The Hash Algorithm of the Network Equipment Performance Evaluation

Xin Zou[1], Li Zhou[1] and Liangyi Gong[2]

[1]*The National Computer network Emergency Response Technical Team Coordination Center of China, P. R. China*
[2]*Harbin Engineering University, P. R. China*
zouxin@cert.org.cn, zhouli@cert.org.cn, gongliangyi@hrbeu.edu.cn

Abstract

At present the hash algorithm analysis is mostly confined to the randomness of the hash values. Researchers have not analyzed the effect of hash algorithm in the network equipment. We propose three evaluation measures of hash algorithm from the perspective of theory analysis and experimental evaluation: the output stream integrity by combining with the dictionary sorting and binary search method to verify, the flow distribution uniformity of hash algorithm, load balancing effect according to the group packet number variance and the max-min packet number ratio. In the experiments, we evaluate the hash algorithm of the current domestic well-known network equipment manufacturers with the backbone network flow. The results show that the evaluation measure can effectively measure hash algorithm. It is helpful for network equipment manufacturers to choose proper hash algorithm.

Keyword: *hash algorithm, stream integrity, load balance, network equipment, evaluation measures*

1. Introduction

With the expansion of the network scale and the development of the application technology, the network flow is also increasing fast. Because the cluster node processing ability is limited, it needs to shunt automatically the high-speed network packets to multiple low-speed network packets.

Backbone network often needs to set data stream distribution rules to distribute the coming message fast and timely. Hash algorithm has many advantages, such as short pretreatment time, low memory consumption and supporting matching rules, *etc.* [1]. So the hash algorithm is widely used in the network equipment, such as ECMP, Port Trunking, Network probe.

Hash algorithm plays an important role in managing the data flow. The evaluation of Hash algorithm provides references for network devices. In the papers [2, 3], authors analyzed five kinds of Hash algorithms based on IP flow field and classified them into two types: one directly used identification field in a message [4], it improved the efficiency but the hash value of the random effect is poor; another one used hash functions to compute hash value [5], good hash function could improve the efficiency and ensured higher randomness of the has value. In the paper [6], researchers analyzed the compute speed of the flow hash functions. In the paper [7], it proposed the random measure for evaluating the performance of hash algorithm, and theoretically proved that it could improve the randomness of hash value using exclusive-or operation and shift operation between bits, then put forward the principle for bit flow hash algorithm. In the paper [8], researchers put forward the hash algorithm evaluation standard from two aspects: memory access efficiency and balanced performance, then

proposed a CRC20 Hash algorithm based on flow quintuple. In the paper [9], it conducted a comparative study on flow hash function, used uniformity, collision rate and active flow estimation and computing speed to evaluate the property of hash functions, and gave the basic theory about choosing hash function through experiment comparison.

At present, the Hash algorithm analysis is based on Hash algorithm performance of the stand-alone. The researchers just considered the randomness of the algorithm itself, memory usage, computing speed and other factors, while the network shunt equipment characteristics was not considered. The demand of the network shunt equipment characteristics contains protocol flow, the output of stream integrity (the same TCP flow data packets go into the same processing unit), load balancing, session management and filter forwarding etc. A good Hash algorithm can not only meet the high randomness but also meet the need of the network shunt equipment characteristics described above.

We propose the three new evaluation measurements according to the characteristics of the network shunt equipment and Hash algorithm evaluation: (1) output stream integrity; (2) flow distribution uniformity; (3) the load balancing of the algorithm. In this paper, we use dictionary sorting to test the stream integrity (流同源同宿), evaluate the flow distribution uniformity of the algorithm combining with the max-min packets number to evaluate the load balancing of the algorithm. Through using the backbone network flow to evaluate the current domestic well-known network equipment manufacturers of hash algorithm, the experiment result shows that our method can effectively evaluate the hash algorithm in the network shunt equipment.

2. Definition

Traditionally, it uses quintuple to define network flow: source IP address (SIP), destination IP address (DIP), source port (SPORT), destination port (DPORT), protocol (PROTOCOL). If the quintuples belong to same flow [10], the source IP and destination IP transport monodirectional message stream in a period. In most processes, it needs bidirectional message stream to restore the protocol, information processing etc. So it needs to consider the integrity of the stream when network devices shunting.

Definition 1 (uniflow) Each message has a quintuple G5=[sip,dip,sport,dport,ptl], the source(or client) direction flow is defined as G5s=[sip,dip,sport,dport,ptl], destination(or server) direction flow is G5d=[dip,sip,dport,sport,ptl], where sip is the source(or client) direction IP address, sport is the source port, dip is the destination(or server) direction IP address, dport is the destination direction port, ptl is the type of protocol.

Definition 2 (stream integrity) Each message is defined as p, the source(or client) direction data flow message of the same session is p_s, destination direction data flow message is p_d, the data set is $M = \{p_1, p_2, \ldots, p_i, \ldots, p_n\}$, where $p_i \in \{p_s, p_d\}$ and n indicates the number of the messages. The output set is $F = f(M) = \{R_1, R_2, \ldots, R_t\}$, where $R \subseteq M$, $i \in [0,t]$, t is the number of group, and $t \geq 1$, f() is the separate function.

If $\forall p_s \in M$ then $p_s \in R$, $i \in [0,t]$ ①

If $p_s \in R, \exists p_d \in S$, $i \in [0,t]$ then $p_d \in R$ ②

If $\forall R_1, R_2, R_1 \in F, R_2 \in F$ then $R_1 \cap R_2 = \varnothing$ ③

If $F = f(M) = \{R_1\}$ then $R_1 = M$ ④

If the separate function meets all the conditions, then the function meets the stream integrity.

Define 3 (packet number ratio) Assume that there are n numbers for 0 ~ (n-1), time period is in $[\tau, t]$, when flow is ended, the receiving packet number of each group are $X_0, X_1, \ldots X_{n-1}$. The average of each group is: $X_{avg} = (\sum_{i=0}^{n-1} X_i) / n$, the variance of each group is $V_i^2 = (Xi - Xavg)^2$. The overall variance is $V = \sum_{i=0}^{n-1} V_i^2 / n$.

The maximum packet number ratio is defined as $P_{max} = Max(X_i) / X_{avg}$, the minimum packet number ratio is $P_{min} = X_{avg} / Min(X_i)$. The maximum packet number ratio and minimum packet number ratio indicates the load balancing of the Hash algorithm.

3. The Evaluation Measures of Hash algorithm

3.1. Stream Integrity verification

Because of the load balancing of the network shunt equipment and the routing asymmetry, it would lead to a series of messages belong to a session distributed to different link, while the back-end server needs to collect the whole session message to complete data processing. The stream integrity guarantees the integrity of the session message and completes the normal message processing.

According to Definition 2, we can know that the output stream integrity must to meet conditions ①②③④. But in the actual network devices algorithm, the Hash function has mapping characteristics, conditions ①, ③, ④ satisfy. It just needs to verify the hash algorithm meets the condition ②, just proving hash function satisfy:

$$Hash(G_{ss}) = Hash(G_{sd}) \qquad (3.1)$$

Such as a network shunt equipment hash function is as follows:

```
Hash5(sip,dip,sp,dp,ptl){
    dsp=dip ^ sip
    dsp_h16 =(dsp&0xffff0000)>>16
    dsp_l16 =dsp&0x0000ffff
    dspt_h16_s=dsp_h16 ^ sport
    dspt_l16_d=dsp_h16_s ^ dport
    dsp16= (dspt_h16_d ^ dsp_l16) ^ ptl
    dsp_h=( dsp16& 0x0f00)>>8
    dsp_m=( dsp16& 0x00f0)>>4
    dsp4=dsp_h ^ dsp_m
    dsp12=dsp_h | (dsp4<<8)
    hash_value= dsp12>>4
}
```

Due to the shift (>>,<<) operation, AND (&) operation and constant in arithmetic operations, it will not affect the stream integrity. So Hash5 can be simplified as:

Hash5(sip,dip,sp,dp,ptl)

$$= \begin{pmatrix} (dip \oplus sip) \oplus (sp \oplus dp) \\ \oplus (dip \oplus sip) \oplus ptl \end{pmatrix} \vee \begin{pmatrix} \begin{pmatrix} (dipsip) \oplus (sp \oplus dp) \\ \oplus (dip \oplus sip) \oplus ptl \end{pmatrix} \\ \oplus \begin{pmatrix} (dipsip) \oplus (sp \oplus dp) \\ \oplus (dip \oplus sip) \oplus ptl \end{pmatrix} \end{pmatrix}$$

Where, a=$(dip \oplus sip) \oplus (sp \oplus dp) \oplus (dip \oplus sip) \oplus ptl$

According to the exclusive or operation meeting Commutative law, (A xor B) xor C= A xor (B xor C).

Hash5(sip,dip,sp,dp,ptl)=

$$a \vee (a \oplus a) = a \vee 0 = a$$

$$= (dip \oplus sip) \oplus (sp \oplus dp)$$

$$\oplus (dip \oplus sip) \oplus ptl$$

$$= (sip \oplus dip) \oplus (dp \oplus sp)$$

$$\oplus (sip \oplus dip) \oplus ptl$$

=Hash5(dip,sip,dp,sp,ptl)

So the Hash algorithm on network device meets the requirement of the stream integrity, formula 3.1.

The famous example of searching hash table is series of CRC algorithm. The famous nginx system adopted CRC32 [11], the algorithm is operated in bytes, while the byte sequence of the G5s and G5d is different, Hash(G5s) $\neq$ Hash(G5d). So CRC32 does not meet the requirements of stream integrity.

We can use the method as follow to verify the stream integrity: First, sorting each group message quintuple record according to the dictionary sort, then using the binary search method to Locate any a record in other groups. If a record appears in two or more groups at the same time, it judges the hash algorithm stream integrity is wrong.

3.2. Flow distribution uniformity of the Hash algorithm evaluation

Generally, the expectation of the hash value of the hash function should be evenly into hash space, and avoid gathering. According to definition 3, the variance of each group is $V_i^2 = \left(Xi - Xavg\right)^2$. The overall variance is $V = \sum_{i=0}^{n-1} V_i^2 / n$.the overall variance reflects the deviation between the samples and theory distribution. The standard deviation $\sigma = \sqrt{V}$ is used to measure if the hash algorithm split evenly. The value is smaller, the result is better.

3.3. Load balancing of the hash algorithm evaluation

A load balancing equipment often contains a flow separator and several output links. As Figure 1 shows, the flow separator receives data packets from high speed network then

distributed them to the low speed links. A good load balancing model for the network equipment should distribute the flow evenly or allocate them as the value predefined.

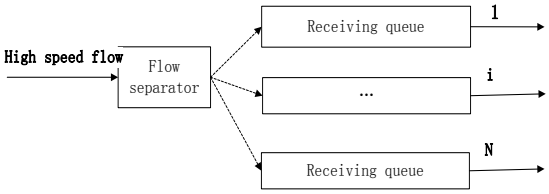


Figure 1. Load Balancing Model for the Network Equipment

Assume that there are N output links, the capacity of each link i is μ_i, $S_i(\tau, \iota)$ is the transmission quantity from flow separator to link i in the period $[\tau, \iota]$. An ideal load balancing equipment should meet the rule [12] in any period$[\tau, \iota]$:

$$\frac{S_i(\tau, \iota)}{S_j(\tau, \iota)} = \frac{\mu_i}{\mu_j} \tag{3.2}$$

The capacity of each link is consistent in the network devices, so an ideal network load balancing equipment should also meet the rule [12] below:

$$S_0(\tau, \iota) = S_1(\tau, \iota) = \cdots = S_{n-1}(\tau, \iota) \tag{3.3}$$

But it can't match the ideal load balancing in actual situation. We use standard deviation and packets number ratio to comprehensively measure the result of load balancing. The smaller of the values of P_{max} and P_{min}, the better of the result of load balancing. But the standard deviation of load balancing can not bigger than the upper limit.

The estimation of load balancing is calculating the index that is the number of packets which each group received every t seconds, and then drawing the curve of the ratio of the standard deviation and packet number according to time. We can estimate the hash algorithm with the curve. The smaller of the standard deviation, the better of the load balancing result of the hash algorithm.

For example 1, the current flow is about 150000pps, assume that computing the index in every t=5 seconds, and then separates the packet into n=16 groups. The number of packet which each group received can not be more than or less than twenty percent of the average, so the P_{max} can not be more than 1.2, while P_{min} can not be less than 1.25, the upper limit of estimated load balancing variance is $\sigma \approx \sqrt{(\frac{150000 \times t}{n} \times \frac{1}{5})^2 \times n / n} = \frac{3000 \times t}{n}$ =9375, and σ can not be more than 9375.

4. Experiments and Analysis

The comparison result of hash function on equipment is got by using the three evaluation measures this paper proposed, the analysis data is from CERNET. First, we verify the stream integrity of the Hash algorithm on network devices from five manufacturers(HUAWEI, ZTE, DPtech, Sugon, Ebright) in China, random evaluation and load balancing according to the configure file , and then give the statistical analysis results, the experimental platform framework shown as Figure 2.

The evaluation adopts the specified card single thread capturing, and comparing the number of network card capturing packets and the number of routers sending packets, verifying the correctness of the capturing interface. We capture 200 million packets as sample evaluation while ensuring network data flow is no packet loss under the condition of validation capture. The data information is as follows: sampling time 35m55s, the number of packets is 200000000, average speed: 92807pps.

We compute the packet number of each group and its standard deviation, the number of packets ratio every 5 seconds. The evaluation proposes three indexes, computes the upper limit of these indexes. Where the time interval t=5s, group number n=16, the standard deviation upper limit of the load balancing algorithm is 5800, the max packet number and the min packet number ratio is 1.2, average packet number and min packet number ratio is 1.25.

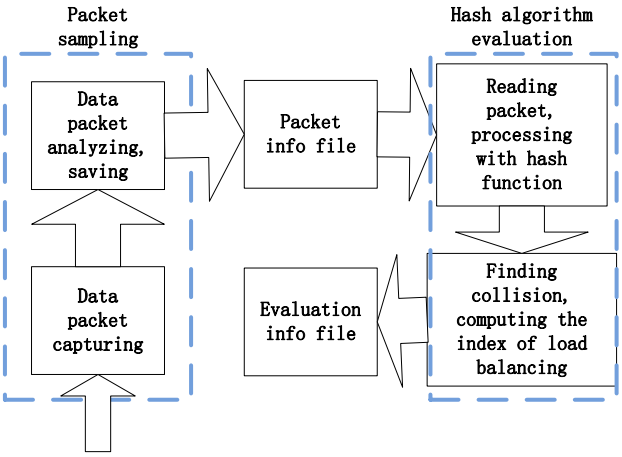


Figure 2. Experimental Platform Framework

4.1. Stream Integrity

We evaluate the 16 groups of the hash algorithm's quintuple in network equipment of five manufacturers,as A1, A2, A3, A4, A5 according to stream integrity authentication algorithm. We use the dictionary sorting method to judge the stream integrity.

```
For each packet in test-set Do
groupid = Hash(packet);
   If groupid < 1 or groupid > max_group_num Then
         Report a error!
Endif
group[groupid].add(packet);
   For i= 1 to max_group_num Do
//if the packet is founded in other groups, there is something wrong with stream integrity.
If i != groupid and FindPacketInGroup(packet, i) == 1 Then
          Report a error!;
  End If
    End For
End For
```

The experiment results show that the all the hash functions are accord with stream integrity correctness devices.

4.2. Flow distribution uniformity of the Hash algorithm evaluation

According to distribution uniformity evaluation algorithm, recording the number of 16 groups received data packets every 5 seconds in collected period 35m55s. And then calculate the variance of each group data. The experimental results are shown in Figure 3. Because the value of variance smaller, the result is better. The standard deviation of A3 is much more than others. The distribution uniformity effect of A3 is poor, while A1、A2、A4、A5 is less the same. From Table 1, the distribution uniformity effect of five equipments Hash algorithm evaluated from excellent to poor is A2、A1、A5、A4、A3.

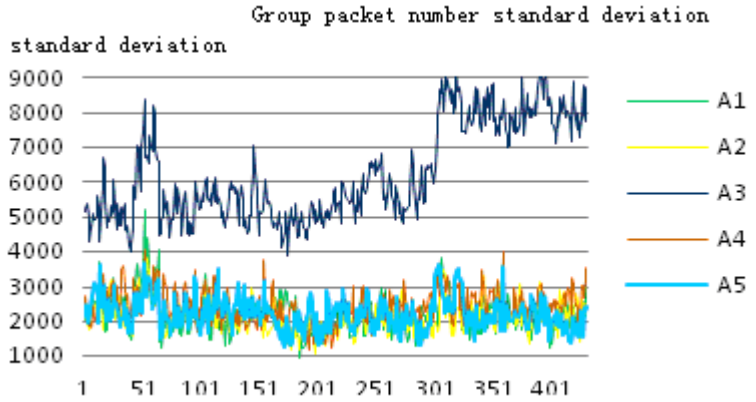


Figure 3. Equipment Hash Algorithm of Distribution Uniformity Evaluation in IPv4 Traffic

Table 1. Average Variance Values of Equipment Hash

facility	A1	A2	A3	A4	A5
Standard deviation	2148	2108.	6052	2368	2160

4.3. Load balancing of the hash algorithm evaluation

Estimate the ability of Hash algorithm load balancing in the IPv4 flow. According to the load balancing evaluation algorithm, in the collected period 35 minutes 55 seconds, in every 5 seconds, we recorded 16 groups receive package. Then, calculate the variance, the maximum number of packages ratio, the minimum number of packages ratio for the data in every group.

According to the example of the 3.3,if the upper limit of estimated load balancing variance is less than 5800 and the max packet number and the min packet number ratio is 1.2, average packet number and min packet number ratio is 1.25, the load balancing effect is good .The experiment results as shown in figure 4-6,the load balancing effect of A3 is very poor, the load balancing effect of A1, A2, A4, A5 is good and just little differences, we can evaluate Hash algorithm load balancing effect in the five facilities from excellence to bad ： A2, A1, A5, A4, A3.

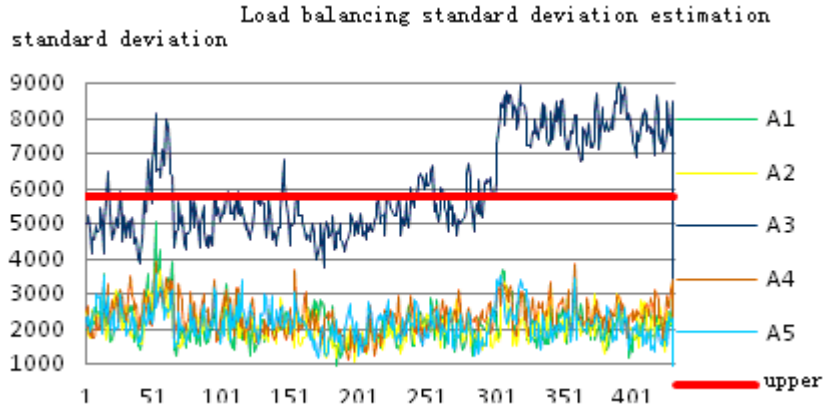


Figure 4. Hash Algorithm Load Balance Standard Deviation Evaluation of Facility

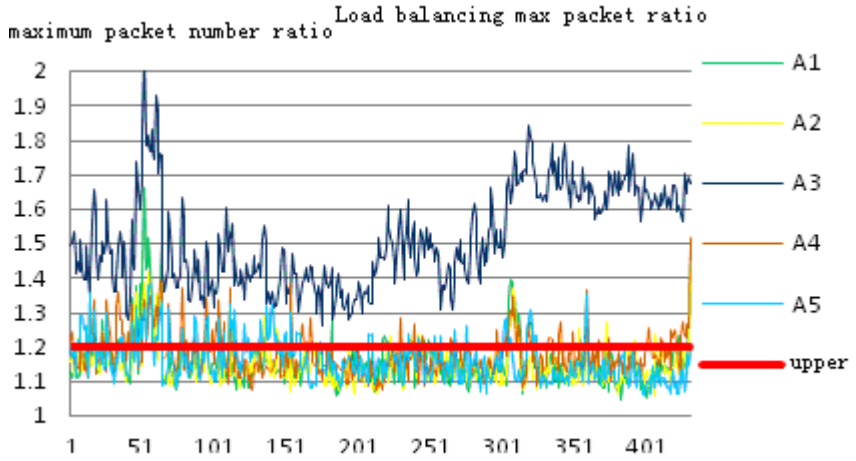


Figure 5. Hash Algorithm the Maximum Number of Packages Ratio Evaluation

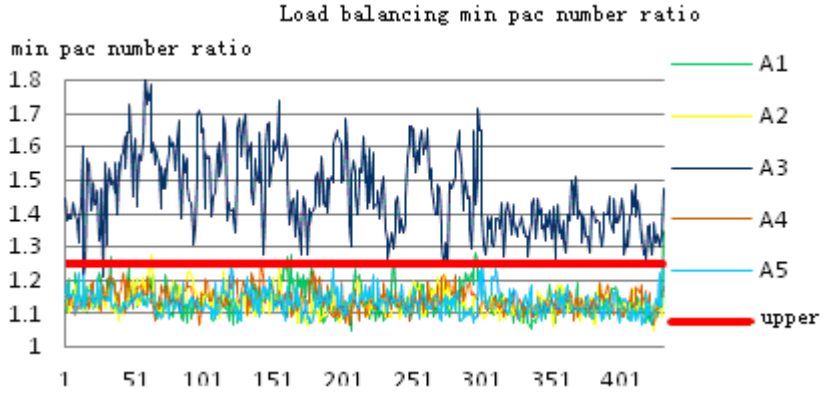


Figure 6. Hash Algorithm the Minimum Number of Packages Ratio Evaluation

Table 2. The Average of Hash Function Load Balance Quota of Facility

facility	standard deviation	the maximum number of packages ratio	the minimum number of packages ratio
A1	2148.525	1.160551	1.139112
A2	2108.323	1.151409	1.134023
A3	6052.576	1.51617	1.46226
A4	2368.904	1.17846	1.142281
A5	2160.207	1.163312	1.134102

5. Conclusions

The experiments show that the hash algorithm assessment method and scheme proposed in this paper is simple, practical, objective and accurate, it can effective assess the hash algorithm in network equipment. From the experimental results, we can learn that the effect of hash algorithm in the major network equipment is better under the traffic of IPv4.With the continuous expansion and popularity of the IPv6 network, the research of hash algorithm on IPv6 flow is less in current domestic, so it is need to study and research the network equipment hash algorithm on IPv6 flow in the future.

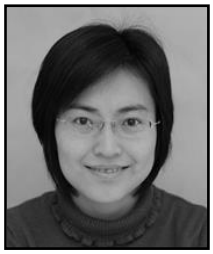
References

[1] G. Zhao and L. Yan, "Keywords Decompound Hash Algorithm for Quick Flow Classification", Computer Engineering.August, **(2010)**.
[2] LiuYan-Hua, ChenGuo-Long and HhuangQiao-Yu, "Research on IP packet dynamic data-distribution based on Hash algorithm", Journal of Fuzhou University (Natural Science Edition), **(2009)**.
[3] R. Jain, "A comparison of hashing schemes for address lookup in computer networks", IEEE Trans. on Communications, vol. 40, no. 3, **(1992)**, pp. 1570−1573.
[4] Z. Cao, Z. Wang and E. Zegura, "Performance of hashing-based schemes for Internet load balancing", In: Nokia FB, ed. Proc. of the IEEEINFOCOM 2000. Piscataway: IEEE Computer and Communications Societies, **(2000)**, pp. 332−341.
[5] G. Cheng, J. Gong and W. Ding, "Distributed sampling measurement model in a high speed network based on statistical analysis", ChineseJournal of Computers, vol. 26, no. 10, **(2003)**, pp. 1266−1273 (in Chinese with English abstract).
[6] N. G. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation", IEEE/ACM Trans. on Networking, vol. 9, no. 3, **(2001)**, pp. 280−292.
[7] M. Molina, S. Tartarelli and F. Raspallm, *et a1*., "Implementation of an IPFIX compliant flow traffic meter： challenges and performance assessment", IEEE Workshop on IP Operations and Management, **(2003)**, pp. 61-67.
[8] S. Demetriades, M. Hanna, S. Cho and R. Melhem, "An Efficient Hardware-based Multi-hash Scheme for High Speed IP Lookup", High Performance Interconnects, **(2008)**.
[9] Y. Chen, X. Lu and Z. Sun, "Research of the Hashing Algorithms Based on IP Flow Management", Computer Engineering and Science, vol. 30, no. 4, (**2008).**
[10] S. Qiang and G. Cheng, "Comparison and Analysis of Hash Algorithm Based on Flows", Journal of Nanjing normal university (engineering and technology), (**2008)** December.
[11] Z. Cao, Z. Wang and E. Zegura, "Performance of Hashing-Based Schemes forInternet Load Balancing", Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings, (**2000)**, pp. 332-341.
[12] Z. Guo and Y. Liu, "Research of High Speed Network Packet Capturing Technology", Science paper Online, **(2010)** December**.**
[13] D. Kotturi and S. -M. Yoo, "High-Speed Parallel Architecture of the Whirlpool Hash Function", IJAST vol. 7, (**2009)** June, pp. 21-26.
[14] D. Dey, P. R. Mishra and I. Sengupta, "HF-hash: Hash Functions Using Restricted HFE Challenge-1", IJAST vol. 37, (**2011)** December, pp. 129-140.
[15] Y. Li, "An Effective TCM-KNN Scheme for High-Speed Network Anomaly Detection", IJAST, vol. 24, (**2010)** November, pp. 11-16.

Authors

Xin Zou

He received the Master degree from the Department of Science and Technology from Harbin Institute of Technology, China in 2003. Since 2008, he has worked in The National Computer network Emergency Response Technical Team Coordination Center of China. He current research interests mainly include routing protocol and algorithm design, performance evaluation and optimization for networks.



Zhou Li

She received the Master degree in Electronic and Information Engineering from Harbin Institute of Technology in 2005, and received Ph.D. from Beihang university in 2011. Her research interests include information security and avionics network, email:zhouli@cert.org.cn.

Gong Liangyi, born in 1987, now is the PhD student of the Computer Science and Technology Department of the Harbin Engineering University. His research interests include information security and wireless sensor network security. email: gongliangyi@hrbeu.edu.cn.