Multilevel Early Packet Filtering Technique based on Traffic Statistics and Splay Trees for Firewall Performance Improvement

Zouheir Trabelsi and Safaa Zeidan
Faculty of Information Technology
UAE University
Al -Ain, UAE
trabelsi@uaeu.ac.ae
safaa_zeidan@hotmail.com

***Abstract*— This paper presents a mechanism to improve firewall packet filtering time through optimizing the order of security policy filtering fields for early packet rejection. The proposed mechanism is based on the optimization of the filtering fields order according to traffic statistics. Furthermore, the mechanism uses multilevel packet filtering, and in each level unwanted packets are rejected as early as possible. So, the proposed mechanism can be considered also as a device protection mechanism against denial of service (DoS) attacks targeting the default policy rule. In addition, early packet acceptance is done through using the splay tree data structure which changes dynamically according to traffic flows. So, repeated packets will have less memory accesses and therefore reducing the overall packets matching time. The proposed technique aims to overcome some of the performance limitations of the previous technique, named Self Adjusting Binary Search on Prefix Length [1] (SA-BSPL). The numerical results obtained by simulations demonstrate that the proposed mechanism is able to significantly improve the firewall performance in terms of cumulative packet processing time compared to SA-BSPL technique.**

Keyword: Packet Classification, Early packet Rejection, Binary Search on Prefix Length, Splay Tree, Hash Table.

I. INTRODUCTION

Firewall packet filtering is performed in a sequential order starting from the first rule until a matching rule is found. If no matching rule is found, the packet is processed by the default rule. Thus, the computational complexity of the filtering process depends on the length of each rule as well as the depth of finding a matched rule in the list. Hence, the order of rules, the order of rule-fields, and the characteristics of the packet flows have a significant impact on packet filtering time.

In addition, unwanted traffic targeting the default rule may cause more harm than others by producing an overhead to the system which is proportional to the number of rules used in the security policy. Such unwanted traffic may cause a denial of service (DoS) attack and degrade considerable the firewall's performance. From this point of view, it is very important to reject such traffic as early as possible as shown in Fig.1.

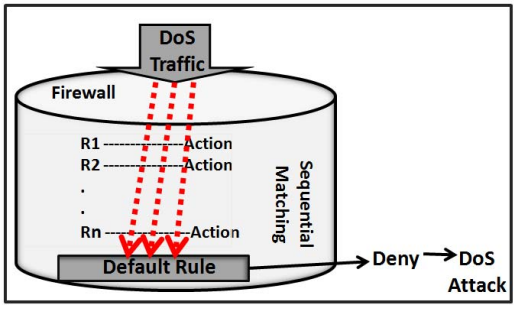

Figure 1. Unwanted Traffic targeting default rule may cause DoS attack

In this paper, we propose an approach to optimize firewall early acceptance path as well as early rejection path. The approach uses splay tree data structure that changes dynamically according to traffic flows. This characteristic of the splay tree allows for early acceptance of repeated packets. On the other hand, the technique uses three levels of filtering to reject the unwanted traffic as early as possible.

Since SA-BSPL [1] depends only on the splay tree data structure for early packet rejection or acceptance, our main contribution in this paper is that the proposed early acceptance and rejection approach is done through both policy fields ordering and cascaded level filtering using traffic statistics.

The paper is organized as follows: Section II discusses the related work. Section III presents the proposed approach. Section IV presents numerical results obtained from experiments based on simulations, in order to evaluate the firewall performance for the proposed mechanism. Finally, Section V concludes the paper.

II. RELATED WORK

The most early research works focus on the improvement of searching times using various mechanisms including hardware-based solutions [6, 7], specialized data structures [8, 9, 5, 10, 11, 12], and heuristics [5]. Research works in [17, 2, 4, 13 and 14] focus on the statistical filtering schemes to improve the average packet filtering time. The structure of searching by taking into account of packet flow dynamics is introduced in [14, 15, 3]. In order to optimize firewall filtering policies by

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

utilizing the characteristics of packet flow over Internet is presented in [16]. Segments-Based Tree Search (STS) scheme [4] uses bounded depth Huffman trees to enhance the searching based on the statistics collected from segments. However, this scheme may need large overheads for maintaining the tree periodically. To reduce the overheads, Segments-based List Search (SLS) [4] by keeping the segments in a most-recently-used (MRU) order instead of using trees. SLS scheme can only be used when packet flows are in steady state. However, in real network environment, attacks such as denial of service attacks (DoS) usually produce huge burst traffic. From this point of view, SLS is not capable to provide a better performance comparing to STS.

The idea of early rejection was introduced in [2, 1, 18]. In [2] a new approach named FVSC is proposed to optimize the rejection path, this technique uses set cover approximation algorithm to construct early rejection rules from the original security policy common field values. PBER technique in [18] is considered as a generalization of FVSC [2] in the sense that FVSC [2] focuses only on rejection path while PBER [18] finds short cuts for both accepted and rejected packets. In [1] a binary search on prefix length algorithm is applied to every policy filtering field along with the property of splaying the search tree nodes to handle the early accepted packets.

III. PROPOSED WORK

The following firewall policy example in Table I will be used throughout this paper.

The proposed scheme is similar to the search mechanism used in SA-BSPL technique [1]. It consists of a set of statistical splaying filters that use binary search on prefix length and it is called: Statistical Splaying Filters with Binary Search on Prefix Length (SSF-BSPL). Since the proposed SSF-BSPL in this paper uses binary search on prefix length [19] with the splay tree [20], we will describe them in more details in the following sections.

A. Binary Search on Prefix Length (BSPL)

The Binary Search on Prefix Length algorithm (BSPL) [19] as shown in Fig. 2 is based on three main ideas: First, it uses hashing to check whether an incoming value matches any prefix of a particular length; second, it uses binary search to reduce number of searches from linear to logarithmic; third, it uses pre-computation (markers) to prevent backtracking in case of failures in the binary search.

TABLE I. EXAMPLE OF A FIREWALL POLICY WITH 8-BIT PREFIXES FOR THE SOURCE AND THE DESTINATION IP ADDRESSES FIELD

Rule no.	Scr Prefix	Dst Prefix	Scr Port	Dst Port	Protocol	Action
R1	1010*	0000*	*	21	TCP	allow
R2	11110*	000111*	*	53	UDP	allow
R3	001101*	101011*	*	80	TCP	allow
R4	10100*	00000*			ICMP	allow
R5	1110*	000*			ICMP	allow
R6	00*	0011*	>1023	80,23	TCP	allow
R7	01*	110*	*	25	TCP	allow
R8	110110*	110010*	*	69	UDP	allow
R9	00110*	1010*	*	21	TCP	allow
R10	110*	11010*	*	[0,100]	TCP	allow
R11	000*	11*	*	*	TCP	allow
R12	0110*	1*	*	*	TCP	Deny
R13	*	0*			ICMP	Deny
R14	*	*	*	*	*	Deny

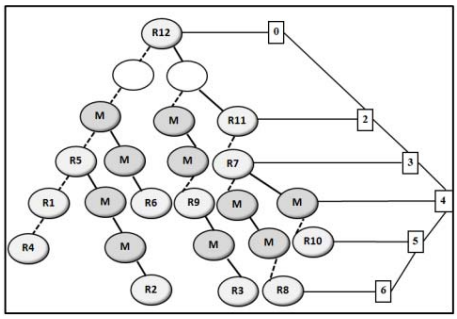

Figure 2. Waldvogel's binary search on prefix length for the destination address field prefixes of Table I.

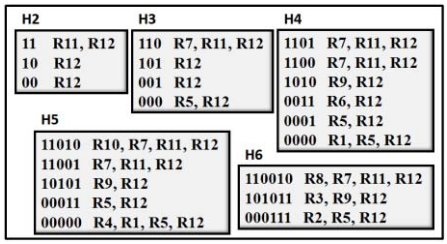

Figure 3. The collection of hash-tables for the destination address field prefixes of Table 1

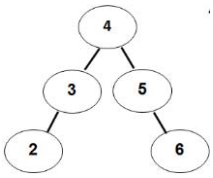

Figure 4. Splay tree according to the hash-tables in Figure 3

Fig. 3 shows the collection of hash tables for the destination address field prefixes in table I. The length of the prefixes is stored in a splay tree as shown in Fig. 4.

B. Splay Tree Data Structure

The accessed node in the Splay Tree becomes the root through applying either a single rotation or a series of rotations (Zig, Zig Zag and Zag Zig operations).

C. Proposed Statistical Splay Tree Policy Filters(SSF-BSPL)

Although packets can be rejected by intermediate deny rules in the policy, this technique optimizes the process of matching time for the unwanted as well as wanted traffic packets. This optimization is done through three filtering levels: Statistical Policy Filtering Level (for early packet rejection), Field filtering level (for early packet rejection and acceptance) and Cascaded Filtering level (for early packet rejection).

1) Statistical Policy Fields Filtering level

In Statistical Policy Field Filtering level filtering fields that can reject most of the traffic is ordered according to the traffic statistics. For a given window of traffic flow, policy fields are arranged in descending order staring from the field with the highest rejection statistics. This would allow reducing memory

access to the splay trees and hash tables and consequently, reducing the filtering processing time for the unwanted packets.

For example, suppose that a security policy consists of filtering fields F1, F2, F3, F4 with traffic rejection percentages 8%, 2%, 30%, 80%, respectively. So, instead of checking packet header fields against (F1, F2, F3, F4), we check them against statistical policy ordered fields F4, F3, F1, F2 as shown in Fig. 5. There is another improvement for the proposed statistical policy field filtering level in Fig. 5(b) explained in cascaded filtering level.

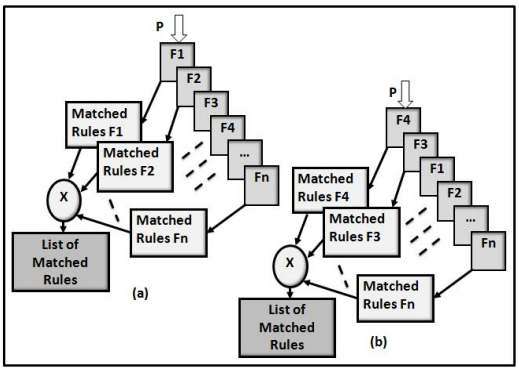

Figure 5. (a) SA-BSPL. (b) Proposed Statistical Policy Field Level Filtering.

2) Cascaded Statistical Policy Fields Filtering level

In Cascaded Statistical Policy Fields Filtering level, list of matched field rules is intersected with previous intersected matched rules list. If there are no common rules between the two lists, the packet will be rejected as early as possible with no need to check other fields. Otherwise next field is checked and if corresponding packet field matches this field, list of matched rules will be intersected with previous list and so on. Fig. 6 illustrates the idea of cascaded policy fields filtering level.

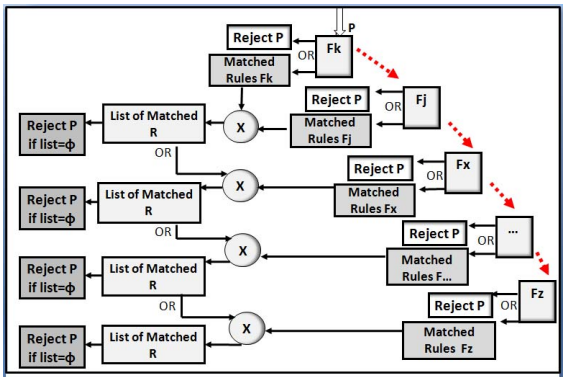

Figure 6. Proposed Cascaded Policy Fields Filtering Level.

3) Field Filtering Level

In Field Filtering Level each Filtering field consists of a collection of hash-tables and a splay tree as proposed in SA-BSPL [1].

In the proposed SSF-BSPL technique, the three filtering levels are combined together to enhance packet processing time as follows: Upon packet arrival, the first statistical field is chosen (Statistical policy fields filtering level), then the field splay tree and hash tables are searched using binary search on prefix length mechanism (Field filtering level). If the packet doesn't match the first statistical policy field it will be rejected by min-node [1]. Otherwise the search continuous to the second statistical policy field and if the packet matches the second field, list of matched rules will be intersected with the list of matched rules from previous field (Cascaded policy fields filtering level). If the intersection is empty the packet is rejected. Otherwise search continuous to the next field and so on. Fig. 7 shows the algorithm for the proposed SSF-BSPL mechanism.

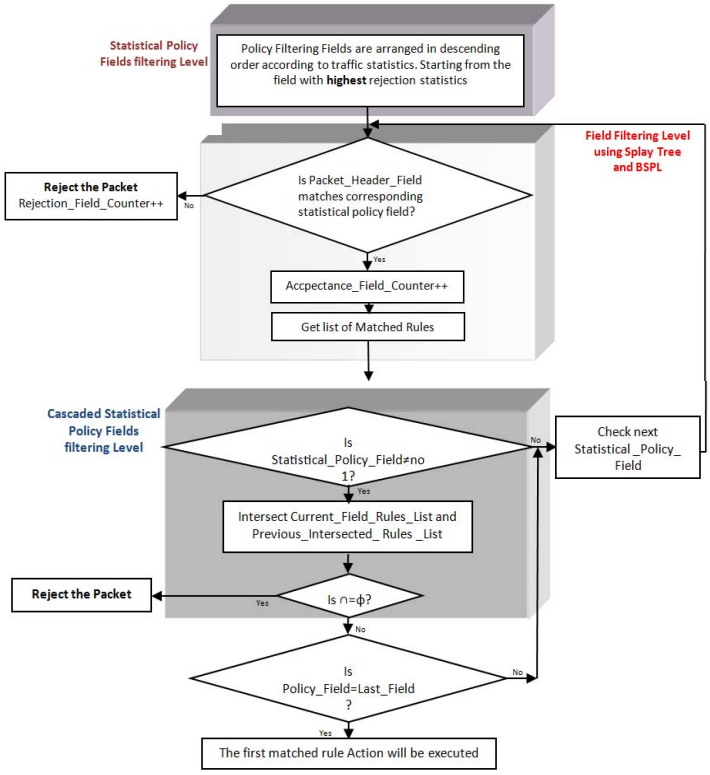

Figure 7. SSF-BSPL packet matching algorithm

IV. MATHEMATICAL MODEL

A. Statistical policy fields reordering decision

Here we will discuss whether to decide to reorder the policy fields or not using the statistical policy field filtering level. Assume that the firewall consists of k filtering fields with certain order in the previous situation. We want to know if this order will be changed or not in the current situation. First let us introduce some notations to be used in table II.

TABLE II. PREVIOUS AND CURRENT SITUATIONS FOR POLICY FILTERING FIELDS.

	F_1	F_2	…	F_k	Total
Previous	$n_{1,1}$	$n_{1,2}$		$n_{1,k}$	N_1
Current	$n_{2,1}$	$n_{2,2}$		$n_{2,k}$	N_2
Total	C_1	C_2		C_k	N

Let $n_{i,j}$ is the number of rejected packets by field F_j, where i refers to the current or previous situation. Let $P_{i,j}$ for i = 1, 2 and j = 1,…, k denotes the probability of field F_j to reject a

packet. We wish to test if there is a change in these proportions between the current and the previous situations. This amount is to test:

$$\begin{cases} H_0 & P_{1,j} = P_{2,j} \quad \text{for } j = 1,\ldots,k \\ H_1 & P_{1,j} \neq P_{2,j} \quad \text{for some } j = 1,\ldots,k \end{cases}$$

which is known in statistics as the test of equality of two multinomial distributions and is carried out using the Chi-square test. To be precise let $E_{i,j}=N_i C_j/N$ for i = 1, 2 and j = 1,…, k. And define:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(n_{i,j} - E_{i,j})^2}{E_{i,j}}$$

We will reject H_0 and conclude that there is a need to re-order if the Value of χ^2 is larger than $\chi^2_{\alpha,k-1}$ for a given value of α like (0. 05, or 0.01). The values of $\chi^2_{\alpha,k-1}$ are obtained from tables of the chi-square distribution. Note also that computer codes to calculate $\chi^2_{\alpha,k-1}$ are also available.

B. *Optimal fields order*

Our main objective in this section is to find the optimal policy fields order that minimizes the total packet filtering time. We assume that the traffic flow is divided into segments of T packets. Let T_i defines the total traffic entering policy field F_i and $T_{i(r,min)}$, $T_{i(r,x)}$ are the total number of packets rejected by F_i field filtering level (min-node) and cascaded filtering level (intersection), respectively. That is,

$$T_i = \begin{cases} T & i = 1 \\ (1 - P_{T_{1(r,\min)}})T_1 & i = 2 \\ (1 - P_{T_{i-1(r,x)}})(T_{i-1} - T_{i-1(r,\min)}) & i = 3,..,k \end{cases}$$

Where $P_{Ti(r,min)}$ is the probability that field F_i rejects a packet using field filtering level (min-node), and $P_{Ti(r,x)}$ is the probability that field F_i rejects a packet using cascaded filtering level (intersection).

$$P_{Ti(r,min)} = T_{i(r,min)} / T_i$$
$$P_{Ti(r,x)} = T_{i(r,x)} / (T_i - T_{i(r,min)})$$

Let $P(F_i)$ is the probability for field F_i to reject a packet. $P(F_i)$ depends on both $P_{Ti(r,min)}$ and $P_{Ti(r,x)}$ that is,

$$P(F_i) = \begin{cases} P_{T_{1(r,\min)}} & i = 1 \\ P_{T_{i(r,\min)}} * P_{T_{i(r,x)}} & i = 2,\ldots,k \end{cases}$$

So, the firewall policy fields will be ordered according to its $P(F_i)$.

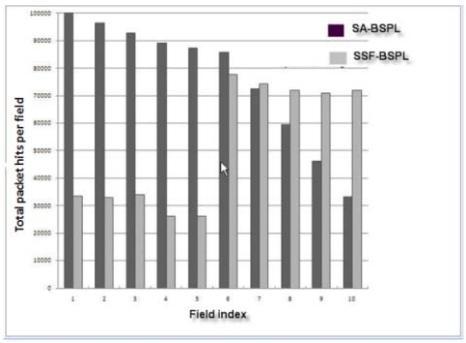

Figure 8. Cumulative field hit ratio

V. EVALUATION

A. *SSF-BSPL vs. SA-BSPL*

Fig. 8 shows the accumulated fields hit ratio for SSF-BSPL and SA-BSPL techniques. To investigate the effect of the proposed SSF-BSPL technique, we generated a special packet flow consisting of only 10% of packets matching the filtering rules, and the other 90% of packets not matching any of the filtering rules. Fig. 9 shows that the cumulative processing time gain provided by SSF-BSPL is significantly less compared to SA-BSPL. This is because the filtering process in SA-BSPL scheme may have to check most of the policy fields to reach a decision regarding a given packet when the firewall is heavily loaded with non-matching malicious packets depending on the position of the field that will reject the packet. However, by applying SSF-BSPL scheme, the position of the policy fields as well as the cascaded level filtering are optimized to make the matching process much faster.

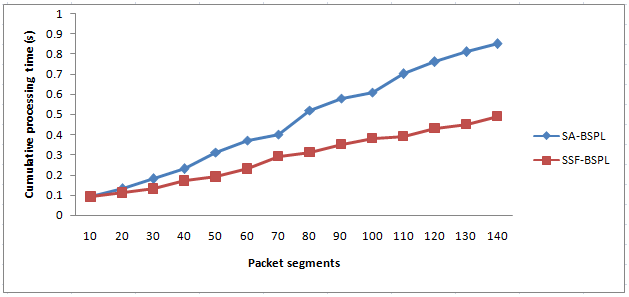

Figure 9. Cumulative processing time for high non-matching traffic

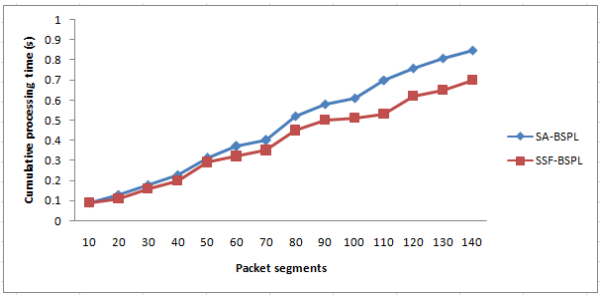

Figure 10. Cumulative processing time for high matching traffic

In the second DoS attack experiment, the firewall is flooded by matching packets, such as SYN flood attack. Special packet flows with 90% of the packets matching the filtering rules, and 10% of the packets not matching any of the filtering rules are used in the experiment. Fig. 10 shows that there is slight difference between the two techniques due to the small packet rejection percentage.

B. *Field Filtering Level vs Cascaded Statistical Policy Fields Filtering level*

In this section we intend to investigate the significant effect of adding the cascaded filtering level on the performance enhancement of the proposed SSF-BSPL technique.

In this experiment, we used the same above traffic flow that consists of only 10% of packets matching the filtering rules, and the other 90% of packets not matching any of the filtering rules. In the first stage, this traffic is applied to the SSF-BSPL technique when only field filtering level is involved. Then, the cascaded filtering level is added to evaluate its effect on improving the overall processing time. Fig. 11 shows the processing time gain when applying field filtering level only and both field filtering level and cascaded filtering level.

VI. CONCLUSION

Data networks may suffer from some traffic flows that are very expensive to classify and filter as they may undergo a longer than average list of filtering rules before being rejected by the default deny rule.

In this paper, we have proposed a traffic statistical scheme, based on multilevel filtering modules using splay trees and hash tables. The proposed scheme can easily reject unwanted traffic in early stages as well as accept repeated packets with less memory accesses, and thus less overall packets matching time. Therefore improving firewall performance especially for traffic flows with high packet rejection percentages as proved by evaluation. From this point of view, the proposed SSF-BSPL scheme can be considered as a device protection mechanism against DoS traffic. The SSF-BSPL scheme uses a mathematical model that can decide whether to change or keep the statistical policy fields order for the next packet segment. In future work, we intend to improve the proposed mathematical model to take into consideration not only ordering the policy filtering field according to their $P(F_i)$, but also the order of each field with respect to pervious fields. This is due to the fact that cascaded filtering level has a significant effect in reducing the cumulative filtering time.

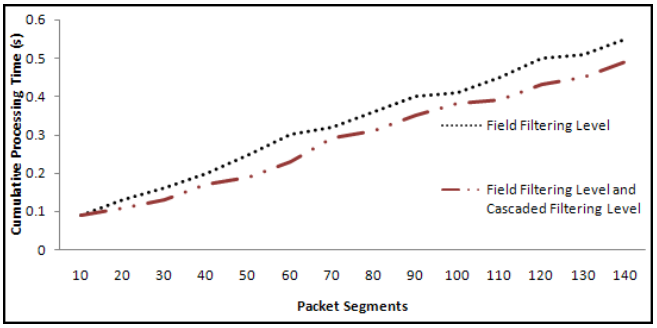

Figure 11. SSF-BSPL Field Filtering Level vs. Cascaded Filtering Level cumulative processing time gain

ACKNOWLEDGMENT

The authors acknowledge the support of Emirates Foundation through Research Grant No. (2009/161).

REFERENCES

[1] N. Neji, A. Bouhououla: Dynamic Scheme for Packet Classification Using Splay trees, Information Assurance and Security, pp. 1-9, 2009.

[2] H. Hamed, A. El-Atawy, and E. Al-Shaer. Adaptive statistical optimization techniques for firewall packet filtering. In *IEEE INFOCOM'06*, April 2006.

[3] K. Lan and J. Heidemann. On the correlation of internet flow characteristics. Technical Report ISI-TR-574, USC/ISI, 2003.

[4] A. El-Atawy, T. Samak, E. Al-Shaer and H.Li. Using online traffic statistical matching for optimizing packet filtering performance. *IEEE INFOCOM'07*, pages 866-874, 2007.

[5] P. Gupta and N. McKeown. Algorithms for packet classification. *IEEE Network*, 15(2):24–32, 2001.

[6] F. Baboescu and G. Varghese. Scalable packet classification. In *ACM SIGCOMM'01*, 2001.

[7] A. J. McAulay and P. Francis. Fast routing table lookup using CAMs. In *IEEE INFOCOM'93*, March 1993.

[8] V. Srinivasan, Subhash Suri, and George Varghese. Packet classification using tuple space search. In *Computer ACM SIGCOMM Communication Review*, pages 135–146, October 1999.

[9] A. Feldmann and S. Muthukrishnan. Tradeoffs for packet classification. In *IEEE INFOCOM'00*, March 2000.

[10] P. Gupta and N. McKeown. Packet classification using hierarchical intelligent cuttings. In *Interconnects VII*, August 1999.

[11] E. Cohen and C. Lund. Packet classification in large isps: design and evaluation of decision tree classifiers. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRIC international conference on Measurement and modeling of computer systems*, pages 73–84, New York, NY, USA, 2005. ACM Press.

[12] Thomas Y. C. Woo. A modular approach to packet classification: Algorithms and results. In *IEEE INFOCOM'00*, pages 1213–1222, March 2000.

[13] P. Gupta, B. Prabhakar, and S. Boyd. Near optimal routing lookups with bounded worst case performance. In *IEEE INFOCOM'00*, 2000.

[14] L. Kencl and C. Schwarzer. Traffic-adaptive packet filtering of denial of service attacks. In *WOWMOM'06: The 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 485–489, Washington, DC, USA, 2006.

[15] S. Acharya, M. Abliz, B. Mills and T.F. Znati, Optwall: a hierarchical traffic-aware firewall, *Proceedings of 14th Annual Network & Distributed System Security Symposium (NDSS)*, San Diego, US, February 2007 .

[16] H. Hamed and E. Al-shear: Dynamic Rule-ordering optimization for High-speed Firewall Filtering. In ASIACCs' 06, March 21-24, 2006, Tuipei, Taiwam.

[17] H. Hamed, A. El-Atawy, and E. Al-Shaer. On Dynamic Optimization of Packet Matching in High-Speed Firewalls, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 10, OCTOBER 2006.

[18] E. Al-Shear, A. El-Atawy, T. Tran: Adaptive Early Packet filtering for Defending firewalls against DoS Attack. In Proceeding of IEEE INFOCOM, pp. 1-9, 2009.

[19] M. Waldvogel, G. Varghese, J. Turner, B. Plattner. "Scalable High Speed IP Routing Lookups". In *Proceedings of the ACM SIGCOMM (SIGCOMM '97)*, pp. 25-36, 1997.

[20] D. Sleator, R. Tarjan. "Self Adjusting Binary Search Trees". *Journal of the ACM*, 32(3), pp. 652-686, 1985.