Communication Minimization for In-Network Processing in Body Sensor Networks: A Buffer Assignment Technique

Hassan Ghasemzadeh*, Nisha Jain*, Marco Sgroi† and Roozbeh Jafari*
*Embedded Systems and Signal Processing Lab, Department of Electrical Engineering
University of Texas at Dallas, Richardson, Texas 75080-3021
http://www.essp.utdallas.edu
† Wireless Sensor Networks Lab sponsored by Pirelli and Telcom Italia
1995 University Ave. Suite 225, Berkeley, CA 94704
marco.sgroi@wsnlabberkeley.com

***Abstract*—Body sensor networks are emerging as a promising platform for healthcare monitoring. These systems are composed of battery-operated embedded devices which process physiological data. The reduction in the power consumption is an important factor to increase the lifetime for such systems and to enhance their wearability through reducing the size of the battery. In this paper, we develop an energy-efficient communication scheme that uses buffers to reduce the number of transmissions among the sensor nodes constrained to limited hardware resources. A direct acyclic graph is used to model the information flow. We define a communication optimization problem and solve it using convex optimization techniques. We present results that support the efficiency of the proposed technique.**

I. Introduction

Technological advances in wireless communication, sensor design and embedded processors have led to the development of pervasive body sensor networks (BSNs) that enable wearable and mobile healthcare monitoring systems. Such platforms consist of a set of miniaturized sensor nodes which sense physiological and environmental data, initiate actions and trigger alarms during an emergency. By providing an efficient way of shifting healthcare from a traditional clinical setting to a home-bound, BSNs can reduce medical expenses and assist early detection of medical conditions.

Despite the variety of potential applications, wide deployment of BSNs is still limited due to computation, communication, battery lifetime and storage due to their small size, which is dictated by wearability factors. Previous studies of embedded sensor nodes have shown that data communication is expensive in terms of energy consumption, whereas data processing is relatively inexpensive [1].

Most BSN applications have certain signal processing requirements. The typical signal processing flow involves several processing blocks which may require collaboration among the sensor nodes. The signal processing can be computationally intensive and may require extensive inter-node communication introduced by dependencies among processing blocks across the network. Frequent inter-node communication leads to increased energy consumption in such applications. Therefore, transmissions in short bursts at a high bit rate can be desirable in order to minimize transmission energy per bit [2].

We focus on physical movement monitoring applications where daily activities of the patients are recognized through analysis of data from a BSN. A final report generated by the system is then sent to a physician at end of the day. The lack of immediate deadlines for processing data and communication in such applications makes them ideal for power optimization through burst transmissions. We propose a communication model for these systems with the objective of minimizing transmissions between sensor nodes. In this model, the inter-node dependency is captured by the dependency of the signal processing tasks across the nodes. We pose an optimization problem which assigns appropriate buffers to each processing block taking into consideration limited memory size on each sensor node. Further, we investigate and verify these communication models using an open source signal processing in node environment (SPINE) [3]. We also evaluate the performance gain achieved in terms of reduced number of packet transmissions.

II. Related Work

A number of projects have recently addressed energy-efficiency of BSNs. Authors in [4] introduce the notion of action coverage to reduce the number of active nodes in movement monitoring applications. A fast method of activity recognition using BSNs is proposed in [5] by developing decision tree models that involve only informative sensor nodes in movement identification. The dynamic power management scheme described in [6] is based on setting the components of electronic system into different states based on their performance, workload and power consumption level.

Burst architectures, where high data rate transmission in short periods of time is supported, are desirable for data transmission in BSNs [7] because of their potential for dramatic reduction in power usage. Burst transmission requires buffers to accumulate results between transmissions [8]. The concept has been utilized for communication optimization in several domains. In distributed embedded software [9],

978-3-9810801-5-5/DATE09 © 2009 EDAA

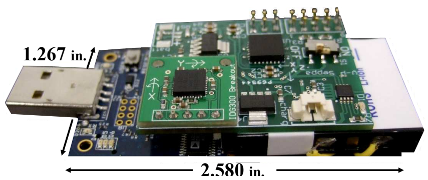

Fig. 1. **Sensor node with inertial sensor board**

buffering techniques are employed to achieve effective communication among asynchronous processes. In VLSI domain, buffer insertion has been in use for power optimization, by reducing switched capacitance, and for synchronizing hardware modules [10].

Nevertheless, to the best of our knowledge, the issue of communication minimization by means of buffers in BSNs has not been investigated previously. We make the use of an efficient buffer assignment technique in distributed lightweight embedded platforms with limited storage. Our technique minimizes the number of wireless transmissions in the network to improve system lifetime.

III. Preliminaries

Body sensor networks are mainly composed of a number of on-body physiological sensors integrated with processing and communication modules. Our pilot study of physical movement monitoring finds applications in fall detection, rehabilitation, balance evaluation and gait analysis [11].

A. System Architecture

Our system consists of several sensor nodes, each equipped with a processing module and a custom-designed sensor board powered by a Li-Ion battery as shown in Fig. 1. The processing module is a mote with embedded radio for communication. Each sensor board has a tri-axial accelerometer and a bi-axial gyroscope.

The system aims at detecting the physical movements of the subject wearing the system. A number of sensor nodes are placed on different joints of human body. Inertial data obtained by each sensor node is subject to physical action recognition and further processing. The local information is derived on each node and transmitted to a base-station in a wireless manner for further analysis. The base-station can be another mote or a PDA which determines the final decision by integrating information from all the nodes.

B. Signal Processing

A typical system of physical activity recognition intends to detect a set of movements of interest. Pattern classification techniques are mostly employed to distinguish each action from the rest. Our signal processing involves several processing tasks as shown in Fig. 2 and described in the following.

Preprocessing: The data is collected from each sensor node and is passed through a moving average filter to reduce high frequency noise.

Segmentation: The signal is divided into segments that represent complete actions.

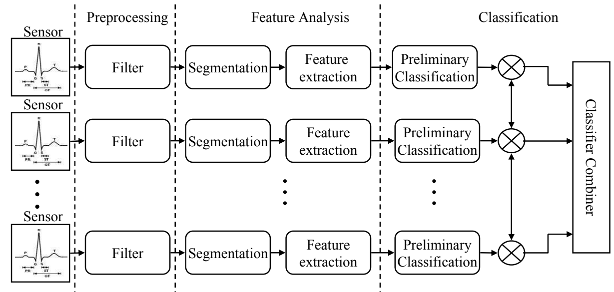

Fig. 2. **Typical signal processing for action recognition**

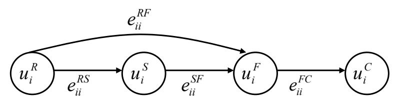

Fig. 3. **Intra-node dependency subgraph**

Feature Extraction: Temporal dimensionality of each segment is reduced by reducing the input data to a set of representative feature.

Classification: All sensor nodes make decisions on the movements performed by the subject. A local decision is made by each node and a final decision based on the all local decisions is made by a central node.

C. Dependency Graph

The signal processing model described earlier is usually implemented in a distributed manner over a BSN. We represent information flow across the network through our dependency graph model. In a system of *n* sensor nodes, dependency graph is composed of *n* subgraphs connected through inter-node links. Each subgraph represents static information dependencies within a node as shown in Fig. 3 where u_i^R, u_i^S, u_i^F and u_i^C denote sensor reading and preprocessing, segmentation, feature extraction and classification blocks respectively.

Definition 1: Given a set of n sensor nodes $s_1, \ldots, s_n$, *intra-node dependency subgraph* for node s_i is defined by $G_i = (V_i, E_i)$ where V_i is the set of four vertices and E_i is the set of four edges. Each vertex, denoted by u_i^{μ}, corresponds to a processing unit, and each edge is denoted by $e_{ii}^{\mu\eta}$ $(\mu, \eta \in \{R, S, F, C\})$ representing intra-node dependencies.

Inter-node links which represent dependencies across sensor nodes are used to connect subgraphs and form a dependency graph. An inter-node link is determined according to dependencies induced by application.

Definition 2: Given a set of *n* sensor nodes $s_1, \ldots, s_n$ and inter-node dependencies, *dependency graph* $G = (V, E)$ is formed by connecting n intra-node dependency subgraphs $G_1, \ldots, G_n$ through dependency links E_b defined by application criteria. The set of edges E is given by

$$E = E_w \cup E_b \tag{1}$$

where the set of intra-node edges E_w is given by

$$E_w = \bigcup_{i=1}^{n} E_i \qquad i = 1, ..., n \tag{2}$$

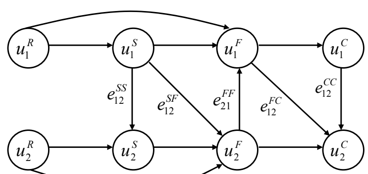

Fig. 4. **Sample dependency graph with five inter-node links**

and the set of vertices V is defined by

$$V = \bigcup_{i=1}^{n} V_i \qquad i = 1, ..., n \tag{3}$$

where each V_i is the set of vertices within subgraph G_i.

As an abstract example, Fig. 4 shows a dependency graph for a network of two sensor nodes where processing units are connected through five inter-node dependency links.

In our system of activity monitoring, practical inter-node links $e_{ij}^{\mu\eta}$ connecting processing block μ at node s_i to processing block η at s_j are presented as follows. A link e_{ij}^{SS} shows that the segmentation block at sensor s_j requires segmentation results from sensor s_i. A lack of prominent patterns prevents s_j from properly determining start and end of actions. When a subject is walking, a node placed on the leg observes high quality pattern, but a head node observes noisy signals and irrelevant patterns. A link e_{ij}^{SF} is required when node s_i becomes a master node for segmentation and provides node s_j with the annotation points (fragment of sensor data stream signifying a certain activity of predefined class type) for feature extraction. When s_j requires information on features calculated by s_i to extract features, a link e_{ij}^{FF} is added. If classification at node s_j is contingent on certain features from s_i, a link e_{ij}^{FC} is required. A segmentation unit may need information on current movement detected by another node. In this case, a link e_{ij}^{CS} is added to the dependency graph. A link e_{ij}^{CF} is required if the feature extraction depends on classification results provided by other nodes. When sensor node s_j is considered as classifier combiner, links of the form e_{ij}^{CC} are considered allowing other nodes to transmit local classification to the central node.

IV. Problem Definition

Data transmissions is one of the major sources of power consumption in BSNs. Transitional movements are mostly non-periodic and discontinuous in time. Therefore, inter-node transmissions are required according to the availability of data due to new human actions. The main drawbacks with such a model include the overhead introduced by the establishment of new communication channels and low bandwidth utilization. To compensate these issues, our model aims to minimize the number of burst transmissions by inserting appropriate buffers between dependent processing units.

The idea behind using buffers is to transmit the maximum amount of data in short time intervals. The large number of data blocks produced by each processing unit is stored locally and is transmitted using available bandwidth. This would conform to real situations of healthcare systems where physicians are interested in receiving reports on daily activities rather than immediate reports. By assuming no immediate deadlines in the system, we maintain individual buffers on each link and transmit the data blocks separately for each inter-node link.

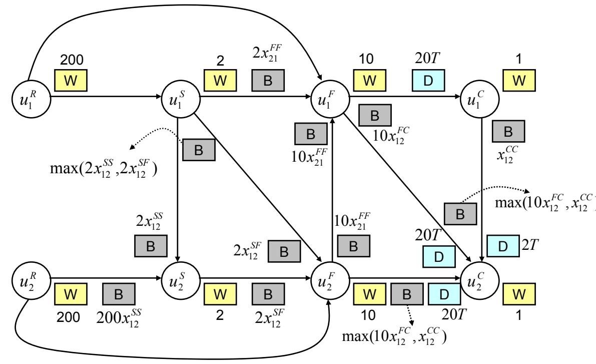

Fig. 5. **Example of buffer assignment method**

Problem 3: Given dependency graph G, each inter-node link $e_{ij}^{\mu\eta}$ is associated with a number $x_{ij}^{\mu\eta}$ denoting the number of actions for which data blocks produced by the source unit u_i^{μ} are buffered prior to every transmission. The objective is to find values $x_{ij}^{\mu\eta}$ that minimize the number of transmissions subject to memory constraints on nodes.

A. Buffer Assignment

Through an illustrative example (Fig. 5), we show our buffer allocation technique in the rest of this section.

The amount of data that can be stored within buffers is constrained by the available memory of the nodes. Different processing units produce data blocks of different sizes. In the example shown in Fig. 5, reading (raw sampled sensor data), segmentation, feature extraction and classification generate data blocks of size 200, 2, 10 and 1 byte respectively.

Definition 4: The amount of data produced by unit u_i^{μ} per action is called *data unit* and denoted by b_i^{μ}.

Our communication model maintains three types of buffers for each sensor node as defined below:

Definition 5: A buffer of size b_i^{μ} associated with processing unit u_i^{μ} is called *intra-node buffer* in order to enable in-node processing. The data stored in this buffer will be consumed by the next processing unit within the same node.

In the above example, the total amount of intra-bode buffers, labeled by W, maintained for nodes s_1 and s_2 is $W_1 = W_2 = 200 + 10 + 2 + 1 = 213$.

Definition 6: A buffer allocated to a processing unit u_i^{μ} due to inter-node dependencies is called *inter-node buffer*. Such a buffer provides data for consequent dependent units.

Inter-node buffers are sized according to the number of actions each link would store. An inter-node link with $x_{ij}^{\mu\eta}$ number of actions allocates a buffer to the source unit and another buffer to the destination unit. When a source unit has more than one outgoing edge, only one buffer is enough to store data for both links. The size of such buffer is determined based on maximum amount of data required to be transmitted among the links. In Fig. 5, the processing unit u_1^S has two outgoing edges. The inter-node buffer (labeled by *B*) is then sized according to the values of x_{12}^{SS} and x_{12}^{SF}. However, only a single buffer with the maximum required size would be sufficient to accommodate the data generated by this unit. That is, a buffer of size $\max(2x_{12}^{SS}, 2x_{12}^{SF})$ is allocated. Destination units, however, require $2x_{12}^{SS}$ and $2x_{12}^{SF}$. Similarly, a destination unit with more that one incoming edge requires a buffer sized according to maximum amount of data enforced by the incoming links. In Fig. 5, such a buffer (with size $\max(10x_{12}^{FC}, x_{12}^{CC})$) is allocated to the incoming links at u_2^C. Moreover, an inter-node dependency prevents the destination unit from performing any processing until it receives data from the source. That is, the data produced by the predecessor of the destination unit ($b_j^{\eta-1}$) must be buffered as well. In Fig. 5, link e_{12}^{SS} enforces a buffer of size $200x_{12}^{SS}$ on the incoming link e_{11}^{RS}.

Computational time varies from one processing unit to another. This implies that consecutive processing blocks may operate at different speeds. When a processing unit is slower than its predecessor, extra buffers are inserted to compensate the timing delay.

Definition 7: *Production rate*, r_i^μ, is defined as the number of actions a unit u_i^μ can process in a given time unit.

Definition 8: A buffer enforced due to data processing time delay between two consecutive units is called *delay buffer*.

In the above example, we assume that the reading, segmentation and feature extraction all have the same production rate $r_i^\mu = 10$. The classifier, however, operates slower with $r_i^C = 8$. This requires delay buffers (labeled by *D*) of size *20T*, to prevent data overflow for extended time period, where *T* refers to the time until memory overflows. The classifier on the second node has two incoming edges which must hold transmitted data for future processing by the classifier.

We make here certain assumptions about the frequency of occurrence of the actions. We assume that the number of movements occurring in a given time period follows a Poisson distribution [12]. The probability of observing x number of actions in a given time interval is represented by

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{4}$$

where λ is the expected number of movements occurred in the given interval. With a confidence level of $1-\alpha$, the Poisson model provides an upper bound k on the number of actions occurred during the interval:

$$p(x \leq k) = \sum_{x=0}^{k} p(x) = 1 - \alpha \tag{5}$$

Definition 9: The upper bound on the number of actions occurred during a given time interval is known as *action rate*, *k*, which is obtained according to Poisson distribution model.

B. Problem Formulation

Given the dependency graph G as described earlier, let $a_{ij}^{\mu\eta}$ be a binary that represents existence of inter-node links and is given by

$$a_{ij}^{\mu\eta} = \begin{cases} 1, & \text{if } s_i \text{ and } s_j \text{ are dependent through } u_i^\mu \text{ and } u_j^\eta \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and let d_i^μ be another binary denoting whether or not a processing unit is slower than its predecessor.

$$d_i^\mu = \begin{cases} 1, & \text{if } r_i^\mu < r_i^{\mu-1} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The number of actions A occurred at action rate k during time period T is given by

$$A = k \times T \tag{8}$$

and the number of transmissions can be calculated by

$$Z = \sum_{i,j} A \frac{a_{ij}^{\mu\eta}}{x_{ij}^{\mu\eta}} \tag{9}$$

The total size of the intra-node buffers W_i for node s_i is given by

$$W_i = \sum_{\mu} b_i^\mu \tag{10}$$

and the total size of the inter-node buffers for node s_i is determined by

$$B_i = \sum_{\mu} (\max_{j,\eta}(a_{ij}^{\mu\eta} b_i^\mu x_{ij}^{\mu\eta}) + \max_{j,\eta}(a_{ji}^{\eta\mu} b_j^\eta x_{ji}^{\eta\mu}) + \max_{j,\eta}(a_{ji}^{\eta\mu} b_j^{\mu-1} x_{ji}^{\eta\mu})) \tag{11}$$

and the total size of delay buffers for node s_i is given by

$$D_i = T \sum_{\mu} d_i^\mu ((r_i^{\mu-1} - r_i^\mu) b_i^{\mu-1} + \sum_{j,\eta} a_{ji}^{\eta\mu} b_j^\eta) \tag{12}$$

Let M_i be the size of the memory on node s_i. The problem of minimizing the number of transmissions can be formulated as a convex optimization problem as follows:

$$Min \quad Z \tag{13}$$

subject to:

$$W_i + B_i + D_i \leq M_i \qquad \forall i \in \{1, ..., n\} \tag{14}$$

$$x_{ij}^{\mu\eta} \in \; Z^+ \quad \forall i, j, \mu, \eta \tag{15}$$

The non-linear constraints due to the *max* functions in (11) can be transformed into linear equations by expanding every function over all values taken by the function. The integrality condition (15) can be relaxed to solve the problem using common convex programming tools.

$$x_{ij}^{\mu\eta} > 0 \quad \forall i, j, \mu, \eta \tag{16}$$

The solution obtained due to integrality relaxation will not carry the optimality condition, but we find a lower bound on the size of memory for which the result is optimal.

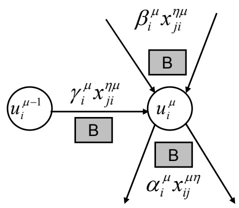

Fig. 6. **Three components of inter-node buffers**

Theorem 10: For each sensor node s_i, the integer relaxed convex optimization finds optimal solutions for memories of size

$$M_i - \sum_{\mu}(1-\varepsilon)\alpha_i^\mu \beta_i^\mu \gamma_i^\mu \tag{17}$$

where

$$\begin{aligned}
\alpha_i^\mu &= a_{ij}^{\mu\eta} b_i^\mu \quad \text{s.t.} \quad j,\eta = \arg\max_{j,\eta}(a_{ij}^{\mu\eta} b_i^\mu x_{ij}^{\mu\eta}) \\
\beta_i^\mu &= a_{ji}^{\eta\mu} b_j^\eta \quad \text{s.t.} \quad j,\eta = \arg\max_{j,\eta}(a_{ji}^{\eta\mu} b_j^\eta x_{ji}^{\eta\mu}) \\
\gamma_i^\mu &= a_{ji}^{\eta\mu} b_j^{\mu-1} \quad \text{s.t.} \quad j,\eta = \arg\max_{j,\eta}(a_{ji}^{\eta\mu} b_j^{\mu-1} x_{ji}^{\eta\mu})
\end{aligned} \tag{18}$$

Proof: Inter-node buffers assigned to each processing unit can have at most three components as given in equation (11). Integer relaxation would increase size of each optimal buffer associated with $x_{ij}^{\mu\eta}$ by factor $1-\varepsilon$ of the unit data (α_i^μ,β_i^μ,γ_i^μ) as shown in Fig. 6. Hence, the memory usage on each node is optimized as illustrated in (17).

V. Experimental Setup

In our experiments, we use TelosB motes which are commercially available form XBow®. Information regarding the hardware specification, power consumption and memory can be found in [13]. These motes have memories of size $M_i = 10KB$. We take a network of three sensor nodes with different configurations to show the effectiveness of our technique. We estimate timing delays based on our preliminary results obtained from SPINE [3], an open source framework for BSNs. According to this investigation, the reading unit operates at $2.1Hz$ for input data of 200 samples. We further assume that segmentation is performed based on standard deviation taken over a moving window [14] which produces annotations at $2.9Hz$. Feature extraction operates at $2.2Hz$ for obtaining 13 statistical features enforcing delay buffers between segmentation and feature extraction. We assume a *k*-NN algorithm for classification that operates at $1.9Hz$ enforcing extra delay buffers between feature extraction and classification.

We solve our optimization problem using CVX [15], a package for specifying and solving convex programs, in two scenarios with respect to the action rate. In the worst case, movements occur continuously with no stall in between. In this case, we assume the rate of $1.8Hz$ (110 actions/min.) for walking actions as measured in [16]. We also solve the problem for the case where the action rate is $0.0167Hz$ (1 action/min.). In the former scenario, the Poisson model is used to achieve a confidence level of 95%. In the later case, however, processing blocks operate fast enough compared with the incoming actions. Therefore, no delay buffers are needed.

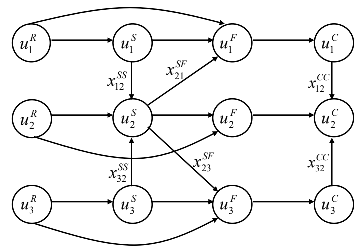

Fig. 7. **Communication network for experimental setup (Conf. #1)**

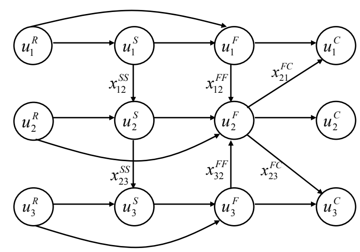

Fig. 8. **Communication network for experimental setup (Conf. #2)**

VI. Experimental Results

We solve the convex optimization problem for several configurations, corresponding to a particular BSN application, of a network composed of three sensor nodes. The first configuration is shown in Fig. 7 where node 2 is the master node for segmentation and classification. This node receives local annotations from the two other nodes and makes a decision on final annotations. The annotation points are then transmitted to the other nodes for feature extraction. This node is also responsible for combining local classifications. Inter-node links are labeled as $x_{ij}^{\mu\eta}$ where their weights denote the number of actions stored in the buffers of each adjacent sensor node. In Fig. 8, another configuration is depicted where sequential dependencies for segmentation are needed. This model also takes into consideration dependency of features of node 2 on features obtained by other two nodes. Finally, nodes 1 and 3 perform a partial data fusion at the feature level for their classification. In the last configuration, shown in Fig. 9, we consider dependency of a feature extraction, node 2, on segmentation and classification of other nodes. A complete data fusion at the feature level is performed by the classifier on node 2. In Table I, the results of our optimization are shown for the three configurations. The last column portrays packet transmission reduction. All the results are calculated for the period of twenty four hours of continuous physical activity monitoring.

VII. Discussion and Future Work

We outline several important issues along with our future plans in the following.

- Our proposed technique has several advantages in addition to the main objective of transmission reduction.

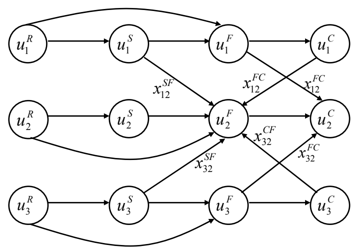

Fig. 9. **Communication network for experimental setup (Conf. #3)**

TABLE I
NUMBER OF TRANSMISSIONS FOR DIFFERENT COMMUNICATION NETWORKS

Configuration	$\lambda(Hz)$	# Packets		Imp
		W/o Buffer	W/i Buffer	
1	1.8	185760	74485	59%
	0.0167	1440	80	94%
2	1.8	46440	31226	32%
	0.0167	1440	153	89%
3	1.8	69660	37297	46%
	0.0167	1440	17	98%

It can simplify the communication, reduce the overhead of communication coordination, and enhance the system lifetime.

- The choice of the number of packets as a performance metric makes our evaluation independent of the communication protocol. Furthermore, larger packet sizes presented by our model can potentially enhance communication by reducing energy per bit for transmission [2].
- The existence of deadlines is not investigated in this study. Nevertheless, immediate reactions can be initiated when certain actions (e.g. falling) are observed. For each such action, one master node can detect the movement locally and initiate the communication.
- Our current proposed solution is based on pre-allocation of buffers. In future, we would like to introduce dynamic allocation of buffers and intelligent routing protocols to determine the event for burst transmission. We also plan on further refinement of our model for actual implementation on the nodes.

VIII. CONCLUSION

In this paper, we presented a communication model to reduce the number of transmissions by introducing buffers. We formulated the problem using integer convex programming and solved using CVX tool. In our work, we took movement monitoring as a pilot application and evaluated the results using our model. Our system model is generic in nature and it can be suitably adapted for various healthcare applications based on BSNs.

ACKNOWLEDGEMENT

The authors would like to acknowledge funding support from Telecom Italia, Texas Instrument and Project Emmitt. They would also thank Ville-Pekka Seppa for his efforts in designing motion sensor board.

REFERENCES

[1] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 40–50, Mar 2002.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "smart dust"," in *MobiCom: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, New York, NY, USA, 1999, pp. 271–278.

[3] R. Gravina, A. Guerrieri, S. Iyengar, F. Tempia Bonda, R. Giannantonio, F. Bellifemine, T. Pering, M. Sgroi, F. G., and S.-V. A., "Demo abstract: Spine (signal processing in node environment) framework for healthcare monitoring applications in body sensor networks," *Proc. of the 5th European conference on Wireless Sensor Networks (EWSN)*, 2008.

[4] H. Ghasemzadeh, E. Guenterberg, K. Gilani, and R. Jafari, "Action coverage formulation for power optimization in body sensor networks," in *ASP-DAC '08: Proceedings of the 2008 conference on Asia and South Pacific design automation*, 2008, pp. 446–451.

[5] H. Ghasemzadeh, J. Barnes, E. Guenterberg, and R. Jafari, "A phonological expression for physical movement monitoring in body sensor networks," *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 58–68, 2008.

[6] L. Benini, A. Bogliolo, G. Paleologo, and G. De Micheli, "Policy optimization for dynamic power management," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 18, no. 6, pp. 813–833, Jun 1999.

[7] A. Dolgov and R. Zane, "Low-power wireless medical sensor platform," *Engineering in Medicine and Biology Society, EMBS: 28th Annual International Conference of the IEEE*, pp. 2067–2070, 2006.

[8] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design considerations for distributed microsensor systems," *Custom Integrated Circuits, 1999. Proceedings of the IEEE 1999*, pp. 279–286, 1999.

[9] A. Benveniste, P. Caspi, M. di Natale, C. Pinello, A. Sangiovanni-Vincentelli, and S. Tripakis, "Loosely time-triggered architectures based on communication-by-sampling," in *EMSOFT: Proceedings of the 7th ACM/IEEE international conference on Embedded software*, New York, NY, USA, 2007, pp. 231–239.

[10] A. Murugavel and N. Ranganathan, "Gate sizing and buffer insertion using economic models for power optimization," *VLSI Design, 2004. Proceedings. 17th International Conference on*, pp. 195–200, 2004.

[11] 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 *The 2nd International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet)*. Breckenridge, CO: ACM, June 2008.

[12] A. Panangadan, M. Mataric, and G. Sukhatme, "Detecting anomalous human interactions using laser range-finders," *Intelligent Robots and Systems, 2004. (IROS). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2136–2141 vol.3, 2004.

[13] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pp. 364–369, April 2005.

[14] E. Guenterberg, H. Ghasemzadeh, R. Jafari, and R. Bajcsy, "A segmentation technique based on standard deviation in body sensor networks," in *Dallas-EMBS: Proceedings of IEEE Dallas Engineering in Medicine and Biology Workshop*, 2007.

[15] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," Recent Advances in Learning and Control, (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, pages 95-110, Lecture Notes in Control and Information Sciences, Springer, 2008.

[16] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive kalman filtering," in *Proceedings of the 13th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS*, 2000.