An MDP-Based Dynamic Optimization Methodology for Wireless Sensor Networks

Arslan Munir, *Student Member*, *IEEE*, and Ann Gordon-Ross, *Member*, *IEEE*

Abstract—Wireless sensor networks (WSNs) are distributed systems that have proliferated across diverse application domains (e.g., security/defense, health care, etc.). One commonality across all WSN domains is the need to meet application requirements (i.e., lifetime, responsiveness, etc.) through domain specific sensor node design. Techniques such as sensor node parameter tuning enable WSN designers to specialize tunable parameters (i.e., processor voltage and frequency, sensing frequency, etc.) to meet these application requirements. However, given WSN domain diversity, varying environmental situations (stimuli), and sensor node complexity, sensor node parameter tuning is a very challenging task. In this paper, we propose an automated Markov Decision Process (MDP)-based methodology to prescribe optimal sensor node operation (selection of values for tunable parameters such as processor voltage, processor frequency, and sensing frequency) to meet application requirements and adapt to changing environmental stimuli. Numerical results confirm the optimality of our proposed methodology and reveal that our methodology more closely meets application requirements compared to other feasible policies.

Index Terms—Wireless sensor networks, dynamic optimization, MDP.

1 Introduction and Motivation

Wireless sensor networks are distributed systems consisting of spatially distributed autonomous sensor nodes that span diverse application domains (e.g., security and defense, industrial automation, and logistics, etc.). However, this wide application diversity combined with increasing complexity, functionality requirements, and highly constrained operating environments make WSN design very challenging.

One critical WSN design challenge involves meeting *application requirements* such as reliability, lifetime, throughput, delay (responsiveness), etc., for myriad of application domains. For example, a vineyard irrigation system may require less responsiveness to environmental stimuli (i.e., decreased irrigation during wet periods), but have a long lifetime requirement. On the other hand, in a disaster relief application, sensor nodes may require high responsiveness but have a short lifetime. Additional requirements may include high adaptability to rapid network changes as sensor nodes are destroyed. Meeting these application specific requirements is critical to accomplishing the application's assigned function. Nevertheless, satisfying these demands in a scalable and cost-effective way is a challenging task.

Commercial off-the-shelf (COTS) sensor nodes have difficulty meeting application requirements due to inherent manufacturing traits. In order to reduce manufacturing costs, generic COTS sensor nodes capable of implementing nearly any application are produced in large volumes, and are not specialized to meet any specific application requirements. In order to meet application requirements, sensor nodes must possess tunable parameters. Fortunately, some COTS have *tunable parameters* such as processor voltage, processor frequency, sensing frequency, radio transmission power, and radio transmission frequency, etc.

Sensor node *parameter tuning* is the process of determining appropriate *parameter values* which meet application requirements. However, determining such values presents several tuning challenges. First, *application managers* (the individuals responsible for WSN deployment and management) typically lack sufficient technical expertise [1], [2], as many managers are nonexperts (i.e., biologists, teachers, structural engineers, agriculturists, etc.). In addition, parameter value tuning is still a cumbersome and time-consuming task even for expert application managers due to unpredictable WSN environments and difficulty in creating accurate simulation environments. Second, selected parameter values may not be optimal. Given a highly configurable sensor node with many tunable parameters and with many values for each tunable parameter, choosing the optimal combination is difficult. In addition, unanticipated changes in the sensor node's environment can alter optimal parameter values. For example, a sensor node designed to monitor a short-lived volcanic eruption may need to operate for more months/years than expected if earthquakes alter magma flow.

To ease parameter value selection, *dynamic optimizations* enable sensor nodes to dynamically tune their parameter values in situ according to application requirements and environmental stimuli. This dynamic tuning of parameters ensures that a WSN performs the assigned task optimally, enabling the sensor node to constantly conform to the changing environment. Besides, the application manager

- *A. Munir is with the Department of Electrical and Computer Engineering, University of Florida, 113 Larsen Hall, Gainesville, FL 32611. E-mail: amunir@ufl.edu.*
- *A. Gordon-Ross is with the Department of Electrical and Computer Engineering and also with the US National Science Foundation (NSF) Center for High-Performance Reconfigurable Computing (CHREC), University of Florida, 221 Larsen Hall, Gainesville, FL 32611. E-mail: ann@ece.ufl.edu.*

Manuscript received 5 Dec. 2010; revised 3 May 2011; accepted 15 July 2011; published online 19 July 2011.
Recommended for acceptance by V. Misic.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-12-0707.
Digital Object Identifier no. 10.1109/TPDS.2011.208.

need not know sensor node and/or dynamic optimization specifics, thus easing parameter tuning for nonexpert application managers.

There is a lot of research in the area of dynamic optimizations [3], [4], [5], [6], but, however, most previous work focuses on the processor or memory (cache) in computer systems. Whereas these endeavors can provide valuable insights into WSN dynamic optimizations, they are not directly applicable to WSNs due to different design spaces, platform particulars, and a sensor node's tight design constraints.[1]

In this paper, we propose an application-oriented dynamic tuning methodology for WSNs based on Markov Decision Process (MDP). Our MDP-based application-oriented tuning methodology performs dynamic voltage, frequency, and sensing (sampling) frequency scaling (DVFS2). MDP is an appropriate candidate for WSN dynamic optimizations where dynamic decision making is a requirement in light of changing environmental stimuli and wireless channel condition. We focus on DVFS2 for several reasons. Traditional microprocessor-based systems use dynamic voltage and frequency scaling (DVFS) for energy optimizations. However, sensor nodes are distinct from traditional systems in that they have embedded sensors coupled with an embedded processor. Therefore, DVFS only provides a partial tuning methodology and does not consider sensing frequency. Sensing frequency tuning is essential for sensor nodes to meet application requirements because the sensed data delay (the delay between the sensor sensing the data and the data's reception by the application manager) depends upon the sensor node sensing frequency as it influences the amount of processed and communicated data. Thus, DVFS2 provides enhanced optimization potential as compared to DVFS with respect to WSNs.

Our main contributions in this paper are

- To the best of our knowledge, we propose for the first time a Markov Decision Process for WSN dynamic optimization. MDP is suitable for WSN dynamic optimization because of MDP's inherent ability to perform dynamic decision making. This paper presents a first step toward MDP-based dynamic optimization.
- Our MDP-based dynamic optimization methodology gives an optimal policy that performs DVFS2 and specifies optimal sensor node parameters for WSN lifetime.
- Our MDP-based dynamic tuning methodology is optimal in any given situation.
- Our MDP-based dynamic tuning methodology can adapt to changing application requirements and environmental stimuli.
- We provide implementation guidelines for our proposed dynamic tuning methodology in sensor nodes.

We compare our proposed MDP-based application oriented dynamic tuning methodology with several fixed heuristics. The results show that our proposed methodology outperforms other heuristics for given application requirements.

1. Additional related work in the literature is presented in Section 2 of the supplementary material document posted online.

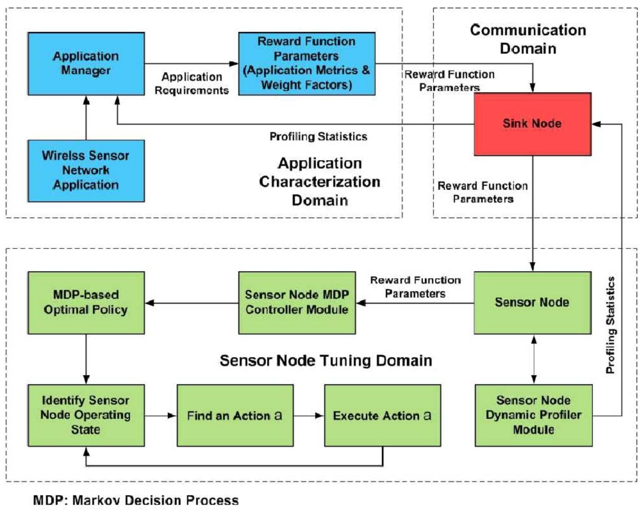
Application Manager
Application Requirements
Reward Function Parameters (Application Metrics & Weight Factors)
Reward Function Parameters
Communication Domain
Sink Node
Profiling Statistics
Wireless Sensor Network Application
Application Characterization Domain
Reward Function Parameters
MDP-based Optimal Policy
Sensor Node MDP Controller Module
Reward Function Parameters
Sensor Node
Profiling Statistics
Sensor Node Tuning Domain
Identify Sensor Node Operating State
Find an Action a
Execute Action a
Sensor Node Dynamic Profiler Module
MDP: Markov Decision Process

Fig. 1. Process diagram for our MDP-based dynamic optimization methodology for wireless sensor networks.

The broader impacts of our research includes facilitating WSN designers (persons who design WSNs for an application) to better meet application requirements by selecting optimal tunable parameter values for each sensor node. As this paper presents a first step toward MDP-based dynamic optimization, our work can potentially spark further research in MDP-based optimizations for WSNs.

2 MDP-Based Tuning Overview

In this section, we present our MDP-based tuning methodology along with an MDP overview with respect to WSNs [7].

2.1 MDP-Based Dynamic Optimization Methodology for Wireless Sensor Networks

Fig. 1 depicts the process diagram for our MDP-based dynamic optimization methodology. Our methodology consists of three logical domains: the application characterization domain, the communication domain, and the sensor node tuning domain.

The *application characterization domain* refers to the WSN application's characterization/specification. In this domain, the application manager defines various *application metrics* (e.g., tolerable power consumption, tolerable delay, etc.), which are calculated from (or based on) application requirements. The application manager also assigns *weight factors* to application metrics to signify the weightage or importance of each application metric with respect to each other. Weight factors provide application managers with an easy method to relate the relative importance of each application metric. The application manager defines an MDP *reward function* which signifies the overall reward (revenue) for given application requirements. The application metrics along with associated weight factors, represent the MDP *reward function parameters*.

The *communication domain* contains the sink node (which gathers statistics from the sensor nodes) and encompasses the communication network between the application manager and the sensor nodes. The application manager transmits the MPD reward function parameters to the sink node via the communication domain. The sink node in turn relays reward function parameters to the sensor nodes.

The *sensor node tuning domain* consists of sensor nodes and performs sensor node tuning (determination of optimal parameter values for the sensor node to meet application requirements). Each sensor node contains an *MDP controller module* which implements our MDP-based dynamic optimization methodology. After a sensor node receives reward function parameters from the sink node through the communication domain, the sensor node invokes the MDP controller module. The MDP controller module calculates the *MDP-based optimal policy*. The MDP-based optimal policy prescribes the optimal sensor node *actions* to meet application requirements over the lifetime of the sensor node. An action prescribes the sensor node *state* (defined by processor voltage, processor frequency, and sensing frequency) in which to transition from the current state. The sensor node identifies its current operating state, determines an action "a" prescribed by the MDP-based optimal policy (i.e., whether to continue operation in the current state or transition to another state) and subsequently executes action "a."

Our proposed MDP-based dynamic tuning methodology can adapt to changes in application requirements (since application requirements may change with time, e.g., a defense system initially deployed to monitor enemy troop position for four months may later be required to monitor troop activity for an extended period of six months). Whenever application requirements change, the application manager updates the reward function (and/or associated parameters) to reflect the new application requirements. Upon receiving the updated reward function, the sensor node reinvokes MDP controller module and determines the new MDP-based policy to optimally meet the new application requirements.

Our MDP-based dynamic optimization methodology reacts to environmental stimuli via a *dynamic profiler module* in the sensor node tuning domain. The dynamic profiler module monitors environmental changes over time and captures unanticipated environmental situations not predictable at design time [8]. The dynamic profiler module may be connected to the sensor node and profiles the *profiling statistics* (e.g., wireless channel condition, number of packets dropped, packet size, radio transmission power, etc.) when triggered by the WSN application. The dynamic profiler module informs the application manager of the profiled statistics via the communication domain. After receiving the profiling statistics, the application manager evaluates the statistics and possibly updates the reward function parameters. This reevaluation process may be automated, thus eliminating the need for continuous application manager input. Based on these received profiling statistics and updated reward function parameters, the sensor node MDP controller module determines whether application requirements are met or not met. If application requirements are not met, the MDP controller module reinvokes the MDP-based optimal policy to determine a new operating state to better meet application requirements. This feedback process continues to ensure that the application requirements are met in the presence of changing environmental stimuli.

2.2 MDP Overview with Respect to Wireless Sensor Networks

In this section, we define basic MDP terminology in the context of WSNs and give an overview of our proposed MDP-based dynamic optimization methodology for sensor nodes. MDPs, also known as stochastic dynamic programming, are used to model and solve dynamic decision making problems. We use standard notations as defined in [9] for our MDP-based problem formulation.[2]

The basic elements of an MDP model are: *decision epochs and periods*, *states*, *action sets*, *transition probabilities*, and *rewards*. An MDP is *Markovian* (memoryless) because the transition probabilities and rewards depend on the past only through the current state and the action selected by the decision maker in that state.

Decision epochs. The *decision epochs* refer to the points of time during a sensor node's lifetime at which the sensor node makes a decision. Specifically, a sensor node makes a decision regarding its operating state at these decision epochs, i.e., whether to continue operating at the current state (processor voltage, frequency, and sensing frequency), or transition to another state. We consider a discrete time process where time is divided into *periods* and a decision epoch corresponds to the beginning of a period. The set of decision epochs can be denoted as $T = \{1, 2, 3, \ldots, N\}$, where $N \leq \infty$ and denotes the sensor node's lifetime (each individual time period in T can be denoted as time t). The *decision problem* is referred to as a *finite horizon* problem when the decision making horizon N is finite and *infinite horizon* otherwise. In a finite horizon problem, the final decision is made at decision epoch $N - 1$; hence, the finite horizon problem is also known as the $N - 1$ period problem.

State space and action set. The system (a sensor node) operates in a particular *state* at each decision epoch, where S denotes the complete set of possible system states (i.e., state space). States specify particular sensor node parameter values and each state represents a different combination of these values.

The state space for our MDP-based tuning methodology is a composite state space containing the Cartesian product of sensor node tunable parameters' state spaces. We define the state space S as

$$S = S_1 \times S_2 \times \cdots \times S_M \; : \; |S| = I, \tag{1}$$

where $\times$ denotes the Cartesian product, M is the total number of sensor node tunable parameters, S_k denotes the state space for tunable parameter k where $k \in \{1, 2, \ldots, M\}$, and $|S|$ denotes the state space S cardinality (the number of states in S).

An *action set* represents all allowable actions in all possible states. At each decision epoch, the sensor node decides whether to continue operating in the current state or to switch to another state. The sensor node state (in our problem) represents a tuple consisting of processor voltage (V_p), processor frequency (F_p), and sensing frequency (F_s). If the system is in state $s \in S$ at a decision epoch, the sensor node can choose an action a from the set of allowable actions A_s in state s. Thus, an action set can be written as

2. Section 3 of the supplementary material document presents detailed mathematical formulation of our MDP-based dynamic optimization.

$A = \bigcup_{s \in S} A_s$. We assume that S and A_s do not vary with time t [9].

Decision rule. A *decision rule* prescribes an action in each state at a specified decision epoch. Our decision rule for sensor nodes is a function $d_t : S \rightarrow A_s$ which specifies the action at time t when the system is in state s for each $s \in S$, $d_t(s) \in A_s$. This decision rule is both *Markovian* and *deterministic*.

State dynamics. When a sensor node selects action $a \in A_s$ in state s, the sensor node receives a *reward* $r_t(s, a)$ and the *transition probability distribution* $p_t(\cdot|s, a)$ determines the system state at the next decision epoch. The real-valued function $r_t(s, a)$ denotes the value of the reward received at time t in period t. The reward is referred to as income or cost depending on whether or not $r_t(s, a)$ is positive or negative, respectively. When the reward depends on the system state at the next decision epoch, we let $r_t(s, a, j)$ denote the value of the reward received at time t when the system state at decision epoch t is s. The sensor node selects action $a \in A_s$, and the system occupies the state j at decision epoch $t+1$. The sensor node evaluates $r_t(s, a)$ using [9]

$$r_t(s, a) = \sum_{j \in S} r_t(s, a, j) p_t(j|s, a), \tag{2}$$

where the nonnegative function $p_t(j|s, a)$ is called a *transition probability function* which governs the state dynamics. $p_t(j|s, a)$ denotes the probability that the system occupies state $j \in S$ at time $t+1$ when the sensor node selects action $a \in A_s$ in state s at time t and usually $\sum_{j \in S} p_t(j|s, a) = 1$. Formally, an MDP is defined as the collection of objects $\{T, S, A_s, p_t(\cdot|s, a), r_t(s, a)\}$.

Policy. A *policy* specifies the decision rule for all decision epochs. In the case of sensor nodes, the policy prescribes action selection under any possible system state. A policy π is a sequence of decision rules, i.e., $\pi = (d_1, d_2, d_3, \ldots, d_{N-1})$ for $N \leq \infty$. A policy is *stationary* if $d_t = d \ \forall\, t \in T$, i.e., for stationary policy $\pi = (d, d, d, \ldots, d)$.

Reward. As a result of selecting and implementing a particular policy, the sensor node receives rewards at time periods $\{1, 2, 3, \ldots, N\}$. The reward sequence is random, because the rewards received in different periods are not known prior to policy implementation. The sensor node's optimization objective is to determine a policy which maximizes the corresponding random reward sequence.

As a result of selecting an action, the sensor node receives a reward $r(X_t, Y_t)$ at time t. The *expected total reward* denotes the expected total reward over the decision making horizon given a specific policy. Let $\upsilon^{\pi}(s)$ denote the expected total reward over the decision making horizon when the horizon length N is a random variable, the system is in state s at the first decision epoch, and policy π is used [10], [9]

$$\upsilon^{\pi}(s) = E_s^{\pi}\left[E_N\left\{\sum_{t=1}^{N} r(X_t, Y_t)\right\}\right], \tag{3}$$

where E_s^{π} represents the expected reward with respect to policy π and the initial state s (the system state at the time of the expected reward calculation), and E_N denotes the expected reward with respect to the probability distribution of the random variable N. We can write (3) as [9]

$$\upsilon^{\pi}(s) = E_s^{\pi}\left\{\sum_{t=1}^{\infty} \lambda^{t-1} r(X_t, Y_t)\right\}, \tag{4}$$

which gives the *expected total discounted reward*. We assume that the random variable N is geometrically distributed with parameter λ and hence the distribution *mean* is $1/(1-\lambda)$ [10]. The parameter λ can be interpreted as a *discount factor*, which measures the present value of one unit of reward received one period in the future. Thus, $\upsilon^{\pi}(s)$ represents the expected total present value of the reward (income) stream obtained using policy π [9].

The *reward function* captures application metrics and sensor node characteristics. Our reward function characterization considers the power consumption (which affects the sensor node lifetime), throughput, and delay application metrics. We define the reward function $f(s, a)$ given the current sensor node state s and the sensor node's selected action a as

$$f(s, a) = \omega_p f_p(s, a) + \omega_t f_t(s, a) + \omega_d f_d(s, a), \tag{5}$$

where $f_p(s, a)$ denotes the power reward function, $f_t(s, a)$ denotes the throughput reward function, and $f_d(s, a)$ denotes the delay reward function; ω_p, ω_t, and ω_d represent the *weight factors* for power, throughput, and delay, respectively. The weight factors' constraints are given as $\sum_m \omega_m = 1$ where $m = \{p, t, d\}$ such that $0 \leq \omega_p \leq 1$, $0 \leq \omega_t \leq 1$, and $0 \leq \omega_d \leq 1$. The weight factors are selected based on the relative importance of application metrics with respect to each other.

We define linear reward functions for application metrics because an application metric reward (objective function) typically varies linearly, or piecewise linearly, between the minimum and the maximum allowed values of the metric [10], [11]. We define the power reward function in (5) as

$$f_p(s, a) = \begin{cases} 1, & 0 < p_a \leq L_P \\ (U_P - p_a)/(U_P - L_P), & L_P < p_a < U_P \\ 0, & p_a \geq U_P, \end{cases} \tag{6}$$

where p_a denotes the power consumption of the current state given action a taken at time t and the constant parameters L_P and U_P denote the minimum and maximum allowed/tolerated sensor node power consumption, respectively.[3]

Optimality equation. The optimality equation, also known as Bellman's equation, for expected total discounted reward criterion is given as [9]

$$\upsilon(s) = \max_{a \in A_s}\left\{r(s, a) + \sum_{j \in S} \lambda p(j|s, a) \upsilon(j)\right\}, \tag{7}$$

where $\upsilon(s)$ denotes the maximum expected total discounted reward.

3 Numerical Results

In this section, we compare the performance (based on the expected total discounted reward criterion (4)) of our proposed MDP-based DVFS2 optimal policy π^* (π^{MDP})

3. Section 3.5 of the supplementary material document presents detailed formulation and corresponding figures for our reward functions.

TABLE 1
Parameters for Wireless Sensor Node State $s_i = [V_p, F_p, F_s]$ (V_p is specified in volts, F_p in MHz, and F_s in KHz)

Notation	Parameter Description	$s_1 = [2.7, 2, 2]$	$s_2 = [3, 4, 4]$	$s_3 = [4, 6, 6]$	$s_4 = [5.5, 8, 8]$
p_i	power consumption in state s_i	10 units	15 units	30 units	55 units
t_i	throughput in state s_i	4 units	8 units	12 units	16 units
d_i	delay in state s_i	26 units	14 units	8 units	6 units

Parameters are specified as a multiple of a base unit where one power unit is equal to 1 mW, one throughput unit is equal to 0.5 MIPS, and one delay unit is equal to 50 ms. Parameter values are based on the XSM mote.

with several fixed heuristic policies using a representative WSN platform. We use the Matlab MDP tool box [12] implementation of our policy iteration algorithm [9] to solve Bellman's equation (7) to determine the MDP-based optimal policy. We select sensor node state parameters based on eXtreme Scale Motes (XSM) [13], [14]. The XSM motes have an average lifetime of 1,000 hours of continuous operation with two AA alkaline batteries, which can deliver 6 Whr or an average of 6 mW [13]. The XSM platform integrates an Atmel ATmega128L microcontroller [15], a Chipcon CC1000 radio operating at 433 MHz, and a 4 Mbit serial flash memory. The XSM motes contain infrared, magnetic, acoustic, photo, and temperature sensors.

To represent sensor node operation, we analyze sample application domains that represent a typical security system or defense application (henceforth referred to as a *security/defense system*) [7], health care application, and ambient conditions monitoring application.[4] For brevity, we select a single sample WSN platform configuration and several application domains, but we point out that our proposed MDP model and methodology works equally well for any other WSN platform and application.

For each application domain, we evaluate the effects of different discount factors, different state transition costs, and different application metric weight factors on the expected total discounted reward for our MDP-based optimal policy and several fixed heuristic policies (Section 3.1). The magnitude of difference in the total expected discounted reward for different policies is important as it provides relative comparisons between the different policies.

3.1 Fixed Heuristic Policies for Performance Comparisons

Due to the infancy of WSN dynamic optimizations, there exist no dynamic sensor node tuning methods for comparison with our MDP-based policy. Therefore, we compare to several fixed heuristic policies (heuristic policies have been shown to be a viable comparison method [10]). To provide a consistent comparison, fixed heuristic policies use the same reward function and associated parameter settings as that of our MDP-based policy. We consider the following four fixed heuristic policies:

- A fixed heuristic policy π^{POW} which always selects the state with the lowest power consumption.
- A fixed heuristic policy π^{THP} which always selects the state with the highest throughput.
- A fixed heuristic policy π^{EQU} which spends an equal amount of time in each of the available states.
- A fixed heuristic policy π^{PRF} which spends an unequal amount of time in each of the available states based on a specified preference for each state. For example, given a system with four possible states, the π^{PRF} policy may spend 40 percent of time in the first state, 20 percent of time in the second state, 10 percent of time in the third state, and 30 percent of time in the fourth state.

4. Numerical results for an ambient conditions monitoring application are presented in Section 6.1 of the supplementary material document posted online.

3.2 MDP Specifications

We compare different policies using the *expected total discounted reward* performance criterion (7). The state transition probability for each sensor node state is governed by state dynamics (Section 2.2). The sensor node's lifetime and the time between decision epochs are subjective and may be assigned by an application manager according to application requirements. A sensor node's *mean lifetime* is given by $1/(1-\lambda)$ *time units*, which is the time between successive decision epochs (which we assume to be 1 hour). For instance for $\lambda = 0.999$, the sensor node's mean lifetime is $1/(1-0.999) = 1{,}000$ hours ≈ 42 days.

For our numerical results, we consider a sensor node capable of operating in four different states, i.e., $I = 4$ in (1). Table 1 summarizes state parameter values for each of the four states s_1, s_2, s_3, and s_4. We define each state using a $[V_p, F_p, F_s]$ tuple where V_p is specified in volts, F_p in MHz, and F_s in KHz. For instance, state one s_1 is defined as $[2.7, 2, 2]$, which corresponds to a processor voltage of 2.7 volts, a processor frequency of 2 MHz, and a sensing frequency of 2 KHz (2,000 samples per second). We represent state $s_i \; \forall \; i \in \{1, 2, 3, \ldots, I\}$ power consumption, throughput and delay as multiples of power, throughput, and delay base units, respectively. We assume one base power unit is equal to 1 mW, one base throughput unit is equal to 0.5 Millions of Instructions per Second (MIPS), and one base delay unit is equal to 50 ms. We assign base units such that these units provide a convenient representation of application metrics (power, throughput, delay). We point out, however, any other feasible base unit values can be assigned [10]. We assume, without loss of generality, that the transition cost for switching from one state to another is $H_{i,a} = 0.1$ if $i \neq a$. The transition cost could be a function of power, throughput, and delay but we assume a constant transition cost for simplicity as it is typically constant for different state transitions [15].

Our selection of the state parameter values in Table 1 corresponds to XSM mote specifications [13], [15]. The XSM mote's Atmel ATmega128L microprocessor has an operating voltage range of 2.7 to 5.5 V and a processor frequency

TABLE 2
Minimum L and Maximum U Reward Function Parameter Values and Application Metric Weight Factors for a Security/Defense System, Health Care, and Ambient Conditions Monitoring Application

Notation	Parameter Description	Security/Defense	Health Care	Ambient Monitoring
L_P	minimum acceptable power consumption	12 units	8 units	5 units
U_P	maximum acceptable power consumption	35 units	20 units	32 units
L_T	minimum acceptable throughput	6 units	3 units	2 unit
U_T	maximum acceptable throughput	12 units	9 units	8 units
L_D	minimum acceptable delay	7 units	8 units	12 units
U_D	maximum acceptable delay	16 units	20 units	40 units
ω_p	power weight factor	0.45	0.5	0.65
ω_t	throughput weight factor	0.2	0.3	0.15
ω_d	delay weight factor	0.35	0.2	0.2

range of 0 to 8 MHz. The ATmega128L throughput varies with processor frequency at 1 MIPS per MHz, thus allowing an application manager to optimize power consumption versus processing speed [15]. Our chosen sensing frequency also corresponds with standard sensor node specifications. The Honeywell HMC1002 magnetometer sensor [16] consumes on average 15 mW of power and can be sampled in 0.1 ms on the Atmel ATmega128L microprocessor, which results in a maximum sampling frequency of approximately 10 KHz (10,000 samples per second). The acoustic sensor embedded in the XSM mote has a maximum sensing frequency of approximately 8.192 KHz [13]. Although the power consumption in a state depends upon not only the processor voltage and frequency but also on the processor utilization, which also depends upon sensing frequency, we report the average power consumption values in a state as derived from the data sheets [15], [16].

Table 2 summarizes the minimum L and maximum U reward function parameter values for application metrics (power, throughput, and delay) and associated weight factors for a security/defense system, health care, and ambient conditions monitoring application. Our selected reward function parameter values represent typical application requirements [17]. We describe below the relative importance of these application metrics with respect to our considered applications.

Although power is a primary concern for all WSN applications and tolerable power consumption values are specified based on the desired WSN lifetime considering limited battery resources of sensor nodes. However, a relative importance in power for different applications can be delineated mainly by the infeasibility of sensor node's battery replacement due to hostile environments. For example, sensor nodes in a war zone for a security/defense application and an active volcano monitoring application makes battery replacement almost impractical. For a health care application with sensors attached to a patient to monitor physiological data (e.g., heart rate, glucose level, etc.), sensor node's battery may be replaced though power is constrained because excessive heat dissipation could adversely affect a patient's health. Similarly, delay can be an important factor for security/defense in case of enemy target tracking and health care for a patient in intensive health conditions whereas delay may be relatively less important for a humidity monitoring application. A data sensitive security/defense system may require a comparatively large minimum throughput in order to obtain a sufficient number of sensed data samples for meaningful analysis. Although relative importance and minimum and maximum values of these application metrics can vary widely with an application domain and between application domains, we pick our parameter values (Table 2) for demonstration purposes to provide an insight into our optimization methodology.

We point out that all of our considered application metrics specifically throughput depends upon the traffic pattern. WSN throughput is a complex function of the number of nodes, traffic volume and patterns, and the parameters of the medium access technique. As the number of nodes and traffic volume increases, contention-based medium access methods result in an increased number of packet collisions which waste energy without transmitting useful data. This contention and packet collision results in saturation which decreases the effective throughput and increases the delay sharply. We briefly outline the variance in WSN traffic patterns for our considered applications. The security/defense application would have infrequent bursts of heavy traffic (e.g., when an enemy target appears within the sensor nodes' sensing range), health care applications would have a steady flow of medium to high traffic, and ambient conditions monitoring applications would have a steady flow of low to medium traffic except for emergencies (e.g., volcano eruption). Although modeling of application metrics with respect to traffic patterns would result in a better characterization of these metric values at particular instants/times in WSN, however, these metric values can still be bounded by a lower minimum and upper maximum value as captured by our reward functions (Section 2.2).

Given the reward function, sensor node state parameters corresponding to XSM mote, and transition probabilities, our Matlab MDP tool box [12] implementation of policy iteration algorithm solves Bellman's equation (7) to determine the MDP-based optimal policy and determines the expected total discounted reward (4).

3.3 Results for a Security/Defense System Application

3.3.1 *The Effects of Different Discount Factors on the Expected Total Discounted Reward*

Table 3 and Fig. 2 depict the effects of different discount factors λ on the heuristic policies and π^{MDP} for a

TABLE 3
The Effects of Different Discount Factors for a Security/Defense System

Discount Factor λ	Sensor Lifetime	π^{MDP}	π^{POW}	π^{THP}	π^{EQU}	π^{PRF}
0.94	16.67 hours	10.0006	7.5111	9.0778	7.2692	7.5586
0.95	20 hours	12.0302	9.0111	10.9111	8.723	9.0687
0.96	25 hours	15.0747	11.2611	13.6611	10.9038	11.3339
0.97	33.33 hours	20.1489	15.0111	18.2445	14.5383	15.1091
0.98	50 hours	30.2972	22.5111	27.4111	21.8075	22.6596
0.99	100 hours	60.7422	45.0111	54.9111	43.6150	45.3111
0.999	1000 hours	608.7522	450.0111	549.9111	436.15	453.0381
0.9999	10,000 hours	6.0889×10^3	4.5×10^3	5.4999×10^3	4.3615×10^3	4.5303×10^3
0.99999	100,000 hours	6.089×10^4	4.5×10^4	5.5×10^4	4.3615×10^4	4.5303×10^4

$H_{i,j} = 0.1$ *if* $i \neq j$, $\omega_p = 0.45, \omega_t = 0.2, \omega_d = 0.35$.

security/defense system when the state transition cost $H_{i,j}$ is held constant at 0.1 for $i \neq j$, and ω_p, ω_t, and ω_d are equal to 0.45, 0.2, and 0.35, respectively. Since we assume the time between successive decision epochs to be 1 hour, the range of λ from 0.94 to 0.99999 corresponds to a range of average sensor node lifetime from 16.67 to 100,000 hours $\approx 4{,}167$ days ≈ 11.4 years. Table 3 and Fig. 2 show that π^{MDP} results in the highest expected total discounted reward for all values of λ and corresponding average sensor node lifetimes.

We calculate the percentage improvement in expected total discounted reward for π^{MDP} for a security/defense system as compared to the fixed heuristic policies as $[(R^{MDP} - R^X)/R^{MDP}] \times 100$ where R^{MDP} denotes the expected total discounted reward for π^{MDP} and R^X denotes the expected total discounted reward for the X fixed heuristic policy where $X = \{\text{POW}, \text{THP}, \text{EQU}, \text{PRF}\}$. For instance, when the average sensor node lifetime is 1,000 hours ($\lambda = 0.999$), π^{MDP} results in a 26.08, 9.67, 28.35, and 25.58 percent increase in expected total discounted reward compared to π^{POW}, π^{THP}, π^{EQU}, and π^{PRF}, respectively. We observe that π^{MDP} shows increased savings as the average sensor node lifetime increases due to an increase in the number of decision epochs and thus prolonged operation of sensor nodes in optimal states as prescribed by π^{MDP}. On average over all discount factors λ, π^{MDP} results in a 25.57, 9.48, 27.91, and 25.1 percent increase in expected total discounted reward compared to π^{POW}, π^{THP}, π^{EQU}, and π^{PRF}, respectively.

3.3.2 The Effects of Different State Transition Costs on the Expected Total Discounted Reward

Fig. 3 depicts the effects of different state transition costs on the expected total discounted reward for a security/defense system with a fixed average sensor node lifetime of 1,000 hours ($\lambda = 0.999$) and ω_p, ω_t, and ω_d equal to 0.45, 0.2, and 0.35, respectively. Fig. 3 shows that π^{MDP} results in the highest expected total discounted reward for all transition cost values.

Fig. 3 also shows that the expected total discounted reward for π^{MDP} is relatively unaffected by state transition cost. This relatively constant behavior can be explained by the fact that our MDP optimal policy does not perform many state transitions. Relatively few state transitions to reach the optimal state according to the specified application metrics may be advantageous for some application managers who consider the number of state transitions prescribed by a policy as a secondary evaluation criteria [10]. π^{MDP} performs state transitions primarily at sensor node deployment or whenever a new MDP-based optimal policy is determined as the result of changes in application requirements.

We further analyze the effects of different state transition costs on the fixed heuristic policies, which consistently result in a lower expected total discounted reward as compared to π^{MDP}. The expected total discounted rewards for π^{POW} and π^{THP} are relatively unaffected by state transition cost. The explanation for this behavior is that these heuristics perform state transitions only at initial sensor node deployment when the sensor node transitions

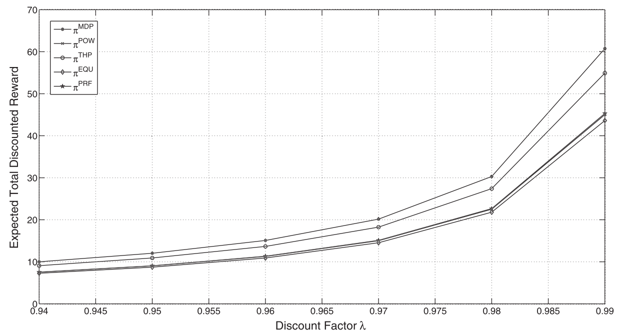

Fig. 2. The effects of different discount factors on the expected total discounted reward for a security/defense system. $H_{i,j} = 0.1$ if $i \neq j$, $\omega_p = 0.45, \omega_t = 0.2, \omega_d = 0.35$.

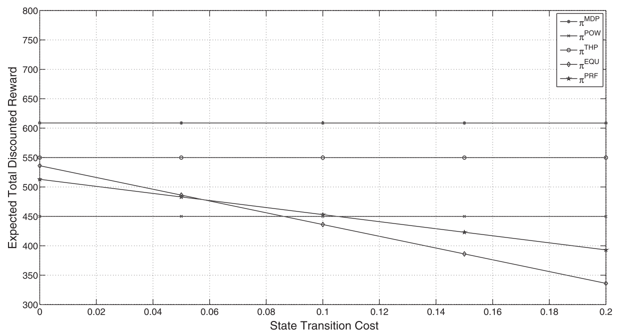

Fig. 3. The effects of different state transition costs on the expected total discounted reward for a security/defense system. $\lambda = 0.999$, $\omega_p = 0.45, \omega_t = 0.2, \omega_d = 0.35$.

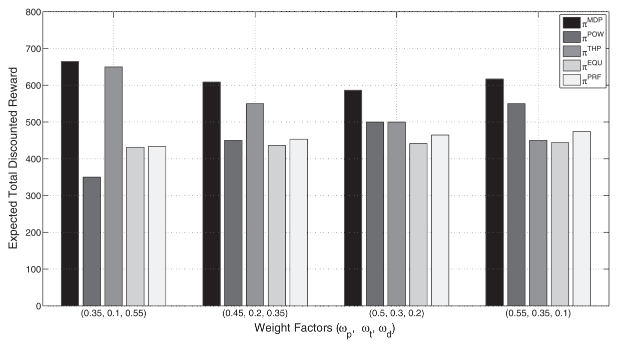

Fig. 4. The effects of different reward function weight factors on the expected total discounted reward for a security/defense system. $\lambda = 0.999$, $H_{i,j} = 0.1$ if $i \neq j$.

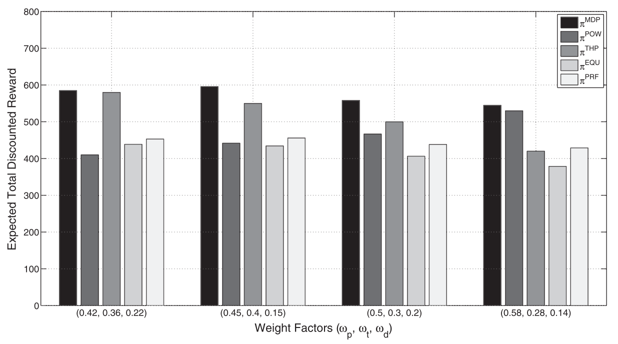

Fig. 6. The effects of different reward function weight factors on the expected total discounted reward for a health care application. $\lambda = 0.999$, $H_{i,j} = 0.1$ if $i \neq j$.

to the lowest power state and the highest throughput state, respectively, and remain in these states for the entire sensor node's lifetime. On the other hand, state transition cost has the largest affect on the expected total discounted reward for π^{EQU} due to high state transition rates because the policy spends an equal amount of time in all states. Similarly, high switching costs have a large affect on the expected total discounted reward for π^{PRF} (although less severely than π^{EQU}) because π^{PRF} spends a certain percentage of time in each available state (Section 3.1), thus requiring comparatively fewer transitions than π^{EQU}.

3.3.3 *The Effects of Different Reward Function Weight Factors on the Expected Total Discounted Reward*

Fig. 4 shows the effects of different reward function weight factors on the expected total discounted reward for a security/defense system when the average sensor node lifetime is 1,000 hours ($\lambda = 0.999$) and the state transition cost $H_{i,j}$ is held constant at 0.1 for $i \neq j$. We explore various weight factors that are appropriate for different security/defense system specifics, i.e., $(\omega_p, \omega_t, \omega_d) = \{(0.35, 0.1, 0.55), (0.45, 0.2, 0.35), (0.5, 0.3, 0.2), (0.55, 0.35, 0.1)\}$. Fig. 4 reveals that π^{MDP} results in the highest expected total discounted reward for all weight factor variations.

3.4 Results for a Health Care Application

3.4.1 *The Effects of Different Discount Factors on the Expected Total Discounted Reward*

Fig. 5 depicts the effects of different discount factors λ for a health care application when the state transition cost $H_{i,j}$ is held constant at 0.1 for $i \neq j$, and ω_p, ω_t, and ω_d are equal to 0.5, 0.3, and 0.2, respectively. Fig. 5 shows that π^{MDP} results in the highest expected total discounted reward for all values of λ and corresponding average sensor node lifetimes as compared to other fixed heuristic policies.

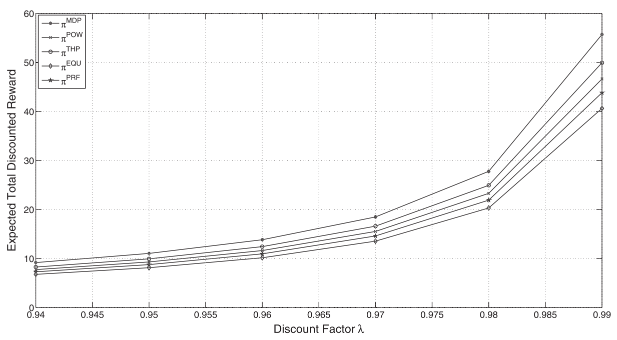

Fig. 5. The effects of different discount factors on the expected total discounted reward for a health care application. $H_{i,j} = 0.1$ if $i \neq j$, $\omega_p = 0.5, \omega_t = 0.3, \omega_d = 0.2$.

We calculate the percentage improvement in expected total discounted reward for π^{MDP} for a health care application as compared to the fixed heuristic policies. We observe that when the average sensor node lifetime is 1,000 hours ($\lambda = 0.999$), π^{MDP} results in a 16.39, 10.43, 27.22, and 21.47 percent increase in expected total discounted reward compared to π^{POW}, π^{THP}, π^{EQU}, and π^{PRF}, respectively. On average over all discount factors λ, π^{MDP} results in a 16.07, 10.23, 26.8, and 21.04 percent increase in expected total discounted reward compared to π^{POW}, π^{THP}, π^{EQU}, and π^{PRF}, respectively.

3.4.2 *The Effects of Different State Transition Costs on the Expected Total Discounted Reward*

We observe the effects of different state transition costs on the expected total discounted reward for a health care application with a fixed average sensor node lifetime of 1,000 hours ($\lambda = 0.999$) and ω_p, ω_t, and ω_d equal to 0.5, 0.3, and 0.2, respectively. Results reveal that π^{MDP} results in the highest expected total discounted reward for all transition cost values. The fixed heuristic policies consistently result in a lower expected total discounted reward as compared to π^{MDP}.

3.4.3 *The Effects of Different Reward Function Weight Factors on the Expected Total Discounted Reward*

Fig. 6 depicts the effects of different reward function weight factors on the expected total discounted reward for a health care application when the average sensor node lifetime is 1,000 hours ($\lambda = 0.999$) and the state transition cost $H_{i,j}$ is kept constant at 0.1 for $i \neq j$. We explore various weight factors that are appropriate for different health care application specifics (i.e., $(\omega_p, \omega_t, \omega_d) = \{(0.42, 0.36, 0.22), (0.45, 0.4, 0.15), (0.5, 0.3, 0.2), (0.58, 0.28, 0.14)\}$). Fig. 6 shows that π^{MDP} results in the highest expected total discounted reward for all weight factor variations.

Figs. 4 and 6 show that the expected total discounted reward of π^{POW} gradually increases with as the power weight factor increases and eventually exceeds that of π^{THP} for a security/defense system and a health care application, respectively. However, close observation reveals that the expected total discounted reward of π^{POW} for a security/defense system is affected more sharply than a health care application, because of the more stringent constraint on maximum acceptable power for a health care application (Table 2). Figs. 4 and 6 show that π^{PRF} tends to perform better than π^{EQU} with increasing power weight factors because π^{PRF} spends a greater percentage of time in low power states.

4 Conclusions and Future Work

In this paper, we present the first (to the best of our knowledge) dynamic optimization methodology for WSNs based on Markov Decision Processes. Our MDP-based dynamic optimization methodology tunes sensor node processor voltage, frequency, and sensing frequency in accordance with application requirements over the lifetime of a sensor node. Our proposed methodology is adaptive and dynamically determines the new MDP-based optimal policy whenever application requirements change (which may be in accordance with changing environmental stimuli). We compared our MDP-based optimal policy with four fixed heuristic policies and conclude that our proposed MDP-based optimal policy outperforms each heuristic policy for all sensor node lifetimes, state transition costs, and application metric weight factors. We provided the implementation guidelines of our proposed policy in sensor nodes. We proved that our proposed policy has fast convergence rate, computationally inexpensive and thus can be considered for implementation in sensor nodes with limited processing resources.

Future work includes enhancing our MDP model to incorporate additional high-level application metrics (e.g., security, reliability, energy, lifetime, etc.) as well as additional sensor node tunable parameters (such as radio transmission power, radio transmission frequency, etc.). Furthermore, we plan to incorporate wireless channel condition in the MDP state space, thus formulating a stochastic dynamic program that enables sensor node tuning in accordance with changing wireless channel condition. We plan to implement our MDP-based methodology on hardware sensor nodes for further verification of results. In addition, we will enhance sensor node tuning automation using profiling statistics by architecting mechanisms that enable the sensor node to automatically react to environmental stimuli without the need for an application manager's feedback. Future work also includes the extension of our MDP-based dynamic optimization methodology for performing *global optimization* (i.e., selection of sensor node tunable parameter settings to ensure that application requirements are met for WSN as a whole where different sensor nodes collaborate with each other in optimal tunable parameter settings determination).

Acknowledgments

This work was supported by the US National Science Foundation (NSF) (CNS-0834080) and the Natural Sciences and Engineering Research Council of Canada (NSERC). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF and NSERC. This paper has supplementary material available at http://ieeexplore.ieee.org, provided by the authors. The supplementary material contains additional discussions, modeling details, results, figures, etc., for our MDP-based dynamic optimization methodology proposed in this paper.

References

[1] M. Horton, "Commercial Wireless Sensor Networks: Status, Issues and Challenges," *Proc. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON): Keynote Presentation,* Oct. 2004.

[2] K. Greene, "Sensor Networks for Dummies," *Technology Rev. (published by MIT),* Mar. 2006.

[3] C.-Y. Seong and B. Widrow, "Neural Dynamic Optimization for Control Systems," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 31, no. 4, pp. 482-489, Aug. 2001.

[4] K. Hazelwood and M.D. Smith, "Managing Bounded Code Caches in Dynamic Binary Optimization Systems," *ACM Trans. Architecture and Code Optimization,* vol. 3, no. 3, pp. 263-294, Sept. 2006.

[5] H. Hamed, A. El-Atawy, and A.-S. Ehab, "On Dynamic Optimization of Packet Matching in High-Speed Firewalls," *IEEE J. Selected Areas in Comm.,* vol. 24, no. 10, pp. 1817-1830, Oct. 2006.

[6] S. Hu, M. Valluri, and L.K. John, "Effective Management of Multiple Configurable Units Using Dynamic Optimization," *ACM Trans. Architecture and Code Optimization,* vol. 3, no. 4, pp. 477-501, Dec. 2006.

[7] A. Munir and A. Gordon-Ross, "An MDP-Based Application Oriented Optimal Policy for Wireless Sensor Networks," *Proc. Seventh IEEE/ACM Int'l Conf. Hardware/Software Codesign and System Synthesis (CODES+ISSS),* Oct. 2009.

[8] S. Sridharan and S. Lysecky, "A First Step towards Dynamic Profiling of Sensor-Based Systems," *Proc. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* June 2008.

[9] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, Inc., 2005.

[10] E. Stevens-Navarro, Y. Lin, and V. Wong, "An MDP-Based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks," *IEEE Trans. Vehicular Technology,* vol. 57, no. 2, pp. 1243-1254, Mar. 2008.

[11] S. Lysecky and F. Vahid, "Automated Application-Specific Tuning of Parameterized Sensor-Based Embedded System Building Blocks," *Proc. IEEE Int'l Conf. Ubiquitous Computing (UbiComp),* Sept. 2006.

[12] I. Chadès, M. Cros, F. Garcia, and R. Sabbadin, "Markov Decision Process (MDP) Toolbox v2.0 for MATLAB," *INRA Toulouse,* INRA, France, http://www.inra.fr/internet/Departements/MIA/T/MDPtoolbox/, Feb. 2005.

[13] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events," *Proc. ACM Int'l Symp. Information Processing in Sensor Networks (IPSN),* Apr. 2005.

[14] P. Dutta and D. Culler, "System Software Techniques for Low-Power Operation in Wireless Sensor Networks," *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design (ICCAD),* Nov. 2005.

[15] ATMEL "ATMEL ATmega128L 8-Bit Microcontroller Datasheet," *ATMEL Corporation,* San Jose, California, http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf, Dec. 2010.

[16] Honeywell "Honeywell 1- and 2- Axis Magenetic Sensors HMC1001/1002, and HMC1021/1022 Datasheet," *Honeywell Int'l, Inc.,* Morristown, New Jersey, http://www.ssec.honeywell.com/magnetic/datasheets/hmc1001-2_1021-2.pdf, Dec. 2010.

[17] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* vol. 38, no. 4, pp. 393-422, Mar. 2002.

Arslan Munir received the BSc degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, in 2004, and the MASc degree in electrical and computer engineering (ECE) from the University of British Columbia (UBC), Vancouver, Canada, in 2007. He is currently working toward the PhD degree in ECE at the University of Florida (UF), Gainesville. From 2007 to 2008, he worked as a software development engineer at Mentor Graphics in the Embedded Systems Division. He was the recipient of many academic awards including the Gold Medals for the best performance in Electrical Engineering and Academic Roll of Honor. He received the Best Paper award at the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. His current research interests include embedded systems, low-power design, computer architecture, multicore platforms, parallel computing, dynamic optimizations, fault-tolerance, and computer networks. He is a student member of the IEEE.

Ann Gordon-Ross received the BS and PhD degrees in computer science and engineering from the University of California, Riverside, in 2000 and 2007, respectively. She is currently an assistant professor of electrical and computer engineering at the University of Florida and is a member of the US National Science Foundation (NSF) Center for High Performance Reconfigurable Computing (CHREC) at the University of Florida. She is also the faculty advisor for the Women in Electrical and Computer Engineering (WECE) and the Phi Sigma Rho National Society for Women in Engineering and Engineering Technology. She received her CAREER award from the US National Science Foundation in 2010 and Best Paper awards at the Great Lakes Symposium on VLSI (GLSVLSI) in 2010 and the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. Her research interests include embedded systems, computer architecture, low-power design, reconfigurable computing, dynamic optimizations, hardware design, real-time systems, and multicore platforms. She is a member of the IEEE.