2011 Eighth International Conference on Information Technology: New Generations

Embedded Network Firewall on FPGA

Raouf Ajami, Anh Dinh
Department of Electrical and Computer Engineering
University of Saskatchewan, Canada

Abstract

This paper 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. An Altera FPGA platform has been used for implementing and evaluating the network firewall.

Keywords: ***network firewall, FPGA, embedded system***

1. Introduction

The estimated cost of recovery for companies and organizations due to malicious software attacks has increased in the order of more than one billion per year since 1995 [1]. From 2005 onward, the loss declined due to increasing and improving network security infrastructure [1]. Using Field Programmable Gate Array (FPGA) devices for network firewall has been reported in the past [3,4,5,6].

This high speed packet processing firewall can be configured in different modes and has the ability to accept and apply configuration changes in real-time along with speed improvement compared to [5] and [6]. The modes, which this firewall operates on, are based on header information in Link, Internet and Transport layers of the Transport Control Protocol/ Internet Protocol (TCP/IP) protocol stack. There are six modes of operation of the firewall: (1) TCP/UDP destination port number, (2) TCP/UDP source port number, (3) Source IP address, (4) Destination IP address, (5) Source Media Access Control (MAC) address, and (6) Combination of source IP address and destination port number.

This is a hardware/software co-design 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.

2. Hardware and software modules

Figure 1 shows the overall structure of the embedded network firewall design. As shown in the block diagram, the main modules in the embedded network firewall are the Nios II 32-bit microprocessor module, the Ethernet module, the Content Addressable Memory (CAM) module, the Netmask RAM module, the Arbiter module and the Network Firewall module (NFM). All of these modules tightly works together to achieve a powerful, flexible and easy to configure packet filtering firewall. The flow chart for the NFM module is shown in Figure 2. Table 1 lists the resource required to implement the design into an Altera FPGA device. Figure 3 shows all of the different software layers that are used in the embedded system network application. These modules handle initialization and configuration of the Nios II and the firewall to coordinate and run Telnet server software for changing the firewall configuration and monitoring the firewall.

Initially, when the system is powered up, the firewall module is in inactive mode as the Nios II has not been programmed to run any code or operating system (OS). Nios II Integrated Development Environment (IDE) development tool is used to upload essential applications including RTOS to the Nios II. The software initializes all of the modules according to the selected mode of operation. When the Ethernet module finds any TCP/IP traffic targeted toward the Nios II, it interrupts the NFM. The NFM extracts the necessary field (based on the initial operation mode) off the Ethernet frame and inquires the permission from the CAM module. If the CAM module finds the selected field in its memory block (match found), a pass permission is given to the NFM and the NFM interrupts the Nios II for a receiving packet. In the case of an invalid packet, the NFM drops the packet and waits for the next interrupt from the Ethernet. After each Ethernet interrupt, the packet status is reported to the Nios II by the NFM for monitoring purposes. The firewall operation mode and configuration can be changed any time on the fly by Telnetting to the Telnet server running on the Nios II.

978-0-7695-4367-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ITNG.2011.178

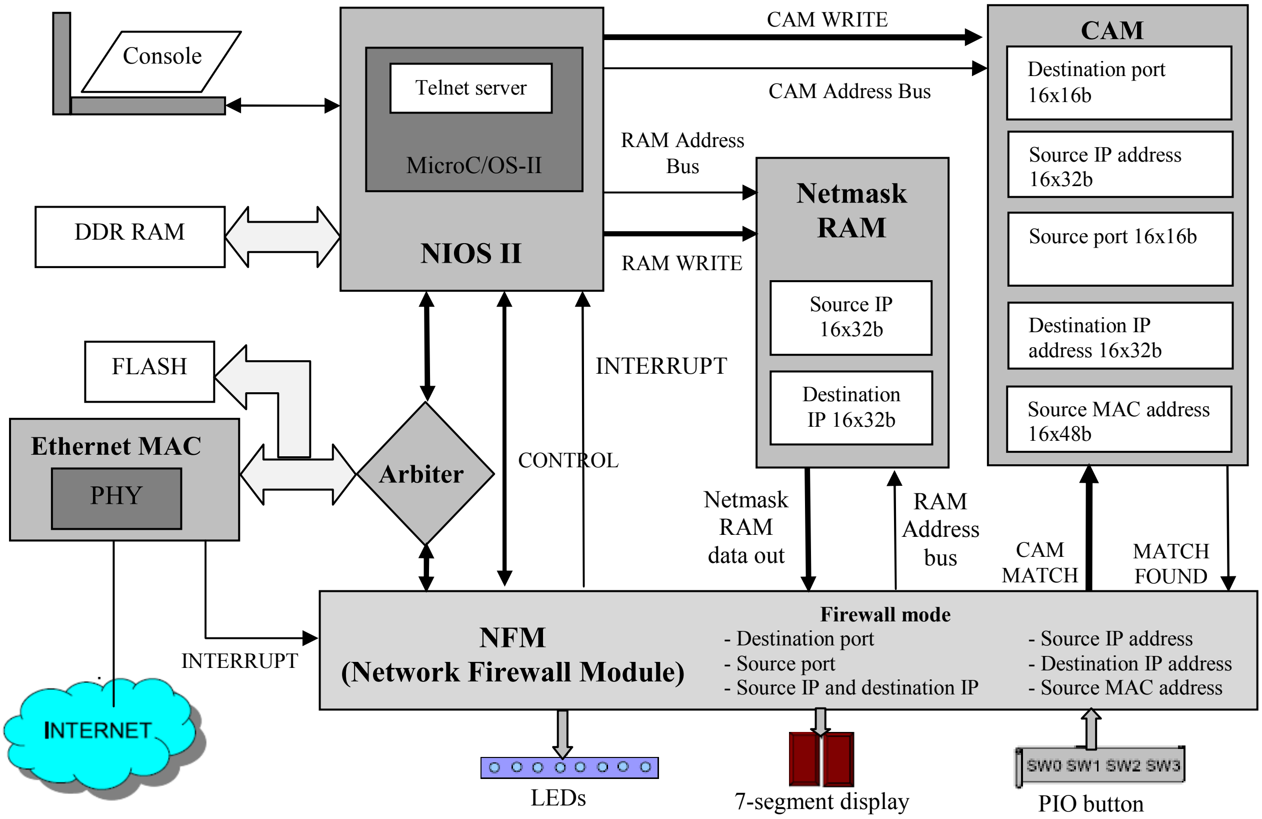

Figure 1. Embedded network firewall block diagram

3. Testing and results

Four very useful and popular software tools were used for this testing: *Wireshark, Tcpdump, Ping, and Network MAPper (Nmap)*. A small size network was setup including two host PCs, the Altera board and a network hub. At first, the FPGA is uploaded with the hardware design and then the application is compiled in the Nios IDE and loaded to the Nios II. Table 3 shows the timing results from different modes of operation. The processing time takes between 1.84µs and 4.2µs. This processing time is not causing any substantial delay for packet processing.

4. Conclusions

This design outperforms the other FPGA firewalls in terms of practical features such as real-time configuration, real-time status report, transport and physical layer packet/frame processing and IP address range capability. A comparison with the *iptables* (the Linux software firewall) also shows this design performs much faster and not sensitive to the packet size or the number of rules in the firewall configuration. The complete design uses a small portion of the Altera StratixII-2S60 FPGA, 11% of the logic blocks and 6% of the memory blocks. Testing speed indicates that the design can process from 2.9 to 6.6 Giga bits of Ethernet data per second. It means that by adding this firewall on a network gateway, the data transfer experiences no delay.

5. References

[1] S. Harris, A. Harper, C. Eagle, and J. Ness, *GRAY HAT HACKING, The Ethical Hacker's Handbook,* McGraw-Hill, 2nd Edition, 2008.

[2] E. D. Zwicky, S. Cooper, and D. B. Chapman, *Building Internet Firewalls*, O'Reilly Media, 2nd Edition, 2000.

[3] Jedhe, G.S.; Ramamoorthy, A.; Varghee, K., "*A Scalable High Throughput Firewall in FPGA*," The 16th International Symposium on Field-Programmable Custom Computing Machines, FCCM'08, Palo Alto, CA, USA, April 14-15, 2008, pp. 43-52.

[4] Kayssi, A.; Harik, L., Ferzli, R., Fawaz, M., "*FPGA-based Internet protocol firewall chip*," The 7th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2000, Jounieh, Lebanon, December 17-20, 2000, pp. 316-319, vol.1.

[5] J. Cheng, "Silicon firewall prototype," Master of Science Thesis, University of Saskatchewan, 2004.

[6] D. Laturnas, R. Bolton, "*Dynamic silicon firewall,*" Canadian Conference on Electrical and Computer Engineering, CCECE 2005, Saskatoon, Canada, May 1-4, 2005, pp. 304-307.

[7] *NIOS II Processor Reference Handbook*, Altera Corp.

Table 1. FPGA resources

Family/Device	Stratix II/EP2S60F672C5
Combinational ALUTs	4595/48352 (10%)
Dedicated logic registers	3169/48352 (7%)
Logic utilization	11%
Total block memory bits	164096/2544192 (6%)

Table 3. Number of clock cycles for invalid packet dropping

Firewall mode	**With monitoring** #clock/ns	**Without monitoring** #clock/ns
Destination port	118/2360	92/1840
Source port	118/2360	92/1840
Source IP address	122/2440	96/1920
Destination IP address	122/2440	96/1920
Source IP and Destination port	153/3060 to 169/3380	127/2540 to 143/2860
MAC address	122/2440	96/1920

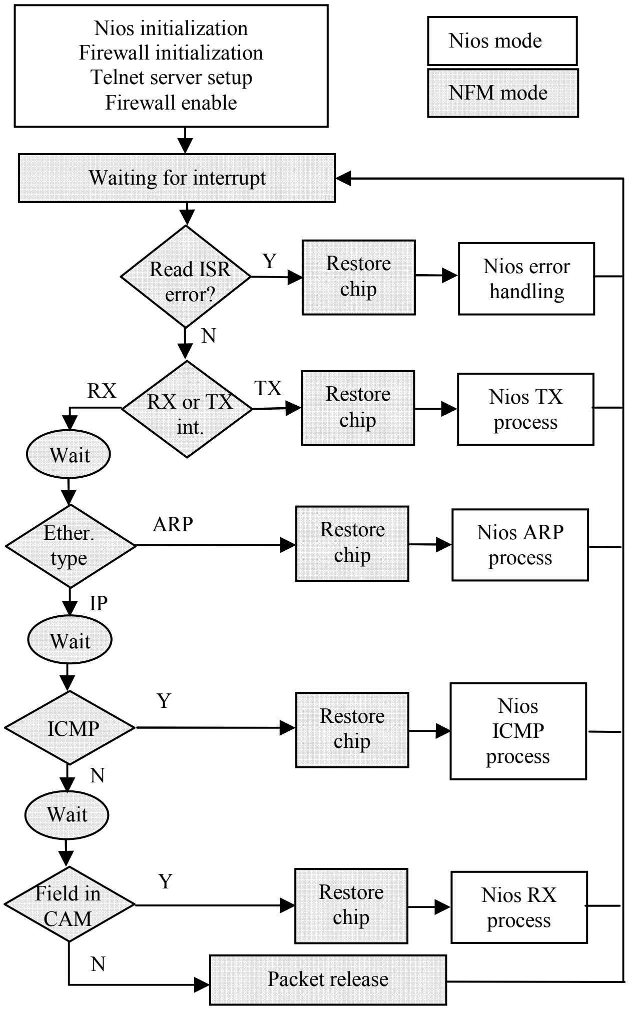

Figure 2. NFM process flowchart

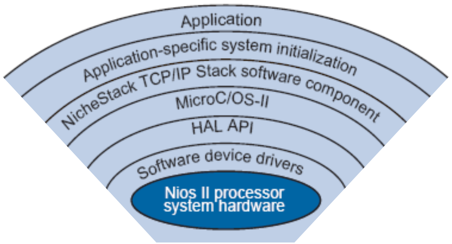
Application
Application-specific system initialization
NicheStack TCP/IP Stack software component
MicroC/OS-II
HAL API
Software device drivers
Nios II processor system hardware

Figure 3. Layered software model for NIOS II network application development [7]