A performance-optimized firewall rules matching algorithm

Li Zhong
Institute of Information Science and Technology
Zhengzhou
Zhengzhou, P. R. China
e-mail: lz196903@163.com

Li Xiao
Institute of Information Science and Technology
Zhengzhou
Zhengzhou, P. R. China
e-mail: lx195803@163.com

***Abstract*—The algorithm of firewall rules matching designed in this paper is based on the idea of divide-and-conquer the rules set. The rules set are divided into multiple sub-sets in accordance with the protocol type. Then, accordance with the relationship between two rules, each sub-set is divided into two groups: disordered group and queue group. Furthermore, hash function is designed to match rules in disorder group, while indexing algorithm is proposed to match rules in the queue group. The analysis shows that the efficiency of the algorithm is much better than similar algorithms, greatly improving the performance of the firewall.**

Keywords- firewall; rule matching; divide-and-conquer; hash

I. INTRODUCTION

With the development of society and the advancement of technology, the technology and scale of Internet is developing rapidly, which has brought great convenience to the work and life. At the same time, it also has bring new challenges for network security. Network firewall is taken as one important security measures, which provides effective security guarantees in large-scale, high-traffic network environment. Rules matching (packet classification) is one of the important aspects, which also is the bottlenecks of performance. The firewall rule set is constructed by large numbers of rules which includes one group of source address, one group of destination address, one group of source port, one group of destination port and the corresponding action. the main function of the rules matching is that: to receive the packet matching the rule set, to find their best match of the rules, and to process the packet, according to the rules defined action.

With the scale of the firewall rule set larger, the average processing time for each packet is increasing, so the performance and efficiency of the firewall directly reduce. Thus, in the case, the size of the rule set becomes larger and lager, there are two ways to maintain the performance of the firewall: 1. Reducing the processing time required that each data packet matching in each rule set, in other words, it is the improvement of the matching algorithm that improve the matching efficiency; 2. reducing the size of rule set that each packet to match.

Matching algorithm is improved in different perspectives[1-10], a faster algorithm is proposed by MEINERS[2], which is based on TCAM (Ternary Content Addressable Memory). In addition, the time complexity of it is generally constant. However, the rules are supported in the algorithm just only expressed by prefix form. The resource consumption of the matching algorithm is very large, which limits the use of its marketing. TCAM matching algorithm is improved by dynamic coding to reduce its energy consumption, but it still just supports only rules prefix form. The research of matching algorithm is conduct by the hash table, which can get well performance guarantees in the worst case. On the other hand, the average performance is very poor[4][5]. Thus, the matching algorithm based on decision tree gets higher average performance[6] [7], but such algorithms in the worst case performance is poor.

Under this background, the idea of divide-and-conquer is use firstly to divide the rules set into multiple sub-set by the protocol in this paper, then each sub-set is divided into two groups according to the relationship between the rules. Furthermore, the two group rule sets is respectively used a different method to match rules according to its own characteristics.

II. BASE CONCEPT

The main contents of the firewall filtering rules is mainly to include: source address, destination address, source port, destination port, protocol type and the measure to take for the convenience of description in this paper, the definition of rule is given just as follows.

Definition 1: If all domains of the rules R_x are not subset, superset or equal with corresponding domains of rules R_y, then, R_x and R_y can be said as completely unrelated. It can be abstracted as mathematical expressions as follows: $\forall i \in \{ip\_s, ip\_d, port\_s, port\_d, pro\}$,and $\forall \Delta \in \{\subset, \supset, =\}$ then, $\exists R_x[i] \Delta R_y[i]$ not established.

Definition 2: If all domains of the rules R_x are equal with corresponding domains of rules R_y, then, R_x and R_y can be said as completely equal. It can be abstracted as mathematical expressions as follows: $\forall i \in \{ip\_s, ip\_d, port\_s, port\_d, pro\}$, $\exists R_x[i] = R_y[i]$.

Definition 3: If all domains of the rules R_x are subset of corresponding domains in rules R_y, then, R_x and R_y can be said as contains relevant. It can be abstracted as mathematical expressions as follows: $\forall i \in \{ip\_s, ip\_d, port\_s, port\_d, pro\}$,if $\exists R_x[i] \subseteq R_y[i]$,and $\exists i$, $R_x[i] \neq R_y[i]$.

Definition 4: If there is at least one domain of the rules R_x that is subset or superset of corresponding domain in rules

R_y , in addition, there is at least one domain of the rules R_x that is not subset or superset of corresponding domain in rules R_y , then, R_x and R_y can be called as partly independent. It can be abstracted as mathematical expressions as follows: $\exists i, j \in \{ip\_s, ip\_d, port\_s, port\_d, pro\}$ and $\exists \Delta \in \{\subset, \supset, =\} \Rightarrow R_x[i] \Delta R_y[i]$, $\forall \Delta \in \{\subset, \supset, =\} \Rightarrow R_x[j] \Delta R_y[j]$ is not exist.

Definition 5: if part domains of the rule R_x is a subset of the corresponding domain in rules R_y , and the other domains of the rules R_x is a superset of the rules R_y , then, R_x and R_y can be called as cross-correlation. It can be abstracted as mathematical expressions as follows: $\forall i \in \{ip\_s, ip\_d, port\_s, port\_d, pro\}$, $\exists \Delta \in \{\subset, \supset, =\} \Rightarrow R_x[i] \Delta R_y[i]$ and $\exists i \neq j : (R_x[i] \subset R_y[i] \ \ and \ \ R_x[j] \supset R_y[j])$

The following example illustrates the above definitions listed in the firewall rules shown in Table 1. And the format of the rules in the table upon the format used in Access Control Lists (ACLl) on Cisco routers.

TABLE I. FIREWALL RULES SAMPLE TABLE

	type	ip_s	port_s	ip_d	port_d	action
1	TCP	1. 2. 3.1/4	1024	5.6.7.8	[1,65534]	accept
2	UDP	7.8.9.10	1025	11.12.13.*	90	refuse
3	TCP	11.12.13.*	*	20.21.*.*	*	refuse

III. DIVIDE-AND-CONQUER THE RULES SET

The protocol types of filtering in firewall most are TCP, UDP, ICMP from statistical analysis of large number of firewall rule sets. Moreover, the proportion of TCP and UDP protocols is respectively between 23% to 47%, and the proportion of ICMP achieves between 6% and 19%. Thus, the data packet filtered by firewall which can be transferred in accordance with its transport layer protocol type to a different set of rules to match.

The idea of divide-and-conquer method is used in this section, which divides the rules of the firewall rule set into four categories such as TCP rules, UDP rules, ICMP rules, other transport protocol and no protocol rules, according to the protocol type. Then, the every kind rules are organized together to form four rule groups according to certain order.

Network packets through the firewall, the first of its transport layer protocol type, enter the appropriate rules set rules match. So that each packet matching the rule set is greatly reduced. Its main process is shown in Figure 1, concrete steps are as follows.

The network packets are entered into the appropriate rules set to match rules, when the packets pass through the firewall. So, the rule sets that each packet to match is greatly reduced. The main process is just shown as Figure 1, and the concrete steps are just as follows.

Step1: Determining the protocol type of the data packet.

Step2: Calculating the size of the rule set which is corresponding with the protocol type, and compare with the size of the rule set which is corresponding with the no protocol type, if the result is smaller, then next step enter Step 3, otherwise jump to Step 4.

Step3: Matching with the rule in the rule set which is corresponding with the protocol type, if it is success to match, next step jumps to Step 5, otherwise Step 4.

Step4: Matching with the rule in the rule set which is corresponding with no protocol type, if it is success to match, next step jumps to Step 5, otherwise Step 6.

Step5: Dealing with the packets according to the action which is defined by the rule.

Step6: Dealing with the packets according to the default action.

Step7: End .

IV. RULE MATCHING ALGORITHM

In the firewall system, most of the relationship between the various rules may belong to part independent, cross-correlation, contains relevant and completely irrelevant, but the condition that it belongs to completely irrelevant is very little. As for completely equal rules, they have been removed in the configuration of the firewall. As for two rules whose relationship is completely irrelevant or part independent, they can not successfully match with the same data packet at the same time. Thus these rules in the firewall can been configured in any sequence, and they don't need to match the same data packet in order. While the rules whose relationship is the cross-correlation, and contains relevant can successfully match with the same data packet at the same time. So, these rules match data packets must be in the strict order. According to above, the rules set which are matching packets with the idea of divide-and-conquer have been split into two groups in this section, such as disorderly group and orderly group. And the disorderly group contains all the rules whose relationship are completely irrelevant or part independent, while the orderly group contains all the rules whose relationship are cross-correlation, contains relevant. Just for this, the five relationship domains of rules should be divided into two groups when the firewall is configured, and the packets must be matched in disorderly group before the orderly group when they are matched with rules. The whole matching process is shown in Figure 1.

A. *Algorithm of disordered group*

The matching process in the disorderly group is matching directly to get the rule location by the hash function. The idea of the matching process is that the large number of rules is mapped to a small amount of hash table, and hash collisions are reduced as far as possible to get better-mixed hash table. Thus, in this paper, the four domains (source IP address, destination IP address, source port number, destination port number) of the rule is taken for the input of the hash function. In addition, each rule set has been divided by the transport layer protocol type in the last section, therefore, the protocol type of rule sets in this section is all the same. The protocol type does not be taken for the input

of the hash function, as result for it could not reduce the hash collisions. The main process of the hash function is getting the value of IP XOR Port, while IP is the sum of the source IP address and destination IP address, and Port is the result of the source port number added with the destination port number. Because the length of IP address is 32 bits, we get the 16 bits address by XOR the high 16 bits and the low 16 bits of the address. The main flow of the hash function is just as Figure 2.

Because the length of the hash table is smaller than the size of the rule set, there will certainly be hash collisions. The solution to the conflict: if two rules that have been hashed and the keywords are equal, the two rules can construct one linked list in accordance with the order of insertion sequential. When the data packets are matching the rules, it can be turn along the linked list.

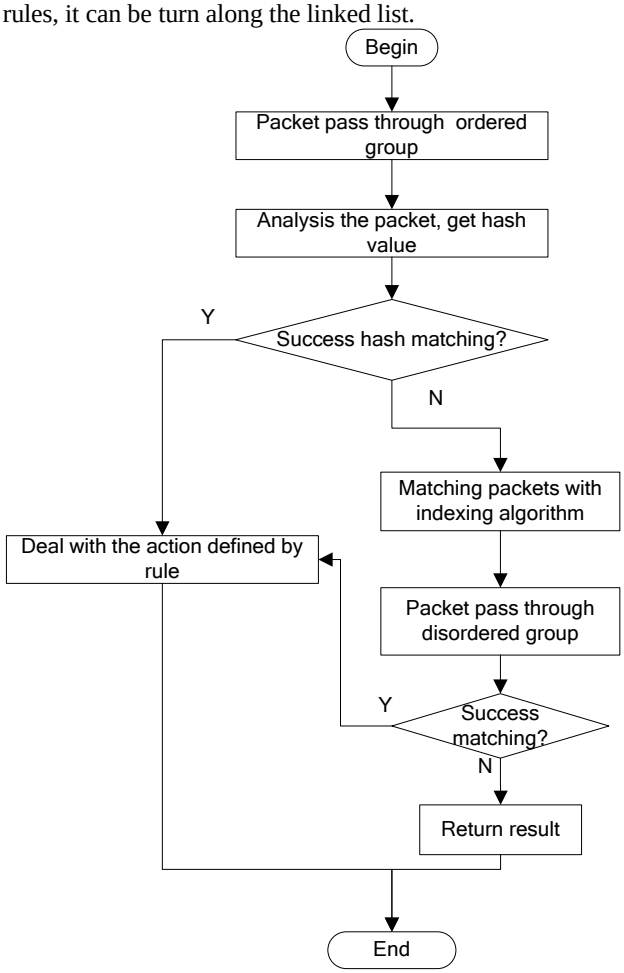

Figure 1. The flowchart of matching algorthim

B. *Algorithm of ordered group*

Because of the strict sequences between every two rules in the orderly group, the rules could not be randomly inserted. Moreover, most of the rules are orderly group, the size of which is very huge. If the linear matching algorithm is used to match the data packet, the efficiency of matching will be very low. Although the relationship of rules in the orderly group is related, it does not mean that any rule in the ordered group is associated with any other rules, just plurality of rules have the correlation that can construct one subset, which is irrelevant with other rules in the orderly group. As a result, the index algorithms can be used to match the data packets in the orderly group. The groups constructed by related rules are arranged in strict sequence accordance with the priority to compose one linked list, which is similar to the single linked list structure. In addition, the linked lists are unrelated to each other; moreover, every linked list can be arranged in any order. Therefore, the speed of matching orderly group can be improved by indexing the linked list. As acquiring one domain in the rule by indexing, the rules are divided into two categories according to the direction of flow, such as the inflow and outflow. General speaking, the inflowing rules concern the destination address, while the out flowing rules concern the source address. Based on this, if the direction of the rules is inflow, the rules can be indexing by the destination address, if the direction is outflow, the rules can be indexing by the source address. The index can be used to sort the table in accordance with the rules of the index key values, corresponding to different rules chain. The indexing value can be used with the key minimum network address in the rules chain. If the rules are extremely abundant, the secondary index or multi-level index can be used to improve the utilization of the index table indexing speed.

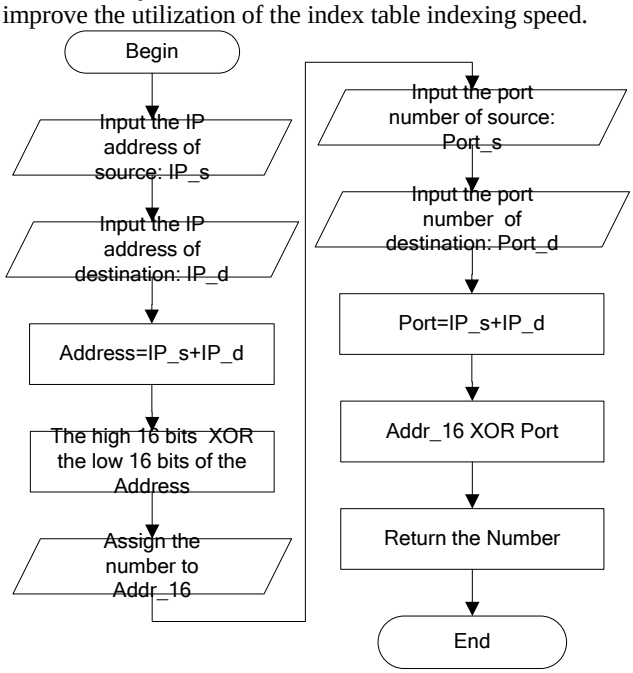

Figure 2. The flowchart of hash function

V. PERFORMANCE ANALYSIS

A. *Divide-and-conquer algorithm performance analysis*

As the divide-and-conquer process is dealing with the rules set in static before the data packets being matched, which will not affect the performance of the firewall. Under

the idea of divide-and-conquer, the size of rule set is significantly reduced, which has laid substantial foundation for the improvement of the matching algorithm.

B. Matching algorithm performance analysis

1) Disorderly group matching algorithm performance analysis

Hash function is mainly constructed by addition, XOR, AND, SHIFT and modulo operation, which are all cost CPU disposable calculation, therefore, execution of hash is only need to perform twice addition, XOR and AND operation and once modulo operation. Thus, the efficiency of the algorithm is very high, and the CPU occupancy rate is very low. The addition of IP address and port in the algorithm satisfy the commutative law, and the entering of the hash function in the data packets send from the two sides of the communication are the same, so we can use the same value to avoid secondary hash calculation to save the system overhead, and improve efficiency of the firewall matching. The storage space of the hash result is 16 bit, which can be expressed hash range is 0 to 65535. So, the value much larger than the size of the general rule set, the probability of hash collision is very small.

2) Disorderly group matching algorithm performance analysis

The rule matching of ordered group is based on the indexing algorithm. The index table of indexing algorithm is sorted by the address in the rules, when the data packet is matched in the ordered group, the algorithm is matching the data packet in accordance with the address from the index table for binary query, its time complexity is $O(\log_2 N)$ (N is the size of ordered group). After the index has been get by Binary query, the related rule chain can be found, and sequential matching in it. Generally, the rule chain is relatively small, typically only two to three rules.

3) The algorithm integrated performance analysis

To sum up, the performance of the matching algorithm designed in this paper is mainly focus on the correlation between each rule. When the correlation of rules is relatively small, most rules can be categorized into disorder group, namely the size of disordered group >> the size of ordered group, and the matching efficiency is nearly equivalent to the hash matching efficiency in disordered group. In addition, the control of particle size in firewall is generally as sites, the relationship between each other is relatively small; normally, the size of disordered group is much larger than the ordered group. On the other hand, when the size of ordered group >> the size of disordered group, the comprehensive performance of the algorithm is close to $O(\log_2 N)$.

VI. CONCLUSION

Based on the in-depth analysis of the relationship between every two firewall filtering rules, the rule sets are classified by the type of protocol in this paper, which greatly reduced the scope of the packet matching rules, and then the subsets are divided into a disorderly group and orderly groups, then, the hash algorithm is designed to match data packets with the rules in disorderly group, while the indexing algorithm for the rules in orderly groups. Above research greatly improve the matching efficiency, so that the firewall work has been greatly optimized. Next, the main subject of the expand firewall research is the renovation of the correlation between rules, which can transform the associated rules to independent rules, so that all of the rules can be carried to the disordered hash matching.

REFERENCES

[1] Yadi Ma, Suman Banerjee. “A smart pre-classifier to reduce power consumption of TCAMs for multi-dimensional packet classification,” SIGCOMM 2012, ACM Press, Dec. 2012, pp. 335-346, doi: 10.1145/2342356.2342428.

[2] Chih-Hsun Chou, Fong Pong. “Speedy FPGA-based packet classifiers with low on-chip memory requirements,” FPGA 2012, ACM Press, Feb. 2012, pp. 11-20, doi: 10.1145/2145694.2145697.

[3] Chih-Hsun Chou, Fong Pong. “Cache Architecture for High-Speed Multidimensional Packet Processing,” Sixth Internet Computing for Science and Engineering (ICICSE2012), April. 2012, pp. 60-65, doi: 10.1109/ICICSE.2012.21..

[4] Hyogon Kim, Sunil Kim, Moon Hae Kim. “Scalable packet classification through rulebase partitioning using the maximum entropy hashing”. IEEE/ACM Transactions on Networking (TON), vol. 17, Dec. 2009, pp. 1926-1935, doi: 10.1109/TNET.2009.2018618.

[5] GUPTA P, MCKEOWN N. “Packet classification using hierarchical intelligent cuttings,” IEEE Micro, vol. 20, Feb. 2000, pp. 34 – 41, doi: 10.1109/40.820051.

[6] Rihua Wei, Yang Xu,Chao, H.J. “Block permutations in Boolean Space to minimize TCAM for packet classification, ” INFOCOM, 2012 Proceedings IEEE, May. 2012, pp. 2561 - 2565, doi: 10.1109/INFCOM.2012.6195653.

[7] TAYLOR D, TURNER J. “Scalable packet classification using distributed cross producing,” IEEE Micro, vol. 90, Feb. 2006, pp. 49 – 60, doi: 10.1109/40.83640721.

[8] CR Meiners, AX Liu, E Torng. “Bit Weaving: A Non-prefix Approach to Compressing Packet Classifiers in TCAMs,” IEEE/ACM Transactions on Networking (TON), vol. 20, April. 2012, pp. 488-500, doi: 10.1109/TNET.2011.2165323D.