TEAN-Sleep for Distributed Sensor Networks: Introduction and α-Metrics Analysis

Manikanden Balakrishnan, Eric E. Johnson and Hong Huang
New Mexico State University

Abstract **– One of the significant applications of Wireless Sensor Networks (WSNs) is threat surveillance, where events are rare and the sensing nodes are idle most of the time. Sleep is essential for conserving energy in such applications, since, for sensor technology, the energy consumption rate during idle state is in the same order of magnitude as during radio communications. This paper proposes Topology and Energy Adaptive, Non-synchronous (TEAN) sleep, a network-level sleep coordination scheme for WSNs that exploits the neighborhood connection redundancies to achieve longer durations of continuous node sleep, while ensuring uniform (close to 100%) network coverage and connectivity for reliable threat surveillance.**

The main objective of TEAN-sleep is to maximize lifetime without degrading the connection reliability of a network. In that perspective, this work adopts sensor-oriented α- metrics, a unified representation of energy conservation, coverage and connectivity efficiency, to quantify the performance of TEAN-sleep. Simulations predict 70% average α-performance improvement when employing TEAN-sleep as compared to the sleep deprived WSNs, under variety of monitoring scenarios.

I. INTRODUCTION

WSNs comprise of several tiny sensor nodes collaborating in their sensing, processing and communication processes to accomplish high-level application tasks. Recently WSNs have found extensive application in threat surveillance (natural disaster, enemy activity monitoring) that demand extended periods of continuous and unattended monitoring for events. Replenishing the battery energy in nodes is generally not possible due to the human inaccessible terrain the nodes are deployed on, and so the effectiveness of a WSN depends on its efficiency in using the limited energy supply. Energy conservation is a forefront research area for sensor network technology.

Threat monitoring applications are characterized by sudden burst of events (enemy detection) or in-frequent updates (such as hourly hurricane updates), where the average network traffic is significantly low and the deployed nodes spend most of their time in idle state, *a state in which the node is powered on, but the radio is not involved in any active communications*. However, due to the unpredictable event patterns and the impact of unnoticed disasters, continual and reliable surveillance is often needed. Consequently, WSNs are densely deployed for robustness against frequent node failures (common in error prone disaster areas) and for timely threat notification. In such cases, only a subset of nodes is required to establish a connection backbone for reliable information supply. The rest of the nodes could sleep (turning off the radio and CPU modules) to conserve energy without impacting network connectivity.

The latest specifications of the popular Mica2 sensor motes [1] show a power consumption ratio of approximately 1:2:4 for the mote in idle, reception and transmission states respectively. Given that the energy spent (in Joules) is the integral of power consumption over time, the idle-mode energy dissipation would dominate the overall energy expenditure in low traffic sensor applications (e.g. threat surveillance). A sleeping node, however, consumes about one tenth the power of an idle node [1].

Hypothesis: *Increasing the time that nodes in idle state can sleep would be significant for extending WSN lifetime, especially in extremely low traffic sensor applications.*

In this paper, we develop a topology level sleep coordination scheme that minimizes the number of active nodes required for maintaining a connected network (active set), thereby providing significant opportunities for sleep.

A. Contributions

1. Design of TEAN-sleep, where nodes employ network connectivity information to coordinate sleep operations, while maintaining a connected network at all times.
2. Performance analysis of TEAN-sleep using a family of α-metrics, which provides an integrated representation of energy efficiency and monitoring reliability of TEAN-sleep.

B. Related Work

Sleep organization can be approached in several dimensions. Heterogeneous approaches use low-powered sensor nodes to sense events and wake up high-powered nodes for intensive processing and long-haul communications [2]. Certain researchers have employed multi-radio motes [3], where a low power control radio is used for continuous monitoring of transmissions of interest and to wake up the data radio and the processor. In this work we consider a homogenous WSN and nodes without any special low-power radio modules.

The popular S-MAC [4] protocol provided a MAC layer approach to reduce idle listening using synchronous listen-sleep cycles, where nodes adaptively sleep when no active transmissions are detected at the beginning of every cycle. Powering up the sensor radio and CPU modules from sleep to active states consumes energy in the order of mJ ([5], [6]), which imply that avoiding frequent module-state transitions might yield better energy efficiency (validation is beyond the scope of this paper). A longer continuous sleep (than waking up in short cycles) could be achieved using network level sleep coordination ([7], [8]) that uses connectivity information to adapt the active participation of network nodes. The design approach of our work is similar to the ad-hoc sleep scheme in [9], but with conceptual modifications to suit WSN characteristics. Simplicity is preferred over complexity (such as the need for synchronization and global connectivity estimates), which is an efficient trade-off for WSNs.

1-4244-1513-06/07/$25.00 ©2007 IEEE

II. TEAN-SLEEP

This section provides a detailed discussion on the characteristics and design of TEAN-sleep.

A. Characteristics

- *Energy adaptive*: Energy-aware sleeping enables uniform energy depletion in the network, and guarantees adequate sensing coverage for large fraction of the operational time.
- *Topology adaptive*: Sleep decisions adapt to the topology dynamics, whereby only redundant nodes are switched off, thus ensuring a connected network.
- *Unsynchronized sleep cycles*: Neighbors do not have to synchronize to common sleep schedules (unlike S-MAC [4]). Sleep decisions are explicitly communicated and the neighborhood locally self-synchronizes to topology changes. Removing the need for synchronization is essential for WSNs, since it either reduces mote size (hardware synchronization) or protocol complexity (software synchronization).
- *Free sleep*: A listen period followed by a sleep period constitutes one *duty cycle*. In TEAN-sleep, the sleep period is not bounded by periodic wakeups imposed by fixed duty cycles. Each node sleeps freely for the entire sleep duration (computed dynamically), and then wakes up to listen before making further sleep decisions. The listen-sleep operations alternate, but the duty cycle length varies in accordance with the adaptive sleep duration. In short, there is no notion of periodic duty cycling as opposed to schemes like [4], where the sleep operations adapt within a fixed length duty cycle.
- *Distributed, minimal overhead*: Nodes locally decide to sleep based on their own perspective of the neighborhood connectivity, estimated from minimal information exchange whose energy overhead would be nullified by sleep.
- ***Goal and tradeoff*:** This research targets threat surveillance applications (rare and unpredictable traffic patterns), where maintaining a connected network consistently is extremely important for reliability. *TEAN-sleep aims to maximize node sleeping time, but employs a conservative (safe) approach to guarantee seamless network coverage and connectivity.*

B. Protocol Architecture

The sleep embedded protocol stack designed for this work is shown in figure 1. The sleep agent could accept run time commands from higher agents to avoid conflicts with the network setup established by the higher layers. For example, routing entities could disable node sleep for a specified duration to prevent breakage of dynamically established paths. Sleep commanding is a cross layer optimization, and is purely application specific. In this work, we do not employ any higher-layer commands, and leave the application framework to repair sleep disruptions (e.g. finding alternate paths in response to link disconnections).

Peer sleep entities communicate using the services of lower layers, and the data link layer of our architecture is slightly modified to direct the sleep and application packets to the corresponding protocol agents. In this design, the sleep entity can operate independently from other layers and can directly embed into most protocol stacks with only a minor modification to the link layer internal packet directing routine.

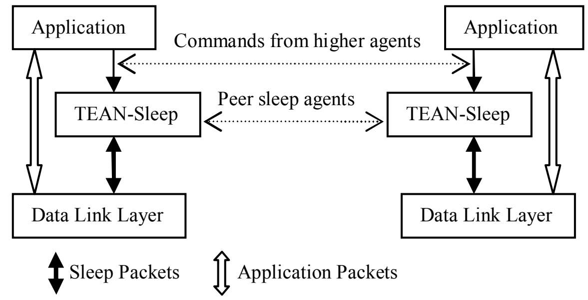

Figure 1. TEAN-sleep Architecture

C. Protocol Overhead

Control packets and the internal data structures for information exchange and maintenance represent the overhead.

1) Control Packets

<u>Hello</u>: Every node periodically broadcasts a hello packet that specifies its current energy level (Joules) and its one-hop neighbors (identified through neighbors' hellos). Hello exchanges provide extended (2-hop) neighborhood information, which is essential for making sleep decisions. Fig. 2 shows the format of hello packet.

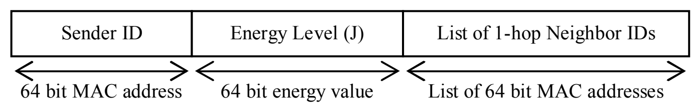

Figure 2. Hello Packet Format

The length of hello interval (HI) depends entirely on the network dynamics. For static sensor networks, the HI could be in the order of many seconds, generating a negligible overall overhead (hello size depends only on the neighborhood size).

<u>Sleep_beacon</u>: Every node broadcasts a sleep_beacon before going to sleep, which specifies the sleep duration. Sleep beacons enable topology awareness, whereby nodes adaptively change their sleep decisions based on the neighborhood state changes. Fig. 3 shows the structure of sleep beacons.

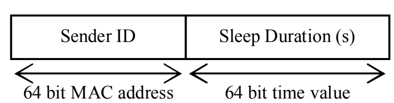

Figure 3. Sleep_beacon Format

2) Internal Data Structures

Node sleep decisions are based on the current neighborhood state, maintained in two internal data tables (not broadcasted).

<u>Neighborhood Information Table (NIT)</u> (figure 4) maintains the local connectivity information, which has the list of 1-hop neighbors and neighbors' neighbor lists.

1-hop neighbor	**Sleep Flag**	**Extended neighbors**	**Expiry**
Neigh 1	YES	List of 1's neighbors	3*HIs
Neigh 2	NO	List of 2's neighbors	3*HIs
…..Neigh n	NO	List of n's neighbors	3*HIs

Figure 4. NIT Structure

A sleep flag indicates the sleep status of the neighbors. NIT is updated on every control packet reception to maintain up-to-date topological information (for reliable sleep decisions). The NIT entries have an implementation specific expiry period, and in this research we consider a neighbor to be lost if no updates have been received for three HIs.

Energy_cache (Figure 5) maintains the current residual energy levels (Joules) of neighbors, which is updated from neighbor hellos. An expired neighbor entry in the NIT causes the removal of the corresponding entry in the energy_cache.

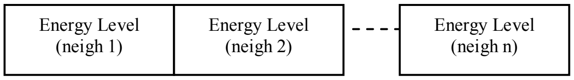

Figure 5. Energy_Cache: Array of neighbor energy levels

D. *Protocol Design*

The key design problem of TEAN-sleep is the formulation of node sleep decisions, such that a connection backbone is always guaranteed. TEAN-sleep defines

Sleep Eligibility Condition (SEC): *A node, n, is eligible to sleep if and only if all 1-hop neighbors in n's NIT, both awake and sleeping, are connected to at least one other currently awake neighbor of n.*

If and only if a node satisfies the above condition, SEC is set to 1, else SEC is 0. Note that a node must consider its sleeping neighbors while evaluating SEC, since the sleeping neighbors should have connectivity when they wake up.

1) SEC Explanation

Figure 6 uses a simple scenario for a clear explanation of the SEC concept (adjacent information represents the NITs)

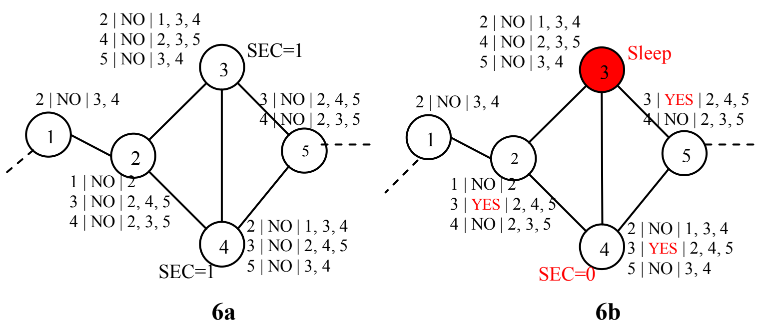

Figure 6. Concept of SEC

Node 1 cannot go to sleep (SEC = 0) since its neighbor 2 is not connected to at least one other of its awake neighbor. Same applies to 2 and 5. Node 3's SEC is 1, since all its neighbors (2, 5 and 4) are connected to at least one other *awake* neighbor of 3 (4 and 2 are mutually connected and 5 is connected to 4). Similarly 4's SEC is also 1.

Assume 3 broadcasts a sleep_beacon (goes to sleep). Figure 6b shows the corresponding topological changes. Nodes 2, 4 and 5 flag 3 as sleeping for the duration specified in the beacon. Node 4 re-computes its SEC to 0, since 2 and 5 are now not connected to any of 4's current "awake" neighbors. Node 4's SEC remains 0 until 3 wakes up (signaled by hello broadcast).

Every node ensures that there is an alternate active connectivity for all its neighbors before deciding to sleep, maintaining an overall connected network (since neighborhoods overlap).

2) Sleep Coordination Procedure

The TEAN-sleep entity can be viewed as a state machine with the following operational states:

- *Sleep*, where the radio and processor modules are switched off for *Sleep_Time* (T_s) duration. The energy-adaptive T_s design is explained in the next section. Sensor modules consume negligible power [1], and so they are always active detecting events and waking up other modules.
- *Awake_RecordUpdate (Awake_RU)* is a state of channel listening for control packets and updating internal records. Nodes listen for *Awake_RU_Time* (T_{ru}), which is set to one HI, since it would be the expected interval to receive all active-neighbor hellos in an unsynchronized neighborhood.
- *Awake_SleepWait (Awake_SW)* is a waiting state before sleep to randomize the sleeping instants of neighbors. Nodes wait for a random *Awake_SW_Time* (T_{sw}), sufficient to receive and process neighbor sleep_beacons and update their SECs. A node could lose its sleep eligibility in the wait state due to neighbor sleep decisions. The main purpose is to prevent two nodes from sleeping simultaneously due lack of topological convergence, which might lead to network partitions. In this work, T_{sw} is computed as P_A*HI/2 (see next section for P_A explanation) to make high-energy nodes wait for longer time and prevent monopolized sleep situations.

Nodes, upon powering on or waking up from sleep, send a hello packet to announce their active participation, and stay in *Awake_RU* for T_{ru} period. The TEAN-sleep agent, while awake, broadcasts a hello packet every HI. Upon T_{ru} timeout, nodes switch to *Awake_SW* and wait for a random T_{sw} period before deciding to sleep. During the active periods (*RU* and *SW*) every control packet is received and processed for updating the internal records (tables, SEC).

Upon T_{sw} timeout, if and only if the SEC = 1, a node broadcasts a sleep_beacon and sleeps for a certain time duration (T_s). Hello packets are never sent during sleep. If the SEC is 0, nodes switch back to *Awake_RU* and the process is repeated. TEAN-sleep is topology aware, since SEC changes dynamically with the neighborhood sleep states. Figure 7 depicts the operation of TEAN-sleep.

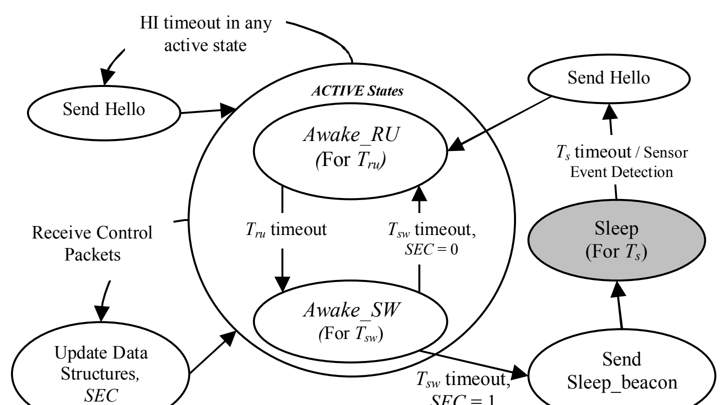

Figure 7. TEAN-sleep: State Machine

3) Energy Adaptive Sleep Duration

Energy aware sleep is essential for sleep fairness and creating a uniform energy distribution in the network. Energy critical nodes could be provided with more chances or long durations of sleep for energy balancing, and TEAN-sleep uses the later method, since more chances cannot be guaranteed in an unsynchronized coordination procedure. Sleep duration (T_s) determines the amount of energy conserved in a node, and is the key energy-balancing parameter of TEAN-sleep.

Through sleep, every node tries to converge to the average energy level of the neighborhood within a finite amount of time ($T_{converge}$), and this process could be mathematically represented as;

$$E_i - \lambda_A P_A T_{converge} = \bar{E} - \lambda_A \bar{P}_A T_{converge} \quad 1$$

$$\Rightarrow P_A = \bar{P}_A - \frac{\bar{E} - E_i}{\lambda_A T_{converge}} \quad 2$$

Where, $\bar{E}$ is the average neighborhood energy level (J), which can be estimated from the energy_cache. λ_A represents the node energy dissipation rate (J/s) in active state, and a maximum value could be assumed to make λ_A a constant. E_i is the current energy level of a node attempting to converge to $\bar{E}$.

P_A is the probability that a node is in active state (the fraction of time a node should be active within a duty cycle) i.e.,

$$P_A = \frac{T_A}{T_A + T_s} \quad 3$$

T_A is the average amount of active time within a duty cycle. In TEAN-sleep, nodes are required to listen for at least one HI before making sleep decisions, and so T_A can be set to one HI.

$\bar{P}_A$ is the average active fraction of the neighbors within a duty cycle. In general, nodes expect to sleep for at least the amount of time they listened, and so setting $T_A = T_s$ in 3 gives $\bar{P}_A = 0.5$ (50% average duty cycling). From 1, P_A represents the fraction of time, within a duty cycle, a node needs to be active so that E_i converges to $\bar{E}$ at $T_{converge}$. P_A can be computed dynamically from 2 before every sleep decision, and T_s could be adapted using 3 to satisfy the required P_A i.e.

$$T_s = R\,T_A \quad ;where \;\; R = \frac{1-P_A}{P_A}$$

In TEAN-sleep, the length of duty cycle varies based on the T_s. R represents the sleep magnitude and is energy adaptive. R increases with increasing energy criticality (low P_A), providing a proportionally higher T_s for sleep starved nodes. Since nodes are not guaranteed to sleep at any instant (due to connected network requirement), the active fraction of nodes randomly extend beyond the average duty cycle (50%), creating an energy imbalance, which emphasizes the need for energy aware sleeping.

$T_{converge}$ can be chosen based on the rate of convergence required by the application, but should be long enough to make P_A in the achievable range [0-100%]. For example, a very short $T_{converge}$ might require a negative P_A, which is logically impossible to achieve. This neighborhood energy balancing method would inherently propagate to the entire network (since neighborhoods overlap), and the performance evaluation section substantiates the efficiency of our design.

4) Key Design Issues

Hello packets are the major communication overhead of TEAN-sleep. A short HI provides quick convergence to topological changes and prevents prolonged network pathologies, but generates abundant overhead traffic. For static sensor networks (considered in this research), where channel errors and node dynamics (death/sleep) are the primary sources of topological changes, a long HI would suffice. However, a long HI implies longer listening periods (*Awake_RU*) and thereby increased energy consumption. HI is a key design factor, which should be selected to satisfy the application demands on efficiency-overhead tradeoff.

T_s is the main performance parameter of TEAN-sleep, which should amortize the energy cost of protocol overhead and further achieve energy savings. However, long T_s results in poor recovery from sudden network partitions, and a maximum limit on T_s could be imposed by the application for reliability. In this work, no such optimizations are investigated.

III. α-METRICS

The crucial requirement of energy-aware sensor protocols is not only to conserve energy but also to ensure that the QoS is not degraded. For example, TEAN-sleep strives to maximize network lifetime, while maintaining a connected network. Therefore, an integrated representation of energy efficiency and connection reliability would be appropriate to analyze TEAN-sleep, and *α-metrics* are designed with that intent.

Developing sensor-oriented metrics has been previously addressed by the sensor research community. [10] designed connected-coverage (CC), a popular term that implies a region is both sensed and connected to the user. Metrics that represent the application-centric performances are also significant for sensor networks. Authors in [11] researched the concept of α-lifetime, the lifetime when only a fraction α of the region is required to be covered at any instant. This work develops a family of α-metrics based on [10] and [11].

A. Definitions

1) Connected-Coverage (CC)

Any point p in the sensor region is *covered* if the Euclidean distance between p and at least one active sensor i is less than the sensing radius of i ($|pi| < R_s(i)$). The point p is *connected* to the user, if there exists a multi-hop path from the user to at least one active sensor covering p. p has CC if it is both covered and connected. CC would accurately measure the reliability of TEAN-sleep in maintaining a connected network.

2) α-lifetime

This research uses the same definition of *α-lifetime* as in [11], but with the coverage requirement replaced by CC in lifetime estimations.

α-lifetime is the total network lifetime (in time units) until only a percentage α (%) of the source region has CC at any time. Even if most of the sensor region is covered, if no information is communicated to the user (due to network partitions), the network is useless. Thus by considering CC, *α-lifetime* represents the "useful" lifetime of a WSN.

3) α-effectiveness

α-effectiveness is the total amount of information (bits) delivered to the user by a WSN until only α % of the source region is required to have CC at any time. It is the total information supplied until *α-lifetime*.

α-effectiveness is an integrated representation of lifetime improvement and seamless user information supply. For the same event rate scenario, high *α-effectiveness* not only means improved energy savings, but also implies consistent CC (monitoring reliability) throughout the lifetime.

B. Why α-effectiveness and α-analysis?

Extending network lifetime for prolonged monitoring is important for threat surveillance applications, and *α-lifetime* would seem adequate in defining the performance of TEAN-sleep. However, seamless CC (another key requirement) may not be truly reflected by *α-lifetime*. For example, in scenarios with very few sensor updates (hourly), the effects of intermittent CC would be negligible, since the rare data events would not require CC on a consistent basis. Disaster monitoring applications are also characterized by periods of heavy traffic (when a threat is detected), and so *α-effectiveness*, a measure of continual information supply, is essential and would ideally complement the *α-lifetime* analysis.

A clearly defined and consolidated depiction of energy, connectivity and coverage efficiency (as a single metric) reduces the quantity and complexity of performance representations, which simplifies, and thereby improves, reader understanding.

C. Estimating the α-metrics

To compute *α-metrics* we need to estimate the CC of a sensor region to the required α fraction and the amount of information contained in the sensor reports delivered to the user.

1) Estimating CC

The entire region is split into discrete points spaced at the required level of quantization (this work uses 1m spacing). The source reports “obtained at the sink” can be processed to identify the coverage status of every point at any instant (by employing the definition), from which the % covered could be computed. Regional coverage estimations are done using the reports obtained at the sink, which implies that the region is also connected to the user. In short, CC is implied if a point is covered, since coverage is estimated from the reports received at the sink. If any point is not reported continuously for certain duration (three HIs used here), it is declared to have lost CC.

2) Information Content in a Report

Correlation (in time and space) between two source reports is the rate of redundancy in the information supplied by them. In this work, we use the correlation model (based on lossless compression) for environmental sensor data developed by [12], since it closely matches our sensor applications. Employing their model for time and space domains, the spatio-temporal message redundancy rate between any two reports i and j can be represented as,

$$R_{s,t}(i,j) \;=\; \frac{1}{\frac{d_s}{c}+1} * \frac{1}{\frac{d_t}{\Delta}+1}$$

Where, d_s is the distance between the source nodes, d_t is the time interval between the reports, c and Δ are the space and time correlation constants respectively. $R_{s,t}$ lies in the range (0-1). If the spatial correlation between two reports is 0 ($d_s >> c$), then the overall redundancy rate is also 0 even if they are generated at the same time ($d_t = 0$) and vice-versa. The amount of uncorrelated data contained in a report i of size M bits is,

$$Information = \left(1 - \sum_k R_{s,t}(i,k)\right) * M \quad (bits)$$

where the summation represents the cumulated redundancy rate of i with k reports received before i until either the information supplied by i becomes 0 or the correlation between the reports becomes negligible ($d_s >> c$ or $d_t >> \Delta$). In simple terms, a report’s spatio-temporal correlation with the previous the reports is estimated and removed for computing the actual information supplied. The model employed here is not strictly based on, but agrees with the concepts of, information theory. The main purpose of this simple design is to aid in α-effectiveness estimation, and not to achieve data compression or aggregation, which is beyond the scope of this research.

IV. PERFORMANCE ANALYSIS

The complete TEAN-sleep model (figure 1) was developed in ns-2 [13] for the purpose of analysis and performance verification. Push diffusion [14] was chosen as the application framework, since it was exclusively developed for source-active sensor applications [14] (such as disaster monitoring). The popular 802.11 DCF [15] was used for channel access, instead of any sensor MAC protocols, to precisely quantify the contributions of TEAN sleep. Table 1 shows the power consumption data of Mica2 sensor technology [1], which was used for the node energy model (the magnitude of idle mote power consumption emphasizes the need for sleep)

Table 1. Power Consumption Model

Mote operating mode	**Power Consumption (mW)** (with typical 3V supply)
Active, Radio Transmit	117
Active, Radio Receive	57
Idle (Active, Radio Idle)	27.2
Sleep (Radio/CPU off)	< 3

A. Sleep Vs. No Sleep

Several previous researches ([7], [9]) have compared the performance of a sleep coordinated network against the operation without sleep, since sleep opens up the possibilities of network partitions. In no-sleep scenarios, reliable CC is often guaranteed throughout the network lifetime. Introducing sleep ensures energy conservation, but inefficient sleep coordination leads to intermittent CC, which would degrade the α-metric performance. For example, with TEAN-sleep, the overall network lifetime would surely be extended when compared to the lifetime without sleep, but the α-lifetime improvement depends entirely on the scheme’s efficiency in conserving energy, while maintaining a connected network (since α-metrics are estimated based on consistent CC). The α-metric comparison between sleep and no-sleep scenarios is essential for substantiating the contributions of TEAN-sleep.

B. Experimental Evaluation

TEAN-sleep was tested under three different surveillance scenarios, 1) Uniform Events 2) Localized Events and 3) Vehicle Tracking (Random Events), for exploring its performance under diverse traffic and energy-depletion patterns. Table 2 lists the common parameters and the scenario specific changes would be discussed in their individual sections. The results were obtained with 90% confidence intervals for the average of 10 simulations.

Table 2. General Simulation Parameters

Topography	1000m x 1000m square region
Number of nodes	100, uniformly distributed 1 sink (user) at the center of the square area, collecting reports
Node Sensing Range Node Radio range Generated Sensing Density	100m 250m (>2*Sensing Range [16]) ~2.5 nodes/sensing radius
Initial energy of nodes	20 Joules (User is wall-powered)
Report size	156 bytes
Channel bandwidth	1.6 Mb/s
Information Model Parameters	Time correlation constant = 60 sec Space correlation constant = half the sensing radius = 50 m
TEAN-Sleep Parameters	HI = 10 sec
Coverage Requirement (α)	Application specific. In this work, we used: α = 95%

1) Uniform Events

In this scenario, 40 nodes are capable of sensing and the rest are relay-only nodes. The sensing nodes provide reports to the user every minute and are deployed uniformly to cover the entire region. This model represents a typical protected area surveillance scenario, where updates are generated evenly throughout the sensor region. Figure 8 shows the performance of TEAN-sleep under such uniform energy depletion patterns.

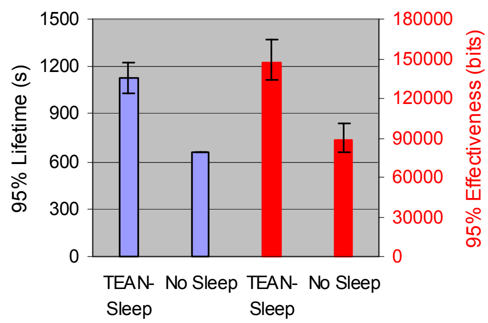

Figure 8. Performance of TEAN-sleep: Uniform Events Scenario

Intuitively, we would expect the information supply to increase proportionally with CC lifetime extension in static networks with periodic traffic (since time and space correlation would not vary drastically). However, packet losses and the rate at which updates are generated influence the data supply at the user. TEAN-sleep achieved 71% and 66% improvement in α lifetime and effectiveness respectively over the performance without sleep, which indicates significant energy savings with consistent CC provision.

2) Localized Events

The purpose of this experiment is to test the efficiency of TEAN-sleep in deterministic, non-uniform traffic scenarios. Three source nodes are isolated within 900x900m to 1000x1000m area of the square region (local source region), and report events at high rate (1/sec) to the user. This scenario mimics disaster periods, where events are localized to the disaster location (imagine a volcanic eruption generating abundant information flow to the user). Figure 9 shows the α-performance of TEAN-sleep, where improvements of 79% in lifetime and 70% in effectiveness were recorded.

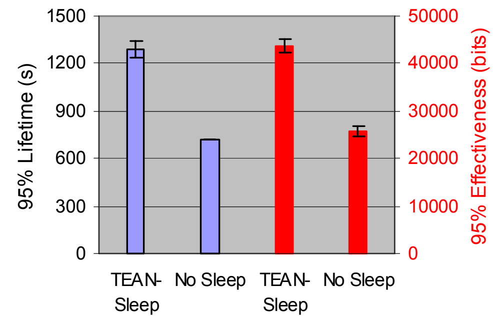

Figure 9. Performance of TEAN-sleep: Localized Events Scenario

TEAN-sleep's % improvement in non-uniform traffic case is marginally higher than the uniform-events case, since maintaining an energy balanced network in non-uniform traffic scenarios is significant to ensure adequate CC for longer durations. In localized events, without sleep coordination, the nodes between the disaster region and the sink (line-of-fire nodes) are prone to faster energy depletion, since diffusion would always choose the lowest latency paths. This energy imbalance degrades the CC % in no-sleep scenarios, escalating the contributions of TEAN-sleep.

To substantiate this claim, figure 10 shows the residual energy distribution of the sensor region, with and without sleep coordination, as a gray scale map. The color bar represents the energy values normalized to 1, where 1 represents the maximum energy level (darker regions are energy critical).

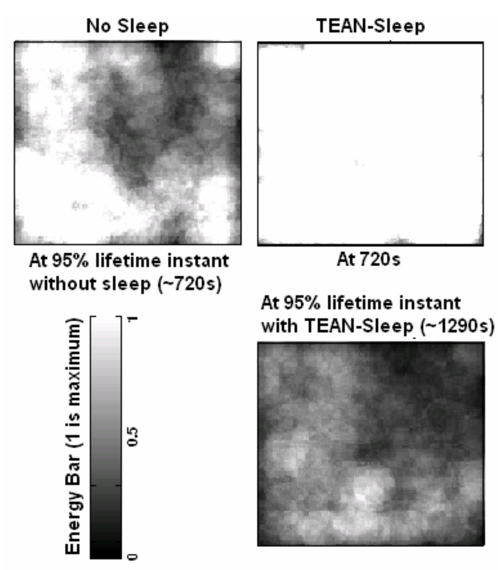

Figure 10. Network Residual Energy Distributions

As expected, without sleep, the line-of-fire nodes (between the center of square and right top corner) tend to die faster, creating localized energy critical regions. Such non-uniform energy depletion behavior decreases the "useful" lifetime of a WSN (due to CC % degradation). The color map of TEAN-sleep exhibits a reasonably energy balanced network even at the α-lifetime instant with sleep, ensuring extended periods of uniform CC. This clearly depicts the efficiency of our T_s design (II.*D.3*) in energy balancing the entire network[1] .

[1]Energy balancing in non-uniform traffic scenarios is more challenging and the performance can certainly be generalized to all cases.

3) Vehicle Tracking (Random Events)

Vehicle tracking is another surveillance pattern, where burst of events occur at random locations (non-uniform traffic in non-deterministic locations). In this experiment, a source node moves randomly at 20 miles/hr within the deployed region, and generates reports at high data rate (1/sec) to the user. This represents a scenario where an enemy vehicle is being tracked and the application demands location updates every second (traffic bursts from locations proximal to the vehicle). The efficiency of TEAN-sleep in maintaining a consistent CC throughout the network (effectiveness in reliably reporting random events) will be precisely quantified in this experiment.

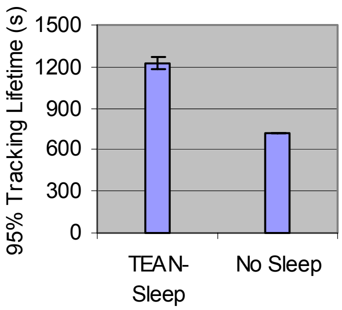

Figure 11. Vehicle Tracking

Figure 11 depicts the performance of TEAN-sleep in tracking applications. 95% tracking lifetime is the lifetime of a WSN until the probability with which a moving vehicle could be tracked reduces below 95% at any time.

Tracking fraction could be easily estimated from the total number of location updates obtained and the expected number of updates within a time quantum (e.g 1 minute). Non-uniform and intermittent CC would result in poor tracking capabilities and consequently would shorten the α-tracking lifetime. TEAN-sleep achieves 71% improvement in α-tracking lifetime, implying that our sleep management method conserves energy as well as maintains a reliable connected-covered network at all times.

V. Conclusions and Future Work

This paper introduced TEAN-sleep, a network level sleep management scheme for WSNs that asynchronously coordinated sleep operations to achieve energy savings, while guaranteeing close to 100% network CC at all times. Further, we adopted a family of well defined α-metrics that depicted the energy, coverage and connectivity efficiency of TEAN-sleep as a consolidated representation (instead of multiple graphs for each performance).

Adaptive sleep duration is the key performance parameter, and was designed to double the lifetime of the sleep deprived networks (50% average duty cycling). However, due to the protocol overhead and CC requirements, ideal conservation cannot be achieved. Performance testing under a variety of surveillance scenarios identified, on average, 70% improvement in network lifetime using TEAN-sleep under 95% CC requirements.

As per the SEC, the probability of node-sleep increases with increasing neighborhood density, and so low node densities would proportionally decrease the average energy savings in a node. Also, the performance of TEAN-sleep does not vary with varying CC demands (α), since the entire network dies gracefully (energy-balanced network). The results under varying node densities and CC demands were not presented due to space constraints.

Incorporating the TEAN-sleep layer into existing protocol stacks will require a small, but crucial implementation of a commanding interface for the higher layers (routing) to control sleep operations dynamically. Since TEAN-sleep is an independent entity, higher layers could use the command interface to pause sleep operations to avoid any service disruptions.

In this introduction work, the initial demands on TEAN-sleep (useful lifetime improvements with continual CC) were clearly validated. The next major step would be to compare the performances of TEAN-sleep with other topology coordination schemes ([7], [8]) operating under the uniform CC constraints.

Investigating the impact of sleep on reporting latency and reliability (prolonged sleep has poor recovery from sudden network partitions) would be useful in adapting the sleep duration based on network failure history.

Though TEAN-sleep is a free sleep scheme, it still alternates the listen-sleep operations as a cycle. Identifying the average duty cycle percentage of TEAN-Sleep in the overall network operation and comparing it with fixed duty cycling schemes [4] would quantify the actual benefits of asynchronous operation.

References

[1] Crossbow Technology Inc., http://www.xbow.com/

[2] A. Spyropoulos, C.S. Raghavendra, V.K. Prasanna, “A Distributed Algorithm for Waking-up in Heterogeneous Sensor Networks”, *Proceedings of Second International Workshop on IPSN*, Palo Alto, CA, April 2003.

[3] E. Shih, P. Bahl, M. Sinclair, “Wake on wireless: An event driven energy saving strategy for battery operated devices”, *Proceedings of ACM/IEEE International Conference on Mobicom*, Atlanta, GA, September 2002

[4] W. Ye, J. Heidemann, D. Estrin, “Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks”, Technical Report ISI-TR-567, USC/Information Sciences Institute, January 2003.

[5] Prabal Dutta et al., “Design of a Wireless Sensor Network Platform for Detecting Rare, Random and Ephemeral Events”, IPSN’05, Los Angeles, CA, April 2005.

[6] D. Lymberopoulos and A. Savvides, “A Motion-Enabled, Power Aware Sensor Platform for Distributed Sensor Network Applications”, IPSN 2005, Los Angeles, CA, April 2005.

[7] A. Cerpa, and D. Estrin, “ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies”, *IEEE Transactions on Mobile Computing* 3(3): 272-285, 2004

[8] B. Chen et al., “Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks”, *Proceedings of ACM Mobicom*, Rome, Italy, July, 2001.

[9] F. Koushanfar et al. “Low Power Coordination in Wireless Ad-hoc Networks”, *ACM ISLPED*, pp. 475-480, August 2003.

[10] X. Wang et al., “Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks”, *ACM SenSys*, LA, CA, November, 2003

[11] H.Zhang and J. Hou. “Maximizing α-lifetime for wireless sensor networks”, *Int. J. Sensor Networks, Vol. 1, Nos. 1/2, 2006*

[12] S. Pattem, B. Krishnamachari and R. Govindan, “The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks”, 3rd Symposium on IPSN, Berkeley, CA, April 2004.

[13] The Network Simulator. http://www.isi.edu/nsnam/ns/

[14] Silva.F, Heidemann.J, Govindan.R, Estrin.D, “Directed Diffusion”, USC/ISI Technical Report ISI-TR-2004-586.

[15] Giuseppe Bianchi, “Performance Analysis of IEEE 802.11 Distributed Coordinated Function”, *IEEE Journal on Selected Areas in Communications 18 (3),* pp. 535-547, March 2000.

[16] H. Zhang and J. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *Wireless Ad Hoc and Sensor Networks: International Journal*, vol. 1, num. 1-2, pp.89-123, January 2005.