gTBS: A Green Task-Based Sensing for Energy Efficient Wireless Sensor Networks

Abdullah Alhalafi, Lokman Sboui, Rawan Naous, and Basem Shihada
King Abdullah University of Science and Technology (KAUST),
Computer, Electrical and Mathematical Science and Engineering Division (CEMSE),
Thuwal 23955-6900, Saudi Arabia.
{abdullah.halafi,lokman.sboui,rawan.naous,basem.shihada}@kaust.edu.sa

***Abstract*—Wireless sensor networks (WSN) are widely used to sense and measure physical conditions within a given region. However due to the limited lifetime of the sensors, many efforts are made to design energy efficient WSN. As a result, many techniques were presented in the literature such as power adoption, sleep and wake-up, and scheduling in order to enhance WSN lifetime. These techniques where presented separately and shown to achieve some gain in terms of energy efficiency. In this paper, we present an energy efficient cross layer design for WSN that we named "green Task-Based Sensing" (gTBS) scheme. The gTBS design is a task based sensing scheme that not only prevents wasting power in unnecessary signaling, but also utilizes several techniques for achieving reliable and energy efficient wireless sensor network (WSN). The proposed gTBS combines the power adaptation with a sleep and wake-up technique that allows inactive nodes to sleep. Also, it adopts a gradient-oriented unicast approach to overcome the synchronization problem, minimize network traffic hurdles, and significantly reduce the overall power consumption of the network. We implement the gTBS on a testbed and we show that it reduces the power consumption by a factor of $20\% - 55\%$ compared to traditional TBS. It also reduces the delay by $54\% - 145\%$ and improves the delivery ratio by $24\% - 73\%$.**

***Index Terms*—Wireless Sensor Networks, Task-Based Sensing, Adaptive Power, Sleep and Wake-up.**

I. INTRODUCTION

Wireless Sensor Networks (WSN) consist of many sensor nodes deployed to perform different sensing tasks. The nodes communicate to either share or forward the collected data to a specified sink. Each node can have one or more sensing capabilities depending on its components and the task assigned to it [1], [2]. Sensors are widely used to offer basic functionalities such as monitoring, positioning, forecasting, etc. Nevertheless, the WSN research still ought to investigate many challenges including their diverse applications, unique network topology, unique traffic characteristics, and most importantly their severe energy resource constraints. When battery powered sensors are deployed in large numbers and in harsh or remote environments, the network lifetime becomes constrained by this limited and inaccessible energy source. It is hard to replace batteries in large magnitudes in such environments. Consequently, energy efficiency can hinder the goal of sensing if the energy source is depleted. It is found that the main cause of sensors energy consumption is idle listening and overhearing [3]. Therefore, further research is needed to reduce the amount of energy wasted during these non-active sensor states [1]. Previous studies on energy efficiency of WSNs employed two major schemes: adaptive powering, and sleep and wake-up. In power adaptation, control for the transmission levels was adopted in [4]–[6], where the power level was optimized based on the network connectivity, link quality and lifetime measures. However, it has an extensive initialization stage and a substantial overhead that increases with the number of nodes. The timing in sleep and wake-up techniques vary as random, synchronous, and periodic sleeping. In [7] random periods of sleep are set based on the lifetime and network coverage parameters, which pose hard constraints when matching the wake-up time and communication. Synchronous sleep [8] builds on virtual clustering which requires in-channel signaling and coordination among the nodes. This requirement proves a bit hard with the increasing nodes interaction. Periodic sleep is a prediction based protocol in which the transmitter holds the burden of predicting the wake-up times for the corresponding receiver [6], [9]. It inherently gives the transmitter the responsibility to synchronize and adjust the clocks accordingly. Authors in [10] adopted task management in WSN but using simulation only where intermediary actors between the sink and the sensors collect data, take decisions, and perform the appropriate processing. In our work, we experimentally integrated the role of actors into sensors which are connected to the sink either directly or through hops [11]–[14].

There has been also quite enough work on the classical tree based routing protocols with regards to energy efficiency. As an example, source based tree creation techniques has been introduced in [15], [16]. Each node calculates its neighboring nodes and takes the path having minimum number of nodes with maximum residual energy. In [17] Energy Efficient Source Based Tree Routing with time stamp in WSN, in which energy consumption and message loss was investigated. By removing the nodes with the least energy resource so that energy efficient path can be selected and more secure and reliable network is provided. The minimum number of hops or nodes for transmitting the packets in the path is also selected to conserve energy. These techniques among many others are investigating the route selection based on the available energy information but did not approach energy efficiency from conservation point of view to elongate the network sensors lifetime, as we do in our approach.

To address the above highlighted problems, we propose a novel gTBS scheme. The conceptual design started with merging power adaptation with sleep and wake-up. This combination poses a challenge in the synchronization process, nodes availability, and overall network efficiency. The idea of utilizing the Task-Based Sensing, on top of the combination, as a technique to handle the synchronization, leads to localized and relaxed constraints. In comparison with alternative techniques, the gTBS benefits from the added advantages of classical approaches while alleviating the underlying difficulties. For instance, we believe our approach is unique as we avoid using addresses for routing [18], and avoided using the lots of details in routing tables used by classical routing techniques [17]. We rather use tasks to increase the probability of having nodes go to sleep whenever the requested tasks are not within the assigned tasks of those nodes. The intended cluster of nodes associated with the task will be triggered for a wake-up while other nodes will remain inactive. The difficulties are alleviated with this optimized cluster-task path setting, paving the way for a novel and significant energy saving design. The proposed protocol forms a combination of a low-complexity and efficient power adaptation, a sleep and wake-up scheme, and a gradient-based unicast. Moreover, adoption of the combination outperformed the individual solutions not only in terms of energy efficiency, but also in delay and packet dropping [4]–[10].

This combination of the classical techniques, in fact, came with challenges that we had to work around. Synchronization as explained above was the most challenging. In addition, determining parameters off-line for the power adaptation settings was not straight forward and we had to use devices from the Micro-sensors Laboratory to perform the measurement. Finally, integrating all the different techniques into a reliable and functioning senors network while keeping in mind energy efficiency was a challenge. We will explain these challenges in details as we go through our work in this paper.

In this paper, the contributions are as follows:

1) Proposing a gradient-based transmission to alleviate signaling within the network and presenting its implementation procedure.
2) Proposing an initialization stage in which a one-time off-line optimum value for the Adaptive Power Transmission parameter is estimated.
3) Proposing a power adaptation scheme based on the estimated off-line parameters that can be used on-the-fly without introducing an overhead. Thus, we obtain a more efficient design and lower energy consumption.
4) Proposing a task driven sleep and wake-up scheme where network modes play no role in activating other nodes. Instead, the route is set by the sensing task and the data to be sent between transmitting or receiving nodes.
5) Performing a representative pilot-test of the proposed sensing network, using TelosB sensors, as a proof of concept.

The rest of this paper is organized as follows. In Section II, the system model is presented. In Section III, the gTBS techniques are described. The evaluation methodology is presented in Section IV. Numerical results are presented in Section V. Finally, the paper is concluded in Section VI.

II. SYSTEM MODEL

A. Task-Based Sensing Description

We consider a WSN topology composed of a sink or gateway and a certain number of sensor nodes. The sensor nodes send data to the sink as shown in Fig. 1. The sensing is performed in a form of tasks initiated by the sink and broadcasted to the rest of the network. Each task is associated with a certain number of nodes depending on particular characteristics (*e.g.,* location, sensor type, etc.). A task is characterized by different parameters: type of sensing, number of sensing operations, period of sensing and the intended nodes (nodes required to sense data). For example, a temperature sensing task for a month can be defined as i) sense the temperature, ii) for 30 times, iii) with a period of 1 day, iv) from nodes located in the rectangle $[x1, x2, y1, y2]$. We assume that the nodes are randomly distributed but have a fixed location. In addition, data can be transmitted to the sink using multi-hop if there is no direct line of sight (LOS) to the sink. We also assume that the sink has a pre-known ID, which is ID=0, and has no power limitations. Each node has a unique ID (set in an ID assignment phase), a hop-order and can execute one task at a time. The hop-order or simply $order$, is defined as the minimum number of hops to reach the sink. So, the sink has order 0, and a sensor node that is in a direct LOS with the sink has order 1, etc.

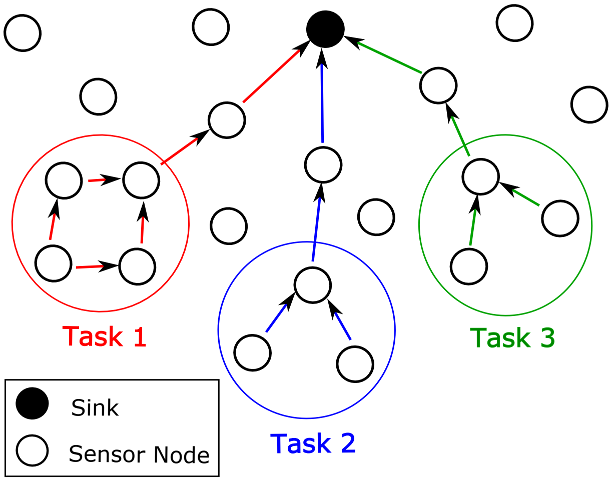

Figure 1. Task-Based WSN topology

B. Packets structure

We define two types of packets used in our protocol: ID ($Control$) packets and ($Data$) packets. The ID packet is composed by three parts, the sender's ID, order and gradient ID. An ID packet is represented as follows [senderID, senderOrder, senderGradientID]. The data packet is composed of $x+2$ components where x is the number of sensed phenomena. For example, if the nodes sense the temperature and the humidity, then $x=2$ and the Data packet is represented as [senderID, senderOrder, sensedTemperature, sensedHumidity].

C. Protocol Design

The proposed protocol involves three different phases: ID assignment, requesting tasks and receiving sensed data.

1) ID assignment phase: We adopt a two level-based ID assignment, with a hierarchy set according to proximity to the sink (order 0, ID=0). All nodes with LOS to the sink are set to order 1, and then the order of further away nodes increases accordingly. Moreover, within each cluster of nodes unique IDs are assigned using a 14-bit random number generation. This mechanism allows to have up to 16,384 IDs leaving a very low probability that two nodes will have the same ID. Henceforth, this negligible conflict will lead to a reduction in retransmission and network overloading in this initialization phase. The ID assignment algorithm implemented at all nodes is given by Algorithm 1.

Algorithm 1 ID Assignment Algorithm

```
Initialize $myOrder = 1, isAssign = 0$.
**ID Assignment Loop**
**if** An ID packet is received, i.e, [$senderID$, $senderOrder$, $senderGradientID$] **then**
    **if** ($isAssign = 0$) **then**
        Generate random ID and affect it to $myID$
        Set $isAssign = 1$
        Set $myGradientID = senderID$
        **if** ($myOrder < senderOrder$) **then**
            Set $myOrder = senderOrder + 1$
        **end if**
        Send [$myID$, $myOrder$, $myGradientID$] to node with ID=$myGradientID$
    **end if**
    **if** ($myOrder > senderOrder$) **then**
        Broadcast [$senderID$, $myOrder$, $myID$]
    **end if**
    **if** ($myOrder < senderOrder$ and $senderGradientID = myID$) **then**
        Send [$senderID$, $myOrder$, $myGradientID$] to node with ID=$myGradientID$
    **end if**
**end if**
**End of ID Assignment Loop**
```

A gradient of a node is defined as the closest node that insures a flow of data to this node and from this node to the sink. Moreover, by simultaneously setting the gradient in this phase, we establish the basis for the gTBS scheme. The gradient of a node is the node through which the ID assignment was forwarded from the sink for the first time. Hence, the ID assignment phase, when based on the gradient concept, significantly contributes to the energy saving both during initialization and throughout the network operation lifetime.

Table I
DATA FORWARDING AND SENSING ALGORITHM

	order>myOrder	order<myOrder
ID $= 0$	**Ignore**	If myID $\in$ iID : **Sense and Send to Gradient**
		If myID$\notin$ iID : **Broadcast**
ID $\neq 0$	**Broadcast**	**Ignore**

2) Task request phase: After receiving all new ID's, the sink starts to broadcast tasks throughout the sensor network. A task contains the required parameters for sensing (type, number, period, and intended nodes). Each node only considers a task received from its gradient. Only intended nodes and their gradient will be active during the task period. The rest of the network is inactive but ready to receive any other task request.

3) Forwarding Sensed Data: The intended sensors perform the sensing at assigned periods and forward the data to the sink via their gradients. When the sink receives a data packet, it identifies the received ID and data, and then stores them with the corresponding time-stamp.

III. IMPLEMENTATION

This section highlights the key implementation challenges including defining the gradient transmission, working with adaptive powering, and finally defining the sleep and wake-up periods.

A. Gradient-Based Transmission

The gTBS scheme utilizes a gradient-based transmission, which allows the node to transmit the sensed data through their gradients till reaching the sink. Hence, a single path is used for each sensed data in a task. This approach reduces the overall network activity as data flow into reduced number of nodes, and consequently reduces energy consumption of the network data diffusion process cumulatively. The challenges behind implementing the gradient concept is i) how to choose the gradient of each node and ii) which forwarding or transmission rules should be applied to the nodes in order to reduce unnecessary transmissions. The gradient of a given node is determined during the ID assignment phase. When the ID assignment request is broadcasted by the sink, the first node that forwards this request to the given node is its gradient. Afterwards, the algorithm that determines the response of the nodes towards any received data packet is summarized in Table I. The node with order $myOrder$ ignores the data packets in two cases: First, if it comes from higher order nodes and its ID=0 (*i.e.,* request broadcasted from nodes further away from the sink). Second, if it comes from lower order nodes and its ID≠0 (sensed data from nodes closer to the sink). In the other cases, if the node ID, $myID$, is included in the indented nodes ID set, iID, the node performs the sensing and broadcasts the data. In the rest of the cases, the node broadcasts the data packet and includes its own order.

B. Adaptive Power Transmission

Adapting the transmission power, denoted P_{tx}, is an important step to save energy. In many cases, using a single power level for all the nodes is inefficient [19] when the received signal to noise ratio (SNR) largely exceeds the SNR threshold of an acceptable reception, (*i.e.,* the signal is totally decoded). An intuitive and low-complexity indicator of the SNR is the received signal strength indicator (RSSI) when the noise is almost the same. In addition, we denote by RSSI_{th} the RSSI corresponding to an acceptable reception. In order to adapt the power, we model the RSSI-P_{tx} relationship using a linear equation as follows,

$$RSSI = aP_{tx} + b. \tag{1}$$

where a and b are empirically determined off-line parameters that need to be determined for each sensor device. In our case we used the TelosB motes (Type: TPR2420). The objective is to reduce P_{tx} but keeping the RSSI above RSSI_{th}. For this purpose, we perform measurements on real TelosB transmissions to determine the values of a and b. We, first, perform RSSI measurements (in dBm) with different P_{tx} values to determine RSSI_{th} and the parameter a using the corresponding slope. Note that RSSI_{th} can be analytically determined as in [20]. Then, after deployment of the sensors, b is determined using, in the first step, the maximum value of P_{tx} and the measured RSSI. In fact, b is related to the distance between the two terminals. Since the nodes have fixed locations, b is a constant. Hence, we determine the optimal P^*_{tx} as,

$$P^*_{tx} = \frac{\text{RSSI}_{th} - b}{a}. \tag{2}$$

In Fig. 2, we perform RSSI measurements as a function of power levels for TelosB motes with various distances (D=1,2,4,5,6,7,9, and 10m). The power levels are integers from 0 to 31 obtained from the CC2420 datasheet [21]. In Table II the corresponding power values in dBm are presented. Finally, we obtain $a = 0.6452$ and $\text{RSSI}_{th} = -70$ dBm. For example, if the RSSI is -45 dBm for $P_{tx} = 31$, then $b = -65$ and the optimal power level is $P^*_{tx} = 8$. Note that broadcasting is performed with the higher power level in order to reach the maximum number of nodes. However, transmission to the gradient is achieved with the adapted power level. Due to channel variation, the power can be adapted each time the node receives a packet from its gradient. However, this may include additional processing and delay that can affect the network activity which introduces a trade-off that should be considered empirically. In our work, we update the power at the start of each new task.

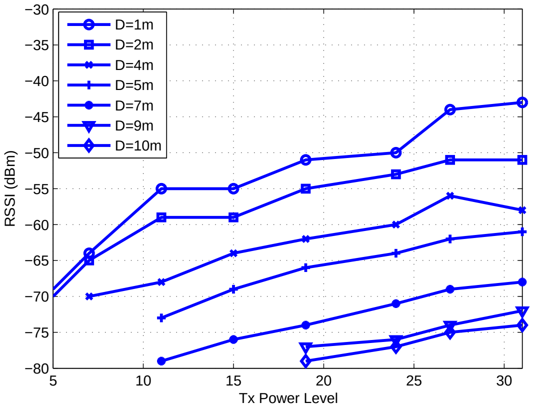

Figure 2. Transmission power levels vs RSSI for different distances.

Table II
CC2420 POWER LEVELS WITH OUTPUT AND CURRENT DRAW.

CC2420 Power Level	Output Power (dBm)	Current Drawn (mA)
31	0	17.4
27	-1	16.5
23	-3	15.2
19	-5	13.9
15	-7	12.5
11	-10	11.2
7	-15	9.9
3	-25	8.5

C. Periodic Sleep and Wake-up

The objective of the sleep and wake-up is to allow the nodes to sleep opportunistically when they are inactive during an execution of a certain task or when waiting for a new task. There are two types of sleep states a node can take:

- Periodic Sleep: Depending on network inactivity (*i.e.,* no tasks are required), the nodes are put into periodic sleeping, where the nodes alternate between sleep and wake-up states. Alternating between the two states takes milliseconds to be able to receive any upcoming task.
- Continuous Sleep: In case of being intended to sense within a task, a node may go into a sleep state between two sensing actions and then wake-up to sense and sleep again. This continuous sleep can be quite extremely large in case the sensing period is in hours or days.
- Task-Based Sleep: The way our approach is linked to Task-Based sensing is explained in the following. The Sink node sends a sleep flag in the data request to indicate whether or not inactive nodes should sleep. Inactive nodes are those not required to sense or are not in the sensing path towards the sink (not gradient). At the start of each task sensing, this flag is set to FALSE to insure all nodes are awake. Once the task is sent to all the nodes in the network, and the nodes associated with that task are identified, and gradients are also identified, only then Sink node broadcasts a data packet including a sleep flag with TRUE status, so that all inactive nodes enter in a sleep mode till the end of the task sensing (as they are not associated with the task and not gradients of other nodes).
- Sleep and Wake-up: On the other hand, when a sensor node is booted and the stay-awake flag is set to FALSE. It stays awake for a certain amount of time and if it receives no packets during that time it goes to sleep for a pre-set amount of time and wake-up periodically to listen for any

incoming packets.

- Sleep and Wake-up combined with Power Adaption: When sensor receives a task then first step checks if it came from its own gradient, and if so, then the adaptive power setting is updated based on the RSSI and power values extracted.

  If this packet happens to be the first one this sensor node receives in the series of data request, it records the start time of duty cycle and sets the corresponding flag to indicate that a cycle has started.

  The second step is to check that it is a data request packet. If it is, the number of received messages in incremented, and if the sender's order is lower than the sensor node's own (i.e. the packet comes from a node closer to the sink than the current sensor node), then the following is executed: the time between the first and second data requests is calculated and stored as the message period (to be used in determining how long the sensor node should sleep). When the third or later data request packet arrives, the packet's sleep flag is checked as well as the node's own staying awake flag. If the first is TRUE and the latter is FALSE, that means the node is neither the requested sensing node nor is it in the path between the sensing node and the Sink node (not gradient). Such node goes to sleep for an amount of time equal to the calculated request period multiplied by the number of data requests left to be sent (which is included in the packet).
- Active Nodes: Our approach identifies the nodes which should stay awake. Active nodes (with awake flag set to TRUE) are either nodes required to sense data within a task sensing request or nodes in the path for the data sensing (gradient for other nodes).

IV. TESTBED SETUP

In the next section, we present the implementation of these techniques along with the task-based sensing on real testbed of sensors. In order to evaluate the proposed gTBS scheme, we adopt a network of TelosB motes. First, we present our evaluation setup, then the methodology we used. We describe the adopted energy model and the way we searched for optimal sleep mode settings in the TelosB motes.

A. Experimental Setup

Our evaluation setup consists of a sink which is connected to a PC surrounded by n TelosB sensor nodes where $n \in \{1, ..., 6\}$. The novelty of our work is in the sensing technique while combining the three schemes. We use the motes merely as tools to proof the concept. These TelosB motes (Type: TPR2420) integrate (TI MSP430) microcontrollers as the main processing unit, referred to as (MCU) in the rest of this paper, and the radio RF chip of type (Chipcon CC2420) as its radio transceiver. Our calculations are based on the datasheets of TelosB motes and Chipcon radio [21]. The datasheet values form the basis for our analysis and facilitates the composition of the appropriate network-related power consumption model. These TelosB mote sensor nodes are connected to a PC running TinyOS application in order to collect the measurements from each node when the sensed data are transmitted to the sink.

B. Methodology

In our testbed, we collect two types of data. First, is the sensed data, which include temperature measurements. Second, is the measurements required for our analysis. These measurements include sink ID number, sensor node ID number, gradient, number of sensor nodes, number of requested events, total transmit time, total delay at the sink, total duty time, sent packets, received packets, supplied voltage, and finally the sleep time of each sensor node. We start our evaluation by assigning ID numbers as explained previously. When IDs assignment is complete, the sink sends a task to the sensor nodes. Afterwards, the intended nodes periodically forward the sensed data through the gradient back to the sink during the task period. We repeat the same steps while incrementing the number of intended nodes by one, then measure and record the required analysis parameters. For benchmarking, we compare the proposed gTBS scheme with a simple TBS scheme where none of the three energy efficient techniques is adopted.

C. Energy Calculation

Energy calculation for each sensor node is based on the summation of the energy of each component obtained from the above measurement steps. Total Energy is denoted E_T.

$$E_T = \sum \begin{cases} \text{Energy of active MCU when processing data} \\ \text{Energy of idle MCU} \\ \text{Energy of Radio when transmitting (tx)} \\ \text{Energy of Radio when receiving (rx)} \\ \text{Energy of Radio when (idle)} \end{cases} \tag{3}$$

$$\begin{aligned} E_T = &\{I_{MCU_{\text{active}}} \times V\} \times T_{\text{total duty}} \\ &+ \{I_{MCU_{\text{idle}}} \times V\} \times \{T_{\text{total duty}} - T_{\text{sleep duty}}\} \\ &+ \{I_{R_{\text{tx (at 0 dBm)}}} \times V\} \times T_{tx} \\ &+ \{I_{R_{\text{rx}}} \times V\} \times T_{rx} \\ &+ \{I_{R_{\text{idle}}} \times V\} \times \{T_{rx} + T_{tx} - T_{\text{sleep duty}}\} \end{aligned} \tag{4}$$

The $Current$ drawn in each state is denoted by I, and the $Supplied\ Voltage$ is V. The variable T represents the $Time$ that the TelosB MCU or radio chip spent in each operation state (idle, active, transmitting tx, receiving rx) or in sleep state.

D. Optimal Sleep Mode

In order to determine the optimal sleeping mode to use in our sleep and wake-up scheme, we establish an experimental setup for analyzing the different sleep modes of the MCU. In addition, we analyze the power consumption when the radio chip (RF) is either on or off. A *Techtronic* power supply is used to provide a $3V$ to the motes, and the current consumption is measured using $LabView$ application.

1) MCU Sleep modes: The MCU includes different levels of sleep based on the functionality and components whose operation is compromised for exchange of power savings. The levels vary in severity starting from LPM0 (minimal power reduction) and up to LPM4.5 (MCU goes into *deep sleep* until activated by an external interrupt). In Fig. 3, we investigate the power consumption of the LMP0 and LPM1. We show that in comparison to the idle state, in which the MCU is not performing any computations, the levels of sleep do not provide any added advantage. On the contrary, a current spike appears whenever the system goes into and out of the sleep state. Since negligible savings were obtained from this analysis, no sleep state for the MCU should be adopted, not to mention the added cost of switching between states. This approach allowed the nodes to be aware of the sensing task at hand for an added simplification of clocking and scheduling processes. Thus, the synchronization is being governed by the task itself, as each node is aware of its own gradient and the route to send the data through.

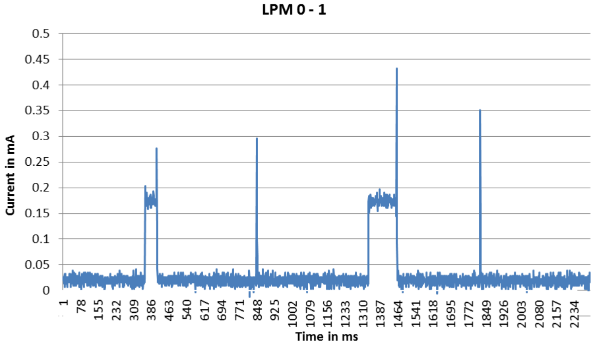

Figure 3. LPM0 and LPM 1 Power Consumption.

2) Radio Sleep Mode: Here, we try to quantify the actual consumption of the radio component and how much actual savings can be achieved from a single device employing sleep and wake-up. The experiment run by toggling the radio between the active and sleep modes. Fig. 4 shows the experimental results of the test where the spikes correspond to the actual switching of the radio component and the step pulse is for the operation of the LEDs. This implies that the radio RF chip consumption is dominant among the other mote components, and the capability of putting it into sleep state is a key factor for further savings. Therefore, we adopted the control of the radio component only. Moreover, the MCU operation was kept intact due to the relevance of the accuracy of the clocking to the overall operation of the system. With this implementation at hand, savings up to 40% were possible in the operation of a single mote.

Table III shows the different current values drawn by the Microcontroller (MCU) and Radio Chip, which are associated with each power mode state of the device when in operation. The table also shows the type of MCU used in the TelosB motes and the Radio chip installed [21], [22].

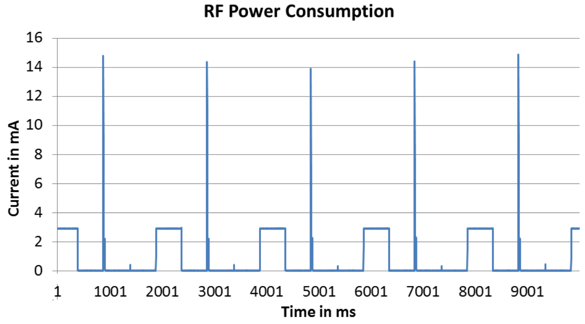

Figure 4. RF Power consumption for the sleep and wake-up states.

Table III
TELOSB DATA SHEET (TPR2420)

MCU (Processing Data)	1.8 mA	Active
Type: TI MSP430 microcontroller	5.1 μA	Sleep
Radio (Sending/Receiving)	17.4 mA	Send
Type: ChipCon CC2420	19.7 mA	Receive
RF Power = 0 dBm	21 μA	Idle
Data Rate = 250 kbps	1 μA	Sleep

V. EVALUATION RESULTS

In this Section, we examine the power consumption of gTBS in comparison to the simple, non-green TBS that does not implement any energy efficient mechanism. Our comparison is based on energy consumption, average delay, and event delivery ratio.

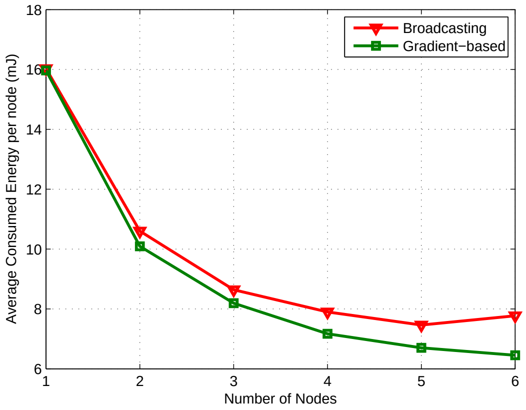

Figure 5. ID assignment energy consumption: comparison between gradient-based and flooding transmissions.

Fig. 5 plots the average consumed energy per node versus the number of sensor nodes during the ID assignment phase. We compare the gradient-based transmission with the classical broadcasting which corresponds to broadcasting any received packet in all the cases mentioned in Table I. The average energy is computed by dividing the total consumed energy (including the energy of the sink) by the number of sensor nodes. Hence, we highlight the total energy needed to assign ID to a certain number of nodes. Since the energy consumed by the sink is high due to the continuous sending and receiving, this average energy decreases as the number of node increases. In addition, we note that using gradient-based transmission reduces the average energy by 5%, 9% and 17% of the broadcasting communication for 2, 4, and 6 nodes, respectively.

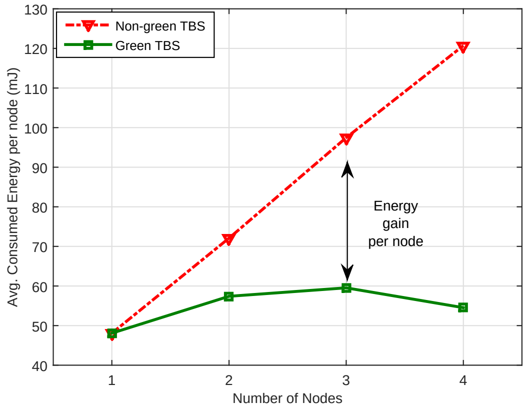

Figure 6. Energy Consumption of gTBS and non-green TBS schemes.

Fig. 6 presents the consumed energy of gTBS versus traditional TBS (non-energy efficient schemes). The gTBS scheme implements the gradient-based transmission, the power adaptation, and the sleep and wake-up schemes. The used task is a temperature sensing task for 10 times within a period of 1 second, and the intended nodes are all available nodes. We notice that gTBS is equivalent to traditional TBS when only one sensor node is used since the node is always awake and transmits with the same power to the sink. However, as the number of intended nodes increases, the consumed power per node increases remarkably for the traditional TBS, but slightly increases for the gTBS. In fact, the energy consumption reduction of the gTBS compared to the traditional TBS is 20%, 39% and 55% for 2, 3, and 4 nodes, respectively. Hence, the gTBS scheme significantly improves the energy savings when the number of nodes increases further. We also observe that the gTBS energy consumption per node starts to decrease when the number of nodes exceeds 3 to match Fig. 5.

In Fig. 7, the event delivery ratio (EDR) is plotted as a function of the number of nodes for gTBS and traditional TBS. The EDR is defined as follows,

$$EDR = \frac{\text{Received Packets}}{\text{Sensing Requests } \times \text{Intended Nodes}} \tag{5}$$

The tested task is the same as in Fig. 6, where the intended nodes are all available. We notice that all the nodes are required to sense and send data which introduces some losses in data delivery for traditional TBS. This observation is mainly caused by two facts. First, when the gradient-based communication is not used, all nodes are broadcasting the sensed data. Hence, each node receives, decodes, and identifies every forwarded packet even if it is not a gradient. This process produces an instantaneous high traffic in the network in a similar fashion as flooding and results in loss of packets. Second, transmitting with the maximum power (non-adaptive) will cause a high interference between the nodes which will then affect the link quality. However, compared to the traditional TBS, the gTBS scheme boosts the EDR gain to 54%, 145% and 137% for 2, 3, and 4 nodes, respectively. This gain is a result of unicasting packets and lowering power transmission as implemented in the gTBS.

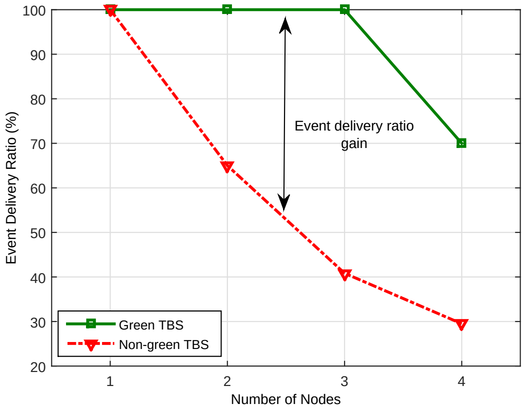

Figure 7. Event Delivery Ratio of gTBS and traditional TBS schemes.

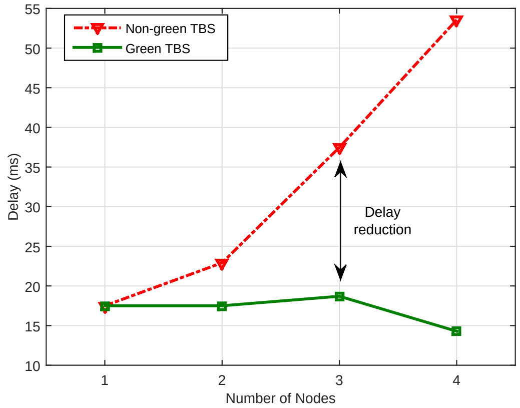

Figure 8. Average delay of gTBS scheme compared to traditional TBS.

Fig. 8 compares the average delay of gTBS and traditional TBS. The delay is the time between transmitting a task and receiving the corresponding data at the sink. We observe that when the number of nodes increases, the delay of the traditional TBS increases. This is due to the high interference and high traffic. In addition to packet loss, due to the CSMA protocol implemented in the TelosB motes, nodes have to wait for the channel to be free in order to send and forward packets which encompass further delays. On the other hand, the delay reduction of the gTBS compared to the traditional TBS is 24%, 50%, and 73% for 2,3, and 4 nodes, respectively. Hence, the gTBS significantly improves the delay.

VI. CONCLUSIONS

This paper introduced a green Task-Based Sensing (gTBS) protocol. This design allows the WSN to sense data in the form of tasks directed to a cluster of nodes. The gTBS adopts gradient-forwarding, power adaption, and sleep and wake-up techniques to increase the energy efficiency of WSNs. We evaluated our gTBS protocol on TelosB motes network. We showed that gTBS reduces the energy consumption by $20\% - 55\%$. Moreover, we also obtained a reduction of the delay by $54\% - 145\%$, and an enhancement of $24\% - 73\%$ of the event delivery ratio. In future work, although we performed as a starting point a representative pilot-test for a proof of concept, we however plan to expand our scalability impact analysis to include a variation of the higher number of nodes in order to claim an overall superiority of our proposed gTBS scheme.

REFERENCES

[1] A. Prayati, C. Antonopoulos, T. Stoyanova, C. Koulamas, and G. Papadopoulos, “A modeling approach on the TelosB WSN platform power consumption,” *Journal of Systems and Software*, vol. 83, no. 8, pp. 1355–1363, 2010.

[2] A. Tripathi, N. Yadav, and R. Dadhich, “Secure-spin with cluster for data centric wireless sensor networks,” in *2015 Fifth International Conference on Advanced Computing and Communication Technologies (ACCT)*. IEEE, 2015, pp. 347–351.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong, and J. Rabaey, “Distributed algorithms for transmission power control in wireless sensor networks,” in *IEEEWireless Communications and Networking (WCNC’03)*, vol. 1. IEEE, 2003, pp. 558–563.

[5] M. Tahir, N. Javaid, A. Iqbal, Z. A. Khan, and N. Alrajeh, “On adaptive energy-efficient transmission in wsns,” *International Journal of Distributed Sensor Networks*, vol. 2013, no. 10, p. 10, 2013.

[6] D. Son, B. Krishnamachari, and J. Heidemann, “Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks,” in *IEEE Communications Society Conference onSensor and Ad Hoc Communications and Networks (SECON’14)*. IEEE, 2004, pp. 289–298.

[7] M. Balakrishnan, E. E. Johnson, and H. Huang, “TEAN-Sleep for Distributed Sensor Networks: Introduction and α-Metrics Analysis,” in *IEEE Military Communications Conference (MILCOM’07)*. IEEE, 2007, pp. 1–7.

[8] W. Ye, J. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[9] B. Jiang, B. Ravindran, and C. Hyeonjoong, “Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 735–747, 2013.

[10] M. Khan and B. Rinner, “Energy-aware task scheduling in wireless sensor networks based on cooperative reinforcement learning,” in *2014 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2014, pp. 871–877.

[11] Y. Song, L. Liu, H. Ma, and A. V. Vasilakos, “A biology-based algorithm to minimal exposure problem of wireless sensor networks,” *IEEE Transactions on Network and Service Management*, vol. 11, no. 3, pp. 417–430, 2014.

[12] Y. Yao, Q. Cao, and A. V. Vasilakos, “Edal: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wire-less sensor networks,” in *10th IEEE MASS*. IEEE, 2013, pp. 182–190.

[13] Y. Xiao, M. Peng, J. Gibson, G. Xie, D. Du, and A. V. Vasilakos, “Tight performance bounds of multihop fair access for mac protocols in wireless sensor networks and underwater sensor networks,” *IEEE Trans. Mob. Comput.*, vol. 11, no. 10, pp. 1538–1554, 2012.

[14] Y. Xiao, Z. Yanmin, K. Linghe, L. Cong, G. Yu, A. V. Vasilakos, and Y. Min, “Cdc: Compressive data collection for wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2015.

[15] S. Charalambos and V. Vasos, “Source-based routing trees for efficient congestion control in wireless sensor networks,” in *IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2012, pp. 378–383.

[16] A. H. Mohajerzadeh, M. H. Yaghmaee, and Z. Eskandari, “Tree based energy efficient and congestion aware routing protocol for wireless sensor networks,” in *11th IEEE Singapore International Conference on Communication Systems (ICCS)*. IEEE, 2008, pp. 1707–1711.

[17] D. Singh, S. Sharma, V. Jain, and J. Gajrani, “Energy efficient source based tree routing with time stamp in wsn,” in *International Conference on Signal Propagation and Computer Technology (ICSPCT)*. IEEE, 2014, pp. 120–124.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *International conference on Mobile computing and networking*. ACM, 2000, pp. 56–67.

[19] J. Kim, S. Chang, and Y. Kwon, “Odtpc: On-demand transmission power control for wireless sensor networks,” in *International Conference on Information Networking (ICOIN’08)*. IEEE, 2008, pp. 1–5.

[20] M. Zuniga and B. Krishnamachari, “Analyzing the transitional region in low power wireless links,” in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (SECON’04)*. IEEE, 2004, pp. 517–526.

[21] Ti.com, “Analog, embedded processing, semiconductor company,” *Texas Instruments - TI.com*, 2015. [Online]. Available: http://www.ti.com. [Accessed: January-2016].

[22] Xbow.com, “Moog crossbow - the smart sensor company - super smart,” 2016. [Online]. Available: http://www.xbow.com. [Accessed: January-2016].