Final Report

August 2005 – August 2008

Department of Energy Early Career Principal Investigator Award (#ER25691)
titled

High Performance Networking Hardware Design

Gokhan Memik
Department of Electrical Engineering and Computer Science
Northwestern University

1. Research Summary and Activities

Overall, the goal in this project is to develop and investigate novel methods to increase the performance of networking hardware. A particular method that is proposed is trading-off correctness for increased performance. In other words, the project investigates how the performance of networking hardware can be improved if it is known that the network can tolerate a larger number of errors. The research in this project is divided into four major tasks: understanding/quantifying application errors, development and analysis of fault throttling techniques (i.e., tools for error vs. performance trade-off), the engineering of hardware that is built with such trade-off mechanisms (i.e., clumsy packet processors), and understanding the effects performance/reliability trade-offs on users.

In addition to this particular approach, the project included the development of other novel optimizations techniques to improve performance of networking hardware. First, we have worked on the design of customized architectures for various purposes. For example, we have developed hardware modules for feature extraction. The module we have developed can summarize network traffic information at speeds exceeding 30 Gbps. We have also developed a hardware anomaly-detection system utilizing the Principal Component Analysis. In addition, we have developed an all-digital software-defined radio. Second, we have developed techniques to improve the overall system performance by optimizing the network processing tasks on the host. Finally, we have worked on techniques to automatically configure networking applications to improve the efficiency of routing. Our detailed findings in these topics are described in Section 3.

2. Outcomes

This project resulted in 1 PhD student graduated and supported two more (who are expected to graduate in the following year). The following papers related to the project have appeared in conferences and journals:

- "Learning and Leveraging the Relationship between Architecture-Level Measurements and Individual User Satisfaction", A. Shye, B. Ozisikyilmaz, A. Mallik, G. Memik, P. Dinda, R. Dick, A. Choudhary, in Proc. of *International Symposium on Computer Architecture (ISCA)*, Beijing, China, June 2008
- "Energy Detection using Estimated Noise Variance for Spectrum Sensing in Cognitive Radio Networks", Z. Ye, G. Memik, J. Grosspietsch, in Proc. of *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, NV, March/April 2008
- "An Efficient FPGA Implementation of Principle Component Analysis based Network Intrusion Detection System", A. Das, S. Misra, J. Zambreno, G. Memik, A. Choudhary, in Proc. of *Design, Automation and Test in Europe (DATE)*, Munich, Germany, Mar. 2008
- "An FPGA-based Network Intrusion Detection Architecture", David Nguyen, Abhishek Das, Joseph Zambreno, Gokhan Memik, Alok Choudhary, *IEEE Transactions on Information Forensics and Security (TIFS)*, Volume 3, Issue 1, Mar. 2008
- "Automated Task Distribution in Multicore Network Processors using Statistical Analysis", A. Mallik, Y. Zhang, G. Memik, in Proc. of *ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, Orlando, FL, Dec. 2007
- "Design and Implementation of an FPGA Architecture for High-Speed Network Feature Extraction", S. Pati, R. Narayanan, G. Memik, A. Choudhary, J. Zambreno, in Proc. of

International Conference on Field-Programmable Technology (FPT), Kitakyushu, Japan, Dec. 2007

- “Spectrum Sensing Using Cyclostationary Spectrum Density for Cognitive Radios”, Z. Ye, J. Grosspietsch, G. Memik, in Proc. of *26th IEEE Workshop on Signal Processing Systems (SiPS)*, Shanghai, China, Oct. 2007
- “Reversible Sketches: Enabling Monitoring and Analysis over High-speed Data Streams”, R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, E. Parasons, Y. Zhang, P. Dinda, M. Kao, G. Memik, accepted by *IEEE/ACM Transactions on Networking*, Volume 15, no. 5, October 2007
- “Digital Modulation Classification Using Temporal Waveform Features for Cognitive Radios”, Z. Ye, G. Memik, J. Grosspietsch, in Proc. of *18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Athens, Greece, Sep. 2007
- “An FPGA Based All Digital Transmitter with Radio Frequency Output for Software Defined Radio”, Z. Ye, G. Memik, J. Grosspietsch, in Proc. of *Design, Automation, and Test in Europe (DATE)*, Nice, France, April 2007
- “User-Driven Frequency Scaling”, Arindam Mallik, Bin Lin, Gokhan Memik, Peter Dinda, Robert Dick, *IEEE Computer Architecture Letters*, Volume 5, no. 2, 2006
- “The User In Experimental Computer Systems Research”, P. Dinda, G. Memik, R. Dick, B. Lin, A. Mallik, A. Gupta, and S. Rossoff, to appear at *Workshop on Experimental Computer Science (Part of FCRC)*, San Diego, CA, June 2007
- “A Reconfigurable Architecture for Network Intrusion Detection Using Principal Component Analysis”, D. Nguyen, A. Das, G. Memik, A. Choudhary, In Proc. of *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, California, April 2006
- “Reverse Hashing for High-speed Network Monitoring: Algorithms, Evaluation, and Applications”, R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, E. Parsons, Y. Zhang, P. Dinda, M. Kao, G. Memik, in Proc. of *25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, Apr. 2006
- “A Reconfigurable Architecture for Network Intrusion Detection using Principal Component Analysis”, D. Nguyen, G. Memik, A. Choudhary, in Proc. of *Fourteenth ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, Monterey, CA, Feb. 2006

3. Details of Discoveries

The details of the discoveries we have made for the topics described in Section 1 are listed in the following sections.

3.1 Dedicated Networking Hardware Designs

Our work in this topic focused on two areas: design of networking hardware for security and design of a digital software-defined radio architecture.

3.1.1 Design of a Hardware-Based Security System

The first step in the anomaly detection was to design a feature extraction module. The goal of feature extraction is to create associations between a series of events or data (e.g. streaming image or video, or network packets) and properties or behavior that is representative in that data set. Specifically, each data element possesses a multi-dimensional set of properties and features can be associated with different subsets and combinations of these properties. Our goal is to design fast and efficient hardware architectures that accomplish this task. We utilized the idea of sketches as a basis. Sketches are data structures, which are used in data stream modeling for summarizing large amounts of information requiring a small, constant amount of memory. They are a probabilistic summary technique for analyzing large streams without keeping per-flow state that make vector projections onto other sketches to infer additional information. Three basic components form our FEM architecture: Feature Controller (FC), Feature Sketch (FS), and Data Aggregate (DA). A generic view of this architecture is shown in Figure 1. The FEM architecture uses multiple hash functions in parallel to achieve high accuracy. This parallel access is particularly suitable for acceleration using hardware implementation. The feature controller (FC) coordinates the inputs to the hash functions, which are read from the atoms. For instance, a feature sketch monitoring traffic data in the highway network of a city could use number of cars passing through control points and their direction . The feature sketch (FS) is an application of sketches used for data stream modeling. Essentially, it is a memory block accessed by passing the input properties through different hash functions. An FS contains H rows each of length K. Finally, the data aggregate (DA) component takes H values and estimates the actual value for a query. There are two main functions supported by the FEM: UPDATE (k, v) to change the value in the sketch and ESTIMATE (k) to correctly retrieve a value from the sketch. Using statistical estimation techniques, we can show that ESTIMATE queries to the FS are accurate. Particularly, our preliminary implementation on a implemented using a Xilinx Virtex-II Pro chip reveal that the FEM architecture with H = 4 and K = 4096 achieves approximately 98% accuracy and 20.8 Gbps throughput for a 6-dimensional input set derived from network packet traces. In addition, the throughput can be increased to approximately 30 Gbps for H = 8 and K = 1024, which achieves 91% accuracy. In general, by simply modifying the H and K values, the trade-off between accuracy and speed can be controlled.

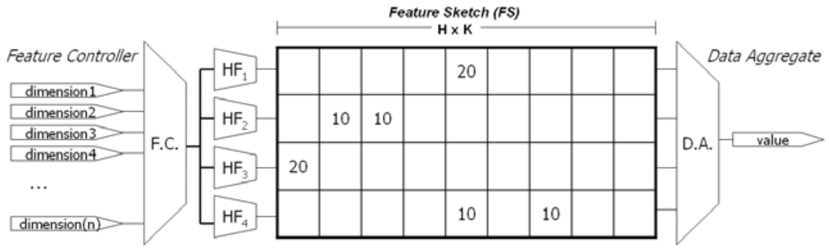

Figure 1. Feature Extraction Module

The feature extraction is followed by preprocessing, where features identified by the first module will be examined in closer detail and some filtering will be performed. This stage transforms the data into a more appropriate format for subsequent analysis. In addition, it increases the efficiency of the analysis stage. For example, by pruning out dimensions that are not important, a principal component analysis (PCA) technique can reduce the occurrence of false alarms, i.e., cases where the system reports change when there is none or vice versa. Some of the key components in this area are aggregation, dimensionality reduction, feature selection, and variable transformation.

Principal Components Analysis (PCA) is a widely used data analysis technique that is useful for reducing the dimensionality of the data and generating a new set of uncorrelated variables that are linear combinations of the original variables. PCA is based on the eigenvalue decomposition of the data covariance matrix and is of interest because typically most of the variation in the data can be captured by projecting the data onto the space defined by the relatively small set of eigenvectors with the largest eigenvalues. This allows the use of techniques that cannot be used for high dimensional data, and also reduces data set size and computational requirements considerably. As an additional benefit, PCA often reduces the amount of noise in the data. We propose to develop a hardware module to perform fast PCA. The preliminary design is depicted in Figure 2. This module reads p-dimensional atomic of feature information as well as the eigenvalues and the eigenvectors that are generated offline. Then, it performs the "outlier detection" analysis (parC in Figure 2), where for each input, a principal component score (PCS) is calculated. PCS aims to capture the distance of a point from the averages of the input set. Then, if the PCS is above a threshold, a change alarm is raised. We have implemented this preliminary design using Xilinx XC2VP100 device from the Virtex-II Pro family. For a 28-dimensional data (each field is 32-bits) and a major component selection of 6, our design was clocked at 75.83 MHz, achieving a throughput of 19.41 Gbps.

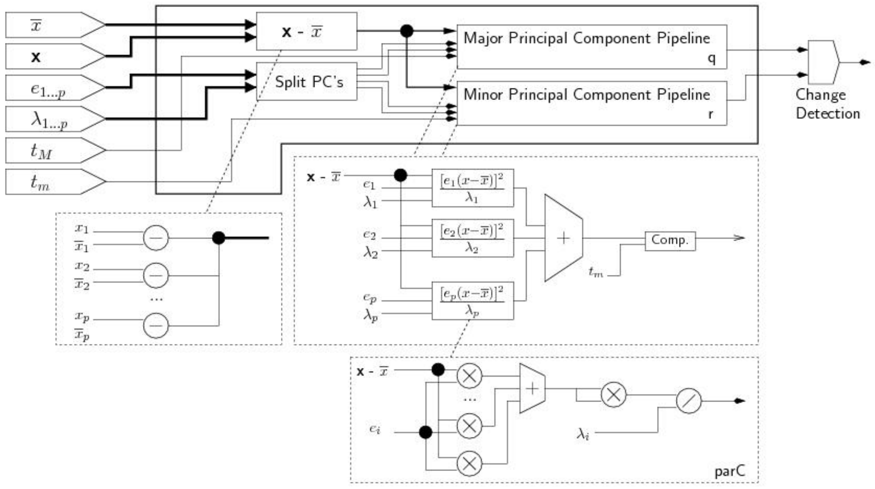
x - x̄
Split PC's
Major Principal Component Pipeline
q
Minor Principal Component Pipeline
r
Change Detection
Comp.
parC

Figure 2. Overview of the PCA implementation

3.1.2 Software-Defined Radio

We have worked on hardware implementation of a software-defined radio. Specifically, we have developed an all-digital software-defined radio architecture. Although there are several SDR implementations available, they still require significant amount of analog components. We tackled this problem and showed the feasibility of an all-digital transmitter. A high-level overview of this transmitter is illustrated in Figure 3. Almost all the transmitter's functionalities can be incorporated in the digital signal processing engine using Field-Programmable Gate Arrays (FPGAs) except the RF power amplification and simple filtering. As technology advances, for an ideal SDR, the digitization might be at, or very close to the antenna, such that almost all the radio communication functionalities can be realized using software based on high speed and reprogrammable digital signal processing engine. The advantages of all-digital transmitters are: potential high efficiency power amplification, the capabilities of digitally combining signals from multiple channels, and software programmability.

The generation of digital RF signals has drawn a lot of interest among researchers and engineers. However, prior to our work, only simulation results or non real-time test results have been presented in literature. In these works, the digital RF signals were computed offline and stored in pattern generator for the purpose of measurement. On the other hand, our work has presented the architecture and implementation of a real-time system that demonstrates the feasibility of digital generation of RF signals.

The entire all-digital transmitter architecture is shown in Figure 4. For brevity, only the functional blocks for inphase path are shown in detail. For the implementation, we have chosen Xilinx's Virtex2pro family: XC2VPX20-FF896, speed grade -7. On chip multi-gigabit transceiver (MGT) is used as the high speed parallel to serial converter. In the 20 MHz passband, the measured ACLR is 45 dB, which can meet the requirement for WCDMA. The noise shaping effects can clearly be seen from the spectrum plot where the noise rises outside the RF signal bandwidth. The EVM is measured to be less than 1%.

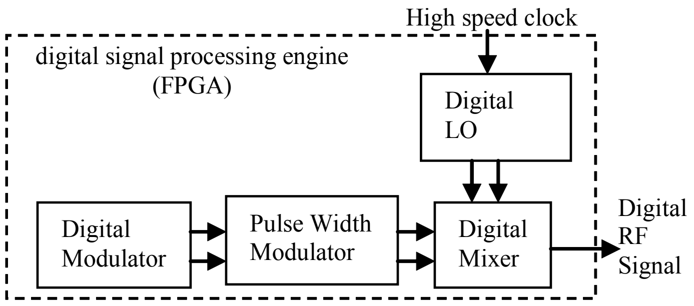

Figure 3. All-digital transmitter architecture

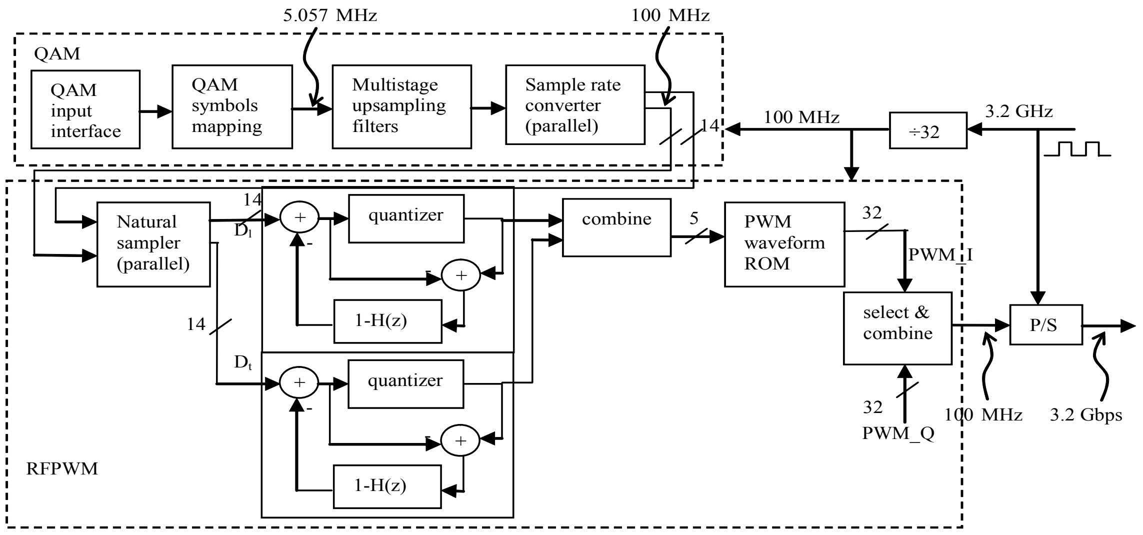

Figure 4. All-digital transmitter architecture implemented on FPGA

3.2 *Correctness Trade-Offs for Improving Performance*

Our work in this topic focused on two aspects. First, we studied fault throttling techniques and developed models for them. Second, we analyzed clumsy packet processors.

3.2.1 Fault Throttling Techniques and Models

The first stage is to investigate techniques that are available to a designer for controlling the fault behavior of a processor or processor component. There is already a set of techniques in the literature that varies the "reliability" of a component. For example, reducing the supply voltage (i.e., voltage scaling) of a component increases its delay and thereby increases the probability that the timing constraint will not be met (i.e., a hardware fault will occur). In the future, we will analyze these tools and investigate their applicability in clumsy execution. For example, in our previous work, we have analyzed the applicability of overclocking for clumsy execution.

3.2.2 Clumsy Packet Processors

The main approach in clumsy packet processors is to push the architecture properties (i.e., fault throttling) to a point where the processor components start making mistakes. Doing this increases the performance and/or reduces the energy consumption according to the particular fault throttling technique. In network processors, many applications are known to be immune to errors in higher levels. However, we still need to investigate clumsy packet processors for different applications and be able to compare different configurations. Therefore, we worked on a) quantify the importance of an application error, b) define metrics for comparing two outputs, and c) measure the change in the output of an application when the fault behavior of a component is changed.

3.3 Understanding User Behavior under Changing Performance Metrics

Once we have established the feasibility of clumsy processors, we have started investigating how users would react to the changes that result from clumsy execution. Particularly, we have worked on the analysis of user tolerance towards performance trade-offs. We have concentrated on developing user-feedback methodologies that can be implemented in different operating systems. The main idea in this work is named User-Driven Frequency Scaling (UDFS), which uses direct user feedback to drive an online control algorithm that determines the performance of a target system. Particularly, our goal in this work is to develop a tool that would interact with the user and change the target system performance according to the feedback it receives. We have first targeted a processor that is used in a laptop. However, our findings can easily be applied to a number of different processing elements. The main problem in this approach is to find a good operating condition.

Our goal is to change the performance visible to the user. To achieve this, we have manipulated the Dynamic Voltage and Frequency Scaling (DVFS) techniques. DVFS is one of the most commonly used power reduction techniques in high-performance processors. DVFS varies the frequency and voltage of a microprocessor in real-time according to processing needs. Although there are different versions of DVFS, at its core DVFS adapts power consumption and performance to the current workload of the CPU. Specifically, existing DVFS techniques in high-performance processors select an operating point (CPU frequency and voltage) based on the utilization of the processor. This approach integrates OS-level control, but such control is pessimistic.

User-Driven Frequency Scaling (UDFS) uses direct user feedback to drive an online control algorithm that determines the processor frequency. Processor frequency has strong effects on power consumption and temperature, both directly and also indirectly through the need for higher voltages at higher frequencies. The choice of frequency is directly visible to the end-user as it determines the performance he sees. There is considerable variation among users with respect to the satisfactory performance level for a given workload mix. We exploit this variation to customize frequency control policies dynamically to the user. In UDFS, the user presses a button when discomforted by the performance of the machine. These input events drive the UDFS algorithm that sets processor frequency. Our approach employs direct feedback from the user during ordinary use of the machine. We implemented UDFS as Microsoft Windows client software that appears as a taskbar task. The F11 key serves as the user discomfort button. We developed two algorithms to control the frequency based on these events. The first one (UDFS1) is loosely related to TCP congestion algorithm, the frequency of the processor corresponding to bandwidth and the user input corresponding to packet losses. The second algorithm (UDFS2), on the other hand, assumes that the pressing of a button means that the user wants the processor remain at the level and adjusts itself to remain at that level for a longer time.

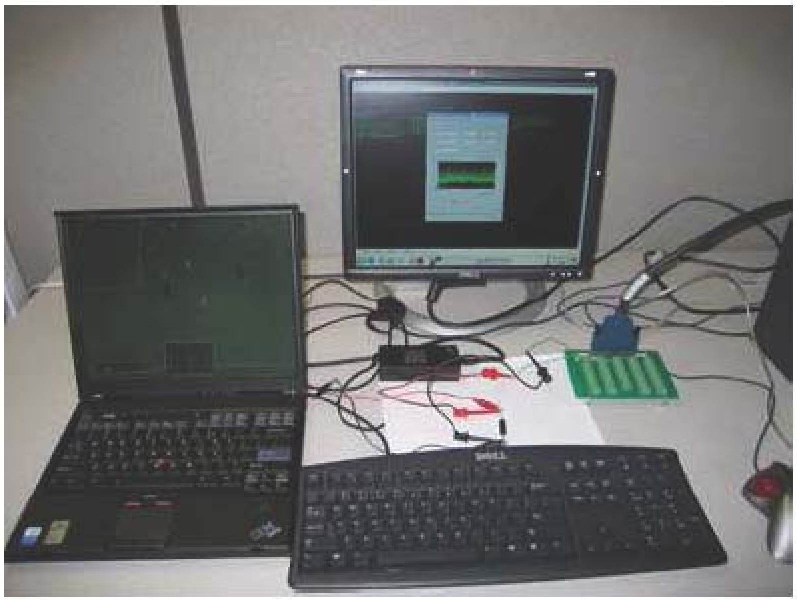

Figure 5. Power Measurement Framework

The UDFS algorithm dramatically reduced typical operating frequencies and voltages while maintaining performance at a satisfactory level for each user. The techniques were evaluated through user studies conducted on a Pentium M laptop running Windows applications. The studies include both single task and multitasking scenarios. The overall system power and temperature reduction achieved by our methods were measured in real time and the framework is pictured in Figure 5. UDFS reduces the overall power consumption by 22.1%, averaged across all users and applications, compared to the Windows XP DVFS scheme. The average temperature of the CPU is decreased by 13.2◦C. Using user trace-driven simulation to evaluate the CPU only, average CPU dynamic power savings was 24.9% UDFS, with a maximum of reduction 83.4%. These results suggest that there is strong variation in user performance expectation.

3.4 Mitigating System Level Bottlenecks - Investigating the Effects of Packet Losses

Finally, we worked on analyzing system level bottlenecks. Packet losses constitute an important problem for communication networks. For unreliable protocols such as UDP, they can degrade the performance of the applications. Some network transport protocols such as TCP provide reliable delivery of packets. In the event of a packet loss, the receiver asks for retransmission or the sender automatically resends any packets that have not been acknowledged. Although TCP can recover from a packet loss, retransmitting missing packets causes more interrupts raised at the host. Since our proposed architecture increases the rate of packet losses, we have to first analyze the effects of packet losses on the end systems. To achieve this, we have conducted a series of experiments. Figure 6 shows the simulation results for Maerts, a microbenchmark of Netperf. The figure presents the average number of interrupts per received packet. The results clearly reveal that even a small change in packet loss rate can cause a significant increase in the average interrupts on the end node: if the packet loss rate is increased to 1%, the average number of interrupts for each received packet goes up by 97.5%. In addition, if the packet loss rate reaches 5%, the increase in the number of interrupts reaches 275.3%. To be able to understand the impact of execution time on the CPU, we have analyzed the distribution of execution time during our simulations described above. Specifically, we measure the total time spent for handling the interrupts. As shown in Figure 7, the processing time of interrupts can increase by more than 62.3% as the packet loss rate increases from 0% to 1% and by more than 180% when the loss rate increases to 5%. In any networked application, interrupt handling constitutes a large fraction of the overall execution time. Hence, this change in the interrupt handling time has an important impact on the overall execution time of the applications.

There are several reasons for such a drastic change in the number of interrupts. The most important one is the inefficiently handling of the packets on the end nodes. Therefore, we have developed an approach that can be readily applied at Network Interface Cards (NICs) to mitigate the effects of packet losses. Our Delayed Processing approach employs a buffer in the NIC to delay the packets transmitted within the Round-Trip Time (RTT) when a packet is lost. Our goal is to minimize interrupts when a packet loss occurs. We achieve this by masking the packet loss information from the kernel whenever possible. In other words, the kernel will not be aware that the arriving packet is out-of-sequence. As a result, no interrupt will be raised when there is a packet missing, hence no special processing will be performed due to the lost packet. By delaying the processing of received packets, the execution on the host will not be interrupted; hence the time spent on interrupt handling is minimized.

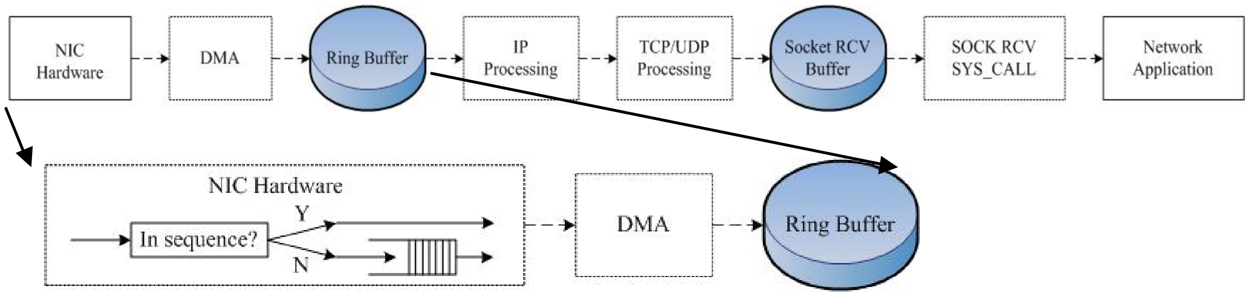

Figure 8. Change in the packet handling process on the end-nodes

Figure 8 shows our modification on the NIC of the receive host. When an out-of-sequence packet arrives, we do not put it into the ring buffer but into a delay buffer, which is a First in First out (FIFO) buffer in the NIC of the receiver side. In this figure, we do not provide the whole view of the process, since the following steps after the ring buffer remain the same as the conventional implementation.

The results show that using our proposed scheme, the number of interrupts can be reduced by up to 40% when packet losses take place. In addition, we also observe that, generally a very small buffer is necessary to achieve the maximum benefits. For example, on a 200 ms latency link, a buffer size of 11 MTU is large enough to achieve the maximum benefits. We have observed similar results after varying the link latency. In other words, the latency does not have a significant impact on the overall buffer requirement. In conclusion, the results suggest that the delayed packet processing can ameliorate interrupt overhead of TCP/IP workloads remarkably.

4. Personnel

The project throughout its span supported three graduate students: Arindam Mallik (graduated), Yu Zhang, and Pan Yan. Arindam Mallik participated in this project since its inception in August 2005. He received his PhD degree in June 2008. His work included the fault tolerance analysis for clumsy processors and code optimizations for networking applications. Yu Zhang received her BS degree from the Electrical Engineering Department of Tsinghua University, which is considered to be the top engineering school in China. Before joining our department in the Spring 2006, she has been a member of the Laboratory on Optical Networks at the Tsinghua University. Yu has worked on the system level studies. Pan Yan joined the project in the Fall 2007 quarter. Pan Yan has a B.S. degree from Shanghai Jiatong University and an M.S. degree from National University of Singapore. He continues to work on understanding the effects of reliability and performance (i.e., effects of clumsy execution) on user satisfaction.

5. DOE Collaborators

The PI collaborated with Dr. Wenji Wu at Fermi National Accelerator Laboratory. The collaboration was in the topic of system optimizations for high-performance networking.