Department of Electrical and Computer Engineering
Howard University

Design Project Proposal

EECE 401 Senior Design I

MEMORANDUM

October 31, 2007

TO: Dr. Charles Kim
Instructor

FROM: Laurence Wilson
Tolu Onibiyo
Marlon Winder
Idris Ozoya
Hassan Ayinde

SUBJECT: Design Project Proposal Submission

Enclosed is our group's design project proposal, Network Packet Inspector and Intrusion Detector. This proposal is submitted in partial fulfillment of the Senior Design requirement outlining the plan for the project pursuit through the problem formulation with functional requirement; alternative solution generation with electrical and computer engineering approaches; project management and milestones; and task assignments and deliverables. We understand this proposal, in written report as attached and oral presentation upon scheduled, would undergo a rigorous Proposal Review Panel assessment. Upon request, we are willing to accept recommendations from the Panel Review and modify and resubmit for final approval.

Design Project Proposal

Network Packet Inspection and Intrusion Detection

Submitted by

Laurence Wilson
Tolu Onibiyo
Marlon Winder
Idris Ozoya
Hassan Ayinde

Approved by

Proposal Review Panel Representative:

____

Name Signature Date

Senior Design I Instructor:

____

Name Signature Date

INTRODUCTION

Increase in network traffic volume and transmission speeds has given rise to the need for extremely fast packet processing. Many traditional processor-based network devices are no longer sufficient to handle tasks such as packet analysis and intrusion detection at multi-gigabit rates. Rapidly increasing network transmission speeds have marked the computationally heavy task of network packet inspection as an obvious bottleneck in the processing and forwarding of information across the network. This project proposes hardware architectures to relieve the computational load of a processor typically housed within network switches and routers. The need for this function cannot be over-emphasized.

As the speed of networks increases, the amount of time available for a network device, such as a router to process each packet decreases. For example, a small 64-byte packet may arrive every 51,200 ns for a 10 Mb/s network under near peak traffic loads. This provides sufficient time to respond to each packet. At 10 GB/s however there are only 51 ns available to respond to each packet. This presents a computational bottleneck to the network, especially when traditional packet monitoring is employed. This issue will be addressed by the proposed hardware architecture.

This project requires knowledge of network infrastructure as well as advanced hardware design. A strong foundation of description languages, using VHDL and Verilog provide a basis for this design and implementation. Classes previously studied here at Howard University such as Digital Systems; Advanced Digital System; ASIC Design; Micro Computer Design; and Introduction to Computer Networks, all play a key role in the problem formulation and problem solving.

Project Objectives

The primary objective of this project is to design a packet processor that will be able to process packets at a line speed of 100 Mbps. To accomplish this primary goal, the hardware area must fit on the Xilinx ML403-XC4VLX25 FPGA board and have a minimum clock frequency of 25 MHz to accommodate line speed of 100 Mbps. In order to produce this end product on time, the objective of meeting our deadlines and milestones are also a priority. This project's estimate completion time is the first week of March.

Engineering Approach

Significant research has been done on intrusion detection methodologies. Perhaps the most common approach is signature-based, which is centered on the assumption that intrusion attempts can be characterized by the comparison of user activities against a database of known attacks that lead to compromised system states. Signature-based intrusion detection is the basis for “Snort”, the software intrusion detection software solution. Snort is a free open source network intrusion detection and prevention system capable of performing packet logging and real-time traffic analysis on IP networks. Snort can perform protocol analysis, content searching and matching, and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, web application attacks, and OS fingerprinting attempts, amongst other features.

In order to identify potentially harmful packets, Snort must search its rule-set, or signature data base, to find any rules that match the packet under inspection. A signature is a known condition or pattern that indicates suspicious activity may be occurring. One example of a signature is port 8080. Many html based attacks falsely attempt to connect to port 8080. If a packet attempts to connect to port 8080, this may indicate suspicious activity.

As previously discussed, faster networks and heavy traffic loads make this software-based approach insufficient. Attempts have been made to implement faster pattern matching algorithms, however improvements have not been significant enough to handle gigabit network rates. This, to some extent, is a result of insufficient capacity by the processor. This proposed solution consists of a processor dedicated to packet inspection and pattern matching. The processor receives packets from a network backbone and classifies the packets into various protocols. Data is subsequently separated into each portion of the protocol. For pattern matching, depending on the keyword or item the user intends to search for, the header and payload of each classified packet is checked for any matches.

Tasks and Deliverables

Our design project focuses on the design and implementation of a software-driven processor implemented on a FPGA board. This entails developing a user interface as well as a circuit to process the packets.

The goals toward this project are to:

1. Hold weekly meetings and additional meetings as required.
2. Complete deliverables at least 2 business days before due date when ever possible.
3. Write code and run simulations for classifier by November 13th 2007.
4. Write code and run simulations for pattern matcher by December 6th 2007
5. Write code and run simulations for UART (Universal Asynchronous Receiver and Transmitter) driver by January 1st 2008.

6. Generate libraries for communication port to be incorporated into the User Interface module by January 31st 2008.
7. Integrate and test hardware and software components by February 28th 2008.
8. Evaluate and compare performance with existing technologies by March 5th 2008.
9. Finalize user guide and specification sheet by March 10th 2008.
10. Complete entire project by March 15th 2008.
11. Demonstrate final product on EE Day.

The final deliverables include:

1. Software program
   a. User interface
   b. Communication port link
2. FPGA board (Processor)
   a. Packet classifier
   b. Pattern Matcher
   c. UART driver

The final product will also include a detailed user's guide describing how to use the entire system. This guide will provide step-by-step instructions on how to load and remove signatures as well as how to view statistical information about signatures that have been matched. This setup will serve as a powerful tool to enable the user to gather information about data traversing the network. A specifications sheet will also be delivered providing a detailed description of the design including schematics of the hardware components. This product will be delivered by March satisfying all the design requirements. It will be tested and evaluated based on the 'Snort' software implementation to ensure improved performance.

A demonstration of the capabilities of this product will be provided. A member of the audience will be allowed to interactively operate the processor through the user interface that we developed. A member of the group will guide the participant through the user interface, demonstrating to the audience the following features:

1. Packet Classification:
    a. The user may enter specific protocols of interest. Once entered, the user may view counters associated with the protocols indicating how many packets match the protocol.
2. Signature Database Manipulation:
    a. Signatures may be loaded through the user interface. These signatures are compared to the parsed Ethernet packets for matches. A counter may be viewed for each signature by the user indicating the number of matches found. An audible signal is also provided to the user indicating that a match has been found. The system will include enough resources for at least 32 signatures.
3. Resource Statistics
    a. The use of and availability of system resource may be viewed through the user interface indicating how many signatures are loaded as well as the number of available signatures. The system will be developed to support a finite number of signatures.

This interactive demonstration will illustrate the use of the processor and the key features and capabilities. This final system and demonstration will verify that the proposed design has been successfully implemented according to specifications.

Project Management

In order to successfully implement this design, a timeline was established detailing the tasks of each member of the group. Also a budget is provided indicating the resources and facilities needed to complete this project.

Delivery Date	Milestone	Primary Team Member
14-Nov	Complete Classifier	Marlon Winder
6-Dec	Complete Pattern Matcher	Hassan Ayinde
31-Dec	Complete UART Driver	Laurence Wilson
15-Jan	Complete User Interface	Tolu Onibiyo
1-Feb	Complete Communication Port Link	Idris Ozoya
15-Feb	Complete User's Guide	The Team
1-Mar	Complete Specification Document	The Team
1-Apr	Deliver Final Product	The Team

Conclusion

We are on the verge of next age communications which require higher bandwidth of information to be processed in a shorter amount of time. IPV6, the next generation Internet protocol, defines a much larger packet size and supports transmission speeds in excess of 40 GB/s. Therefore preparations must be made in advance to deal with challenges that may arise in the future. The proposed design attempts to lay the foundation for offloading network intrusion detection from the CPU to a dedicated processor. This will free up computer resources allowing maximum CPU utilization.

Software packet inspection algorithms are consistently the bottleneck of typical network intrusion detection systems. This project presents a solution to the problem of the increasing need for faster packet processing and is needed if the desired level of network security is to be maintained at those speeds of future networks. This deep packet inspector project requires a significant amount of time and resources, but it will save time in the long run while keeping networks safe. This hardware architecture classifies every protocol field of an incoming packet as the packet is captured. Each classified protocol field has the ability to be analyzed and compared against signatures providing matches, thus enabling search at line speed of at least 100 Mbps. This particular approach is very adaptable and flexible. Because of our approach, it is simple to include additional resources requested as a result of the panel review.

Attachment

Synergy Overload	REQUIREMENT LIST FOR CONTENT PROCESSOR	**2nd Version: 10/23/2007**
Marlon Winder Hassan Ayinde Tolu Onibiyo Laurence Wilson Idris Ozoya		
DATE UPDATE	**REQUIREMENTS**	**SOURCES**
	Overall Function: *A microprocessor that is capable of deep packet inspection at wire speed*	
	SIZE	
10/01/2007	• Processor would be implemented on a single Virtex4 XC4VLX25 FPGA board.	The Team
	PERFORMANCE	
	Hardware:	
10/17/2007	• Processor must be dynamically configurable.	Marlon
10/17/2007	• Ability of the processor to search for patterns contained in the payload portion of IEEE 802.3 Ethernet frames.	Marlon, IEEE Standard
10/17/2007	• Support of relational operations and expressions in accordance with the Portable Operating System Interface (POSIX) expression lexicon.	Marlon
10/17/2007	• Ability to search for “don’t care” bytes within frames.	Laurence
10/20/2007	• Provides signals at the top/UI level to indicate expression match, prescribed action to be taken, and frame location where match was found.	Tolu
10/23/2007	• Processor must have the ability to search for matches given data input at a rate in excess of 100 Mbps.	Hassan
10/17/2007	• Ability to provide audible notification when match found.	Tolu
	Software:	
10/20/2007	• The UI must allow configuration and manipulation of search patterns.	Hassan
10/20/2007	• Ability to receive signals from the hardware device indicating an expression match, prescribed action to be taken, and frame location where match was found.	Laurence, Tolu &

10/20/2007	• Provide statistics regarding match count.	Marlon Idris
10/19/2007	**SAFETY** • Device must not generate heat in the excess of 40^{o}C from 6 inches away	The Team
10/20/2007 10/23/2007	**COST AND SCHEDULE** • FPGA board must cost less than $1000 • Content Process design must be completed and ready for testing by 3/1/2008	Team
10/23/2007 10/01/2007	**INTERFACES** • FPGA board must interface with user via EIA232 RS232 serial communication port. • FPGA board must interface with internet via standard Ethernet Port.	Marlon Hassan Tolu IEEE Standard
	COMPLIANT STANDARDS • IEEE 802.3 Ethernet • EIA232 RS232 Serial Port • IEEE 1003 POSIX expression lexicon.	IEEE Standard