Journal of Engineering Mechanics and Machinery (2016) 1: 34-39
Clausius Scientific Press, Canada

A Real-time IP Packet Flow Match Algorithm Based on Two-layer XOR Hash and TCAM

Zhen Zuo[1,a], Xin Pang[1,b] and Zhiping Huang[1,c]

[1]College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, China

[a]673213286@qq.com, [b]zuozhenwork@163.com, [c]kdhuangzp@sina.com

Keywords: Flow match; Real-time performance; Hash function; Flow measurement

Abstract: In the field of real-time flow measurement for high-speed network, the performance of real-time IP packet flow match algorithm determines the performance of the measurement to a large extent [1]. By analyzing the Hash function, a real-time IP packet flow match algorithm RFMA-HT based on two-layer XOR Hash and TCAM is put forward. Test results show that the RFMA-HT algorithm can meet the real-time performance of flow match in high-speed network and can greatly reduce the conflict rate in the meantime.

1. Introduction

Network measurement is the process of testing and verifying the indexes which can represent the performance of network by the certain method and technology, using software and hardware tools [2]. Network measurement is an important auxiliary means of network management, the basis for rational allocation of resources, the requirements for improving Quality of Service, and an important method for ensuring network security. In the field of network measurement, it is common to compress the network packets to IP flows according to a certain standard and measure the IP flows continuously. To measure the IP flows, we first need to match the incoming packets to IP flows. The performance of the flow matching algorithm is the key to measure the high-speed network flow accurately.

The traditional methods for real-time flow match of IP packet include IP packet classification algorithm, state detection technology, BPF algorithm, SRL language, multi-domain packet classification algorithm, TCAM device and so on. But these methods cannot be directly used for flow match in high-speed network because they do not support large matching rule sets and they take long time to match and occupy large memory. In order to match the massive and high-speed data packets, the real-time flow match algorithm has become a research hotspot. Normal Hash function has so high conflict rate that it needs to be improved. In this paper, a real-time IP packet flow match algorithm RFMA-HT based on two-layer XOR Hash and TCAM is proposed [3], which can meet the real-time performance and reduce the conflict rate.

2. Problem analysis

In order to realize the real-time performance of flow match, the time complexity of the match algorithm must be $O(1)$. Hash function can complete the query match with the time complexity of

$O(1)$ by mapping a binary value to another binary value called hash value which has shorter length. It is called conflict if $f(a) = f(b)$ after performing hash operation $h = f(x)$ to the binary values a and b ($a \neq b$) [4].

Under certain constraint of timeout policy, a set of IP packets with the same IP packet head is called an IP flow. Each IP packet head (source address, destination address, source port, destination port, protocol number) has a total length of 104 bits. Considering that the type of IP packet protocol is limited and more than 99% packets are TCP and UDP packets, we commonly do not perform hash operation on protocol number which accounts for 8 bits. Therefore, only the source address, destination address, source port and destination port of the IP packet are involved in the hash operation.

When each IP packet arrives, hash function can convert the 96-bit IP head to an a-bit hash value as the ID of this IP packet. So it can achieve query match of a-bit address lines with time complexity $O(1)$. However, there must be conflicts in this hash function, which means two different flows have the same a-bit address so that they enter the same spatial region of the memory. In order to achieve the query match with time complexity of $O(1)$, we must open up the second layer of storage space and operate the second hash function for the IP packets that cannot enter the first layer of storage space because of conflicts. But there are still conflicts in the second layer of storage space, so multiple layers of storage space may are required to reduce the number of conflicting IP packets. No matter how many times hash function there are, we cannot guarantee 0% conflict rate in theory. But the number of IP packets that conflict after many times of hash operation is very few, so we can use TCAM device whose time complexity is $O(1)$ to complete the query match [3].

Through the above analysis, we can obtain that hash function is operated at least once to each IP packet. Therefore, the complexity of the hash operation itself should be as low as possible. In order to reduce the number of hash operations and compare operations, we should minimize the conflict rate of hash function.

3. Hash function selection

Hash function has to meet the certain conditions. (1) In order to reduce the conflict rate of match algorithm, we must choose a hash function that has high degree of randomness. (2) In order to improve the match instantaneity, we have to select a low-complexity algorithm to realize the hash function, such as simple bit operations including Non-operation, and operation, or operation and XOR operation. In order to measure the conflict rate of the algorithm, the concept of random degree of hash function is adopted in this paper [5]. A certain bit x of IP packet can be 0 or 1 in random. Assuming the probability of being 1 is p, then the entropy of is defined as

$$H(x)=-(p\log_2 p+(1-p)\log_2(1-p)) \tag{1}$$

When $p = 1/2$, $H(x)$ reaches the maximum value $H_{max}(x) = 1$. The degree of randomness of x is defined as

$$R(x)=H(x)/H_{max}(x)=H(x) \tag{2}$$

After calculation, XOR operation can improve the degree of randomness of hash function. To verify the calculation results, we carried out an experiment in which we perform XOR operation to the destination port and the source port of the IP packets collected from the Equinix-Chicago link [6]. It is shown from Fig.1 and Fig.2 that the probability of each bit of the destination port and the source port being 1 fluctuate with the bit and the randomness is relatively low. After the XOR operation, the probability of each bit being 1 fluctuates slightly at 1/2 and the randomness of each bit fluctuates slightly at 1, which indicates the randomness of the data is greatly improved. Since the randomness of the port number is particularly poor in some locations, XOR operation once cannot improve the randomness of all bits. We can compensate that by a plurality of XOR operations.

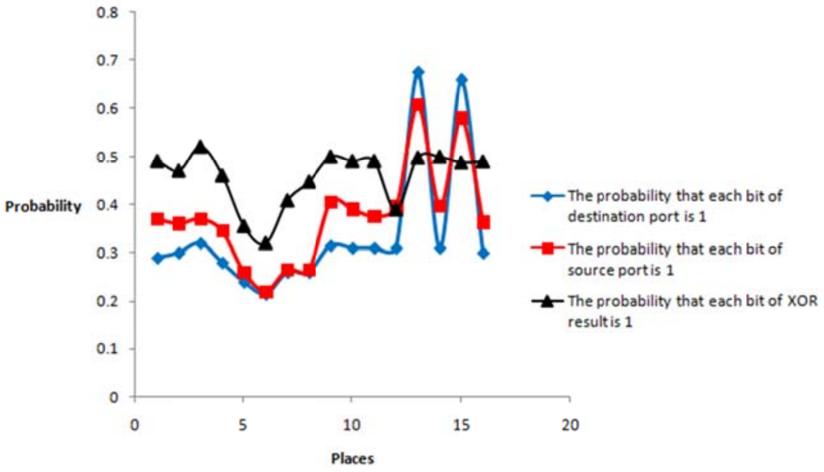

Fig.1 The probability that each bit of destination port/source port and XOR result is 1

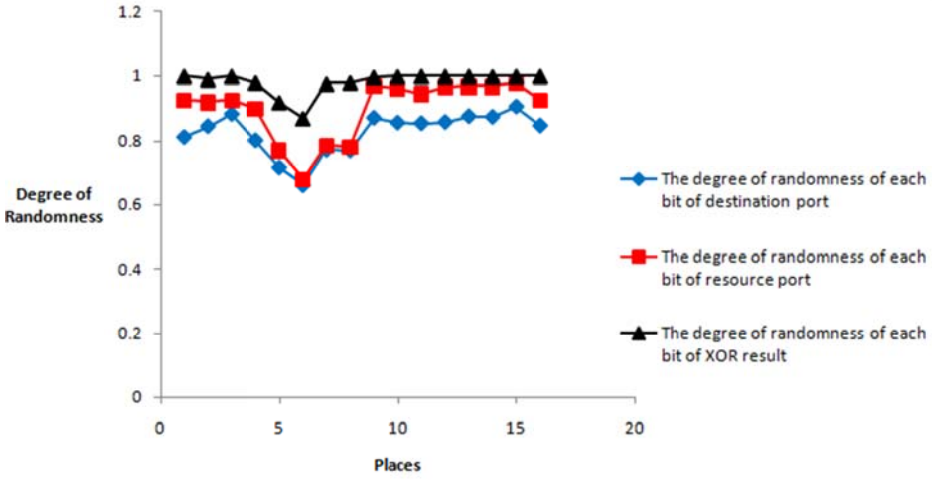

Fig.2 The degree of randomness of each bit of destination port/source port and XOR result

4. The proposed RFMA-HT algorithm

4.1 The overall design scheme

It is calculated that the number of IP flows that are still conflicting after hash function twice is smaller than the maximum match number of TCAM device in the optical fiber backbone network with the number of current flows in the order of millions. The overall design scheme of RFMA-HT is shown in Fig.3.

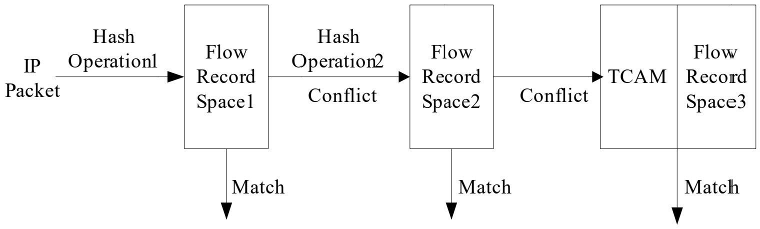

Fig.3 The overall program of RFMA-HT

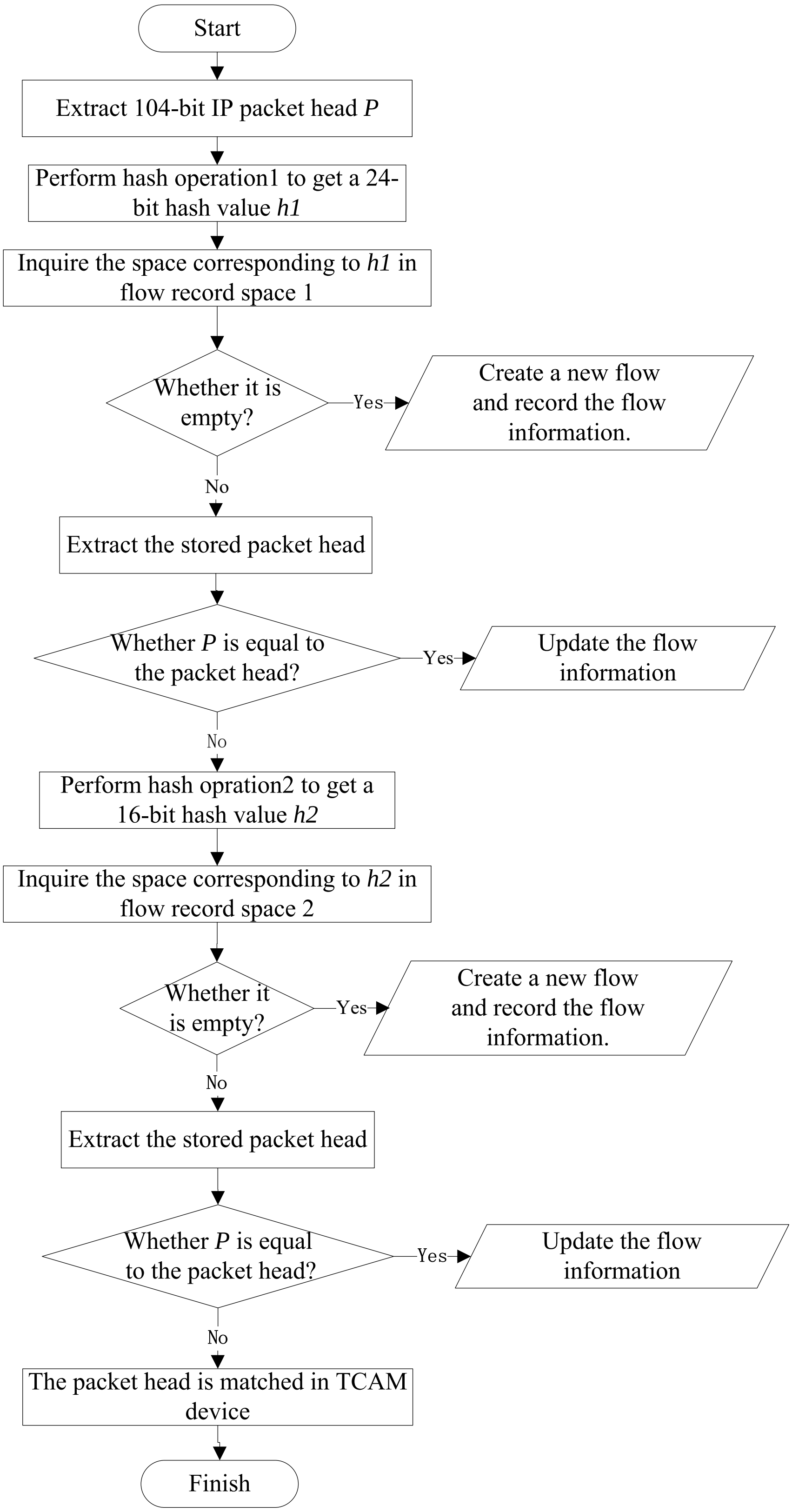
Start
Extract 104-bit IP packet head *P*
Perform hash operation1 to get a 24-bit hash value *h1*
Inquire the space corresponding to *h1* in flow record space 1
Whether it is empty?
Yes
Create a new flow and record the flow information.
No
Extract the stored packet head
Whether *P* is equal to the packet head?
Yes
Update the flow information
No
Perform hash opration2 to get a 16-bit hash value *h2*
Inquire the space corresponding to *h2* in flow record space 2
Whether it is empty?
Yes
Create a new flow and record the flow information.
No
Extract the stored packet head
Whether *P* is equal to the packet head?
Yes
Update the flow information
No
The packet head is matched in TCAM device
Finish

Fig.4 Flow chart of RFMA-HT algorithm

We extract 104-bit IP packet head of the collected IP packets and perform the first hash operation to the specific 96 bits and obtain a 24-bit hash value. This hash value is matched in the first flow record space. If there is a conflict in the process of matching, we perform the second hash function to the IP packet head to obtain a 16-bit hash value, which is matched in the second flow record space. If there are still conflicts after these hash operations, the IP packet head will be matched through the TCAM device.

4.2 Implementation of RFMA-HT

In RFMA-HT algorithm, we conduct xor hash operation to the source address, destination address, source port and destination port that extracted from the IP packet head and obtain the hash value. The hash value is matched in the flow record space as an address: check whether there is a flow in the address corresponding to the hash value. If not, the IP packet head information is recorded. If there is a flow, compare the information of this flow and the IP packet. If they are equal, update the flow information. If not, it indicates that there is a conflict. Then conduct the second hash operation. The remaining conflicting IP packets are matched in the TCAM device by IP packet head directly(Shown in Fig 4).

4.3 Feasibility analysis of the algorithm

IP packets need to match the flow by and operations, xor operations, shift operations, add operations, and compare operations. Some IP packets also need to be matched by TCAM devices. Although multiple operations are required for each IP packet, it is only one clock cycle per IP packet to complete the real-time flow match with parallel pipelining. The overall time complexity of the RFMA-HT algorithm is $O(1)$.

In the 40 Gbit/s fiber backbone link at full load, it takes 8ns to process the IP packet with the shortest packet length of 40 bytes. The RFMA-HT algorithm can meet the match need in the fiber backbone network as long as the memory frequency up to 125MHz. At present the SRAM memory frequency has generally reached more than 200MHz. In addition, the RFMA-HT algorithm reduces the time complexity by increasing the space complexity, and the required large-capacity memory space can be obtained by multiple memory cascades.

5. Experiment and test results

The basic parameters of the experimental data are shown in Table 1.

Table 1.The parameters of the experimental data

Trace	Speed	Duration	IPv4 packet number	Flow number
Equinix-Sanjose(Direction A)	10Gbit/s	60s	22374606	954374

The test results are shown in Table 2.

Table 2.The test results

	Flow number	Proportion	IP packet number	Proportion
Cache1	913653	95.73%	22046970	98.54%
Cache2	30521	3.20%	247517	1.11%
Cache3	10200	1.07%	80119	0.36%

It can be seen from Table 2 that RFMA-HT identifies all the IP packets and match all 22374606 IPv4 packets into 954374 flows. The numbers of the flow and IP packet matched in the first flow record space take proportion of 95.73% and 98.54% respectively. The proportions are 3.20% and

1.11% respectively in the second flow record space. They are 1.07% and 0.36% respectively in the third flow record space. The first hash function has matched more than 98% of the IP packet, which means the conflict rate is relatively low. Since the bit of the hash value in the second hash function is shorter than that in the first hash function, the conflict rate in the second hash function is relatively high. However, the number of IP packet that needs to be matched in the second hash function is so small that it has little impact on the overall conflict rate.

6. Conclusions

This paper put forward the RFMA-HT algorithm directing at IP packet flow match. Test results show that the RFMA-HT algorithm has lower computational complexity and lower conflict rate, which can meet the requirements for IP packet real-time flow matching in optical fiber backbone network.

References

[1] Hong Tang, Yongjun Wu and Guofeng Zhao: Research on Stochastic Matrix Mapping Hash for Specific Flow Matching, Journal on Communication, 2007(02), p17.

[2] Guoping Ren: Netwok Measurement Review, Scientific and Technology Information: Academic Research, 2008(04).

[3] Zhen Zuo: Research on the Key Technologies of Real-time Flow Measurement on Optical Fiber Backbone Network, National University of Defense Technology, 2012.

[4] Guang Cheng, Jian Gong and Wei Ding: A Hash Algorithm for IP Flow Measurement, Journal of Software, 2005(05), p652.

[5] Bogang Lin: Network and Information Security, Beijing: China Machine Press, 2004(in Chinese).

[6] Cooperation Association for Internet Data Analysis (CAIDA), http://www.caida.org.