Dynamic Election-Based Sensing and Routing in Wireless Sensor Networks

Sharief M. A. Oteafy
School of Computing
Queen's University
oteafy@cs.queensu.ca

Hosam M. AboElFotoh
Dept. of Math. and Computer Sci.
Kuwait University
hosam@sci.kuniv.edu.kw

Hossam S. Hassanein
School of Computing
Queen's University
hossam@cs.queensu.ca

Abstract—**Decentralized protocols offer high adaptability to topology changes prominent in Wireless Sensor Networks (WSN). Protocols resilient to topology changes stemming from nodes dying, being added, relocating or duty cycling, improve network performance in terms of lifetime and percent of events sensed and reported. Topology dependant protocols, such as cluster-based, face many hindrances especially in terms of scalability, dynamicity, and adapting to varying traffic rates. Accordingly, a novel approach is introduced in sensing, whereby a single node is elected to report a sensed event, in a decentralized manner, thereby avoiding redundant reports by other nodes which exhaust network resources. Election is based on the node with the highest likelihood of successfully reporting the event. This protocol is coupled with a localized multi-hop routing protocol, to route that report back to the sink, by electing the most reliable next-hop neighbor to relay the report. Simulation results demonstrate the increase in network lifetime, detection/reporting efficiency, and resilience to varying node density.**

Keywords- Sensor Networks; Decentralized protocols; Dynamic Election; Decentralized Sensing; Dynamic Routing.

I. INTRODUCTION

The task of adapting Wireless Sensor Network (WSN) protocols to dynamic environments, ensuring high levels of resilience to network changes, is non-trivial. After deployment, Sensor Nodes (SNs) are susceptible to failure for many reasons; including running out of battery energy (death) or destruction by physical factors. In addition, nodes within communication range may fail to communicate due to temporary factors; such as a node being "off" in its duty cycle, or due to the onset of an event leading to signal attenuation beyond which the receiver cannot interpret the message sent to it.

WSNs are expected to self-configure upon deployment, and adapt their topology to perform their duties without external intervention. In case of changes in network structure/topology, resulting from duty cycled/dead/introduced nodes, the SNs are expected to dynamically adapt and reconfigure to carry on their assigned tasks. Thus adaptability of WSNs, which is highly coupled with network topology [12], is of utmost importance [1] - [3].

Protocols governing WSNs are mainly classified into two categories: centralized and decentralized [10] & [11], along with other hybrid architectures. In a centralized approach, either the sink or selected cluster heads are responsible for controlling routing, duty cycling and data aggregation from cluster members [3] & [2]. Cluster heads could be static throughout network lifetime, or could change after designated (time) rounds [3].

Centralized approaches suffer from degradation in performance mostly when the network architecture changes rapidly, or the number of *active* nodes vary over time. This includes short term variations due to duty cycling, or long term ones caused by a decrease from death or increase by introducing new nodes for replacement. A major factor of hindrance is bottlenecks. That is, as the tasks of the network increase (as from high event occurrence rates), the centralized controllers (sink and/or cluster-heads) become less capable of handling the amount of traffic generated. This affects network reliability, lifetime, power consumption, and overall efficiency in detecting the events, especially in time (reporting) latency.

On the other hand, decentralized protocols adapt to high traffic rates, increased event occurrence rates and other common hindrances in an efficient manner; by localizing their effect to their respective origins and hence sparing the network as a whole from performance degradation [6]. Issues such as duty cycling & data aggregation in addition to MAC protocols benefit by reducing the problem size to a local scale [12].

Many WSN operations could be optimized by planning (in deterministic deployment [8]) or detecting (in random deployment [4]) the number of sensing nodes in a given area. These operations include duty cycling and data aggregation to improve coverage, reliability and energy efficiency.

Most sensing schemes would have nodes report each event they sense when active. To conserve network resources, reducing redundancy in reports was explored by two main methods. The first exploits duty cycling, and ensures that a small percent of the nodes in the vicinity of the event would be active and hence detect and report the event. Duty cycling schemes vary in the methods by which nodes are selected for duty cycling, and the durations they spend active and idle [3].

The other main stream in reducing redundant reports depends on data aggregation. That is, numerous reports of the same event, traversing the network towards the sink, would be aggregated together (e.g. by averaging the readings) by nodes on their merging paths to the sink. This is done in a hierarchical scheme with the existence of local cluster heads, or in a flat scheme by nodes on the path to the sink [2]. In addition, other protocols, such as Directed Diffusion, are query based and assume that only nodes relevant to an event (in vicinity or other metrics) will respond; hence reducing reports.

In broader terms, detection and reporting redundancy result needlessly in significant power loss over the network. It is a major issue to address and hereby we propose a novel direction in eliminating redundancy in reports without jeopardizing reliability issues in terms of data fidelity and accuracy.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

The main contribution of this paper is establishing a decentralized dynamic sensing scheme which safely eliminates redundancy in reporting events, which exhausts network resources and hence affects its lifetime; especially as the rate of event occurrence increases. This scheme is then completed by introducing a dynamic election based routing scheme which routes the report effectively back to the sink.

Our proposed solution deals with an ad hoc WSN and makes no assumptions about the positions of nodes, duty cycling scheme, node density and its variation over the region and event occurrence rates. The presented scheme elects a single node for each event occurring to generate a report and sends it towards the sink, where a single node at each hop is chosen to relay the report. To ensure the reliability of the scheme in reporting each event back to the sink, a *fitness* measure is introduced to elect (in a decentralized manner) the most fit node to accurately generate or relay the report.

The remainder of this paper is structured as follows: The proposed sensing protocol is formalized and elaborated upon in Section II. Similarly, Section III presents the election-based routing protocol. Section IV details the simulation scenarios and results obtained from performance analysis, and Section V contains the conclusions of this work.

II. DECENTRALIZED ELECTION-BASED SENSING

One of the most prominent factors to consider in sensing is the density of nodes; given as the number of nodes in a given unit area [1] and [2]. It is vital to exploit redundancy in nodes to reduce the total energy spent to report an event, as well as increasing reliability in reporting it. Yet, doing so should follow a dynamic scheme which will not be hindered by high densities of nodes.

When a phenomenon event ***E*** occurs, all the *active* nodes in its vicinity, denoted by $\mathbf{v}_\mathrm{E}$**,** will detect it. Since some nodes in the network could be duty cycling and "*off*", the remaining nodes with their sensors "*on*" would be deemed *active*.

The goal of this sensing protocol is to elect the most fit node $\boldsymbol{n_f}$ in $\mathbf{v}_\mathrm{E}$ to generate a report of ***E*** and forward it towards the sink. This eliminates the redundancy of similar reports traversing the network and needlessly exploiting its resources, as shown in the remainder of this section. The symbols used hereafter are listed in Table I.

A. Assumptions

Define the WSN as a set of homogenous nodes ***N***. Each node has a unique ***ID***. For a node $\boldsymbol{i} \in \boldsymbol{N}$, its radius of sensing and transmission are denoted by $\mathbf{R_S}(\boldsymbol{i})$ and $\mathbf{R_{Tx}}(\boldsymbol{i})$ respectively, and its remaining battery energy as ***batt***(***i***). The sink's power supply is assumed to be unlimited, and it is placed on the border of the deployment region. No restrictive assumption is made on node placement, thus random deployment is assumed.

The *Euclidian* distance between two nodes ***i*** and ***j*** is given by **dist**(***i***, ***j***). For any two nodes ***i*** and ***j***, iff

$$\mathbf{dist}(\boldsymbol{i},\boldsymbol{j}) \le \mathbf{R_{Tx}}(\boldsymbol{i}) \tag{1}$$

then ***j*** is one-hop-away from ***i***. Starting with a hop value of zero for the sink, and increasing it by 1 for each one-hop-away neighbour, each node retains its minimum hop value relative to the sink, denoted as ***hop(i).***

TABLE I
NOTATION AND SYMBOLS

Symbol	Represents
$\boldsymbol{E}$	A phenomena Event detectable by sensing nodes
$\boldsymbol{N}$	Set of nodes in the deployed WSN.
$\boldsymbol{R_S(i)}$	Sensing range of node ***i***
$\boldsymbol{R_{Tx}(i)}$	Transmission range of node ***i***
$\boldsymbol{dist(i, j)}$	Euclidian distance between nodes ***i*** and ***j***
$\boldsymbol{hop(i)}$	Hop value of node ***i***
$\boldsymbol{batt(i)}$	Remaining battery energy in node ***i***
$\mathbf{v}_\mathrm{E}$	Set of *active* nodes in vicinity of ***E*** ($\subseteq N$)
$\boldsymbol{n_f}$	Most fit node which will report ***E*** ($\in \boldsymbol{v_E}$)
$\boldsymbol{\lambda}$	Mean inter-occurrence time of events
$\boldsymbol{\tau}$	Average inter-occurrence rate of events
$\boldsymbol{\xi_i}$	Fitness of node ***i*** to report an event
$\boldsymbol{\omega_i}$	Waiting duration of node ***i*** before declaring itself as winner
$\boldsymbol{\gamma}$	weight (importance) assigned to battery value w.r.t hop value
$\boldsymbol{\delta_E}$	Delay in reporting an event ***E*** by winner $\boldsymbol{n_f}$

Events occurring in the region being sensed are approximated according to a Poisson distribution with mean $\boldsymbol{\lambda}$, and the average inter-occurrence rate is denoted by $\boldsymbol{\tau}$. The locations of events over the region follow a uniform distribution.

B. Initialization

A setup phase takes place once the sensor nodes are deployed. We assume that all sensors initially have the same TX andRX range *R*. The target is setting the hop values of all the nodes in the network in a shortest-path method. That is, each node's hop value should be the least possible. Formally:

$$\forall i \;\; \nexists k \;\; \text{s.t.} \;\; hop(i) > k \tag{2}$$

where $\boldsymbol{i} \in \{1,\ldots,\boldsymbol{N}\}$, and ***k*** is the number of hops on a valid path reaching the sink.

This phase is triggered by the sink when it sends a **SetH**

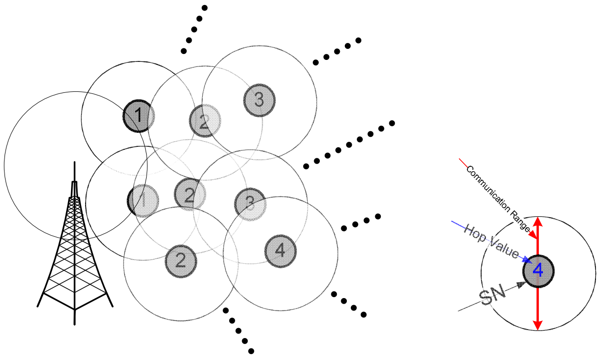

Figure 1: Setup phase: Setting hop values of nodes

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

message with its hop value of 0 to its one-hop-away neighbours. The sink's transmission distance is equal to *R*. Each node starts with a hop value of ∞. When node ***i*** receives a **SetH** message from node ***j***, its hop value is updated iff

$$(hop(i) - hop(j)) > 1 \tag{3}$$

in which case it *broadcasts* its new hop value in a new **SetH** message. The cycle then repeats for its one-hop-away neighbours. The result of this setup phase is depicted in Fig. 1.

Because of the update mechanism, and since the hop difference value is used to decide if a new **SetH** should be broadcasted, this algorithm will converge with each node setting its least possible hop value; assuming a finite number of nodes. This is presented in Algorithm 1.

Algorithm 1: Node ***i*** receiving a **setH** message

```
Procedure Receive_setH()
Receive setH message S
If hop(i) > S.hop +1 then
    hop(i) ← S.hop +1
    create new setH S' message
    S'.hop ← hop(i)
    Broadcast(S')
else
    ignore
```

C. Sensing protocol

The goal of this sensing scheme is to select one node, among the set $\mathbf{v}_E$ of active nodes able to detect an event ***E***, to report it back to the sink. The challenge is choosing the most reliable node, while sustaining efficiency and load balancing. Reliability here is a measure of the probability of the message reaching the sink if it was relayed through that node. This measure hereafter is referred to as the *fitness* of the node; denoted by ξ.

Fitness is an aggregation of the node's hop value and its remaining battery energy. These values are normalized prior to aggregation, by dividing them by their respective maximum values. Thus the aggregation function assigns weights to both normalized factors, allowing the protocol to effectively stress more importance on either.

Formally, the fitness of a node ξ_i is defined by:

$$\xi_i = \frac{\text{batt}(i)}{hop(i)} * \gamma \tag{4}$$

where $\gamma > 0$ represents the weight (importance) assigned to battery value w.r.t. the hop value.

For a time critical scenario, a smaller weight will be assigned to γ. This will orient the algorithm towards a shortest path approach. On the other hand focusing on the battery energy, by increasing γ, will favor nodes with larger energy reservoirs, hence distribute the load of the network over the nodes in a fair method.

After each node computes its fitness ξ_i it shall set a timer to a waiting duration ω_i, *inversely proportional* to its ξ_i, after which it will declare itself as the winning node. That is, the higher the fitness of the node, the shorter the waiting duration. Thus, the most *fit* node with the shortest ω_i – denoted by ω_f – shall declare itself first, and the rest of the nodes competing to report that event will quit. Formally:

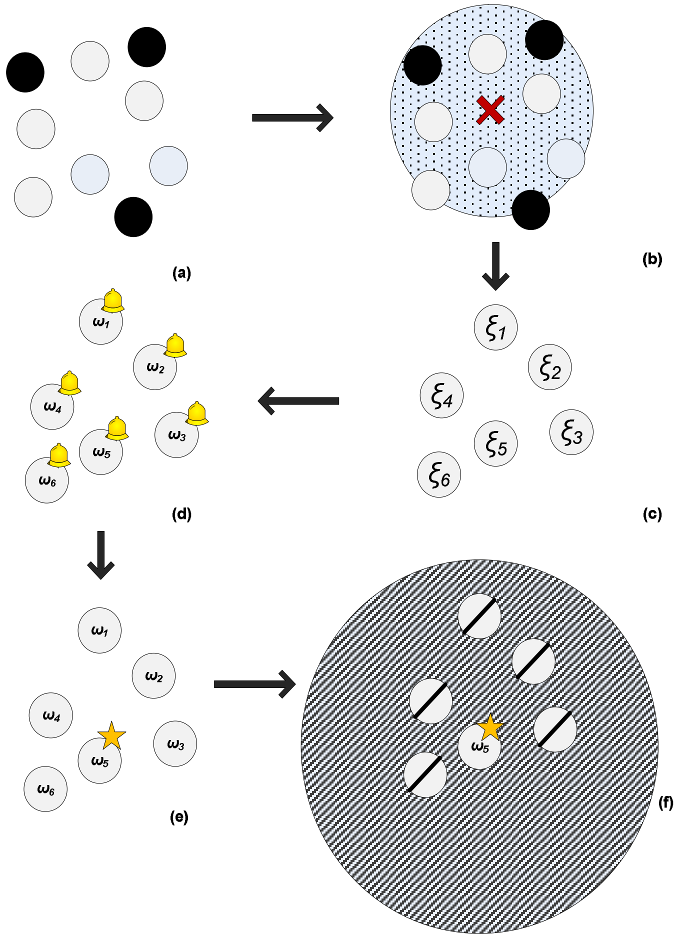

Figure 2: Sensing scheme. In (a) the nodes are in their initial states, where the black ones are not active. When an event ***E*** occurs in (b), marked by an X, the active nodes in its vicinity v_E – within the circle – set their willingness values ξ_i as in (c). Each of those nodes then sets its waiting timer ω_i to start competing for reporting ***E***, shown in (d). The node with shortest wait time wins – marked by * in (e) – then it broadcasts a message to forward the report at **twice** the sensing range $R_S(i)$, declaring itself the most fit n_f node; as in (f).

$$\omega_i \propto \frac{1}{\xi_i'} \tag{5}$$

where ξ'_i is the normalized value of ξ_i after being spread over the [0,1) range, defined as:

$$\xi_i' = \frac{\xi_i}{\sqrt{\xi_i^{\,2} + s}} \tag{6}$$

where ***s*** is a spreading factor for the resulting normalized value. The larger the value of ***s*** the more spread the value of ξ'_i would be over that range.

By normalizing the values composing the fitness measure, an upper bound is placed on the delay in reporting the event (to next neighbor towards the sink). Formally, the delay in reporting an event E, denoted by δ_E:

$$\delta_E \le \left(\omega_f + \hat{N}_Q\right) \tag{7}$$

where ω_f is the waiting time of the most fit node (winner) n_f and $\check{N}_Q$ is the average delay in the queue of n_f before transmission (depends on MAC protocol) .

Finally, the winning node n_f will transmit its request, "Who Is Ready" message (**WIR**) to forward the report towards the sink, but at a transmission range equal to twice the sensing range. This will ensure that all possible nodes in $\mathbf{v}_E$ who were competing to report, realizing that another node is more fit, will quit the contention upon hearing that request.

The dynamics of this sensing protocol is depicted in Fig. 2. Also, Algorithm 2 presents the pseudocode of the operations performed by an active node when it detects an event ***E***.

Algorithm 2: Active node n_i sensing an event ***E***

```
Procedure compete_to_report()
compute $\xi_i$
$\omega_i \leftarrow \xi_i + \Delta_i$
set waiting timer to $\omega_i$
while ($\omega_i > 0$) do
    if WIR message heard then
        exit compete_to_report()
    else
        pause(1 time unit)
        decrement $\omega_i$
end while
create report message of this event
transmit report message at 2* $R_S(i)$
```

D. Avoiding Multiple Winners

When the virtual set of active nodes $\mathbf{v}_E$ competing to report event ***E*** forms, it is quite possible for more than one node in $\mathbf{v}_E$ to have the same fitness value. In such case, if in (5)the waiting timer ω_i simply relied on the inverse of fitness; those nodes with equal values will declare themselves as unique winners even though they are not unique. This could cause many reliability issues and redundancy in reports, which contradicts the purpose of this algorithm.

To solve this problem, a randomness factor has to be included in computing ω_i. Using a pseudorandom generator with a sufficiently large period, and knowing that the cardinality of $\mathbf{v}_E$ is limited, this would suffice to satisfy with high certainty that the resulting value for the winning node ω_f is unique. Hence, there will only be one winner to report each event.

In addition, nearly all MAC protocols used in WSNs employ algorithms to avoid/prohibit collisions [3] & [9], and they all basically give preference to one transmission over the other in the time domain. Even though our proposed protocol does not rely on that aspect of MAC protocols, but it serves in emphasizing the negligibility of the probability of more than one node declaring themselves as a winner at the same time.

To introduce randomness in calculating ω_i, a Linear Congruential Generator (LCG) [8] is used in calculating a random offset in time, denoted as Δ. Each node ***i*** can set its time offset Δ_i accordingly. Thus, with a sufficiently large value for the LCG's period, the waiting time of each node in $\mathbf{v}_E$ is computed as:

$$\omega_i = \frac{1}{\xi'_i} + \Delta_i \tag{8}$$

III. DECENTRALIZED ELECTION-BASED ROUTING

In routing protocols, tailored parameters are derived to optimize their performance according to each targeted application. Mainstream protocols however target energy efficiency; mainly to prolong network lifetime.

Here we present a dynamic routing protocol, based on the election scheme previously discussed, to select at each stage the most reliable node to relay a message in a multi-hop scheme. The protocol facilitates dynamic switch of emphasis on either shortest path routing or load balancing routing. This section presents the assumptions made, scheme of the routing protocol, and an example of how it works.

A. Model Overview

In essence, we follow the same general model presented in Section 2.1. Once node $\mathbf{n}_f$ declares itself as the winner for reporting an event ***E***, it broadcasts a message declaring itself as the winner, hence signalling to all the other nodes in $\mathbf{v}_E$ to halt their competition. This is where the sensing protocol transitions to the routing protocols by triggering a multi-hop scheme to route the message back to the sink.

Since not all the nodes on the physical path to the sink would be active, the protocol will have no assumptions regarding the density nor distance between neighboring nodes. Instead, it will target the most *fit* node on a route to the sink (i.e. with a smaller hop value). We assume that nodes have the capability to increase their transmission range as needed, within the boundaries of their remaining power.

Hence, the inherent selection of the most *fit* node to carry on the task of relaying a message implies that it would be the node with the larger energy reservoir making it capable of increasing R_{Tx} as needed to ensure the delivery of the message at hand. Then the receiver will initiate the same cycle to find the most fit node on the next hop towards the sink, until the message finally reaches the sink.

B. Routing Protocol

When node n_i has a message to relay towards the sink, it will broadcast a WIR message at its initial transmission range. This message will be heard by all the neighboring nodes one-hop-away from the n_i, and hence will create a virtual set of nodes $\mathbf{v}_E$. The next hop node will be chosen from $\mathbf{v}_E$.

The target would be to select the most *fit* node $n_f \in \mathbf{v}_E$, which will follow the same scheme explained in Section II.C, yet now the nodes in $\mathbf{v}_E$ are contending to relay a message instead of reporting an event. Node n_f would then reply back with an **IR** message (**I**'m **R**eady) declaring itself as the most fit. The IR message would be transmitted at twice the transmission range at which the triggering WIR message was sent at (which would be integrated into the WIR message). This suffices to make all contending nodes in $\mathbf{v}_E$ quit competing.

If after a given time out period, n_i does not receive a message, then it would assume that there are no active nodes in its vicinity. In this case, n_i would increase its transmission range by a constant increment Δ_d given that it has enough battery reservoir to do so, and then resend the WIR at this new range. If it cannot afford the power expenditure resulting from such an increase in transmission range, then the packet is dropped. This cycle would keep recurring until either a winning node n_f would reply with an IR message, hence successfully completing its duty, or the relaying node n_i could no longer increase its transmission distance and thus dropping the report.

Algorithm 3: Active node n_i transmitting a **WIR** message

```
Procedure find_next_hop()
timer ← 0
R_Tx ← min_Tx_range
Transmit WIR at R_Tx
while message not sent
    while (timer < time_out_thrshld) do
        if IR message received from node n_j
            forward report packet to n_j
            exit find_next_hop()
        else
            pause(1 time unit)
            Increment timer
    end while
    If remaining energy suffices
        R_Tx ← R_Tx + Δ_d
        timer ← 0
        Transmit WIR at R_Tx
    else
        drop_packet()
end while
```

Algorithm 2: Active node n_i receiving **IR** message from n_j

```
Procedure compete_to_relay()
compute $\xi_i$
$\omega_i \leftarrow \xi_i + \Delta_i$
set waiting timer to $\omega_i$
while ($\omega_i$ > 0) do
    if IR message heard then
        exit compete_to_relay()
    else
    pause(1 time unit)
    decrement $\omega_i$
end while
create IR message
transmit IR message at $R_S(j)$
```

IV. PERFORMANCE ANALYSIS

To analyze the effect of the proposed sensing and routing schemes, the following model has been adopted and standardized over all the simulation scenarios. Our analysis is based on the simulation results generated by the Event-Driven Wireless Network Simulator (EDWiNS), detailed in [10], to test the different protocols in comparison schemes.

The model assumes a deployment over a 100 x 100 m area with 200 nodes positioned randomly over that region, and the sink resides in a corner of the region. Each node has $\mathbf{R}_{Tx}(i)$= $\mathbf{R}_S(i)$=20 m, and starts off with 100 J of energy. The mean inter-occurrence time of events follows a Poisson distribution with mean τ. To accurately measure and compare the performance of the different protocols, we use the same set of events for all simulations. Data messages are 1 kB in size, and control messages are 200 bits.

The Friss-free space model is used for our power-model functions, defined as:

$$E_{T_x}(k,d) = k * \left(E_{elec} + d^{\theta} * E_{amp}\right) \tag{9}$$

$$E_{R_x}(k) = k * E_{elec} \tag{10}$$

estimating the energy required for transmitting, E_{Tx}, a data packet of size k bits to a distance d meters and the energy spent receiving it; E_{Rx}. This model is adopted from [7], where E_{Elec} is the energy dissipated by the transceiver circuitry, either in transmitting or receiving, and E_{amp} is the transmitter amplifier power. The path lost exponent θ is 2. In the power model in [7], the cross-over distance set as the cut-off point to decide on θ was determined to be 86.2 m. Thus, the d^2 attenuation model is a valid approximation; since the distance between nodes is seldom greater than this in most WSNs.

Network lifetime performance of our protocol was compared to both the direct-transmission and the multi-hop shortest path protocols, using the IEEE 802.11 CSMA/CA MAC protocols with RTS/CTS. This is shown in Figure 3.

Evidently network lifetime is significantly reduced by resorting to direct transmission versus multi-hops schemes, yet the gain in lifetime achieved by our protocol, by reducing redundant reports, is evident in the figures, especially as message size increases (5 Kb in our experiments) .

On an equally important measure, the percentage of events successfully detected and reported by the network account for a crucial caliber to the performance of WSN protocols. Figure 4 demonstrates that measure for the three schemes. It is important to note two main points. First, the dynamic election based sensing scheme did not suffer any significant degradation in its ability to successfully detect and report events, even when compared to a scheme where redundancy is heavily exploited to ensure a high percentage.

Secondly, as the rate of event occurrence decreases (its value increases), the network suffers energy loss due to idle waiting time more than the power needed to report such events. Nevertheless, our protocol suffers the least from this variance.

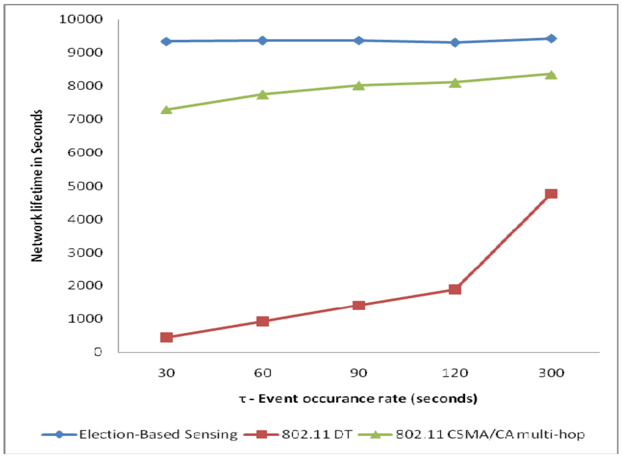

Figure 3: Effect of event occurance rate τ on network lifetime.

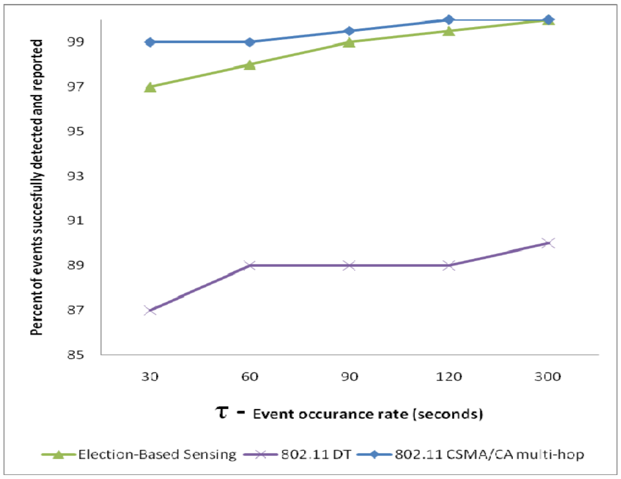

Figure 4: Effect of event occurance rate on the percent of events succescully decteted and reportad back to sink.

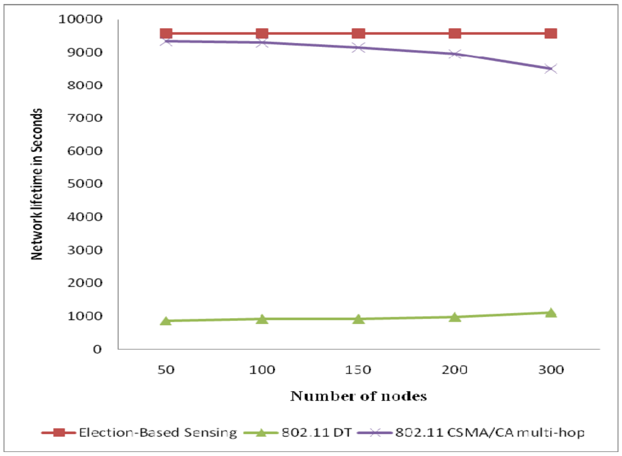

Figure 5: Effect of network density on network lifetime.

Finally, node density, being one of the common parameters exploited to ensure reliability in detection and increase in network lifetime, is investigated. Node density is defined as the number of nodes in a given unit area. In our simulations the sensing region area was fixed to 100x100 m. Increased node density is usually justified for achieving higher coverage. Nevertheless, if a protocol does not take into account the overhead of coordination between nodes, loss of energy caused by redundant messages, and the effect on duty cycling, then increased node density would be a hindrance to performance.

The effect of node density on network lifetime is depicted in Figure 5. Our protocol is resilient to changes in node density, since for any given event only one node would report it, whereas the other protocols suffer due to the depletion of network energy caused by the increase in number of messages to report and relay. This also incurs significant reporting delay, since all nodes will contend to access the medium.

V. CONCLUSIONS

We proposed a novel sensing protocol that would alleviate the degrading effect of varying node density, network exhaustion due to redundant reports, and prolonged reporting time due to heavy contention to access the medium while increasing network lifetime and not sacrificing detection and reporting efficiency. Our protocol stems from a decentralized dynamic architecture which is indifferent to the underlying topology, duty cycling, and adapts to varying (short and long term) node densities and availability. Our sensing scheme was coupled with a routing protocol which dynamically routes reports back to the sink by emphasizing either time latency or efficient load balancing using an aggregation function which adapts to different WSN scenarios as required.

REFERENCES

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

[2] 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.

[3] A. Boukerche, and S. Nikoletseas, “Protocols for data propagation in wireless sensor networks: a survey,” *Wireless Communications Systems and Networks.* 718 – 732.

[4] M. Cardei, and J. Wu. *Coverage in Wireless Sensor Networks.* Handbook of Sensor Networks. CRC Press, 2004.

[5] 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.

[6] H. Hassanein, J. Luo, "Reliable Energy Aware Routing In Wireless Sensor Networks," *2nd IEEE Workshop on Dependability and Security in Sensor Networks and Systems DSSNS*, April 2006.

[7] 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.

[8] H. Leeb, and S. Wegenkittl, “Inversive and linear congruential pseudorandom number generators in empirical tests,” *ACM Transactions on Modeling and Computer Simulation (TOMACS)* Vol. 7, No. 2, pp. 272-286, Apr. 1997.

[9] D. Niculescu, “Communication Paradigms for Sensor Networks”, IEEE Communications Magazine, March 2005.

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

[11] S. Sahni and X. Xu , “Algorithms for wireless sensor networks,” *International Journal on Distributed Sensor Networks*, vol.1, no. 1, pp.35-56, 2005.

[12] 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.