

Addressing the Energy-Delay Tradeoff in Wireless Networks with Load-Proportional Energy Usage

Jie Chen and Biplab Sikdar

Department of Electrical, Computer and Systems Engineering

Rensselaer Polytechnic Institute, Troy, NY, 12180

Email: {chenj24,sikdab}@rpi.edu

Abstract—Hardware techniques such as dynamic voltage and frequency scaling may be used to reduce the energy consumption of network interfaces and achieve load-proportional energy usage. These techniques slow down the operation of the hardware and thus their power savings come at the cost of increased packet delays. This paper presents a methodology to address the energy-delay tradeoff while achieving load-proportional energy usage in wireless networks. The proposed system uses a pipelined implementation of the functional blocks of the medium access control (MAC) layer. Each functional block has its own job queue and is treated as an individual system with an independent clock. The clock frequency of each functional block is dynamically selected based on the length of its job queue. While the pipelined implementation reduces the MAC layer processing delays, the use of queue-length based frequency scaling provides load-proportional energy usage and enhances the overall stability of the system. The performance of the proposed system has been verified through extensive simulations.

I. INTRODUCTION

The energy efficiency of wireless networks has become increasingly important in recent years due to the environmental concerns as well as technological and financial considerations. Power management is not one of the design principles for existing networking hardware such as those for wireless local area networks. As a result, their energy consumption rates are independent of the traffic they are carrying, implying that these devices consume close to the maximum energy rating even in low load situations [1], [2]. Recent work has shown that load-proportional energy usage may be achieved in both wired and wireless network interfaces by controlling the operating frequency or voltage of these devices based on the existing load conditions [3], [4]. However, the use of dynamic voltage or frequency scaling (DVFS) for reducing the energy consumption reduces the operating speed of the devices and comes at the cost of increased packet delays. To objective of this paper is to develop a mechanism that can reduce the packet delays while maintaining load-proportional energy usage in wireless local area networks.

Energy efficiency of protocols and hardware for wireless networks has received considerable attention in literature. The protocol aspects of the energy efficient design of wireless networks focuses on a number of topics such as choice of paths with lowest energy consumption, transmission power control, energy-efficient scheduling and MAC protocols etc. However, while these techniques may reduce the overall energy consumption of the network, as long as the devices are on,

the rate of energy consumption remains constant and does not depend on the offered load. As a result, the energy consumed by the devices when they have low loads is the same as the energy consumed while handling heavy loads. To address this issue, hardware level techniques such as dynamic voltage and frequency scaling have been proposed for achieving load-proportional energy usage [3], [4]. While reducing the voltage and/or the clock frequency reduces the energy consumption, it also reduces the rate at which packets are processed by the MAC layer (by reducing the rate at which instructions are executed), thereby increasing the packet delays. In this paper our focus is on addressing this delay penalty by developing a network interface card architecture that minimizes the packet delays while providing load-proportional energy usage.

The main contribution of this paper is the development of a pipelined architecture for MAC layer packet processing where the functional blocks use dynamic frequency scaling (DFS) to achieve load-proportional energy usage and low packet delays. The packet processing steps for transmitting a packet at the network interface card are broken into five functional blocks: (1) packet read from the transmit buffer, (2) IPsec and management functions, (3) packet framing, (4) MAC layer packet encryption, and (5) modulation and coding. To reduce the packet delays, the processing steps are pipelined through the five stages. Each packet is passed sequentially through the functional blocks and at any given point in time, different stages of the pipeline may be processing different packets. The processing speed of each step is independently controlled using DFS to ensure load-proportional energy usage. A queue length based frequency selection mechanism is presented for the DFS which minimizes the energy consumption of each stage while working to maintain queue stability. The performance of the proposed architecture has been evaluated using extensive simulation. Our results show that the proposed framework outperforms traditional network interfaces for the IEEE 802.11 protocol as well as interfaces that implement DFS, in terms of the energy consumption, average packet delays, and packet drop rates.

The rest of the paper is organized as follows. Section II presents the related work and background information. Section III presents the proposed architecture for the pipelined dynamic frequency scaling system. Section IV presents the simulation results to evaluate the proposed mechanism. Finally, Section V concludes the paper.

II. BACKGROUND AND RELATED WORK

This section presents the related work in the development of mechanisms for load proportional energy usage in computer networks. We also present an overview of the wireless system considered in this paper.

A. Background and Related Work

The use of DVFS to reduce to energy consumption of electronic systems is a well investigated technique that is also implemented in commercially available hardware. The majority of the existing literature in this area addresses the problem of selecting the operating voltage and/or frequency for various scenarios of interest, with the primary objective of reducing the overall energy consumption. Existing approaches for voltage and frequency selection are primarily based on workload decomposition [5], [6], [7] and differ in their ability to accommodate different constraints and objectives.

In the context of networking devices, DVFS has been applied for energy savings in wired as well as wireless networking devices [8], [4]. Mechanisms for selecting the operating frequency in networking devices based on predicted values of future traffic have been developed [9]. DFS for network routers is considered in [10] while [11] addresses the problem of implementing DVFS schemes in parallel network processors by modeling the system as a $M^x/D/1/SET$ queuing system.

In all the systems developed and considered in existing literature, the energy savings resulting from the use of DVFS comes at the cost of increased packet delays due to the reduction in the processing speeds associated with voltage and frequency scaling. In contrast, the focus of this paper is on developing a system that minimizes the packet delays while ensuring the load proportional energy usage facilitated by DFS.

B. System Model

This paper considers a wireless network interface card for local area networks based on the IEEE 802.11 protocol [12]. Note that the architecture based on pipelining of the functional blocks proposed can also be applied to network interfaces for other protocols.

In general, a network interface card for an IEEE 802.11 based network consists of functional blocks for the MAC and the physical (PHY) layers. At the PHY layer, the network interface card includes functional blocks for modulation and coding, and radio frequency (RF) transmit and receive circuitry. At the MAC layer, the network interface card includes functional blocks for the transmit (TX) and receive (RX) buffers and the associated buffer manager, blocks for packet level processing such as framing and encryption, and a control unit. The network interface card may also include blocks for higher layer functionalities such as Internet Protocol Security (IPsec) and management. Finally, the network interface card also includes functional blocks for interfacing with the wired network (usually Ethernet) in the case of wireless access points and base stations. For network interface cards designed for

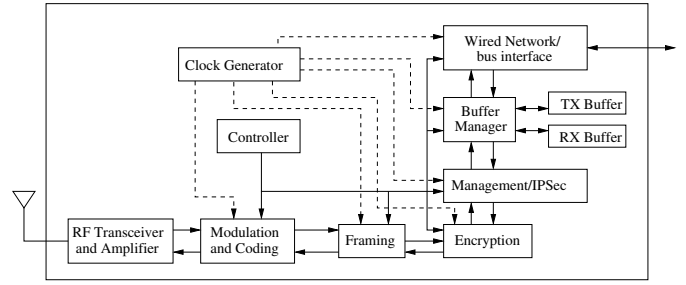


Fig. 1. Overview of the proposed system architecture.

use in computing devices (e.g. laptops), functional blocks for interfacing with the computer's system bus are included.

III. PROPOSED SYSTEM ARCHITECTURE

This section presents the proposed pipelined architecture for a wireless network interface card. A description of the architecture is followed by the frequency selection mechanism for the DFS mechanism.

The proposed system architecture is presented in Figure 1. The figure shows the basic functional blocks of the network interface card and the flow of data and control information between them. Note that additional functionalities, if desired, may be easily added while maintaining the pipelined nature of the architecture. Any packet to be transmitted through the wireless network is first sent to the wireless network interface card through the system bus or wired network interface. The received packet is then stored in the TX buffer using coordination provided by the buffer manager. Once this packet is ready for transmission (usually when it moves to the head of the queue of the TX buffer), the network interface card first does the tasks, if any, related to IPsec and management (such as updating statistics). The packet is then passed on to the MAC layer encryption block. This functional block encrypts the packet as per the security protocol implemented by the MAC layer, such as Wired Equivalent Privacy (WEP) and Temporal Key Integrity Protocol (TKIP). The encrypted packet (i.e. payload) is then passed on to the framing functional block which adds the necessary MAC layer headers and trailers. The complete MAC layer packet is then passed on to the physical layer where the first functional block takes care of the modulation and coding. Finally the packet is passed to the RF circuitry for transmission.

In traditional systems, each packet processed individually and passed sequentially through the functional blocks. Thus at any point in time, only one of the functional blocks is active. While this is inefficient use of system resources, it is generally acceptable since the processing times are usually smaller than the packet transmission times. However, with the use of DFS, the processing speed of each block may be decreased to reduce the energy consumption. As a result, the packet processing times increase and become a significant part of the packet delays.

To reduce the impact of the increased delays associated with the use of DFS, we propose a pipelined implementation

TABLE I
PIPELINE STAGE FUNCTIONS AND LENGTHS

Block	Length	Factors
Buffer Read	Variable	Packet length
IPSec/Management	Fixed	-
Encryption	Variable	Packet length, protocol
Framing	Variable	Packet length
Modulation	Variable	Packet length, code rate

of the functional blocks inside the network interface card. Each functional block is a stage of the pipeline and has its own, local queue of packets to be processed. Packets still pass sequentially through the functional blocks but the use of a pipelined architecture allows each functional block to be simultaneously active and process a different packet. Consequently, the rate at which the interface card is able to process packets and line them up for transmission increases.

To fully exploit the benefits of pipelining, we treat each functional block as an independent system from the perspective of DFS. Thus each functional block has its own independent clock whose frequency is selected based on the policy specified in Section III-A. The use of an independent clock for each stage serves two purposes. First, we note that the length of each stage of the pipeline (i.e. the time taken to process a packet by a stage) is different. Thus, the slowest functional block would become the bottleneck and its queue may overflow, leading to either a slowdown or loss of packets. By treating each block independently, a higher clock rate may be used for the longer pipeline stages to speed them up. At the same time, lower clock rates may be used for shorter pipeline stages, thereby ensuring the largest possible energy savings. Secondly, the use of independent clocks tends to make the lengths of the pipeline stages as equal as possible, thereby increasing the pipeline efficiency.

The length of each stage of the pipeline depends on the nature of the tasks that are conducted in that stage. Table I lists the nature of tasks carried out in each stage and the factors that impact the length of the stage. To evaluate the length of the functional blocks in the pipelined architecture, we have used results from [16], [15], [17]. The clock cycles associated with buffer read/write operations and packet framing in IEEE 802.11 devices has been investigated in [16] and these have been used for calculating the lengths of the corresponding functional blocks of this paper. The length of the IPSec/management stage in the pipeline is independent of the packet length and is assumed to be a constant. The complexity of this task is comparable to that of buffer management and we assume similar values for this stage. In this paper we assume that the TKIP mechanism is used for encrypting the packet. To calculate the length of the encryption stage, we use results from [15] where the number of clock cycles necessary for executing the TKIP encryption protocol in IEEE 802.11 has been evaluated. Finally, for the modulation block, we use the results of [17] where the computational cycles required by IEEE 802.11 compliant digital baseband transmitters has been

evaluated.

In the proposed system, the functional block related to the system bus/wired network interface has not been included in the pipeline or the DFS. This is because this functional block needs to interact with a standardized protocol (such as the Peripheral Component Interconnect (PCI) bus) and may not have the flexibility to arbitrarily change the rate at which it handles data. Consequently, we fix the clock rate for this block and allow it to function independently.

Finally, we note that the proposed system incurs additional complexity in terms of maintaining pipelined stages and separate clock frequencies at each stage. Technically these are feasible and easy to implement. The benefits provided in terms of lower packet delays and greater energy savings compensates more than adequately for the additional complexity.

A. DFS Frequency Selection

This section present the methodology for selecting the clock frequency of each pipelined stage. We assume that the frequency adapter associated with the clock generator is capable of providing a set of N operating frequencies \mathcal{F} , with $|\mathcal{F}| = N$, that may be chosen for operating each functional block. The basic problem is to select the appropriate operating frequency for each functional block and the objectives are to ensure the stability of the packet queues at each block, while maximizing the energy savings.

Consider the operation of any functional block in the pipeline, say block k , with $1 \leq k \leq 5$ (we have five blocks in the pipeline). Let the available clock frequencies be arranged in decreasing order, f_1, f_2, \dots, f_N , with $f_1 > f_2 > \dots > f_N$. Let the rate of energy consumption of block k when frequency f_i , $1 \leq i \leq N$, is selected by denoted by P_i^k . The energy consumption of a device is usually modeled as [13]

$$P = \alpha C_{eff} V^2 f \quad (1)$$

where α is the switching factor, C_{eff} is the effective capacitance, V denotes the operating voltage and f is the operating frequency. Since changes in the operating frequency also affect the required clock frequency, we have [13]:

$$V^2 \propto f. \quad (2)$$

Thus we have

$$P \propto f^2. \quad (3)$$

Then, for operating frequency f_i , (1) can be written as

$$P_i = \eta f_i^2, \quad (4)$$

where η is a constant. Since the power consumption is directly related to the operating frequency, we have $P_1^k > P_2^k > \dots > P_N^k$, for all k . Define r_i^k to be the energy saved when frequency f_i is used instead to frequency f_1 :

$$r_i^k = P_1^k - P_i^k, \quad (5)$$

for $1 \leq i \leq N$. We have $r_i^k < r_2^k < \dots < r_N^k$.

We use a queue length based frequency selection mechanism for each functional block. Let the number of packets currently

TABLE II
AVAILABLE CLOCK FREQUENCIES

Index	Frequencies
1	100MHz
2	150MHz
3	200MHz
4	250MHz
5	300MHz

queued at the k -th functional block be Q_k . Then, the i that satisfies $(N - i)\xi < Q_k \leq (N - i + 1)\xi$, where ξ is a fixed positive integer, is used to select the clock frequency f_i to be used for the functional block. This frequency selection mechanism maps the queue length at a functional block to the set of clock frequencies in blocks of ξ packets. The mapping ensures that higher clock frequencies are selected when the queue length at the functional block increases. As a result, the workload is processed quickly, thereby reducing the queue lengths and moving the system towards stability. On the other hand, when the queue length is small, lower clock frequencies are selected. This results in greater energy savings without resulting in excessive queue lengths and packet delays. Finally, we note that ξ is a design parameter in the frequency selection mechanism. ξ may be selected as $\lfloor Q_k^{max}/N \rfloor$ where Q_k^{max} is the maximum queue size of functional block k .

IV. SIMULATION RESULTS

This section presents the simulation results to evaluate the proposed architecture. The proposed system is also compared with traditional IEEE 802.11 devices (no DFS and no pipelining) as well as systems with DFS but no pipelining. The simulations were conducted using the NS2 simulation software. The proposed pipelined system architecture was implemented in the NS2 platform and a set of five operating frequencies, as shown in Table II, are provided for selection by the functional blocks with DFS. For calculating the system level energy consumption, a model based on the equations in Section III-A was implemented.

We report results for three types of traffic: Constant bit rate (CBR), Poisson distributed traffic, and Pareto on-off traffic. The user datagram protocol (UDP) was used as the transport layer for our simulations. The duration of each simulation run was 100 seconds and the average of 10 runs is used for the results. The length of each packet was 1000 bytes. The simulations consider a two node scenario with one access point and one subscriber station.

We consider the following metrics for evaluating the performance of the proposed architecture:

- The average packet delay: The packet delay is calculated as the total time spent by a packet in the network interface card. This includes the time spent in the processing as well as channel contention and transmission. The results show the packet delays averaged over all packets transmitted by a node.
- The average energy consumed per packet: The average energy per packet is computed as the total energy consumed by a node divided by the number of packets handled by the node (including lost packets). To keep the results independent of the parameters such as device capacitance, we show the results for the energy consumption in terms of the average operating frequency. From Section III-A, the rate of energy consumption of the wireless network interface card is directly dependent on the operating frequency. Thus the reported results in terms of the operating frequency can be directly interpreted in terms of energy in Joules.
- The packet loss rate: The packet loss rate is defined as the number of packets lost divided by the total number of packets delivered to the MAC layer by the upper layer.

A. Average Energy Consumption

Figure 2 shows the average energy consumed the proposed system (labeled: pipelined DFS), a system with DFS but no pipelining (labeled: DFS), and a traditional system without pipelining or DFS (labeled: without DFS), for the three traffic types. While the systems with DFS can dynamically choose from the set of five clock frequencies shown in Table II, the traditional IEEE 802.11 system in our simulations (without DFS) used the middle frequency of 200 MHz. The results show that the energy consumption of the traditional system is independent of the offered load. On the other hand, the energy consumed by the systems with DFS is load-proportional. Also, we note that the proposed system has the lowest energy consumption.

The energy consumed by the system with only DFS and no pipelining increases beyond that of traditional systems at high loads. This is because at high loads, a system with DFS switches to higher clock frequencies. While this increases the energy consumption, it reduces the packet delays and losses, as can be seen from Figures 3 and 4. On the other hand, while the traditional system may have lower energy consumption at high loads, this benefit is offset by the larger packet delays and loss rates.

Figure 2 also shows that while the energy consumption of the proposed system increases with the load, it still stays below the traditional system and the system with only DFS. The key to this performance is the speedup in the packet processing rate provided by the pipelined architecture. As a result of the faster packet processing, the queue lengths at the functional blocks are smaller. As a result, the functional blocks choose lower clock frequencies for their operation, leading to a reduction in the energy consumption.

B. Average Packet Delay

Figure 3 shows the average packet delay for the three systems under the traffic models considered for our simulations. We observe that the proposed system leads to the lowest packet delays at all loads. The reduction in the packet delays is a direct consequence of the increased packet processing rate facilitated by the pipelined architecture.

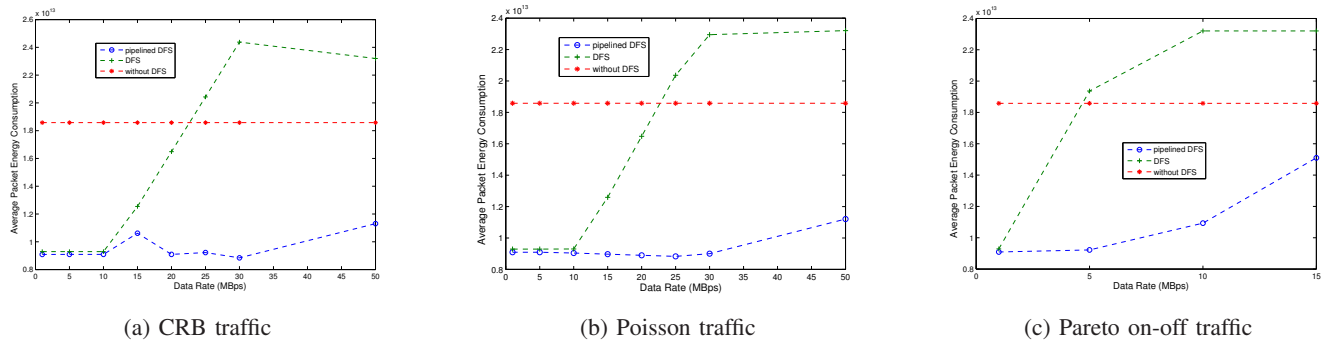


Fig. 2. Average energy consumed per packet.

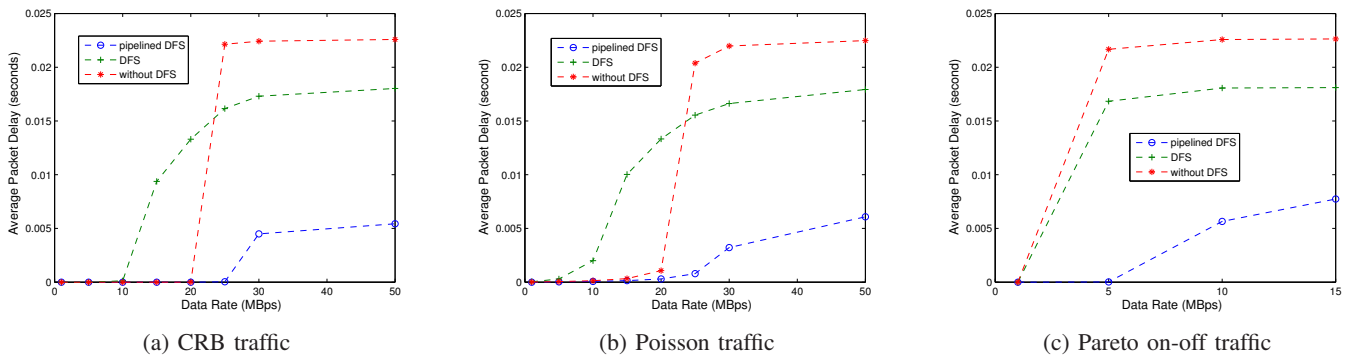


Fig. 3. Average delay per packet.

As the load starts to increase (data rates of 10 to 20 Mbps for the CBR and Poisson traffic), the system with only DFS has higher delays than the traditional system. This is because while the load is increasing, the queue lengths are still small enough for the DFS to select frequencies lower than 200 MHz (the frequency used by the traditional system). As a result the delays with DFS are higher. However, from Figure 2 we note that the energy consumed by DFS in this region is lower than that for traditional systems. As the arrival rates increase further, the delays with DFS are lower than that of the traditional system since higher frequencies are now used by the DFS.

C. Packet Loss Rate

Figure 4 shows the packet loss rates for the three systems. We observe that the traditional system has the highest loss rates. This is because its clock rate stays constant and does not adapt to the large queue lengths as the traffic rate increases. On the other hand, the two systems with DFS select their operating frequencies by looking at the queue lengths. As a result, when the queue lengths increase, higher operating frequencies are chosen which serves to increase the rate at which packets are processed and thereby reduce the queue lengths. Finally, we note that the proposed system has the lowest loss rates due to its joint use of DFS and pipelining which ensures the fastest processing of packets.

Finally, we note that the simulations in this paper used the middle frequency of 200 MHz from Table II for the traditional system. The trends in results stay the same irrespective of

the choice of the frequency. In case a lower frequency is chosen, the traditional system would lead to a lower energy consumption as compared to that shown in Figure 2. However, the corresponding packet delays and loss rates will increase beyond those in Figures 3 and 4. Similarly, the use of a higher frequency would increase the energy consumption but lead to lower delays and loss rates. The proposed system provides the best mix of energy, delay and loss performance while providing load-proportional energy usage.

V. CONCLUSIONS

This paper presented an architecture for wireless network interface cards that provide load-proportional energy usage while addressing the energy-delay tradeoff. To reduce the delays associated with the use of DVFS, we proposed an architecture where the functional blocks inside the interface cards are pipelined. In addition, DFS is used in each block to provide load-proportional energy usage. The performance of the proposed system has been evaluated using simulations and it has been shown to outperform existing systems in terms of the energy consumption, packet delays and loss rates.

REFERENCES

- [1] B. Sikdar, "A study of the environmental impact of wired and wireless local area network access," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, February 2013.
- [2] O. Arnold, F. Richter, G. Fettweis and O. Blume, "Power consumption modeling of different base station types in heterogeneous cellular networks," *Proceedings of Future Network and Mobile Summit*, Florence, Italy, June 2010.

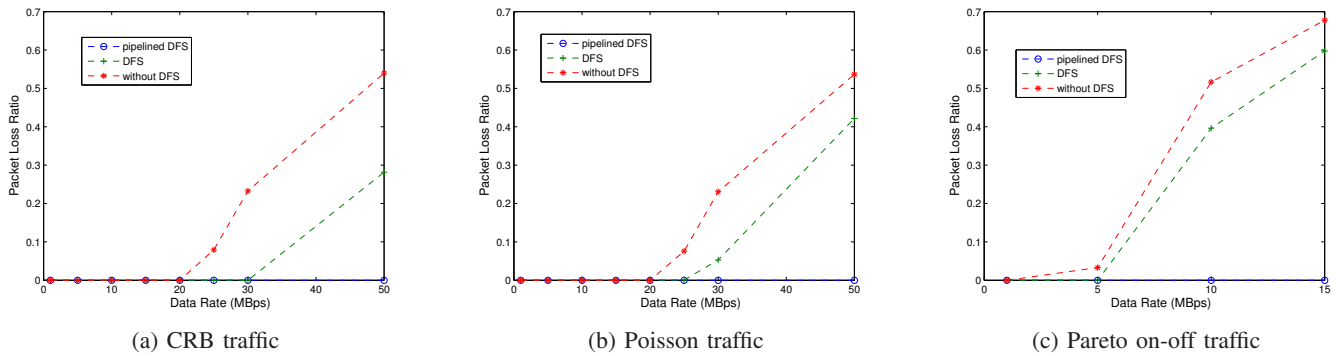


Fig. 4. Packet loss rate.

- [3] H-S. Jung, A. Hwang and M. Pedram, "Predictive-flow-queue based energy optimization for gigabit Ethernet controllers," *IEEE Transactions on VLSI Systems*, vol. 17, no. 8, pp. 1113-1126, August 2009.
- [4] J. Chen and B. Sikdar, "A Mechanism for Load Proportional Energy Use in Wireless Local Area Networks," *Proceedings of IEEE GLOBECOM*, Atlanta, GA, December 2013.
- [5] K. Choi, R. Soma and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," *Proceedings of Symposium on Low Power Electronics and Design*, pp. 174-179, Newport Beach, CA, August 2004.
- [6] M. E. Salehi, M. Samadi, A. Afzali-Kusha, M. Pedram and S. M. Fakhraie, "Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs," *IEEE Transactions on VLSI Systems*, vol. 19, no.10 pp. 1931-1935, October 2011.
- [7] M. Lim, and V. Freeh, "Determining the Minimum Energy Consumption using Dynamic Voltage and Frequency Scaling," *Proceedings of IEEE IPDPS*, pp. 1-8, Long Beach, CA, March 2007.
- [8] M. Gupta, S. Grover and S. Singh, "A feasibility study for power management in LAN switches," *Proceedings of ICNP*, pp. 361-371, Berlin, Germany, October 2004.
- [9] L. Shang, L-S. Peh and N. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," *Proceedings of Symposium on High-Performance Computer Architecture*, pp. 91-102, Anaheim, CA, February 2003.
- [10] A. Mishra, R. Das, S. Eachempati, K. Ravishankar, N. Vijaykrishnan and C. Das, "A case for dynamic frequency tuning in on-chip networks," *Proceedings of IEEE/ACM International Symposium on Microarchitecture*, pp 292-303, New York, NY, December 2009.
- [11] R. Bolla, R. Bruschi and C. Lombardo, "Dynamic Voltage and Frequency Scaling in Parallel Network Processors," *Proceedings of IEEE HPSR*, pp. 242-249, Belgrade, Serbia, June 2012.
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE standards 802.11, January 1997.
- [13] J. Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.
- [14] J.Chen and B.Sikdar, "A Mechanism for Load Proportional Energy Use in Wireless Local Area Networks" *IEEE GlobeCom 2013*
- [15] J. Lee, S.Yoon, K. Pyun and S. Park, "A Multi-Processor NoC Platform Applied on the 802.11i TKIP Cryptosystem" *Proceedings of the IEEE Asia and South Pacific Design Automation Conference* pp. 607-610, June 2008.
- [16] L. Fang, H. Wu and Z. Huang, "Performance Modeling and Analysis of IEEE 802.11 Protocol using POOSL" *Proceedings of IEEE ICIEA 2009*, pp.1330-1335, 2009.
- [17] Y. Tang, L. Qian and Y. Wang, "Optimized Software Implementation of a Full-Rate IEEE 802.11a Compliant Digital Baseband Transmitter on a Digital Signal Processor" *Proceedings of IEEE GLOBECOM*, St. Louis, MO, December 2005.