DynAHeal: Dynamic Energy Efficient Task Assignment for Wireless Healthcare Systems

Priti Aghera[1], Dilip Krishnaswamy[12], Diana Fang[1], Ayse Coskun[3], Tajana Rosing[1]

[1] Department of Computer Science and Engineering, Univ of California San Diego, CA 92093, USA
[2]Qualcomm Research Center, San Diego, CA 92121, USA [3]ECE Dept, Boston University, MA 02215, USA

***Abstract*—Energy consumption is a critical parameter in wireless healthcare systems which consist of battery operated devices such as sensors and local aggregators. The system battery lifetime depends on the allocation of processing, sensing, and communication tasks to devices of the system. In this paper, we optimize the battery life of a wireless healthcare system by efficiently assigning tasks to the available resources. There are several dynamically changing characteristics in the system, such as task parameters (processing complexity, arrival rate, and output data), each device's available battery capacity, varying wireless channel conditions, and network load. Our dynamic task assignment algorithm, "*DynAHeal*" adapts to such changing conditions, and improves the battery life. Our experiments show that the task assignment given by *DynAHeal* improves the overall system lifetime under varying dynamic conditions on an average 60% relative to sending all the data for processing to the base station, and 35% with respect to an optimal static assignment.**

I. INTRODUCTION

The new generation of wireless mobile systems with seamless integration of 2.5G and 3G cellular systems, wireless LAN, Bluetooth, and Zigbee provides wide coverage and an improved capacity to run various types of wireless applications including healthcare [1]. In addition, advances in integrated circuit design and bioengineering have led to the design of low-cost, miniature, and lightweight physiological sensors that can be seamlessly integrated into a body area network for human health monitoring. The MobiHealth project [6] provides an example of a healthcare system, where the main goal is to continuously sense and send the data to a backend server for analysis by healthcare professionals. CardioNet [2] provides 24x7 cardiac monitoring service with beat-to-beat, real time analysis, automatic arrhythmia detection, and wireless ECG transmission. There are many systems like Alarm-Net [3], I-Living [4], and PAMM [5] to aid the elderly in their daily life. Most of the work on wireless healthcare systems today assumes sense-&-forward mode where data is gathered and sent to the backend server for processing. Sending all data to the backend leads to significantly lower system battery lifetimes and does not leverage the processing capabilities available in sensors and mobile phones today. Task assignment and scheduling onto multiple resources are well-studied problems in traditional computation and VLSI computer-aided design [7]. Previous work has shown (e.g. [8], [9]) that the cost of transmitting or receiving a bit wirelessly can be significantly more expensive than processing the bit locally on the CPU. In [10] and [11], static algorithms are introduced for energy-efficient task assignment and scheduling for a single-hop network with homogeneous resources. However, these algorithms are not applicable to systems with dynamically changing characteristics and heterogeneous resources.

In this work, we propose an adaptive runtime task assignment technique for wireless healthcare systems with the primary goal of improving the battery lifetime of the overall system while meeting task dependency and deadline constraints. We first optimize task assignment using integer linear programming (ILP), which serves as a baseline of comparison for our novel dynamic algorithm. The ILP solution is optimal for an *a priori* known static set of static tasks. However, the computationally expensive nature of the ILP makes it inefficient as workload and system characteristics change at runtime. Our novel algorithm, called *DynAHeal (Dynamic Task Assignment for Wireless Healthcare Systems)* adapts quickly to dynamic changes in workload and system characteristics. *DynAHeal* estimates how long the system would last for each task assignment option, and selects the assignment that provides the highest system lifetime. To evaluate our algorithm for a variety of task sets with different set of sensors and under various wireless channel conditions, we implement it using Qualnet [13] discrete event wireless simulator. We show that our dynamic scheduling technique is able to improve the battery life on an average 60% in comparison to sending all the data for processing to the base station, and by up to 35% in comparison to the design time task assignment given by the ILP.

II. SYSTEM MODEL AND TASK ASSIGNMENT ALGORITHMS

Figure 1: Wireless healthcare system architecture

Figure 1 shows the architecture of a typical wireless healthcare system. The system consists of three main components: Sensors, Local Aggregators (LAs), and a Backend Server (BE). Sensors form the Body Area Network (BAN). Their function is to sense, detect specific events (e.g., low blood sugar). Sensors send processed and/or raw sample data to their respective LA wirelessly. There is one LA per BAN. The LA aggregates data collected by sensors and may process the data before sending it to a centralized BE for analysis by a health professional. Alternatively, the LA might send all the data to BE for analysis. The LA can typically contain multiple wireless technologies such as Bluetooth, Zigbee, WLAN, and WWAN. The BE receives information sent by the LA and stores the information in a database which can be accessed for further analysis. In addition, the BE may also process the information that it receives in real-time.

Tasks of a wireless healthcare application can be modeled as a Directed Acyclic Graph (DAG) $G = (T, E)$ in which each vertex represents a periodic task $T_i \in T$. Each edge in the graph $E_{ij} \in E$ represents a precedence relation between tasks $T_i, T_j \in T$ and an inherent communication task. Tasks that do not have any predecessors are called source tasks, and tasks that do not have successors are called sink tasks. The weights W_{ij} on edge E_{ij} represent the amount of data that needs to be communicated from task T_i to T_j. We define system life as the minimum of battery life of all the mobile components per patient. ***Given a DAG and a set of heterogeneous resources R, our goal is to map the given DAG onto the set of resources R such that the system battery life is maximized.***

A. *Static Task Assignment Algorithm - ILP:*

The ILP-based static task assignment maximizes the battery life for a known set of tasks and resources. We formally define the assumptions and the objective function as the following:

Given: A DAG contains a set T of n tasks, and a set R of m (heterogeneous) resources. For each resource, the fraction of energy consumed out of the total battery lifetime available Bat_r by a specific task assignment (*Assign:* $T \rightarrow R$) running on resource r is given by E_r / Bat_r.

Goal: Ensuring that the rate of depletion of batteries throughout the system is balanced by minimizing the maximum fraction of energy consumed by a specific task assignment for each resource subject to deadline, task precedence and allocation (i.e., a task having to run on a particular resource) constraints. The ILP objective function for m sources is:

$$\min_i \left\{ \max_r \frac{E_{r,i}}{Bat_r} \mid r = 1,2,\ldots m-1 \right\}$$

By minimizing the maximum energy depletion rate, the ILP balances the rate of battery energy consumption among the resources. This ensures that a resource is not over-used and that its battery does not get depleted significantly earlier than others. While maximizing system lifetime, the precedence and deadline constraints of the tasks are preserved.

B. *Dynamic Task Assignment - DynAHeal*

Due to the various sources of runtime variations, such as changes in wireless channel conditions or task execution times, performing static task assignment is not an efficient solution for real-life systems. Since solving the ILP for every change in the conditions is computationally very costly (i.e., taking minutes to hours depending on the task graph), we instead design a fast and energy-efficient task assignment algorithm called *DynAHeal* that runs on the LA. Figure 2 outlines our algorithm. Starting with the source task, for each task we select an assignment that meets the deadlines and maximizes the system battery life in comparison to all other task assignments possible at a given stage. We perform this operation iteratively until all tasks are assigned. After the initial assignment we iterate over the assignments and reexamine each task's assignment to see if an alternative assignment improves system life. We end the algorithm when further iteration does not improve system battery life. We use following equations to compute estimated battery life, T_{bat}, of a resource.

$$T_{bat} = \frac{E_{bat}}{P_{Tot}} \quad (1)$$

$$P_{tot} = P_{CPU} + P_{radio} \quad (2)$$

$$P_{CPU} = P_{CPUa} \times CPU_{util} + P_{CPUi} \times (1 - CPU_{util}) \quad (3)$$

$$P_{radio} = P_{tx} \times Tx_{active} + P_{rx} \times Rx_{active} + P_{idle} \times R_{idle} \quad (4)$$

Estimated battery life T_{bat} of a resource is defined as the remaining battery energy E_{bat} divided by the total power P_{Tot} consumed by the resource as shown in (1). P_{tot} is the sum of computation power P_{CPU} consumed by a CPU and communication power P_{radio} consumed by a radio on a platform. Equation (3) defines P_{CPU}, where P_{CPUa} and P_{CPUi} represent power consumed by a CPU in active and idle states respectively and CPU_{util} represents CPU Utilization. Equation (4) defines P_{radio}, where P_{tx}, P_{rx} and P_{idle} are the powers consumed by radio in transmit, receive and idle state respectively and Tx_{active}, Rx_{active} and R_{idle} are percentage of time that the radio is in transmit, receive and idle state respectively. All these parameters are affected by task assignments on a resource and radio link condition. We use the following task set definitions in describing the DynAHeal algorithm.

- T : set of n tasks in the system
- P $\subseteq$ T: set of pre-allocated tasks.
- A $\subseteq$ T : set of already assigned tasks
- R $\subseteq$ T: set of unassigned tasks
- E $\subseteq$ R: set of eligible tasks. All remaining tasks whose predecessors have been assigned.

Step 1: Assign the tasks of set P to specified nodes of the system. P is used to put user-defined constraints on task assignment and typically contains sensing tasks bound to specific sensors and data logging tasks bound to the BE.

Step 2: Update the sets A, E and R as per their definition.

Step 3: Compute *Battery Life Estimation* table as shown in Table I for all tasks in E. We estimate a sensor's battery life and the LA's battery life for each possible assignment of tasks (on sensor, LA or BE) using (1) to (4). We then compute the *system battery life* as minimum of *sensor battery life* and *LA battery life* for each possible assignment. The maximum of these three system life values is the maximum system battery life that we can achieve for the given task. We store the node ID of the best task assignment as *best assignment node ID*. We also compute *deadline slack* for each possible task assignment. If a particular task assignment does not meet the deadline, then this value will be negative, and we do not consider such an assignment possibility.

Step 4: Sort the *Battery Life Estimation* table based on maximum *system battery life* and select the task with the longest *system battery life* for assignment. This results in the maximum system life achievable at this stage. After this assignment, we go to step 2 and the process is repeated until all the tasks are assigned.

Step 5: Iterate over each task assignment to find a better solution by re-examining assignment of an individual task starting from the source.

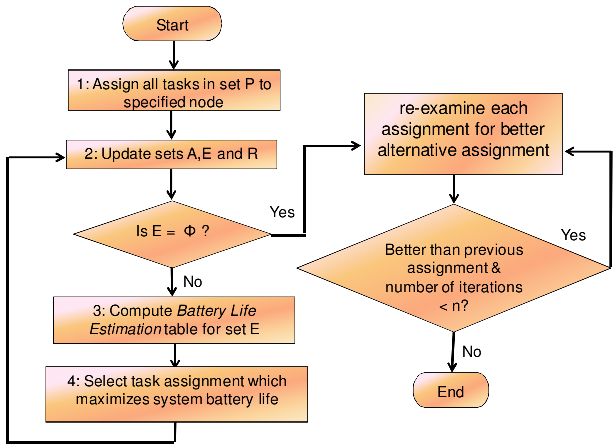

Figure 2: DynAHeal Algorithm

We bound the number of iterations to n to have an upper bound on execution time. In practice we haven't observed more then 2-3 repetitions of step 5. The complexity of the entire *DynAHeal* algorithm is $\mathbf{O(n^3 * Log(n))}$, leading to much lower run-time overhead compared to the ILP (which is NP-hard).

III. EXPERIMENTAL RESULTS

In this section we evaluate our dynamic task assignment technique in terms of its effect on the overall system battery life, and its dynamic adaptability.

The four task graphs we used in our experiments are described in Table I.

Table I: EXPERIMENTAL WORKLOAD

Task graph		# of sensors	# of tasks	Application
A	HR + Arrhythmia + Activity	2	8	ECG sensor detects heart rate per minute and arrhythmia while the accelerometer keeps track of activity.
B	Activity Detect	3	8	Detects a person's activity using three accelerometers.
C	GPS + HR + Activity	4	17	Correlates heart rate and activity of a person based on GPS data as context.
D	All-Vital	5	23	Logs all vital signs such as heart rate, blood pressure, and activity in addition to location.

We use Qualnet [13], a state of the art discrete event wireless network simulator, to simulate wireless healthcare systems. Sensor nodes are modeled as MicaZ nodes with a Zigbee radio. They use small coin cell batteries, with 560 mAh capacity and have CPU speed of 8MHz. The Local Aggregator is modeled as a UMTS-UE (User Equipment i.e. Handset) with an additional Zigbee radio interface, a 900mAh battery and CPU speed of 400MHz. The handset is connected to the Backend Server via the UMTS network and with sensors via Zigbee radio. Tasks assigned to the resources send data to next resource over UDP if any of its successor tasks is not assigned on the same resource. We implement logic for periodic task execution, data transmissions, radio link parameter measurement and reporting, task assignment control messaging and execution of the *DynAHeal* algorithm at the application layer. We formulate an ILP for each task graph and use an open source ILP solver called *lp_solve* [12] to get the optimal task assignments for each of the formulations.

A. *Static vs dynamic algorithms*

In Table II, we show percentage improvement in the system life (*SysLifeImprov*) achieved based on the task assignments by the ILP and the *DynAHeal* algorithms with respect to a simplistic assignment in which all data is sent to the back end server for processing, *All-On-BE*. System parameters such as execution time and arrival rate of the tasks are constant, and LA and sensors are stationary to show that *DynAHeal* improvement in the system battery lifetime are nearly identical to the optimal ones obtained by the ILP (both get an average 60% improvement). However, the computational time of ILP is in the order of seconds compared to *DynAHeal*'s execution time which is in milliseconds.

Table II: AllOnBE vs ILP VS DYNAHEAL COMPARISIONS

TG	SysLifeImprov (ILP)	SysLifeImprov (*DynAHeal*)	Exec Time (ILP) sec	Exec Time (*DynAHeal*) sec
A	5%	5%	0.38	0.002
B	20%	20%	8.45	0.003
C	82%	78%	194.80	0.005
D	135%	135%	286.53	0.018

B. *DynAHeal adaptability*

Dynamic load balancing is required in the system as the processing complexity of a task may change depending on the amount of processing required on the data, or due to increased load on a resource in the system such as the LA. If the sensed information increases due to a change in the patient's health condition or due to patient mobility, then the task complexity increases. Additionally, if a cell-phone is used as the LA, it may have additional processing to perform during the day for the user, in addition to supporting the health-care tasks. As a result of this varying load, a current task assignment may no longer be energy efficient.

In our experiments we increased the execution time of processing tasks by a factor of 5 after 12 hours and then adjusted it back to the original setting for the next 12 hours to simulate two levels of processing load. In Figure 3 we compare the system lifetime based on new task assignments obtained by *DynAHeal* algorithm with a task assignment obtained by the ILP that is based on initial task execution times. We see up to 35% increase in system life when we use the *DynAHeal* algorithm. Similar gains are observed when other task parameters such as task's output data and arrival rate are changed.

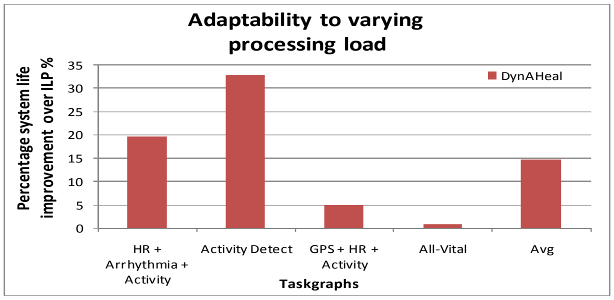

Figure 3: System life improvement achieved by *DynAHeal* over ILP when task execution times change at run-time

Wireless channel conditions change over a period of time due to factors such as the mobility of the person using the system, weather conditions, and extra traffic on the system. To simulate this change we assume that the user is at home for 14 hours of the day, 8 hours at work, and that commuting to and from work takes 1 hour in each direction. We also assume that link conditions are better at home compared to work. The LA detects these changes dynamically and updates the task assignments (as 2 iterations of *DynAHeal* takes close to8 msec for a typical task set on a 400MIPS processor). The cost of UMTS wireless transmissions could increase by a factor of 10 depending on user's distance from the base station. Figure 4 shows that the system life is increased by approximately 16% using the *DynAHeal* algorithm to handle changes in link conditions due to the mobility of the LA.

In another set of experiments we assume the user is in a densely populated area like a downtown location in a city, where there are multiple LAs/cellphones in the same coverage area, sharing the data channel for 3 hours per day. This simulates a change in the WWAN channel utilization due to the presence of multiple users. The results of this experiment are shown in Figure 4 , with DynAHeal execution based on multiple LAs in the system in Figure 4 sharing the same WWAN channel. Under varying channel loading we achieve an improvement of up to 10% and on an average 6% by dynamically changing the task assignment using *DynAHeal* in comparison to static allocation provided by the ILP.

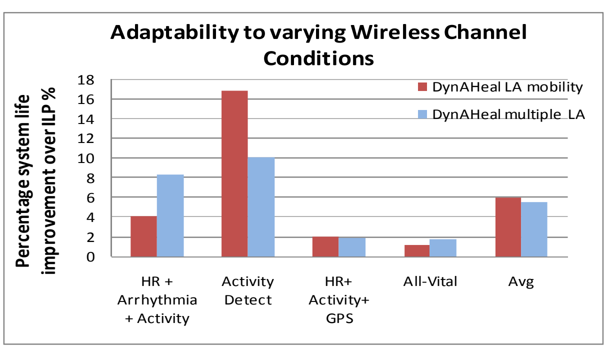

Figure 4**:** System life improvement achieved by *DynAHeal* over ILP under dynamic link condition change

IV. Conclusion

Task assignment in a healthcare system consisting of heterogeneous resources significantly impacts the battery life of the different components in the system, and the overall battery life of the entire system. To address this problem, a dynamic algorithm, *DynAHeal*, for energy-efficient task assignment was presented. An optimal task assignment for static conditions using an ILP was formulated and solved as a baseline for comparison. The *DynAHeal* algorithm was implemented in the Qualnet discrete event wireless simulation environment to evaluate various healthcare system scenarios. It was shown that the task assignment generated by *DynAHeal* performs very close to the optimal task assignment given by the ILP in static conditions. In dynamic situations, *DynAHeal* dynamically recomputes the task assignment and outperforms the static ILP-based solution by up to 35% under varying load and by up to 16% under varying wireless channel conditions such as user mobility and increased channel utilization.

V. Acknowledgments

This work has been funded by NSF project, "Citisense" grant CNS-0932403, Center for networked systems (http://cns.ucsd.edu) grant CNS08TR and Qualcomm.

References

[1] Kamugisha Kazaura, Liu Jun, and Mitsuji Matsumoto, "A Simulation Based Evaluation on the Performance Integrated 3G - Wireless LAN Network", TENCON 2004, IEEE Region 10

[2] CardioNet, http://www.cardionet.com/

[3] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L.,et al., "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring", University of Virginia

[4] Qixin Wang, Wook Shin, Xue Liu, Zheng Zeng, Cham Oh, Bedoor K. et al."I-Living: An Open System Architecture for Assisted Living",

[5] Hamid Aghajan, Juan Carlos Augusto, Chen Wu, Paul McCullagh, and Julie-Ann Walkden, "Distributed Vision-Based Accident Management for Assisted Living"

[6] A.V.Halteren, R. Bults, K. Wac, D. Konstantas, and I. Windya, "Mobile Patient Monitoring: The MobiHealth System" The Journal of Information Technology in Healthcare, vol. 2, 2004.

[7] Giovanni De Micheli, "Synthesis And Optimization Of Digital Circuits" McGraw-Hill, Inc.

[8] E. Jeannot, B. Knutsson, and M. Bjo¨rkman, "Adaptive Online Data Compression," Proc. IEEE Int'l Symp. High Performance Distributed Computing '02, July 2002.

[9] Rong Xu, Zhiyuan Li, Cheng Wang, Peifeng Ni, "Impact of Data Compression on Energy Consumption of Wireless-Networked Handheld Devices", ICDCS'03

[10] Yuan Tian Ekici, E. Ozguner, F, "Energy-constrained task mapping and scheduling in wireless sensor networks" in Mobile Adhoc and Sensor Systems Conference, 2005

[11] Y. Yu and V. K. Prasanna, "Energy-balanced task allocation for collaborative processing in wireless sensor networks," in Mobile Networks and Applications, Feb. 2005.

[12] Lp_solve - http://lpsolve.sourceforge.net/5.5/

[13] I. Scalable Network Technologies, "QualNet Simulator," Scalable Network Technologies, Inc., http://www.qualnet.com

[14] Alive Heart Sensor – http://www.alivetec.com