Design and Implementation of Stateful Packet Filtering Firewall and optimization using Binary Decision Diagram

A THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

Bachelor of Technology

in

Computer Science & Engineering

By

Anil Kumar

(Roll No. 110CS0041)

Department of Computer Science & Engineering

National Institute of Technology

Rourkela (769008), India

May 2014

Design and Implementation of Stateful Packet Filtering Firewall and optimization using Binary Decision Diagram

A THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTSFOR THE DEGREE OF

Bachelor of Technology

in

Computer Science & Engineering

Submitted to

National Institute of Technology, Rourkela

By

Anil Kumar

(Roll No. 110CS0041)

Under the supervision of

Dr. S. K. Jena

Professor

Department of Computer Science & Engineering

National Institute of Technology

Rourkela (769008), India

May 2014

Department of Computer Science & Engineering

National Institute of Technology

Rourkela (769008), India

C E RT I F I C A T E

This is to certify that the thesis entitled ***Design and Implementation of Stateful Packet Filter Firewall and optimization using Binary Decision Diagram***, submitted by **Anil Kumar** (Roll No: 110CS0041) has been carried out under my supervision in partial fulfilment of the requirements for the degree of ***Bachelor of Technology*** during session 2010 - 2014 in the Department of Computer Science & Engineering, National Institute of Technology, Rourkela.

To the best of my knowledge, this work has not been submitted to any other University/Institute for the award of any degree or diploma.

Dr. S.K. Jena

Professor

Department of Computer Science & Engineering

National Institute of Technology, Rourkela

India 769008

Place: Rourkela

Date:

A C K N O W L E D G E M E N T

I would like to express my deep sense of gratitude and respect to my supervisor **Prof. S. K. Jena** for his invaluable guidance, motivation, constant inspiration and above all for his ever co-operating attitude that enabled me in bringing up this thesis in the present form. I consider myself extremely lucky to be able to work under the guidance of such a dynamic personality.

I am grateful to **Prof. S. K. Sarangi**, Director, National Institute of Technology, Rourkela who has been a constant source of inspiration for me. I am also thankful to Prof. **S. K. Rath**, H.O.D, Department of Computer Science & Engineering, National Institute of Technology, Rourkela for his constant support and encouragement.

Last but not the least I would like to thank **Ashish Kumar** (Ph.D. Scholar, Information Security), and my all friends with whose additional help this study has been a succulent one.

Place: Rourkela

Date:

Anil Kumar

Roll No. 110CS0041

Department of Computer Science & Engineering

National Institute of Technology, Rourkela

India 769008

Author's Declaration

I hereby certify that all the work contained in this report is done by me unless otherwise acknowledged. Also, all of my work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

Anil Kumar

110CS0041

NIT ROURKELA

Figure List

Figure 1. Firewall Schematics
Figure 2. Packet Fitler
Figure 3. Circuit Level
Figure 4. Application Level Gateway
Figure 5. Stateful Inspection Firewall
Figure 6. Server Report
Figure 7. client window
Figure 8. Firewall Rules
Figure 9. log window
Figure 6. A Simple Decision Diagram for ($x1$ or $x2$) and $x3$
Figure 7. Example of BDD
Figure 8. ROBDD Representation
Figure 9. Flow Chart of Current Proposed System
Figure 10 number of nodes in Binary Decision Diagram using user order and CUDD Generated Order
Figure 11 Comparison of graph for most-rejected packets
Figure 12 Graph Comparison between list based and BDD based

Table List

Table 1 For the representing the Access List total no of Boolean variables required
Table 2 Sample Access List containing Rules
Table 3 Using User Order and CUDD Order number of nodes varies
Table 4 Comparison for Most Reject Packets
Table 5 Comparison for Most Accept Packets

Contents

Chapter no.	Description	Page no.
Chapter 1	**Introduction**	**9-13**
	1.1 **Firewall**	**9**
	1.2 Types of Firewall	**9**
	1.3 Packet Fitler	**9**
	1.4 Circuit Level	**10**
	1.5 Application Level Gateways	**11**
	1.6 Stateful Packet Fitler	**12**
	1.7 Firewall Limitation	**13**
Chapter 2	**Implementation of Stateful Packet Filtering**	**15**
Chapter 3	**Optimization of Firewall Rule**	
	3.1 Binary Decision Diagram	**17**
	3.2 Ordered BDD (OBDD)	**18**
	3.3 Tradition List Based Packet Filtering	**18**
Chapter 4	**Design and Development Details**	
	4.1 Development of a Module to Generate .blif files	**21**
	4.2 Development of a Packet Filter	**22**
Chapter 5	**Results and Analysis**	
	5.1 Result of Real World Access List	**24**
Chapter 6	**Conclusion**	
	6.1 Conclusion	**27**
Chapter 7	**Reference**	**28**

A B S T R A C T

Today internet is the most useful and big source of knowledge. We can find any information on the internet. But at the same time we are exposed to different types of attacks such as spoof Packet filtering, Denial of Service Attack and so on. So we have to secure the network from this type of attack so that we can easily find information without any hiccups. Through Firewall we can secure our network form this type of attack. There are so many types of Firewall currently exist. But we focus specially on Stateful Packet Firewall.

Stateful Packet Filtering in improved version of packet filter firewall in which it validates the first packet of the new connection according to the firewall rule. If that packet is satisfied by the firewall rule policy than corresponding entry is created in state table so that for consecutive packet of the same connection will not be validated by firewall rule. It checks only that packet is corresponding to the existing connection or not. If packet is of existing connection then it will immediately passed through firewall, no need to check according to firewall rule and if packet is of the new connection then it is passed through firewall if and only if it validates the rule and accordingly it will create entry in state table.

But there exist problem when the rule list is large in number. Today firewall rules contains thousands or lacks of rule. So it will take long time to decide for a packet to be allowed or not. So we can improve this look up time by using Binary Decision Diagram (BDD). BDD is compressed data structure that will decide immediately that if packet should be passed or not. Operation are performed directly on compressed data structure. On testing on millions of packets the look up time is decreases up to 74%.

Chapter – 1

INTRODUCTION

1.1 Firewall

Security remains a top concern with expanding dependence on the Internet because of the expansion in number of potential wellsprings of assault systems oblige insurance from unintentional episodes and pernicious acts. Any new neighborhood is associated with the Internet hosts display in that system get powerless against the assault. Subsequently, to conquer this security issue, these frameworks is arranged precisely so the configuration of the system frameworks gets vigorous. A large portion of the complexities in utilizing firewalls today exist in dealing with an extensive number of firewalls and guarantee whether they uphold a predictable arrangement over an association's system.

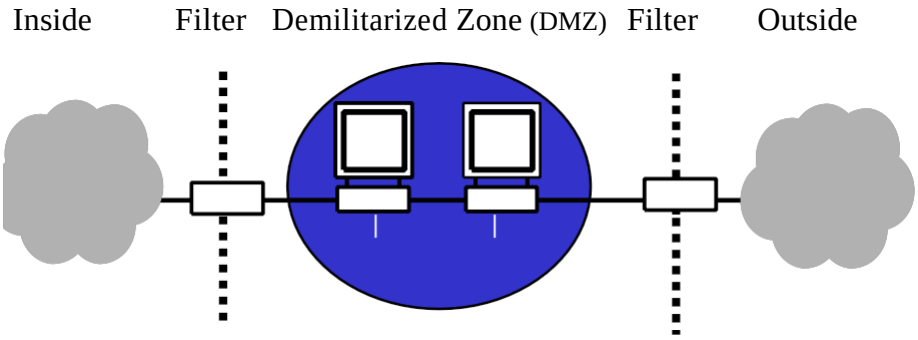


Figure 1: Firewall Schematics

1.2 Firewall Types

There are numerous four sorts of firewall are we utilized. In this segment we demonstrate each of them to sum things up.

1.3 Packet Fitler

Packet Filter Firewall is the simplest and fastest firewall which is used to decide if packet is allowed through firewall or not. It operates on network layer of the OSI model. Packet Filter firewall works on five tuples i.e. source address, destination address, source port, destination port and protocol. For a new connection of client each packet is individually checked by firewall rules if it satisfied the rule then it is accepted otherwise it is rejected. It is also called stateless protocol firewall. Because it doesn’t maintain any state of the connection i.e. it does not remember that consecutive packet is of the same connection or not. It simply checks each

and every packet of the connection. At any time if any packet doesn't satisfy the rule it simply discards the packet.

Advantage of this packet filter firewall is that it is easy to implement and easy to understand and it is fast enough if client is less in number.

Disadvantage of Packet filter firewall is that if it does not maintain any state of connection and attack known as Anti-Spoofing can easily be done on this firewall.

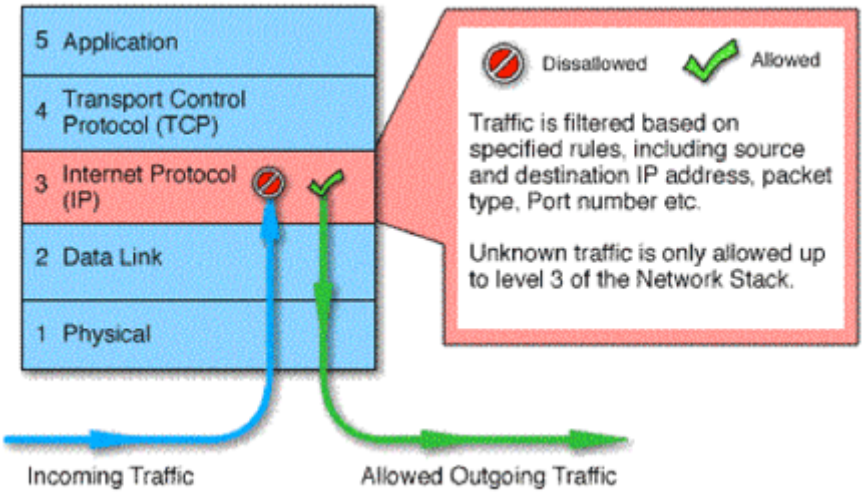


Fig 2. Packet Fitler

1.4 Circuit level(CL)

CL gateways operate in session layer of Open System Interconnection model, or the Transmission Control Protocol layer of TCP protocol. To check whether a session is genuine they screen TCP handshaking between packets. Information coming to a remote computer through a CL gateway seems to have begun from the same gateway. This is valuable for concealing important data about secured networks. CL gateways are moderately modest in cost. It has the advantage of concealing data of the private network they ensure. Then again, individual packet filtering is not done by them.

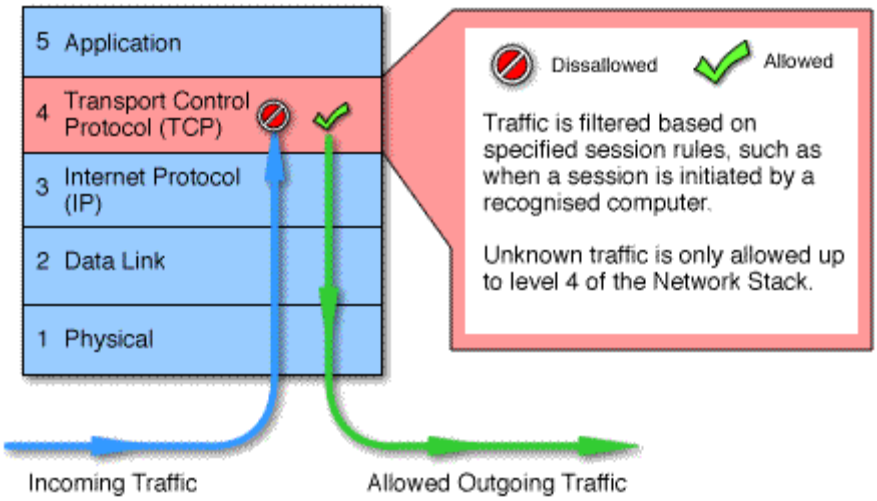


Fig 3. Circuit Level

1.5 Application level gateways

Application level gateways are like circuit-level gateways with the exception of that they are application particular. They are also known as proxies. Packet filtering is done by them at the application layer. If there is no proxy, incoming or outgoing packets can't accesses to the services. In layman terms, an application level gateway is arranged to be a web proxy won't permit any gopher, ftp, telnet or other movement through. Since they look at packets in application layer, they can refine commands, for example, http:post and http:get, and so on. Gateways of application level can likewise be utilized to log client action and login. They offer huge amount of security, yet have a critical effect on network performance. This is a direct result of context switches that reduce the network access significantly. They are not transparent to end clients and hence they need manual configuration of every end machine.

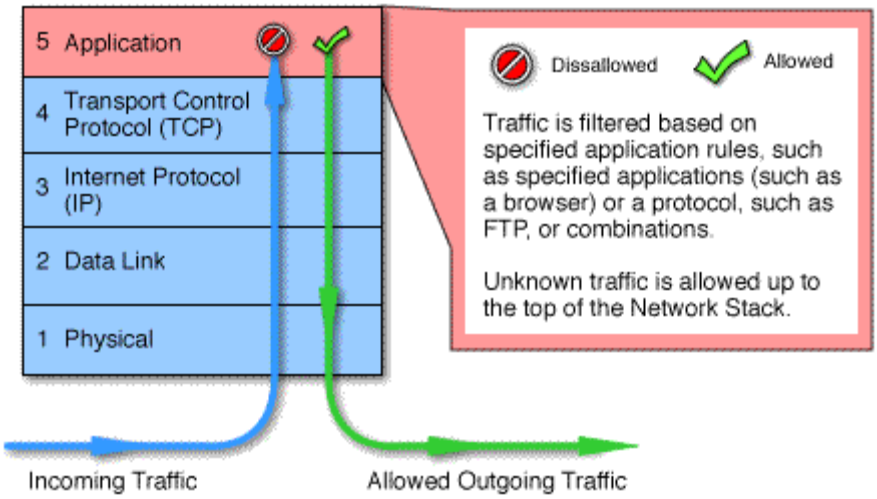


Fig 4.Application Level Gateway

1.6 Stateful Packet Filter

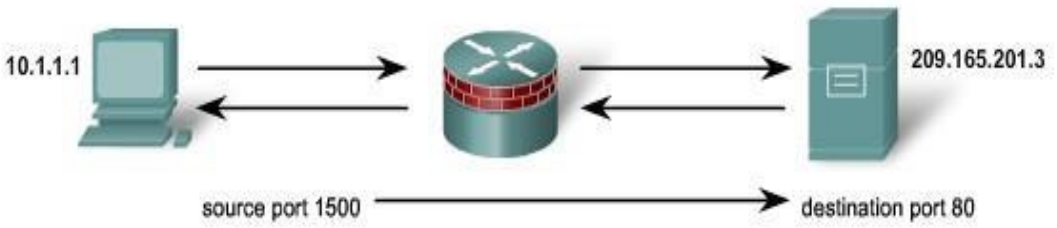
Stateless packet filtering otherwise called static packet filtering can't perceive whether the packet is some piece of a current stream of information or not. Permit or deny decisions are made on packet by packet basis instead of focus around past permitted or denied packets. As come about, a stateless packet filtering firewall is not intelligent and could be tricked by expert hackers. To tackle this issue, stateful inspection firewall was firstly created by Bill Cheswick and Steve Bellovin at At&t Bell Laboratories in 1989.

Stateful inspection firewalls are focused around packet filtering firewall and can screen the connection state, for example, initiation, information exchange and termination. To bring up that, the terms to describe the association state may be a bit not the same as different firewall items however very comparative. This sort of firewall is otherwise called dynamic packet filtering firewall and it utilizes information at layer 3, 4 and 5 of the OSI model.

Communication process is tracked by Stateful investigation firewalls by utilizing a state table. Following data is about the Cisco stateful firewall's state table. A stateful firewall will log data in the session flow table each time a TCP or UDP connection is established. The session stream table incorporates source and destination address, TCP sequencing data , port numbers, and flags for TCP or UDP association in each session. At the point when firewall accepts a packet from outside network, the accepted packets will be compared with saved state table data to settle on the allowed or denied choices. Also, for stateful inspection

firewalls which include the network address translation (NAT) characteristic, their state tables will incorporate some NAT data.

More progressive stateful inspection firewalls can upgrade state table and permit returning movement started from inside network. This diminishes DNS cache poisoning (which could result in the name server to return erroneous address which could make the attackers' device act as Domain Name Server to trap clients trying to comunicate with them) and TCP RST flood attacks (a sort of Denial of Service strike by sending multiple SYN packets to targeted clients). In any case, as stateful inspection offers transparency, interior IP addresses are uncovered as indicated in Figure 5 below. This could result in potential dangers for the network.



Inside ACL (Outgoing Traffic)	Outside ACL (Incoming Traffic)
permit ip 10.0.0.0 0.0.0.255 any	Dynamic: permit tcp host 209.165.201.3 eq 80 host 10.1.1.1 eq 1500 permit tcp host 209.165.201.3 host 10.1.1.1 eq 22 deny ip any any

Figure 5. Stateful Inspection Firewall

1.7. Firewall Limitation

Data security experts frequently end up working against confusion and well known estimations structured from inadequate information. Some of the opinions result more from trust than actuality, for example, the thought that inward system security might be comprehended basically by sending a firewall. While history has proven time and again that firewalls play a vital and focal part in the upkeep of system security and any sanctioned n that overlooks them, does so at its risk, they are not, one or the other the solution of each

security part of a system, nor the lone sufficient rampart against interruption. Comprehending what firewalls can't do is as critical as realizing what they can.

2. Implementation of Stateful Packet Filtering Firewall

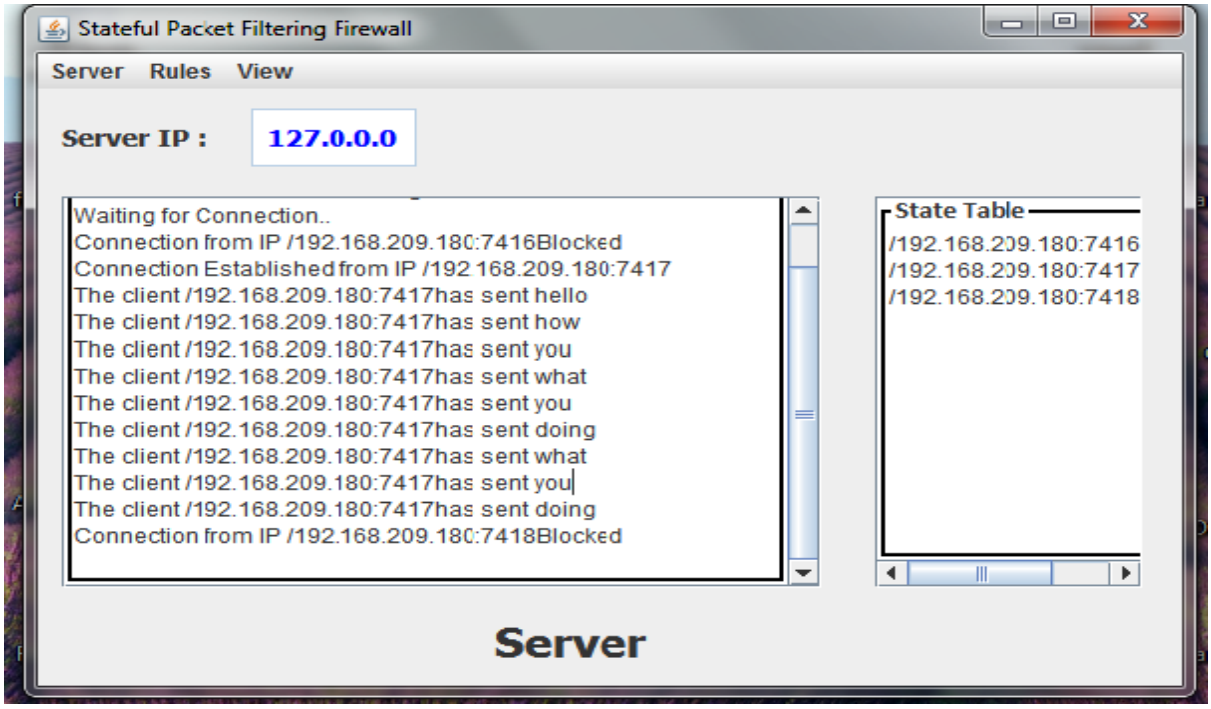


Fig 6. Server Report

It is the GUI showing how any client is connected to the server and how they generate the report. If a connection satisfies the policy rule then then it will show that connection established and if it does not satisfied the rule then it will show that connection is blocked.

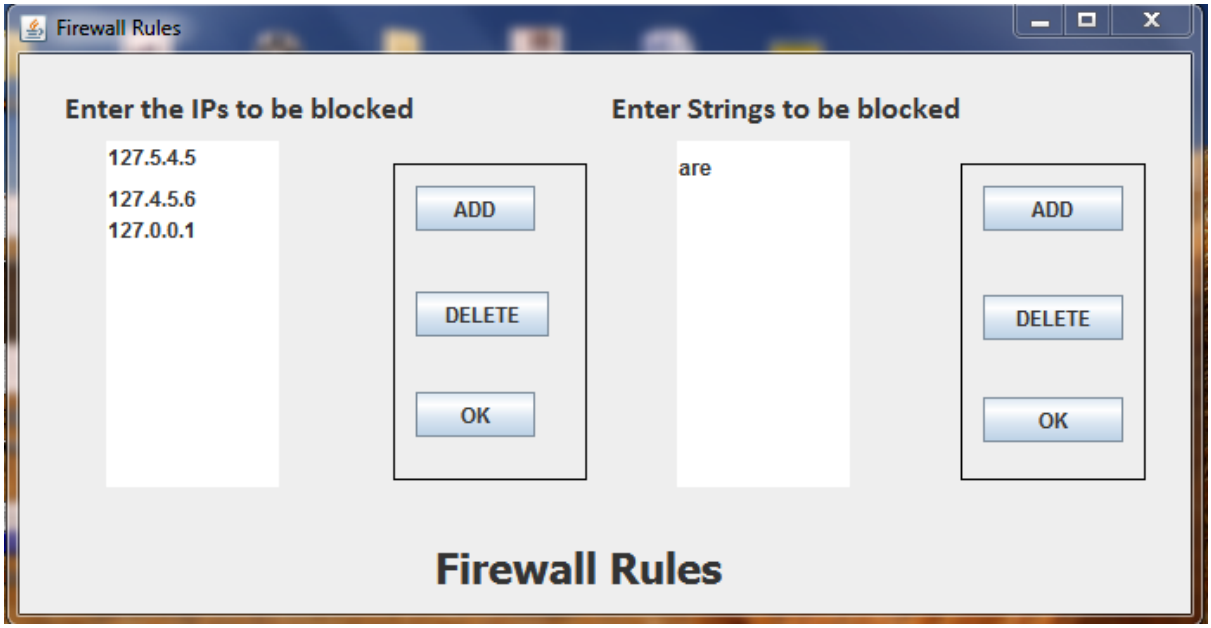


Fig 7. Firewall Rules

This window has two main functions done by the server administrator. One part is for IPs to be blocked and other part is for the strings to be blocked in the message. Additional rules can be added or deleted.

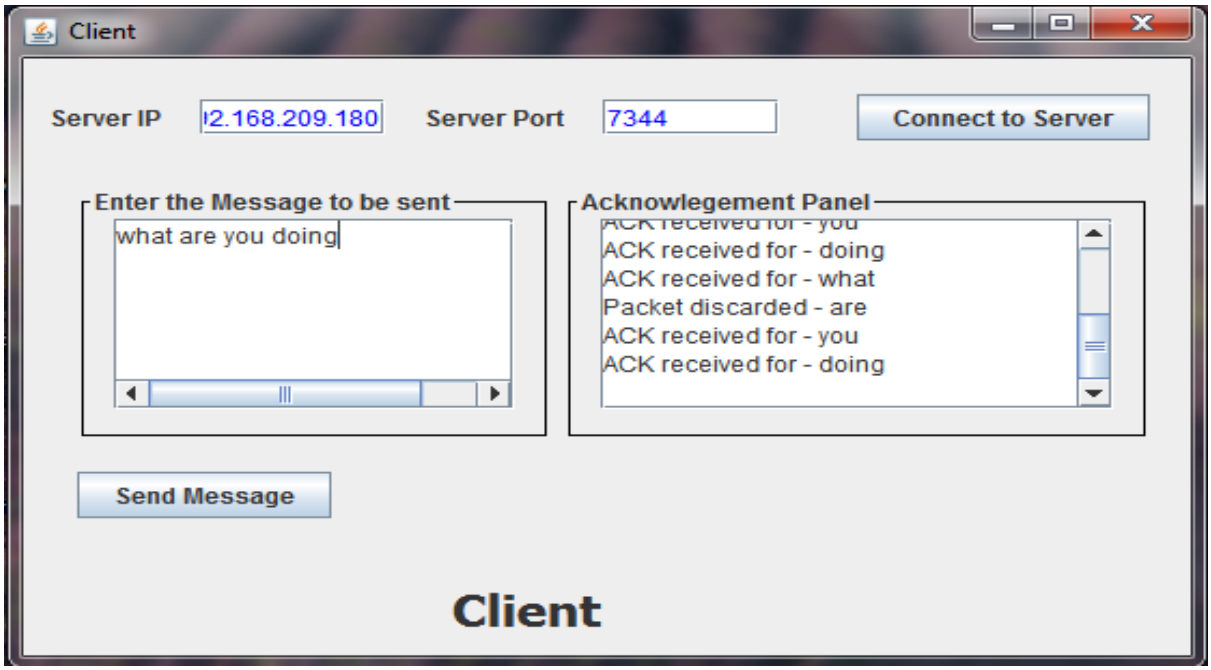


Fig 8.client window

This window shows the Client window. It takes server IP and Port number as texbox input to connect to the server. One part is for sending the message and other part is the acknowledgement panel in which it shows which packet is accepted or denied.

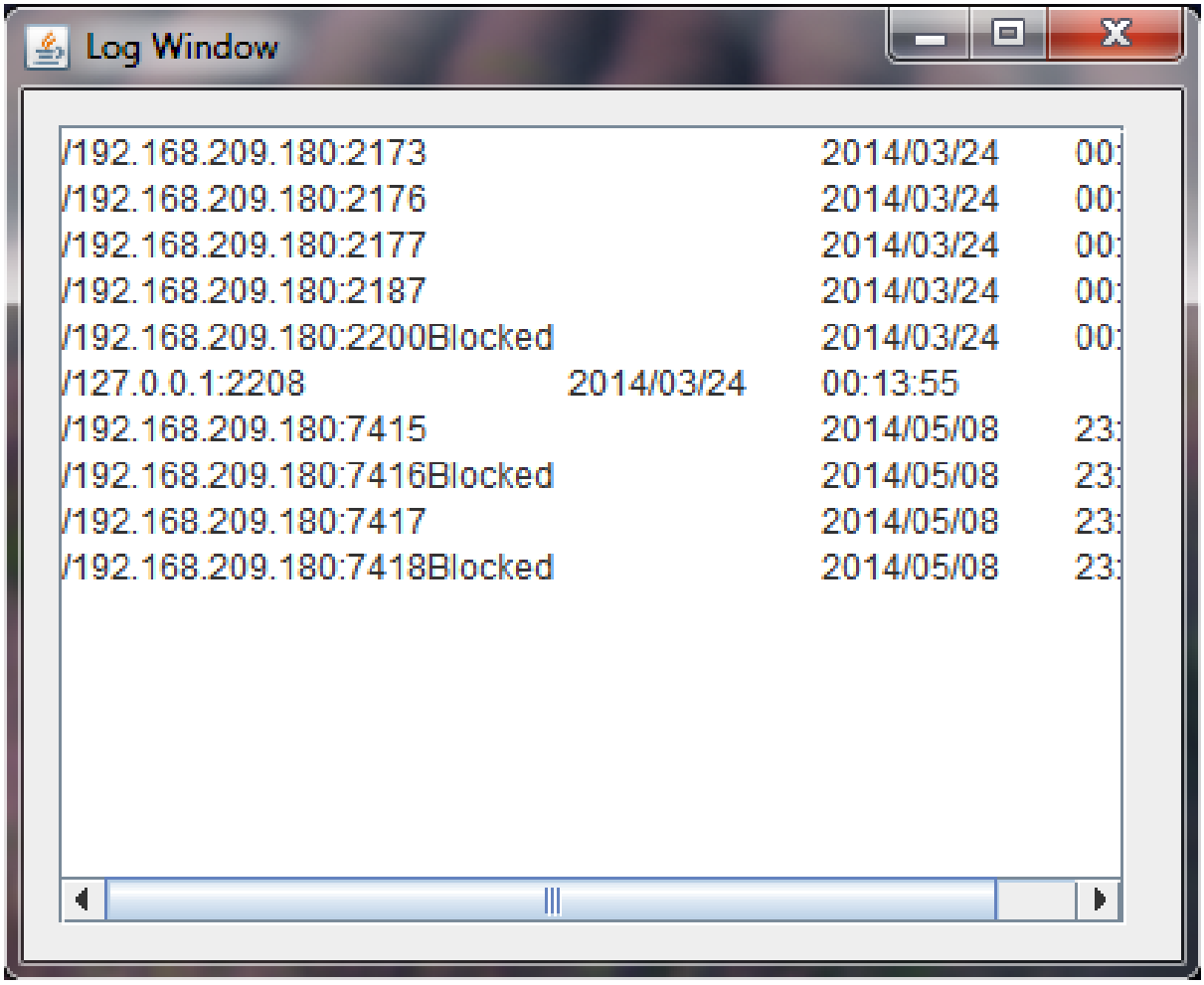


Fig 9.log window

This window is Log Window. It can only be opened by the server administrator. It shows which client has connected when in addition to their connection status i.e. whether they are accepted or denied.

3. Optimization of Firewall rule

3.1 Binary Decision Diagram (BDD)

Binary Decision Diagram (BDD) or Duple Decision plan or fanning system is an information structure that is utilized to speak to a Boolean capacity [bryant, 1986; Bryant, 1992]. Not at all like other layered representations, operations are performed specifically on the packed representation. A Boolean capacity might be spoken to as an established, guided non-cyclic chart, which comprises of a few choice nodes and terminal nodes. There are two sorts of terminal nodes called 0-terminal and 1-terminal.each choice node N is named by Boolean variable VN and has two tyke nodes low kid and high kid. The edge from node VN to a low (or high) youngster speaks to a task of VN to 0 (resp.1).to assess articulation given a translation of the variables, you begin at the root and move downwards. For instance the Boolean outflow (x1or x2) and x3 could be spoken to by the choice graph in figure 1.

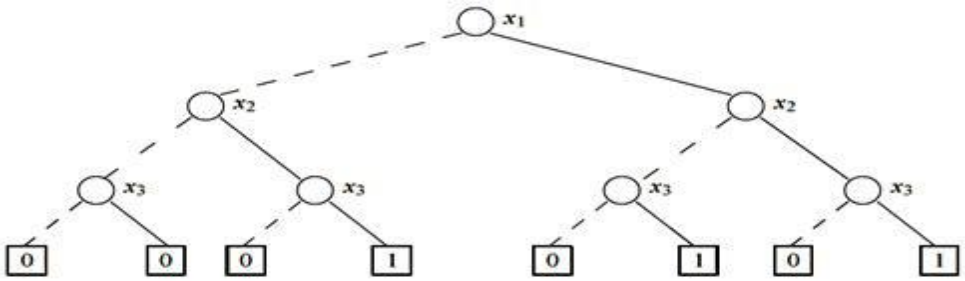


Figure 9 – A Simple Decision Diagram for (*x*1 or *x*2) and *x*3

Table 1 – For the representing the Access List total no of Boolean variables required

Dimension Field	Boolean Variables	Total Number
Protocol Type	p7,p6,……..p1,p0	8
Source IP Address	sa31,sa30,……..sa1,sa0	32
Destination IP Address	da31,da30,…….da1,da0	32

Source Port	sp15,sp14,……..sp1,sp0	16
Destination Port	dp15.dp14,……..dp1,dp0	16
Total		104

3.2 Ordered BDD (OBDD)

in the event that every variable of the BDD shows up at most once in each one complete way and if the variables show up in the same request in all other complete ways. Out of these variants of Bods ROBDD are most utilized sorts of Bods. These ROB Dds could be acquired utilizing CUDO bundle by giving BOD as data. Case of such ROBOD got from Fig. 1 is demonstrated in Fig 2. We can watch that number of nodes in the ROBOD is lessened by two as contrasted and unique BOO. Client characterized variable requesting can additionally be utilized to build the Bods. TABLE I is the sample of such variable requesting utilized within BDD development.

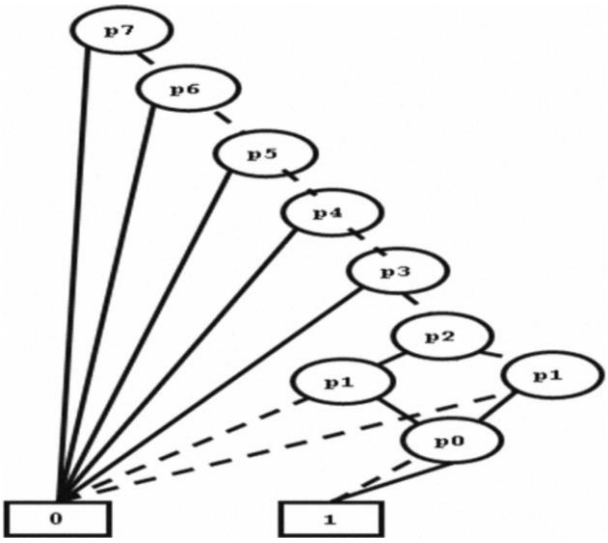


Fig 10.Example of BDD

Each packet of new connection contains header information like protocol, source_address, source_port, and destination_address and destination_port. All of them are represented using decimal numbers and can be changed in to the corresponding binary format. Consider, the protocol to be TCP and its number is 3, whose binary form is 00000011. 8 variables are

needed to store these 8 bits and their named can be as p7, p6, p5, p4, p3, p2, p1, p0. Top to bottom traversal done on the Binary Decision Diagram.

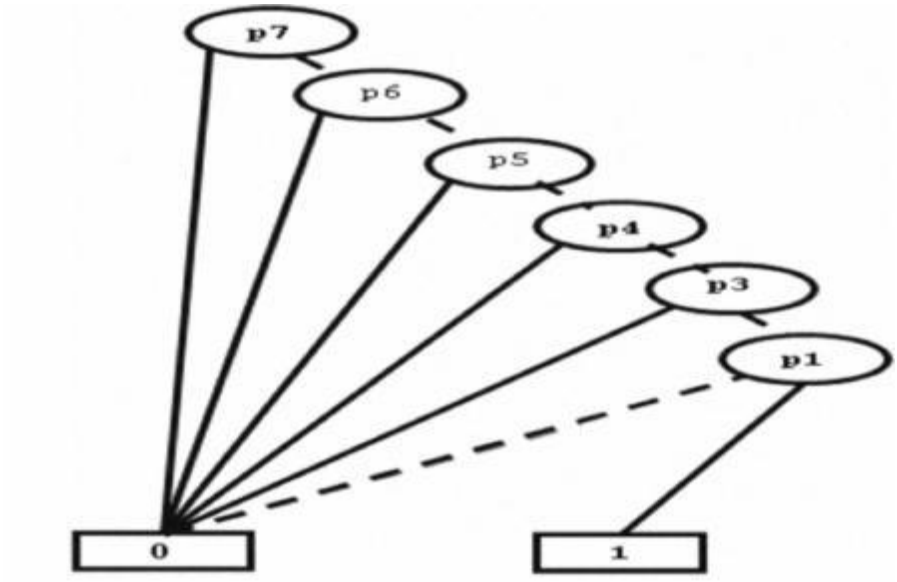


Figure 11 – ROBDD Representation

3.3 Traditional list based packet filtering

In traditional list based packet filtering, rule sets are in proprietary formats and rules are expressed in the form of if condition then action. To take decision rules are searched one by one to see whether the condition matches the incoming packets: if it does, the packet is accepted or rejected depending upon the action. If the condition does not match the rule, the search continues with the following rules. Thus the whole list is being traversed in serial fashion which takes too much lookup time to decide the fate of packet. The main disadvantage of list-based algorithm is that rules are placed without any prior knowledge like channel characteristics. This can cause the increase in time for making decision. To achieve a significant amount of gain in decision making the most occurring rule in accepting or rejecting a packet can be placed at the beginning of the rule set, this is called list-based approach with promotion. List-based approach is the best approach but it lacks in accuracy.

Algorithm/or list-based packet filter with promotion

1. Extract the incoming packet header data.
2. while access list is not empty final result = 1 do
3. if action = PERMIT then
4. ACCEPT the packet
5. rule hit count[rule no]++
6. else if action = DENY then
7. REJECT the packet
8. rule hit count[rule no]++
9. else
10. rule action is not defined
11. end if
12. end while
13. //for the default deny case
14. if access list is empty final result = 0 then
15. REJECT the packet
16. end if
17. //In the next iteration alter access list according to the rule hit count.

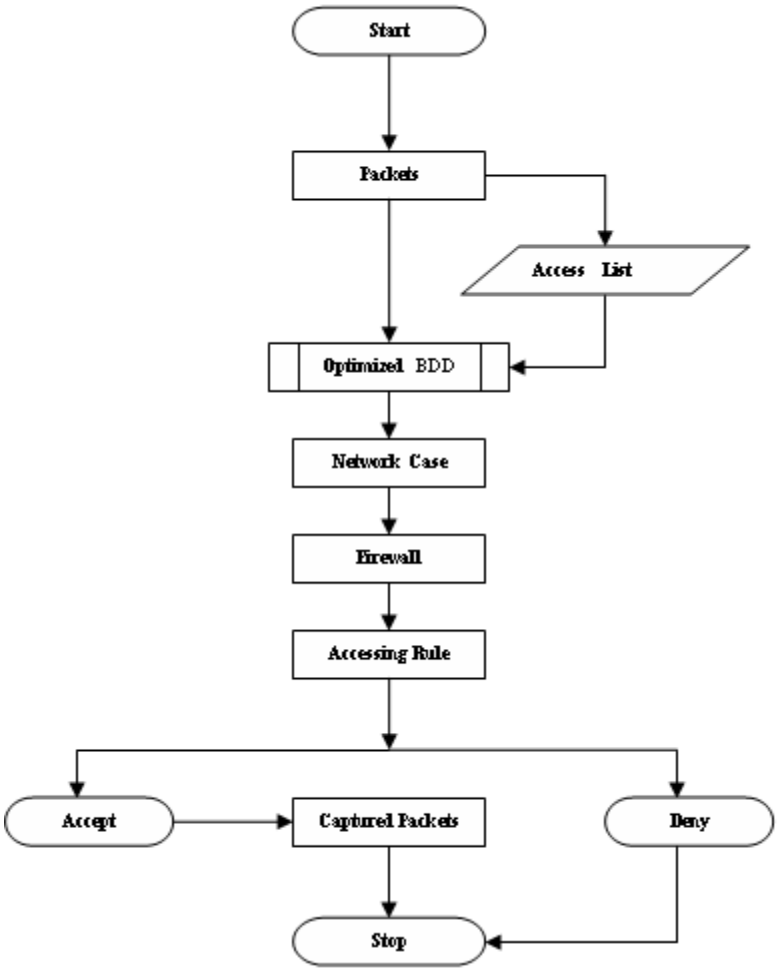


Figure 12 – Flow Chart of Current Proposed System

In our framework we will utilize the aforementioned BDD based methodology to speak to the guideline set in a Boolean declaration. A recursive capacity will be characterized that will change over the guidelines into Boolean statements. Once the BDD representation of the principle set has been constructed, it could be utilized for lookup. On the landing of packet the significant packet data might be concentrated. This header data gives a mapping from variables to truth values. Utilizing this mapping we begin at the foundation of BDD and works downwards until we get to a leaf. On the off chance that the variable at the current node is false move to the left, on the off chance that it is genuine move to the right. For choosing the activity of the packet the find calculation perform the processing without any redundancy. BDD speaks to multidimensional sifting and holds the all measurement values in binary arrangement. The expense of building BDD is extremely sensible and practical and lookup is possible rapidly Doing the same with straight representation of the principles is substantially more troublesome and expensive. The proposed framework will have particular

possibilities that can block and log movement passing over a computerized system or some piece of a system. As information streams stream over the system, the sniffer catches every packet and in the end unravels and investigates its substance can perform Deep Packet Inspection (DPI) to audit system packet information as per the suitable RFC or different details.

4. DEVELOPMENT AND DESIGN DESCRIPTION

In this area we are going to talk about the usage subtle elements and what are the modules actualized for the fruitful finish of the bundle channel firewall venture.

4.1. Algorithm for generating .blif File

I have executed a module which consequently takes the right to gain entrance sequence as data and gives the comparing .blif record as yield. We can input this .blif document as information to the CUDD bundle to get the advanced type of BDD as yield documents.

Algorithm: .blif File Generation

// Input: Rule List

// Output: .blif file containing the binary form of the rule list.

Do While (Rule is exist in the rule list)

1. Examine each rule from the rule list.
2. Take the protocol number, source_address and port, and destination_address and port.
3. Change each part in to its binary format in required number of bits
4. Store the each part of the converted binary form in to the .blif file.
5. end while

4.2. Stateful Packet Development

In this algorithm packet channel firewall takes the .dot document produced from CUDD as the info record and chooses the activity of the approaching and cordial packets.

Algorithm for BDD Lookup Algorithm
// Input: CUDD order, dot file.
// Output: ACCEPT or REJECT the packet.
Check the head of the BDD given by the CUDD
If (solid_line) then final_output = 1
else final_output = 0
lookup_pttr=first node of the BDD
while(lookup_ptr == 1 OR lookup_ptr == 0)
if(header lookup_ptr = low (lookup_ptr)
if(lookupytr == NULL)
then **ACCEPT** the packet
else **REJECT** the packet

Calculation looks the Binary Decision Diagram produced by the CUDD as indicated by the Binary Decision Diagram set. From that point it takes the every bit of the header and contrasts and the BDD. When each and every bit of the header are validated by crossing the BDD from top of tree to bottom of tree node the decision of packet's worthiness is acknowledged. Regularly the packet channel data is put away in a 2-D vector, which holds the data all BDD nodes and their high and low edge values.

5. RESULTS AND ANALYSIS

We have done numerous investigates the right to gain entrance records created by our own particular and additionally from this present reality access sequence. We have utilized Protocol Source-Destination (PSD) request for the analyses and discovered critical result in the execution of packet channel firewall. At first get to sequence length is situated to four to separate between a settled request and the best variable request produced by CUDD. Case in point, table 2 and table 3 delineate two sets of access records and the relating near effect is demonstrated in figure 8.

Table 2 – Sample Access List containing Rules

Rule#	Proto type	src_addr	src_mask	src_port	dest_addr	dest_mask	dest_port	Action
1	TCP	0.0.0.0	255.255.255.255	1023	146.141.0.0	0.0.255.255	25	Permit
2	IP	146.141.0.0	0.0.255.255	80	146.141.0.0	0.0.255.255	120	Permit
3	TCP	0.0.0.0	255.255.255.255	1023	146.141.0.0	0.0.255.255	80	Permit
4	IP	0.0.0.0	255.255.255.255	25	146.141.0.0	0.0.255.255	80	Permit
3	TCP	0.0.0.0	255.255.255.255	1023	146.141.0.0	0.0.255.255	1023	Permit

Table 3 –Variation of nodes using List based and BDD order

Number of Access List	Length of List	Number. of Nodes using List Based	Number. of Nodes using BDD based
1	11	94	94
2	22	192	128
3	33	204	132
4	44	210	138

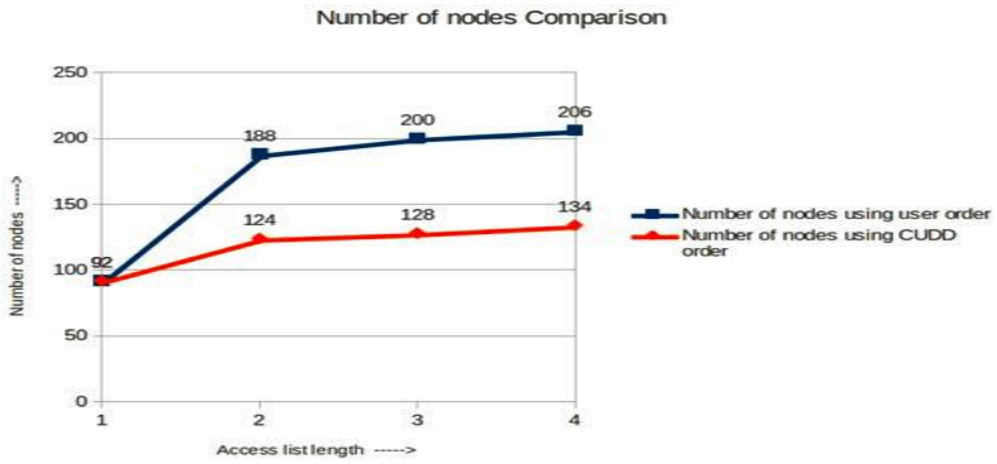


Figure 10 – Number of Nodes in BDD using User Order and CUDD Generated Order

From figure 8, it is understood that the amount of nodes utilizing the CUDD created request is much lower than the client given altered request. In this manner, stockpiling required for the comparing packer channel firewall might be decreased essentially.

5.1. Results for Real World Access List

For certifiable access sequence we have acknowledged the right to gain entrance control sequence from a genuine LAN, which has around 262 tenets. I have taken introductory 44 tenets as first get to sequence. We have utilized this first get to sequence for the both the firewalls (BDD based and sequence-based with advancement) as information. After that, in each one stage, we have added 44 more leads to get an alternate set of effects. For everyone right to gain entrance sequence we have examined the amount of correlations needed for the deny or acknowledgement of the given bunch of data packets, that are produced arbitrarily. In BDD-based packet channel the amount of correlations is the amount of nodes went by the firewall to choose the acknowledgement or dismissal of the packets. In sequence-based packet channel number of correlations is the amount of guidelines it has gone by and in each one guideline what number of packets it has checked. We have created two separate sorts of convention; one is "most-acknowledge packets" and an alternate is "most-reject packets". On account of "most-reject packets" convention BDD-based packet channel will navigate lesser nodes in the BDD to choose the destiny of the packet. At the same time on account of sequence-based packet channel the entire list of guidelines in the current access sequence will be navigated. Another kind of info information we utilized is the "most-acknowledge packets" convention. For this situation BDD-based packet filer the calculation navigates the

BDD from the start to finish to choose the packet's acknowledgement. However if there should arise an occurrence of sequence-based methodology the approaching packet may match with any standard in the right to gain entrance sequence. Thus, the aggregate number of correlations obliged will be relying upon the position of the matching manage in the right to gain entrance sequence. Interestingly, in the second case likewise our outline gives better come about contrasted with sequence-based packet channel firewall. Table 4 and 5 show the normal number of examinations taken by both the firewalls to choose it acknowledgement or dismissal on the provided for one million arbitrary packets. On account of the "most-reject packets" the execution of the BOD-based packet channel is higher as the amount of examinations is altogether less. For one million approaching packets with access sequence sets from 44 to 262 the normal diminishment in number of examinations for BDD-based packet channel firewall is 75% when contrasted and sequence-based packet channel firewall and "most-reject packets" convention is picked. For "most-acknowledge packets" convention the normal diminishment is almost 34%, however for 80-262 right to gain entrance sequence sets this lessening is surprisingly better, about half. Thus, for both the conventions the execution of BDD-based packet channel firewall is fundamentally superior to the schedule-based packet channel firewall framework. Table 4 and 5 Comparative comes about between BDD based and List based methodologies Figure 9 and figure 10 represent these near effects in graphical configuration.

Table 4 – Comparison for Most Reject Packets

Most Reject Packets	No. of Compares on node with 1 Million Packets		
Length of Access List	Based on BDD	Based on List	Reduction in percentage
40	32054234	55668183	63.46
80	32026791	120134708	73.34
120	32537347	171710572	81.07
160	42643811	198408729	78.50
200	48068402	215368648	77.68
267	49690340	220103588	77.42
		Average Reduction = 75.24	

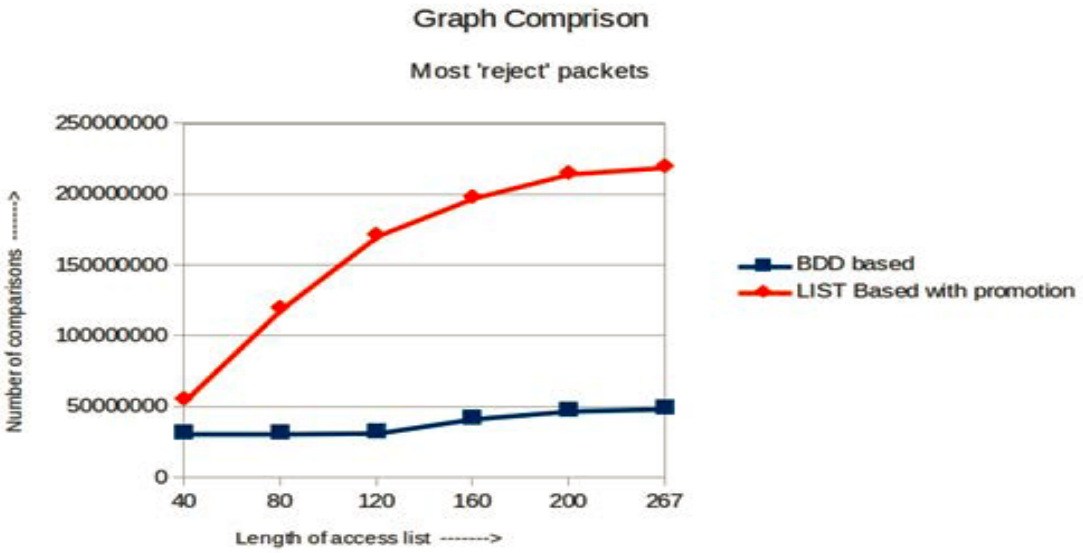


Figure 13 – Graph Comparison for "Most-Reject Packets"

Table 5 – Comparison for Most Accept Packets

Most Accept Packets	No. of Compares on node with 1 Million Packets		
Length of Access List	Based on BDD	Based on List	Reduction in percentage
40	60386771	40360524	-49.61
80	58431012	68786826	15.05
120	57783157	100364081	42.47
160	58144886	129886903	55.23
200	55868916	164023455	65.93
267	55555756	208888173	73.40
		Average Reduction = 33.74	

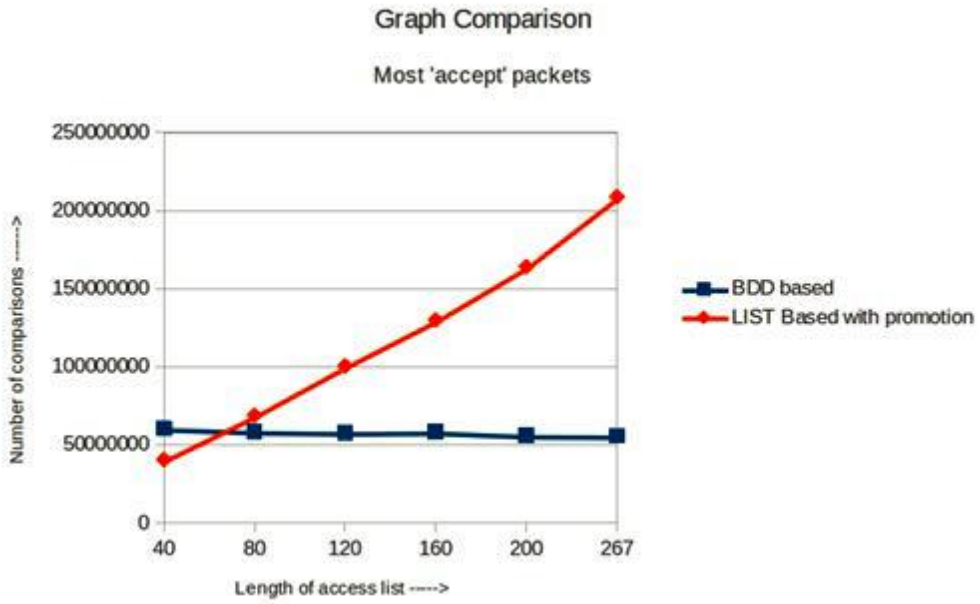


Fig.14. Graph Comparison between list based and BDD based

6. CONCLUSION

Aim of this work is to explore basic design and implementation of stateful packet filtering firewall which is more secure than packet filtering firewall and alternative representations for rule sets. We use Binary Decision Diagram for effective and efficient representation of rule set and to reduce the lookup time for satisfying the rule. It will helps in making fast decision to accept or reject the packet. At this stage the focus is exploring the „back end" – internal representations of the rule set that can be used to improve lookup performance and provide network managers a tool to understand and analyse the rule set. The efficiency is considerably decreases when we compared List-based packet filter with Binary Decision Diagram-based. We remove the redundant computation using BDD approach and implement the multidimensional filtering. Our duple decision diagram or Binary Decision Diagram approach takes less space in terms of memory and takes less time to decide a packet that it is accepted or not.

****

References

[1] Paul, G. ; Dept. of Comput. Sci. & Eng., IIT Kharagpur, Kharagpur, India ; Pothnal, A. ; Mandal, C.R. ; Bhattacharya, B.B.

[2] R.E. Bryant (1986) "Graph-based Algorithms for Boolean Function Manipulation", *Transaction on Computers*, Vol. C- 35, No. 8, Pp. 677–691.

[3] J. Mogul, R. Rashid & M. Accetta (1987), "The Packet Filter: An Efficient Mechanism for User-Level Network Code", *Proceedings of the Eleventh ACM Symposium on Operating Systems Principles*, Pp. 39–51.

[4] R.E. Bryant (1992), "Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams", *ACM Computing Surveys*, Vol. 24, No.3, Pp. 293–318.

[5] A. Henwood & Hazelhurst, A. Fatti (1998), "Binary Decision Diagram Representations of Firewall and Router Access Lists", jtp:/(ftp.cs. wits.ac.zalpublresearchlreportsITR-Wi.)

[6] T.V. Lakshman & D. Stidialis (1998), "High Speed Policy-based Packet Forwarding using Efficient Multi-Dimensional Range Matching", *Proceedings of SIGCOMM*, ACM Press, Pp.203–214.

[7] V. Srinivasan, S. Suri & G. Varghese (1999), "Packet Classification using Tuple Space Search", *Proceedings of SIGCOMM*, ACM Press, Pp. 135–146.

[8] P. Eronen & J. Zitting (2001), "An Expert System for Analyzing Firewall Rules", *Proceedings of the 6th Nordic Workshop on Secure IT Systems (NordSec 2001)*, Copenhagen, Denmark, Pp. 100–107.

[9] M. Christiansen & E. Fleury (2004), "An MTIDD based Firewall using Decision Diagrams for Packet Filtering", *Telecommunication Systems*, Vol. 27, No. 2–4, Pp .297–319.

[10] E. Al-Shaer & H. Hamed (2004), "Modeling and Management of Firewall Policies", *IEEE Transactions Network and Service Management*, Vol. 1, No. 1, Pp.2–10.

[11] Adel El-Atawy, Hazem Hamed & Ehab Al-Shaer (2005), "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering", *IEEE Infocom*, Pp.1–15.

[12] Errin W. Fulp & Stephen J. Tarsa (2005), "Trie-based Policy Representations for Network Firewalls", *Proceedings of the IEEE International Symposium on Computer Communications*, Pp. 434–441.

[13] S. Acharya, J. Wang, Z. Ge, T. Znati & A. Greenberg (2006), "Traffic Aware Firewall Optimization Strategies", *Proceedings of ICC*, Pp. 2225–2230.

[14] C. Shen, T. Chung, Y. Chang & Y. Chen (2007), "PFC: A New High Performance Packet Filter Architecture", *Journal of Internet Technology*, Vo1.8, No.1, Pp. 67–74.

[15] G. Paul, A. Pothnal, C.R. Mandal & Bhargab B .Bhattacharya (2011), "Design and Implementation of Packet Filter Firewall using Binary decision diagram", *Proceedings of Student Technology Symposium*, IIT, Kharagpur, Pp. 17–22.