Dynamic Power Management in Wireless Sensor Networks: State-of-the-Art

Waltenegus Dargie, *Member, IEEE*

***Abstract*—In the last few years, interest in wireless sensor networks has increased considerably. These networks can be useful for a large number of applications, including habitat monitoring, structural health monitoring, pipeline monitoring, transportation, precision agriculture, supply chain management, and many more. Typically, a wireless sensor network consists of a large number of simple nodes which operate with exhaustible batteries, unattended. Manual replacement or recharging the batteries is not an easy or desirable task. Hence, how energy is utilized by the various hardware subsystems of individual nodes directly affects the scope and usefulness of the entire network. This paper provides a comprehensive assessment of state-of-the-art of dynamic power management (DPM) in wireless sensor networks. It investigates aspects of power dissipation in a node and analyses the strength and limitations of selective switching, dynamic frequency, and voltage scaling.**

***Index Terms*—Clock gating, dynamic frequency scaling, dynamic power management, dynamic voltage scaling, embedded systems, power gating, selective switching, wireless sensor networks.**

I. Introduction

Wireless sensor networks are one of the technologies that are gaining a considerable attention. They have been deployed to monitor the activities of animals and plants whose behavioural patterns or distributions can easily be affected by human presence [19]; to inspect the structural integrity of bridges and buildings [18], [33] as well as of pipelines [26]; to capture the presence and extent of active volcanoes [32]. Likewise, in precision agriculture, they have been used to monitor soil moisture, radiation, pH, and humidity [7], [6]. Other applications include healthcare [29] and supply chain management [20].

Typically, a wireless sensor network consists of a large number of nodes each of which integrates one or more sensors, a processing subsystem and a short range transceiver. The nodes are capable of organizing themselves to establish and maintain a network and carry out reliable sensing. However, when considered individually, each node is a simple device; the components that make up its subsystems are commonplace, off-the-shelf components. Ideally, the network should have a long life and operate unattended, but several factors put a limit to the energy source:

1) Considering the complexity of the task for which they are deployed – namely, sensing, processing, and communication –, the nodes are very small in size to accommodate high capacity batteries.
2) Given the size of the network and its deployment setting, manually replacing or recharging batteries on a periodic basis is a formidable challenge.
3) Whereas research is being conducted to employ renewable energy and self-recharging mechanisms, still the size of presently available nodes makes the task difficult.
4) The failure of a few number of nodes may fragment the entire network prematurely.

The problem of power consumption has been addressed in two different ways in the literature. In the first, a large number of energy-efficient communication protocols – most significantly, MAC, routing and self-organization protocols – that take the peculiarities of wireless sensor networks into account have been proposed. In the second, local dynamic power management (DPM) strategies are developed to recognize and minimize the impact of wasteful and inefficient activities within an individual node.

Wasteful and inefficient activities can be accidental side-effects or results of non-optimal software and hardware configurations. For example field observations revealed that some nodes exhausted their batteries prematurely because of unexpected overhearing of traffic that caused the communication subsystem to become active for a time longer than originally anticipated [17]. Similarly, some nodes exhausted their batteries prematurely because they aimlessly attempted to establish links with a network that has become no longer accessible to them.

Most of the time, however, inefficient power consumption results due to not-optimal configurations in hardware and software components. For example, a considerable amount of power can be dissipated in an idle processing or communication subsystem. Similarly, a receiver that aimlessly receives packets that are not destined to it; or overhears while neighboring nodes communicate with each other consumes a significant amount of power.

A local DPM strategy ensures that such wasteful activities are avoided and power is consumed frugally. Ideally, it provides each subsystem of a node with the amount of power that is sufficient enough to carry out a task at hand. When there is no task to be processed or executed, it forces the subsystem to operate at the most economical power mode or turns it off altogether.

There has been a considerable interest in the past, and as a consequence, a significant body of work, in dynamic power management, particularly, in the context of embedded systems. But wireless sensor networks bring their own challenges and

Manuscript received May 03, 2011; revised September 28, 2011; accepted October 18, 2011. Date of publication October 28, 2011; date of current version April 13, 2012. The associate editor coordinating the review of this paper and approving it for publication was Prof. Tianhong Cui.

The author is with the Chair of Computer Networks, Faculty of Computer Science, Technical University of Dresden, 01062 Dresden, Germany (e-mail: waltenegus.dargie@tu-dresden.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2011.2174149

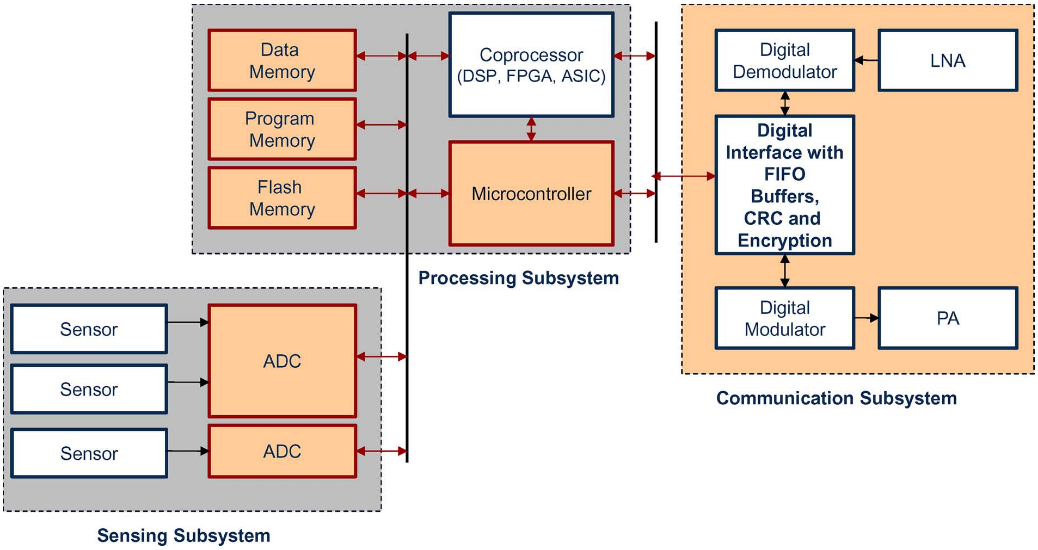


Fig. 1. A partial view of the system architecture of a wireless sensor node. The components with gold background are those to which DPM can be applied.

peculiarities into the research field. To begin with, unlike embedded systems, which function, by and large, stand-alone, no individual node is of interest in and of itself. Secondly, a local decision made by a node can have a global impact. This paper attempts to provide a comprehensive insight into aspects of DPM in wireless sensor networks. It presents the challenges, the results that are achieved so far, and some outstanding research issues in need of attention.

The remaining part of the paper is organized as follows: In Section II, sources of power dissipation in a wireless sensor node are discussed in detail. In Section III, different DPM strategies and their side-effects are presented. In Section IV, a conceptual architecture for a DPM is discussed. In Section V, hardware and software prototype implementations of dynamic power management systems in wireless sensor networks are discussed. Finally, in Section VI, some outstanding research issues are discussed and concluding remarks are given.

II. Power Dissipations

Power in a sensor node can be inefficiently dissipated for various reasons. Fig. 1 shows four of the main subsystems of a wireless sensor node, namely, the sensing, memory, processor and communication subsystems. Different types of communication interfaces (serial buses) interconnect these subsystems, but the most frequently used are the serial peripheral interface (SPI) and the inter-integrated circuit (I^2C). A SPI bus is useful for high speed communication (for example, between the communication and processing subsystems), whereas the half-duplex I^2C is suitable for low speed communication (mostly between the ADCs of the sensing subsystem and the processing subsystem).

A. Processor Subsystem

At the very low-level, undesirable power dissipation occurs due to various intrinsic leakage components in the CMOS transistors. Some of these are weak inversion, drain-induced barrier lowering, gate-induced drain leakage, and gate oxide tunneling [23]. Finding the right balance between the various transistor design parameters is a formidable challenge; despite remarkable achievements in semi-conductor technologies, there are still lossy components. A related source of power dissipation (known as dynamic dissipation power) is the charging and discharging of load capacitances [2]. This power is quadratically proportional to the DC supply voltage and linearly proportional to the operating frequency.[1] While decreasing the bias voltage reduces the dynamic power dissipation, there is a side effect to it, however, as it also means that the threshold voltage – the voltage required to turn on the transistor – should also be reduced, which, in turn, results in a significant amount of leakage current.

TABLE I
Nominal Current Draw of the ATmega128L Microcontroller

Power Mode	Current
Active	8.0 mA
Idle	3.2 mA
ADC Noise red.	1.0 mA
Power-down	0.103 mA
Power-save	0.110 mA
Standby	0.216 mA
Extended Standby	0.223 mA

Most existing processing subsystems employ microcontrollers, notably Intel's StrongARM and Atmel's AVR. These microcontrollers enable some of their internal components to be turned-off completely when they are idle. For example the ATmega128L microcontroller provides six different configurations, each of which has a different power dissipation profile: *idle*, *ADC noise reduction*, *power save*, *power down*, *standby* and *extended standby*. The *idle* mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port and the interrupt system to continue functioning. The *ADC Noise Reduction* mode stops the CPU and all *I/O* modules, except the asynchronous timer and the ADC. The aim is to minimize switching noise during ADC conversions. In *Power save mode*, the asynchronous timer continues to run, allowing the user to maintain a timer base; the remaining components of the device

[1] This knowledge is the basis for dynamic frequency and voltage scaling, to be discussed in Section III-B.

TABLE II
SIX DIFFERENT POWER MODES OF THE ATMEGA128L MICROCONTROLLER. THE "X" INDICATES THE ACTIVE (ENABLED) HARDWARE COMPONENTS. THERE ARE SOME ADDITIONAL COMPONENTS WHICH ARE ACTIVE (ENABLED) IN ALL POWER MODES, BUT THEY ARE REMOVED FROM THE TABLE

Sleep Mode	Active Clock Domains			Oscillators		Wakeup Sources				
	clk_{IO}	clk_{ADC}	clk_{ASY}	Main CLK	Timer Oscil	INT7:0	Timer 0	SPM/ EEPROM	ADC	Other I/O
Idle	X	X	X	X	X	X	X	X	X	X
ADC noise. Redu.			X	X	X	X	X	X	X	
Power Down						X				
Power Save				X		X	X			
Standby					X	X				
Ext. Standby					X	X	X			

enter into a sleep mode. The *Power down* mode saves the registers' content while freezing the oscillator and disabling all other chip functions until the next interrupt or Hardware Reset. In *Standby* mode, a crystal/resonator oscillator runs while the remaining hardware components enter into a sleep mode. This allows very fast start-up combined with low power consumption. In *Extended Standby* mode, both the main oscillator and the asynchronous timer continue to operate. Table I summarizes the nominal current consumption of the ATmega128L microcontroller in the different power modes. Table II displays the types of hardware components which are active (enabled) in each power modes.

Additional to these configurations, present day microcontrollers, including the ATmega128L microcontroller, can be configured to operate at different supply voltages and clock frequencies.

B. Communication Subsystem

Unlike the processor subsystem, the tasks of the communication subsystem are known at the time of deployment. Consequently, the low-level software code as well as the hardware is fine tuned for an optimal performance. Even so, the communication subsystem can aimlessly consume a significant amount of power. There are two main reasons for this: (1) idle listening, and (2) overhearing. Idle listening occurs when a node does not have knowledge about the arrival of packets which are directly addressed to it, and therefore, the receiver remains idly powered. Overhearing occurs when the receiver receives and processes packets which are not intended to it. In both cases, the active components such as voltage and intermediate amplifiers, the local oscillator and the phase-locked-loops (PLLs) are all active and quiescent currents flow in their circuits. Knowledge of packet reception and transmission rates is useful for defining a sleeping schedule for the communication subsystem. For this purpose, many commercially available transceivers provide different power modes. For instance, the CC2420 transceiver [22] can be configured in one of the eight discrete transmission levels when it is active (-24 to 0 dBm). Moreover, it can be configured to operate in one of the three low power modes, namely, Off (voltage regulator off), Power down (voltage regulator enabled), and Idle (crystal oscillator running) when it is idle.

Compared to all the other subsystems of a wireless sensor node, the radio subsystem consumes a significantly large amount of power when it receives, transmits, or idly listens, which is why it is necessary to make the communication subsystem sleep on a periodic basis. Almost all medium access control protocols in wireless sensor networks (such as XMAC [5] and RI-MAC [27]) enable a node to periodically switch off the communication subsystem. This will be further discussed in Section V.

C. Communication Interfaces

Power is consumed when the processor subsystem interacts with the other subsystems via the internal high speed buses. The specific amount depends on the frequency and bandwidth of the communication. These two parameters can be optimally configured depending on the interaction type, but bus protocol timings are usually optimized for particular bus frequencies. Moreover, bus controller drivers require to be notified when bus frequencies change to ensure stable performance.

D. Memory

On account of space and cost constraints, the memory unit of most commercially available wireless sensor nodes contains a Dynamic RAM (DRAM) in which one bit of information is stored in a transistor-capacitor pair. Since real capacitors are lossy, they require to be recharged (refreshed) periodically. The refresh rate has a direct bearing on the power consumption of the memory unit. A DRAM can be configured to operate in different power modes: Temperature compensated self-refresh mode, partial array self-refresh mode or power down mode. The standard refresh rate of a memory unit can be adjusted according to its ambient temperature. For this reason, some commercially available DRAMs integrate temperature sensors. Apart from this, the self-refresh rate can be reduced if the entire memory array is not used to store data. In other words, the refresh operation can be limited to the portion of the memory array in which actual data are stored. This approach is known as partial array self-refresh mode. If no actual data storage is required, the supply voltage of most or the entire on-board memory array can be switched off. The partial array self-refresh mode becomes efficient when the sensor data are stored in a contiguous block of the memory. This can be achieved if the operating system supports dynamic memory allocation[2].

The performance of the processing subsystem depends on how fast data and instructions can be transferred between the processor subsystem and the memory subsystem. Each memory access and each bus transaction results in energy consumption in the bus drivers and the memory unit [30]. The RAM timing,

[2] Not all operating systems in wireless sensor networks support dynamic memory allocation. A good summary of the operating systems in wireless sensor networks and a comparison between them can be found in [10, Ch. 4].

TABLE III
PARAMETERS OF RAM TIMING

Parameter	Description
RAS	Row Address Strobe or Row Address Select
CAS	Column Address Strobe or Column Address Select
t_{RAS}	A time delay between the precharge and activation of a row
t_{RCD}	The time required between RAS and CAS access
t_{CL}	CAS Latency
t_{RP}	The time required to switch from one row to the next row
t_{CLK}	the duration of a clock cycle
Command Rate	The delay between Chip Select (CS)
Latency	The total time required before data can be written to or read from memory

another parameter that influences the power consumption of the memory unit, refers to the latency associated with accessing the memory unit. Before a processor subsystem accesses a particular cell in a memory, it should first determine the particular row or bank and then activate it with a *row access strobe (RAS)* signal. Once a row is activated, it can be accessed until the data are exhausted. The time required to activate a row in a memory is t_{RAS}, which is relatively small but could impact the system's stability if set incorrectly. A memory cell is activated through a *column access strobe (CAS)*. The delay between the activation of a memory cell and the writing of data into or reading of data from the cell is given as t_{RCD} – it is called RAS to CAS delay. This time can be short or long, depending on how the memory is accessed. If it is accessed sequentially, it is insignificant, but, if the memory is accessed in a random fashion, the current active row must first be deactivated before a new row is activated, in which case, the t_{RCD} latency can be considerable.

The delay between the *CAS signal* and the availability of valid data on the data bus is called *CAS Latency*. Low *CAS latency* means high performance but also high power consumption. The time required to terminate one row access and begin the next row access is t_{RP}. In conjunction with t_{RCD}, the time (or clock cycles) required to switch banks (rows) and select the next cell for reading, writing, or refreshing is expressed as $t_{RP} + t_{RCD}$. The duration of time required between the active and precharge commands is called t_{RAS}. It is a measure of how long the processor must wait before the next memory access begins. Table III summarizes the quantities that express RAM timing.

Besides how memory is accessed, several decode and multiplex (switching) stages have to be passed through the bus to move data from the memory unit to the processing subsystem and vice versa [30]. The dynamic power consumption of the memory unit depends on the rate at which the decoding and multiplexing operations take place.

E. Power Subsystem

The power subsystem supplies DC voltage to all the other subsystems and consists of a battery and a DC-DC converter, among other things. The DC-DC converter is responsible for providing the right amount of supply voltage to each individual hardware component by transforming the main DC supply voltage into a suitable level. The transformation can be a step-down (buck), a step-up (boost), or an inversion (flyback) process, depending on the requirements of the individual subsystems. The transformation process has its own power consumption, even though this is a small amount.

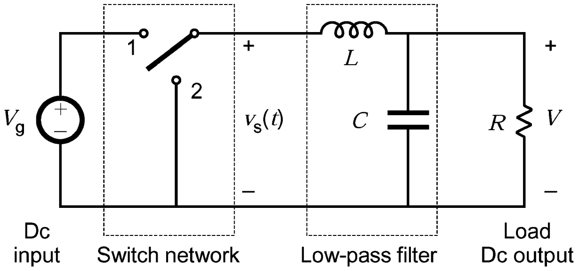


Fig. 2. A DC-DC converter consisting of a supply voltage, a switching circuit, a filter circuit and a load resistance.

1) Batteries: Batteries are specified by a rated current capacity, C, expressed in *Ampere-Hour*. This quantity describes the maximum amount of energy that can be withdrawn from a battery under a specified discharge rate and temperature. Most portable batteries are rated at $1C$, which means a 1000 *mAh* battery provides 1000 *mA* for one hour, if it is discharged at a rate of $1C$. Ideally, the same battery can discharge at a rate of $0.5C$, providing 500 *mA* for two hours; and at $2C$, 2000 *mA* for 30 minutes and so on. $1C$ is often referred to as a one-hour discharge. Likewise, a $0.5C$ would be a two-hour discharge and a $0.1C$ a ten-hour discharge.

In reality, batteries perform at a rate below the prescribed rate. Often, the Peukert Equation [11] is applied to quantify the capacity offset.

$$C_p = I^k t \tag{1}$$

where C_p is the Peukert Capacity expressed in *Ampere-Hours*; I is the discharge current in Ampere; k is a dimensionless constant that refers to the internal resistance of the battery (known as the Peukert constant). This value indicates how well a battery performs under continuous heavy currents. A value close to 1 indicates that the battery performs well; the higher the number, the more capacity is lost when the battery is discharged at high currents. k is determined empirically. For example, for lead acid batteries, the number is typically between 1.3 and 1.4. Finally, t is the discharge time expressed in hours.

Drawing current at a rate greater than the discharge rate results in a current consumption rate higher than the rate of diffusion of the active elements in the electrolyte. If this process sustains for a long time, the electrodes run out of active material even though the electrolyte has not yet completely exhausted the active material. This situation can be overcome by intermittently drawing current from the battery.

2) DC-DC Converter: The DC-DC converter transforms one voltage level into another voltage level. It is the equivalent of an AC voltage transformer. The main problem with a DC-DC converter is its conversion efficiency. A typical DC-DC converter consists of a power supply, a switching circuit, a filter circuit and a load resistor. Fig. 2 illustrates the basic circuit structure of a DC-DC converter.

As can be seen in the figure, the circuit consists of a single-pole, double-throw (SPDT) switch that is connected to a DC supply voltage, V_g. Considering the inductor, L, as a short circuit and the capacitor, C, as an open circuit for the DC supply voltage, the output voltage of the switch, $V_s(t)$, equals to V_g

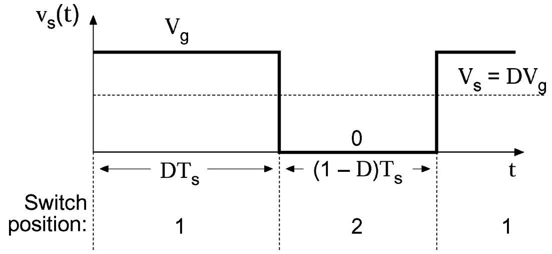


Fig. 3. Output voltage of the switching circuit of a DC-DC converter.

when the switch is in position 1 and 0 when it is in position 2. Varying the position of the switch at a frequency, f_s, yields a periodically varying square wave, $v_s(t)$, that has a period $T_s = 1/f_s$.

$v_s(t)$ can be expressed by a duty cycle D, which describes the fraction of time that the switch is in position 1, such that $0 \leq D \leq 1$. The output voltage of the switching circuit is displayed in Fig. 3.

A DC-DC converter is realized by employing active switching components, such as diodes or power MOSFETs. Typically, the switching frequencies range from 1 *kHz* to 1*MHz*, depending on the speed of the semiconductor devices. Using the inverse Fourier transformation, the DC component of $v_s(t)$ (V_s) is described as:

$$V_s = \frac{1}{T_s}\int_0^{T_s} v_s(t)dt = DV_g \tag{2}$$

which is the average value of $v_s(t)$.

In other words, the integral value represents the area under the waveform of Fig. 3 for a single period, or the height of V_g multiplied by the time T_s. It can be seen that the switching circuit reduces the DC component of the supply voltage by a factor that equals to the duty cycle, D. Since $0 \leq D \leq 1$ holds, $V_s \leq V_g$ holds as well.

Ideally the switching circuit should not consume power. In practice, however, due to the presence of an internal capacitive load, there is some power dissipation. As a result, the efficiency of the switching circuit is between 70 and 90%.

In addition to the desired DC voltage, $v_s(t)$ contains undesirable harmonics of the switching frequency, f_s. These harmonics must be removed so that the converter's output voltage $v(t)$ is essentially equal to the DC component $V = V_s$. For this purpose, a DC-DC converter employs a lowpass filter. In Fig. 2, a first-order LC low-pass filter is connected to the switching circuit. The filter's cut-off frequency is given by:

$$f_c = \frac{1}{2\pi\sqrt{LC}}. \tag{3}$$

The cut-off frequency, f_c, should be sufficiently less than the switching frequency, f_s, so that only the DC component of $v_s(t)$ is allowed to pass to the next stage. Once again, in an ideal filter, there is no power dissipation in the passive components. In practice, however, there is some dissipation.

The DC-DC converter produces a DC output voltage whose magnitude can be controlled by the duty cycle, D, using circuit elements that (ideally) do not dissipate power. The conversion ratio, $M(D)$, is defined as the ratio of the DC output voltage, V, to the DC input voltage, V_g, under a steady-state condition:

$$M(D) = \frac{V}{V_g}. \tag{4}$$

For the buck converter shown in Fig. 2, $M(D) = D$.

III. Dynamic Power Management

There are two basic premises for implementing DPM in wireless sensor networks: (1) Wireless sensor networks are predominantly event driven and the events occur infrequently; and (2) nodes experience non-uniform workload. Hence, the main goal of a DPM strategy is to identify idle and underutilized hardware components and adapt their power requirements accordingly. This entails determining the type and timing of the power transition based on the system's history, workload, and performance constraints [24]. Power transitions can take place by selectively switching off/on hardware components or through dynamic frequency and voltage scaling [3]. The former is sometimes referred to as power gating [4] while the latter as clock gating [9], [34].

A. Selective Switching

In Section II, it has been shown that some of the hardware components of a wireless sensor node can be configured to operate at different power modes, depending on their present and anticipated workload. The decision for a particular power mode should take the cost of power transition and the associated latency into account. There are several factors which influence these costs. For instance, the processor subsystem has to save and load context (state) information during power mode switching; the communication subsystem has to start up and synchronize some devices before actual transmission and reception begin – the frequency synthesizer's phase-locked loop (PLL) of the widely used Chipcon CC2420 transceiver requires 192 μs to lock up. Similarly, switching the StrongARM-1100 microcontroller from an active mode ($\text{nominal power consumption} = 400$ mW) to an idle mode ($\text{power consumption} = 50$ mW) takes 10 μs whereas a transition from an active mode to a sleep mode ($\text{power consumption} = 160\ \mu$W) requires 90 μs. To bring the same processor from an idle mode to an active mode will take 10 μs whereas from a sleep mode to an active mode will take 160 μs [3]. For a DRAM, transition from an active mode ($\text{power consumption} = 300$ mW) to a nap mode (power consumption = 3 mW) saves a significant amount of power, but bringing the memory unit back to the active state during data access can incur a transition cost of 165 *mW* and a delay of 120*ns* [14].

In a wireless sensor network, the interesting phenomena to be captured (for example, a leakage in a pipeline, a fracture in a structure, or a pestilence in a farm) cannot be modeled as deterministic events. Otherwise there would be no need for setting up a monitoring sentinel. Therefore, estimation of the arrival of events in the network should be probabilistic. The associated uncertainty forces a trade-off to be made between energy efficiency and the potential to miss vital events.

1) Transition Costs: Consider a hardware component inside a node that operates in just two different power modes – i.e., *on* or *sleep*. For simplicity, the transition from *on* to *sleep* does not have an associated power cost, but the reverse transition (from *sleep* to *on*) results in power and delay costs. These costs are justified, if the energy that can be saved in the *sleep* state is a large amount. It is useful to quantify these costs and set up a transition threshold.

Let t_{sleep} denote the minimum time the hardware component stays in a *sleep* state; the power consumed during this time is P_{sleep}; the transition time is $t_{sleep,on}$; the power consumed during the transition is $p_{sleep,on}$; and the power consumed in an *on* state is P_{on}. Hence:

$$P_{sleep} \cdot t_{sleep} + P_{sleep,on} \cdot t_{sleep,on} \leq P_{on} \cdot (t_{sleep} + t_{sleep,on}) . \tag{5}$$

Therefore, t_{sleep} is justified if [8]:

$$t_{sleep} \geq \max\left(0, \frac{(P_{on} - P_{sleep,on}) \cdot t_{sleep,on}}{P_{sleep} - P_{on}}\right). \tag{6}$$

Equations (5) and (6) can easily be generalized to describe n distinct operational power modes, in which case a transition from any state i (signifying a higher operational power mode) to j (signifying a lower operational power mode) resulting a reverse transition time $t_{j,i}$. Hence, the transition is justified if (7) is satisfied.

$$t_j \geq \max\left(0, \frac{(P_i - P_{j,i}) \cdot t_{j,i}}{P_j - P_i}\right) \tag{7}$$

where t_j is the duration of the subsystem in state j.

The equations above assume that the transition cost from a higher power mode (*on*) to a lower power mode (*off*) is negligible. If this is not the case, the energy that can be saved through a power transition (from state i to state j, $E_{saved,j}$) is expressed as:

$$E_{saved,j} = P_i \cdot (t_j + t_{i,j} + t_{j,i}) - (P_{i,j} \cdot t_{i,j} + P_{j,i} \cdot t_{j,i} + P_j \cdot t_j) . \tag{8}$$

If the transition from state i to state j costs the same amount of power and time delay as the transition from state j to state i (a symmetric transition cost), (8) can be expressed as:

$$\begin{aligned} E_{saved,j} = {} & P_i \cdot (t_j + t_{i,j} + t_{j,i}) \\ & - \left(\frac{P_i + P_j}{2}\right)(t_{i,j} + t_{j,i}) \\ & - (P_i - P_j) \cdot t_j . \end{aligned} \tag{9}$$

Obviously, the transition is justified if $E_{saved,j} > 0$. This can be achieved in three different ways, namely, by:

1) increasing the gap between P_i and P_j;
2) increasing the duration of state j, t_j; and,
3) decreasing the transition times, $t_{j,i}$ and $t_{i,j}$.

Algorithm: Consider a wireless sensor node that consists of an ATmega128L microcontroller, a CC2420 transceiver, and a DRAM memory unit. The different power modes of the individual components are summarized in Fig. 4. A DPM based on the Selective Switching technique can have at least $3 \times 3 \times 6 = 54$ different operation points: $\{P_{operation} : P_0, P_1, \ldots, P_{53}\}$. A transition from P_i to P_j has always an associated power and delay cost.

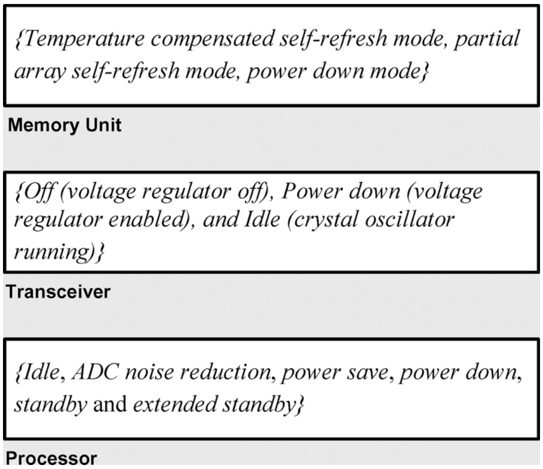
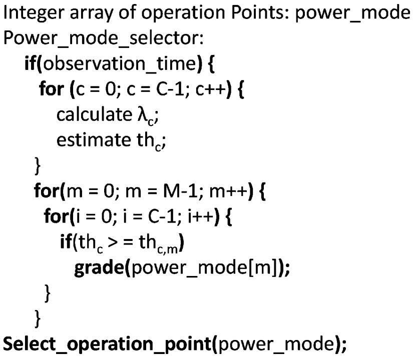

{Temperature compensated self-refresh mode, partial array self-refresh mode, power down mode}
Memory Unit
{Off (voltage regulator off), Power down (voltage regulator enabled), and Idle (crystal oscillator running)}
Transceiver
{Idle, ADC noise reduction, power save, power down, standby and extended standby}
Processor

Fig. 4. Power configuration space of a wireless sensor node.



```
Integer array of operation Points: power_mode
Power_mode_selector:
  if(observation_time) {
    for (c = 0; c = C-1; c++) {
      calculate $\lambda_c$;
      estimate $th_c$;
    }
    for(m = 0; m = M-1; m++) {
     for(i = 0; i = C-1; i++) {
       if($th_c$ > = $th_{c,m}$)
         grade(power_mode[m]);
     }
    }
Select_operation_point(power_mode);
```

Fig. 5. An example realisation of a DMP (selective switching) as a part of a first-in-first-out (FIFO) scheduler.

The task of the DPM is, therefore, to observe the activity (workload) of each hardware component for a set period of time and estimate the task arrival rate in the future. Based on this estimation, the optimal power mode and the associated transition time is computed for each hardware component. Then, the aggregate transition time is calculated to determine the appropriate operation point of the node.

One way of implementing the DPM is as a part of an energy-aware scheduler, since the scheduler has complete knowledge of the tasks that should be executed and the type of hardware resources they require. Moreover, schedulers inherently define queues for scheduling tasks (no task will be lost even when a hardware component is sleeping or is switched off) and, therefore, the DPM implementation does not require a modification of the scheduler's architecture.

Fig. 5 displays an example DPM realisation as a part of an energy-aware FIFO scheduler. Based on knowledge of each manageable hardware component, $\{\mathrm{c} : \mathrm{c} = 1, .., \mathrm{C}\}$, the DPM defines a minimum dwell time ($th_{c,m}$) for each component c to remain in the state m, operating at operating power P_m, using (6) or one similar to it. The DPM periodically examines the task arrival rate at each component (λ_c) and estimates the average time (th_c) between two consecutive tasks until the next observation time. Alternatively, th_c can be the average latency for device c that can be tolerated by the application. Once th_c is obtained, the DPM grades (bonuses) each operating point if $th_c \geq th_{c,m}$.

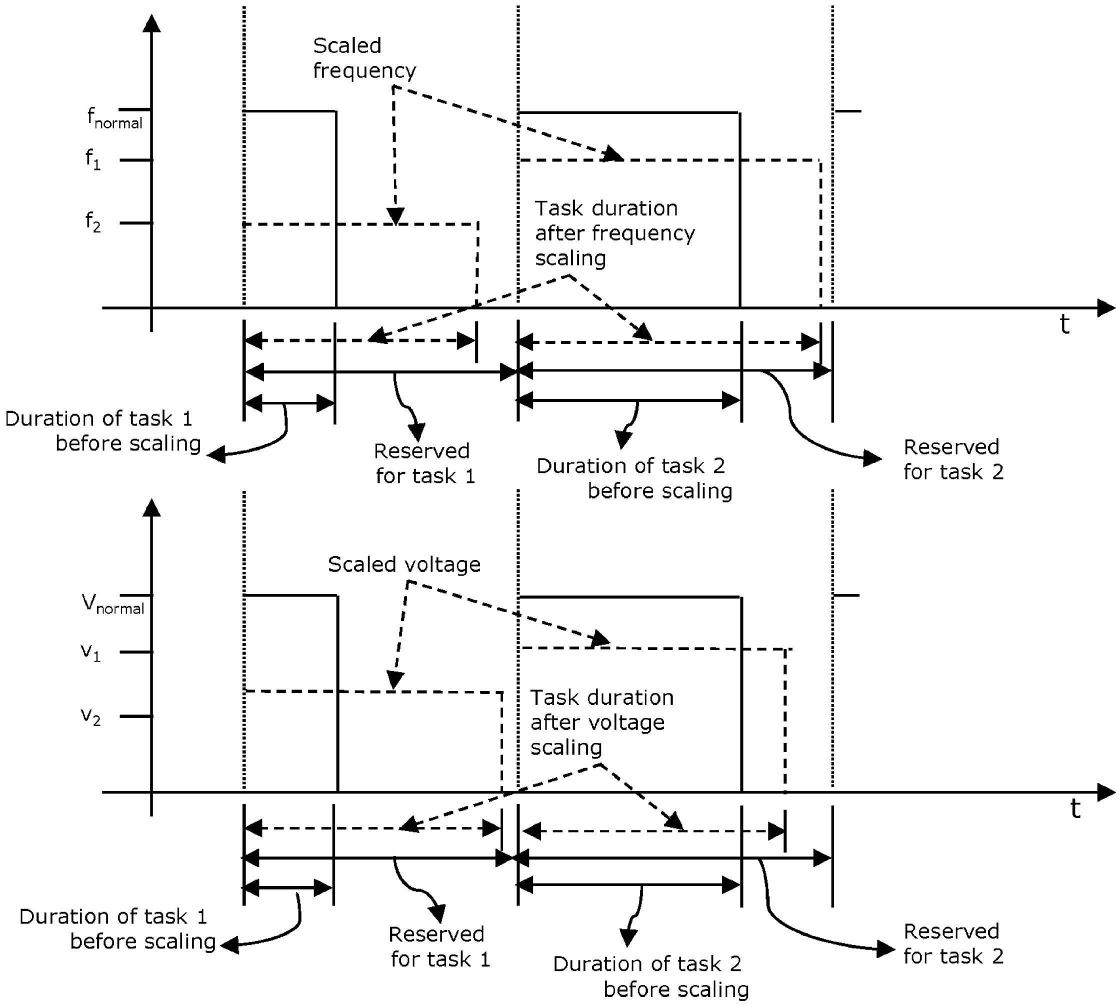

Scaled frequency
f_normal
f_1
f_2
Task duration after frequency scaling
t
Duration of task 1 before scaling
Reserved for task 1
Duration of task 2 before scaling
Reserved for task 2
Scaled voltage
V_normal
V_1
V_2
Task duration after voltage scaling
t
Duration of task 1 before scaling
Reserved for task 1
Duration of task 2 before scaling
Reserved for task 2

Fig. 6. A processor subsystem operating at its peak performance (solid line). The power consumption of the processor can be improved by applying either frequency (above) or voltage (below) scaling (indicated by the dash lines). As can be seen, a scaling process results in a delay in execution time. Ideally, this latency is still below the time set by the scheduler when no voltage or frequency scaling is applied, but in reality, it is greater than this limit. This is the cost of scaling and its significance is application-dependent.

Finally, the operating point that obtains the highest grading will be selected as the next operating point.

In our example, there are $M = 54$ different operating points and $C = 3$ manageable hardware components.

B. Dynamic Scaling

Dynamic voltage scaling (DVS) and dynamic frequency scaling (DFS) are complimentary to Selective Switching. These two approaches aim at adapting the performance of the processor core (as well as the memory unit and the internal communication interfaces) when the node is active.

Most of the time, the tasks scheduled by the operating system (runtime environment) do not require the processor to execute at its peak capacity. Rather, some tasks are completed ahead of their deadline and the processor enters into a low-leakage idle mode for the remaining time. As illustrated by Fig. 6, even though the two tasks are completed ahead of their schedule, the processor runs still at peak frequency and supply voltage, which is wasteful.

With DVS and DFS, the supply voltage and clock frequency of some of the subsystems of a wireless node are scaled down according to the present and anticipated workload, so that each task is stretched to its planned schedule. While reduction in the operation frequency results in a linear energy saving, reduction in the supply voltage results in quadratic saving. However, the reduction in magnitude in both cases cannot take place arbitrarily; only specific discrete amounts are permitted to ensure a stable operation. Moreover, the reduction cannot take place endlessly. For example, the minimum operating voltage for CMOS logic to function under a stable condition was first derived by Swanson and Meindl [28] and is expressed as follows:

$$V_{dd,limit} = 2 \cdot \frac{kT}{q} \cdot \left[1 + \frac{C_{fs}}{C_{ox} + C_d}\right] \cdot \ln\left(1 + \frac{C_d}{C_{ox}}\right) \quad (10)$$

where C_{fs} is the surface state capacitance per unit area; C_{ox} is the gate-oxide capacitance per unit area; and C_d is the channel depletion region capacitance per unit area.

Unlike Selective Switching, there are two types of costs with dynamic scaling. Firstly, power supplies require a finite amount of time to settle to the new operating voltage; the delay being a function of the load on the supply voltage. During this time, some hardware components (in some cases, the processor subsystem itself), should be halted and isolated to avoid unreliable operation. This requires an external hardware. Secondly, due to

TABLE IV
FILTER COEFFICIENTS FOR TASK ARRIVAL RATE ESTIMATION

Filter Type	Filter Coefficients
Moving Average	$h_k(i) = \frac{1}{N}$
Exponential Weighted Average	$h_k(i) = a^{-i}$
Least Mean Square	$h_{n+1}(k) = h_n(k) + \mu w_e(n) w(n-k)$

the capacitive load inside the CMOS circuitry, power transition does not take place at once. The switching delay can be approximated by the following equation:

$$t_{delay} = \frac{C_s \cdot V_{dd}}{I_{d-sat}} \tag{11}$$

where C_s is the source capacitance; V_{dd} is the supply voltage at the drain; and I_{d-sat} is the drain saturation current. According to Sinha and Chandrakasan, the relationship between the energy cost, the operation frequency and the supply voltage [25] can be expressed as follows:

$$E(r) = CV_0 2 T_s f_{ref} r \left[\frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + \left(\frac{r}{2}\right)^2} \right] \tag{12}$$

where, C is the average switching capacitance per cycle; T_s is the sampling period; f_{opp} is the operating frequency at V_{dd} supply voltage; r is the normalized processing rate ($r = f/f_{opp}$); and $V_0 = (V_{dd} - V_t)^2/V_{dd}$, where V_t is the threshold voltage.

Dynamic voltage and frequency scaling require stable clock generator and DC-DC converter, which imply development cost.

C. Task Scheduling

As mentioned earlier, the estimation of the task arrival rate at the scheduler and the anticipated workload of the different subsystems are crucial preconditions for a DPM technique. Sinha and Chandrakasan [25] investigate several types of filters as estimation techniques. These filters take the past N tasks at the scheduler – tasks that are executed to the end in a First In First Out (FIFO) scheduler – into account and estimate the workload of the processor for the next observation period:

$$W_p[n] = \sum_{k=0}^{N-1} h_n[k] W[n-k]. \tag{13}$$

The filter's coefficients in (13) can have different complexity, depending on the type of filter chosen. For a moving average filters, all filter coefficients have the same values, whereas for an Exponential Weighted Average filter, the coefficients of recent workloads have more significance than previous workloads. For Least Mean Square filters, the values of the filter coefficients are set based on the difference between the estimated and actual workloads of past observations. The coefficients of the different filters are summarized in Table IV. Experiment results show that large values of N do not necessarily correspond to more accurate estimation [25]. Moreover, Least Mean Square filters are more reliable than other, but they are also more complex and computational intensive.

A complementary approach is to estimate the workload of each hardware resource during a specified observation time and estimate the future workload independently. This is particularly useful for Selective Switching. Merkel and Bellosa propose a "task activity vector " (TAV) as a part of the runtime context of a task [21]. The vector has a dimension that is equal to the number of hardware components which support dynamic power management. Each component of the vector denotes the degree of utilization of a corresponding hardware component when the task is executed. The idea is to provide the scheduler with detailed information regarding the resource requirement of each task. Hardware event counters can be employed to determine the frequency and duration of hardware access by each task [16]. This way, the scheduler is able to determine the right type of power adaptation for each hardware component. The side effect of this approach is its disregard of dependency between the different hardware components.

IV. CONCEPTUAL ARCHITECTURE

The implementation of a DPM technique should address three essential concerns:

1) How much is the net power that can be gained by a DPM technique? How much is the extra computational overhead that should be introduced?
2) Should the technique be centralized or distributed?
3) If it is a centralized approach, which of the subsystems of a wireless sensor node should be responsible for the task?

A typical DPM technique monitors the activities of individual subsystems and makes decisions pertaining to the suitable operational power modes. Since this process consumes certain amount of power and requires additional resources, it can be justified if the net power gain is significantly large.

The decision whether a DPM technique should be central or distributed depends on several factors. One advantage of a centralized approach is that it is possible to achieve a global view of the power consumption of the node. On the other hand, a global technique can add a significant computational load on the subsystem that undertakes the management. A distributed approach scales well by authorizing individual subsystems to carry out local power management strategies. The problem with this approach is that local strategies may contradict with global goals. Given the relative simplicity of a wireless sensor node and the quantifiable tasks that should be processed, most existing power management strategies are based on centralized solutions.

A. Architectural Overview

Though the aim of a DPM technique is to optimize the power consumption of a node, it should not affect the system's stability. Furthermore, the application requirements in terms of the quality of sensed data and latency should be satisfied. Fortunately, in most realistic situations, a wireless sensor network is deployed with a specific task in mind. This task does not change, or it changes gradually. Knowledge of the sensing task, the deployment setting and the network density simplifies the workload estimation. This is summarized by Fig. 7.

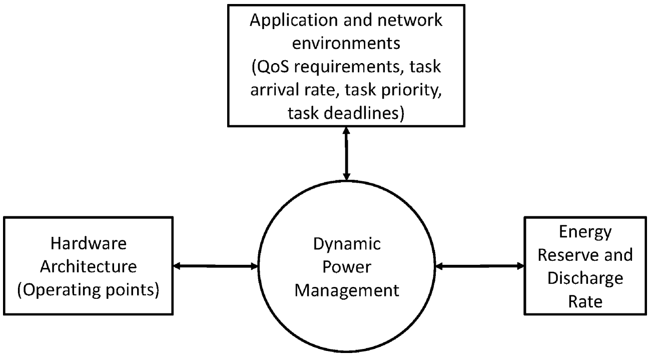

Application and network environments (QoS requirements, task arrival rate, task priority, task deadlines)
Hardware Architecture (Operating points)
Dynamic Power Management
Energy Reserve and Discharge Rate

Fig. 7. Factors affecting a DPM technique.

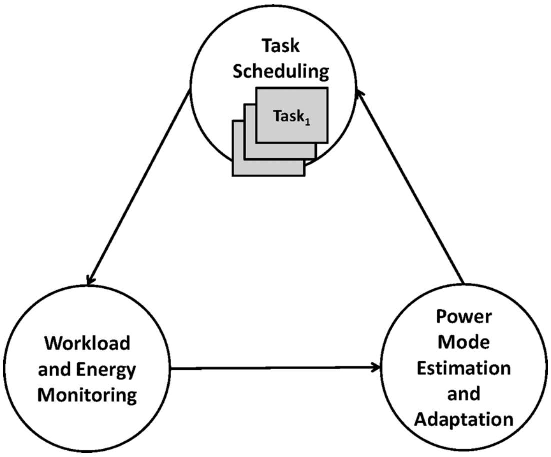

Task Scheduling
Task$_1$
Workload and Energy Monitoring
Power Mode Estimation and Adaptation

Fig. 8. A DPM technique modelled as a state machine.

A DPM technique will take several factors into account, including the system's hardware architecture and the various operation modes of individual hardware components; the available resources (CPU workload and memory) to carry out various computations pertaining to the management task; the energy reserve of the node as well as the current discharge rate and the load of the supply voltage; and the application's quality of service requirements. Once the workload of the node and the power modes of the different subsystems for the next observation period are estimated, the DPM technique may reconfigure the hardware components, reschedule tasks, and adjust biasing voltages and the frequency of clock generators, which is why the arrows in Fig. 7 indicate in both direction.

The process can be understood as a circular process consisting of three basic operations: workload and energy monitoring, power mode estimation and adaptation and task (re)scheduling. The energy consumption of the different hardware components should be monitored to determine the deviation between the approximated and the actual energy consumption of the node. This will be useful to adjust the filter coefficients of the task arrival estimation filter. The circular process is illustrated in Fig. 8.

V. Implementation

Decisions with regard to the implementation of the dynamic power management architecture entail the choice between Selective Switching of hardware components and Dynamic Voltage and Frequency Scaling. Whereas the former can be realised with a software component that collaborates with the operating system or runtime environment (scheduler), the latter may require an external hardware component, depending on the sensitivity of the processing subsystem. There are mainly three reasons for this:

1) During voltage scaling, most practical power supplies require a certain amount of time to settle to the new voltage level. This time depends on the load on the supply voltage bus.
2) When voltage and frequency scaling are applied, the processor subsystem may not operate reliably and should therefore be halted during the transition.
3) A change in the operating frequency of a CPU may affect the operation of internal phase-locked loops (PLLs), which may mean reprogramming the PLLs. In fact, this limits the range of frequencies that can be supported by a dynamic frequency scaling.

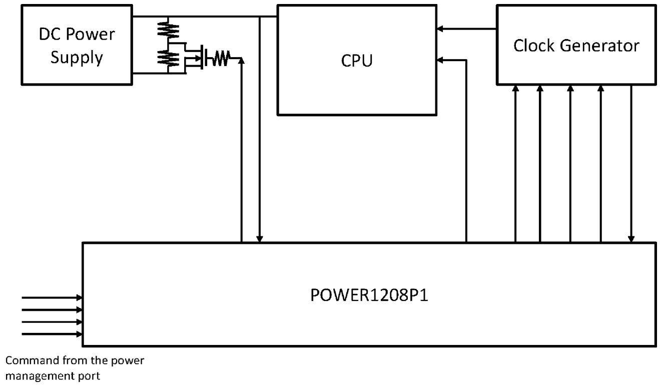


Fig. 9. A partial view of a hardware realisation of dynamic voltage and frequency scaling.

Fig. 9 displays the Lattice Semiconductor Corporation hardware subsystem that supports dynamic frequency and voltage scaling. The subsystem consists of the supply voltage (which outputs two different voltage levels, depending on whether the MOSFET transistor connected to the DC supply voltage is on or off), a clock generator (which outputs 20 different clock frequencies), and the power controller (POWER1208P1) which provides all the logic for frequency and voltage scaling functions. The controller receives instructions (from the processor subsystem) through the four pins at the left side. The subsystem decouples the effect of voltage and frequency transitions from the processing subsystem and provides stable inputs to it. One of the side effects of employing an external hardware subsystem for a DPM is the additional space requirement.

Almost all Medium Access Control (MAC) protocols in wireless sensor network support periodic sleeping of the communication subsystem of a node to frugally utilise power (thereby avoiding idle listening and overhearing). These protocols define a duty cycle, $D = T_{Active}/(T_{Active} + T_{Sleep})$, which is less than 10%. The duty cycle depends on the data traffic within the network and the maximum end-to-end delay in packet delivery. Fig. 10 displays a software realisation of a DPM based on periodic sleeping. The link layer defines three radio states, namely, on (active), off (sleep), and waiting states. If the communication subsystem is in the "on" states, it participates in multi-hop communications by forwarding packets to and from neighbour

```
state: {ON, WAITING, OFF}
radio_wake_eventhandler:
  if (state = ON)
    if (expired(timer))
      timer ← $t_{sleep}$
      if (not communication_complete())
        state ← WAITING
        wait_timer ← $t_{wait\_max}$
      else
        radio_off()
        state ← OFF
  elseif (state = WAITING)
    if (communication_complete() or
        expired(wait_timer))
      state ← OFF
      radio_off()
  elseif (state = OFF)
    if (expired(timer))
      radio_on()
      state ← ON
      timer ← $t_{awake}$
```

Fig. 10. Software implementation of a DPM at the Link Layer [13], [10].

nodes. It remains active until the time set for the "on" state expires. If, however, the node is receiving data at the time the "on" state time expires, it will remain in that state until all the data have been received. A node wakes up at any time if it has data to transmit. This type of DPM does not require additional hardware and the program overhead is not considerable.

In general, however, knowledge of the application requirements is essential to make the appropriate choice. For some applications, for example, power adaptation based only on task arrival rate estimation does not save power. Weissel *et al.* [31] illustrate how a DPM technique employed to control the sleeping schedule of an IEEE 802.11 based transceiver can cost different applications different amount of power. Subsequently, the authors recommend that power management techniques should take several context information – average size of packets received; ratio of average length of inactive to length of active periods; ratio of average size of packets received to size of packets sent; ratio of traffic volume received to traffic volume sent; average size of packets sent, etc. – into consideration.

VI. Conclusion

An efficient use of energy is a crucial concern in wireless sensor networks. So far, the research community has tried to address the problem of power dissipation in two ways: (1) developing energy-efficient communication protocols and data processing algorithms; and (2) implementing and executing power management policies. Whereas the former approach has usually a global scope (at any rate, a scope that goes beyond a single node), the latter has usually a local scope, limited to a single node. Based on the hardware architecture of a wireless sensor node, it has been shown that a DPM technique can ensure the efficient use of power by monitoring the activities of the processor, memory unit, transceiver, and communication interfaces. Additionally, battery capacity can be enhanced by controlling how current is discharged.

Two types of DPM techniques are introduced. In Selective Switching, the main idea is to minimize the idle state power dissipation of hardware components. This is achieved by estimating the appropriate power mode of individual components and configuring them to operate in thes states. The advantage with this approach is that the technique can be implemented with software components only, since most of the hardware components provide well-defined interfaces to be dynamically configured. The main problem with the approach is the cost of power transition, both in terms of power and delay. Some of the existing operating systems in wireless sensor networks, for example, TinyOS [1], [15] and Contiki [12] provide application developers with hardware abstractions for implementing Selective Switching.

In dynamic power and frequency scaling, the active state power requirement of a hardware component is adapted to its present and anticipated workload. If the workload of a node changes slowly over time, dynamic scaling can be efficient. The problem with this approach is its requirement for stable clock generator and power supply, each of which is capable of providing different output levels. Some power supplies require a finite amount of time to settle to the new operating voltage. The delay is a function of the load on the supply voltage. In this case, it may be necessary to isolate the processor subsystem during the transition, i.e., an extra hardware is required to undertake this job.

Workload (task arrival rate) estimation is vital to strike a balance between the power that can be saved and the latency that comes from power transition (which in turn, may introduce its own power cost). Even though there are a large number of estimation techniques, inherently, complex estimation techniques are computation-intensive and require a large amount of memory, which is not available in a wireless sensor node. As a result, only simple filters can be realized inside a node.

While DPM is an extensively investigated subject in the context of embedded systems, wireless communications, peer-to-peer communications and wireless sensor networks, work still remains to quantitatively describe the resource demand, implementation complexity and processing time of the approaches proposed. In the end, these non-functional aspects determine the scope and usefulness of the DPM techniques.

References

[1] "T. T. 2.x Working Group. Tinyos 2.0," in *Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, 2005, pp. 320–320.

[2] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in cmos vlsi circuits by input vector control," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 12, no. 2, pp. 140–154, Feb. 2004.

[3] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 8, no. 3, pp. 299–316, Mar. 2000.

[4] A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *Comput. J.*, vol. 57, no. 7, pp. 1–15, 2009.

[5] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble mac protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, 2006, pp. 307–320.

[6] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," *IEEE Pervasive Comput.*, vol. 3, no. 1, pp. 38–45, Jan./Mar. 2004.

[7] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. L. P. Corrêa, "From wireless sensors to field mapping: Anatomy of an application for precision agriculture," *Comput. Electron. Agric.*, vol. 58, no. 1, pp. 25–36, 2007.

[8] C. Chiasserini and R. Rao, "Improving energy saving in wireless systems by using dynamic power management," *IEEE Trans. Wireless Commun.*, vol. 2, no. 5, pp. 1090–1100, Sep. 2003.

[9] K. Choi, R. Soma, and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, New York, NY, 2004, pp. 174–179.

[10] W. Dargie and C. Poellabauer*, Fundamentals of Wireless Sensor Networks: Theory and Practice*, 1 ed. New York: Wiley, 2010.

[11] D. Doerffel and S. A. Sharkh, "A critical review of using the peukert equation for determining the remaining capacity of lead-acid and lithium-ion batteries," *J. Power Sources*, vol. 155, no. 2, pp. 395–400, 2006.

[12] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki – A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw. (LCN)*, Washington, DC, 2004, pp. 455–462.

[13] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: Simplifying event-driven programming of memory-constrained embedded systems," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, 2006, pp. 29–42.

[14] X. Fan, C. Ellis, and A. Lebeck, "Memory controller policies for dram power management," in *: Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, New York, NY, 2001, pp. 129–134.

[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proc. 9th Int. Conf. Archit. Support Program. Languages Oper. Syst. (ASPLOS-IX)*, New York, NY, 2000, pp. 93–104.

[16] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchit. (MICRO)*, Washington, DC, 2003, p. 93.

[17] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker, "An architecture for energy management in wireless sensor networks," *SIGBED Rev.*, vol. 4, no. 3, pp. 31–36, 2007.

[18] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. 6th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, New York, NY, 2007, pp. 254–263.

[19] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM Int. Workshop Wireless Sensor Netw. Appl. (WSNA)*, 2002, pp. 88–97.

[20] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso, "Cargonet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events," in *Proc. 5th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, 2007, pp. 145–159.

[21] A. Merkel and F. Bellosa, "Task activity vectors: A new metric for temperature-aware scheduling," in *Proc. 3rd ACM SIGOPS/EuroSys Eur. Conf. Comput. Syst.*, New York, NY, 2008, pp. 1–12.

[22] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Piscataway, NJ, 2005, pp. 364–369.

[23] K. Roy and S. Mukhopadhyay, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits," *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[24] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, "Dynamic voltage scaling and power management for portable systems," in *Proc. 38th Annu. Design Automat. Conf. (DAC)*, New York, NY, 2001, pp. 524–529.

[25] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 62–74, Mar./Apr. 2001.

[26] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "Pipenet: A wireless sensor network for pipeline monitoring," in *Proc. 6th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, New York, NY, 2007, pp. 264–273.

[27] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 1–14.

[28] R. Swanson and J. Meindl, "Ion-implanted complementary mos transistors in low-voltage circuits," *IEEE J. Solid-State Circuits*, vol. 7, no. 2, pp. 146–153, 1972.

[29] U. Varshney, "Pervasive healthcare and wireless health monitoring," *Mob. Netw. Appl.*, vol. 12, no. 2–3, pp. 113–127, 2007.

[30] A. Weissel and F. Bellosa, "Process cruise control: Event-driven clock scaling for dynamic power management," in *Proc. Int. Conf. Compilers, Archit., Synthesis Embedded Syst. (CASES)*, New York, NY, 2002, pp. 238–246.

[31] A. Weissel, M. Faerber, and F. Bellosa, "Application characterization for wireless network power management," in *Proc. Int. Conf. Archit. Comput. Syst. (ARCS)*, 2004, pp. 231–245.

[32] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 18–25, Mar./Apr. 2006.

[33] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst.*, New York, NY, 2004, pp. 13–24.

[34] Y.-W. Yang and K. S.-M. Li, "Temperature-aware dynamic frequency and voltage scaling for reliability and yield enhancement," in *Proc. Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Piscataway, NJ, 2009, pp. 49–54.

Waltenegus Dargie (M'08) received the B.Sc. degree in electrical engineering from the Nazareth Technical College, Ethiopia, in 1997, and the M.Sc. degree from the Technical University of Kaiserslautern, Germany, in 2002, both in electrical engineering, and the Ph.D. degree in computer engineering from the Technical University of Dresden, Germany, in 2006.

He was a Researcher at the Department of Electrical Engineering and Computer Science, University of Kassel, Germany, from 2002 to 2005, and also at the Fraunhofer Institute of Experimental Software Engineering, Kaiserslautern, Germany, from 2002 to 2003. He is currently an Associate Professor at the Technical University of Dresden, Germany. His research interests include wireless networks, wireless sensor networks, and digital signal processing.