with the indication of COV, we select some large flows in which packets is larger than 200. The total number of flows is 31,466. Among them, the number of UDP flows is 15,696, and the number of TCP flows is 15,770. Fig.1 also shows that the number of TCP flows is greater than the number of UDP flows when the COV counts less than 3. Conversely, the number of UDP flows is greater when the COV exceeds 3. This phenomenon reveals that the packets inter-arrival times in UDP flows is more instability than TCP flows. Due to its dynamism of packets inter-arrival times, we get the conclusion that the existing MBET algorithm is not appropriate for UDP flows.

IV. ADAPTIVE TIMEOUT STRATEGY

A. Classification of MPIT

An ideal adaptive timeout strategy should accurately predict the inter-arrival for the next packet and set the timeout to this duration. While for the last packet, the timeout value ought to be zero. In this manner, the state holding time equals to the flow active duration. Obviously, to accurately estimate the packets inter-arrival time is impossible, therefore we think use the maximum packet inter-arrival time(MPIT) as its timeout value is a comparatively better choice. Because it can not lead to the frequently termination and recreation of flows but also may bring the system resource wasted on the end flows staying in measurement system smaller.

TABLE III
THE PARTITION GRANULARITY FOR FLOW MAXIMUM PACKET INTER-ARRIVAL(IN SECOND)

Class-ID.	I	II	III	IV	V	VI
Interval	(0,2]	(2,4]	(4,8]	(8,16]	(16,32]	(32,64]

To well reveal the typical flow feature for flow termination decision, we divide the whole UDP flows into six classes whose id named from I to VI according to its maximum packet inter-arrival time within the flow duration. As Table III shows, the corresponding partition granularity are 2s, 4s, 8s, 16s, 32s. There five parameters have divided the whole range of the maximum packet inter-arrival time into six subranges{(0,2], (2,4], (4,8], (8,16], (16,32], (32,64]}. By this means, we have changed the problem from predicting adaptive timeout value to the problem of multiclass classification. Using the SVMs(Support Vector Machines) techniques, we want to accurately predict its class-id of its maximum packet inter-arrival time corresponding based on its flow characteristic.

B. Adaptive Timeout Strategy

In MBET algorithm, the timeout value of flows is initialized to its maximum. However, for short flows, due to its small duration, the timeout value is much more longer than its duration, and thus causing the end flows staying on memory for much unnecessary time and leading the efficiency to the measurement system.

Therefore, the best scheme is to dynamic adapt its timeout value with its packets increase. In our strategy, we train six classifiers based on its flow characteristic such as bytes in flow, flow duration, max/min/average of packet sizes and max/min/average packet inter-arrival time. Using these six classifiers, we dynamic predict and adapt its timeout value when packets in flow reaches 5, 10, 50, 100, 500, 1000. Our scheme can deal with the long flows and short shows as well, because it can dynamic adapt its timeout value with the increase of the packets in flows.

To build our classifiers, we randomly select 10,000 flows and collect its corresponding flow characteristic information when packets in flows reaches 5, 10, 50, 100, 500, 1000. We use the technical of multiclass SVMs to train these data to get some models and use these models to predict the class-id of its maximum packet inter-arrival time corresponding. Because it is a multiclass problem of six classes, the accurate rating is not so high. Therefore, we present another concept of adjust accuracy rating. In general, we commonly reference accurately predict as our predicted class-id equals the actual class-id. If the accuracy rating higher, due to its higher ability

to accurately predict its maximum packet inter-arrival time, the mean flow extra retaining time will be shorter. For this problem, we defined another concept of adjust accuracy rating when our predicted class-id greater or equal to the actual class-id. When we choose a larger class-id, it can not lead to the frequently termination and recreation of flows but also may bring the system resource wasted on the end flow staying in measurement system smaller. In common sense, if the adjust accuracy rating higher, it means that one flow is less likely to be cut into multiple short flows. Therefore, we think the adjust accuracy rating is a appropriate indication for designing adaptive timeout value.

In view of this, we present our probability-guaranteed adaptive timeout strategy named MSVM. The key notion behind our strategy is to improve the accurate rating on the premise of a certain probability-guaranteed(0.90,0.95,0.98) to make sure one flow can not be cut into short flows. Table IV shows the accuracy rating and its corresponding adjust accuracy rating. The accurate rating decrease with the increase of the probability-guaranteed. When the probability-guaranteed equals 0.9 which means that the adjust accuracy rating is higher than 90%, the accurate rating exceeds 50% in most cases. When the probability-guaranteed equals 0.98, the accurate rating is around 45% in most cases, and its maximum does not exceeds 46.5%.

V. Experiment Results and Analysis

In this paper, we assess and compare the performance of our adaptive timeout strategy MSVM with other fixed and adaptive timeout strategy based on two metrics: average hold factor $\bar{F}_{hold}$ [6]and thrashing.

We define $F_{hold} = D_{hold}/D_{act}$[6], D_{hold} represents the sum of the flow duration and the flow timeout, D_{act} represents the flow duration. The smaller ratio between the time wasted in memory(the flow timeout) and the useful time (the flow duration) reveals that the measurement system is more efficient. For the flow set formed by N flows. The average F_{hold} is calculated as:

$$\bar{F}_{hold} = (\frac{1}{N}\sum_{n=0}^{N} F_{hold}^{-1}(n))^{-1} \quad (1)$$

We note that performance improves (gain) when $\bar{F}_{hold}$ decreases, while it is degraded (loss) when the degree of thrashing increases. For the fixed timeout strategy, both gain and loss move in the same direction; as timeout becomes smaller, the average hold factor decrease(higher gain), but thrashing also increases(higher loss). For this reason, we define the overall performance metric M[6] as:

$$M(T) = \frac{\alpha G(T)}{\beta L(T)} \quad (2)$$

where G(T)[6] and L(T)[6] are relative performance gain and loss defined as

$$G(T) = \frac{\bar{F}_{hold}(T_{ref}) - \bar{F}_{hold}(T)}{\bar{F}_{hold}(T_{ref})} \quad (3)$$

$$L(T) = \frac{N(T) - N(T_{ref})}{N(T_{ref})} \quad (4)$$

TABLE V
The experiment result versus different strategies for trace I-1

Strategy	UDP Flows	F_{hold}	Large UDP Flows	Performance
Fixed-2	83658624	15.321001	2177607	0.074746
Fixed-4	70617495	13.032734	1876887	0.226040
Fixed-8	53605692	10.791272	1866625	0.652153
Fixed-16	42924057	10.682923	1927441	1.469059
Fixed-32	36910931	13.422735	1641154	2.690103
MBET	34954395	12.994489	1797045	10.135670
MSVM-90	34496741	14.092347	1509351	16.89788
MSVM-95	34316581	15.209276	1459979	21.402370
MSVM-98	34175071	16.328326	1436746	40.673270

with T_{ref} being the fixed timeout value used as a reference case and N(T) being the number of total flows created with timeout T (for fixed timeout) or configuration (for our adaptive timeout strategy). The weight factors α and β may be used to assign non-uniform weights to each metric depending on their relative importance. In this study, we use α = β = 1, treating gain and loss equally. To evaluate the algorithm performance, we choose the parameter CFG-2[6] for the MBET algorithm.

Table V shows the experiment result such as the number of UDP flows, $\bar{F}_{hold}$ factor, and the number of the large UDP flows versus different strategies for trace I-1. As Table V shows, the number of UDP flows is a decreasing function of the timeout value. In addition, the $\bar{F}_{hold}$ is a increase function of the timeout value in most cases. Therefore, what we need to do is to select the best strategy which can reach the equilibrium state and get the maximum performance metric. For fixed timeout strategies, Table V also reveals that the performance metric increases with the increase of the timeout value. At the timeout value of 32s, it gets the maximum performance metric for the fixed strategies. However, the performance of the MBET algorithms is five times bigger than the fixed timeout schemes. In addition, it also shows that our MSVM algorithm outperform than the commonly used adaptive timeout strategies(MBET). Commonly, its performance metric is two-four times than the MBET.

Fig.2 shows the comparison of performance metric over different strategies for trace I-1, I-2, I-3 and I-4, which is drawn in a log scale. It clearly indicates our MSVM outperforms than other fixed and MBET timeout schemes. The MSVM algorithm achieves its high performance metric by significantly reducing the $\bar{F}_{hold}$ factor with only a slight increase in the number of flows. As Fig.2 shows, the performance metric of our MSVM algorithm is usually two-four times as the MBET algorithm and twenty-thirty times as the fixed timeout schemes. We also note that its performance metric increases with the probability-guaranteed rating increase from 0.90 to 0.98.

VI. Conclusion

In this paper, we mainly solve two important questions. The first, compared with TCP flows, we find the COV(coefficient of variation) of the packet inter-arrival time is more unsteadily. Due to this great differences, the existing adaptive timeout

TABLE IV
THE RESULT OF SUPPORT VECTOR MACHINES

Experiment ID	Length	Best c	Best g	Accuracy rating	Adjust accuracy rating
MSVM-98	5	0.01625	0.25	41.30%	98.20%
	10	0.5	8.0	35.10%	98.00%
	50	0.0625	0.0078125	45.91%	98.37%
	100	0.125	0.00390625	46.38%	98.25%
	500	0.5	0.0009765625	48.06%	98.10%
	1000	0.5	0.001953125	43.75%	98.85%
MSVM-95	5	0.25	1.0	47.11%	95.19%
	10	0.25	0.25	41.18%	95.09%
	50	0.125	0.001953125	52.32%	95.05%
	100	0.5	0.0078125	50.43%	95.49%
	500	1.0	0.00390625	49.44%	96.24%
	1000	1.0	0.001953125	45.02%	95.32%
MSVM-90	5	2.0	2.0	52.41%	90.22%
	10	0.125	0.015625	49.06%	90.40%
	50	0.25	0.001953125	54.51%	90.40%
	100	0.5	0.001953125	53.78%	90.49%
	500	1.0	0.0009765625	51.33%	91.67%
	1000	1.0	0.0009765625	46.41%	93.50%

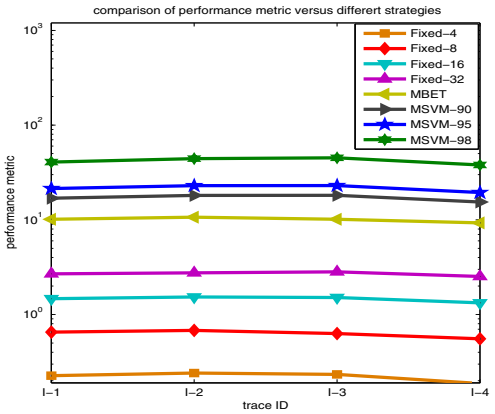

Fig. 2. Comparison of performance metric between fixed and adaptive timeout schemes for different traces. The reference case was with T = 64 sec. In both sets, the MSVM significantly outperforms the fixed and other adaptive timeout schemes.

strategies which mostly design for the TCP flows are not appropriate for UDP flows.

Second, we present our adaptive timeout strategy named MSVM. The key notion behind our strategy is that we use the maximum packet inter-arrival time as its timeout value. Using the Support Vector Machine techniques, we build six classifiers to accurately predict its class-id of its maximum packet inter-arrival time corresponding and adapt its timeout value with the increase of the packets in flows. Limited to its low accurate rating, we present another concept of adjust accuracy rating. It is a probability-guaranteed(90%,95%,98%) strategy to avoid long flow to be cut into short flows. Through analysis and experiment, we prove our scheme achieving significant performance advantages over widely used fixed and other adaptive timeout strategies. And the performance of our strategy increase with the increase of the probability-guaranteed.

ACKNOWLEDGMENT

Our work is supported in part by the National Basic Research Program ”973” of China(Grant No.2007CB311100) and the National Science Foundation of China(Grant No.61003167).

REFERENCES

[1] K.C. Claffy, *Internet traffic characterization* . [Ph.D. Thesis],San Diego: University of California, 1994.
[2] K.C. Claffy, H.W. Braun, and G.C. Polyzos, *A Parameterizable Methodology for Internet Traffic Flow Profiling[J]*. IEEE Journal on Selected Area in Communications, 1995,13(8):1481-1494.
[3] K. Sripanidkulchai, B. Maggs, and H. Zhang, *Analysis of Live Streaming Workloads on the Internet*. In Proc. of IMC’04, October, 2004, pp. 41-54.
[4] CAIDA. http://www.caida.org/research/traffic-analysis/tcpudpratio/
[5] G. Iannaccone, C. Diot, and I. Graham, *Monitoring very high speed links*. In Proc. of the First ACM SIGCOMM Workshop on Internet Measurement Workshop 2001, San Francisco, California, USA, November 2001, pp. 267-271.
[6] B. Ryu, D. Cheney, and H.W. Braun, *Internet flow characterization: adaptive timeout strategy and statistical modeling*. In: Workshop on Passive and Active Measurement (PAM), 2001.
[7] J.F. Wang, L. Li, F.C. Sun and M.T. Zhou, *A probability-guaranteed adaptive timeout algorithm for high-speed network flow detection*. Computer Networks,2005,48(2):215-233
[8] M.S. Kim, Y.J. Won, and J.W. Hong, *Characteristic analysis of internet traffic from the perspective of flows*. Computer Communications, 2006,vol.12:1639-1652
[9] C.W. Hsu and C.J. Lin, *A comparison of methods for multi-class support vector machines*. IEEE Transactions on Neural Networks, 13(2002), 415-425.
[10] N. Hohn and D. Veitch, *Inverting sampled traffic*. In Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. 2003. 222-233.
[11] M. Rey, *Transmission control protocol*. RFC793,1981.
[12] R. Jain and S.A. Routhier, *Packet Trains-Measurements and a New Model for Computer Network Traffic*. IEEE Journal on Selected Area in Communications, Vol. SAC-4, No. 6, Sep, 1986, pp. 986 995.
[13] D.D. Clark, *The design philosphy of the Darpa Internet protocols*. In Proc. of ACM SICCOMM ’88, Aug. 1988, pp. 106-114.
[14] LIBSVM: A Library for Support Vector Machine,C.-C. Chang and C.-J. Lin. (2001). [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm/
[15] A. Este, F. Gringoli, and L. Salgarelli, *Support Vector Machines for TCP Traffic Classification*. Elsevier Computer Networks (COMNET), Vol. 53, No. 14, pp. 2476-2490, Sep. 2009.