31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Motion-based Wake-up Scheme for Ambulatory Monitoring in Wireless Body Sensor Networks

Isaac Pek, Agustinus Borgy Waluyo, Wee-Soon Yeoh, Xiang Chen
Institute for Infocomm Research, A*STAR, Singapore
Email: {ipek,awaluyo,wsyeoh,xchen}@i2r.a-star.edu.sg

***Abstract*— Given that wearable sensors that are attached on patients for the purpose of continuous real-time medical monitoring typically need to remain operational for periods of up to 24 hours before a battery change or recharge, power preservation schemes play a critical role in minimizing any possible disruption to a patient's daily activities. In this paper, we propose a motion-based wake-up scheme, a feature which combines motion detection with existing power preservation schemes in order to achieve a balance between energy saving and data timeliness, particularly in critical situations. As a showcase, we have integrated this feature with a healthcare application and demonstrate the capability of the scheme to deal with critical events, e.g., when a patient falls down from the bed. This showcase affirms the effective uses of our proposed motion-based wake-up scheme.**

I. Introduction

THE current lightweight design of modern sensors has allowed such sensor devices to be deployed as part of a Wireless Body Area Network (WBAN) which can be easily worn on a patient's body. With an array of heterogeneous sensors within the WBAN collecting a variety of physiological data in real-time, caregivers will be able to monitor patients' health conditions. Continuous patient health monitoring not only facilitates real-time diagnosis of a patient's current health status, but also supports early detection of symptoms related to various medical conditions which are not easily detected and require long-term monitoring to be detectable, e.g. cardio-vascular disease, etc.

With a rising number of medical applications utilizing data collected from WBAN sensors, the use of a middleware layer to simplify application development is considered ideal. For this purpose, we have developed a lightweight middleware called SLIM [5]. Our SLIM middleware supports various features such as, timely data acquisition, sensor resource management, sensor power preservation and secured wireless data transmission.

The purpose of our work described in this paper is to balance sensor power preservation schemes with a sensor node's ability to transmit data in a timely manner, ensuring that higher level applications are able to detect critical events in real-time, even while power preservation is on-going. We have extended this new motion-based wake-up scheme from our previous work on our SLIM middleware.

In this paper, we briefly provide an overall background of our work in Section II. A more in-depth description of the proposed motion-based wake-up scheme can be found in Section III, followed by a showcase of how a health monitoring application utilizes the newly implemented feature in Section IV. Finally, Section V concludes the paper.

This work was supported by Science and Engineering Research Council (SERC) of the Singapore Agency for Science, Technology and Research (A*Star) on Embedded & Hybrid Systems II (EHS-II) programme under grant 052-118-0058.

II. Background

A. SLIM Middleware

SLIM is designed to support multi-modal sensors; allowing different sensors types (e.g. accelerometers, ECG and Sp02) to co-exist and transmit to a common gateway. SLIM is divided into two portions; the Lower Middleware resides in the sensor nodes, while the Upper Middleware is deployed on a mobile device such as a PDA.

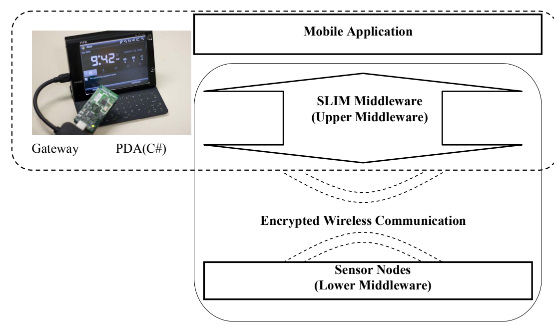


Figure 1. Overview of SLIM Middleware.

As shown in Figure 1, the Lower Middleware (sensor nodes) programs are coded in nesC using TinyOS 2.x, while the Upper Middleware is programmed in C#. The upper and lower layers communicate over wireless IEEE 802.15.4 (Zigbee) protocol, passing through a base station node. This node utilizes runs the BaseStation program, provided by TinyOS. Our sensor hardware is a set of EHSII sensors, developed by the Embedded & Hybrid Systems II programme in A*Star which have similar specifications as the Imperial College BSN motes [2,3].

B. Existing power preservation schemes in SLIM

Wireless radio transmissions generally consume significant amounts of power, and are the primary cause of battery power depletion as the radio consumes more power than the microcontroller [4,6,7].

When deployed in medical monitoring applications, wireless sensor nodes typically need to transmit data on a constant basis to higher-level applications residing on devices which have a higher processing power for more complex decision making or data logging purposes. Frequent radio transmission, however, depletes a sensor's battery rapidly, and it is logical for the CC2420 radio transceiver to be deactivated when it is not needed. For example, when a patient has been lying down for an hour, constant data transmission from certain accelerometers sensors may not be required, the application can thus activate the power preservation scheme on these sensors to conserve energy.

The power preservation method we have included as part of SLIM is called sleepRadioToggle and is illustrated in Figure 2. It incorporates a time-based wake-up scheme where the CC2420 radio periodically toggles between the active and inactive states. While the radio is toggled, the MSP430 microcontroller remains operational to continuously sample data from the sensor. Only sensor data that is sampled when the radio is active is transmitted. Radio transmission only takes place at regular intervals, instead of all the time, thus saving energy while providing a window for applications to process the data and terminate any on-going toggling via a radio message to the sensor node, if necessary.

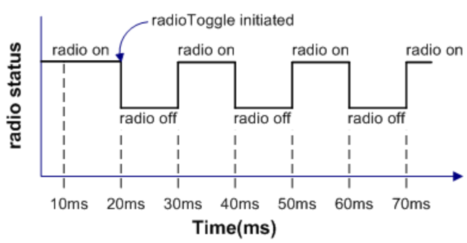


Figure 2. In a sleepRadioToggle power preservation scheme, the radio is activated / deactivated at periodic intervals.

The power consumed by an EHS-II accelerometer sensor node (comprising of the MSP430 microcontroller, CC2420 radio transceiver, accelerometer sensor board and battery board) sampling at 25 Hz both with and without wireless data transmission is shown in Table 1. It can be calculated that during power preservation, the power consumed by the sensor node is ~78.3 mW. In terms of battery operating duration with the same setup and continuous data transmission, the 800 mAh lithium polymer battery is able to sustain operation for ~21 hrs. With our power preservation scheme applied, the operating duration of the accelerometer sensor node can be extended to ~33 hrs.

Radio Status	**Power consumption**
Radio toggled active (data transmitted)	123.75 mW (37.5 mA * 3.3 V)
Radio toggled inactive (no data transmitted)	33 mW (10.0 mA * 3.3 V)

Table 1. Power consumption during active and no transmission periods (measured from EHS-II accelerometer sensors operating on a 3.3 V, 800 mAh lithium polymer cell).

The main drawback of the sleepRadioToggle scheme is that data cannot be transmitted when the radio is turned off. Therefore, if a critical event were to occur when the radio is inactive, data pertaining to such an event would never reach the application layer since the sensor node is unable to transmit any data. Take for example, a scenario where a patient is initially lying down on a bed and the accelerometer sensors he/she is wearing are toggling their radio to conserve power. At some point, the patient falls off the bed and lies unconscious on the floor. Although the fall is detected by the sensors, data cannot be transmitted because the fall occurred when the radio was toggled inactive. Subsequently, when the sensor node wakes up and resumes data transmission, the application would detect that the patient was still lying down and would have no knowledge that the patient had previously fallen. To deal with such a situation, a wake-up scheme is necessary to ensure that sensor nodes are able to wake up and resume normal data transmission in a timely manner.

C. *Related Work*

One approach to waking up a sensor node is to overlap the radio wake up timings for sensors in the network, allowing a node to wake its neighbors should the need arise [11]. This method however, is not applicable in our case as a radio message is needed to wake up the sensors, and our nodes cannot receive any radio messages while their radio is inactive. Other methods involve the use of a hardware RF module which wakes up a node when the RF module receives a signal [12]. This approach might be feasible in our situation, but cannot achieve the same degree of power conservation considering the RF module still draws power from the sensor's batteries. We therefore propose a motion-based wake up scheme to deal with critical events during a radio's inactive period, which is further described in the next section.

III. MOTION-BASED WAKE-UP SCHEME

Our approach to address the problem of not receiving data is to delegate minor data processing to the sensor whenever power preservation is active. This approach uses the MSP430 microcontroller (which remains active) on the sensor nodes to independently determine if the degree of motion (measured in terms of acceleration) of the accelerometer crosses a motion threshold that is pre-specified by the application. If the threshold is exceeded, the radio is re-activated immediately and the sampled data is transmitted out to the receiving base station node. This eliminates the need for a radio signal instructing the sensor node to wake up. A flow-chart describing the motion-based wake-up scheme, when applied for tri-axial accelerometer sensors is shown in Figure 3.

The motion-based wake-up scheme firstly determines the net acceleration from data sampled by each tri-axial accelerometer. This initial value is subsequently updated whenever a new set of accelerometer data is received (1 reading from x, y and z axis each). The amount of deviation of the newly received readings from the net acceleration is checked against a motion threshold to decide if the accelerometer sensor should wake up. The larger the amount of force applied on the accelerometer sensor, the larger the resulting deviation.

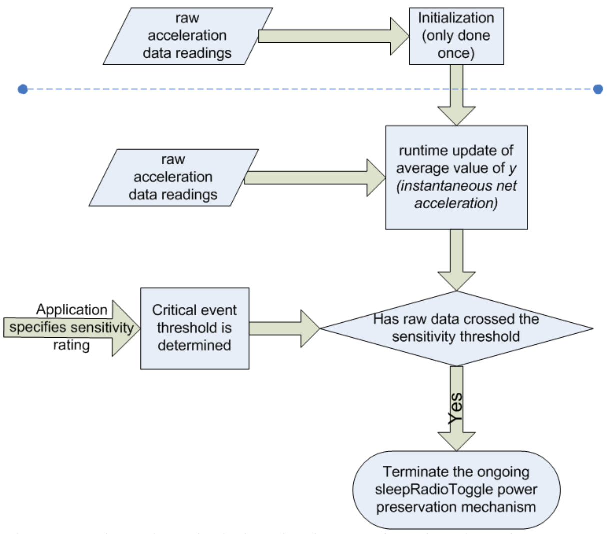


Figure 3. Flow-chart depicting the degree of motion detection process for accelerometer sensors.

```
each time 3 data (x, y and z axis) arrive
    check if the net acceleration ā_net(t) has been initialized
    (init.)
                    if ā_net(t) has not been init.:
            add the square of each of the new data readings
            a_net(t) to ā_net(t) and divide based on the
            length of computation
              if ā_net(t) has been init:
            runtime update of the ā_net(t)
    compare the fluctuation of the new set of data and the net
    acceleration against the critical threshold, i.e.,
    (a_net(t) − ā_net(t))^2 > Th
        if the fluctuation is larger than the critical threshold:
            terminate the timer regulating the toggling and
            reactivate the sensor node's radio
```

Figure 4. Pseudo-code describing the degree of motion detection process.

The application controls the wake-up scheme through a sensitivity rating (expressed in terms of a percentage value) for each accelerometer. This rating is transmitted via a radio message to each accelerometer sensor node before power preservation commences. Each node uses the rating value in the message to determine its own motion threshold. Any data that crosses this threshold is transmitted immediately when the radio is re-activated, as shown in Figure 5.

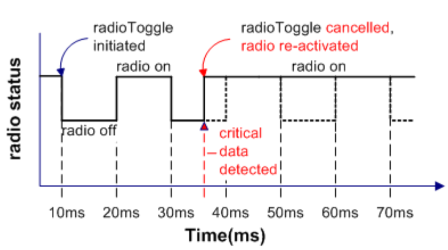


Figure 5. Motion-based wake-up scheme reactivating the radio once data has been detected has crossed a critical threshold.

Figure 6 shows a test where an accelerometer sensor was moved with varying degrees of motion (large/small) at periodic intervals. Each fluctuation in the upper figure corresponds to a movement of the accelerometer sensor, while spikes represent the larger movements. As shown in the lower figure, minor movements do not cross the threshold, whereas vigorous movements will exceed the pre-set motion threshold.

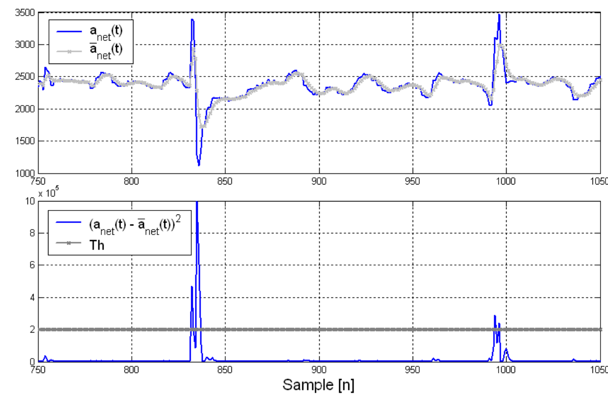


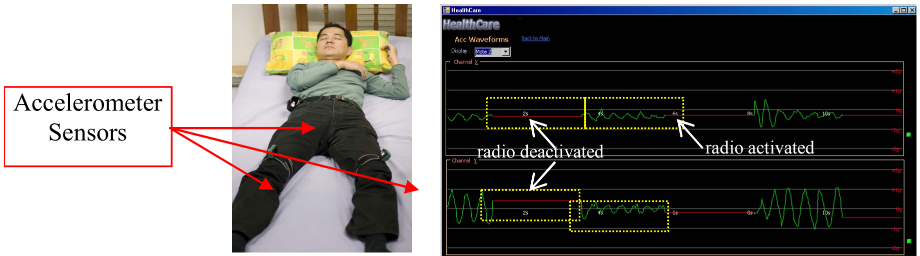
Figure 6. Motion-based wake-up scheme detects the crossing of a threshold under vigorous accelerometer movement, but not under minor movement.

IV. Integrated with a Healthcare Application

This section discusses the integration of our SLIM middleware with a healthcare application deployed on a mobile device (PDA), performing real-time patient activity classification for the purpose of ambulatory monitoring. This allows a patient's current health status to be remotely monitored by staff in the hospital. The application utilizes 3 tri-axial (x, y and z-axis) accelerometer sensors operating at a sampling frequency of 25 Hz. Each sensor transmits a 36 byte long radio packet comprising of a header, trailer and a 26 byte data payload. 16 bytes of the data payload comprise of 8 data readings (2 bytes each), while the remaining 10 bytes contain housekeeping information such as the mote ID, packet number, sensor type and channel numbers.

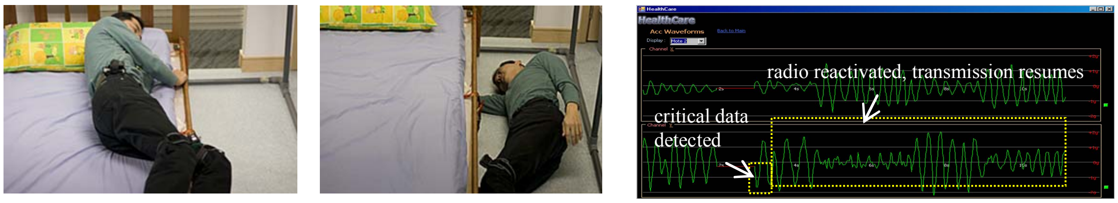
The accelerometer sensors are mounted on a male subject's left thigh, right thigh and torso respectively. The application classifies the activity of the patient into either, standing, sitting, lying down or walking, based on the aggregation of the acceleration data received [9]. As shown in Figure 7a, once the application has detected that the subject has been sleeping for a period of time (about 1 hour), the application instructs the sensors to initiate sensor power preservation via the sleepRadioToggle scheme provided by SLIM. The resulting waveform of data that arrives from the accelerometer sensor is displayed in Figure 7b.

In this scenario, the motion-based wake up scheme was preset by the application with a sensitivity rating of 40%. At this setting, it can be observed that the critical event detection does not trigger when minor movements occur as the subject changes positions during his sleep as depicted in Figure 8a.



(a) Subject is detected by the application as 'Lying Down'. **(b) Sensor is in power preservation mode (x and y axis shown only)**

Figure 7. Prolonged activity detected and sensor's radio transceiver is turned to power preservation mode.



(a) sensor does not activate on minor motion (b) Bodily impact with the ground and the sensor's radio wakes up (c) Accelerometer data transmission resumes when large motion is detected (x and y axis shown only)

Figure 8. Showcase of motion-based wake-up scheme in real-life application.

At some point in time, the subject rolls off the bed and impacts against the floor as shown in figure 8b. The net acceleration of the patient as he rolls from the bed to the floor exceeds the motion threshold and the power preservation scheme is terminated. Once terminated, the sensor node's radio transceiver is reactivated and data transmission resumes as shown in Figure 8c.

This scenario has highlighted the significance of such a feature when used to balance maintaining sensor power preservation while ensuring continued detection at the sensor node level.

V. Conclusion and Future Work

The need for continuous real-time monitoring to diagnose or treat the various harmful medical conditions that can affect the human body have shown the important role that wireless wearable sensors play in such applications. For applications that depend on timely data arrival to function optimally, it is important for sensor nodes to be able to balance power saving schemes with timely data acquisition.

In this paper, we have presented a novel motion-based wake-up scheme which allows our SLIM middleware to ensure timely delivery of acceleration data to applications in the application layer even when a sensor node's radio transceivers are deactivated to conserve battery power. This feature is able to resume the transmission of sensor data when the sensor detects a large degree of motion.

For our future work, we will extend the critical event detection feature to provide support for more types of sensors such as the ECG or Sp02 sensors.

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of Sensor Networks," Computer, vol. 37, no. 8, 2004, pp. 41–49.

[2] B. Lo, S. Thiemjarus, R. King, and G-Z. Yang, "Body Sensor Network – A Wireless Sensor Platform for Pervasive Healthcare Monitoring," in *Proc. of the 3rd International Conference on Pervasive Computing*, 2005, pp.77–80.

[3] B. Lo and G.Z. Yang, "Architecture for Body Sensor Networks", IEE Proceedings of the Perspective in Pervasive Computing, 2005, pp.23–28.

[4] B.d. Silva, A. Natarajan, M. Motani, and K-C. Chua, "Design Considerations of Body Sensor Networks," in *Pro*c. of the Tenth IEEE International Conference on e-Health Networking, Applications & Services (IEEE Healthcom), 2008.

[5] A.B.Waluyo, I. Pek, and X. Chen, "SLIM: A Secured Lightweight Interactive Middleware for Wireless Body Area Network," in *Proc.* of 30th Annual International Conference of the IEEE EMBS, 2008, pp. 1821–1824.

[6] Texas Instruments [Online]. Datasheet Available at: http://www.ortodoxism.ro/datasheets2/9/0oe5ltue2s77tejsk7j5xs0j72fy.pdf. Last accessed: April, 2008.

[7] Texas Instruments Chipcon CC2420 [Online]. Datasheet Available at: http://focus.ti.com/docs/prod/folders/print/cc2420.html. Last accessed: April, 2008.

[8] UC Berkeley. TinyOS Community Forum [Online]. Available: http://www.tinyos.net. Last accessed: April, 2008.

[9] W.S. Yeoh, I. Pek, Y.H. Yong, X. Chen, and A.B. Waluyo, "Ambulatory Monitoring of Human Posture and Walking Speed Using Wearable Accelerometer Sensors," in *Proc.* of 30th Annual International Conference of the IEEE EMBS, 2008, pp. 5184–5187.

[10] Xue Yang and Nithin H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving Balance between Energy Saving and End-to-end Delay," in *Proc.* of 10th Annual International Conference of the IEEE RTAS, 2004, pp. 19-26.

[11] Shan Liang, Yunjian Tang and Qin Zhu, "Passive Wake-up Scheme for Wireless Sensor Networks," in *Proc.* of 2nd Annual International Conference of the IEEE ICICIC, 2007, pp. 507.