Design and Analysis of a Dynamically Reconfigurable Network Processor

I.A. Troxel, A.D. George, and S. Oral
HCS Research Lab, ECE Department, University of Florida, Gainesville, FL
{troxel,george,oral}@hcs.ufl.edu

Abstract

The combination of high-performance processing power and flexibility found in network processors (NPs) has made them a good solution for today's packet processing needs. Similarly, the emerging technology of reconfigurable computing (RC) has made advances in packet processing as well as other point-solution markets. Current NP designs offer configurable elements but generally do not use dynamic RC techniques for run-time reconfiguration. Incorporating RC into NP designs to enhance packet processing is a natural progression for both of these emerging technologies. This paper presents the simulation results of a novel design for a RC-enhanced NP based on the Intel IXP1200 NIC design philosophy. The enhanced NP's performance is compared to that of the baseline NP in terms of three normalized traffic patterns and a case-study traffic pattern based on a military application. The results demonstrate that the enhanced NP significantly outperforms the baseline NP design in terms of latency for prioritized traffic that is non-uniform.

1. Introduction

Advances in chip technology and wire speeds have driven the need for faster packet-processing devices. Today's network devices are required to do more complex operations on an increasing number of packets in a more flexible manner for a lower cost in a shorter amount of time than any previously. To further complicate the issue, the user's appetite for additional bandwidth appears insatiable -- history has shown that as network technology catches up to demand, new ways to use the additional functionality tend to push the envelope even further.

NPs have their origins within the vast sea of on-line niche markets, dot-com catastrophes and high-speed access to the desktop, as a revolution among the top providers of the Internet's infrastructure gained momentum. A strong divide existed in the early to mid 1990s between two main factions of protocol-based packet processor developers along the dimensions of flexibility verses speed. On one hand, developers who saw a need to accommodate a large diversity of protocols chose to use a General-Purpose Processor (GPP) to handle network traffic in order to sacrifice speed for flexibility. Those who fell into this group believed the future shape of the Internet to be a sea of protocols with intelligent translation between intranets. This group's focus, therefore, was on serving the need for flexibility of the periphery consumer, and largely overlooked the needs of the backbone suppliers. On the other hand, developers who wished to capitalize and improve upon the increased network speeds obtained during this era chose to produce Application-Specific Integrated Circuits (ASICs) for network traffic processing in order to sacrifice flexibility for speed. Those who fell into this camp saw the Internet as a close-knit community, moving toward a single unifying standard. This group's focus, therefore, was on serving the need for speed of the backbone suppliers, and largely overlooked the needs of the periphery consumer.

As these two camps strayed further apart, the inherent problems each faced by maintaining a hard-line viewpoint of the emerging Internet that did not materialize spelled drastic consequences for each. The GPP group saw a consolidation of protocols rendering their flexibility niche a moot point, while at the same time link speeds began to exceed processor clocks ten-fold rendering wire speeds unattainable. At the same time, the ASIC faction saw the same consolidation of protocols stop far short of the one or two protocols they believed would define the Internet. Consequently, the fixed nature of the ASIC has rendered their designs too inflexible for some applications. In addition, the tremendous production cost involved in creating ASIC designs has produced a high entrance barrier in the market, ensuring their use by only large-scale corporations. The one savior of the ASIC market has been their ability to keep in step with the increases in link speed in recent years. While this group has not suffered the hardships that befell the GPP group, the recent decline in sales of all traditional networking equipment may have signaled a shift in the trends of networking. Vendors are now realizing that optimizing for speed or flexibility alone will not meet tomorrow's market demands for routing switches, edge switches, network interface cards (NICs) and nodes that offer the best of both options. To meet these and other challenges,

both groups have begun to work toward the common goal of a next-generation NP.

Meanwhile, due to technological advances over the past decade, Reconfigurable Computing (RC) has garnered a great deal of attention from the academic community as well as industry [1]. RC systems have been shown to provide a computation speedup, in some application-specific domains, as large as 100 times as compared to GPPs with comparable resources [2]. In still other domains, RC device implementations have mimicked GPP performance at two-thirds the cost [3]. The most remarkable speedup has been seen in traditionally ASIC-laden markets such as digital signal processing (DSP) [4] and cryptography [5].

While this emerging technology has grown at an accelerated pace, there are still numerous obstacles to overcome. Producing a set of language standards, defining network protocols, standardizing benchmarks and even solidifying a name for the technology are all still to be accomplished [6].

However, RC designs have produced significant performance speedup in point-solution markets that have the same processing trends as the application domain of network processing. Therefore, RC-enhanced NP designs are poised to make an impact on future packet-processing systems in so-called active networks (AN). Some of the general topics within AN include resource management [7], adaptive flow control, adaptive error recovery, adaptive mesh interconnections, adaptive routing [8-9], adaptive node topologies and reconfigurable network links [10].

The organization of the remainder of this paper is as follows. Section 2 presents related research as it pertains to current generation NPs and how RC has entered NP designs. Section 3 outlines the proposed RC-enhanced NP model and the simulation environment in which it was created. Section 4 discusses the experiments by which the new model is compared to the baseline system. Section 5 presents a case-study analysis of the new model, and Section 6 discusses conclusions and directions for future work.

2. Related Research

A host of large-scale companies as well as numerous start-ups are shaping the face of new-generation NPs [11]. Each of the major players has been lured to new NPs due to their faster time-to-market as compared to traditional ASIC designs as well as the flexibility they provide akin to past GPP designs [12]. Also, market forecasters predict a coming surge in revenues for the NP market niche. In fact, combined revenues are expected to climb to $2.9 billion by 2004 [13]. Such potential market growth has garnered the respect of many of the industries biggest players.

The NP market is rich with a variety of designs. While fragmentation has meant the lifeblood of some of the smaller players in the NP design market, with numerous designs realized or in production, it is difficult to adequately judge the benefits of each. This fragmentation of the market has left much to be desired in terms of accurate head-to-head performance analyses of the notable models. However, there exists a great need to conduct such tests so that future designs will not make the same mistakes as those that preceded them. In addition, while it can be surmised that new NPs will outperform their ASIC and GPP equivalents in terms of flexibility coupled with speed [14, 15], the fact that they are better (and if so by how much) cannot be determined without structured comparisons. Of the proposed designs, there are three in particular that warrant more attention.

The Motorola/C-Port C-5 Digital Communication Processor (DCP), shown in Figure 1, represents one extreme of the NP market as a highly distributed architecture. The NP consists of 16 channel processors (CPs) and five co-processors, all connected through a 60Gbps bus. The channel processors, each of which consists of a 32-bit RISC core and two serial data processors (SDPs), are the heart of the unit. The SDPs are microcode-programmable to implement link-layer interfaces including Ethernet, SONET and serial data streams. Since each RISC core can execute a different program, and the channel processors share a common bus, there is a great deal of flexibility in distributing processing across the chip. There can be a parallel processing arrangement where identical programs can be execute on several CPs, or a pipelined arrangement where each processor is dedicated to a particular task and passes its output to the input of the next processor [16]. The C-5 DCP offers a wide range of processing options from network edge to core.

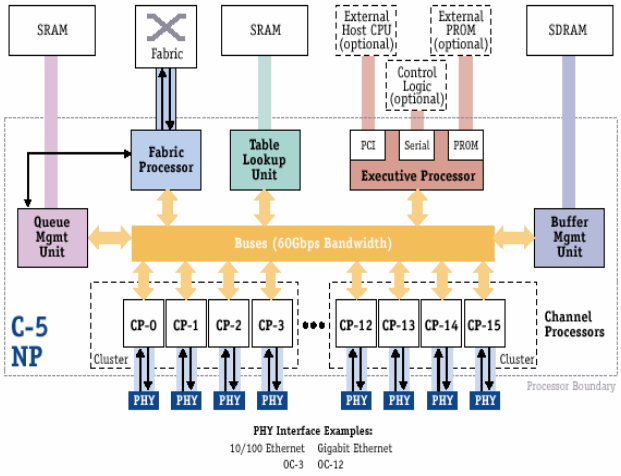

Figure 1. Motorola/C-Port DCP architecture
(Courtesy: Motorola Inc./C-Port Corp.)

The Intel IXP 1200 represents the middle of the road in terms of distributed versus tightly coupled processing. The device is a highly integrated, hybrid data processor that delivers high-performance parallel processing power and flexibility to a wide variety of networking, communications and other data-intensive applications. The IXP1200 is designed specifically as a data control element for applications that require access to a fast memory subsystem, a fast interface to I/O devices, and processing power to perform efficient manipulation of various data sizes [17].

The IXP1200 combines a StrongARM microprocessor with six independent 32-bit RISC data engines possessing hardware multithread support that, when combined, provide over 1 Giga-operations per second. The six MEs are reportedly capable of packet forwarding of 3 million Ethernet packets per second at Layer 3. The StrongARM processor is used for more complex tasks such as address learning, building and maintaining forwarding tables, and network management. Figure 2 shows the Intel IXP1200 architecture [18].

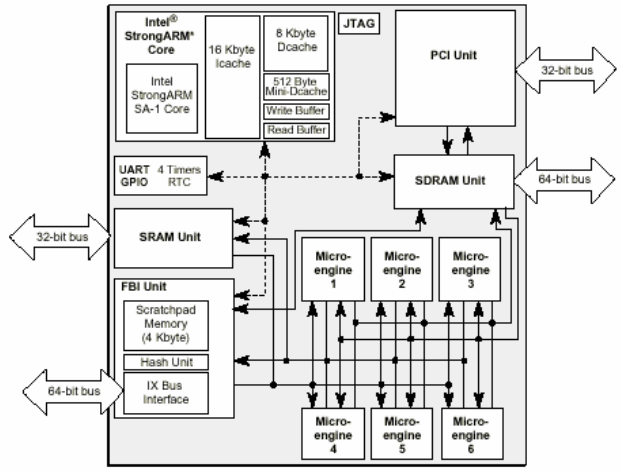

Figure 2. Intel IXP1200 NP
(Courtesy: Intel Corp.)

The Lucent/Agere PayloadPlus processor family represents the other market extreme as a tightly coupled NP solution. The architecture includes the Fast Pattern Processor (FPP), Routing Switch Processor (RSP) and the Agere System Interface (ASI). The PayloadPlus processor is designed to handle wire-speed data streams at up to OC-48c rates. Each specialized chip provides a complementary function to work in concert: the FPP for high-speed classification, the RSP for processing and routing traffic, and the ASI to provide policing, manage state information and provide a PCI connection to a host processor [19]. The PayloadPlus architecture is shown in Figure 3.

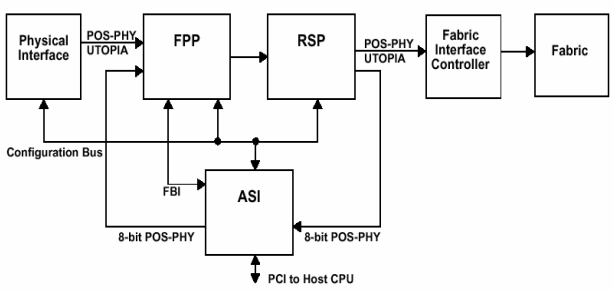

Figure 3. Lucent/Agere PayloadPlus NP
(Courtesy: Lucent Technologies/Agere Systems)

Within the realm of RC-enhanced NPs, the Reconfigurable Communications Processor (RCP) from Chameleon Systems is the first industry NP that uses dynamic reconfiguration as part of normal system operation. The architecture of the RCP is shown in Figure 4. The RCP consists of a 32-bit ARC processor, memory units and a 32-bit reconfigurable processing fabric that consists of 108 parallel computation units [20]. The RPC has been able to bridge the configuration latency problem that plagues RC systems by multiplexing contexts in the processing fabric. After initialization, context switching can be performed in a single clock cycle [21].

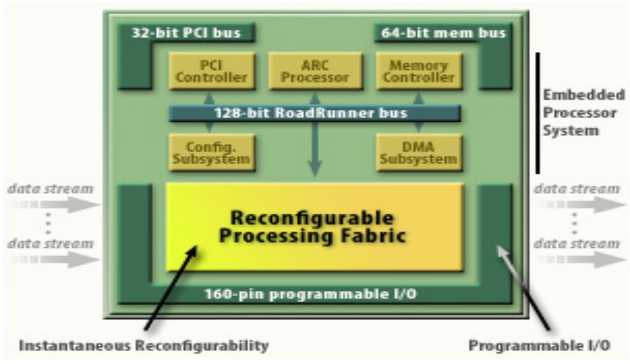

Figure 4. Chameleon Systems RPC
(Courtesy: Chameleon Systems Inc.)

For our research, a novel design and simulation model has been developed for a dynamically reconfigurable NP by adapting the architecture of the Intel IXP1200. Its purpose is to support the study of design options and tradeoffs in dynamic NP architectures versus static ones and help develop an understanding of how RC-enhanced NP devices may influence future systems. The IXP1200 was chosen as a basis due to the nature of its fixed processor coupled with flexible microengines that can be readily replaced with reconfigurable units. This design philosophy offers potent tradeoffs between performance, software tool support, flexibility, and versatility. Section 3 provides a description of the architecture for this RC-enhanced NP.

3. Simulation Model Description

A description of the new model and the simulation environment in which it was produced is presented in Sections 3.1 and 3.2, respectively. Section 3.3 details terminology that is specific to the model's architecture. Section 3.4 describes the values assigned to model parameters as well as how the model was verified against the Intel IXP1200.

3.1. Model overview

The moderately coupled, distributed processing approach highlighted in the Intel IXP1200 NP forms the basis of the new RC-enhanced NP. However, the six MEs (dictated by the IXP1200 design) that perform packet processing in the new system are provided with the capability to be dynamically reconfigured in the manner described in Section 3.3. A functional description of this new design is shown in Figure 5.

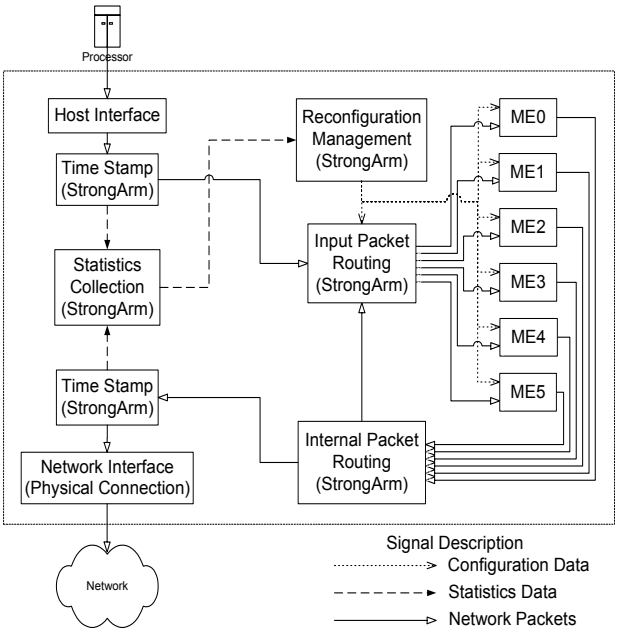

Figure 5. Functional diagram of RC-enhanced NP

The MEs in the new design perform pipelined packet processing as in the original IXP1200. However, in the new design, the MEs are not fixed components, but instead dynamically reconfigurable devices such as FPGAs. To function properly, runtime reconfigurable systems use a statistics gathering mechanism to determine when system adaptations are necessary. For the new design, the additional functions of internal packet routing, statistics analysis, and reconfiguration management is performed by the StrongArm processor. In the original Intel design, the StrongArm has much the same role in terms of packet routing and statistics management, so the addition of the reconfiguration management to its workload is considered to be a reasonable addition.

3.2. Simulation environment

Simulation is used to develop the RC-enhanced NP in order to accurately model the level of detail and flexibility inherent in RC systems. To model and simulate the device, the Block-Oriented Network Simulator (BONeS) Designer tool from Cadence Design Systems was used.

BONeS is an integrated software package for event-driven simulation of data transfer systems. It allows for a hierarchical, dataflow representation of hardware devices and networks with the ability to import finite-state machine diagrams and user-developed C/C++ code as functional primitives. BONeS was developed to model and simulate the flow of information represented by bits, packets, messages, or any combination of these. An overview of BONeS can be found in [22].

The simulation model constructed using BONeS to evaluate the RC-enhanced NP is of a high fidelity, consisting of approximately 3300 primitive blocks in six layers of depth that can be simulated to the accuracy of a clock cycle. The entire system model was constructed, tested and experiments performed in approximately 1500 hours over the course of 1.5 years.

3.3. Architecture terminology

An *ME pipeline* is a collection of MEs that performs packet processing in a pipelined manner. Three ME pipelines, with a depth of at least one ME per pipeline, exist in the model simulation. Due to the fact that the RC-enhanced NP has a total of six MEs, the upper bound for the depth of an ME pipeline is four. Numerous possible ME pipeline configurations exist for the model. One possibility allows only one ME to be assigned to each ME pipleline. Another possibility allows multiple MEs to be assigned to each ME pipeline. The use of additional computing resources would tend to increase an ME pipeline's processing throughput. Other possibilities include hybrid designs that offer mixtures of the two previous possibilities. The convention for describing the number of MEs allotted to each ME pipeline is given by [*x,y,z*] where *x*, *y* and *z* denote the number of MEs allotted to the first, second, and third pipeline respectively.

ME pipeline configurations for the NP system fall into two main types. The first type, *baseline configuration*, is a collection of all three available ME pipelines that cannot be reconfigured during the course of packet processing. Baseline configurations represent the NP system without any RC enhancement. An *optimal baseline configuration* is the one that is best suited to process a specific set of packets.

The second type, *dynamic configuration*, is a collection of all three available ME pipelines that can be reconfigured during the course of packet processing. The decision process for possible reconfiguration occurs periodically at the *decision interval*. This interval is defined as the period between decisions and lasts the length of time it takes for the slowest ME pipeline to process five packets. At the end of a decision interval, the *per-pipeline latency* is polled. This latency is defined as the number of clock cycles it takes for a packet to move from the input of a ME pipeline to the output. The per-pipeline latency is compared to the *latency threshold* for each ME pipeline to determine the need to reconfigure the system. A separate latency threshold value exists for each ME pipeline. If a per-pipeline latency value is above the latency threshold for that ME pipeline then a reconfiguration is attempted. The manner in which the NP performs reconfiguration is detailed in Section 4.2.

3.4. Parameters and verification

All system delays are defined as multiples of the ME clock period of 8.33ns (i.e. frequency of 120MHz). Packets arrive at the ingress point of the NP at a rate of one packet per ten ME clock cycles. This rate is faster than any ME pipeline can process a single packet, ensuring the ME pipelines are the bottleneck of the system. The PCI host interface, memory access delay, and physical network interface are not included in our model for simplicity. Rather, these values are given a fixed latency value of one ME clock cycle in order to keep packets from being overwritten in the simulation. The reconfiguration latency, or the number of ME clock cycles it takes to change ME pipeline configurations, is assumed to be one ME clock cycle as in Chameleon Systems' RPC mentioned in Section 2.

In order to gauge the relative performance of the simulation of the baseline configuration as compared to the original IXP1200 design, a verification of the system was performed as follows. First, a baseline configuration of our NP model with two MEs per ME pipeline is created. Second, 5000 Gigabit Ethernet (GigE) packets are passed through this NP model. It was found that the device maintained a steady-state packet-processing rate of 2.65 million packets per second. This value is comparable to the 3 million Ethernet packets per second asserted by the Intel documentation [17].

4. Simulation experiments

The following section details the manner in which the baseline configurations are compared to the dynamic configurations. Section 4.1 describes the packet processing methodology used for the experiments, and Section 4.2 introduces the terminology used to characterize the packet processing. Section 4.3 presents the *baseline experiments* in which the optimal baseline configuration for each of the different cases is observed. The results from the baseline experiments are compared to the dynamic configurations in the *head-to-head experiments* detailed in Section 4.4.

4.1. Packet processing description

The NP creates the GigE packet header information for data and destination address pairs that are passed to it from the host processor. Upon close analysis of the work involved in packet header construction, it is observed that a majority of processing time is spent computing the Cyclic Redundancy Check (CRC). Also, within the CRC operation, the majority of processing is taken up performing a 32-bit polynomial division. Therefore, system reconfiguration in order to better adapt a given ME pipeline is centered on the polynomial divide operation. Four possible divide operations were chosen for this research as described in [23]. For each operation, the cost, represented as a number of transistors, as well as the performance, represented as processing latency, is shown in Table 1. The values of transistor size and processing latency for each operation are converted to a number of MEs and clock cycles to produce meaningful values for the purposes of the simulation. Integer values of MEs were used since they represent the number of stages in the ME pipeline.

Table 1. Divide operation performance vs. area

Divide Operation	Cost (Area)		Performance	
	Transistors	MEs	Latency (ns)	CCs
32-array	32,896	4	160	20
16-digital serial	11,386	3	220	26
Modified booth	3,808	2	320	38
Quasi bit-serial	1,944	1	640	77

There exists a direct relation between cost and performance for the divide operations as can be observed in Table 1. Our RC-enhanced NP relies upon the assumption that the divide operation can be partitioned without significant loss of performance. Previous research has shown that breaking up such complex polynomial computations can be accomplished with little additional cost from communication overhead [24].

4.2. Traffic flow, mixtures and priorities

A *traffic flow* is the basic unit of differentiation between types of network traffic that are processed by the NP. Each traffic flow originates from the host processor.

For the purposes of this research, three traffic flows are used and each is mapped exclusively to one of the three ME pipelines. A *priority scheme* is used to denote the relative importance of the traffic flows, and each traffic flow is assigned an exclusive priority within a priority scheme. Three priority schemes have been defined for our NP model. The *1PC, 2PC and U* schemes allocate priority to the traffic flows as follows:

One-priority critical (1PC): P0 > P1 > P2

Two-priority critical (2PC): P0 = P1 > P2

Uniform (U): P0 = P1 = P2

For a dynamic configuration, priority schemes are also used to determine the manner in which MEs are added to a given traffic flow's ME pipeline. If the per-pipeline latency of a given traffic flow's ME pipeline is above the latency threshold value for the priority assigned to that traffic flow, the NP attempts to allocate more MEs to that ME pipeline. If a free ME exists, the NP will assign it to the requesting ME pipeline. However, if no MEs are free, then the NP will take away a ME of another traffic flow's ME pipeline if two conditions are met. First, the priority value assigned to that traffic flow must be less than the priority value of the requesting traffic flow as dictated by the priority scheme equations. Second, the traffic flow's ME pipeline depth must be greater than one ME.

A *traffic mixture* describes the relative number of packets from each traffic flow that makes up the total number of packets for the NP to process. A traffic mixture is expressed as the relative percentage of the number of packets from each traffic flow out of a total of 3000 packets. Three traffic mixtures (*1-Hot*, *2-Hot*, *3-Hot*) have been defined for the NP simulation model. Within the 1-Hot traffic mixture, a single priority dominates during any given time period. 1-Hot contains the following traffic flow percentages: the first 1000 packets are made up of 90% P0, 5% P1 and 5% P2; the second 1000 packets are made up of 5% P0, 90% P1 and 5% P2; and the third 1000 packets are made up of 5% P0, 5% P1 and 90% P2.

Within the 2-Hot traffic mixture, two priorities dominate during a given time period. 2-Hot contains the following traffic flow percentages: the first 1000 packets are made up of 45% P0, 45% P1 and 10% P2; the second 1000 packets are made up of 45% P0, 10% P1 and 45% P2; and the third 1000 packets are made up of 10% P0, 45% P1 and 45% P2.

Finally, the 3-Hot traffic mixture provides a near-uniform traffic mixture in which all priorities are equal. Due to limitations of the simulation's traffic generators, only increments of 5% were possible, resulting in a distribution of 35%, 35%, and 30%. However, it was determined that the affect of the small imbalance in the distribution is negligible. The 3-Hot traffic mixture contains the following traffic flow percentages: the first 1000 packets are made up of 35% P0, 35% P1 and 30% P2; the second 1000 packets are made up of 35% P0, 30% P1 and 35% P2; and the third 1000 packets are made up of 30% P0, 35% P1 and 35% P2.

4.3. Baseline experiments

In the baseline experiments, the optimal baseline configuration for each of the three traffic mixtures (1-Hot, 2-Hot, 3-Hot) is determined for each of the three priority schemes (1PC, 2PC, U). The observed values are given in Table 2. The name for each priority scheme in the baseline experiments is preceded by "*B_*" for clarity. While the total execution time, per-pipeline latency, per-pipeline throughput, and pipeline utilization were all measured and taken into consideration, only the per-pipeline latency values are presented in this paper as a means of comparison between the configurations. As evidenced in Table 2, B_2PC and B_U have identical optimal baseline configurations. Therefore, the B_U case will be used to represent both these priority schemes for the head-to-head experiments described in following subsequent sections.

Table 2. Optimal baseline configurations

Traffic Mixture	Priority Scheme		
	B_1PC	B_2PC	B_U
1-Hot	[4,1,1]	[3,2,1]	[3,2,1]
2-Hot	[4,1,1]	[3,2,1]	[3,2,1]
3-Hot	[4,1,1]	[2,2,2]	[2,2,2]

4.4. Head-to-head experiments

In the head-to-head experiments, the optimal baseline configurations for each of the three traffic mixtures and the three priority schemes are compared to the dynamic ones. The three priority schemes for the dynamic configurations are preceded by a "*D_*" for clarity. The latency thresholds for each of the three priority schemes are given in Table 3.

Table 3. Dynamic priority latency thresholds

	Priority		
Priority Scheme	P0	P1	P2
D_1PC	50 CCs	70 CCs	100 CCs
D_2PC	80 CCs	80 CCs	100 CCs
D_U	120 CCs	120 CCs	120 CCs

4.5. Head-to-head results

The head-to-head experiments are presented next. The latency values used to compare the optimal baseline

configurations to the dynamic configurations are averaged over the per-pipeline latency for each traffic flow. In addition, only the results of the best dynamic configuration for each experiment are shown (e.g. D_1PC scheme in the case of the 1-Hot traffic mixture).

Results from the 1-Hot experiment

The average latency results for the 1-Hot experiment are shown in Figure 6. D_1PC produces low latency values for each priority because MEs are reallocated to each ME pipeline as needed. P2 suffers a severe latency with the B_U scheme, and both P1 and P2 suffer severely with B_1PC, due to the fact that they cannot reallocate MEs. The dynamic system is found to reconfigure a total of only 17 times in this experiment.

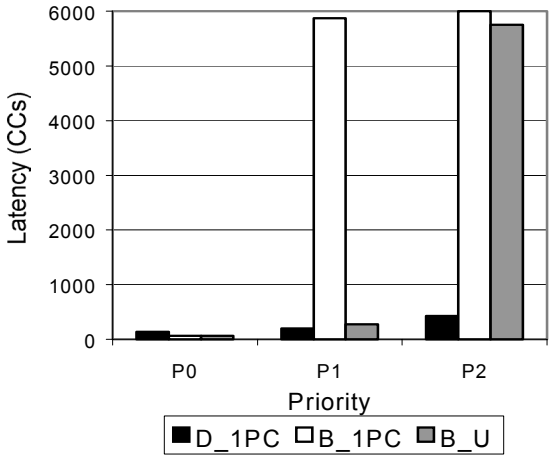

Figure 6. Average latency results with 1-Hot

Results from the 2-Hot experiment

The average latency results for the 2-Hot experiment are shown in Figure 7. D_2PC produces latency values that are fairly comparable to the previous experiment for each priority despite a substantial increase in the number of system reconfigurations. Due to the fact that the traffic mixture is becoming more uniform, the optimal baseline configurations are less subject to the latency penalties inherent in having ME pipelines that are fixed. The B_U scheme, in particular, produces good latency values for all priorities because of this trend. The dynamic system is found to reconfigure a total of 90 times in this experiment.

Results from the 3-Hot experiment

The average latency results for the 3-Hot experiment are shown in Figure 8. D_U produces poor latency values for each priority due to a large number of system reconfigurations. The relative effect versus the baseline is more severe in this experiment as compared to the previous one because the reconfigurations are creating ME pipelines that are not ideally suited for the current traffic mixture in addition to consuming processing cycles during reconfiguration. This condition occurs because the traffic mixture at any given time changes so rapidly that the dynamic system cannot adapt in a meaningful and efficient manner. The B_U scheme produces low latency values for all priorities because it is always ideally suited for a uniform traffic mixture. B_1PC continues to produce a poor latency value for P2 because this priority scheme is not well suited for a uniform traffic mixture. The dynamic system is found to reconfigure a total of 110 times in this experiment.

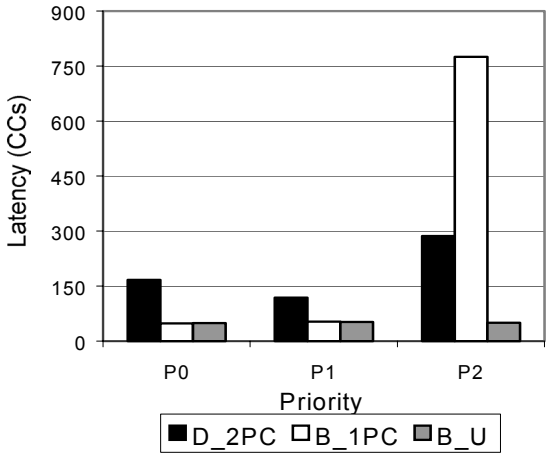

Figure 7. Average latency results with 2-Hot

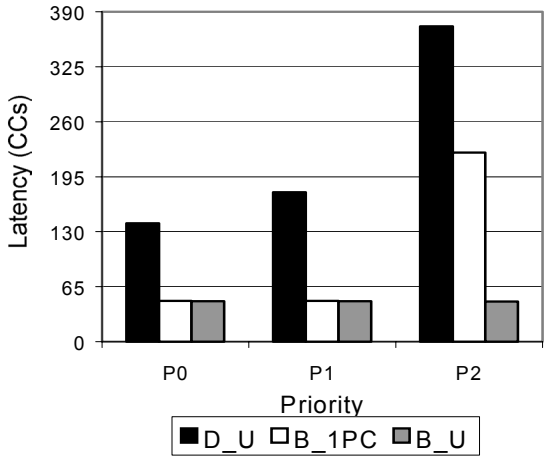

Figure 8. Average latency results with 3-Hot

It should be noted that the scale of the previous three charts has been significantly reduced in the progression from 1-Hot to 2-Hot to 3-Hot results. Therefore, latency penalties that occur in the 3-Hot experiment are clearly much less severe than those in the 1-Hot experiment.

These results illustrate three main points. First, dynamic configurations perform far better than the baseline configurations when traffic mixtures are non-uniform. Second, dynamic configurations suffer from resource thrashing if traffic mixtures become uniform. Third, the decrease in performance incurred by using dynamic configurations when processing a uniform traffic mixture is far less severe than the decrease in performance incurred by using a baseline configuration when processing non-uniform traffic mixtures.

In studying a wide variety of real-world network traffic patterns, the argument could be made that the majority of network traffic mixtures are non-uniform over an arbitrary time period. Given this assumption, the use of dynamic NP resource allocation is likely to have a large impact on tomorrow's networking environments. In the next section, a description of the traffic mixture for a case study used to stimulate our NP model is presented.

5. Case study

The case-study experiments employ the same testing strategy used in the previous section. First, the optimal baseline configuration for the case study's traffic mixture is found for each of the priority schemes. Second, optimal baseline configurations are compared to the three dynamic configuration priority schemes in head-to-head experiments.

5.1 Description

The traffic mixture for this case study is inspired by packet types used in the network of a Navy system for theatre air and missile defense known as Cooperative Engagement Capability (CEC). This system, under development and deployment by the U.S. Navy for the past decade, supports the total automation of fleet defense systems. The basic concept is illustrated in Figure 9.

Each player in the theatre of operations communicates using three traffic flows defined as missile detection and tracking (denoted as radar), cooperation instructions (denoted as command) as well as engagement information (denoted as weapon). For the purposes of this case study, the three traffic flows are mapped onto the three priorities such that radar is P0, command is P1 and weapon is P2.

There exists a series of actions through which all players must possess in order to destroy a missile that is known as the "kill chain." This series of actions involves three phases (detect, control and engage) illustrated in Figure 10. The 3000 packets from which the traffic mixture is calculated have been divided equally between the three phases of the "kill chain."

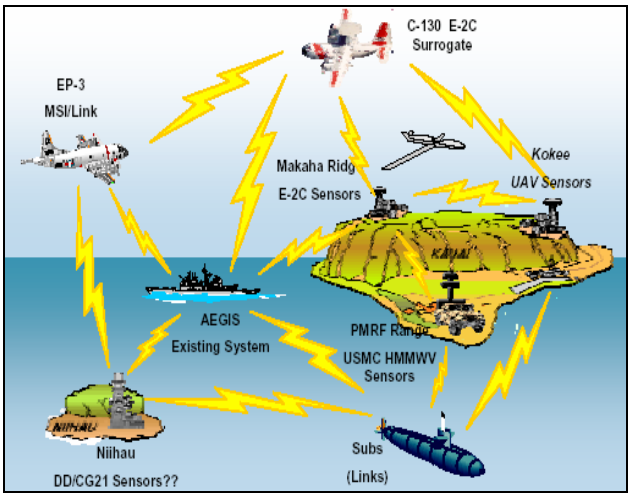

Figure 9. Theatre missile defense system
(Courtesy: ONR)

As an actual battle traffic mixture is not publicly available, a candidate traffic mixture was developed. This traffic mixture contains the following traffic flow distribution: the first 1000 packets (detect phase) are made up of 85% P0, 10% P1 and 5% P2; the second 1000 packets (control phase) are made up of 35% P0, 60% P1 and 5% P2; and the third 1000 packets (engage phase) are made up of 15% P0, 30% P1 and 55% P2.

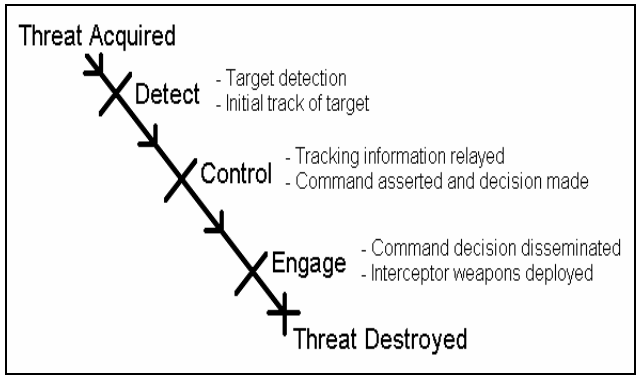

Figure 10. Kill chain of theatre missile defense

5.2 Baseline results

The optimal baseline configurations for the case study were determined and are given in Table 4. Note that since B_2PC and B_U have identical optimal baseline configurations, the B_U case will represent both priority schemes in the subsequent experiment.

Table 4. Optimal baseline configurations

Traffic Mixture	Priority Scheme		
	B_1PC	B_2PC	B_U
Case Study	[4,1,1]	[3,2,1]	[3,2,1]

5.3 Head-to-head results

The optimal baseline configurations for each of the three priority schemes are compared to the three priority schemes for the dynamic configurations. The average latency results for the case study are shown in Figure 11.

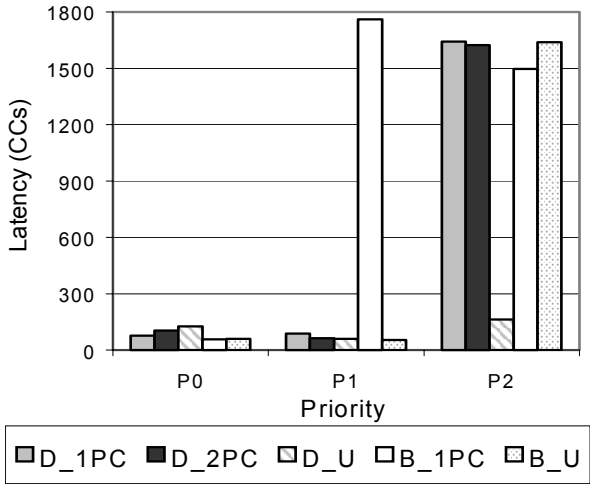

Figure 11. Average packet latency results

All configurations produce low latency values for P0. However, B_1PC produces high latency values for P1 and P2, while the B_U scheme produces high latency values for P2, both due to the fact that MEs cannot be reallocated. For the dynamic cases, D_U produces low latency values for all three priorities while the other two dynamic configurations did not produce low latency values for P2. D_U performed well regardless of having a high number of system reconfigurations because these configurations were meaningful and produced ME pipelines that were well suited for the current traffic mixture even after reconfiguration. D_1PC and D_2PC did not produce configurations that were well suited to handle P2 traffic. The reconfiguration count for the dynamic systems is found to be 16 times for D_1PC, 18 times for D_2PC and 63 times for D_U.

The case study results illustrate two main points. First, the need for dynamic configurations to allow MEs to be available for any traffic flow's ME pipeline to use is demonstrated. This point is evidenced in the fact that the D_U scheme did not suffer from system resource thrashing (as seen in the 3-Hot head-to-head experiment in Section 4.5) even though it reconfigured four times more often than the other two dynamic configurations. Second, the dynamic configurations performed as well, if not better, overall than each of the ideal baseline configurations.

6. Conclusions

In this paper, the functionality of an RC-enhanced NP model is simulated using BONeS Designer, a software tool for event-driven simulation of computer network systems that offers a high degree of fidelity. The accuracy of the baseline model was verified in terms of a nominal value for sustained packet processing asserted in the Intel IXP1200 documentation. The RC-enhanced NP dynamic configuration is compared to a baseline configuration in order to gauge relative performance. In addition, a case study is presented to highlight the effects of dynamic reconfiguration on a traffic mixture patterned after a complex system design. This research has shown the potential advantages from incorporating RC design techniques into next-generation NPs.

In the baseline experiments, the ME pipelines were not reconfigurable. This type of system mimics the behavior of today's NPs that are not RC enhanced. The baseline experiments produced the optimal baseline configurations used for comparison of the baseline NP to the dynamic NP in the head-to-head experiments.

A head-to-head comparison of the baseline NP and dynamic RC-enhanced NP illustrated three main points. First, dynamic configurations perform far better than baseline configurations when traffic patterns are non-uniform due to the fact that dynamic configurations can reallocate MEs. Because the baseline configurations cannot reallocate MEs, many processing clock cycles are wasted due to an imbalance of system resources.

Second, dynamic configurations may suffer from resource thrashing if traffic patterns become uniform. This situation may occur if the decision interval assumed in this research is too fine grained as compared to the time it takes to process the total number of packets per experiment. By having a small window in which to determine if a system reconfiguration is necessary, the dynamic configurations reconfigured the system based on transient traffic mixtures rather than adapting to the overall traffic mixture over time. In fact, the results demonstrate that it is better to not reconfigure the system at all for uniform traffic mixtures (U).

Third, the decrease in performance incurred by using dynamic configurations when processing uniform traffic mixes is far less severe than the decrease in performance incurred by using a baseline configuration when processing non-uniform traffic. This phenomenon is due to the fact that the baseline systems are only optimized for one traffic mixture and are therefore inadequately suited to process packets in other traffic mixtures.

The case study illustrated two points. First, a high number of ME pipeline reconfigurations does not necessarily imply resource thrashing and performance loss in dynamic configurations. This fact is observed when the configuration for the D_U scheme performed

well for all priorities even though it reconfigured four times more than other dynamic configurations. The D_U configuration accomplished this by reconfiguring the system in a meaningful way in order to produce ME pipelines that were well suited for the current traffic mixture even after reconfiguration. Therefore, the decision interval used in the case study experiment has a better granularity for the traffic mixture as compared to the head-to-head experiments in Section 4. Second, the dynamic configurations were found to perform at or above the level of the baseline configurations for the case-study traffic mixture due to the fact that resources in the NP can be reallocated to the ME pipelines of other traffic flows as needed.

Future directions for this research include adding the capability to simulate full-duplex packet processing for multiple RC-enhanced NPs in a complex network. In creating a network of nodes, implementing additional functionality such as network routing information, flow control and quality of service (QoS) mechanisms would be a logical next step. Another area for future work is to study issues with the migration of RC techniques to other NP designs. This direction would help to ascertain the relationship between the emphasis of the NP architecture (e.g. loosely coupled versus tightly coupled processing) and the most effective methods for augmenting them to achieve increased performance through adaptive, reconfigurable processing.

7. Acknowledgements

This work was sponsored in part by the U.S. Dept. of Defense.

8. References

1. A. DeHon and J. Wawrzynek, "Reconfigurable Computing: What, Why and Implications for Design Automation," *Proc. 36th ACM Design Automation Conference (DAC)*, New Orleans, LA, June 21-25, 1999.
2. J. Vuillemin, P. Bertin, et al., "Programmable Active Memories: Reconfigurable Systems Come of Age," *IEEE Transactions on VLSI Systems*, Vol. 4, No. 1, March 1996, pp. 56-69.
3. Star Bridge Systems: *Theory of Hypercomputing*, Technology Overview, 2002.
4. H. Choi and J. Kim, et al., "Synthesis of Application Specific Instructions for Embedded DSP Software," *IEEE Transactions on Computers, Vol. 48, No. 6, June 1999*, pp. 603-614.
5. O. Mencer, M. Morf and M. Flynn, "Hardware Software Tri-design of Encryption for Mobile Communications units," *Proc. IEEE Inter. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Seattle, WA, May 1998.
6. B, Dipert, "Figuring out reconfigurable logic," *EDN Magazine*, August 5, 1999, pp.103.
7. A. Staicu, J. Radzikowski, et al., "Effective Use of Networked Reconfigurable Resources," *Proc. Int. Conf. on Military Applications of Programmable Logic Devices (MAPLD)*, Laurel, MD, September 2001.
8. A. Dollas, D. Pnevmatikatos, [et all], "Architecture and Applications of PLATO, a Reconfigurable Active Network Platform," *Proc. 9th Annual IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM)*, Rohnert Park, CA, April 29-May 2, 2001.
9. A. DeHon, R. Huang and J. Wawrzynek, "Hardware-Assisted Fast Routing," *Proc. 10th Annual IEEE Symp. on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, April 21-24, 2002.
10. S. Johansson, "Transport Network Involving a Reconfigurable WDM Network Layer – A European Demonstration," *IEEE Journal of Lightwave Technology*, Vol. 14, No. 6, June 1996, pp. 1341-1348.
11. X. Zhu, "Network Processor Directory," *Gigascale Silicon Research Center*, 2001, available at: http://www.gigascale.org/mescal/forum/110.html
12. J. Caruso, "Network Processors Combining the Speed of ASICs with the Flexibility of General-Purpose Processors," *Network World Fusion*, September 15, 1999.
13. D. McEuen, "Network Processor Revenues Climb from Obscurity in 1999 to $2.9 Billion by 2004," *In-stat MDR*, Press Release, March 6, 2000.
14. E. Rothfus, "The Challenge for Next Generation Network Processors," Agere, White Paper, 1999.
15. E. Rothfus, "Building Next Generation Network Processors," Agere, White Paper, 1999.
16. Motorola: *C-5 Network Processor*, Product Summary, 2002.
17. Intel: *IXP1200 Network Processor*, Product Brief, 2002.
18. Intel: *IXP1200 Data Sheet*, Product Data Sheet, 2002.
19. Lucent/Agere: *PayloadPlus NPs*, Family Product Brief, 2002.
20. X. Tang, M. Aalsma and R. Jou, "A Compiler Directed Approach to Hiding Configuration Latency in Chameleon Processors," *Proc. 10th International Conference on Field Programmable Logic and Applications*, Villach, Austria, 2000.
21. Chameleon Systems: *CS2000 Reconfigurable Communications Processor*, Family Product Brief, 2000.
22. K. Shanmugen, V. Frost and W. LaRue, "A Block-Oriented Network Simulator (BONeS)," *Simulation*, Vol. 58, No. 2, 1992, pp. 83-94.
23. R. S. Martin and J.P. Knight, "Power-Profiler: Optimizing ASICs Power Consumption at the Behavioral Level," *Proc. 32nd ACM Design Automation Conference (DAC)*, San Francisco, CA, June 1995, pp. 42-47.
24. C. Tanougast, Y. Berviller and S. Weber, "Optimization of Motion Estimator for Run-Time-Reconfiguration Implementation," *Proc. 7th Reconfigurable Architectures Workshop (RAW)*, Cancun, Mexico, May 1-5, 2001.