# A Framework of a Internet Firewall for IPv6 using FPGA

Gouri Shankar Prajapati
Department of Computer Science and Engineering
MANIT, Bhopal

Nilay Khare
Department of Computer Science and Engineering
MANIT, Bhopal

## ABSTRACT

As the communication via internet is growing very fast, network security becoming the essential need of an organization or user. it include protecting data from unauthorized access, protecting data from damage and implementing policies and procedures for network security breaches and data losses. Due to exhaustion problem of IPv4 addresses we will soon switch over IPv6. To solve this problem we are presenting a Framework of a firewall for IPv6 and IPv4 networks using a field-programmable gate array (FPGA). The FPGA implements, the accept or deny rules of the firewall in Hardware using Verilog Hardware Description Language. A hardware based firewall offers the advantages of speed over a software firewall, in addition to direct interfacing with network devices, such as an Ethernet. This firewall would have the ability to process the data packets based on source and destination TCP/UDP port number, source and destination IPv4 and IPv6 address, and combination of source IP address, and destination port number. Incoming and outgoing IPv6 packets addresses first converted into IPv4 addresses for filtering decisions.

## Keywords

Firewall, IP, FPGA, Protocol, ROM, Packet

## 1. INTRODUCTION

A firewall is a device (usually a router or a computer) installed between the internal network of an organization and the rest of the Internet. It is designed to forward some packets and filter (not forward) others. The firewall can be hardware or software, and the protected computer can be a typical PC, network equipment, or embedded device. In all cases, the firewall controls communications to and from devices. It allows or blocks certain types of Internet Protocol (IP) packets based on information contained in the packet, and more specifically in the header of the packet. This information is usually contained in: IP source address, IP destination address, Protocol: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), source port number and destination port number [1]. A firewall can be used as a packet filter. It can forward or block packets based on the information in the network layer and transport layer headers: source and destination IP addresses, source and destination port addresses, and type of protocol (TCP or UDP). A packet-filter firewall is a router that uses a filtering table to decide which packets must be discarded (not forwarded).

**Table 1. Packet filter firewall rule list**

| Interface | Source IP | Source Port | Destination IP | Destination Port |
|---|---|---|---|---|
| 1 | 131.34.0.0 | * | * | * |
| 1 | * | * | * | 23 |
| 1 | * | * | 194.78.20.8 | * |
| 2 | * | 80 | * | * |

1. Incoming packets from network 131.34.0.0 are blocked (security precaution). Note that the * (asterisk) means "any."
2. Incoming packets destined for any internal TELNET server (port 23) are blocked.
3. Incoming packets destined for internal host 194.78.20.8 are blocked. The organization wants this host for internal use only.
4. Outgoing packets destined for an HTTP server (port 80) are blocked. The organization does not want employees to browse the Internet.

A software-based firewall is flexible in terms of rule definition and change, the application of the rule and the speed of the firewall are both a function of the speed of the CPU and the percent of the time devoted to the firewall function. These two disadvantages are not present in a hardware-based firewall, where the packet comparison is performed directly by hardware[1].

The Internet Protocol version 4, or IPv4, is the defined standard in the world today, but it is being replaced by the more advanced IPv6, to help solve the IP address exhaustion problem that is looming on the horizon. IPv4 uses 32 bits to define each address, which, in total, is roughly four billion addresses. This was a huge number during its inception, but with the internet boom, this address pool is expected to run-out in 2010 or 2011. IPv6 uses 128 bits for each address. Internet Protocol version 6 (IPv6) is a version of the Internet Protocol (IP) that is designed to replace Internet Protocol version 4(IPv4). The Internet operates by transferring data in small packets that are routed across networks as specified by an international communications protocol known as the Internet Protocol. Each data packet contains two numeric addresses that are the packet's origin/destination devices. Since 1981, IPv4 has been used version of the Internet Protocol, and it is currently the foundation for most Internet connections. The Internet's growth has created a need for more addresses than IPv4 can offer. IPv6 allows approximately 340 undecillion addresses, but switching from IPv4 to IPv6 may be a difficult process.

It may take some years to completely switch from IPV4 to IPV6 for the time being we can use the IPV4 to IPv6 convertor and IPV6 to IPV4 to run the communication successfully.

## 2. PRIOR WORK

Ayman Kayssi el at.[1] shows design of a firewall for IP networks using a field-programmable gate array (FPGA) The FPGA implements, the accept or deny rules of the firewall in hardware. They make use of static firewall rule list and reflexive firewall rule list to store the rules. a static list of rules that do not change from session to session.They make use of static firewall rule list and reflexive firewall rule list to store the rules. a static list of rules that do not change from session to session. Reflexive firewall rule lists allow IP packets to be filtered based on upper-layer session or connection information. They shows how the rules are translated to VHDL and then implemented in hardware, and how the hardware is utilized to filter network traffic in a packet-by-packet fashion, or based on connection information, with a speed of more than 500,000 packets per second.

Rajanish K. Kamat el at.[3] reported a novel design framework for creation of behavioral design. We have examined the opportunities brought about by finite state machines and to harness them into a synthesizable register transfer level (RTL) architecture. We discuss a case study of packet parser its finite state machine (FSM), data path controller architecture and issues related to its Handel-C implementation.

Arief Wicaksana el at.[4] present the architecture of fast and reconfigurable Packet Classification Engine (PCE).This engine is used in FPGA-based firewall. PCE inspects multi-dimensional field of packet header sequentially based on tree-based algorithm. Ethernet packet is examined with PCE based on Source IP Address, Destination IP Address, Source Port, Destination Port, and Protocol fields of the packet header. A rule of firewall is made from combination of some fields from packet header. Rules simplification is approached using an algorithm which uses tree as a basic form; we called it Tree-based algorithm. This algorithm simplifies overall system to a lower scale and leads to a more secure system.

Raouf Ajami el at. [5] describes the design of a highly customizable hardware packet filtering firewall to be embedded on a network gateway. This firewall has the ability to process the data packets based on source and destination TCP/UDP port number, source and destination IP address range, source MAC address and combination of source IP address, and destination port number. It is capable of accepting configuration changes in real time. A hardware/software co-design is implemented in which the main hardware blocks were built using Verilog Hardware Description Language (HDL). A processor based embedded system with real-time operating system has been designed to achieve highly customized and on the fly configuration change in the firewall. Content Addressable Memory (CAM) was used to improve speed of the packet matching. The whole design has been implemented and evaluated on an Altera FPGA device.

All the work which has been done so far is only for IPv4. No one has discussed to implement the IPv6 addresses to IPv4 addresses conversion and vice versa till now. And implement internet firewall for the coming version of IP i.e. IPv6 in FPGA.

## 3. FPGA BASED FIREWALL FRAMEWORK FOR IPV6

Here framework for hardware based firewall is given that filter the ipv4 and ipv6 packet on the basis of information contains in packet headers. A hardware based firewall offers the advantages of speed over a software firewall. Given framework will easily implemented using verilog hardware description language.

Proposed firewall filter the packets on the basis of IPv4 addresses (32 bit) and port numbers according to the reconfigurable filtering rules stored by the administrator in the rule base. Incoming IPv6 packets header information first converted in IPv4 compatible addresses and then given to the firewall for filtering decisions.

1. Firewall takes all incoming & outgoing packets either IPv4 or Ipv6 and stores into packet buffer.

2. Packet extractor takes packet one by one from the packet buffer and extract the filtering relevant information (IP address and port numbers etc.) and extracted information forwarded to the IP version checker.

3. IP version checker checks the version of incoming packets if 3.1: It is IPv4 packets then forwarded to the comparator.

3.2 It is IPv6 Packet then forward to the IPv6 to IPv4 convertor that converts IPv6 address into compatible IPv4 addresses and then forward to the comparator.

4. The validity of the source IP address and port number is then compared with the IP address and port number stored in the access deny rule list. If match is found then drop the packet form the packet buffer otherwise forward the packet.

### 3.1 Packet Buffer

It is a memory (ROM) which stores the whole packet comprising data part and header part.

### 3.2 IP header Extractor

It separates the header part from the data part. And extract the filtering information needed by the filtering unit from the header i.e. Protocol Version, Source IP address, Source Port number, Destination IP, Destination Port etc.

### 3.3 IP version checker

IP version checker checks the version of incoming IP whether it is IPv4 or IPv6 by counting the number of bits of the Source IP or Destination IP. If there are 128 bits in Source IP then it is IPv6 and if total bits are 32 then it is IPv4.

### 3.4 IPv6 to IPv4 conversion

IPv4 uses four 1 byte decimal numbers, separated by a dot (i.e. 256.256.256.256), while IPv6 uses hexadecimal numbers that are separated by colons. Due to the incompatibility of IPv4 and IPv6, translations have been made to enable their interoperation, which leads to addresses that look like fe80:0:0:0:0:0:C0A8:6301.IPv6 address divided into eight blocks separated by a colon. How IPv6 address is converted into IPv4, extract the last 32 bits from the IPv6 address (last two block from the address). i.e. C0A8:6301 first divide the address into group of two HEX character C0 A8 63 01. Take C0 and multiply the first character ‘C’ by 16 and the second character ’0 by 1. add the two decimal values together to get the IPv4 decimal value i.e ((C=12)*16) + (0*1) = 192 repeat the same process with A8, 63 and 01 we get 168 , 99,and 1.concatenate all to form IPv4 address 192.168.99.1 . IPv4 to IPv6 conversion it is a reverse process 192.168.99.1.Divide the first octet (192) by 16 ( Hex is a Base-16) i.e 192/16 = 12 times exactly with 0 left over- 12 in Hex is represented as C - 0 (zero) in Hex is, represented as , 0 Thus 192 in HEX is C0. Repeat the same with the second octet (168), third octet (99) and fourth octet then we will get corresponding Hex A8 63 01.now concatenate all with the colon C0A8:6301.

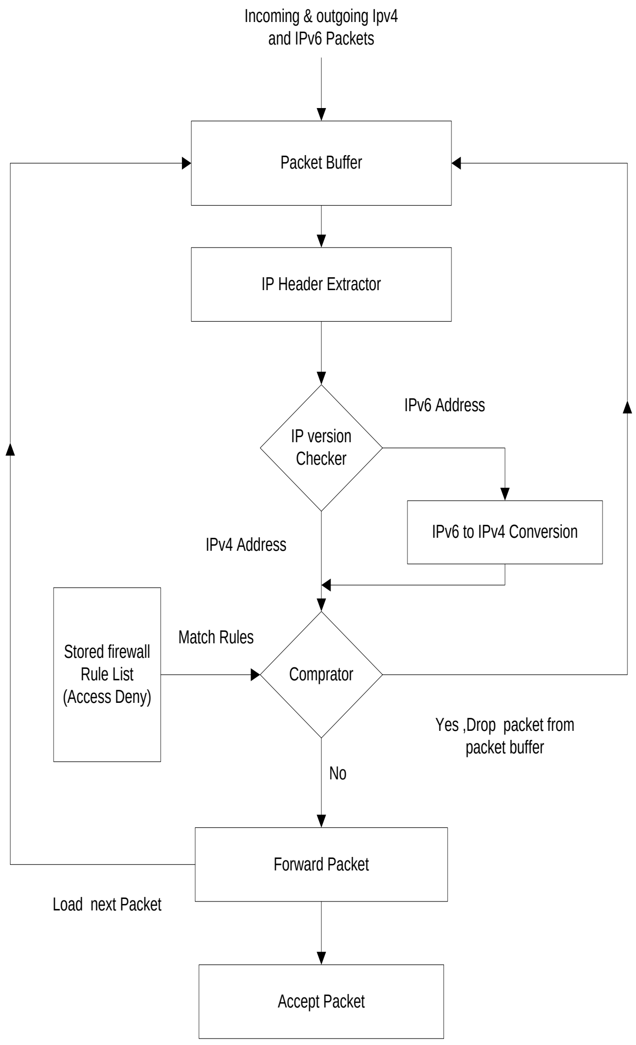

**Fig 1: Architecture of firewall**

### 3.5 Comparator

The network must be able to classify and filter this packet. Generally, network devices classify packets into two categories, permitted packets and blocked packets. Permitted packets are forwarded to the next step while blocked packets are removed from the line. There are two basic approaches for security policy, ‘default deny’ and ‘default allow’. ‘Default deny’ means packet is always blocked unless it is specified, and ‘default allow’ is the opposite. Comparator applied list of rules to all packets going through the firewall. Both inbound (from external network to internal network) and outbound (from internal network to external network) rules have the same basic building block. As an example, if we save the list of denied IP addresses in a memory buffer, and we compare each incoming packet source IP address to the stored ones, the packet is denied whenever there is a match.

### 3.6 Stored Firewall Rule List (Deny List)

It is a memory which is used to store the set of rules. Rule list is formed using Source IP, Destination IP, Source Port, Destination Port and Protocol. This memory can be ROM or CAM. For fast searching of the stored rule we can use CAM memory. Content Associative Memory (CAM) is a type of memory system for fast searching application in hardware. It acts differently from the other types of memory like RAM and ROM. Unlike other memory systems, CAM is fed with data instead of address. CAM searches the entire memory for data and if the data is available in memory (data already stored previously), the location of data is returned. the rule list is updated periodically.

### 3.7 Forward Packet

If rule match is found than comparator discard the packet ,otherwise incoming packet is accepted and send the control signal to packet buffer to load the next packet.

## 4. CONCLUSION

In this paper, we have proposed a frame work that can work with IPv6 as well as IPv4. Here we are mainly focusing on IPv6 to IPv4 conversion and vice versa. Due to exhaustion problem of IPv4 addresses we will soon switch over IPv6. At present 99% internet firewall works on IPv4. It will have some time to completely switch from IPv4 to IPv6. For this duration this framework will be helpful to solve this problem. Software firewalls are applications that run on the host system’s CPU, while hardware Firewalls consist of dedicated hardware. Software firewalls can drag down system performance under stressful network conditions, such as a denial of service attack, because the host system’s CPU is executing the filtering rules. Dedicated hardware firewalls designed to manage a large network are often expensive and meant to be located between a private network and the Internet. The above firewall model will be implemented in FPGA technology.

## 5. REFERENCES

[1] Ayman Kayssi, Louis Harik, Rony Ferzli, Mohammad Fawaz,” FPGA-BASED INTERNET PROTOCOL FIREWALL CHIP” 0-7803-6542-9/00 IEEE 2000

[2] Darrell Laturnas, Ron Bolton, “Dynamic Silicon Firewall”, CCECE/CCGEI , IEEE 2005

[3] Rajanish K. Kamat, Pawan K. Gaikwad, Santosh A. Shinde, “Implementation of FPGA based Firewall Using Behavioral Synthesis”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010

[4] Arief Wicaksana, Arif Sasongko, “Fast and Reconfigurable Packet Classification Engine in FPGA-Based Firewall”, International Conference on Electrical Engineering and Informatics, 978-1-4577-0752-0/11 IEEE 2011

[5] Raouf Ajami, Anh Dinh, “Embedded Network Firewall on FPGA”, Eighth International Conference on Information Technology: New Generations IEEE 2011