A COLLISION FREE MAC LAYER PROTOCOL FOR EVENT BASED DATA GATHERING IN WIRELESS SENSOR NETWORKS

Shafaq B. Chaudhry, Ratan Guha, Seema Bandyopadhyay
School of Computer Science
University of Central Florida
Orlando, FL 32826
Email: {shafaq, guha, seema}@cs.ucf.edu
Phone: 407-823-2956, Fax: 407-823-5419

Yi Qian
Department of Electrical and Computer Engineering
University of Puerto Rico at Mayaguez
Mayaguez, PR 00681
Email: yqian@ece.uprm.edu
Phone: 787-458-4901, Fax: 787-833-3331

Keywords – wireless sensor network, MAC protocol, data gathering, reactive networks, clustering.

ABSTRACT

In this paper, we have proposed a collision-free medium access control (MAC) protocol for wide area data gathering applications in wireless sensor networks. The proposed MAC scheme is specifically designed for reactive networks and follows an event-based data-gathering paradigm that takes into consideration the spatial and temporal correlations of the process being monitored. Sensors periodically monitor their surroundings and react to the environment either by waking up if an abnormality is detected or by decreasing their sleep cycle if an abnormal trend is detected. Simulation studies show that the proposed scheme performs better than similar existing schemes in terms of network lifetime and the responsiveness of the network to the events of interest.

INTRODUCTION

Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life, from tasks such as surveillance and security, to widespread environmental monitoring, manufacturing and business asset management, and automation in the transportation and health-care industries, etc. They can be used in virtually any environment, even where wired connections are not possible, and the terrain inhospitable or physical placement difficult. Wireless sensor networks consist of battery-operated sensor devices with computing, data processing, and communicating components. Energy conservation is a critical issue in wireless sensor networks since batteries are the only limited life energy source to power the sensor nodes.

Like in all shared-medium networks, medium access control (MAC) is important for the successful operation of sensor networks. The MAC protocol in a wireless sensor network must achieve two goals. The first objective is the creation of a network infrastructure which is achieved by establishing communication links between thousands of sensor nodes that are scattered in a sensor field. The second aim is to share communication resources between sensor nodes fairly and efficiently.

Constraints on energy resources available in sensor networks have necessitated the design of MAC protocols that are energy-efficient. Several MAC protocols for wireless sensor networks with an ultimate goal of increasing the network lifetime by conserving energy have been proposed in the literature. S-MAC (Ye et al. 2004) and T-MAC (Dam and Langendoen 2003) are two such energy-efficient MAC protocols that save energy by introducing sleep cycles in sensors. DMAC (Lu et al. 2004) addresses the issue of handling delay while periodically reporting data to the base station. All these protocols were developed for proactive* networks. In (Begum S. et. al., 2004), the authors proposed a scheme, called ELECTION, for reactive* networks. The ELECTION scheme saves energy by introducing sleep cycles for both the sensing and communication circuitry of the sensors. When an event occurs that requires immediate attention, nodes switch on their communication radios, form clusters, aggregate data and report to the base station. It is assumed that all nodes within a cluster wakeup at the

---

* In Reactive Networks, sensors wakeup immediately to sudden and drastic changes in the sensed attribute. In Proactive Networks, sensors periodically sense the environment and transmit data of interest. (Mangeshwar and Agrawal 2001)

same time because of the spatial and temporal correlation of the phenomenon being monitored.

Our goal in this paper is to develop a delay-sensitive MAC protocol for a reactive sensor network deployed for monitoring of a wide area. In our scheme, sensors are organized into clusters after deployment and then they enter a periodic sleep and monitor phase. The sleep duration is adaptive; it reduces when the sensors sense drastic changes in the environment. When an event of interest occurs, the sensors around the location of the event sense it and communicate their readings to the cluster head, which in turn communicates the aggregated data to the user/base station. Since the clustering infrastructure is already in place, unlike ELECTION there is no unnecessary delay in reporting the event to the base station. Also, there is no wastage of energy for reforming the clusters every time an event occurs – another improvement upon ELECTION. The energy savings in our scheme becomes significant when multiple clusters are involved. The proposed scheme also handles a naturally occurring phenomenon that changes with time and diffuses through the field, more efficiently and naturally than ELECTION. We also propose a simple load-balancing scheme to prolong the network lifetime. This load-balancing scheme takes into consideration the fact that the cluster heads drain more energy than the non-cluster head nodes, and hence proposes that a set of candidate cluster-heads (instead of just one) be chosen at the time of cluster formation. Each candidate node in this set takes turns to be the cluster-head for the cluster, and thus useful load balancing can be achieved that prolongs the network lifetime.

The main contributions of this paper are:

- Design of a clustering algorithm for sensor nodes deployed to monitor a wide area application with a simple and efficient load-balancing scheme.
- Design of a collision-free, energy-efficient MAC protocol for successful communication within the clusters.
- Comparison of the proposed scheme's performance in terms of energy consumption and delay with ELECTION.

In the remainder of the paper, we discuss the advantages and disadvantages of some relevant protocols in the Related Work section. Then we give a detailed description the proposed MAC protocol, followed by simulation results, analysis and discussion. Finally, we draw our conclusions and discuss possible future work.

RELATED WORK

Over the past decade, many MAC protocols for sensor networks have been proposed that promise energy-efficiency. S-MAC (Ye et al. 2004) proposes periodic sleep-listen schedules to reduce energy. Neighboring nodes synchronize together to form a virtual cluster and set up a common sleep schedule. Collision is avoided by using the RTS/CTS and the overhearing problem is avoided by means of a Network Allocation Vector. Since different neighbors may have different schedules, S-MAC suffers from "sleep delay" which increases latency in multi-hop routing algorithms. T-MAC (Dam and Langendoen, 2003) improves upon the S-MAC protocol to achieve better latency and better throughput under variable traffic load. T-MAC reduces idle listening by transmitting all messages in bursts at the start of the frame and sleeping between bursts. The length of the burst is determined dynamically. For reducing the delay in unidirectional data gathering sensor network application, DMAC (Lu et al. 2004) proposed a "staggered wakeup" scheme which assigns subsequent active slots to subsequent nodes in the data transmission path. Data prediction is used to enable active slot request when multiple siblings have packets to send in the same sending slot. When two nodes at the same level but different parents wish to transmit in the sending slot, an explicit More-to-Send control packet is used. DMAC does not avoid collisions, and so it wastes energy because of re-transmissions.

LEACH (Heinzelman et al.) is an energy-efficient communication scheme that employs periodic cluster formation. It assumes a single-hop sensor network in which each sensor can communicate with any other sensor or with the base station directly. Each node makes the decision of becoming the cluster head independent of its neighbours.

SMAC, TMAC, DMAC and LEACH all deal with proactive networks that require periodic monitoring and data collection/reporting. Secondly, they do not assume the sensor to be independent of the radio and never shut off their sensor.

For reactive networks, the routing protocol TEEN (Mangeshwar and Agrawal 2001) considered a MAC model based on periodic sleep and wakeu. Nodes form clusters at fixed intervals similar to LEACH. During sleep, the communication radios are switched off but the sensors monitor the environment. When an event exceeding a given threshold is detected, nodse switch on their radios and start reporting to the base station.

ELECTION (Begum et al. 2004) is an energy-efficient MAC protocol designed for gathering data from the network as and when special events are detected. ELECTION adaptively schedules the sleep cycles of both the radio and the sensor in the sensor node based on samples taken from the environment. Thus, it achieves energy-efficiency, and ensures low latency and high responsiveness when abnormalities occur in the phenomenon being monitored. Each node adjusts its sleep cycle independently. The sensors turn their radios on when the sensed intensity of the event has exceeded a pre-defined threshold, form single-hop clusters and communicate their readings to their respective cluster-heads using a TDMA schedule. The disadvantage of the single-hop ELECTION is that it assumes that all nodes that become active are within the transmission range of each other. Since ELECTION forms clusters only when abnormalities are detected, problems with synchronization can arise when the nodes are left sleeping for a long time. Moreover, ELECTION is not suitable for applications where a priori information is not available and thresholds cannot be set or the underlying phenomenon does not exhibit spatial-temporal correlation.

Bandyopadhyay and Coyle (2003) devised a distributed algorithm for organizing sensors into a hierarchy of clusters taking into consideration the energy constraints of wireless sensors. They showed that minimum communication energy is achieved in a network with clustering hierarchy rather than a non-clustered network, and computed the optimal number of cluster heads at each level that will minimize the total energy spent by the sensors while reporting to the base station.

THE PROPOSED MAC LAYER PROTOCOL

In this paper, we propose a collision-free MAC protocol for wide area data gathering applications in wireless sensor networks. The proposed MAC scheme is specifically designed for reactive networks and follows an event-based data-gathering paradigm that takes into consideration the spatial and temporal correlations of the process being monitored. The following assumptions have been made:

i. The sensors are uniformly distributed with density ρ covering a certain area A.

ii. All sensors are homogeneous, static and bear the typical constraints of a wireless sensor network.

iii. The fixed base station is far from the area A, so each sensor has to expend the same amount of energy to communicate with the base station.

iv. The sensor and radio of each sensor can be switched off independent of each other.

v. The event under study exhibits spatial and temporal correlation and it is possible to define a data threshold, a gradient threshold and a tolerance threshold for the event.

Modeling an Event in the Network

All the nodes initially sense the normal intensity. An event *e* can occur at any location $P_{ev}(x,y)$ within the area A at any time t_{ev}. With the passage of time, the event increases in intensity by a certain factor f and also “propagates” from its location source. The intensity I at a point $P(x,y)$ at a time t is a function of the distance from the event source and the time elapsed since the generation of the event.

$$I(P,t) = f\left(dist\left((P, P_{ev}\right), duration\left(t, t_{ev}\right)\right)$$

So, different nodes will sense different values of the event intensity at different times depending upon their distance from the event location. However, the intensity measured at points equidistant from the source will be the same at a given time. If we were to draw concentric circles around the event source, the value of the intensity in each ring will be the same at a given time. Since the event tends to “diffuse”, therefore, at sufficiently large distances from P_{ev}, and when sufficient time has elapsed, no effect of the event may be felt at point $P(x,y)$. Consequently, nodes that lie in the same concentric circle as $P(x,y)$ and beyond will sense the “normal” intensity.

Details of the Proposed Scheme

The scheme has three phases. The details of these phases are given below.

Phase 0 – Synchronization and Cluster Formation

Initially, after the nodes have been deployed, nodes organize themselves into clusters according to a clustering algorithm that is similar to the algorithm proposed in (Bandyopadhyay and Coyle, 2003). Each node becomes a ‘volunteer’ cluster head with a probability *p*. The “volunteer” cluster head advertises this fact to neighboring nodes. Non-cluster head nodes join the cluster head that is closest to them based on the radio transmission range. If a node does not receive a cluster head advertisement within time t, it knows that it is not 1-hop reachable by any of the “volunteer” cluster heads, and so becomes a “forced” cluster heads. Within each cluster, a set of nodes is chosen to be included in the list of candidate cluster heads. These nodes are situated close enough to the original “volunteer” head so that all the cluster members are reachable from each of the candidate cluster head nodes. CSMA is used for all messages sent in the cluster formation phase. Each cluster head then formulates a TDMA schedule for the nodes in the cluster by assigning a number to each member node that determines the TDMA slot for that node. At the end of Phase 0, nodes synchronize with each other. The synchronization can be achieved using the Reference Broadcast Synchronization i.e. RBS (Elson et al. 2002), and other such synchronization protocols. Then, the nodes enter into Phase 1.

Phase 1 – Periodic Sleep and Monitor

In this phase, nodes follow a periodic sleep and wakeup cycle. When a node goes to sleep, it shuts off both its radio and its sensor. Periodically, the node switches on its sensor to sense the environment and take readings. If the perceived data *d* exceeds or equals the pre-defined data threshold D_{th}, the node transits into the active state and gets ready for the data-reporting phase. If the current reading is drastically different from the reading taken in the previous wake up cycle i.e. the gradient $g(t)$ exceeds the value of the gradient threshold G_{th}, but *d* lies within the data threshold D_{th}, the node reduces its sleep cycle. We have adopted the geared sleep reduction function proposed in ELECTION. According to this function, the sleep cycle for time $(t+1)$ depends on the gradient $g(t)$ and the previous sleep cycle $s(t)$ and is calculated as follows:

$$s(t+1) = \begin{cases} s(t) & g(t) < 0.0 \\ \frac{1}{2}s(t) & 0.0 < g(t) < 0.005 \\ \frac{1}{4}s(t) & 0.005 < g(t) < 0.01 \\ \frac{1}{8}s(t) & 0.01 < g(t) < 0.02 \\ \dots \end{cases}$$

Phase 2 – The Active State

In the active state, both the radio and sensor of the node is switched on. In this phase, nodes send information to their respective cluster heads by following their TDMA schedules. If a scheduled node is asleep, the slot is wasted. After the cluster head has received data from all the cluster members that are awake, the cluster head processes this data and communicates it to the base station. If there are no members in the cluster, the cluster head just sends the data to the base station. Aggregated data is sent to the base station only if it is different from the previous reading sent to the base station. Note that according to assumption (v), the events exhibit spatial and temporal correlation and hence if a sensor is sensing the event, in all probabilities its cluster head is also sensing it and hence is alive. In rare occasions, when the cluster head is not alive, the sensors in its cluster will wait for it to sense the event and come alive before transmitting their data to it. In the active phase, when the data sensed by a node falls below the data threshold, the node goes back to Phase 1. We suggest that each cluster be assigned a unique channel for communication. This allows a node in a cluster to transmit at the same time as a node in another cluster without fear of interference. Thus, if an event or phenomenon affects many clusters, or if different events affect different clusters, each cluster can communicate with the base station independent of the other clusters in a collision-free manner.

The Proposed Load Balancing Scheme

As the cluster head suffers considerable energy drainage, the responsibility of the cluster head is re-delegated after it has reported to the base station. This re-delegation occurs in a round-robin fashion to another node that is a member of the set of candidate clusters. The TDMA schedule is also re-adjusted to reflect this change. This load-balancing scheme prolongs the network lifetime, which is especially apparent when the density of the network was increased during the simulation.

Comparison with Election

In this section, we would like to highlight the major differences between our scheme and ELECTION.

In ELECTION, the cluster head is chosen to be the node that reads the highest value of the sensed event. Every time an event is detected, clusters are re-formed, cluster-heads are re-elected and the TDMA schedule of the cluster members is re-created. Then data aggregation and reporting of the event takes place. In the scheme proposed in this paper, the inefficiencies that arise from dynamic formation of clusters are mitigated by forming the clusters statically.

Further, ELECTION raises some interesting questions in scenarios of more realistic events: After the first node wakes up, it is assumed that after the duration t_d of one sleep cycle at worst, sufficient nodes will be awake for the cluster formation and data aggregation/report to occur. What happens after t_d, when surrounding nodes start to wake up as the event propagates? Do the newly awake nodes form clusters among themselves or do they try and join the existing closer clusters? If new nodes join existing cluster heads, the cluster-heads will have to re-generate the TDMA schedule. Dynamic TDMA re-scheduling is not an easy task. The ELECTION scheme does not address these issues clearly.

This problem does not arise in our scheme because the nodes, as and when they wake up, do not have to make any decision of joining any cluster-head. They already know which cluster-head they should report to. Therefore, the cluster heads start to aggregate data immediately and report to the base station. Thus, our scheme differs from ELECTION since cluster formation occurs initially at the time of network deployment and the cluster heads are chosen in a probabilistic manner. The clusters are not dynamically re-formed at the onset of every event and the TDMA schedule is not re-formulated.

SIMULATION AND RESULTS

Simulation Setup

We have implemented both the proposed scheme and the ELECTION protocol in C++. We consider a sensor network of 1000 sensors uniformly distributed over a 100 x 100 sq. units area. The radio range of the sensors was set to a radius of 5 units. Each node becomes a cluster head with a probability of 0.1 We assume that the nodes take some amount of time t_s to synchronize based on the synchronization scheme used (e.g. RBS etc.). The initial sleep cycle time for the nodes was set to 256 time units.

We assume that the ‘base station’ is a special receiving station, mounted on towers and located within the 100 x 100 sq. units. We assume that the case when a node has data to send and its cluster head is asleep rarely occurs. This is because of the spatial and temporal correlation of the underlying phenomenon. However, if this does happen, the node just waits until the cluster head wakes up which will eventually happen as the event propagates.

Event source was generated at a random location $P_{ev}(x,y)$ in the 100 x 100 sq. units region. The event that we are trying to sense is temperature increase. Since heat diffuses from this point in all directions with a certain rate, we set the intensity of the process to increase logarithmically with time. Consequently, at a certain point in time, a sensor node that is within the radius of the event P_{ev} may or may not be awake depending upon its location from the point P_{ev}, given the rate of diffusion of the event. We set initial temperature to be “normal” at 80 degrees and D_{th} at 90 degrees. We set the gradient threshold G_{th} to 5 and the tolerance threshold T_{th} to 2 to allow more nodes to be in the active state at a given time.

Each node starts off with a store of 5000 energy units. We assume that each sensor uses 1 unit to sense the environment and 10 units to send data to the cluster head. Each cluster head uses up 5 units of energy to receive data from a member, and 50 units to send to the base station. All control messages like sending the cluster head advertisement and cluster membership, consume 1 energy unit. We assume no message queuing, hence no queuing delay. We assume a constant processing delay t_p for cluster head before it sends to the base station.

We further assume that if there are n sensors in a cluster, then the cluster formation will take *O(n)* time units. To form the cluster each sensor has to send at least one message in the ELECTION scheme using CSMA, hence the above assumption is obviously an underestimate of the time taken to form clusters. We also assume that the time taken to form the TDMA schedule in the cluster is 1 time unit and the energy spent in forming the schedule is negligible.

Results and Discussion

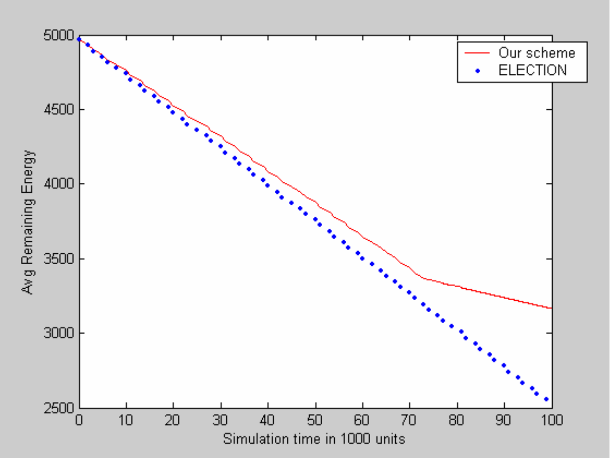
To have a fair comparison, we assumed a simple event; we assumed that the event will affect all the nodes under study and hence all the nodes in the network sense the same intensity of the phenomenon. This assumption was made because it is not known how ELECTION would handle event propagation through the network. Multiple clusters were formed within each scheme. We assumed that each cluster will have a channel allocated to it for communication and hence there is no inter-cluster interference. No load balancing technique was employed for either of the schemes; we only considered clusters with a single cluster head. Starting from an initial intensity of 80, we generated an event repeatedly by letting it increase beyond the threshold of 90 and then decreasing the event back to normal and reiterating over the whole cycle. This kind of an event caused the nodes to wakeup, report to the base station and go back to sleep periodically. Figure 1 shows the comparison of average energy remaining for both the schemes.

The average remaining energy here is defined as the total remaining energy of all the nodes over the total nodes in the network. Figure 1 shows that our scheme consistently performs better than ELECTION. This saving increases as time progresses because the event is being frequently generated. In ELECTION, each time an event is detected energy will be spent in forming a cluster, before aggregating and reporting to the base station. If a huge network is considered, like in this simulation, multiple clusters will be formed, and energy will be spent for every cluster. This energy will include the messages sent to discover neighbors, the processing to find the neighbor node with the highest sensed value, sending the cluster head advertisement and cluster membership messages, and TDMA schedule formulation. CSMA is used for all messages sent to and fro in the cluster formation phase. If we assume that a single control message sent or received consumes 1 unit of energy, then in the best case scenario of CSMA when there are no re-transmissions and all messages are successfully sent and received, the cluster formation will consume

$$N_n \times (1 + O(m)) + N_c \times (1 + O(n)) + N_c \times O(n)$$

units of energy where N_n is the number of nodes, N_c is the number of clusters, m is the maximum number of neighbors of a node in the network, and n is the maximum size of a cluster. On the contrary, in our proposed scheme, the energy wasted while forming and re-forming the clusters is obviated. Cluster formation is a one-time procedure that takes place in phase 0, and thus energy is only wasted once.

Figure 1. Comparison of average remaining energy for ELECTION and our proposed scheme. N=1000, CH=1, event=unnatural, A=100x100 sq. units, D_{th}=90, D_{in}=80



In Figure 1, we see an abrupt change in the slope of our scheme around 70 thousand time units. The reason for this is that, around this time, cluster heads start to die out. Since they no longer report to the base station, the nodes that belong to these clusters ‘conserve’ their energy because they cannot send data to a dead cluster head. On the other hand, the curve for ELECTION proceeds with the same slope because clusters are being formed dynamically on demand. However, when we implement our scheme with the load balancing feature, this problem is solved, because cluster-heads are dynamically being switched and clusters do not suddenly become inactive. Cost of switching the cluster heads in our scheme will be much less as compared to re-forming all the clusters in ELECTION. In fact, cost of switching the cluster-head entails one control message sent by the current cluster head to the next-in-line cluster-head and one cluster head advertisement message sent by the newly registered cluster head to all the members of the cluster.

Figure 2. Comparison of delay for ELECTION and our proposed scheme. N=1000, CH=1, event=unnatural, A=100x100 sq. units, D_{th}=90, D_{in}=80

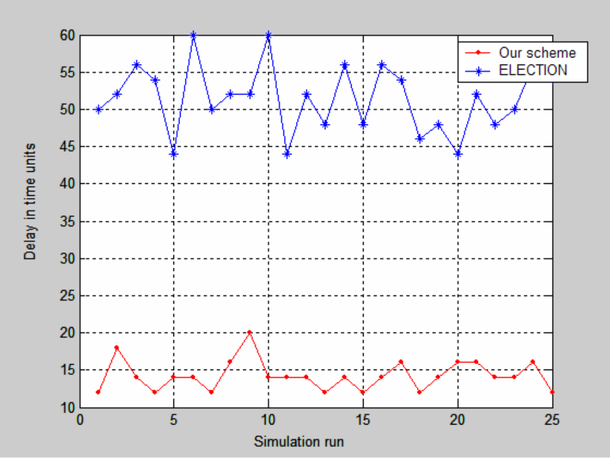
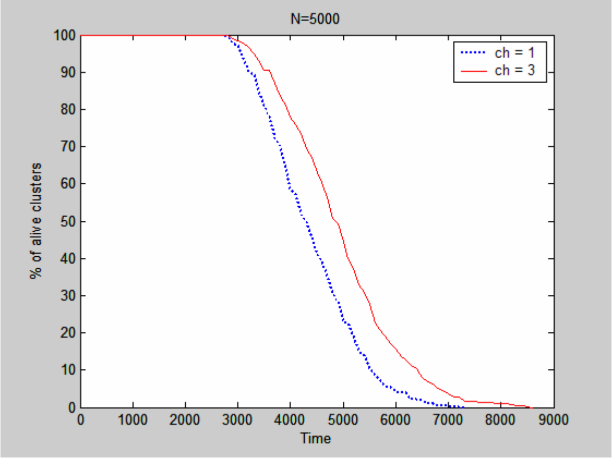


Figure 2 shows the delay comparison of our scheme with ELECTION where delay is defined as the amount of time between the first detection of an event to the time the data actually reaches the base station. The increased delay encountered in ELECTION is due to the fact that clusters are reformed every time the nodes wake up. . Another implication of this is that the reading reported to the base station does not accurately depict the reading of the environment at that time; the reading may have changed during this time. Since cluster formation uses CSMA scheme, so, in the best case scenario, when there are no re-transmissions and all messages are successfully sent and received, if we assume a message transmits in 1 time unit, then the cluster formation stage accounts for

$$O(m)+k+2\times O(n)$$

time units where k is the time after which the nodes in a cluster are guaranteed to have received the cluster head advertisement, m is the maximum size of a neighborhood and n is the maximum size of a cluster. On the other hand, in our scheme, once the nodes wake up, it will take $O(n)$ time units to get data from all member nodes in the TDMA schedule. It is for this reason that the delay encountered in ELECTION is almost 3 times the delay in our scheme. In our proposed scheme, the clustering structure, the cluster-heads, and the TDMA schedule is already in place when the nodes wake up, so unnecessary delay while sending report to the base station is avoided. This ensures that the reported data matches closely with the current data reading of the event being sensed.

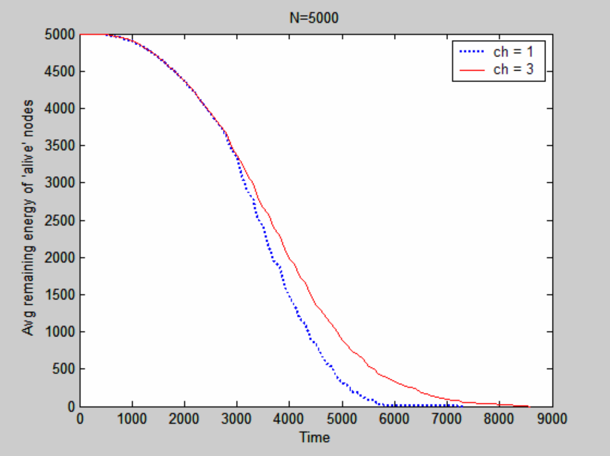
Figure 3. Comparison of % of alive clusters with different no. of candidate cluster heads. N=5000, CH=1 and CH=3, event=natural, A=100x100 sq. units, D_{th}=90, D_{in}=85



One of the main features of our proposed scheme is its simple yet effective load balancing. Figures 3 and 4 show the effect of this load balancing scheme. We considered a network of 5000 nodes. Candidate cluster heads were chosen from member nodes that lay within 1.5 units of distance from the original volunteer cluster heads. To ensure a fair comparison, we generated the same network configurations, the same event characteristics, and the same clustering structure for both the simulations. We set the initial intensity of all the nodes to 85 and the threshold intensity to be 90. When the event exceeds the threshold, somewhere around 3000 time units, cluster heads start to collect data from their cluster members and then send a report to the base station. Nodes continue reporting until no more clusters are alive. It is clear from Figure 3 that increasing the candidate clusters from 1 to 3 increases the percentage of live clusters and thus extends the network's duration significantly.

Figure 4 shows the plot of average remaining energy of "alive" nodes for the same simulation of 5000 nodes. A node is considered "alive" if the cluster head of the cluster to which it belongs to is alive. More specifically, here the average remaining energy is equal to the total remaining energy of "alive" nodes over the total number of nodes in the network. Once the event is detected to cross the threshold somewhere around 3000 time units, the average remaining energy of "alive" nodes for the network with a single candidate cluster-head goes down more rapidly than the average remaining energy of "alive" nodes for the network with three candidate cluster-heads. At 5000 time units, for example, the average remaining energy for $ch = 1$ is around 250 units, while the average remaining energy for *ch = 3* is around 750 units. Thus, our scheme exhibits a considerable savings of 500 units of energy on average for this network of 5000 nodes.

Figure 4. Comparison of average remaining energy of alive clusters with different no. of candidate cluster heads. N=5000, CH=1 and CH=3, event=natural, A=100x100 sq. units, D_{th}=90, D_{in}=85



CONCLUSIONS AND FUTURE WORK

We have proposed a collision-free MAC protocol for wireless sensor networks that gathers data from the affected part of the network in an efficient manner. The scheme saves energy by organizing the networks into clusters and also introducing adaptive sleep cycles for both the radio and sensor in the node. The scheme also increases the network lifetime by employing a load balancing scheme. The proposed scheme is observed to perform better in terms of achievable network lifetime and responsiveness to events as compared to similar existing schemes like ELECTION.

In the future, we intend to investigate more sophisticated load-balancing schemes. We also plan to look at schemes which are capable of handling multiple (overlapping) events occurring at the same time.

ACKNOWLEDGEMENTS

This work was partially supported by the US Army Research Office under grant DAAD19-01-1-0502 and by the US National Science Foundation under grants EIA 0086251 and 0424546. The views and conclusions herein are those of the authors and do not represent the official policies of the funding agencies or the University of Central Florida.

REFERENCES

Bandyopadhyay, S. and E. J. Coyle. 2003. “An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks.” *Proceedings of IEEE INFOCOM’03.*

Bandyopadhyay, S. and E. J. Coyle. 2004. “Spatio-Temporal Sampling Rates and Energy Efficiency in Wireless Sensor Networks.” *Proceedings of IEEE INFOCOM’04.* 4(1):1—16.

Begum, S.; S. C. Wang; B. Krishnamachari; and A. Helmy. November 2004. “ELECTION: Energy-Efficient and Low-Latency Scheduling Technique for Wireless Sensor Networks.” *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN’04)*, pp.60-67.

Elson, J; L. Girod; and D. Estrin. 2002. “Fine-Grained Network Time Synchronization using Reference Broadcasts.” *OSDI* 2002, Boston, Massachusetts.

Heinzelman, W. R.; A. Chandrakasan; and H. Balakrishnan. 2000. “Energy-efficient communication protocol for wireless microsensor networks.” In *Proceedings of 33rd Annual Hawaii International Conference on System Sciences*.

Lu, G.; B. Krishnamachari; and C. S. Raghavendra. April 2004. “An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks.” *Proceedings of the 18th International Parallel and Distributed Processing Symposium.*.

Manjeshwar, A. and D. P. Agrawal. 2001. “TEEN: a routing protocol for enhanced efficiency in wireless sensor networks.” *IPDPS*. San Francisco, California.

Van Dam, T. and K. Langendoen K. November 2003. “An adaptive energy-efficient MAC protocol for wireless sensor networks”, *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, pages 171–180, Los Angeles, California.

Ye, W.; J. S. Heidemann; and D. Estrin. June 2004. “Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks”, *IEEE/ACM Transactions on Networking*, Volume: 12, Issue: 3, Pages: 493 – 506.