ELECTION: Energy-efficient and Low-latEncy sCheduling Technique for wIreless sensOr Networks *

Shamim Begum, Shao-Cheng Wang, Bhaskar Krishnamachari, Ahmed Helmy
Email: {sbegum, shaochew, bkrishna, helmy}@usc.edu
Department of Electrical Engineering-Systems
University of Southern California

Abstract

We propose ELECTION, a new sleep scheduling scheme that adaptively schedules the sleep cycles of both communication radios and sensors in wireless active sensor networks. Taking advantage of spatial and temporal correlations in the underlying physical phenomenon, our scheme controls sleeping schedules of radios and sensors, and adaptively meets the energy efficiency, latency and responsiveness needs of applications. During the normal phase of operation, sensors take samples of the environment once at each wakeup time, and based on the perceived environment they adapt their sleep cycles. When an abnormality is perceived from the sampled data, sensors communicate with their neighbors to form a cluster and report to the base station. Analysis and simulation results show that ELECTION outperforms existing protocols significantly in terms of energy savings as well as delay and responsiveness.

1 Introduction

Recent advances in microelectronics, integrated circuits and communications have allowed sensor integration with processing and communicating capabilities into low-cost embedded sensor devices. These devices are capable of monitoring a wide variety of ambient conditions: temperature, pressure, humidity, soil makeup, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects [1]. These devices are empowered with certain processing, memory, and communication capabilities. Networking such devices serve a wide varieties of applications ranging from environmental [2], structural [3], factory, and seismic [4] monitoring to target tracking.

Each of these deployments involve a large number of sensor devices and is expected to last as long as possible. In past few years achieving energy efficiency for these wireless sensor networks (WSN) has been the most important research challenge. Energy efficient protocols have been proposed for MAC [5, 6, 7], topology control [8, 9, 12], and data aggregation [16, 17]. The main focus of these works is in the design of novel sleep scheduling schemes wherein nodes turn off their communication radios during the sleep. Typically these works assume passive sensors where sensor themselves consume very insignificant amount of power.

In contrast, we present a new sleep scheduling scheme that schedules both communication radios and the sensors. We assume smart (or active) sensors where sensors act as smart agents, and are able to sense the environment in a responsive and timely manner. They communicate to each other using communication radios to perform collaborative and integrated sensing. These sensors are massively deployed in an environment and configured in a network to communicate to a base station to report the accumulated data. The sensor pod developed by NASA's Jet Propulsion Laboratories is an example of smart sensor [18]. These sensors can be used in similar deployments for environmental and structural monitoring to monitor physical phenomenons. When the processor and communication radios are off, these sensors consume about 20% energy of transmission, and therefore controlling these sensors has been a source of significant energy savings compared to traditional passive systems [19].

We take an example of a wireless network of active sensors deployed to monitor some phenomenon in a chemical powerplant. The network is monitoring the environment to report any abnormalities of the underlying phenomenon. Energy efficiency, delay and responsiveness of such applications vary with different modes or phases of the applications. For example, during normal operation, achieving energy-efficiency is more important than assuring low latency or high responsiveness. While the phenomenon tends to increase because of some abnormalities in the environment, it is more important to ensure low latency and high responsiveness than energy-efficiency. Designing a sleep scheduling scheme that ad-

*This work is supported in part by NSF grants 0325875, and 0347621.

dresses such dynamic requirements of energy-efficiency, responsiveness and delay-sensitivity of active sensor network applications, and provides design choices for wide varieties of applications motivates this work.

Measurements of a physical phenomenon in a geometrically closed area exhibit both spatial and temporal correlation. Spatial correlation is well-exploited in the design of wireless sensor network protocols, e.g. data aggregation [20], and coding schemes [21]. Consider a sensor network in Figure 1 deployed to monitor a region R in a chemical powerplant. Assuming all sensors are calibrated, at any point in time, all sensors in a small area r in R measure the same phenomenon. When some abnormal chemical reaction in r causes the phenomenon to increase, all sensors in r read this increasing phenomenon, and perceive the increase. Taking advantage of the spatio-temporal correlation of physical phenomenon the WSN is monitoring, we propose a new sleep scheduling scheme. Each node independently adjusts its sleep cycle based on its sensor measurement and perception of the environment during normal phase of operation, therefore, achieves energy efficiency. When a node perceives that an **event** is approaching, it starts communicating with its neighbors. During this phase of operation, a neighborhood of sensors selects the node which perceives the abnormality most severely as cluster head. The cluster head forms a TDMA-based schedule to collect data from its cluster members which it aggregates into a single signal, and transmits to the closest base station directly. During this phase, the network achieves low latency and high responsiveness.

We run high level simulations of our scheme to evaluate its performance, and to compare with existing schemes. The simulation results show that ELECTION is much more energy efficient, delay-sensitive, and responsive than other protocols.

The rest of the paper is outlined as follows. Section 2 presents related work. A detail description of our protocol is presented in section 3. Section 4 presents an analysis of our protocol. The simulation results are presented in section 5. Section 6 concludes with a brief outline of our future work.

2 Related Work

Wireless sensor networks are typically deployed for some data gathering applications. There are two paradigms of data gathering: periodic, and event-based data gathering. In the periodic data gathering paradigm, all sensor nodes collect data all the time, and route the data generally through a tree. Scientific data collection is an example of periodic data gathering. In event based data gathering, the interest of the end user is in specific kind of information or **event**. Chemical detection is an example of event based data gathering.

There are quite a significant amount of work done in past few years in the design of energy-efficient, low-latency, and fault tolerant communication protocols. These work design scheduling schemes to control duty cycles of communication radios. S-MAC [5] is a periodic sleep scheduling scheme where nodes turn their radios off during sleep. S-MAC reduces energy waste of idle listening significantly, however, it increases latency and reduces throughput. T-MAC [6] dynamically adapts the active cycles. Latency in T-MAC is expected to increase compared to 802.11 like MAC because the data arrived during sleep is queued until the next active cycle. D-MAC [7] solves the high latency problems of periodic sleep by giving periodic active/sleep schedule an offset that depends upon its depth on the data gathering tree.

LEACH [9] is a hierarchical clustering protocol that minimizes global energy usage by distributing the energy load among all sensor nodes at different points in time. Cluster formation is periodic and is done at an *a priori interval*. At these intervals, each node locally and independently decides to be cluster head, and takes the burden of acquiring data from the nodes in the cluster, fusing the data to obtain an aggregate signal, and transmitting this aggregate signal to the base station. Span [10] is a distributed randomized topology control protocol. In Span, each node locally decides whether to sleep or stay awake depending on an estimate of how many of its neighbors will benefit from its being awake, and its remaining energy. [11] presents two topology control protocols to conserve energy usage by turning off *redundant nodes* of the network. Geographic adaptive fidelity (GAF) identifies redundant nodes by their physical location and a conservative estimate of radio range. Cluster-based energy conservation (CEC) determines redundancy by directly observing radio connectivity. S-MAC, T-MAC, D-MAC, LEACH, Span, GAF, and CEC fit into the periodic data gathering paradigm.

TEEN [12] and APTEEN [13] are most related to our work in terms of design philosophy. These protocols fit into the event driven data gathering paradigm. Periodically nodes form cluster similar to LEACH. Unlike LEACH, nodes sleep periodically instead of staying awake. During sleep, nodes turn their communication radios off leaving the sensors on. Nodes sense the environment continuously and wake up only when the event threshold is detected. APTEEN is an extension of TEEN with the addition of features for data querying. The main similarity of TEEN/APTEEN and ELECTION is that they use the threshold of physical phenomenon as **event**. However, there is quite significant difference between these two protocols. Our protocol takes advantage of spatio-temporal correlation of

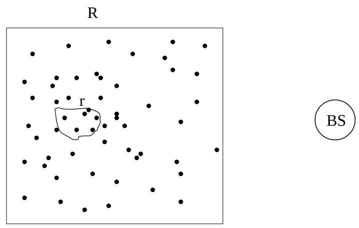

Figure 1: Sensor network to monitor a region R

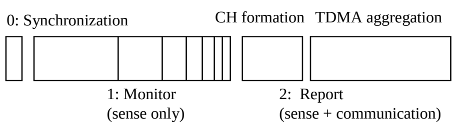

Figure 2: Timing diagram

the phenomenon to adapt its sleep cycle as well to communicate with neighbors.

3 Protocol Details

3.1 Assumptions

Figure 1 is a model of a wireless sensor network deployed for a monitoring/detection application. The network is continuously monitoring some physical phenomenon within a region R to report an event e which, we assume, would occur in a sub-region r in R. All the sensors are homogeneous, fixed, and energy-constrained. The fixed base station BS is located far from the region R. We assume that both the communication radio and the sensor of a sensor node can be turned off to save energy. They can be turned off independently. Certain characteristics of application is assumed. End user should be able to specify a specific value or a range of value that is alarming for the environment the WSN is monitoring. A threshold tolerance should also be specified.

3.2 Basic Mechanisms

Following is a list of system parameters we use. These parameters are tuned depending on the application, and its responsiveness, delay, and energy-efficiency requirements.

1. Durations of initial sleep cycles: S_{in} and S_{in}^0. To collect initial sensed data S_{in}^0 is set very small compared to S_{in}. A larger S_{in} saves energy, however, it increases latency.
2. Initial active period: A_{in}. It is set according to the average node degree so that a node gets enough time to communicate with all its neighbors.
3. Data threshold: D_{th}.
4. Gradient threshold: G_{th}.
5. Sleep reduction function: F_{sr}: It is a monotonically decreasing function of previous sleep duration.
6. Threshold tolerance: Δ_{th}.

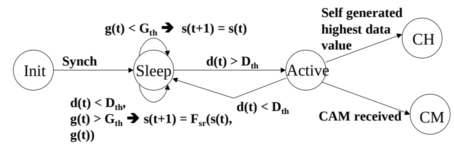

Figure 3: State transition diagram

The sensor network operates into following three phases. Figure 2 presents timing diagram and Figure 3 illustrates state transition diagram of a single node.

Phase 0: Synchronization: Initially nodes synchronize with each other using existing synchronization protocols, e.g. RBS [14], pairwise synchronization [15] etc.

Phase 1: Periodic sleep and monitor: When the network is synchronized, a sensor node s randomly goes to sleep for a period of S_{in}^0. At the end of sleep cycle S_{in}^0, the node wakes up, senses and monitors the environment, and sleeps for a duration of S_{in}. During sleep, it turns both of its sensor and communication radio off. When the node wakes up, it turns its sensor on, records the sensed data of the phenomenon, calculates the gradient considering sensed values in current and last wakeup time. The gradient indicates the trend of sensed environmental phenomenon during the last sleep cycle. Depending on the sensed data d and gradient g, a sensor does its next state transition as follows:

- If d exceeds the data threshold D_{th} (within a tolerance of Δ_{th}), the node changes to active state, turns both its sensor and communication radio on, and enters into second phase. The threshold tolerance Δ_{th} allows more nodes to be responsive to the event.
- If d is below D_{th}, however, g exceeds the gradient threshold G_{th}, the node remains in this phase. It reduces its sleep cycle using a sleep reduction function F_{sr}. We describe this function is section 3.3.
- In all other cases, it remains in this phase without changing the sleep cycle.

Phase 2: CH formation, data aggregation, and report: A node enters this phase either if it detects the sensed data has exceeded the threshold or perceives an alarming increase. At the beginning of this phase, a node waits for an active period A_{in} plus a random period (to avoid synchronization), and then starts communicating with its neighbors by sending neighborhood advertisement message. The advertisement contains instantaneous sensor reading collected during the last wakeup time. After sending the advertisement, a node waits some time to receive advertisements of its neighbors. After this interval, the node compares the sensor readings of its own and all its neighbors. If the node has highest reading, it elects itself as cluster head, and broadcasts the cluster head advertisement message (CAM) to all its neighbors. The node with highest sensor reading is probably the node closest to the region where the phenomenon is increasing. A Node receiving multiple CAMs selects the CH with highest signal strength as the CH to join. It sends a cluster membership message to the CH. After receiving the membership messages the cluster head creates a TDMA schedule, and broadcasts the schedule to all its cluster members. A cluster member transmits its data during its assigned slot, and sleeps otherwise. The cluster head aggregates data from all its cluster members, and transmits the aggregated data to the closest base station directly. Broadcast of the advertisement and membership messages are communicated using a CSMA MAC protocol. If the sensed data d drops below threshold D_{th} in active state, a node enters into the first phase.

3.3 Adapting the sleep cycle using F_{sr}

One fundamental difference between ELECTION and other protocols is that ELECTION turns sensors off during sleep. It saves energy, however, it loses the environmental information. So it is very important that the sleep cycle adapts with the changes in the phenomenon the WSN is monitoring. We ensure this by introducing a sleep cycle reduction function that is a function of current sleep cycle and gradient of the environment. Let $s(t)$ and $g(t)$ be the sleep cycle duration and gradient at time t respectively. $s(t+1)$, the sleep cycle duration at time $t+1$ is determined using the sleep reduction function F_{sr} as follows.

$$s(t+1) = F_{sr}(s(t),\ g(t)) \tag{1}$$

In order to assure the desired adaptivity, we design two types of sleep reduction function.

1. Exponential F_{sr}:

$$s(t+1) = \begin{cases} \frac{1}{2}\ s(t) & G_{th} < g(t) < 2G_{th} \\ \frac{1}{4}\ s(t) & 2G_{th} < g(t) < 3G_{th} \\ \frac{1}{8}\ s(t) & 3G_{th} < g(t) < 4G_{th} \\ \ldots \end{cases}$$

Exponential F_{sr} is good with respect to latency, however, it reduces the sleep cycle very aggressively that may cause a node to sleep with a very small sleep cycle which is quite energy expensive. Moreover, it wastes energy when the phenomenon increases temporarily and decreases again in which case it does not increase the sleep cycle.

2. Geared F_{sr}:

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

Geared F_{sr} reduces the sleep cycle similar to a gear. It alleviates the problems of the exponential function and is more suitable for most applications. When the phenomenon starts decreasing after an abrupt increase, geared F_{sr} increases the sleep cycle accordingly. The multiplicative decrease is changed from the factor of 2 to the factor of 4 when the phenomenon is fast approaching to the threshold in order to increase its responsiveness.

4 Protocol Analysis

4.1 Performance Metrics

We evaluate our protocol and compare with existing protocols using energy efficiency, delay/latency, and responsiveness as performance metrics. We define energy efficiency in terms of remaining energy of all nodes in the network and the number of nodes still alive at any point in time. We define *latency* as the delay of an event detection and report. In other words, latency is the time it takes to detect the event that has already occurred and to report it to the BS. However, purely evaluating the delay between the timing of report generated and the actual timing of data threshold being reached is not fair since our protocol adaptively adjusts the sleep cycle. Therefore, we propose another metric called *responsiveness* which measures the difference between reported data value and the data threshold (e.g. in a temperature monitoring applications, it's the degree). For example, a temperature monitoring application in a chemical powerplant sets the temperature threshold to 100^oF. However, the report sent by the cluster head to the BS indicates that the threshold is reached at 101^oF. The responsiveness in this case is 1^oF.

4.2 Strengths and Limitations

Sensor nodes in ELECTION do not communicate in the first phase (phase 1). They start communication only when they perceive a sudden increase (exceeding D_{th}) or an alarming increasing trend (exceeding G_{th}). It saves the communication energy in this cycle. We trade energy with latency at this phase. If the physical phenomenon is easily recordable and the system parameters are input correctly, the latency is expected to be insignificant. In the second phase, we trade low-latency with energy.

One novelty of our algorithm is: it guarantees that all neighboring sensors in a region r enter into the second phase at the same time. Our guarantee is based on the spatial correlation of the underlying phenomenon. If the phenomenon increases in a region r, all nodes in the region r perceive the increase, and therefore, enter into the second phase whenever their last sleep periods end. Thus, the worst case delay is in the order of duration of the last sleep cycle. Moreover, because of the temporal correlation of the phenomenon, the last sleep cycle is expected to be small compared to initial sleep cycle S_{in}.

The main limitation of our schemes is that it depends a lot on the underlying phenomenon. For industrial applications, *a priori* information of the physical phenomenon is available, however, it may not be available in other contexts. The protocol is also not suitable for applications where the phenomenon does not exhibit spatio-temporal correlation, e.g. seismic monitoring applications. Moreover, the idea of nodes sleeping for a long time may lead to large clock skews and synchronization problems. ELECTION is a single hop protocol which may limit its scalability. We discuss techniques to improve its scalability is Section 6.

4.3 Analytical Comparison

In this section, we present an analytical comparison of total energy, latency, and responsiveness of ELECTION, LEACH, and TEEN. Let us define T as the total period that the network is expected to operate. T is divided into two parts: T_1 is the duration of first phase at the end of which the data threshold is reached, and T_2 is the duration of second phase that starts when the threshold is reached. D_{th} and G_{th} are protocol parameters representing the data threshold and the gradient threshold respectively. The network is defined by three parameters: node density (ρ), average node degree (δ), and total area of the network (A) . Let us define a function $f(t)$ that increases monotonically in time. $f(t)$ represents the underlying environment the sensor network is monitoring.

Let E denote the total energy dissipation of the network. We can divide E into three components: sensing energy, communication energy, and reporting energy. In ELECTION, sensor nodes do not communicate during the first phase, and take samples once only when they wake up from the previous sleep cycle. During the second phase, both the sensors and communication radios are on. During this phase, nodes form cluster and report to the base station. All these three protocols form cluster dynamically, and the mechanisms involved in cluster formation are same. Therefore, for comparison purposes, we are only interested in number of clusters and frequency of cluster formation in these protocols. Let T_r be the reporting interval at which cluster heads report aggregated data to the BS in these protocols.

ELECTION

$$E_{election} = E^{I}_{sense} + E^{II}_{sense} + E^{II}_{cluster} + E^{II}_{report} \quad (2)$$

Let $g(t)$ be the function representing gradient of the environmental phenomenon. Sleep duration at time t is represented by a function $s(t)$.

$$g(t) = \frac{df(t)}{dt}$$

$$s(t) = \begin{cases} s(t-1) - 1 & s(t-1) \geq 1 \\ F_{sr}(g(t),\ S_{in}) & s(t-1) = 0 \end{cases}$$

Let us define a function $C(t)$ that represents a counter that counts total number of times a node wakes up (or sleep) during the first phase.

$$C(t) = \begin{cases} 0 & t = 0, \text{s(t)} \geq 1 \\ 1 & s(t) = 0 \end{cases}$$

N_s, the number of times a node takes samples of the environment, is given by

$$N_s = \int_0^{T_1} C(t)dt \quad (3)$$

In ELECTION, the duration of sleep cycle varies with the behavior of the underlying phenomenon. Let λ_s represent the expected sleep duration. Therefore,

$$N_s = \frac{T_1}{\lambda_s}$$

Let E_s be the energy dissipation for a single sensing operation.

$$E^{I}_{sense} = \rho A E_s N_s = \rho A E_s \frac{T_1}{\lambda_s}$$

In the second phase, nodes sense at every reporting period. Therefore,

$$E^{II}_{sense} = \rho A E_s N_s = \rho A E_s \frac{T_2}{T_r}$$

N_c, the total number of clusters, is given by,

$$N_c = \frac{\rho A}{\delta}$$

Let energy dissipation of a single cluster formation and that of a single report are represented by E_c and E_r respectively. Therefore,

$$E^{II}_{cluster} = \frac{\rho A}{\delta} E_c$$

$$E^{II}_{report} = \frac{\rho A}{\delta} E_r \frac{T_2}{T_r}$$

Let L, η and β represent average latency, worst case latency, and worst case responsiveness respectively. Let G_{max} represent the maximum gradient threshold ELECTION can respond to.

$$\begin{aligned} E_{election} &= \rho A E_s \tfrac{T_1}{\lambda_s} + \rho A E_s \tfrac{T_2}{T_r} + \tfrac{\rho A}{\delta} E_c \\ &\quad + \tfrac{\rho A}{\delta} E_r \tfrac{T_2}{T_r} \\ L_{election} &= \tfrac{1}{2} \text{ Last sleep duration} \\ \eta_{election} &= \text{Last sleep duration} \\ \beta_{election} &= G_{max}\ S_{in} \end{aligned} \quad (4)$$

LEACH and TEEN

Let α be the percentage of nodes cluster heads in these protocols. T_c is the period at which cluster is formed. Let S be the fixed sleep cycle. In LEACH, nodes sense and communicate at every reporting period. In TEEN, nodes communicate at every reporting period, however, they sense at a fixed, high rate (every 5 seconds). In TEEN, report to BS is sent only when the threshold is crossed like ELECTION.

$$\begin{aligned} E_{leach} &= \rho A E_s \frac{T}{T_r} + \frac{\rho A}{\alpha} E_c \frac{T}{T_c} + \frac{\rho A}{\alpha} E_r \frac{T}{T_r} \\ L_{leach} &= \frac{1}{2}\, T_r \\ \eta_{leach} &= T_r \\ \beta_{leach} &= G_{max}\ \eta \end{aligned} \tag{5}$$

$$\begin{aligned} E_{teen} &= \rho A E_s T + \frac{\rho A}{\alpha} E_c \frac{T}{T_c} + \frac{\rho A}{\alpha} E_r \frac{T_2}{T_r} \\ L_{teen} &= \frac{1}{2}\ \text{Clock tick} \\ \eta_{teen} &= \text{Clock tick} \\ \beta_{teen} &= G_{max}\ S \end{aligned} \tag{6}$$

From the above equations, it is evident that ELECTION saves significant energy of cluster formation (T_2 vs. T) as well as sensing. When $E_c \gg E_s$, savings in cluster formation predominate. When $E_s > E_c$, savings in sensing predominate. Thus energy savings in ELECTION depend on the ratio of communication and sensing energy. The upper bound of responsiveness (β) in ELECTION can be guaranteed as long as the gradient of the underlying phenomenon remains within G_{max}. ELECTION presets the initial sleep cycle (S_{in}) to ensure a responsiveness of β if the gradient of the underlying phenomenon remains within G_{max}. On the other hand, LEACH and TEEN do not adjust sleep cycles on perceiving the phenomenon, and therefore, no such guarantee on responsiveness is assured.

5 Simulation Results

We perform a high level simulation of ELECTION, TEEN and a Hybrid protocol using a high level event simulator. The Hybrid protocol is a mix of TEEN and ELECTION. In this protocol, sleep cycle is fixed similar to TEEN, however, the cluster is formed on a demand basis similar to ELECTION. The network is composed of 36 uniformly distributed sensor nodes, and one faraway base station. The actual area covered by the network is divided into four quadrants. Each quadrant is assigned a sensing pattern; all nodes in this quadrant sense the same phenomena. A cluster is formed within each quadrant. The cluster head aggregates the data and reports to the base station.

We run each simulation for 600K seconds. We simulate the geared sleep reduction function with an initial sleep cycle of 256 seconds. The nodes in ELECTION form a cluster when the phenomenon exceeds the threshold. In TEEN, nodes form a cluster every 6000 seconds (T_c). The sleep cycle is fixed to 50 seconds. The cluster formation is similar to LEACH [9]. Table 1 presents the simulation parameters.

We simulate two different environments. In the first, the phenomenon changes 100 times during the entire simulation. The second phenomenon changes 20 times. Figure 4 and 5 present total remaining energy of all the nodes at different times of simulation for these two environments. TEEN forms a cluster at every cluster formation interval (T_c) independent of the environmental condition it is monitoring. Thus the major energy costs in TEEN are sensing and cluster formation. ELECTION outperforms TEEN because nodes do not sense the environment at fixed high rate. It also takes advantage of the spatial correlation to reduce the sleep cycle, and form the cluster on-demand. The Hybrid protocol saves energy of cluster formation, however, it spends significant amount of energy for the continuous sensing task. The second phenomena moves slower than the first, and therefore, the expected sleep duration becomes larger, which in turn results in significant energy savings in ELECTION. On the other hand, since TEEN/Hybrid has fixed sleep duration and fixed cluster re-forming cycle, changing the phenomena does not result in significant energy savings.

Figure 6 presents the number of nodes alive at different times of simulation. Because of periodicity of cluster formation and frequent sensing, nodes in TEEN die faster than in Hybrid and ELECTION. The death of nodes in Hybrid protocol is faster than ELECTION because of the energy waste due to the sensing.

Figure 7 and 8 present delay and responsiveness of ELECTION, TEEN and the Hybrid protocol. We simulate different sensing patterns in different simulation runs (X-axis) and plot average delay and responsiveness in the figures. As shown in Figure 7, the average delays in TEEN and Hybrid protocols are, as expected, around 25 seconds (1/2 of the fixed sleep cycle). The delay in ELECTION is highly dependent on the sensing phenomena, however, in most cases outperforms TEEN and the Hybrid protocol. On the other hand, Figure 8 clearly shows the superiority of ELECTION protocol with respect to responsiveness. We observe a stable 0.1 degree difference between reported data by ELECTION and the preset data threshold. This is a desirable feature if the application requires the protocol to provide certain guarantees that the report will not be generated after the environmental data sensed has gone far beyond the threshold. Nevertheless, the responsiveness of TEEN and Hybrid is highly dependent on the sensing pattern.

We also investigate the ratio of sensing energy con-

Parameter	Values
Simulation time	600K seconds
Initial energy of a node	5000 units
Sensing energy	1 unit
Reception energy	5 units
Tx. energy (between nodes)	10 units
Tx. energy (between node and BS)	50 units
Reporting interval (TEEN, Hybrid)	6K seconds
Initial sleep cycle (ELECTION)	256 seconds

Table 1: Simulation parameters

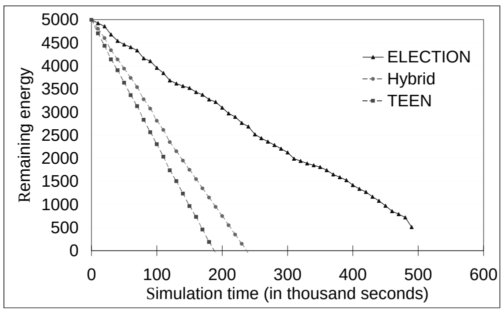

Figure 4: Remaining energy of nodes (E_s/E_{tx}=10%): Phenomenon 1

sumption (E_s) to transmission energy consumption (E_{tx}) using simulation. ELECTION is much energy efficient than existing protocols for sensors that consume a nontrivial portion of total energy cost, e.g. active sensors. However, we are also interested in the scenarios where sensors consume insignificant energy compared to communication, e.g. passive sensors. We perform simulations for such sensors where we assume sensors consume 1% of transmission energy (e.g. temperature sensors). Figure 9 presents the simulation result. It shows that the energy savings of ELECTION are visible even when the sensing energy is insignificant. Another interesting observation is that the energy cost difference between TEEN and Hybrid becomes larger. Cluster formation energy is the dominant factor in this scenario. ELECTION and Hybrid form cluster on-demand and therefore, are more energy efficient compared to TEEN.

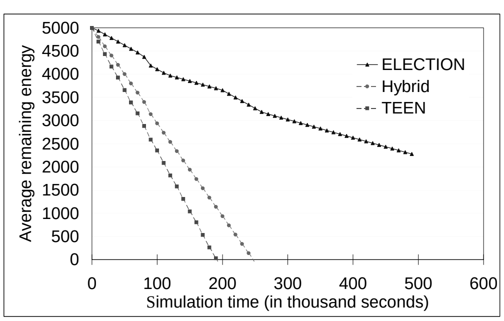

Figure 5: Remaining energy of nodes (E_s/E_{tx}=10%): Phenomenon 2

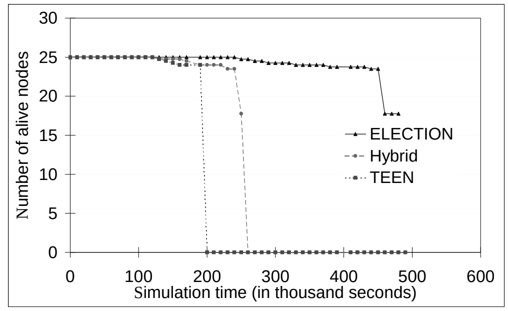

Figure 6: Number of alive nodes

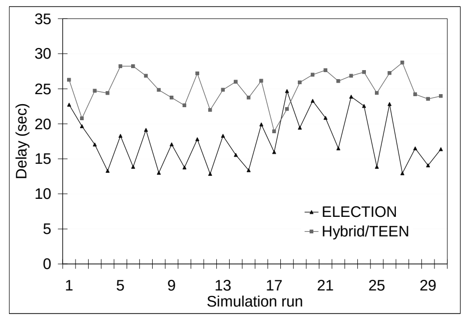

Figure 7: Delay

6 Conclusion and Future Work

We propose ELECTION, a new sleep scheduling scheme for wireless active sensor network taking advantage of spatio-temporal correlation of underlying physical phenomenon the network is monitoring. In this protocol, a node operates in three phases: synchronization, periodic sleep and monitoring, and cluster formation and reporting. Once synchronized, a node sleeps periodically, turning both of its sensor and communication radio off. When it wakes up in the first phase, the node turns the sensor on and senses the environment. If it perceives that the phenomenon is slowly increasing, it goes to sleep without changing the sleep cycle. If it perceives a drastic increasing trend in the phenomenon, it reduces its sleep cycle. If it perceives that the phenomenon has crossed the threshold, it changes to active mode when it turns both the sensor and communication radio on. It communicates with its neighbors to form a cluster. The cluster head creates a TDMA schedule. A node transmits its data during its schedule, otherwise sleeps. Both simulation and analytical results show that ELECTION outperforms existing protocols both in terms of energy savings, latency, and responsiveness.

In the future, we plan to enhance the design of our protocol as well as perform detailed simulations. A hier-

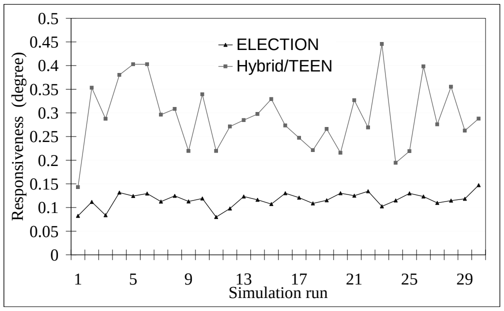

Figure 8: Responsiveness

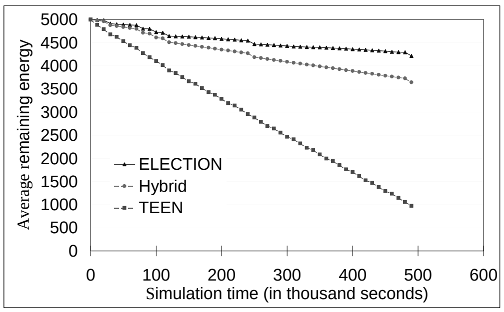

Figure 9: Remaining energy of nodes (E_s/E_{tx}=1%): Phenomenon 1

archical organization of cluster heads would allow multiple communication with BS, and improve its scalability. Currently cluster heads are responsible for creating TDMA schedule, collecting reports, and sending the reports to the BS. To balance the energy distribution of all nodes, the cluster head would assign the node with highest remaining energy to be the node to report to. Load balance would improve its scalability even in single hop mode. If nodes have enough storage to store history of data and its trend/gradient, they can predict the time at which the phenomenon is expected to exceed the threshold. Event prediction can save significant amount of energy in our protocol, however, it requires proper implementation of estimation theory. Our high level simulation does not simulate a realistic environmental phenomenon. We are planning to do detail simulations using network simulator ns [22] to compare the performance of LEACH, TEEN and ELECTION, and also to investigate the effect of MAC layer issues on the proposed scheme.

References

[1] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks", MOBICOM 1999, Seattle, Washington.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, "Wireless Sensor Networks for Habitat Monitoring", WSNA 2002, , Atlanta, Georgia.

[3] C. Lin, C. Federspiel, and D. Auslander, "Multi-Sensor Single Actuator Control of HVAC Systems", 2002.

[4] UC Berkeley, "Smart Building Admits their Faults", http://coe.berkeley.edu/labnotes/1101smartbuildings.html, November 2001.

[5] W. Ye, J. Heidemann, D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", INFOCOM 2002, New York, NY.

[6] T. Dam, K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", SenSys 2003, Los Angeles, California.

[7] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks", WMAN 2004, Santa Fe, New Mexico.

[8] Alberto Cerpa and Deborah Estrin, "ASCENT: Adaptive Self-Configuring sEnsor Network Topologies", INFOCOM 2002, New York, NY.

[9] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Sensor Networks", HICSS 2000, Mauli, Hawaii.

[10] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad HoC Wireless Networks", ACM Wireless Networks Journal, Volume 8, Number 5, September 2002.

[11] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks", CENS Tech Report 0006, 2003.

[12] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks", IPDPS 2001, San Fransisco, California.

[13] A. Manjeshwar and D. P. Agrawal, "APTEEN: A Hybrid Protocol for Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", IPDPS 2002, Fort Lauderdale, Florida.

[14] J. Elson, L. Girod and D. Estrin, "Fine- Grained Network Time Synchronization using Reference Broadcasts", OSDI 2002, Boston, Massachusetts.

[15] S. Ganeriwal, R. Kumar and D. Estrin, "Timing-Sync Protocol for Sensor Networks", SenSys 2003, Los Angeles, California.

[16] C. Intanagonwiwat, R. Govindan, and D. Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks.", MOBICOM 2000, Boston, Massachusetts.

[17] S. Madden, M.J. Franklin, J.M. Hellerstein, W. Hong. "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks.", OSDI 2002, Boston, Massachusetts.

[18] V. Tao, "The Smart Sensor Web: A Revolutionary Leap in Earth Observation", http://www.geoplace.com/gw/2003/0309/0309swb.asp.

[19] S. Madden, M. J. Franklin, and J. M. Hellerstein, "The Design of an Acquisitional Query Processor For Sensor Networks", SIGMOD 2003, Sandiego, California.

[20] S. Pattem, B. Krishnamachari, R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks", IPSN 2004, Berkeley, California.

[21] C. Tang, C. S. Raghavendra, V. K. Prasanna, "An Energy Efficient Adaptive distributed Source Coding Scheme in Wireless Sensor Networks", ICC 2003, Anchorage, Alaska.

[22] The Network Simulator ns-2. http://www.isi.edu/nsnam/ns/index.html.