Threat-Aware Clustering in Wireless Sensor Networks

Ryan E. Blace[1], Mohamed Eltoweissy[2] and Wael Abd-Almageed[3]

[1] BBN Technologies LLC, Columbia, MD, reblace@bbn.com
[2] Electrical and Computer Eng., Virginia Tech, toweissy@vt.edu
[3] UMIACS, University of Maryland, wamageed@umiacs.umd.edu

Abstract. Technological advances in miniaturization and wireless networking have enabled the utilization of distributed wireless sensor networks (WSN) in many applications. WSNs often use clustering as a means of achieving scalable and efficient communications. Cluster head nodes are of increased importance in these network topologies because they are both communication and coordination hubs. Much of the research into maximizing WSN longevity and efficiency focuses on dynamically clustering the network according to the residual energy contained within each node. This is a result of the commonly held assumption that battery depletion is the primary cause of node failure. In this work, we consider that there are applications in which threats may significantly impact node survival. In order to cope with these applications, we present a threat-aware clustering algorithm, extending the Hybrid Energy Efficient Distributed clustering algorithm (HEED) that minimizes the exposure of cluster heads to threats in the network environment. Simulation results indicate that our extended threat-aware HEED, or t-HEED, improves both the longevity and energy efficiency of a WSN while incurring minimal additional overhead. Our research demonstrates and motivates the need for a general framework for adaptive context-aware clustering in WSNs.

Keywords: context awareness, threat model, clustering, sensor networks

1. Introduction

Wireless sensor networks (WSN) are being employed in numerous environments and applications. Often, WSN implementations utilize clustering techniques as a method of achieving self-organization and scalability in their communication model [1-5]. WSNs are energy and resource constrained and their effectiveness and longevity is subject to that of their participant nodes.

Clustering techniques introduce heterogeneity to the service profile of the network, which has the side effect of creating nodes that can be considered 'more critical' than others. This is due to the fact that cluster head node serve as central hubs or super-peers for node management functions such as communications, organization, and security. In a clustered network, each sensor node forwards all communications toward the cluster head to which it is assigned. This can occur in one step or many steps, depending on the clustering implementation. It is the cluster head's job to route all received communications towards a destination or network sink. As a result, cluster heads are of increased importance to the proper functioning of the WSN. From this fact, it is apparent that the compromise or loss of a cluster head will have a greater impact on the overall effectiveness and longevity of the network.

One of the primary design goals for clustering algorithms in WSNs is to maximize the lifespan of cluster heads [6-8]. By maximizing cluster head lifespan, one can minimize the need for the rearrangement of clusters, which is an expensive process involving, at a minimum, a number of communications between a cluster head and a sensor. Most research into clustering algorithms makes the assumption that the primary factor influencing the longevity of sensors in a WSN is the residual energy of the node.[7]. In other words, the primary cause of node failure is a depleted battery.

Increasingly, sensor networks are being used in environments where energy constraints are not the only threat to nodes. For example, in battlefield contexts, enemies may be actively searching for and destroying sensors. In the wilderness, firefighters may use air-deployed sensor networks to track the movements of wildfires that may overwhelm and destroy some of the nodes. In each of these contexts, it is likely that a node may be destroyed due to contextual factors other than energy levels.

This work explores the potential for incorporating a contextual threat-based element into an existing clustering technique that emphasizes energy conservation. Our hypothesis is that threat-aware clustering enhances network longevity and energy efficiency. As part of the work, we define a threat model and simulate the initial distribution and clustering of a sensor network using a clustering technique based on the Hybrid Energy Efficient Distributed clustering algorithm (HEED) [6], and our own proposed algorithm, t-HEED, which is an extension of HEED that uses contextual threat awareness to minimize the threat level of cluster heads. The network is then attacked by adversaries who perform the attacks based on a pre-determined threat distribution. We measured node count, residual energy, and other metrics after a number of epochs to assess the effectiveness of the contextual threat enhanced HEED implementation.

The remainder of the paper is organized as follows. Section 2 summarizes related work. Section 3 presents an overview of HEED [6] and describes our threat-aware clustering approach. Section 4 presents our network and threat models. Section 5 reports on our comparison between HEED and our extension, t-HEED. Finally section 6 concludes the paper and highlights future work.

2. Related Work

Clustering has been demonstrated as an effective technique for achieving prolonged network lifetime and scalability in WSNs [11]. Parameters to include the node degree, transmission power, battery level, or processor load usually serve as metrics for choosing the optimal clustering structure. Recent initiatives address the problem of clustering and reclustering based on application specific attributes and network conditions. Bouhafs et al. [12] propose a semantic clustering algorithm for energy-efficient routing in WSNs. Nodes join the clusters depending on whether they satisfy a particular query inserted in the network. The output of the algorithm is called a semantic tree, which allows for layered data aggregation Siegemund [13] proposes a communication platform for smart objects that adapts the networking structure depending on the context. A cluster head node decides which nodes can join the cluster, based on similar symbolic location. Strohbach and Gellersen [14] propose an algorithm for grouping smart objects based on physical relationships. They use associations of the type “objects on the table” for constructing the clusters. A master node has to be able to detect the relationships for adding/deleting the nodes to/from the cluster. Perianu et al [15] propose a clustering scheme where the network is dynamic, the context is permanently changing and every pair of nodes is capable of understanding the physical relationships and thus the common context. Youssef et al [16] propose dynamic cluster head relocation based on a tradeoff between safety and performance. A cluster head moves closer to an event for enhanced performance while considering the threat level along the path between the cluster head and the monitored event. Up to our knowledge, our work in this paper is the first treatment of threat-aware clustering.

3. Threat-aware Clustering

To investigate the effectiveness of contextual threat-aware clustering, we modified HEED’s clustering algorithm to consider a node’s context when electing cluster heads. HEED is a distributed clustering algorithm that uses the residual energy of each node as a primary factor in deciding whether or not to become a cluster head. By introducing the threat weighting, our construction discourages nodes in high threat areas from becoming cluster heads. The next two subsections describe the HEED clustering algorithm and the modifications that were required to add context awareness.

3.1 HEED Clustering

The HEED clustering algorithm can be divided into three major steps: 1) Tentative cluster head distribution 2) Iterative CH election and balancing, and 3) Finalization and membership establishment. The algorithm is entirely distributed. All information must be transmitted between nodes, or known locally.

In the first step, each node decides whether or not to become a tentative cluster head based on a weighted probability of some ClusterHeadProbability*(Residual Energy/Max Energy). Essentially, each node has a fixed probability of becoming a cluster head, weighted by a dynamic measurement of the node’s current residual energy. When a node elects to become a tentative cluster head, it broadcasts that information to all nodes within communication range.

In the second step, each node goes through an iterative process of deciding whether or not to become a final cluster head based on the cost of the nodes within its communication range and its own cluster head probability. Each node doubles its probability of becoming a cluster head after each successive iteration in which no decision is made. If a node determines it is the optimal cluster head, it will elect to become a tentative cluster head. Once a node’s probability of becoming a cluster head has reached 1, it will assert itself as a final cluster head. Each node that is not ‘covered’ will repeat this process. A node becomes ‘covered’ when it is within communication range of either a final or tentative

cluster head. Any time a node changes state during this phase, the node broadcasts the state change to all neighbors within communication range.
During the final phase of HEED clustering, each node decides to join the least cost cluster head. HEED uses one of two cost functions for determining cluster head membership, least degree and most degree. The goal of least degree is to balance load across all clusters. The goal of most degree is to provide dense clusters.

3.2 Threat-Aware HEED (t-HEED)

At first glance, the contextual threat HEED implementation appears to make minor modifications to the underlying HEED algorithm. While subtle, the changes significantly affect the clustering process.

First, we altered the initialization phase to assert the initial node distribution according to both the residual energy ratio and the contextual threat level. For our purposes, we assume that the local contextual threat level of each node can be objectively determined by each node. The new formula for cluster head probability is expressed in Equation (1). Since we are adding a new weight to the initial cluster head probability function, it may be necessary to increase the baseline cluster head probability C_{prob}. $E_{residual}$ and E_{max} refer to the current battery level and the maximum battery level of the node. $Threat_{prob}$ is the context-based threat value that defines the distribution of potential attacks. P_{min} is a minimum value that is needed so that CH_{prob} never becomes 0

$$CH_{prob} = \max\left(C_{prob} * \frac{E_{residual}}{E_{\max}} * \left(1 - Threat_{prob}\right), p_{\min} \right) \qquad (1)$$

Second, we altered the cost function that is used in both the Repeat phase and the Finalize phase of the algorithm. The cost function is used to rank the neighbors of a node according to their suitability as cluster heads. HEED provides the guidance of using either 'node degree' or '1/node degree' depending on whether the goal is to create dense clusters or to distribute load. we replace this with a simple measurement of the threat level of a node.

The final modification to HEED involves its determination of when a node should stop repeating the middle phase. HEED specifies that a node should repeat until it is 'covered'. A node is considered 'covered' if it is within communication range of a final or tentative cluster head. A node will stop trying to process its local maximum cluster head candidate at this phase because the presence of a tentative or final cluster head implies that the maximum is already known. This implies that the node can simply accept one of the already existing cluster head candidates as its cluster head.

In order for t-HEED implementation to properly propagate contextual information and determine the optimal solution, we had to loosen the definition of 'covered' to include only those nodes that have a final cluster head within communication range. Eliminating the tentative cluster heads from the list allows the algorithm to converge on a more optimal solution rather than be subject to the initial CH distribution which is based solely on local information.

In summary, the contextual threat information is used during each phase in order to modify the probability that a given node will become a cluster head based on its threat level. This has the effect of moving cluster heads away from high threat areas.

4 Network and Threat Models

4.1 Network Model

The simulated network model, Figure 1, is a simple, grid arranged, statically placed, sensor network. The simulations were performed with a grid size of 30 by 30 units. Nodes are distributed throughout the grid based on a normal random distribution. The simulations were performed with a node density of 0.35. It is assumed that nodes have a fixed communication range, simulated as 5 units, and communicate reliably. Each node contains a battery that is modeled as a float value initialized to 1. Each communication incurs a battery cost on both sender and receiver, simulated as 0.005 of the max battery capacity.

The network model simulates a clustered organization and communication scheme, as shown in Figure 2. During network operation, each node will be a cluster head or a sensor. Each sensor must be a member of exactly one cluster, attached to the cluster head by a maximum of one step. After deployment and clustering, any sensor that cannot reach a cluster head and is not a cluster head itself must turn itself off. Cluster heads should be placed such that they are uniformly distributed throughout the topology unless intentionally otherwise arranged.

For this simulation, I assume that after the initial clustering process, there is no reclustering. I am not attempting to test the effectiveness of HEED, simply to investigate the effects of adding the contextual threat awareness. In order to mitigate the potential problems associated with not supporting reclustering, the network model does provide node reassignment.

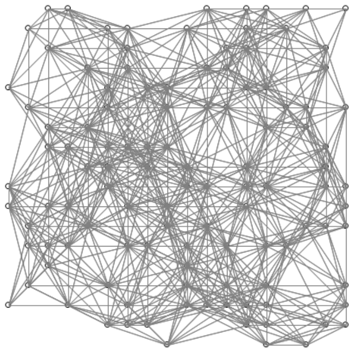

Fig. 1. Network Topology - initial distribution of nodes and node interconnectivity

The network model includes simulated traffic. Traffic is generated at a fixed rate and submitted to a random node in the network. If the source of the traffic is a cluster head, the traffic will incur a send cost on the CH. If the source of the traffic is a sensor node, a send cost is incurred on the sensor and a receive cost is incurred on the cluster head of the sensor. To clarify this point, the model simulates traffic that is routed from sensors to cluster heads. Cluster head to base station communication is ignored.

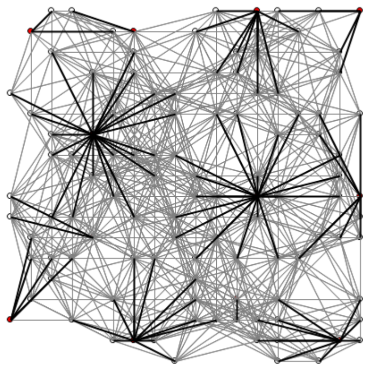

Fig. 2. Clustered organization of sensor nodes

When any node is killed either by a depleted battery or an adversary, it is turned off and effectively removed from the network. Sensor nodes that are killed have no effect on the other nodes in the network. Each cluster head that is killed has the effect of abandoning the nodes in its cluster. Abandoned nodes will attempt to reattach to other clusters within communication range. If no new cluster head is within range, the node will turn itself off.

4.2 Threat Model

The simulated threat model used against the network model can be characterized by a number of adversaries who travel fixed paths through the grid of nodes. Adversaries travel straight and consistent longitudinal and latitudinal paths through the network. Adversaries are generated at a fixed rate in bursts. At the start of the attack simulation, a number of adversaries are created and released. The adversaries travel their paths through the network, taking discrete steps of 0.1 units, until they have traveled across the network and traveled out the network boundaries. Once adversaries have left the network boundaries, the adversaries are destroyed. The attack simulation consistently maintains the number of adversaries until a specified max number of entities have traveled through the network. For the simulations performed in this work, adversaries were simulated at three different rates: 1 adversary at a time with a total of 10, 5 adversaries at a time with a total of 50, and 10 adversaries at a time with a total of 100.

When an adversary comes within a certain distance (simulated as 0.5 units) of a node in the network model, it probabilistically decides whether or not to attack the node. The probability is described in Equation (2). This probability ensures that an adversary is only going to attack a node once during its multi-step encounter with a node. The equation assumes that adversaries will travel directly over each node; however, the equation is a satisfactory way of maintaining the mode important probability: the attack success. The attacks of adversaries will succeed with a random probability that is according to the contextual threat level of the position that the adversary and node occupy. The probability is modeled in Equation (3).

$$P_{Attack} = \frac{1}{\left(\frac{\text{KillDistance} * 2}{StepSize} + 1\right)} \quad (2)$$

$$P_{Success} = P_{Attack} * Threat_{\Pr ob} \quad (3)$$

When an adversary successfully attacks a node, whether it is a sensor or a cluster head, the node in question is immediately killed and cannot be revived. Adversary attacks are modeled as independent events. A successful attack does not alter the behavior of the adversary, it simply continues on its path.

The contextual threat probability distribution is modeled as a matrix of threat values that is 2 units larger in both dimension than the network grid. The distribution has two primary features: points and lines. Initially, the distribution is uniform, with each element in the matrix having a threat level of 0. First, a specified number of random point-based threats are established in the distribution, boosting the threat level of small, symmetric areas throughout the network model. Second, a specified number of random lines are established along latitude and longitude lines of the distribution. This simulates shape-based contextual threat elements (i.e. Roads or buildings). Two specific sets of parameter values were considered for simulation: a point-centric and a line-centric threat distributions. Simulation tests indicated that the distributions produced consistent results. As a result, the tests were performed against the line-centric model.

5 Simulation and Results

The simulation is developed in C# using the .Net platform 3.0. The simulation has a visual component and a date model component. The visual component can be used to observe a simulation, as shown in Figures 3 and 4. Future efforts will likely include an implementation in TinyOS.

We ran extensive simulations using the network and threat models with the goal of gathering performance metrics from both the baseline HEED implementation and the extended threat-aware t-HEED implementation. The metrics are divided into two categories. The first category of metrics attempt to illustrate any effects the HEED modifications have on the behavior of the algorithm. These metrics include:

- The initial cluster head count
- The number of iterations required to converge on a clustering arrangement
- The average cluster head threat level

Ideally, the changes to HEED should not impact the initial cluster head count significantly. A significant change to cluster head count may bias other metrics and simulation results. The number of iterations required to converge on a cluster arrangement is important because it directly affects the time and energy required to perform clustering. Finally, the average cluster head threat level is an important metric that indicates how well the modifications are achieving their goal of minimizing cluster head threat exposure.

The second set of metrics capture the performance of the clustering technique by measuring the overall longevity and effectiveness of the WSN. These metrics include the number of nodes and cluster heads alive in the network, and the average residual energy of the nodes and cluster heads in the network. The number of nodes and cluster heads still alive in the network provide a simple indication of the remaining effectiveness of the network. The average residual energies of nodes and cluster heads in the network indicate the energy efficiency of the network and its potential longevity.

5.1 Clustering Metrics

Initial Cluster Head Count

Table 1 shows the initial cluster head count for two series of simulations. In the first series, all parameters were left at their regular levels and the density of the network was adjusted. In the second series, the transmission distance of each node in the network was adjusted. The results show that x HEED implementation created slightly fewer cluster heads in every test. The differences are marginal for the parameters that are closest to the defaults – Density 0.45 and transmission distance 6. An interesting observation to make is that the number of cluster heads increases with network density and decreases with transmission range.

Table 1: Initial Clusterhead Counts

t-HEED	**HEED**	
10.76	12.84	Density - 0.05
24.1	25.3	Density - 0.45
26.02	26.86	Density - 0.85
49.52	55.82	Trans – 3
16.58	16.84	Trans – 6
7.22	7.46	Trans – 10

Number of Iterations to Converge

Table 2 shows the number of iterations required to converge on a cluster arrangement for the same two series of simulations as for the initial cluster head count metric. In this case, t-HEED implementation took approximately double the iterations to converge than the baseline HEED implementation. Density does not appear to have significantly effected either implementation, and the transmission range seems to have an inverse relationship to convergence iterations.

Table 2: No. of Iterations before Convergence

t-HEED	HEED	
11.84	6	Density - .05

11.9	6	Density - .45
12	6	Density - .85
12	6	Trans – 3
11.6	6	Trans – 6
8.78	6	Trans – 10

Average Cluster Head Safety

This following table shows the average threat level of each cluster head in the WSN after clustering. The metric was measured for the same set of simulations as the other metrics. This is an important metric because it illustrates the effectiveness of the modified clustering algorithm at reducing the threat level of cluster heads. The numbers in the table are measurements of cluster head safety. The threat level is actually (1-x) where x is the value in the table. The data shows that t-HEED implementation is effective at reducing the threat level of cluster heads in the network.

Table 3: Average Threat Level for each Cluster

t-HEED	HEED	
0.6856	0.4555	Density - .05
0.7082	0.4622	Density - .45
0.7039	0.47	Density - .85
0.6472	0.4591	Trans – 3
0.7298	0.445	Trans – 6
0.7832	0.46	Trans – 10

Using the visualization tool, it is easy to see how the context extended implementation of HEED has performed Figure 4. The cluster heads have moved into the low threat areas and are connected to sensor nodes that are in the high threat areas.

5.2 WSN Performance Metrics

Each implementation was subject to a series of simulations that measured the sensor and cluster head longevity and residual energy. The simulations were performed with the default parameters, except for the adversary count, which was tested at 1 at a time for a total of 10, 5 at a time for a total of 50 and 10 at a time for a total of 100 (see the section on the threat model for details). Each simulation was run 100 times and all values for all metrics are the average values over the course of the trials.

Figure 5 shows the number of sensors that remain alive at a given epoch. As the simulation runs, and adversaries attack the network, the number of live sensors decreases as expected. Figure 6 shows the same metric, only for cluster head nodes. The legends are the same for both diagrams. As observed, the benefits that t-HEED introduces are greater as the number of adversaries increases.

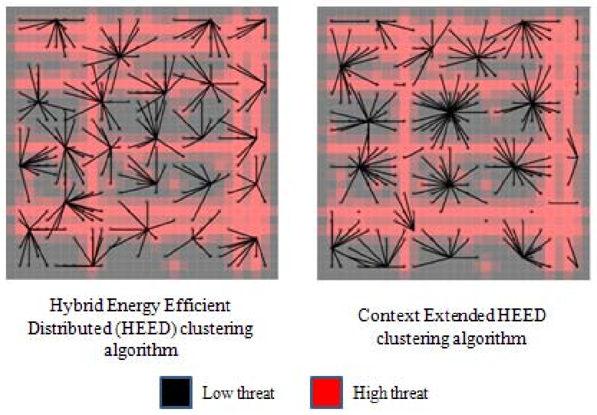
Hybrid Energy Efficient Distributed (HEED) clustering algorithm
Context Extended HEED clustering algorithm
Low threat
High threat

Fig. 4. Comparison between HEED and threat-extended t-HEED clustering

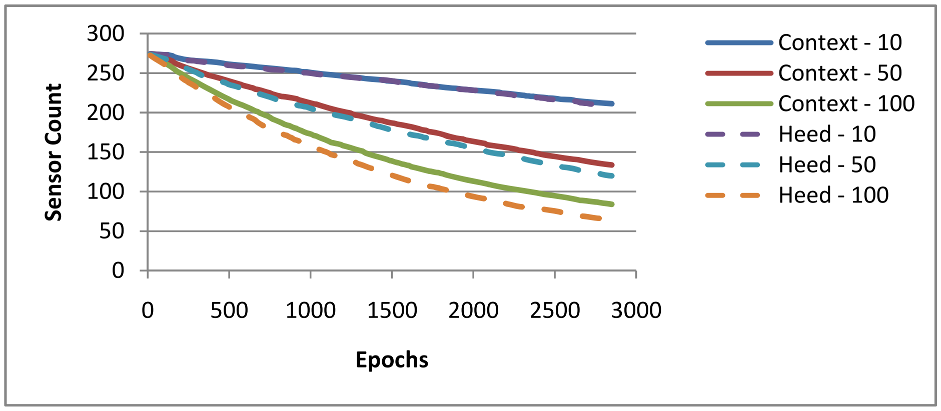

Fig.5. WSN Sensor Count

Figure 7 shows the residual energy for the same simulation. The source of the shift in the graph is uncertain, although it indicates that HEED uses more energy during the initial clustering phase than the context enhanced approach. Figure 8 shows the average residual energy for cluster head nodes in the network. The graph looks very similar to the alive node graph. This is due to the fact that the two metrics are closely correlated.

Additional simulations were performed with various adjusted parameters, including disabling node reattachment, disabling traffic generation, and tweaking node transmission range, network density, and initial cluster head probability. All of the results were consistent with those that have been presented. In every simulation, the context enhanced t-HEED implementation matched or outperformed the baseline HEED implementation.

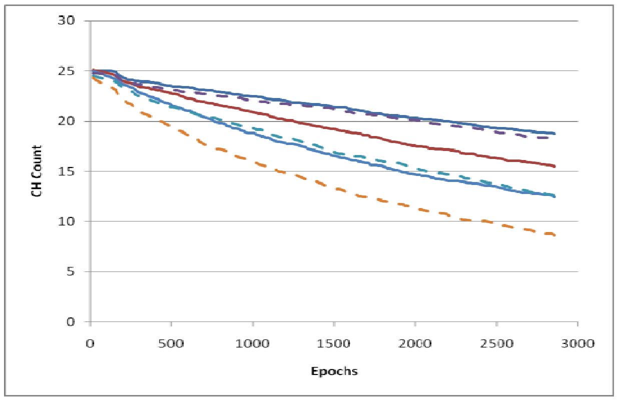

Fig. 6.WSN Cluster Head Count

6. Conclusion

We proposed threat-aware clustering to minimize the risk to cluster heads. The results show that using such an approach can increase the efficiency and longevity of a WSN without incurring an increased amount of overhead on the network. The primary conclusion that can be derived from this work is that context is a suitable parameter to be included in a clustering algorithm. Context need not be limited to a threat profile. The context awareness can extend to include expected locations of traffic generating events, or performance affecting factors like interference.

Future work will include extending our proposed framework to a general framework for adaptive clustering based on context changes. Also, implementation in an emulation environment such as Tiny OS and TOSSIM is a necessary future step in developing this work.

Acknowledgement

This work is supported in part by the National Science Foundation award No. 0721523.

References

[1] S. Selvakennedy, S. Sinnappan, and Y. Shang, "A biologically-inspired clustering protocol for wireless sensor networks." vol. 30: Butterworth-Heinemann, 2007, pp. 2786-2801.

[2] N. Vlajic and D. Xia, "Wireless Sensor Networks: To Cluster or Not To Cluster?," in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*: IEEE Computer Society, 2006.

[3] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks." vol. 30: Butterworth-Heinemann, 2007, pp. 2826-2841.

[4] C.-M. Liu, C.-H. Lee, and L.-C. Wang, "Distributed clustering algorithms for data-gathering in wireless mobile sensor networks." vol. 67: Academic Press, Inc., 2007, pp. 1187-1200.

[5] C.-Y. Wen and W. A. Sethares, "Automatic decentralized clustering for wireless sensor networks." vol. 5: Hindawi Publishing Corp., 2005, pp. 686-697.

[6] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," vol. 3, pp. 366-379, 2004.

[7] S. Yi, J. Heo, Y. Cho, and J. Hong, "PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks." vol. 30: Butterworth-Heinemann, 2007, pp. 2842-2852.

[8] W.-T. Su, K.-M. Chang, and Y.-H. Kuo, "eHIP: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks." vol. 51: Elsevier North-Holland, Inc., 2007, pp. 1151-1168.

[9] "NS2 Homepage," 2007.

[10] "Tiny OS Homepage," 2007.

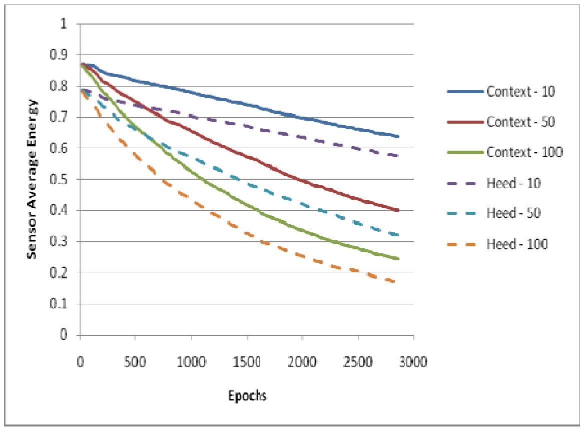

Fig. 7.WSN Sensor Node Average Energy

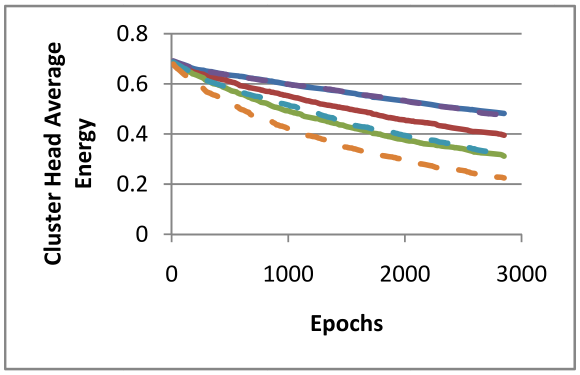

Fig. 8.WSN Cluster Head Average Energy

.