Decentralized multi-level duty cycling in sensor networks

Sharief M. A. Oteafy*, Hosam M. AboElFotoh+, and Hossam S. Hassanein*

* School of Computing, Queen's University,
Kingston, Ontario, Canada K7L 3N6
{oteafy,hossam}@cs.queensu.ca

+ Dept. of Math. and Comp. Science
Kuwait University, P.O. Box 5969
hosam@sci.kuniv.edu.kw

Abstract— Prolonging network lifetime while efficiently detecting and reporting events are arguably the most important objectives of Wireless Sensor Networks (WSNs). Different WSN protocols aim to improve such measures, yet partially focus on certain aspects (eg. reliability and time latency) and sacrifice others (eg. power efficiency) in application specific approaches. We present DMULD (Decentralized Multi-Level Duty cycling), a cross-layer design paradigm aiming at raising performance measures of general WSNs. It integrates tailored multi-level sleep states having varying levels of performance (hence energy consumption) with novel sensing, Medium Access Control (MAC) and routing protocols. Nodes carry on tasks in a decentralized manner with efficient load balancing. DMULD is a dynamic model which is adaptable to application specific requirements, through fine tuning its parameters. DMULD was thoroughly simulated, examining the effects of varying its parameters on network lifetime and efficiency. It achieved over double the lifetime of multi-hop CSMA/CA.

I. INTRODUCTION

Most Wireless Sensor Network (WSN) protocols are application dependent, demonstrating benefits only under specific assumptions. Consequently, protocols addressing general WSNs have seen little improvement. Generalized protocols have many advantages, such as facilitating the implementation on generic embedded systems, making mass production of Sensor Nodes (SNs) more feasible [1]. Also, such generalized protocols raise the upper bound on efficiency of WSNs in general, which sets higher standards for specific WSN protocols which follow it.

The main aim for a WSN is to detect and report the sensed events in its deployment region. It should do so with minimal energy consumption, so as to extend the lifetime of nodes, and hence network lifetime. These constitute the most important performance measures of WSN protocols. Yet, merging efficient protocols for each measure does not imply optimal efficiency for the network as a whole. Paradigms need to be considered in a broad comprehensive view to optimize energy efficiency across its layers.

We target many sources of hindrance to WSN performance, and later describe our contribution to improve them. First is redundancy in sensing; where each sensed event is usually reported by all the nodes in its vicinity. This creates multiple reports of the same event, generating multiple messages which traverse the network towards the sink(s), exhausting its resources. Redundancy in nodes should be optimized to only use the subset of nodes which are required to efficiently perform a task, and putting the rest to lower sleep states. On the MAC level control messages have to be minimized to conserve energy for the main task which is relaying and delivering data messages. Consequently, redundant nodes which do not contribute to reporting an event will not suffer from overhead of control messages.

WSN protocols which aim to improve a subset of properties of the network, may hinder overall network operation. For example energy-aware routing which exhausts shortest paths in a network, yield frequent node failures early in network lifetime, thus affecting connectivity and coverage for the whole network [2]. Similarly protocols which aim at absolute fairness in distributing tasks over all nodes will impose significant time latency in network operation [1].

Thus a great need to devise general WSN protocols which holistically view the main tasks of the network exists; optimizing its internal activities to efficiently perform these tasks. Taking into consideration aspects such as energy-efficiency, reliability, load balancing and network lifetime.

We present DMULD (Decentralized MUlti-Level Duty cycling), a cross-layer model for general purpose WSNs, which addresses the aforementioned problems. DMULD modifies and integrates the concept of multi-sleep states proposed by Sinha and Chandrakasan in [3], with tailored sensing, MAC and routing protocols, to build a general WSN model. At any given time, each node will deterministically be in a certain sleep state, which as explained in Section III, is distinct in its energy consumption based on the components of the node being active/idle/off. Our MAC protocol adopts minimal communication between nodes; and routing is based on dynamically choosing the best route to the sink minimizing both energy consumption of nodes (on path or surrounding it) and the time latency to deliver the event to sink.

Hence three main contributions are presented. First a multi-level duty cycling cross-layer approach is adopted, treating sensor nodes as components rather than whole nodes, and fine tuning the power consumption of each to satisfy its given tasks. This is integrated with a new sensing protocol which aims at conserving network resources by generating only one report per sensed event in the network in a decentralized manner. A modified MAC protocol is derived to account for such a multi-level mechanism. Finally a decentralized approach engulfs all operations of DUMLD to increase its dynamicity in adapting to varying network parameters and preserving high fault resilience.

The remainder of the paper is structured as follows. Section II covers the main preliminaries and work related to our model, which is stated and formalized in Section III. It also presents a Deterministic Finite State Machine (DFSM) for the node states and their transitions. Following that Section IV presents analysis and simulation results of testing DMULD and comparisons with other protocols;

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

namely two variants of the CSMA/CA protocol. Finally Section V states our conclusions.

II. Preliminaries and Related Work

The most stringent constraint of WSNs is the limited amount of energy nodes posses. Because of the large number of nodes [4], and the possible harshness of environment where they are deployed, we cannot practically assume that the batteries of nodes will be replenished after depletion [3], [4], and [5]. Power management in WSNs has been investigated on many levels. Decreasing the power consumption at each of these levels targets prolonging network lifetime. The remainder of this section describes two main methods of power management in WSNs.

A. Duty Cycling

Nodes should not remain active all the time. If all the components of a node are on, over a relatively short period of time, the node will run out of energy and die. *Duty cycling* is the mechanism of putting the node in a sleep state for a certain duration, during which its activity would be superfluous (or non-critical) and switching it back to active state when it is needed. This depends on many factors [1], [6]. The most important factors to optimize are the energy gained from, and overhead of, switching from a state to the other. This would be in terms of power consumption as well as time wasted during switching, called *switching latency*. Different protocols control duty cycling to optimize their performance, as compared in [6]. A *duty cycle* represents the proportion of time a node is active [6].

Multi-sleep states presented in [3] aimed at introducing the concept of partitioning the sensor node into three main components (Microcontroller, sensor and transceiver units), and presenting the feasible combinations of activity levels for each component in the node. Thus duty cycling shifted from simply switching the node on and off, to controlling the activity level of each of its components. Yet the authors presented the scheme as a power model, without addressing how these power models could be utilized in WSN protocols.

B. Mutli-hop Routing

The distance between nodes and sink(s), and the dominance of power consumed in transmission relative to that distance, dictate multi-hop routing as being essential for efficient power management in general WSNs [7]. Although routing is highly application dependant, it should not overlook energy-awareness to improve other parameters of the network. Two important aspects of routing are the time delay in reporting a message back to sink (approximated by number of hops on path and delay in each hop) and the total energy spent sending this packet; and balanced on the number of nodes contributing in relaying.

An example of such a routing mechanism is implemented in the multi-hop Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA). It follows shortest path routing to deliver data packets back to the sink [8].

TABLE I
Sleep States of a Sensor Node

State	Memory	Sensor	Processor	Radio	Power Consumption (mW)
S_{Tx}	Active	On	Active	T_X	27.46[a]
S_{Rx}	Active	On	Active	R_X	22.2
S_{Listen}	Idle	On	Active	listen	22.06
$S_{Compete}$	Idle	Off	Active	listen	≈18
S_{wait}	Idle	Off	Idle	Off	5.92
S_{Off}	Off	Off	Idle	Off	0.02

[a] for transmission power of 0.7368 mW at 19.2 kb/s using ASK

III. Multi-Level Duty Cycling Model

Our Decentralized Multi-level Duty cycling approach (DMULD) is a cross-layer paradigm designed to span general WSNs. It integrates sensing, routing and MAC protocols in a multi-level duty cycling mechanism. Each node can reside in one of six different states, distinct by their power consumption and active components of the node. Each node decides its sleep state based on events occurring in its surrounding environment, communication with neighbouring nodes, and internal triggers. The remainder of this section presents a detailed description of the cross-layer model components, and the different states.

A. Assumptions and Definitions

To achieve generality in DMULD, only broad assumptions spanning general WSN applications are considered. Nodes are assumed to be homogenous, have unique ID's, and their transmission range could be varied. The sink has unlimited power supply and is placed on the border of the deployment region. No assumption is made on node placement and is assumed random. For a WSN with a set of **N** nodes, the *Euclidian* distance between two nodes ***i*** and ***j*** is represented as **dist**(***i***, ***j***).

For a given node ***i***, its radius of transmission and sensing are denoted by $\mathbf{R_{Tx}}(\boldsymbol{i})$ and $\mathbf{R_S}(\boldsymbol{i})$ respectively, and its battery energy as $\boldsymbol{n_i}$(batt). The set of neighbours of node ***i*** denoted by $\boldsymbol{ns_i}$ is calculated as:

$$ns_i = \left\{ j : j \in N \wedge \text{dist}(i, j) \leq R_{Tx}(i) \right\}$$

For any two nodes ***i*** and ***j***, if $\mathbf{dist}(\boldsymbol{i}, \boldsymbol{j}) \leq \mathbf{R_{Tx}}(\boldsymbol{i})$ then ***j*** is one-hop-away from ***i***. Hop value of node ***i*** is denoted by $\boldsymbol{n_i}$

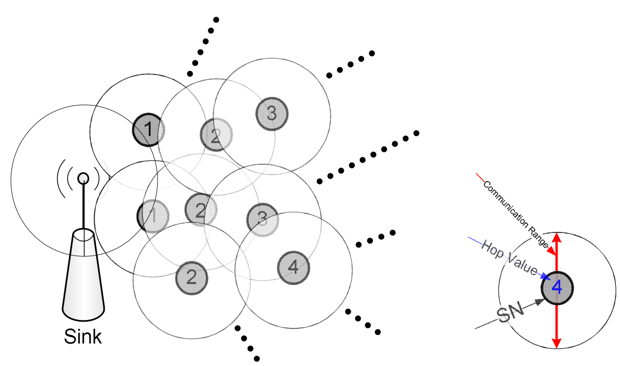

Figure 1: Phase I: Setting hop values of nodes

TABLE II
MESSAGES USED IN DMULD

Message	Purpose	Type
SetH	1. Modify neighbors' hop value 2. Populate table of neighbours at each node	Broadcast
Req	Initiate a Request to send a data message	Unicast
Ack_Req	Acknowledge Req	Unicast
DataP	Holds the actual data (to be reported back to sink)	Unicast
Update_ns	Re-populate a node's list of neighbours	Broadcast
Rem_n	Notify neighbours that this (sender) node is dead	Broadcast

(hop). Event occurrence is approximated by a Poisson distribution with mean λ, and the average inter-occurrence rate is denoted by τ. The position of those events is uniformly distributed over the sensed region.

B. Sleep states and D-FSM

At any given time, each node could be deterministically in one sleep state S, depending on the current event/task and its previous state. All the sleep states are listed in Table I, with relative power consumption values. The sleep states are tailored to only activate the components needed by a node to perform its task. So instead of duty cycling them on and off as a whole, specific components are targeted. When an event occurs, the node checks its current state and condition of the event, and thus decides on the next state it should transition to. The DFSM describing such states and transitions, associated with Table II, is depicted in Fig. 2.

C. Phase I – Initial Setup

This setup phase has two goals. It assigns to every node a value indicating its relative distance – hop count – from the sink, using a minimum spanning tree. The sink triggers this phase by sending its hop value of 0 to its one-hop-away neighbours in a SetH message. The sink maintains a transmission distance equal to that of a node. Each node starts with a hop value of ∞. When node n_i receives SetH from n_j, it updates its hop value iff (n_i(hop) - n_j(hop)) >1. In that case it broadcasts its new hop value in a new SetH message. Fig. 1 shows an example for setting the hop values of nodes in a WSN.

Secondly, this phase populates a *table of neighbours* for each node; as a by-product of updating its hop values based on the SetH messages received from neighbours. This table records for each neighbour, its ID and its *readiness value* ρ. This value indicates the neighbours' ability to relay a message towards the sink. The readiness value aggregates two normalized values: the hop value of the node and its battery reservoir. That is, ρ is the result of a function which sums both these normalized values after multiplying each with a given weight (eg. 50% to balance between both components). Thus if more importance is to be given to the battery value (if balancing loads over all nodes is the main target) then its weight factor would be higher.

Neighbours are ordered in non-increasing order based on their ρ values the highest being in row one. Nodes are initially in state S_{Listen}.

D. Sensing protocol

When a Sensed EVent e (SEV$_e$) occurs, all the nodes in its vicinity which detected it, represented by $\eta(\text{SEV}_e) = \{ j : j \in N \wedge \text{dist}(\text{SEV}_e, j) \le R_S(j) \}$ will contend to report it back to the sink. Only the most *fit* node n_i will carry on this task, the others will be put back to a lower power state. This is done in a decentralized manner.

Fitness – denoted by ω_i – is an aggregation of the node's hop value and its remaining energy. These values are normalized prior to aggregation; which then stresses the weight on either factors. So if battery value had a greater influence in the function then a longer path, through nodes which have more energy, will be more probable. All other nodes which competed to report SEV$_e$ and failed will go to state S_{off} for a duration of $\alpha\tau$, where α is a parameter referred to as the sleep factor.

E. Routing and MAC protocols

The application of DMULD could be generalized to any routing protocol. Here we associate DMULD with dynamic routing tables. When a message is sent, contention to access the wireless medium is resolved using the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol with Request-to-send/Clear-to-send (RTS/CTS).

Now when node n_i has a packet to send, it will look up the first node in its table of neighbours, and send a request Req message to it, and wait for a certain *timeout* period. If the receiver is ready to relay the upcoming data packet, then it acknowledges the request by an Ack_Req message. But if the timer expires without an acknowledgment, then n_i repeats the same procedure with the second neighbour in its table, and so on until a neighbour is ready to relay the message. When a neighbour confirms by the Ack_req message, n_i sends the data packet DataP and then goes to sleep state S_{off} for duration $\alpha\tau$.

If n_i goes over all its neighbours without receiving an Ack_req, then its neighbours are no longer connected to it at the moment, and it will seek other neighbours farther away by increasing its transmission radius by a given increment δ, and re-populate its table of neighbours by sending an Update_ns message. Nodes in state S_{Listen} whom hear this message will reply with their SetH messages.

The specific operations done by a node, and the states in which it will transition from and to, are depicted in the DFSM in Fig. 2, and elaborated in Table III. All the messages communicated in the DMULD protocols are listed in Table II.

F. Accommodating new/dead nodes

When a node's battery level drops below a certain small threshold, it will transmit one final message (Rem_n) before it dies out to its neighbours. This message will trigger its neighbours to remove this node from their list of neighbours, if it is there. On the other hand when a new

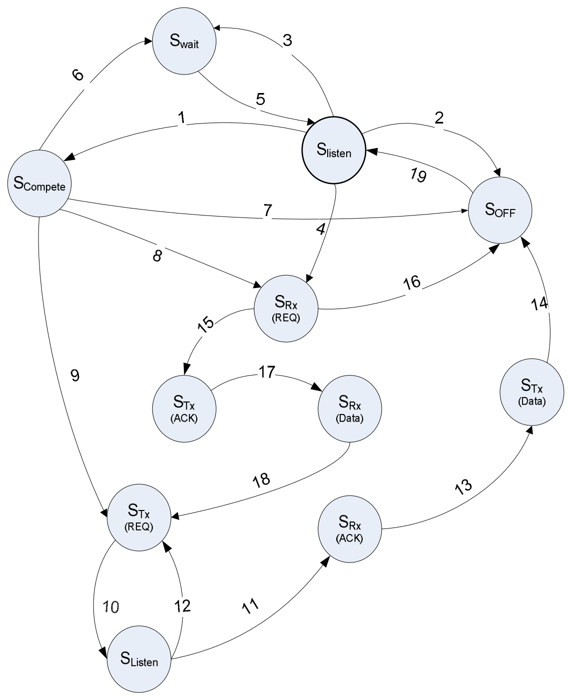

Figure 2: DFSM of a sensor

node is introduced to the network, it will send an Update_ns message to populate its table of neighbours and hop value.

IV. Analysis and Simulations

The performance of DMULD has been tested using EDWiNS (Event-Driven Wireless Network Simulator), which facilitated the simulation of a fine-tuned multi-level power model; implemented in C++, described in [1].

Without loss of generality, the following parameters where set for the simulation model. The simulated network is deployed in a 100x100 m region; populated with 200 nodes randomly positioned over the whole region. Each node has $\mathbf{R_{Tx}}(\boldsymbol{i})$= $\mathbf{R_S}(\boldsymbol{i})$=20 m, yet $\mathbf{R_{Tx}}(\boldsymbol{i})$ can increase later on as stated in protocol. The sink is placed in a corner, and $\mathbf{R_{Tx}}(\boldsymbol{sink})$= 10 m. Each node is initially powered with 1 kJ. Events arrive according to a Poisson distribution with mean inter-occurrence time of $\boldsymbol{\tau}$. Data messages are 2000 bits long, and control messages are 400 bits.

The following power-model functions, adopted from [9], represent the energy to transmit ($\boldsymbol{E_{Tx}}$) and receive ($\boldsymbol{E_{Rx}}$) a data packet of size $\boldsymbol{k}$ bits:

$$E_{T_x}(k,d) = k * \left(E_{elec} + d^{\theta} * E_{amp}\right)$$

$$E_{R_x}(k) = k * E_{elec}$$

where $\boldsymbol{E_{Elec}}$ represents the energy dissipated by the transceiver electronics (equal in T_x and R_x), $\boldsymbol{E_{amp}}$ is the transmitter amplifier power, and $\boldsymbol{\theta}$ is the path lost exponent. In our radio model $\boldsymbol{\theta}$=2; following from the Friss free space model (d^2 attenuation), since the distance between the nodes is usually short, much less than the cross-over distance beyond which it is reasonable to substitute it for 4 (two-ray ground propagation model), calculated as 86.2 m in [9]. Network lifetime is considered over when 75% of the nodes die. It is inherent from DMULD that increasing the sleep factor $\boldsymbol{\alpha}$ of nodes is beneficial in terms of power conservation and thus prolonging network lifetime, since

TABLE III

Deterministic State Transitions of a Sensor Node $\boldsymbol{N_I}$ *(for Fig. 3)*

Trans	*Current state*	*Event*	*Condition*	*Action*	*New state*
1	S_{listen}	**SEV** detected		Set timer to ω_i	$S_{compete}$
2	S_{listen}	Heard **Ack_Req**		Set wake-up timer to $\alpha\tau$	S_{off}
3	S_{listen}	Heard **Req**	Req msg ***not*** for n_i **and** Req sender's hop ≥ n_i*(hop)*	No channel access for Tx time	S_{wait}
4	S_{listen}	Heard **Req**	Req msg for n_i	Rx **Req**	S_{Rx} (Req)
5	S_{wait}	Timer up			S_{listen}
6	$S_{compete}$	Heard **Req**	Req msg ***not*** for n_i **and Req** sender's hop ≥ n_i*(hop)*	No channel access for Tx time	S_{wait}
7	$S_{compete}$	Heard **Req**	Req msg ***not*** for n_i **and Req** sender's hop < n_i*(hop)*	Set wake-up timer to $\alpha\tau$	S_{wait}
8	$S_{compete}$	Heard **Req**	Req msg for n_i	Rx **Req**	S_{Rx} (Req)
9	$S_{compete}$	Timer up		Initiate Tx **Req**	S_{Tx} (Req)
10	S_{Tx} (Req)	**Req** Tx over		Set timer to TIMEOUT	S_{listen}
11	S_{listen}	Heard **Ack_Req**		Receive **Ack_Req**	S_{Rx} (Ack)
12	S_{listen}	Timer up		Tx **Req** to next neighbor in table	S_{Tx} (Req)
13	S_{Rx} (Ack)	**Ack_Req** Rx over		Tx **DataP**	S_{Tx} (Data)
14	S_{Tx} (Data)	**DataP** Tx over		Set wake-up timer to $\alpha\tau$	S_{off}
15	S_{Rx} (Data)	**Req** Rx over	Ready to handle report	Initiate Tx of **Ack_Req**	S_{Tx} (Ack)
16	S_{Rx} (Data)	**Req** Rx over	**Not** Ready to handle report	Set wake-up timer to $\alpha\tau$	S_{off}
17	S_{Tx} (Ack)	**Ack_Req** Tx over		Receive **DataP**	S_{Rx} (Data)
18	S_{Rx} (Data)	**DataP** Rx over		Initiate Tx of **Req**	S_{Tx} (Req)
19	S_{off}	Timer up	n_i*(batt)* > min_batt_thrshold		S_{listen}

each node will spend more time in state $\mathbf{S_{off}}$. Nevertheless, as $\boldsymbol{\alpha}$ increases, the sleeping nodes sacrifice some coverage of the region, and thus some SEV may pass undetected. This effect was tested as shown in Fig. 3, which showed a small decrease in the percentage of SEV successfully detected and reported back to the sink. Thus it is noted that even at very long sleep factors such as 0.95, the percentage remained over 90%, over different values of $\boldsymbol{\tau}$. Another important factor tested was the effect of increasing the initial battery energy of nodes on network lifetime. The importance of this measure lies in investigating the load balancing mechanisms of DMULD and how it maintains performance over the long run (steady-state behavior). The two main parameters, $\boldsymbol{\tau}$ and $\boldsymbol{\alpha}$ were set to 60 seconds and 0.95. The simulations were carried out for varied initial

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

battery values from 100 to 10000 Joules. As shown in Fig. 4, the increase in network lifetime caused by increase in

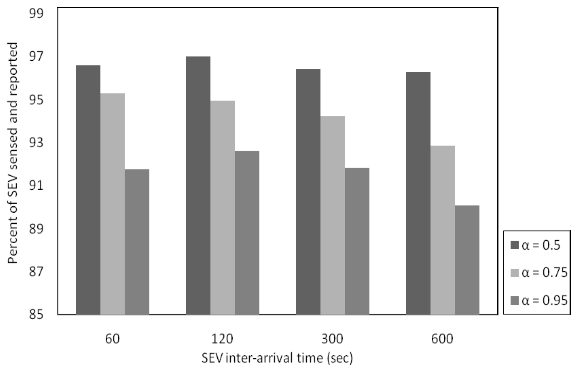

Figure 3: Effect of sleep factor ***α*** on percent of SEV sensed & reported

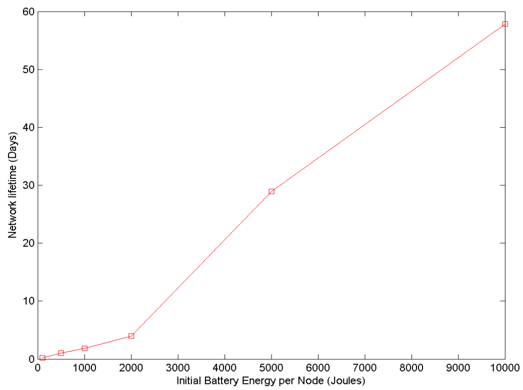

Figure 4: Effect of initial battery energy on network lifetime

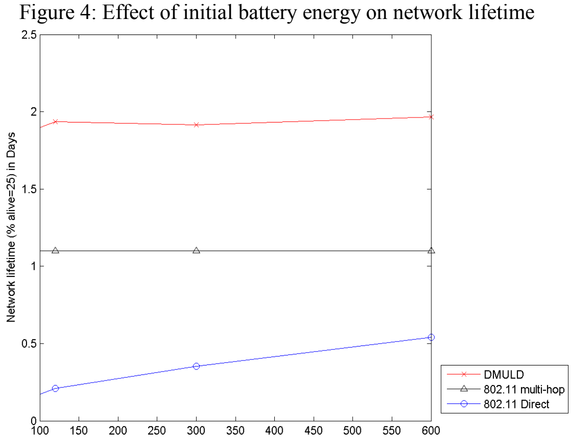

Figure 5: Comparison of network lifetime over three protocols

initial battery energy is nearly linear, facing no saturation.

Finally the performance of DMULD is compared to that of CSMA/CA. Two variants of the latter are tested. Both only depend on a two state duty cycle, either the node is on or off. The first adopts a one-hop mechanism in reporting an event back to the sink. This approach is beneficiary in that it conserves the energy of idle listening by transceivers, since nodes need not relay the packet. Yet this approach suffers from energy loss in higher power transmission to reach the sink in one hop. A more conserving approach is the multi-hop CSMA/CA, which allows the data packets to go over multi-hops towards the sink, using the non-increasing order of hop values of nodes on the routes back to the sink to establish shortest path routes and use them. Request to Send (RTS) and Clear to Send (CTS) packets are used to handle the hidden node problem. Fig. 5 depicts the difference in network lifetime, for the same environment, between DMULD, CSMA/CA with multi-hop routing and one-hop (direct) CSMA/CA. As shown, DMULD achieved more than twice the lifetime of CSMA/CA with multi-hop routing, and more than three times that of direct CSMA/CA. This is due to frequent sleeping of nodes in addition to the power conserving sensing protocol.

V. Conclusion

We proposed DMULD – a novel cross-layer design aimed towards implementing generic WSNs, optimizing the energy spent per event, and load balancing that energy on as many sensor nodes as possible. This facilitates increasing the number of redundant nodes in the WSN and benefiting from doing so, without introducing much overhead from control messages and protocols. That is, DMULD minimizes control messages and the overhead of centralized protocols. DMULD demonstrated the benefits of a multi-level duty cycle cross-layer paradigm which tailors the state of each node's components to activate only the needed ones, and put the other components to lower power states. It is the core of a cross-layer model, which minimizes power consumption when performing tasks, by activating only the necessary components to fulfil the task. DMULD eliminates redundant reports of the same event and focuses on assigning that reporting duty to the most "fit" node. That node would be the one with most energy reservoir and shortest distance from the sink. This approach eliminates a large number of repeated messages exhausting its resources and reducing its lifetime. The results obtained from DMULD when compared to common CSMA/CA protocols show elevation in performance with dynamicity to adapt to different scenarios without affecting overall performance.

References

[1] S. M. A. Oteafy, "A decentralized coordinated multi-level duty cycling approach for wireless sensor networks", M.S. thesis, Dept. Math. & Comp. Sci., Kuwait Univ., 2007.

[2] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, "On lifetime-oriented device provisioning in heterogeneous wireless sensor networks: approaches and challenges," *IEEE Network Magazine*, vol. 20, no. 3, pp. 26-33, May 2006.

[3] A. Sinha, and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 62-74, 2001.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[5] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. on Networking*, vol. 12, no. 3, 493-506, 2004.

[6] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Comm. Magn.*, vol. 44, no. 4, pp. 115- 121, 2006.

[7] J. Al-Karaki, and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Comm*., vol. 11, no. 6, pp. 6-28, 2006.

[8] V. Raghunathan, C. Schurgers, S. Park, and M. B. Sri-vastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Mag,* vol. 19, no. 2, pp. 40–50, 2002.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," Proc. of Hawaiian Int. Conf. on Systems Sci, pp. 2-10, 2000.