A Load Balancing Algorithm for High Speed Intrusion Detection

LU Sheng, GONG Jian, RUI Suying

Department of Computer Science and Engineering, Southeast University, Nanjing, China

Eastern China (North) Network Center of CERNET (210096)

86-25-3614718

{shlu, jgong, syrui}@njnet.edu.cn

ABSTRACT

Load balancing is applied to the development of network-based Intrusion Detection System (NIDS) to fit the performance problem caused by traffic in high bandwidth network. Inspired from the concept of bit entropy and bit flow entropy, a novel load-balancing algorithm named Dimension-based Classification Algorithm (DCA) is introduced in this paper. Based on the contents of fields in IP packet header and some simple operations, this algorithm can keep the relativities among packets in a high bandwidth network environment while distributing workload to different processing node. It has a fairly good load-balancing feature in both macroscopical and microscopical senses for high speed intrusion detection. The selection of operation and operand of DCA is discussed in detailed, and their efficiency is evaluated.

Keywords Load Balance, Intrusion Detection, High Bandwidth Network, Packet Classification, Bit Entropy.

1 Introduction

Continued progress of communication technology enlarges bandwidth of network. The traffic in high bandwidth network increases from Mbps to Gbps, which causes a number of performance problems in NIDS, and makes many traditional methods of IDS unfeasible any more.

Nowadays, attackers can find more and more valuable targets in the Net. Network attack incidents keep happening almost all the time. So NIDSes have to analyze more security related audit data in a shorter instant ever than before. Meanwhile, deliberated attacks become more sophisticated and sneakier, which makes the design of IDS more challenging. Detecting algorithms have to be more complex and accurate, which would stay abreast of the emerging of network attacks. Therefore, the conflict between performance of NIDS and the arrived mass of packets must be dealt with, which is obvious and exigent in high bandwidth network.

When processors' performance does not meet the requirement, using clustered architecture for load balance is a very common solution. But IDS has some specific requirements of load distribution that should be met. That is, the context-sensitive relationship among packets must be kept and the communication among processors should be minimized. For example, some sophisticated attackers will divide their remote exploit packets into fragments; and NIDSes should have the ability to reassemble those packets in order to detect such an action. If one wants to correlate attacks or intrusions, all the related packets have to be sent to the same processing node. Those requirements definitely restrict the selection of load balancing algorithms for NIDSes. Round-robin and some other traditional methods may not be fit for such application environments. New algorithm is required to meet these special requirements.

Inspired from the concept of bit entropy【1】, this paper proposes a new load balancing algorithm, named **Dimension-based Classification Algorithm (DCA)**, to be used in a high bandwidth network for NIDS and other applications in which the context-sensitive feature of flows need to be kept. With this algorithm, a good load balance can be

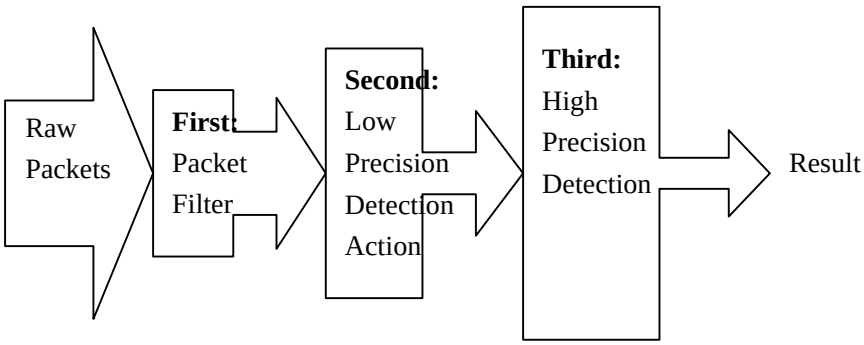

Figure 1: A cascade model of High Speed Intrusion Detection

achieved and the integrity of context-sensitive packets can be maintained as well. It should be noted that the algorithm discussed in this paper is misuse detection oriented.

In section 2, a cascade model is introduced which is typical cluster structure for high bandwidth NIDS implementation, and the basic concepts of DCA are defined. The operations and operands to be used in DCA are studied in section 3. The features of macroscopic and microscopic load balance of DCA are discussed in section 4 and section 5 respectively. Section 6 compares DCA with some traditional load balancing algorithms. Some conclusions are given in section 7.

2 Load Balancing for High Speed Intrusion Detection

2.1 A cascade model

There are approaches to resolve the performance problem in network-based intrusion detection, one of which is to sample the packets at various interval【2】. However, this method has high probability of missing individual attack, which will bring about high false positive rate. Therefore, most of the NIDSes analyze every packet that they captured in the network, but that will probably suffer severe performance problem when used in high bandwidth network, e.g. OC48. To handle this issue, a cascade model, shown in fig. 1, can be used. By this way, the front processor filters the packets in simple way, and drops all the packets that are unrelated with any intrusion. The followed processors analyze the packets left with higher precision and do more complex detection. Here, higher precision detection means more cost in storage and computation.

Although this cascade model does some work, it cannot settle all performance problems. Load-balancing is still needed in such an architecture, because certain processing phase may still be overloaded on some occasions, e.g. a traffic burst occurs.

2.2 Load balancing

Static and dynamic load balancing algorithms are widely used in many areas, from parallel computing to cluster computing environments. And they can be classified as geometric based algorithms, graph theory based algorithms and node migration algorithms etc. 【3】. All of these algorithms focus on task schedule and task distribution. Load balancing is also used in server cluster environment, such as distributed FTP servers, Web server clusters and so on. These clustering servers often use some kinds of algorithms that need less communication among nodes, e.g. random, round-robin, weighted round-robin, high availability etc. 【4】【5】. There are programmatic algorithms used to select a working processor according to IP address or some other information. Beyond those algorithms, there are also some solutions used for web load balancing. For example, mirroring, multi-homed/multi-provider network, content distributed server, server cluster etc.

All the load-balancing algorithms mentioned above dispatch the workload on a context-free basis with the emphasis on the load fairness on each processor. However, IDS requires that the relationship among packets cannot be lost when sharing the processing load with each processor. Therefore, it is more a classification than just randomly balancing the workload, so as to keep the integrity of context-sensitive packets.

2.3 Classification based load balance

G.Cheng, et.al. proposed a packet classification model which can be used for sampling network traffic in high bandwidth network. With this model, certain bits in packet are used to classify packets into different group for processing, so that the effect of load balancing is achieved. The chosen bits, e.g. Identification field in IP packet header, have a good randomness, and are suitable for sampling. The speciality of this model is that it can maintain the consistency of the packets sampled at different sampling point. That is, if one packet is sampled at a point, it will be sampled at all the points, with which packet can be distinguished and classified in high speed【1】. This feature can be applied to IDS load balance. Instead of selecting a set of samples by specific bits value, one can just separate the set of packets into subsets by the value of a bit or a set of bits, so that these packets are grouped for load balancing while the (context-sensitive) relativity among them is reserved.

Generally, if one wants to sample the same traffic with different sampling rate *Aratio* and *Bratio* (*Aratio* >= *Bratio*), the formula 1 must be satisfied.

$$\Omega(Aratio) \supseteq \Omega(Bratio) \qquad (1)$$

In this formula, Ω is the set of samples under given sampling ratio【6】. But in load balancing algorithm, if there are *n* processors with IDs from *1* to *n*, and $\Omega(i)$ is the packets set that the *ith* processor deals with. Then the formula 2 should be satisfied (*I* is corpora, Φ is empty set).

$$\bigcup_{i=1}^{n} \Omega(i) = I \qquad (2)$$
$$\forall i, j, 1 \le i \le n, 1 \le j \le n, i \ne j, \Omega(i) \bigcap \Omega(j) = \Phi$$

Using the concepts of Field, Rule List, etc. defined in 【7】, the basic idea of load balance using packet classification can be formally described as below[1]:

Packet Classification (PC): PC(*O*, Л, *M*)->i, i∈{1, 2, …, m}; m is the number of processors}; where three operations are involved.

- ***Field Generate Operation*** *(O)*: O(F_1, F_2, …, F_n)->F
  F_1, F_2, …, F_n are *n* fields in packet *P*. *O* is an operation on F_1~F_n. F is a set of bits, which is the result of operation *O*. Let F be a specific field of IP header in this specific context.
- ***Classification Operation*** (Л): Л($\Re$, F)->R
  $\Re$={R_1, R_2, …, R_m}, F is the result of operation *O*. R is a subset of $\Re$, and is the result of operation Л.
- ***Mapping Operation*** *(M)*: M(R)->i
  |R|=1, i∈{1, 2, …, m} is required in this specific context.

Obviously, each packet should send exactly to one processor (|R|=1), and each processor should be sent some packets aggregated by the sessions which belonged to for processing (|$\Re$|=m).

It is assumed that the processors are homogeneous because NIDS is supposed to be implemented with a cluster in this specific context. For homogeneous processors, balance of task schedule is most significant while unbalance among processors will decrease the total ability of the processor cluster remarkably. Therefore the task each processor takes must be almost in same quantity in order to achieve a good load balance in both macroscopical and microscopical scopes.

[1]We do not use the whole definition set of【7】, because not all the concepts are required in the paper.

2.4 Dimension-based Classification Algorithm

With the concepts described above, the dimension-based classification algorithm (DCA) can be defined generally as following: For each incoming packet P

1) Get the values of F_1, F_2, …, F_n in P;
2) Perform the operation O on F_1, F_2, …, F_n, and gain the result field F;
3) According to the rule set $\Re$ and field F, do the classification operation Π, and gain a result rule set R;
4) Based on a predefined mapping operation *M*, map the result rule set R to a classification number *i* (i=M(R));
5) Using the classification number *i*, assign P to *ith* processor.

Notice: any well-known packet classification algorithms, e.g. given in 【9】, could be directly used in step 3.

From above one can see that no communication among processors is required when dealing with the incoming packet. The relationship among packets will be assured by the selection of operation O and field F_i ($1 \leqslant i \leqslant n$) since the context-sensitive relationship is maintained by invariant of field. For example, for the segmented IP packet, most of the packet header content will be the same; and for a TCP session, all packets will at least have same source IP address and destination IP address. Because |R|=1, the integer *i* is just the identification of the processor which will handle the packet P, so that mapping operation *M* will be very simple. Therefore, the selection of Fields F_i, Field Generate Operation O, and Rule Set $\Re$ become the key points of the algorithm.

3 Selection of Operations and Operands in DCA

3.1 Bit Entropy and Bit Flow Entropy

Entropy, an important concept of information theory, is being used to measure randomness of various random experiments, and this concept can be extended to estimate bit randomness as well【1】. The following concepts are from【1】.

Definition 1: Bit Entropy, the entropy value of a bit, is defined as:

$$H(b) = -(p \log_2 p + (1-p)\log_2(1-p))$$

Where p is the probability of b=0, and (1-p) is the probability of b=1.

H(b) is the average indeterminacy of bit occurrences and can be used as a metrics of the randomness of bit *b*. The probabilities of bit *b=0 or b=1* are almost the same, so that for higher stochastic bit *b*, its bit entropy is larger and vice versa. Meanwhile, H(b) represents the average amount of information that each bit provided within a bit stream, so it is also a metrics of information measurement.

In reality, a flow of bits is more useful than a single bit, so that the randomness of a bit flow must also be defined.

Definition 2: Bit Flow Entropy is defined as:

$$H(s) = -\sum_{i=0}^{2^s-1} p_i \log_2 p_i \ ,$$

where *s* is the length of a bit flow which has $n=2^s$ events all together[2], and $p_0, p_1, \ldots, p_{n-1}$ are probabilities of each event.

According to the Maximal Bit Flow Entropy Theorem, the maximum of bit flow entropy is $H_{max}(s)=s$. Then the Information Efficiency E of a bit flow is defined by 【1】 as:

[2] 【1】defines the length of s as $n+1=2^s$, the bit flow from p_0 to p_n. We defines the length of s is n, the bit flow from p_0 to p_{n-1}.

Table 1 Truth table of two bits operations

0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
A	B	0	&		A		B	$\oplus$	|		$\overline{\oplus}$	$\overline{B}$		$\overline{A}$			1

Definition 3: Information Efficiency E of A Bit Flow, a metric of bit flow randomness, is the ratio between H(s) and $H_{max}(s)$: $E = H(s)/H_{\max}(s) = H(s)/s$.

The instance of DCA is definitely determined by the selection of operators and operands, which could be seen as the arguments of algorithm. To choose best-fit fields in a packet when instance an algorithm, bit entropy and bit flow entropy analysis is a very useful method. Larger bit flow entropy of a field is a strong assurance of good macroscopic load balance. After fields and operations are decided, another performance metric of microscopic load balance will be considered.

3.2 Selection of Fields F_i and Field Generate Operation O

The DCA does not use the fields in packet directly, but a result of an operation on them instead. Because the result of a bit flow operation is also a bit flow (field), its Information Efficiency E could also be used to test such a special field.

The Maximal Bit Flow Entropy Theorem shows that the Information Efficiency E of a bit flow is satisfied if and only if all the composing bits of this bit flow have large Bit Entropy【1】. That is, good stochastic field is composed of good stochastic bits.

To find good operations and good fields for DCA, one can look at two bits operations first. There are only $2^2 \times 2^2 = 16$ types of operators between two bit operands shown in table 1.

If the result of the operation is all 0 or 1, the entropy of result bit is 0. It is obvious that such two operations aren't fit for DCA. There are 8 types of operations that the ratio of 0 and 1 in result is 3:1 or 1:3 ($C_4^1 + C_4^3$). For those operations, if the entropies of A and B are both large and the correlation between A and B is nearly zero, the difference of the ratio of 0 and 1 in the result could be significant, so that the entropy of result bit should not be large enough. Of course, one could try to gain a large result entropy by using those operations when the entropies of A and B are both less. For example, if both A and B have high probability of being 1, the bit entropy of the result of operation (A & B) could be much larger than that of A or B. However, it is very difficult to find the bit A and bit B that has such a strange characteristics to be used to those operations.

So, using the operations which have two 0s and two 1s in result is a more reasonable choice. There are only six (C_4^2) types of operations have such feature. For operations A and $\overline{A}$, the bit entropies are both as same as A 's bit entropy, because $H(\overline{A}) = -((1-p)\log_2(1-p) + p\log_2 p) = H(A)$. For the same reason, the bit entropies of operations B and $\overline{B}$ are both as same as the bit entropy of operation B. Thus, there are only two types of operations with same entropy will need to be analyzed. They are exclusive OR (XOR) and NOT exclusive OR.

Let the probability that result equals to 1 is p, p_1 is the probability of A =1 and p_2 is the probability of B =1. Suppose that there is no correlation between A and B, then $p = p_1 p_2 + (1-p_1)(1-p_2)$, and the entropy of p is:

$$H = -(1-p_1)(1-p_2)\log_2(1-p_1) - p_1 p_2 \log_2 p_1 - (1-p_1)(1-p_2)\log_2(1-p_2) - p_1 p_2 \log_2 p_2$$

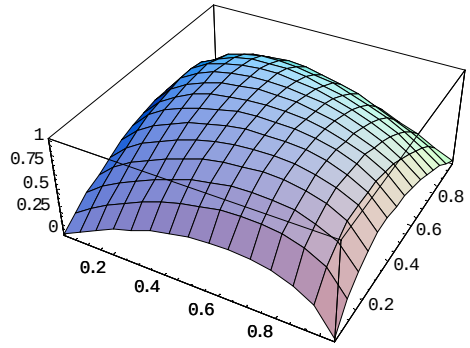

Figure 2 Bit Entropy of Two Bits Exclusive OR

For $p_1 \in (0,1)$ and $p_2 \in (0,1)$, the entropy of p is shown as figure 2.

The maximum point is mounted at $p_1 = p_2 = 0.5$, where the entropy H equals to 1. Because $\frac{\partial^2 H}{\partial p_1} = \frac{p_1 + p_2 - 2p_1 p_2}{-p_1 \ln 2 + p_1^2 \ln 2}$ will less than 0 in area $p_1 \in (0,1)$ and $p_2 \in (0,1)$, so that H shows as an arch structure.

With the discussion above, one can draw the conclusion that if fields with large Information Efficiency are chosen and XOR or NOT XOR operations are used over them, the Information Efficiency of the result will be good. Fortunately, those fields we need to keep the relativities among packets all have good bit flow entropy and Information Efficiency, such as source IP address, destination IP address, source port, destination port, IP identification field and so on【1】.

3.3 Choose Rule Set ℜ, Classification Operation Л and Mapping Operation *M*

There are less constraints about the selection of rule set ℜ, classification operation Л and mapping operation *M*. If these operations do not influence the efficiency of load balancing, the simpler, the better. A simple method described below is effective. Choose $\log_2 m$ (m must be an exponential of 2) bits in field F based on the number m of processors, and select the processors by those bits. For example, it would select two bits to denote 0~3 when m equals to 4, or select three bits to denote 0~7 when m equals to 8. If there are m processors, ℜ={R1, R2, …, Rm}, each Ri is a sequence of bits with length $\log_2 m$, as ℜ={0..0, 0..1, …, 1..1} (each rule of ℜ has $\log_2 m$ bits). The mapping operation is just the EQUAL, using the value of Ri directly.

Classification operation Л could be any one as long as it meets the performance and precision requirement of the problem, so it will not be discussed in this paper any more.

In the following discussion, only fields Fi and field generate operation *O* will be concerned.

4 Macroscopic Load Balance of DCA

To compare the load balancing performance of instances with different arguments, some metrics are needed. In this paper, two types of metrics are put forward, macroscopic metric and microscopic metric. Macroscopic metric is just the randomness of packets. If the packets dealt to the processors with similar probability, it is axiomatic that each processor will handle the packets with almost same amount in a long period. In this section, we will show that DCA can achieve good load balance in macroscopy (in a long

period) by choosing the arguments properly. Because the calculation of the macroscopic load balance is very simple, it would also be a good way to select the fields and field generate operation for an instance of DCA. 67,870,553 packets were analyzed that gained from CERNET backbone in one week with the static interval of dumping all packets in 2 seconds for every 75 seconds.

Only those fields that keep the relative information of the packets will be analyzed, which yields to different algorithms.

Algorithm 1: Let F_1 be the identification field of IP head, field generate operation O be EQUAL. O: (=, F_1)[3]

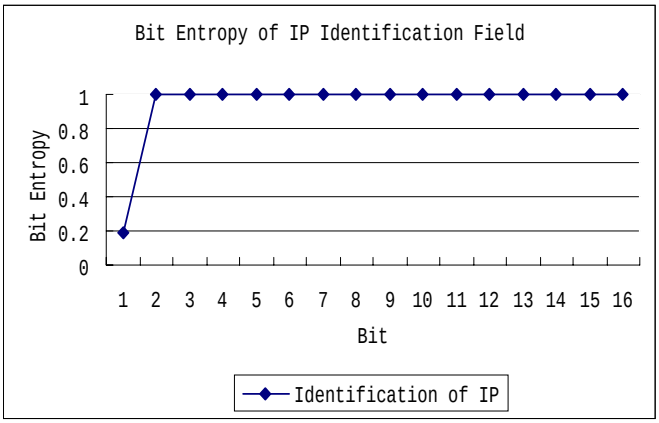

Figure 3 Bit Entropy of IP Identification Field

【1】had gained a conclusion that identification field of IP header has very good Information Efficiency. Figure 3 proved such declaration. This field is analyzed just because it involved in some attacks toward TCP/IP protocol implementations. Of course, the highest bit is not a fair bit that would be chosen.

Algorithm 2: Let F_1 be the source IP address, F_2 be destination IP address, and field generate operation O be exclusive OR (XOR). O=(^, F_1, F_2)

Both Source IP address and destination IP address are good stochastic variables. Figure 4 shows bit entropies of source IP address, destination IP address and the result of XOR operation between them.

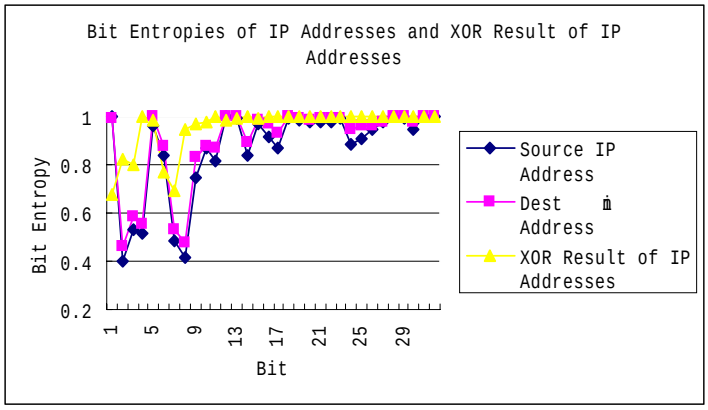

Figure 4 Bit Entropies of IP Addresses and XOR Result of IP Addresses

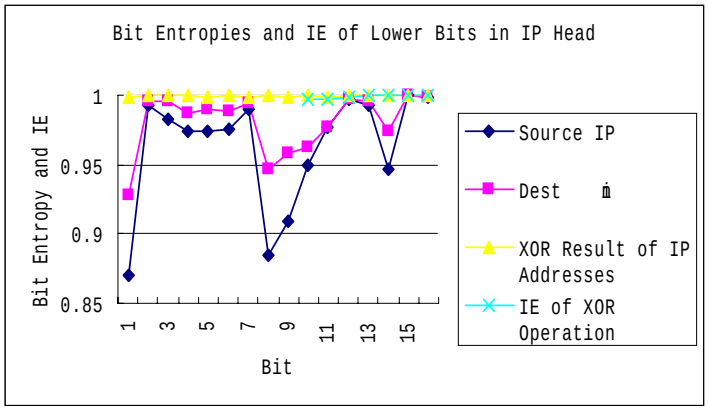

Figure 5 Bit Entropies and IE of Lower Bits in IP Head

The entropies of lower 16 bits and the Information Efficiencies (IE) of lower 7 bits are shown in figure 5. (The Information Efficiency point at the 16th bit is the Information Efficiency of the 16th bit, The Information Efficiency point at the 15th bit is the Information Efficiency of the bit flow consisted of the 15th bit and the 16th bit, and so on)

Figure 5 illustrates that the entropy of the result is better than any other operands. It means that the XOR operation will ameliorate the randomness of bit flows.

Algorithm 3: Let F_1 be source port field, F_2 be destination port field, and field generate operation O be XOR. O=(^, F_1, F_2)

Algorithm 4: Let F_1 be source IP address, F_2 be

[3] O is represented by s-expression leaded by operator, same as in other algorithms' description

destination IP address, F_3 be source port, F_4 be destination port, and field generate operation O be XOR. O=(^, F_1, F_2, F_3, F_4)

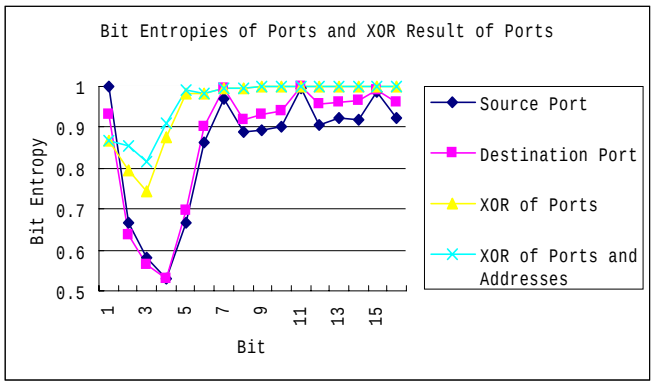

Figure 6 Bit Entropies of Ports and XOR Result of Ports

Port fields are used in algorithm 3 and algorithm 4. Figure 6 illustrates the entropies of source port, destination port, result of XOR operation between ports and result of XOR operation among ports and addresses. It proves that use XOR operation on good stochastic field, the entropy of the result will become larger.

The Information Efficiency of last seven bits is shown in figure 7. The point at the 16th bit is the Information Efficiency of the 16th bit, the point at the 15th bit is the Information Efficiency of the bit flow consisted in the 15th bit and the 16th bit, and so on.

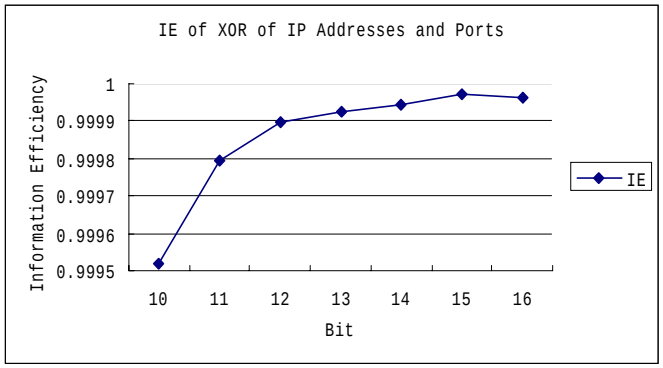

Figure 7 Information Efficiency of Algorithm 4

Figure 7 tells if more bits are used, the Information Efficiency will decrease. It means that increasing the number of processors will decrease the macroscopic load balance.

For all of the algorithms mentioned above, algorithm 2 will guarantee that all packets with same source IP address and destination IP address will be assigned to same processor. Algorithm 4 will promise that all packets in one TCP session will be assigned to same processor, and both of them would achieve good load balance. They both fit for high speed intrusion detection and other environment where such features are needed. But it is very hard to image the usage of algorithm 3.

5 Microscopic Load Balance of DCA

Good bit entropy and Information Efficiency promise a good macroscopic load balance, but they cannot assure the load balance in microscopicy (short time period). And some other metrics in parallel computing cannot be used also, for example the machine balance metric and other benchmarks【10】. Two measures of load balance in dealing with packets are defined to settle this problem, with the basic idea borrowed from【5】. The basic definitions of this two measures are same as in【5】, but the final metrics has been changed accordingly to be applicable for packet and flow.

Definition 4:

- $load_{i,j}$ - Load of processor i (of n processors) at the jth sampling point (of m such points)
- $peak\_load_j$ - highest load on any processors at the jth sampling point
- pead_to_mean ratio:- $\frac{peak\_load_j}{(\sum_{i=1}^{n} load_{i,j})/n}$

LBM(Load Balance Metric) -

$$\frac{\sum_{j=1}^{m}\left(\frac{peak\_load_j}{\left(\sum_{i=1}^{n} load_{i,j}\right)/n}\right)\times\frac{\sum_{i=1}^{n} load_{i,j}}{n}}{\sum_{j=1}^{m}\sum_{i=1}^{n} load_{i,j}/n} \quad (3)$$

$$=\frac{\sum_{j=1}^{m} peak\_load_j}{\left(\sum_{j=1}^{m}\sum_{i=1}^{n} load_{i,j}\right)/n}$$

The major difference between 【5】 and this paper is the definitions of $load_{i,j}$. $pps_{i,j}$(packet per second) and $bps_{i,j}$(bits per sencond) are defined here for different usages. Two types of LBMs are discussed, PLM (pps Load Balance Metric) and BLM (bps Load Balance Metric).

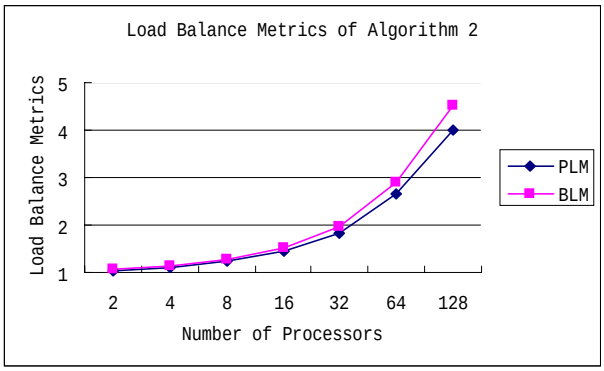

Figure 8 LBM of Algorithm 2

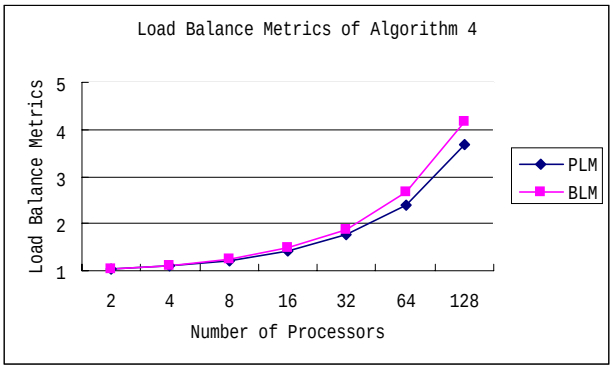

Figure 9 LBM of Algorithm 4

The PLM and BLM of algorithm 2 are shown in figure 8, and the PLM and BLM of algorithm 4 are shown in figure 9.

These two figures show that the increasing of the number of processors decrease the microscopic load balance. This conclusion is similar to the one obtained from macroscopic analysis.

Comparing figure 8 with figure 9, it is clear that the microscopic load balance of algorithm 4 is better than that of algorithm 2. It is another proof that using more stochastic variable will improve the load balance.

6 Conclusion

DCA deals the packets to different processors. It is very similar with the algorithms in server cluster, but quite different from the algorithms in parallel computing. Because the arrival speed of the packets is much faster than any other requests in server cluster, it lacks the time of communication among processors, so that the communication must be restricted to almost none.

Based on fields in IP packet header and the concepts of bit entropy and bit flow entropy, this novel load-balancing algorithm can keep the relativities among packets. And with a cascade model, DCA can be applied an NIDS in a very high bandwidth network environment.

The load-balancing features of DCA in both macroscopical and microscopical senses are analyzed in the paper. It is shown that the higher the randomness of the bits chosen, the better the balance achieved. And the XOR operation can also give help to improve the macroscopical balance. Both the macroscopical and microscopical analysis shows that the size of the cluster used to balance the work is limited, and the best number of node is between 4 to 8 in context-free environment. However, this conclusion also suggests that if some context is introduced, e.g. the current workload of nodes, or the prediction of workload in certain period of time, the size of the cluster could be expanded, which means that more workload could be handled.

The algorithm described in the paper not only can be used in high speed Intrusion Detection System, but also can used in any other situation need to keep the relativities among packets, for example, the

generation of flow, flow analysis etc. in network measurement.

References

【1】 CHENG Guang, GONG Jian, DING Wei, “Network Traffic Sampling Measurement Model on Packet Identification”, Acta Electronica Sinica (Tien Tzu Hsueh Pao), Vol.30, No.12A, Dec. 2002: 89-93(in Chinese)

【2】 James Cannady, “Intrusion Detection – Capabilities and Considerations”, Global Inforsecurity 2002

【3】 Y. F. Hu, R. J. Blake, “Load Balancing for Unstructured Mesh Applications”, Parallel and Distributed Computing Practices, Vol. 2, No. 3, September 1999

【4】 “Growing Your E-Business with IBM Server Load Balancing and Caching embedded solutions”, IBM White Paper. http://www.networking.ibm.com/white/serverload.html

【5】 Richard B. Bunt, Derek L. Eager, Gregory M. Oster, and Carey L. Williamson, “Achieving Load Balance and Effective Caching in Clustered Web Servers”, Proceedings of the forth International Web Caching Workshop, San Diego, California, April 1999, 159-169

【6】 CHENG Guang, GONG Jian, DING Wei, “A Time-series Decomposed Model of Network Traffic Macro-Behavior Analysis”, Acta Electronica Sinica (Tien Tzu Hsueh Pao), Vol.30 No.11, Nov. 2002(in Chinese)

【7】 Sundar Lyer, Ramana Rao Kompella, Ajit Shelat, “ClassiPI: An Architecture for Fast and Flexible Packet Classification”, IEEE Network, March/April 2001, pp.33-41

【8】 “IP Denial-of-Service Attacks”, CERT Advisory, CA-1997-28

【9】Pankaj Gupta, NIck McKeown, “Algorithms for Packet Classification”, IEEE Network, March/April 2001, pp.24.

【10】 Kai Hwang, Zhiwei Xu, “Scalable Parallel Computing: Technology, Architecture, Programming”, China Machine Press, ISBN 7-111-07176-X, pp.91, May. 1999