Evaluation of Body Sensor Network Platforms: A Design Space and Benchmarking Analysis

Sidharth Nabar[*], Ayan Banerjee[†], Sandeep K.S. Gupta[†] and Radha Poovendran[*]
[*]Network Security Lab (NSL), University of Washington, Seattle, USA
[†]IMPACT Lab, Arizona State University, Tempe, USA
snabar@uw.edu, abanerj3@asu.edu, sandeep.gupta@asu.edu, rp3@uw.edu

ABSTRACT

Body Sensor Networks (BSNs) consist of sensor nodes deployed on the human body for health monitoring. Each sensor node is implemented by interfacing a physiological sensor with a sensor platform consisting of components such as microcontroller, radio and memory. Diverse needs of BSN applications require customized platform development for optimizing performance. In this paper, we propose a two-phase framework to evaluate the performance of sensor platforms to match a BSN's computation, communication and sensing requirements: 1) **Design Space Determination**, wherein we investigate salient features of BSN platforms and quantify them as design coordinates through evaluation metrics such as SPSW (Samples Processed per Second per Watt) and EPC (Expected Power Consumption). To measure these metrics for a platform under typical BSN application workloads, we propose BSNBench, a benchmarking suite composed of basic tasks that occur in diverse BSN applications. BSNBench enables an accurate profiling of platforms based on the design coordinates ; 2) **Design Space Exploration**, wherein we explore the design space to find the most suitable platform for a given application. We demonstrate the usage of our framework through a case study, where we consider two practical BSN applications and choose suitable platforms for them.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Experimentation, Measurement, Performance

Keywords

Body Sensor Networks, design space, system performance, hardware systems, benchmark, wearable BSNs

1. INTRODUCTION

Body Sensor Networks (BSNs) consist of miniature wireless sensors that are deployed on a person's body to collect data related to physiological parameters such as temperature, blood glucose level, or heart rate. This data is then transmitted to a central gateway device such as a cellphone or PDA, which in turn can convey it to a healthcare provider or physician over the internet. Starting from this basic design, several diverse applications have been proposed for BSNs, including chronic disease management [23], remote elderly care and human motion analysis [17].

As Body Sensor Networks (BSNs) move from the research stage towards a widely used practical technology, experimental deployments and clinical trials need to be performed in order to validate proposed sensor designs, network architectures or wireless communication protocols. These experiments require the use of programmable sensor platforms[1] that can be used as wearable body sensors. These platforms consist of a microcontroller, radio, antenna, external memory and other peripheral components integrated on a single circuit board. Due to the advances in embedded devices technology, several such platforms have been designed for use in BSN trials [12, 21]. Although different BSN applications impose highly diverse requirements on the underlying platform, it is often not feasible to develop customized hardware platforms for each application due to time and cost constraints. A well thought out choice from existing sensor platforms, instead, can be a good starting point for application specific design of BSNs.

The state-of-the-art BSN platforms display significant diversity in their architecture, processing capability, energy consumption and form factor. Given such diversity in platform characteristics, a standardized approach is required to analyze a given platform and evaluate it as a wearable BSN node. The current approach to platform selection, however, is ad hoc and lacks a well-defined methodology. Defining a standard evaluation method or framework for BSN platforms will enable accurate profiling of existing platforms for various BSN applications. Further, such a framework would help researchers study the performance of existing architectures and develop more efficient systems. Finally, BSN device manufacturers can use this framework to benchmark their devices and compare them to competing devices.

The goal of this paper is *to develop a framework to evaluate sensor platforms based on a set of BSN-specific system characteristics and identify the most suitable platform for a given BSN application*. The framework consists of two phases:

Design Space Determination - In this phase, we identify the features of a sensor platform that determine its performance as a BSN node. These features are then defined as *design coordinates* for BSN platforms, and are further quantified using *evaluation met-*

[1]In this paper, we use the terms 'sensor platform' and 'BSN platform' interchangeably.

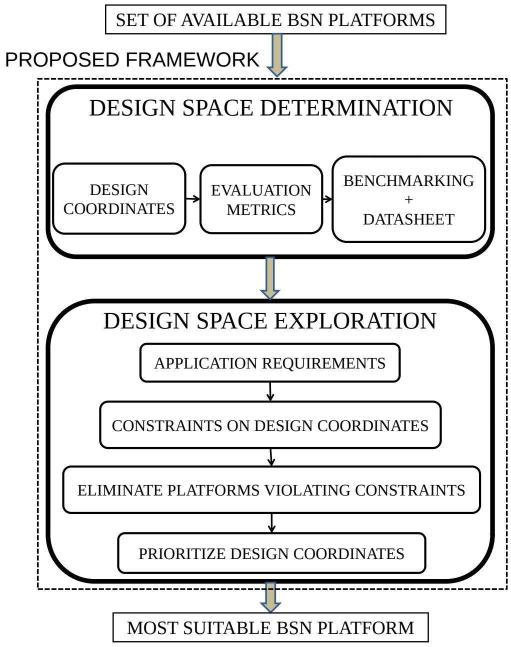

Figure 1: BSN Platform Evaluation Framework

rics. For example, the battery used in a platform is considered to be a design coordinate, and is quantified in terms of its energy capacity and size as evaluation metrics. As a part of this mapping, we introduce two new metrics: **SPSW (Samples Processed per Second per Watt)** and **EPC (Expected Power Consumption)** for measuring the processor performance and radio power consumption, respectively. SPSW captures the tradeoff between power consumption of the processor and the speed of execution, while EPC measures average radio power consumption at different duty cycles.

It is observed that several design coordinates are application dependent, and hence the corresponding metrics must be evaluated under typical BSN workloads. To enable such an evaluation, we define BSNBench, a BSN-specific benchmarking suite consisting of basic, independent tasks that occur in several diverse BSN applications. Using BSNBench, the design coordinate values for several existing platforms are computed in order to determine a *design space*. This mapping of BSN platforms in the design space is the final output of this phase.

Design Space Exploration - In this phase we explore the aforementioned design space to determine the most suitable platform for a given application. Requirements of the application are used to formulate constraints on individual design coordinates. These constraints define a subspace within the overall design space, and the platforms lying inside this subspace are considered suitable for the application. Further, priorities can be set among the remaining design coordinates to identify the single most suitable platform. Figure 1 summarizes the two-phase structure of our framework.

We make the following contributions in this paper:

- We identify a set of design coordinates that characterize a wearable sensor platform and determine its performance in BSN applications. Using these coordinates, we perform the design space analysis of BSN platforms.

- We propose two new metrics for evaluating BSN platforms: SPSW (Samples Processed per Second per Watt) and EPC (Expected Power Consumption). SPSW characterizes the overall processor performance, while EPC measures the average radio power consumption in an application.

- We present a BSN-specific benchmarking suite, called BSNBench, that enables a practical evaluation of platforms based on the proposed design coordinates. BSNBench is intended to be an initial effort towards the goal of defining a standard BSN benchmark, and will be extended in our future work.

- We consider two practical BSN applications - Blood glucose monitoring and Epileptic seizure detection - to show how our framework enables an application-dependent selection of platforms.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the set of design coordinates and the corresponding evaluation metrics used in the design space determination phase. Section 4 describes the design and implementation of BSNBench. In Section 5, we present results obtained using BSNBench for a set of commonly used sensor platforms. Section 6 illustrates the process of design space exploration through a case study of two BSN applications. Section 7 concludes the paper.

2. RELATED WORK

At the time of its conception, a BSN was considered to be a specific application in the broader class of Wireless Sensor Networks (WSNs). Concepts such as use of miniature sensor nodes, processing and aggregation of sensor data and energy-efficient communication protocols were well studied in WSNs and were directly applicable to BSNs. As a result, initial BSN deployments used generic WSN platforms such as TelosB [1], Mica2 [2] and Imote2 [3]. Benchmarks such as TinyBench [9], WisenBench [15] and SenseBench [19] were proposed as an evaluation method for these platforms. However, these benchmarks do not provide a deep insight into the performance of these platforms in BSNs, since they focus on generic WSN applications.

As BSN research advanced towards a medical technology, specialized healthcare applications emerged [11], such as remote monitoring of elderly patients [21], epileptic seizure detection [10] and continuous monitoring of patients with chronic diseases [23]. In addition, traditional medical applications such as EMG measurements, ECG analysis, and glucose monitoring were implemented using BSNs [14, 22, 23]. With these developments, unique aspects of BSNs such as physiological signal processing, need for wearable form factor and interfacing with medical sensors, were identified, which differentiated them from generic WSNs. As a result, several BSN-specific sensor platforms such as BSN node [4], Eco node [21] and SHIMMER [12] were designed.

In spite of such diversity in applications and platform designs, there are some common underlying tasks that occur in all BSN applications, and must be performed efficiently by the sensor platforms. The performance of a platform in executing these tasks eventually determines its effectiveness in the overall application. For example, tasks such as sensing of physiological parameters, basic signal processing, and reliable wireless data transmission are common to most BSN applications. Thus, while evaluating a set of candidate BSN platforms, it is crucial to compare their performance using these basic, representative tasks as benchmarks. However, no such evaluation benchmark currently exists for BSNs.

3. DESIGN COORDINATES

In order to generate a design space of BSN platforms, we need to define a set of orthogonal design coordinates that characterize a BSN platform. These coordinates are essentially features of the platform that determine its performance in BSN applications. For ease of discussion, we divide the coordinates into four groups: *Computation*, *Wireless Communication*, *Energy Source* and *Hardware and Physical Considerations*.

3.1 Computation

We start our design space analysis by focusing on the microcontroller and other related components of the platform that define its computing capabilities.

3.1.1 Processor Performance

In the case of BSNs, performing basic data processing and thresholding operations on the sensor node itself can help the overall application respond faster to changes in the sensed parameter. Further, it avoids transmission of raw data to the gateway device, thus reducing communication energy consumption. However, supporting this increased on-board processing while ensuring timeliness of operation implies high processor utilization. Furthermore, this high performance must be achieved within a highly restricted power budget, since wearable body sensors are typically severely energy-constrained. Thus, for BSNs, it is important to consider the speed and power consumption of the processor together. We consider this combined performance metric as our first design coordinate.

A key observation is that most of the workload of BSN platforms occurs in the form of computations performed on the collected data samples. Hence, measuring the processor speed in units of samples processed per second is much more applicable to BSNs, than traditional metrics such as MIPS (Million Instructions Per Second) or MIPS/Watt. Further, metrics proposed in SenseBench [19] for WSN processors do not capture the tradeoff between processor speed and power consumption. As a result, a new metric needs to be developed for accurate evaluation of the processor performance in BSN platforms.

The proposed metric must measure number of samples processed within a certain power budget and a fixed time interval. This approach motivates the design of the metric SPSW (Samples Processed per Second per Watt) that measures the number of data samples that can be processed in one second per unit watt of power. The exact interpretation of "processing a sample" depends on the application used for evaluation, and thus, the numerical value of this metric is application-dependent. For example, processing a sample might involve querying the sensor ADC, compressing the data and buffering it in memory.

It can be easily verified that the numerical value of SPSW increases with an increase in number of samples processed and reduces with an increase in power consumed or time taken for processing. Although the exact SPSW value for a given processor is application-dependent, the relative trend among processors is uniform across applications with similar computational workload. That is, if processor P_1 has a higher SPSW than processor P_2 in application A_1, it will also have a higher SPSW value for application A_2 as long as A_1 and A_2 are similar in terms of processor workload. This enables an approximate evaluation of a processor to be performed by running a benchmarking task and measuring the SPSW. In order for this evaluation to be accurate, the benchmarking task must be representative of the target application.

We use BSNBench, defined in Section 4.2, to evaluate the SPSW metric for a processor for a given application.

3.1.2 Available Memory

In order to reduce the number of transmissions to the central gateway device, individual sensor nodes must have enough data memory to buffer the collected samples. Further, larger program memory size enables developers to write advanced, memory intensive programs on the sensor node itself, which in turn enables applications to be more responsive. Hence, it is important to consider the available memory space in a BSN platform.

Most of the currently available platforms have multiple types of memory - RAM, ROM and Flash. Further, in some platforms, the flash memory is subdivided into program flash memory and measurement or external flash. The ROM is used for one-time mote configuration code, while the RAM is used for storing run-time variables and the stack. The executable of the application is stored in the program flash memory and hence, its size is restricted by the memory available in this section. Finally, the sensor measurement data can be stored either in the RAM or in the external flash, depending on the developer's choice. However, if stored in the RAM, the data is lost after a reset. The memory specifications for a platform can be obtained using its datasheet.

3.1.3 Signal Processing Capabilities

Basic signal processing, such as peak detection or feature extraction on sensed physiological signals, are often performed on the sensor node itself. For example, an ECG sensor could be programmed to extract the time period and the peak-to-peak amplitude of the sensed ECG waveform. The node can then transmit only these features, rather than the entire waveform, to the gateway device, thus reducing transmission cost significantly. Further, low pass filtering is used in several sensors to eliminate the noise in the measured data.

Most existing platforms perform signal processing algorithms using the main processor itself, while platforms such as Imote2 provide a separate DSP coprocessor. The instruction set of this coprocessor allows for efficient implementations for algorithms such as FFT [16]. Since signal processing computations are generally memory-intensive [7], the available RAM is also an important criterion for evaluation.

Signal processing is essentially a form of computation, and hence the SPSW (Samples Processed per Second per Watt) metric introduced in Section 3.1.1 is used for measuring the signal processing performance of a processor. Like processor performance, the signal processing capability of a platform is also evaluated using the BSNBench suite, introduced in Section 4.

3.2 Wireless Communication

Wireless communication between individual sensor nodes and the central gateway device is a crucial part of BSN applications, and significantly affects the overall power consumption and reliability of the BSN operation. In this section, we discuss the design coordinates pertaining to the radio module of a sensor platform.

3.2.1 Power consumption

As in the case of WSNs, wireless communication in BSNs consumes several orders higher power than computation and hence has a significant effect on the overall energy consumption of a sensor node. Power consumption of radios is generally compared using the current draw ratings in various modes such as *Receive (RX)*, *Transmit (TX)*, *Idle* and *Sleep*. However, this comparison is not suitable for BSN platforms, since it does not consider the duty cycle of the application. For example, a temperature sensor which reports measurements every hour spends much more time in SLEEP state than a real-time ECG sensor in an ICU, which may transmit data every 10 seconds. Clearly, the SLEEP current rating of the radio is of greater importance for the temperature sensor, while the TX current rating is more important for the ECG sensing application.

In this paper, we propose EPC (Expected Power Consumption), a novel metric for evaluating the power consumption of a radio module for a given application. EPC considers the duty cycle of the radio in a given application and accordingly calculates the expected power consumption based on the current drawn in each state. This can be interpreted as a weighted average of the power consumption in each radio state, where the weights are determined based on the time spent by the radio module in that state. Given an application, the developer can consider a representative period of operation, and estimate the intervals spent by the radio module in different states (T_{RX}, T_{TX}, T_{IDLE} and T_{SLEEP}). These estimates are combined with the current draw for each state (I_{RX}, I_{TX}, I_{IDLE} and I_{SLEEP}) to obtain the EPC for that application as:

$$EPC = V_{CC} \cdot \frac{T_{RX}I_{RX} + T_{TX}I_{TX} + T_{IDLE}I_{IDLE} + T_{SLEEP}I_{SLEEP}}{T_{RX} + T_{TX} + T_{IDLE} + T_{SLEEP}};$$

where EPC is the expected power consumption in milli Watts, V_{CC} is the supply voltage, T_{RX}, T_{TX}, T_{IDLE} and T_{SLEEP} are the intervals spent by the radio in the corresponding states, in seconds and I_{RX}, I_{TX}, I_{IDLE} and I_{SLEEP} are the current draws for each state in milli Amperes. Since the EPC value measures the average power consumed by the radio in the application, it enables the total radio lifetime to be easily calculated as:

$$Radio\ Lifetime(s) = \frac{Battery\ Energy(Joules)}{EPC(Watts)}.$$

Akin to the SPSW metric presented in Section 3.1.1 EPC is also application-dependent and can be evaluated using benchmarks. We define a radio benchmarking task in BSNBench, and use it to demonstrate the computation of EPC metric in Section 5.

3.2.2 Reliability

Since BSNs are used for health monitoring and possibly life-critical applications, it is necessary to ensure that the sensor platform provides reliable communication performance. This is especially important since the sensors in a BSN generally use low transmit power in order to conserve energy, and the signal suffers significant attenuation when transmitted across the human body.

The antenna design and the sensitivity of the receiver module in a platform are the main factors that determine its communication reliability. The reliability of the radio module can be measured in terms of the Packet Delivery Rate (PDR), which is the fraction of packets sent by the sender that are successfully received at the receiver. For BSN platforms, it is important to investigate this PDR performance when the platform is deployed on the body [18]. In this paper, we evaluate the reliability performance of platforms through a benchmarking task included in BSNBench. Details of the task are given in Section 4.

3.2.3 Interoperability

Interoperability refers to the seamless integration of a variety of sensors and gateway devices, possibly designed by different manufacturers. It is an important factor in enabling the pervasive adoption of BSN technology and has attracted a lot of attention in recent literature. Since the primary mode of interaction between multiple BSN devices is wireless communication, the communication protocol used in BSNs is the main factor in ensuring interoperability.

Most existing BSN platforms, such as TelosB, Mica2 and BSN node use the IEEE 802.15.4 protocol. This protocol was defined specifically for wireless sensor networks, and provides PHY and MAC layer support to the popular Zigbee standard. However, mobile phones and PDAs, which are generally used as gateway devices, do not support this protocol. Instead, these devices use the Bluetooth standard for implementing Personal Area Network applications. Since these mobile devices are a mature technology, and will not be modified to suit BSN platforms, Bluetooth seems to be the appropriate choice for BSNs. In fact, the Bluetooth consortium has defined a low-energy version for wireless sensors.

Additionally, the *Task Group 6 of IEEE 802.15* is trying to define a low-power, low-frequency standard for Body Area Network (BAN) devices, such as body sensors, mp3 players and other personal wireless devices, and this may become the preferred choice for BSNs. Another possible approach for future BSN platforms is to incorporate multiple radio modules which support different communication protocols, thus providing excellent interoperability. For example, the SHIMMER platform [12] provides connectivity through Bluetooth as well as 802.15.4 protocols.

In this paper, we consider interoperability as a qualitative parameter, and do not define a numerical evaluation metric for it.

3.3 Energy Source

For BSN sensor nodes, lifetime can be defined as the duration for which the sensor can remain operational, before its battery needs to be recharged or replaced. Along with the power consumption of the platform, the attached energy source is an important factor in determining its lifetime. In addition, it also affects the overall form factor of the platform. In this section, we discuss the design coordinates pertaining to the energy source of BSN platforms.

3.3.1 Battery

All currently available BSN platforms run on batteries and most of them use customized battery packs to connect the battery to the main processor board. In most BSN applications, sensor nodes are intended to provide uninterrupted operation over extended lifetimes, which requires use of high capacity batteries. At the same time, use of bulky batteries is discouraged since it increases the overall size and weight of the platform. As a result, it is important to consider the battery capacity (in milli-Ampere-hours, mAh) as well as the dimensions of the battery pack (in mm). Details about the batteries used by a BSN platform can be obtained from its datasheet.

3.3.2 Energy Scavenging

As an energy source for body sensors, energy scavenging is an extremely attractive prospect. Energy scavenging refers to harvesting energy from on-body sources such as thermal gradient, vibrations or ambulatory motion [20]. Significant research efforts have been directed towards investigating different sources of energy and designing efficient scavenging circuits. However, since most scavenging techniques cannot guarantee a continuous supply of power, it is expected that body sensors will retain batteries as the primary energy source at least in the near future.

To the best of our knowledge, none of the current BSN platforms provide support for energy scavenging. As energy scavenging research progresses, and standard, efficient scavenging circuits are developed, future platform designs should provide support for interfacing with energy scavenging circuits. This could be done by providing terminals for interfacing a scavenging circuit with an on-board storage capacitor. This capacitor can then be used to intermittently charge the on-board battery, thus increasing its lifetime.

Like interoperability, support for energy scavenging is also considered a qualitative coordinate in this paper and hence, no numerical metric is devised to measure it.

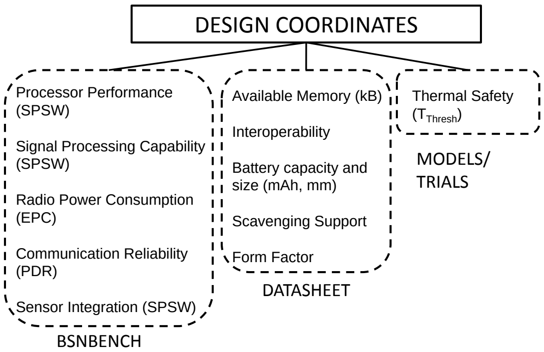

Figure 2: Summary of Design Coordinates and corresponding Evaluation Metrics. Design coordinates are grouped based on the methods used for evaluation.

3.4 Physical and hardware considerations

In this section, we discuss the aspects related to the physical design of BSN platforms.

3.4.1 Thermal Safety

In several BSN applications, the wearable sensor nodes are operational over extended periods of time, which can lead to heating up of the processor or other on-board components. Further, providing a heat sink is impractical for miniature sensors and hence the temperature rise of the platform becomes an important issue. Since the sensors are worn on patients' bodies, it is extremely crucial to consider thermal heating effects and ensure that the temperature of all parts of the platform are always within a safe, approved range.

The *Medical Electrical Equipment: Part 1 - General Requirements for safety (IEC 60601)* standard imposes a temperature threshold for a medical device during its normal operation. This threshold is based on the possible thermal damage to the human skin for a given time of operation of the device at a particular temperature. Thermal damage of the skin is evaluated based on the study performed by Moritz and Henrique [24]. The authors suggest *Threshold Temperature* T_{thresh} of the skin as a metric for thermal damage, where if the skin temperature exceeds the *Threshold Temperature* coagulation occurs leading to blisters on the skin. The parameter is dependent on the time of exposure of the skin to the heat source and is higher for lesser exposure times.

The evaluation of a BSN platform with respect to thermal safety can be performed by model-based verification [24] or through experimental trials. We evaluated the TelosB, Mica2, Imote2 and BSN v3 node platforms by running the BSNBench tasks for an extended period of time and measuring the temperature of the processor using the internal temperature sensor. However, since no significant temperature rise was recorded, these platforms were considered thermally safe.

3.4.2 Form Factor

Since body sensors are intended to be worn by patients over extended periods of time, they must be unobtrusive and easy to wear. Thus, it is important to consider the overall form factor (size and weight) of the platform. Although this design coordinate is much more important for final, market-ready prototypes, it is certainly applicable to research platforms as well, in order to facilitate clinical trials and deployment experiments. As a result, recent BSN-specific platforms such as BSN node v3 [4] and Eco node [21] assign significant importance to the wearability aspect in their design. The size and weight of a platform is specified in the datasheet.

3.4.3 Sensor Integration

Since the primary function of a BSN node is to sense physiological data, it is crucial for a BSN platform to provide efficient sensor interfaces. The sensing function generally occurs through a two-step process: The sensor continually collects data and updates the output of the attached Analog-Digital-Converter (ADC). The microcontroller on the platform then obtains this data by querying the ADC. The efficiency of this querying operation can be measured in terms of speed and power consumption. Hence, we use the SPSW metric, introduced in Section 3.1.1 to evaluate the sensor interface of a platform, and include a Sensor Query task in BSNBench.

Further, in case of BSN platforms, it is important to provide interfaces for connecting medical sensors such as accelerometers, ECG sensors or pulse oximeters. Most currently available BSN platforms have special connection interfaces that work only with corresponding daughter cards. However, in order to facilitate interoperability and easy extensibility, generic connection ports must be provided for use with off-the-shelf medical sensor devices. For example, the BSN node v3 [4] uses a prototype board for connecting any generic sensor to the main node.

A summary of the design coordinates discussed in this section is presented in Figure 2. It can be seen that several design coordinates, such as processor performance and radio power consumption, require a BSN-specific benchmark for evaluation.

4. BSNBENCH: A BENCHMARK FOR BSN PLATFORMS

In this section, we present the design of our proposed benchmarking suite.

4.1 Process of Benchmarking

Benchmarks have commonly been used to measure the performance of processors, embedded systems and other computing systems. Benchmarking a system refers to running a set of tasks or programs on the system and measuring variables of interest, such as power consumption or processing time. For the results of a benchmark to be useful, the tasks must be representative of the final target applications for that system.

The tasks in a benchmark can be complete end-to-end applications, which allows the user to capture the complete performance characteristics of the system for these applications. Such benchmarks are called application benchmarks, and are suitable for systems with a limited number of applications. An example is the 3DMark suite [5] for Windows graphics. The other approach is microbenchmarks, which are comprised of small, atomic tasks that commonly occur in a variety of target applications for the system. Although such benchmarks may not represent a complete application faithfully, they are easier to create than application benchmarks, and provide a good approximation of the system performance in several diverse applications, rather than focusing on a narrow set. For example, BDTI's DSP [8] suite is a widely used microbenchmark in the DSP industry. We choose a microbenchmark approach for BSNBench, since there is a large, growing diversity in BSN applications, and defining an exhaustive application benchmark would be infeasible.

4.2 BSNBench: A BSN-specific microbenchmark

BSNBench is designed as a suite of basic, standalone tasks which serve as building blocks in more complex, full-fledged BSN appli-

Type of Operation	Task	Details	Example BSN Applications
Data Operations	Statistics	Given an array of samples, calculate the mean and standard deviation	Glucose monitoring, Heart rate monitoring
	Out-of-Range	Given an array of samples and a MIN and MAX value, return number of samples outside this range.	Analyzing accelerometer data for posture monitoring
	Differential Encoding	Given an array of samples, compress them using differential encoding.	Temperature recording
Signal Processing	Fast Fourier Transform (FFT)	Perform a 256-point FFT on a given array of samples.	Electromyography (EMG) analysis, Human Motion Tracking
	FIR filtering	Low-pass filter a waveform represented by an array of data points.	Motion analysis [17], Removal of noise from measured data
	Peak Detection	Detect peaks in a waveform.	EMG analysis [22], ECG analysis [10]
Radio Communication	Duty-cycled handshake	Send a packet to base station, sleep. Receive reply on wakeup, sleep again. Repeat in a cycle.	All wireless BSN applications
	Reliable communication	Send given number of packets to base station, from various locations on human body.	All applications with gateway device on body
Sensor Interfacing	Sensed data query	Query on-board sensor and collect given number of data samples.	All sensing applications

Table 1: Composition of BSNBench: Tasks included in BSNBench

cations. These tasks were chosen by investigating diverse BSN applications, breaking them down into independent tasks, and selecting the important commonly occurring tasks. The tasks included in BSNBench suite are shown in Table 1, along with the BSN applications that they occur in. It is clear that the selected set of tasks covers several classes of BSN applications and hence is representative of the platform workload in BSNs.

Since BSNBench is intended to profile all the components of a platform, it includes system-level tasks such as sensor query, purely computational tasks such as differential encoding, basic signal processing tasks such as FFT, and radio communication tasks such as on-body PDR measurement. These tasks are grouped into types based on the aspect of a platform that they help to evaluate.

For evaluating a platform using BSNBench, each task is run on the platform and its performance is measured in terms of a set of quantitative variables of interest, such as power consumption or execution time. The metrics used for measuring these outcomes are based on the design co-ordinates discussed earlier, in Section 3. Thus, the benchmark compliments the design space analysis and helps provide a quantitative evaluation of several aspects of BSN platforms. Table 2 shows the link between design co-ordinates, metrics and the different types of tasks proposed in the benchmarking suite. We now discuss the different types of BSNBench tasks in detail:

4.2.1 *Data Operations*

The tasks of this type are purely computational, and comprise of analysis or manipulation of a given set of data samples. These are aimed at evaluating the performance of the processor in terms of power consumption and execution time. These tasks commonly occur during the basic pre-processing of the sensed physiological data or as the computations required to serve the queries issued to the sensor. For example, the *Out-of-Range* task is used to answer the query "How many readings of blood sugar level were higher than 150 mg or lower than 70 mg?". As another example, *Differential Encoding* can help compress the raw data obtained from a body temperature sensor, since successive temperature readings vary only slightly. Lastly, the *Statistics* task can be used in Heart Rate monitors to obtain the average heart rate and its variability.

4.2.2 *Signal Processing*

These tasks test the performance of the platform in executing basic signal processing algorithms. The sensor readings are interpreted as samples of a waveform, and are passed into the task as a time domain signal. Signal processing is used in several BSN applications, either for removal of noise by filtering or for extracting features from the collected data. For example, the *Peak Detection* task helps to identify the occurrence of peaks in an ECG waveform. Lowpass FIR filtering is used for rejecting high-frequency noise, while FFT analysis is used for frequency domain analysis.

4.2.3 *Radio Communication*

These tasks are used for benchmarking the radio power consumption and reliability of a platform. In most BSN applications, the primary functions of the sensor radio are transmitting data to the gateway device, and receiving queries issued by the device. When not transmitting or receiving data, the radio is set to SLEEP mode in order to conserve energy.

This functionality of the radio is represented by the *Duty-cycled Handshake* task in BSNBench. In this task, the radio sends a packet to the base station and then switches to SLEEP state for a predefined interval T. On waking up, it receives the reply from the Base Station, and again goes into SLEEP mode for time T. This cycle is repeated periodically. The parameter T controls the duty cycle of the radio and can be swept over a range to evaluate the EPC of the radio at different duty cycles. This variation in duty cycle is essential for benchmarking diverse BSN applications. For example, an elderly care monitoring application might buffer the sensed data for a long period of time, compress it and transmit it infrequently. On the other hand, sensors used to monitor an athlete's performance during exercise might stream data almost in real-time.

In order to evaluate the performance of radio modules in terms

Design Co-ordinate	BSNBench Section	Measured Quantities	Metrics Used
Processor Performance	Data Operations	Power consumption, Time required	SPSW
Signal Processing Capabilities	Signal Processing	Power Consumption, Time required, Memory footprint,	SPSW kB
Radio Power Consumption, Reliability	Radio Communication	Power consumption, Received packets	EPC PDR
Sensor Integration	Sensor interfacing	Power consumption	SPSW

Table 2: Design co-ordinates and metrics evaluated using BSNBench

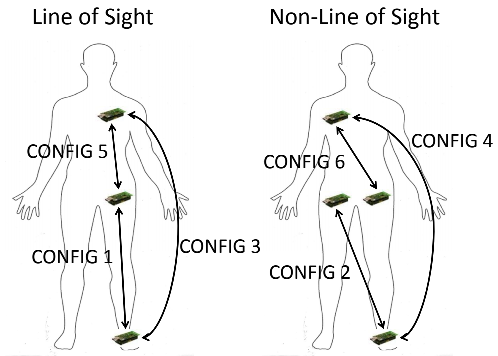

Figure 3: Depiction of the six configurations used to evaluate the On-body PDR for different platforms, in the radio reliability task of BSNBench.

of on-body PDR [18], we define the *Reliability* task. In this task, a pair of sensor nodes are deployed on the human body in 6 different configurations, as shown in Figure 3. Three of these configurations (*Left Foot to Left Hip*, *Left Foot to Left side of chest* and *Left Hip to Left side of chest*) are chosen as Line-of-Sight (LOS) configurations while the remaining three (*Left Foot to Right Hip*, *Left Foot to Right side of chest* and *Left Hip to Right side of chest*) are non-LOS configurations. Each node then sends 500 packets to the other, and the number of successfully received packets is recorded. This is repeated for two postures of the subject: standing and sitting. In this paper, we present results only for indoor environments. Future work will include evaluation in outdoor environments with lesser multipath effects.

4.3 Sensor Interface

The *Sensed Data Query* task involves the microcontroller querying the sensor ADC for its current reading, and storing it in the RAM as a variable. This simple task occurs in almost every BSN application, and is the first step in the flow of information from the patient's body to the caregiver or physician. In the current version of BSNBench, we perform this task using the on-board sensors of a platform. However, it is also important to evaluate the performance with external medical sensors such as ECG or SpO2, which will be included in the next version of BSNBench.

4.4 Implementation Details

Since most existing BSN platforms support TinyOS, we implemented the tasks in BSNBench as standalone applications in TinyOS 2.x. Each data operation task is run 10^5 times inside a FOR loop, and the average time per iteration is calculated. The radio tasks use the *ActiveMessageC* interface provided in the TinyOS library for sending and receiving packets over the radio. In the *Duty-Cycle Handshake* task, the TinyOS Low-Power-Listening (LPL) interface was used to set the radio to sleep for a given interval. Finally, the *Sensed Data Query* task uses the *DemoSensorC* interface.

5. BENCHMARK EVALUATION METRICS AND RESULTS

In this section, we present the results obtained by evaluating four commonly used sensor platforms - TelosB [1], Mica2 [2], Imote2 [3] and BSN node v3 [4] - using BSNBench.

5.1 Data Operation

These tasks are used to evaluate the processor performance of the platforms, and the results are shown in Table 3. The Imote2 platform uses a frequency-scaled processor, and we investigate its performance at 13 MHz as well as 104 MHz.

It was observed that the higher MIPS processors in Mica2 and Imote2 outperformed the TelosB and BSN v3 motes in terms of execution time. However, the high processor speed also leads to greater power consumption. The SPSW metric, defined in Section 3.1.1 effectively captures this tradeoff and assigns a resultant score to each platform. From the SPSW values, it can be seen that the choice of platform is strongly dependent on the application. The Mica2 gives the highest SPSW rating for *Statistics* and *Out-of-Range* tasks, while TelosB performs best in *Differential Encoding*. The performance of BSN node is comparable to TelosB in most tasks, which is expected since they use the same microcontroller. The Imote2 platform at 104 MHz was observed to consistently have a higher SPSW rating than at 13 MHz, since the speed increases by a factor of 8, while power consumption is less than twice.

The Imote2 platform was observed to draw excessive current (above 30 mA) compared to the other platforms, which is due to multiple on-board peripherals being powered up during the active mode of the processor. Further, in tasks involving floating point arithmetic, the Imote2 platform was observed to perform consistently slower. Additional experiments using a Linux-based Imote2 enabled us to attribute this performance degradation to the TinyOS 2.x compiler for Imote2.

The results from this section show that a comparison of processors based on MHz rating and power consumption figures is not accurate for BSN platforms. Instead, the relative performance among platforms is highly dependent on the type of application.

5.2 Signal Processing Operations

The TelosB, Mica2 and BSN v3 nodes use the main processor for Signal Processing algorithms as well, while the Imote2 has a Wireless MMX DSP coprocessor which is optimized for operations such

Platform (Processor)	Task	Sample Size	Consumed Power (mW)	Execution Time (ms)	SPSW (samples/mJ)
Mica2 (ATmega 128L)	Statistics	1000	22.98	450	96.7
	Out-of-Range	200	23.19	5	1724.88
	Differential Encoding	1000	23.706	2.5	16873.36
TelosB (MSP 430)	Statistics	1000	4.98	2298	87.38
	Out-of-Range	200	5.4	35	1058.2
	Differential Encoding	1000	5.31	5	37664.78
Imote2 (13MHz) (PXA271 Xscale)	Statistics	1000	162	2200	2.81
	Out-of-Range	200	157.05	31.6	40.3
	Differential Encoding	1000	161.1	5.1	1217.12
Imote2 (104 MHz)	Statistics	1000	269.1	280	13.27
	Out-of-Range	200	269.55	4	185.49
	Differential Encoding	1000	263.7	0.664	5711.1
BSN v3 (MSP 430)	Statistics	1000	6.7712	2070	71.67
	Out-of-Range	200	6.4768	35	882.27
	Differential Encoding	1000	6.55	5.5	27758

Table 3: BSNBench Data Operation Evaluation Results

Platform	Task	Sample Size	Power Consumed (mW)	Execution Time (ms)	SPSW (samples/mJ)	RAM Footprint (kB)
Mica2	FIR	100	23.01	0.4	10864.84	843
	FFT	256	-	-	0	4425*
	Peak Detection	200	21.66	24	384.73	1048
TelosB	FIR	100	5.349	0.25	74780.33	845
	FFT	256	5.1	427.6	117.4	4404
	Peak Detection	200	5.58	94	381.3	1051
BSN v3	FIR	100	6.3296	0.224	70530.29	845
	FFT	256	6.5136	425	92.47	4419
	Peak Detection	200	6.808	91	322.83	1049
Imote2 (13MHz)	FIR	100	165.15	0.22	2752	1634
	FFT	256	162	95.5	16.504	6604
	Peak Detection	200	156.6	154	8.29	1866
Imote2 (104 MHz)	FIR	100	288.4	0.028	12383	1634
	FFT	256	279	11.8	77.76	6604
	Peak Detection	200	270.9	19.8	37.28	1866

* This is greater than the RAM size (4kB) of Mica2.

Table 4: BSNBench Signal Processing Evaluation Results

as FFT. In our experiments, the coprocessor enhancements were enabled in TinyOS, but in order to ensure fairness, the code was not hand optimized for the MMX processor instruction set [16].

Table 4 shows the RAM footprint and SPSW for the signal processing tasks performed on each platform. The 256-point FFT task could not be implemented in Mica2 motes due to insufficient RAM. The Imote2 platform was seen to outperform TelosB and BSN v3 nodes in the FFT task in terms of execution time due to use of the coprocessor. However, the excessive power consumption of the platform lowers its SPSW rating. Again, since Peak Detection involved extensive floating point arithmetic, the Imote2 performance was lower than expected. In signal processing tasks also, the Imote2 platform shows higher SPSW performance at 104 MHz than at 13 MHz.

5.3 Radio Communication

All of the chosen platforms use the Chipcon CC2420 radio which implements the IEEE 802.15.4. protocol. However, the antenna designs on these platforms are significantly different and this leads to differences in their overall communication performance.

The *Duty-Cycled handshake* task was performed with 2 different values of sleep interval: 2s. and 5s., while the packet send and receive times were both 110 ms. This leads to duty cycles of 10% and 4.21% respectively. The EPC obtained for the four platforms is shown in Table 5. In order to provide a uniform comparison, the lifetime for each platform is calculated for an energy source comprised of 2 AAA E92 [6] cells, with average discharge voltage of 1.2 V. The capacity of each cell is taken as 1050 mAh for current draw below 100 mA.

For the reliability task, the sender power was set at -25 dBm, and all the readings were taken in an indoor lab environment. The results of the experiment are shown in Figure 4. It was observed that the achieved PDR varied significantly over the six different configurations, with a much more reliable connection obtained in Line-of-Sight (LOS) configurations than the non-LOS ones. Further, there was a significant difference in the PDR for the different platforms, with the BSN v3 node being unable to receive any pack-

Platform	Duty Cycle (%)	EPC (mW)	Lifetime (hrs)
Mica2	10	9.41	267.6
	4.21	8.79	286.6
TelosB	10	3.24	777
	4.21	1.69	1486.5
BSN v3	10	4.128	610.4
	4.21	2.64	954.5
Imote2	10	117.28	21.4
	4.21	111.92	22.5

Table 5: BSNBench: Duty-Cycled Handshake Results

Platform	Consumed Power(mW)	Time (s)	SPSW * (samples/mJ)
Mica2	5.46	3.5	272.88
TelosB	10.47	5	366.3
BSN v3	6.586	2.35	646.11
Imote2 (13 MHz)	156.15	2.86	22.391
Imote2 (104 MHz)	289.8	0.368	93.768

* 10000 samples were collected.

Table 6: BSNBench Sensor Query Evaluation Results

ets at -25 dBm[2]. This is due to the difference in the antenna design of the platforms. For example, the BSN v3 node has a miniaturized chip antenna that provides a 5 m. range at 0 dBm [4], while TelosB has an inverted-F antenna with an indoor range of 20m.

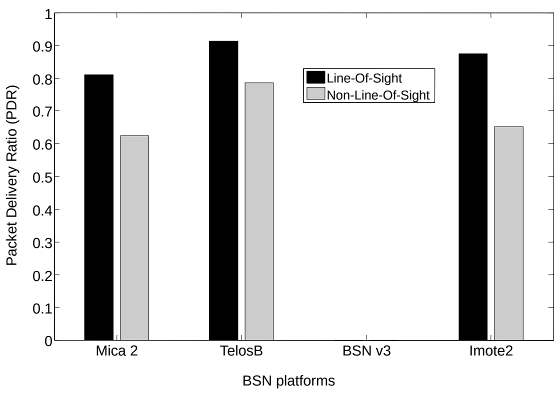

Figure 4: PDR observed in the reliability experiment for sending power = -25 dBm. At this power, no packets were received in case of BSN v3 node.

5.4 Sensor Integration

In the Sensor Integration task, the inbuilt temperature sensors provided on each platform were queried 10^4 times and the data was stored in the RAM. The results for different platforms were measured in terms of SPSW and are shown in Table 6. The BSN v3 node significantly outperforms the other platforms in this task, indicating an efficient sensor interface.

[2]The power had to be increased to -7 dBm to achieve performance comparable to other platforms.

We note that our key contribution in BSNBench is the design of the benchmarking tasks and not the specific implementation. The test results pesented in this section are operating system dependent, but similar analysis can be carried out for benchmarking platforms using different operating systems.

6. DESIGN SPACE EXPLORATION

In this section, we illustrate the process of design space exploration through two case studies representative of typical BSN applications:

1. Continuous Glucose Monitoring [13] (CGM): A long term monitoring application where a sensor measures the blood glucose level and transmits the measured values to a gateway device. The sensor node also checks if the blood glucose level goes above or below user-defined thresholds, and sounds an alarm. The sensor is intended to be compact and easy-to-wear, and must run continuously for 4 days on 2 AAA batteries.

2. Epileptic Seizure Detection (ESD): An application to detect the onset of seizures in epileptic patients. A wearable ECG sensor node collects ECG data of the patient and filters it to remove noise and signal artifacts. The filtered signal is then passed through a peak detector which identifies the peaks to compute the RR intervals. These are then transformed to the frequency domain using FFT. Finally, the FFT coefficients are sent to the gateway device for further processing.

We consider the set of four platforms that are used in the benchmark evaluation section (Section 5) and choose the most suitable platform for each application.

The long term monitoring and data reporting nature of CGM application imposes a strict limit on power consumption. This requirement is mapped to an upper bound on the EPC value for the sensor radio. We assume that 2 Energizer E92 AAA cells are used to power the node. From [6], the service time of each cell is 150 hours for current draw in the order of 10 mA. Using average discharge voltage as 1.2 V, the total energy available is:

$$Battery\ Energy = 2 \cdot 10\ mA \cdot 150\ hrs \cdot 1.2\ V \cdot 3600\ s/hr = 12.96\ kJ$$

We assume that radio duty cycling is used to conserve power, and the radio spends 96% time in SLEEP mode, thus giving a duty cycle of 4%. Further, since communication is the main factor of energy consumption, we assume that 95 % of the total available energy from the battery is used by the radio. Now, since the node must run for 4 days without battery replacement, the bound on the EPC of the radio is given by:

$$EPC \leq \frac{0.95 \times Battery\ Energy}{4 \times 3600 \times 24} \leq 35.62mW$$

Further, we restrict the form factor of the platform to 50×50×50 mm^3, to ensure unobtrusive operation when worn by the patient.

Next, we consider each of these constraints and eliminate platforms which do not comply. The EPC constraint eliminates the Imote2 platform (Table 5) while the form factor constraint eliminates the Mica2 and TelosB motes, leaving the BSN v3 node as the most suitable platform for the CGM application.

For the ESD application, the signal processing requirements , especially the 256-point FFT, enforce a constraint on available memory. This constraint eliminates the Mica2 platform due to its low RAM resource (Table 4). Further, we assume that the PDR for such a critical medical application is required to be at least 0.7 at -25 dBm transmit power. The BSN v3 node is eliminated by this constraint due to its low PDR performance. At this stage, relative priorities can be assigned to the remaining design coordinates. If the highest priority is assigned to processor performance (SPSW), the TelosB platform would be chosen since it outperforms the Imote2 platform for both, Peak Detect as well as FFT tasks.

From these two case studies we observe that for different types of applications the design space exploration leads to different solutions. These insights on the performance of different platforms in a BSN environment cannot be obtained from their datasheet information, which emphasizes the need for our proposed framework.

7. CONCLUSION

In this paper, we presented an evaluation framework for determining the most suitable sensor platform for a given BSN application. The framework incorporates determination of the design space for BSN platforms based on key design coordinates that were identified in this paper. To aide in this process, we developed a benchmarking suite applicable to BSNs, called BSNBench. We then used BSNBench to map four commonly used sensor platforms in the proposed design space. Further, we considered two different types of BSN applications and demonstrated how our framework incorporates application-specific requirements into the platform selection process.

The proposed framework is an initial step towards creating a standardized evaluation method for BSNs, and will be further extended in our future work by adding tasks related to data security and privacy, medical sensor integration and other such aspects that are crucial for BSNs.

Acknowledgments

The work of authors from ASU was supported in part by National Science Foundation grant CNS #0831544.

8. REFERENCES

[1] http://www.xbow.com/Products/Product_pdf_files/Wireless _pdf/TelosB _Datasheet.pdf.

[2] http://www.xbow.com/products/product_pdf _files/wireless _pdf/mica2 _datasheet.pdf.

[3] http://www.xbow.com/Products/Product_pdf_files/Wireless _pdf/Imote2_Datasheet.pdf.

[4] http://ubimon.doc.ic.ac.uk/bsn/a1875.html.

[5] http://www.futuremark.com/products/3dmark03/.

[6] http://data.energizer.com/PDFs/E92.pdf.

[7] A. Banerjee et al. Challenges of implementing cyber-physical security solutions in body area networks. In *Proc. of Intl. Conf. on Body Area Networks*, 2009.

[8] J. Eyre, J. Bier, and B. Inc. DSP processors hit the mainstream. *Computer*, 31(8):51–59, 1998.

[9] M. Hempstead, M. Welsh, and D. Brooks. TinyBench: The case for a standardized benchmark suite for TinyOS based wireless sensor network devices. 2004.

[10] H. in't Veld, O. M.H.A., S.C.M.A., and et al. Context aware algorithm for epileptic seizure detection. In *Awareness deliverables*, 2005.

[11] P. Kulkarni and Y. Öztürk. Requirements and design spaces of mobile medical care. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):30, 2007.

[12] K. Lorincz, B. Kuris, S. Ayer, S. Patel, P. Bonato, and M. Welsh. Wearable wireless sensor network to assess clinical status in patients with neurological disorders. In *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007.

[13] J. Mastrototaro. The MiniMed continuous glucose monitoring system. *Diabetes technology & therapeutics*, 2(1, Supplement 1):13–18, 2000.

[14] J. Mihel and R. Magjarevic. FPGA based two-channel ECG sensor node for wearable applications. In *4th European Conference of the International Federation for Medical and Biological Engineering*, pages 1208–1211. Springer.

[15] S. Mysore, B. Agrawal, F. Chong, and T. Sherwood. Exploring the Processor and ISA Design for Wireless Sensor Network Applications. *vlsid*, pages 59–64, 2008.

[16] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. Imote2: Serious computation at the edge. In *International Wireless Communications and Mobile Computing Conference*, pages 1118–1123, 2008.

[17] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. Bula, and P. Robert. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *IEEE Transactions on Biomedical Engineering*, 50(6):711–723, 2003.

[18] A. Natarajan, B. Silva, K. Yap, and M. Motani. To hop or not to hop: Network architecture for body sensor networks. In *IEEE SECON*, 2009.

[19] L. Nazhandali, M. Minuth, and T. Austin. SenseBench: toward an accurate evaluation of sensor network processors. In *Workload Characterization Symposium, 2005. Proceedings of the IEEE International*, pages 197–203.

[20] J. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive computing*, 4(1):18–27, 2005.

[21] C. Park, J. Liu, and P. Chou. Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring. In *IPSN 2005.*, pages 398–403.

[22] R. Ramachandran, L. Ramanna, H. Ghasemzadeh, G. Pradhan, R. Jafari, and B. Prabhakaran. Body sensor networks to evaluate standing balance: interpreting muscular activities based on inertial sensors. In *Procs. of the 2nd Intl. Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, page 4. ACM, 2008.

[23] A. Van Halteren, R. Bults, K. Wac, D. Konstantas, I. Widya, N. Dokovsky, G. Koprinkov, V. Jones, and R. Herzog. Mobile patient monitoring: The mobihealth system. *The Journal on Information Technology in Healthcare*, 2(5):365–373, 2004.

[24] S. Weininger et al. Factors to consider in a risk analysis for safe surface temperature. In *Product Safety Engineering, 2005 IEEE Symposium on*, pages 83–91, Oct.