International Journal of Security and Its Applications
Vol.9, No.11 (2015), pp.35-46
http://dx.doi.org/10.14257/ijsia.2015.9.11.04

Optimization of Dynamic Programming to the Multimedia Packets Processing Method for Network Intrusion Detection System

Xu ZHAO[1], Jin Jiang[2] and Max Stinnett[3]

[1]*Department of Computer Science,*
Xi'an Polytechnic University, Xi'an, China
[2] *Xi'an Polytechnic University, Xi'an, China*
[3]*The Teacher College,*
Emporia State University,
Emporia, Kansas, USA
37274679@qq.com

Abstract

Network Intrusion Detection System (NIDS) is an important network security system. There always appears high packet loss rate in NIDS, especially when the network traffic is high. We raised Multimedia Packets Processing Method to reduce the packet loss rate and received good results. On this basis, Dynamic Programming is applied to optimize the decision-making steps of this method. This improvement can help the system to find an optimum solution to select the highest risk of multimedia data packet sequence in each time slice, moreover, the system load capacity can also be considered. In this way, the limited processing power of NIDS can be focused on the more dangerous multimedia data packets. A series of experimental results indicate this optimization can help the system to improve the detection rate of the high risk of multimedia information.

Keywords*: Network Intrusion Detection System (NIDS), multimedia packets, dynamic programming, Multimedia Packets Processing Method, optimization*

1. Introduction

1.1 Background and Problem Statements

NIDS is used to monitor network traffic and detect attack attempts. As network speeds increase, the requirements of the NIDS's processing efficiency also increase. How to improve the NIDS in the processing capacity per unit time becomes a research hotspot in the field.

In order to solve this problem, many researchers have studied it in different ways. Here are several common research directions.

1. Clustering Algorithm

JIANG Shen [1] proposed an improved ant colony clustering method for NIDS. This method has not only improved the detection rate, but also reduced the fault detection rate. However, this method has low detection efficiency for unknown attacks. LIU Feng-zhu [2] proposed an improved k-means clustering algorithm for NIDS, whose algorithm can effectively improve detection accuracy. But her algorithm has certain instability when the value of K is between 3 and 6. ZHAI Guangqun [3] proposed a new intrusion detection algorithm based on the combination of K-prototypes and fuzzy evaluation. This algorithm has many advantages, but it has some disadvantages. For instance, it might mistake dubious data for normal one.

ISSN: 1738-9976 IJSIA

2. Artificial Neural Networks [4-5]

He Liang [5] presented a new detection algorithm—T-S FNN based network intrusion detection algorithm in order to overcome the current high rate of false positives, low detection rate. This algorithm uses T-S FNN to classify objects, divides eigen-space of objects and recognizes normal behaviors and intrusions. This new method is effective and feasible.

3. Particle Swarm Optimization [6-7]

LI Zhengjie [7] proposed an intrusion detection model based on immune agent and particle swarm optimization immune principle by means of combining mobile agent and quantum-behaved particle swarm optimization. This system can improve the low detecting speed and high false positive rate of traditional NIDS. However, this approach is easy to fall into local optimum.

4. Support Vector Machine [8-9]

LIU Ming-zhen [9] put forward a network intrusion detection model based on the Chaotic Particle Swarm Optimization (CPSO) algorithm and Least Squares Support Vector Machine (LSSVM). This model can select the optimal feature subset and LSSVM parameters. The detecting speed and network intrusion detection accuracy are improved by using this model. However, the detection rate of this model is not ideal for U2R and R2L.

5. Improved Pattern Matching Algorithm [10-14]

Some researchers have improved the efficiency of intrusion detection system by using improved pattern matching algorithm which is an indispensable part in NIDS. For example, Lu Linlin [11] presents an improved multi-pattern matching algorithm which is based on deterministic finite-state automaton. However, this algorithm is only suitable for finding the small character sets pattern string in large character set text string.

6. Other Research Directions [15-20]

In addition, some researchers improve the NIDS by adapting different methods. For instance, Sravani.K [15] proposed the use of a classification algorithm for network traffic data classification. M.Arun [16] discussed the signature detection technique (SDT) used in NIDS, whose approach involves a unique combination of algorithmic and architectural techniques that outperform some of the previous techniques in terms of performance, speed and power-efficiency. Zhicai Shi [20] proposed a simple and fast discretization algorithm based on information loss by fusing Rough Set theory with Entropy theory. This algorithm is applied to different samples with the same attributes from KDDcup99 and intrusion detection systems. However, the proposed discretization algorithm is sensitive to the initial samples only for part of all condition attributes.

Since the multimedia packets occupy a larger proportion in network flow, the method of particular processing on them can greatly improve the efficiency of NIDS. Among various studies on NIDS, we started from the study of multimedia data in network flow and proposed an identifying method and two separate processing methods [21] for multimedia packets to raise the efficiency of the NIDS. On this basis, we have proposed the Multimedia Packets Processing Method [22] and gained very satisfying results. Based on these studies, the idea of optimization in Dynamic Programming is applied to optimize the decision-making steps of the Multimedia Packets Processing Method to further enhance the efficiency of this method.

1.2 Contributions of this Paper

To summarize, we mainly make the following contributions in this paper:

(a). Among many studies on NIDS, we started from the study of multimedia packets in the network. On the basis of proposed Multimedia Packets Processing Method, we

propose an optimization scheme, which add the idea of dynamic programming to the main decision-making steps of Multimedia Packets Processing Method. By using this optimization scheme, NIDS can make limited processing capability focus on the more risky multimedia packets.

(b). We establish a mathematical model of the abovementioned optimization scheme and prove its feasibility

(c). We introduce the concrete implementation of the optimization scheme and design several experiments which compare the detection rate, packet loss rate and other indicators of NIDS before and after optimization and prove its effectiveness.

1.3 Paper Organization

The rest of the paper is organized as follows: First, related works are discussed in Section 2; Section 3 presents a brief introduction of Dynamic Programming; Section 4 describes the optimization scheme to the Multimedia Packets Processing Method with Dynamic Programming and the proof of feasibility to optimization scheme; the experiment and result analysis for contrast the differences before and after optimization is in Section 5; Section 6 summarizes the whole paper and presents some directions of the future work.

2. Related Works

2.1 Two Separate Processing Methods for Multimedia Packets

In the NIDS, the conventional detection method performs pattern matching on all packets with thousands of rules. It does not distinguish between the multimedia packets and non-multimedia packets. Actually, multimedia packets account for a larger proportion in network traffic and are relatively safe [21]. Thus, this makes multimedia packets consume a lot of the system's resources.

We has proposed and designed an identifying method and two separate processing methods [21] for multimedia packets to solve this problems, and on this basis, the Multimedia Packets Processing Method has been realized [22]. These two processing methods are as follows:

1. The releasing method [21]: When the multimedia packet is found in the net flow, this method identifies multimedia packets and allows them to bypass the conventional detection method in NIDS. Though this method is simple and efficient, its security is lower.

2. The corresponding media type detection method (CMTDM) [21]: This method is a safer and more efficient method than the Releasing method. To initiate this method, a multimedia rule base is created. The multimedia rule base stores the rules which are specifically collected for multimedia packets. The CMTDM can be used to choose the corresponding multimedia rules according to the specific multimedia type that packets carry in order to pre-detect aggressive characteristics. If no aggressive characters are identified in the multimedia packet, the packet is released. However, if aggressive characters are identified by The CMTDM, the packet will be put into the conventional detection method in NIDS. Because the number of multimedia rules contained in the CMTDM is far less than the rules for conventional detection method in the NIDS, this method can significantly improve the detective efficiency for most of the safe multimedia packets. The security of this method is also higher than that of the releasing method.

Compared with the conventional detection method *(i.e.* performing pattern match to all packets with thousands of rules) in the NIDS, both of the two processing methods can obviously reduce the packet loss rate, ss shown in Figure 1 and improve the threshold value of dropping packets , as shown in Figure. 2.

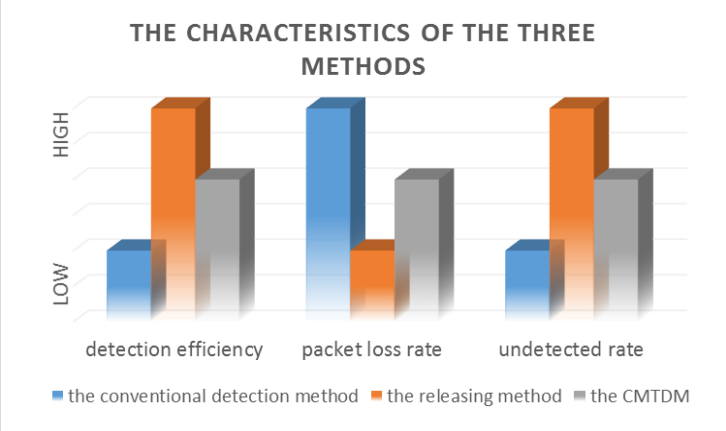

Figure 1. The Characteristics of the Three Methods in the Detection Efficiency, Packet Loss Rate and Undetected Rate

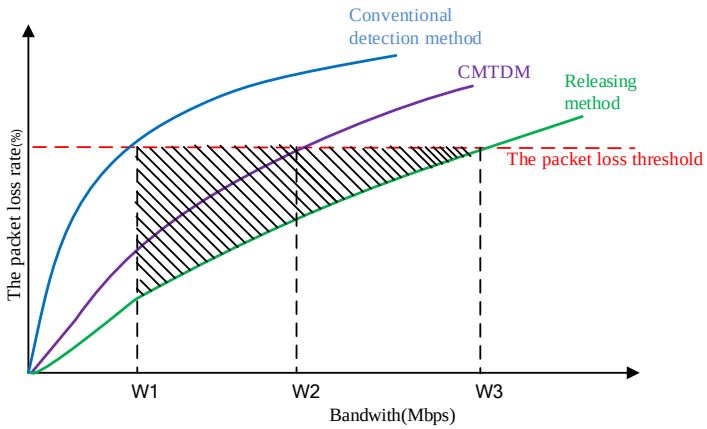

Figure 2. The Packet Loss Threshold by Using Three Different Methods

2.2 The Multimedia Packets Processing Method

The relationship between the packet loss rate by using three different methods and bandwidth can be shown in Figure 2. According to this relationship, the author has proposed the Multimedia Packets Processing Method, whose basic idea is as follows [22]:

1. When the system starts, it works with the conventional detection method and meanwhile statistics stable network traffic in a specific period of time. Then, according to the speed of current network traffic (*i.e.* less than W1, or betweenW1 and W2, or betweenW2 and W3, or more than W3), the system can select the most appropriate method to work from the conventional detection method, the releasing method and the CMTDM. If the speed of network traffic crosses from one interval to another interval, the system should reselect the method.
2. If the chosen method is the CMTDM, and the speed of current network traffic between W1 and W2 (within this interval the system has spare processing capacity), the system should process increasing multimedia packets by using

the conventional detection method under the premise of no packets loss. If packet loss arises, the system should process decreasing multimedia packets.

3. If the chosen method is the releasing method, and the speed of current network traffic between W2 and W3 (within this interval, the system has spare processing capacity), the system should process increasing multimedia packets by using the CMTDM under the premise of no packets loss. If packet loss arises, the system should process decreasing multimedia packets [22].

2.3 Why to Optimize?

Although multimedia packets are relatively safe, they are not absolutely safe. So, in steps [2] and [3] of the Multimedia Packets Processing Method, we hope that those more dangerous multimedia packets are processed preferentially by adapting more rigorous methods (the Conventional detection method and the CMTDM) within the range of the maximum load in the system. However, the key is how to choose these multimedia packets within a time slice. Dynamic Programming is considered for use.

3. Dynamic Programming

Dynamic programming is a method for solving complex problems by breaking them down into simpler sub problems. It is applicable to problems exhibiting the properties of overlapping sub problems and optimal substructure.

Dynamic programming usually refers to simplifying a decision by breaking it down into a sequence of decision steps over time. This is done by defining a sequence of value functions $V_1, V_2, \ldots, V_n$, with an argument y representing the state of the system at times i from 1 to n. The definition of $V_n(y)$ is the value obtained in state y at the last time n. The values V_i ($i = n-1, n-2, \ldots, 2, 1$) at earlier times can be found by working backwards, using a recursive relationship called the Bellman equation. For $i = 2, \ldots, n$, V_{i-1} at any state y is calculated from V_i by maximizing a simple function (usually the sum) of the gain from a decision at time $i-1$ and the function V_i at the new state of the system if this decision is made. Since V_i has already been calculated for the needed states, the above operation yields V_{i-1} for those states. Finally, V_1 at the initial state of the system is the value of the optimal solution. The optimal values of the decision variables can be recovered, one by one, by tracking back the calculations already performed [23].

The fundamental principle of dynamic programming

For the initial state $x_1 \in X_1$, policy $p_{1n}^* = u_1^*, \ldots, u_n^*$ is the necessary and sufficient conditions for the optimal policy. For arbitrary k, $1 < k \le n$, there is:

$$V_{1n}(x_1, p_{1n}^*) = \phi(opt_{p_{1k-1} \in p_{1k-1}(x_1)}[V_{1k-1}(x_1, p_{1k-1})], opt_{p_{kn} \in p_{kn}(x_k)}[V_{kn}(x_k, p_{kn})]) \quad (1)$$

If $p_{1n}^* \in p_{1n}(x_1)$ is the optimal strategy, then for arbitrary k, $1 < k < n$, its sub-policy p_{kn}^* is equally the optimal strategy.

The fundamental principle of the dynamic programming can be described briefly as the following: as for the current state formed by previous decisions whatever the past state and decision was, the rest of each decision must be composed of the optimal strategy.

The fundamental equation of dynamic programming is as follows:

$$\begin{cases} f_k(x_k) = opt\{\varphi(v_k(x_k, u_k), f_{k+1}(x_{k+1}))\}, x_{k+1} = T_k(x_k, u_k), k = 1, 2, \cdots, n \\ f_{n+1}(x_{n+1}) = \delta(x_{n+1}) \end{cases} \quad (2)$$

$f_n+1(x_n+1)=\delta(x_n+1)$ is the terminal condition of the decision process and δ is a known function. When x_n+1 is fixed, it is called a fixed terminal. When x_n+1 can change in terminal collection X_n+1, it is called a free terminal. The eventually required optimal indicator function satisfies the following formula:

$$opt\{V_{1n}\}=opt_{x_1\in X_1}\{f(x_1)\} \tag{3}$$

Formula (2) is a recursive formula. If the target state is determined, this formula can be used to find the optimal value. But in practical application this recursive formula is usually replaced by recursion formula to make it more efficient.

4. Optimization of Dynamic Programming to the Multimedia Packets Processing Method

4.1 Optimization Scheme

It's expected that the selected multimedia packets within each time slice can maximize the degree of risk of them, at the same time, it will not cause the system overload. How to achieve this goal is the focus of this chapter

Firstly, the following parameters need to be defined:

M: The maximum load of the NIDS, *M* can be obtained by counting statistics the stable simulated network traffic over a period of time under the condition of the same bandwidth;

X_i: A multimedia packet to be selected within each time slice

W_i: The load to the system which is caused by multimedia packet X_i, because there are significant positive correlations between the time complexity of the pattern matching algorithm and length of the string to be matched, W_i is determined by the ratio of the actual length of the packet load and the total length.

P_i: The degree of risk of multimedia packet X_i. The value of P_i depends on the degree of risk of multimedia data type which the packets carry. If the packet contain media type such as x-javascript, octet-stream, html, jpeg, gif, x-shockwave-flash, *etc*., then its P_i value will be higher.

Accordance with the idea of dynamic programming, this optimization problem can be described as follows:

In the interval of W1~W2(or W2~W3) in Fig.1, the identified n multimedia packets($X_1, X_2, \ldots, X_n$) within each time slice needs to be picked out to be dealt with by the conventional detection method(if in the interval of W2~W3, the CMTDM should be used). We need to find an optimal selection method to make the objective function:

$$max\sum_{i=1}^{n} P_iX_i,\ Xi\{0,1\},P_i>0,1\le i\le n \tag{4}$$

The constraint function:

$$\sum_{i=1}^{n} W_iX_i\le M, M>0, W_i>0, X_i\{0,1\}, 1\le i\le n \tag{5}$$

Wherein variable $X_i=0$ means the multimedia packet X_i is not selected, $X_i=1$ means it is selected.

4.2 The Proof of Feasibility of Optimization Scheme

Based on the essential principles of dynamic programming, whether dynamic programming can be adapted to optimize Multimedia Packets Processing Method depends on if this problems exhibit the properties of optimal substructure. We spell out the details as below.

Let us suppose $(X_1, X_2, \ldots, X_n)$, $X_i \in \{0,1\}$ is the optimal solution to the method how to select the riskiest multimedia Packets in each time slice, so $(X_2, X_3, \ldots, X_n)$must be the optimal solution in sub time slice. Now the maximum load of the NIDS is $M - W_1X_1$. There are $n-1$ multimedia Packets in this sub time slice, the load to the system caused by the ith $i(1 \le i \le n)$ multimedia packet is W_i. The degree of risk of this multimedia packet is $P_i(P_i > 0)$. Otherwise, supposing $Z_2, Z_3, \ldots, Z_n$ is the optimal solution in sub time slice, $(X_1, X_2, \ldots, X_n)$ is not the optimal solution in sub time slice. Therefore, there appears

$$\sum_{i=2}^{n} P_iZ_i > \sum_{i=2}^{n} P_iX_i, \text{ and } \sum_{i=2}^{n} W_iZ_i \le M - W_1X_1$$

Thus, there is

$$P_1X_1 + \sum_{i=2}^{n} P_iZ_i > \sum_{i=1}^{n} P_iX_i, \text{ and } W_1X_1 + \sum_{i=2}^{n} W_iZ_i \le M$$

Obviously,$(X_1, Z_2, Z_3, \ldots, Z_n)$is the optimal solution and it's more dangerous than$(X_1, X_2, \ldots, X_n)$. Thus, $(X_1, X_2, \ldots, X_n)$is not the optimal solution in sub time slice, which contradicts the assumption above. Therefore, $(X_2, X_3, \ldots, X_n)$must be the optimal solution in sub time slice. This shows that dynamic programming can be used to optimizing Multimedia Packets Processing Method, this problem has the properties of optimal substructure.

4.3 The Process of Specific Optimization

The state transition equation can be obtained as below based on dynamic programming and optimization program

$$f_i(M) = Max\begin{cases} f_{i-1}(M) \\ f_{i-1}(M - W_i) + P_i \end{cases} \quad (6)$$

$f_i(M)$ represents the greatest degree of risk of prior i multimedia packets selected in NIDS whose maximum load is M. According to this formula, we can obtain the optimal choice sequence of Multimedia Packets to be detected.

Here is the key code of optimization program by means of recursive fashion according to state transition equation.

```
for(i=1;i<=n;i++)
 for(j=1;j<=m;j++)  // m represents the maximum load of NIDS
 {
 if(w[i]<=j) /*if the load to the system caused by current multimedia packet is less than that of the system */
   {
    if(p[i]+c[i-1][j-w[i]]>c[i-1][j])  /*p[i] represents the degree of risk of current multimedia packet */
    /*If the degree of risk of current sequence is greater than that of last sequence, c[i][j] should be updated*/
```

```
      c[i][j]=p[i]+c[i-1][j-w[i]];
    else
      c[i][j]=c[i-1][j];
    }
   else
c[i][j]=c[i-1][j];
 }
```

Because there is a nested loop in the above program and the number of looping is n*m, the time complexity of this optimization method is $O(nm)$

5. Experiment

5.1 Experiment Design

The experiments reported here demonstrate a variety of changes in performance before and after optimization. Experimental data is a mixture of MIT Lincoln Laboratory KDD CUP 99 data sets and the background traffic. KDD CUP 99 data sets include four types of network attacks [24-25], DoS, R2L, U2R and PROBE. The types and quantities of attack category are shown as follows:

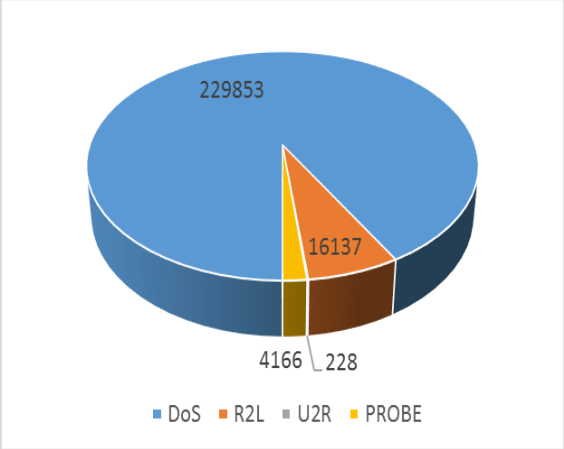

Figure 3. The Types and Quantities of Attack Category in Attack Traffic

Background traffic is the real flow captured before in the network, containing a large number of multimedia data packets. In the experiment, the mixed flow will be used to contrast the effect before and after optimization. Before testing, the value of P should first be set for different media types according to the degree of risky information carried by packets (as shown in Table 1). For instance, executable files can appear in multimedia file of octet-stream type, the value of P of octet-stream type can be set higher. The Table below shows the value of P for several common multimedia types.

Table 1 the Value of P for Several Common Multimedia Types

multimedia types	P
x-shockwave-flash	1.7
x-JavaScript	2.2
html	1.3
jpeg	1.5
gif	1.5
octet-stream	2.4
……	…..

5.2 Detection Rate

The first experiment is to compare differences in the detection rate of different multimedia types of packets before and after optimization, as shown in Figure 4.

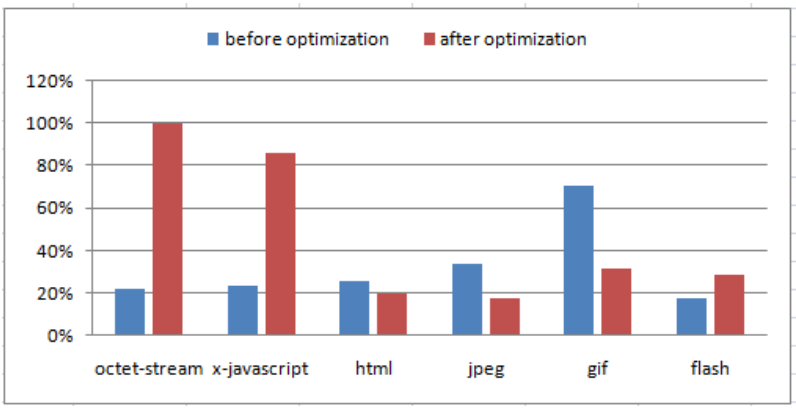

Figure 4. The Differences in the Detection Rates of Different Multimedia Types of Packets Before and After Optimization

As can be seen from Figure 4, the detection rate of multimedia packets with higher degree of risk P shown in Table 1, has been generally improved after optimization. For example, the octet-stream type has increased by 80% and the x-JavaScript type has increased by 59%. In addition to the effect of the optimization, the reason for its great improvement is that the detection rate is low before optimization because these multimedia packets have been selected randomly to be detected. On the other hand, it shows that the detection rate of multimedia packets with lower degree of risk has decreased. For example, the detection rate of "gif" packets decreases by 34%.

5.3 Degree of Risk

The second experiment is to compare differences between the degree of risk which is determined by multimedia packets selected in different time slices before and after optimization.

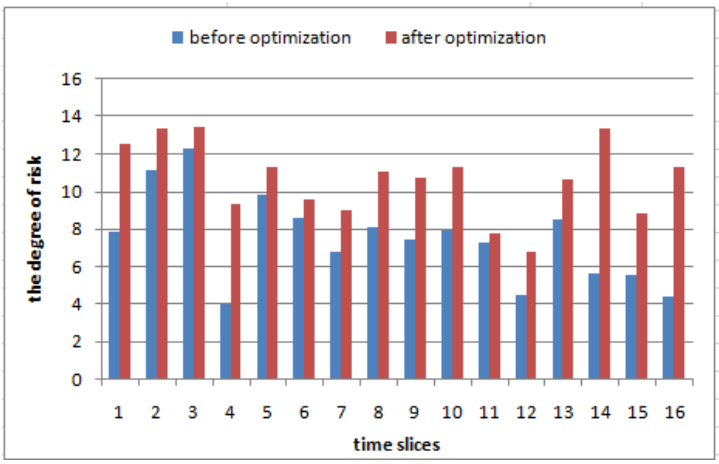

Figure 5. The Differences Between the Degree of Risk in Different Time Slices Before and After Optimization

As shown in the Figure 5, the degree of risk of multimedia packets selected in different time slices after optimization is significantly higher than that of before optimization. This is because multimedia packets are selected by chronological order regardless of the degree of risk of multimedia packets within a time slice before optimization. So the curve fluctuates significantly before optimization and shows its random, while the curve is relatively stable and higher after optimization. This can also be seen by the sample variance of results. According to the calculation formula of sample variance $S^2 = \frac{\sum_{i=1}^{n}(x_i - E(x))^2}{n-1}$, Sample variance is 7.1 before optimization, while it is 4.0 after optimization.

5.4 Packet Loss Rate

The third experiment is to compare the change of the packet loss rate before and after optimization.

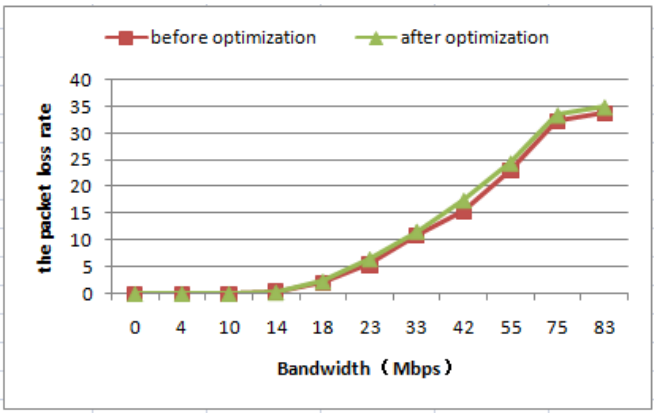

Figure 6. The Contrast of Packet Loss Rate Before and After Optimization

As shown in the Figure above, the change of the packet loss rate before and after optimization is not very obvious. The reason for its slightly higher packet loss rate

after optimization is its consumption of system resource in the optimal choice sequence of multimedia packets.

6. Conclusion and Future Work

With the high development and the widely application of network technology, network invasion is becoming an increasingly serious problem for network engineers and managers. Intrusion detection becomes a critical component of network security administration.

This paper addresses the performance challenges of NIDS in high-speed networks by proposing an optimization scheme by dynamic programming. By using this optimization scheme, within a network flow range, NIDS can make limited processing capability focus on more dangerous multimedia packets. Various experiments have shown that the optimization can effectively improve the detection rate of multimedia packets with dangerous information while the change of the packet loss rate before and after optimization is not very obvious.

As for future work, firstly, I will compare continually the advantage with disadvantage of the optimization proposed in the paper with those of other experiments to get more objective evaluations. Secondly, facing the high bandwidth network, I will conduct research on the optimization based on genetic algorithm and evaluate the performance of the optimization in the higher speed network environment.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61201118), Science and Technology Planning Project of Shannxi Province of China (No.2014K05-43), Xi'an Beilin District Science and Technology Bureau (GX1509).

References

[1] J. Shen, W. Dawei, 'An Improved Ant Colony Clustering Method for Intrusion Detection', Computer Technology and Development , vol. 23, no.12, **(2013)**, pp.139-142.
[2] L. Feng-zhu, 'A Clustering Method for Anomaly Intrusion Detection', Computer Security, vol.15, no.2, **(2013)**, pp.156-161.
[3] ZHAI Guangqun, 'Intrusion detection algorithm based on fuzzy evaluation and clustering analysis', Computer Engineering and Applications , vol. 48, no. 21, **(2012),** pp. 99-102.
[4] W. Wen-Tie, L. Min, L. Bo, 'Intrusion Detection Scheme based on IPSO-RBF', Journal of networks, vol. 8, no. 10, **(2013)**, pp. 2269-2276.
[5] H. Liang, 'An Improved Intrusion Detection based on Neural Network and Fuzzy Algorithm', Journal of networks ,vol. 9, no. 5, (**2014)**, pp. 1274-1280.
[6] W. Qingtao, C. Jibang, Z. Ruijuan, *et al*. 'Intrusion feature selection algorithm based on Particle Swarm Optimization', Computer Engineering and Applications, vol. 49, no. 7, **(2013)**, pp. 89-92.
[7] LI Zhengjie, LI Yongzhong, XU Lei, 'Research of intrusion detection method based on particle swarm optimization and immune Agent', Computer Engineering and Applications, vol. 48, no. 1, **(2012)**, pp. 102-104.
[8] ZHOU Guangping, Anup Shrestha, 'Efficient Intrusion Detection Scheme based on SVM', Journal of networks , vol. 8, no. 9, **(2013)**, pp. 2128-2134.
[9] LIU Ming-zhen, 'Network Intrusion Detection Based on CPSO-LSSVM', Computer Engineering, vol. 39, no. 11, **(2013)**, pp. 131-135.
[10] S. Tian, L. Dong-Ni, 'Memory Efficient Algorithm and Architecture for Multi-Pattern Matching', Journal of Software ,vol. 24, no. 7, **(2013),** pp. 1650-1665.
[11] L. Linlin, T. Ye, 'Research on Proving Multi-pattern Matching Algorithm Based on Deterministic Finite-state Automation', Computer Applications and Software, vol. 30, no. 7, **(2013)**, pp. 321-323.
[12] L. Wei-guo, H. Yong-gang, DHSWM, 'An improved multi-pattern matching algorithm based on WM algorithm', Journal of Central South University, vol. 4212, **(2011)**, pp. 3765-3771.
[13] Y. Hongwen, 'Research on improved BMH single-pattern matching algorithm based on Snort', Computer Engineering and Applications, vol. 48, no. 31, **(2012)**, pp. 78-81.
[14] D. Shibo, L. Xungen, Y. Zhenzhen, 'Improved string matching algorithm', Computer Engineering and Applications , vol. 49, no. 8, **(2013)**, pp. 133-137.

[15] K. Sravani, P. Srinivasu, 'Comparative Study of Machine Learning Algorithm for Intrusion Detection System', Advances in Intelligent Systems and Computing , vol. 247, **(2014)**,pp.189-196.
[16] M.Arun , A.Krishnan, 'Functional verification of signature detection architectures for high speed network applications', International Journal of Automation and Computing, vol. 9, no. 4**, (2012)**, p. 301.
[17] L. Yue, W. Rui, G. Jiping, *et al*, 'Intrusion detection model of abnormal event fusion based on gray correlation', Computer Engineering and Applications, vol. 49, no. 12, **(2013)**, pp. 86-89.
[18] H. Altwaijry, K. Shahbar, '(WHASG) Automatic SNORT Signatures Generation by using Honeypot', Journal of Computers,vol. 8, no. 12, **(2013)**, pp. 3280-3286.
[19] W. Ren, L. Hu, K. Zhao, 'Intrusion Classifier based on Multiple Attribute Selection Algorithms', Journal of Computers, vol. 8, no 10, **(2013)**, pp. 2536-2543.
[20] Z. Shi, Y. Xia, F. Wu, J. Dai, 'The Discretization Algorithm for Rough Data and Its Application to Intrusion Detection', Journal of networks , vol. 9, no. 6, **(2014)**, pp. 1380-1387.
[21] Z Xu, W. Changshan, 'The Improvements to Snort Intrusion Detection System', Journal of Xi'an Polytechnic University, vol. 21, no. 6, **(2007)** , pp. 859-863.
[22] Z. Xu, 'Research on Dynamic Self-Adapting Multimedia Data Processing Method Based on Snort', Computer Systems & Applications), vol. 20, no. 4, **(2011)**, pp. 211-213.
[23] C. Thomas H.,Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford, Introduction to Algorithms (2nd ed.), MIT Press & McGraw–Hill, **(2001)**.
[24] Z. Xin-you, 'Research of intrusion detection system dataset-KDD CUP99', Computer Engineering and Design vol. 31, no. 22**, (2010)**, pp. 4809-4812.
[25] M. Mahoney, P. K. Chan, 'An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection', RAID **(2003)**, pp. 220-237.

Authors

Xu Zhao, he is a Lecturer in the Department of Computer Science, Xi'an Polytechnic University, Shannxi, China. He received the M.S. degree from Xi'an Electronic Technology University, Xi'an City, Shannxi Province, China in 2007. He has some projects in research supported by provincial funds. His research interest is Network Security.

Jin Jiang, he is a Lecturer in the Xi'an Polytechnic University, Shannxi, China. She received the M.E. degree from Xi'an Technological University, Xi'an City, Shannxi Province, China in 2010. She has some projects in research supported by provincial funds. Her research interest is Network Security.

Max Stinnett, he is an instructor in the Emporia State University, Emporia, Kansas, USA. His research interest is multimedia communication.

