ISSN 2047-3338

A Comparative Analysis of Deep Packet Inspection String Matching Algorithms Performance

Muhammad Asif Latif
Department of Computer Science and Engineering, University of Engineering and Technology, Lahore, Pakistan
asif.latif109@gmail.com

Abstract— **These days Deep packet inspection is the more enriched way of filtering networks at application layer level of OSI architecture. Deep packet inspection detects the packet content and the packet ID. Deep packet inspection devices deal with streaming packets in real time. DPI used for virus detection and for other harmful packets by looking for protocol non-compliance, intrusion or other described rules and principles on the on the behalf of which we can choose the packet or to be neglected or assign another destination. For signature comparison, many algorithms are used such as regular expression and string matching algorithms and many others which are discussed later. Now these days many applications depend upon the deep packet inspection for the inspection of the data packets which are passing through the network stream. This paper gives a brief idea about the challenges which are currently facing in the deep packet inspection and some of the design issues. After this some of the string matching algorithms are shortly discussed**.

Index Terms— **Deep Packet Inspection, DPI Challenges, Intrusion Detection System, Network Security and String Matching Algorithms**

I. INTRODUCTION

DEEP packet inspection perform network filtering by examining the packet ID of the Payload packet either by using the String matching algorithms like Wu-Manber etc. Or by using the regular expressions algorithms used in the NIDS of the Snort [1] and Bro [2] and L-7 Filter [3] in the Linux.

There are also some DPI Advantages like finding, categorizing, detecting and redirecting the packets which are not detected by the normal filtering methods. Also, Support Communication service providers to assigns different accessible resources to streamline traffic flow. It is also used to control transferring data to avoid network abusing in order to improve the network performance

Deep packet inspections also have some Disadvantages computer performance decreased with the increase of processor load. Deep packet inspection also increases the Complexity of firewall and other security related software’s. There are two methods to collect packets to DPI: (1) Port Mirroring and, (2) Optical Splitter.

The remaining paper includes the DPI implementation. In section III some of the Deep packet inspection challenges are discussed. Section IV illustrates the deep packet inspection design issues and objectives. Section V Illustrates the DPI string matching algorithms. Section VI Illustrates the performance analysis of different string matching algorithms. Finally section VII Concluding remarks.

II. DPI IMPLEMENTATION

Deep packet inspection (DPI) is classified into the three types of implementations, which are following [4].

Signature Based Identification: In the DPI different applications use different protocols and every protocol is labeled with different signatures. Every signature contains firm port or a bit sequence or string order. The detection is performed on the basis of the matching signature of the different packets which depends on the matching deep packet inspection. Matching DPI used to recognize the data flow which is going to the applications.

Behavior Based Identification: If the data flow is not recognized by the any of the protocol then we use the behavior based identification method is used to implement. Behavior based identification recognize based on the user behavior or specific terminal such as SPAM mails. H323 and Session initiation Protocol (SIP) are the applications which are recognized by the applications gateway.

Application layer Based identification: In many applications some of the control and service features are not available which cost’s a lot. For this case recognition of the application gateway is the only best way to recognize the control flow and the services flow at the defined protocol. H323 and session initiation protocol (SIP) are the applications which are recognized by the applications gateway. It will works by transforming data channels by exchanging it with specific signals.

III. DPI CHALLENGES

Deep packet inspection challenges are categorized into two groups, first challenges which are faced during the development of DPI system and second one is to build DPI [4].

- Challenges to develop a deep packet inspection system.
- DPI signature series challenges.

A) Challenges to Develop a Deep packet Inspection System

There are many challenges which are faced during the development of the DPI System [7] for the high performance it deals with the high speed links. This is one of the main challenges on the behalf of the some issues which are given below.

- Huge number of Signatures
- Difficulty level of the Signatures pattern
- Randomness of the Signature location in the packet data and in the stream of network
- How the Deep Packet Inspection implemented in the Hardware and in the OS

B) Deep Packet Inspection Signature Challenges

In the last few years, signatures are increased dramatically as comparing to the previous years. This happens due to the development of the advance systems, malicious Software's as well as other advance facilities. Deep packet inspection is used to verify the series of packet signature which are used in the millions of application signatures. As the Signatures are increased DPI performance is decreased. Deep packet inspection has the most complex rules as compared to the other systems like SNORT, L7 filters and XML filters.

C) Intrusion Detection System Challenges in DPI

When we talk about intrusion detection system in DPI we face many challenges. The challenges which are faced by INDS [5], [6] are summarized below:

The Complexity of Search Algorithm: Most Important factor of the Deep packet Inspection (DPI) is that it has different searching algorithms with different complexity. DPI is signature based system which is busy for the most of time because of the string matching. String Matching and the Running time both are very important for this an effective algorithm is essential.

Large number of Intruder Signatures: These days, more advance types of attacks has been developed. Due to this reason, Intruder signature series should be increased which distress the deep packet inspection system (DPI). The whole system should be scalable [6].

Signature Location: As comparing to the other systems deep packet inspection (DPI) checks the whole packets and the pattern as well.

Invalid/Incorrect Alarms: Intrusion detection Systems Produce many alarms from these Alarm's some of will be false alarms. All of these alarms should be reviewed.

Speed of Inspection: Speed of the communication is about 10GBE/OC192 and looming to 40GBE/OC768 [5]. For this deep packet speed should be increased.

Encryption of the data: Encrypted data is not inspected by the Deep Packet Inspection (DPI). So, inspection process would be implemented after the data decryption

IV. DESIGN ISSUES AND OBJECTIVE

In designing deep packet inspection faces many design issues; there should be group of things covered to meet with the pace of attack growth and growing the channel communication. There are following design objectives [5]:

System Scalability: The processor should be fast as comparing to the Scalability of the system with the software system which will not affect the system. In the scalability big problem is the restructuring as comparing to the hardware speed.

Quality of Service (QoS): Deep Packet Inspection is to the Provide the ISP services to their Customers. To meet this problem Deep packet Inspection Should Address the Quality of Service (QoS) problems like bandwidth management.

Availability of Signatures: For the secure communication deep packet inspection signature intruders must be always updated with the new intruders for the detection of the different types of the signature of intruders.

Memory Efficiency: Memory efficiency is very risky in the hardware and Software Insufficiency. Memory access time is the primary bottleneck in the deep packet inspection.

V. ALGORITHMS

There are many types of the String Matching Algorithms (SMA) [11].

The Naive SMA: The naive string matching algorithm (SMA) also known as string searching algorithm used to search one or many patterns of similar strings within a string or text, as you can see in the algorithm mentioned in Fig. 1. Naive SMA have no need of extra pre-processing, phase needs extra space. It shifts the one string to the right. The naïve SMA needs 2N text/String characters comparisons [8], [9], [10].

P = Pattern, T=String/Text

S = Shift, P [m] = Length of Pattern, P [0, 1…… m-1]

T [n] =Length of Text or String, T [0, 1…… n-1]

P occurs with Shift in T If 0≤ S ≥ n-m

T[s+0…s+n-1] = P [0, 1…..m-1]

If P occurs with shift S in T Then we call S is Valid Shift
Otherwise Sis Invalid Shift

Consider the given example.

T = ANPANMAN

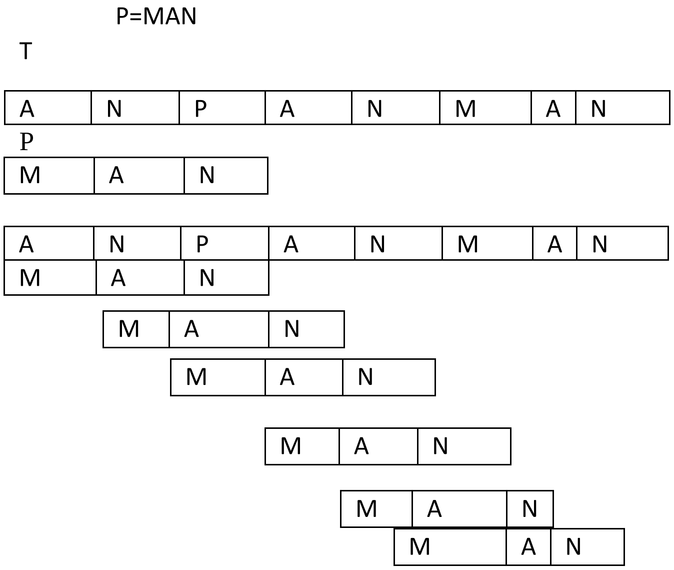

Fig. 1: Naive string matching

String Matching with Finite Automata: String matching with finite automata (SMFA) checks all the characters/text accurately once and show all the valid shifts in the O (n) time. Every Character/text has a State and every state sends the automation to the next state. If pattern of the text/character matches then the Automaton goes into the accepting state. Otherwise Automaton goes into the Appropriate State according to the current state or the input character repaid the maximum benefit we can take from the prior matching [10].

The Rabin-Krap SMA: The naive string matching algorithm (SMA) is used to search one or many patterns of similar strings within a string or text. The Naïve String Matching Algorithm slides the whole pattern one by one and checks the match of the pattern. But in the Rabin-Krap SMA the hash value of the pattern is matched with the hash value of the current substring of the character/text. If the match found then it will start matching all the characters/text individually as you can see in the Fig. 2 [11], [12].

```
Text/Character: L L M L L N L L O L L M L L M L

Pattern:          L L M L
   L L M L           L L M L
   L L M L L N L L O L L M L L M L
   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Result: Pattern Found at Step 1,9,12.
```

Fig. 2: Rabin-Krap SMA

The Knuth-Morris-Pratt SMA: The Knuth-Morris-Pratt SMA algorithm was perceived in 1970 by Donald Knuth, Vaughan Pratt and James H.Morris. The Knuth-Morris-Pratt string searching algorithm checks the existence of word w from the main string, if any mismatch occurs, the word w themselves determine whether to begin next or not through this way re-examination of the previous match could be avoided.

VI. CONCLUSION

This paper introduces the Comparative performance analysis of the Deep packet Inspection string matching Algorithms. The paper also focuses on the various design and implementation issues, limitations as well as the Performance of the different string matching algorithms. Deep packet inspection plays a vital role in the Field of network security can detect any malicious viruses or packets by using the different algorithms. By using the different string matching algorithms performance could also be increased as well as the network security.

REFERENCES

[1] "Snort v2.9.9.9," 2016. [Online]. Available: http://www.snort.org.

[2] "Bro Intrusion Detection System," 2014. [Online]. Available: http://www.bro.org/.

[3] "Application Layer Packet Classifier for LINUX," 2009. [Online]. Available: http://www.l7-filter.sourceforge.net/.

[4] R. T.-M. R. T. El-Maghraby, "A survey on deep packet inspection," 2017 12th International Conference on Computer Engineering and Systems (ICCES), pp. 188-197, 2017.

[5] A. A. Chaudhary, "Software Based Implementation Methodologies for Deep Packet Inspection," 2011 International Conference on Information Science and Applications2011 International Conference on Information Science and Applications, pp. 1-10, 2011.

[6] T. T. AbuHmed, "A Survey on Deep Packet Inspection for Intrusion Detection SystemsA Survey on Deep Packet Inspection for Intrusion Detection Systems," 2008.

[7] S. Rafael Antonello, "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends," Journal of Network and Computer Applications, vol. 35, no. 6, pp. 1863-1878, 2012.

[8] S. R. Janani, "String matching algorithms for reteriving information from desktop— Comparative analysis," in 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, 2016.

[9] Q. Y. Liu, "A factor-searching-based multiple string matching algorithm for intrusion detection," in 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, 2014.

[10] H. N. AbdulRashidx, "Maximum-shift string matching algorithms," in 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2014.

[11] J. L. a. C. Trefftz, "A comparison of the performance of four exact string matching algorithms," in 2007 IEEE International Conference on Electro/Information Technology, Chicago, IL,, 2007.

[12] D. L. Zhang, "Improvement on Wu-manber multi-pattern matching algorithm," in Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, Dalian, 2013.

[13] R. D. Pao, "Optimized Aho-Corasick string matching algorithm for smart phones," in 2016 IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, 2016.

[14] H. N. AbdulRashidx, "Maximum-shift string matching algorithms," in 2014 International Conference on Computer and Information Sciences (ICCOINS).

Table I: Performance Analysis of Different String Matching Algorithms

Algorithms	Application Method	Complexity			Remarks
		Processing		Searching	
		Space	Time	Time	
Aho-Corasick Algorithm	This algorithm finds finite set of strings from the given text. It bouts all the pattern's at once only. So algorithm has a linear complexity if we see length of pattern including the length of examined text and output matched [12].	A:O(i+k)	A:O(i+k)	A:O(j)	For the substrings matches there will be the Quadratics number of matches available
Wu-Manber Algorithm	This algorithm is Hash Based and it is modified form of the Boyer-Moore Algorithm. This algorithm works on the skipping characters concept and it also treasures matching candidate [13].	A:O(i+k)	A:O(i+k)	A:O(Bj/i)	This algorithm is used for large patterns for the fastest searching.
Quick Search Algorithm	This is the Calmest application of BM algorithm and it works only on bad-character Shift table.	A: O(I +σ)	A:O(σ)	A: O(I j)	very firm for short patterns
Optimal Mismatch Algorithm	Another type of simplified Quick Search algorithm, Effective when the Patterns are examined from least recurrent to most recurrent one.	A: O(i2+σ)	A:O(I +σ)	A: O(I j)	All the frequencies should be available in the advance for all the characters of the alphabets.
Maximal Shift Algorithm	In this algorithm character are scanned from the larger to the shorter shift. By using this method shift length would be maximized [14].	A: O(i2+σ)	A:O(I +σ)	A: O(I j)	Worst time complexity in the case of Quadratic.
Karp-Rabin Algorithm	This algorithm uses the Hashing method for Preprocessing to decrease the Quadratic complexity.	Persistent space	A:O(i)	A: O(I j)	predictable running time W:O(j+i)
KMP Skip Search	Advanced form of Skip Search algorithm. This algorithm uses two tables from the Morris-Pratt and KMP for the Linear Skip Search.	A: O(i)	A:O(i)	A: O(I j)	The comparison of the text character is following: W:O(log σ (I).(j / (I log σ (I))))

W: Worst Case, i: Length of Pattern, a: Average case, j: Length of text, B: best case , (σ): size of alphabet No: description is not available, K: z is the number pattern occurrence, W: Worst Case, i: Length of Pattern, a: Average case
j: Length of text, B: best case , (σ): size of alphabet, No: description is not available , K: z is the number pattern occurrence