Fault-Tolerant Clustering Models for Wireless Sensor Networks

Akshaye Dhawan and Stephan Hennion
Ursinus College
Department of Mathematics and Computer Science
Collegeville, PA 19426
{adhawan, sthennion}@ursinus.edu

Abstract

Wireless Sensor Networks are an important focus of research in distributed computing due their many envisioned applications. Self-organizing these networks using clustering algorithms has been studied extensively in the literature as a means to conserve energy in sensor nodes. In clustering, the network organizes around a small set of cluster-heads which then gather data from their local cluster, aggregate this data and transmit it to the base station. In this paper we present two models for adding fault-tolerance to clustering algorithms. Since, sensor nodes are often deployed in harsh environments, they are prone to failure. Cluster-head failure can leave a cluster disconnected from the base station until the network reorganizes again. We present in this paper two models for adapting to the failure of a cluster-head. In the Inter-Cluster Recovery model, the failure of a cluster-head is handled by having the nodes in that cluster join an adjacent cluster. On the other hand, the Intra-Cluster Recovery model, replaces a failed cluster-head with another node in the same cluster. We implement our models for cluster-head failure detection and recovery over the LEACH clustering protocol and compare them to LEACH. We achieve fault-tolerance with a $7-10\%$ *reduction in lifetime as a trade-off.*

1 Introduction

Wireless Sensor Networks (WSNs) have attracted a lot of recent research interest due to their applicability in security, monitoring, disaster relief and environmental applications. WSNs consist of a number of low-cost sensors scattered in a geographical area of interest and connected by a wireless RF interface. Sensors gather information about the monitored area and send this information to an external node known as the base station. The radio on board these sensor nodes has limited range and allows the node to transmit over short distances. In most deployment scenarios, it is extremely expensive for each node to communicate directly to the sink and hence, the model of communication is to transmit over short distances to other peers.

In order to keep their cost low, sensors are equipped with limited energy and computational resources. The energy supply is typically in the form of a battery and once the battery is exhausted, the sensor is considered to be dead. Sensor nodes are also limited in terms of memory and processing capabilities. Hence, harnessing the potential of these networks involves tackling a myriad of different issues from algorithms for network operation, programming models, architecture and hardware to more traditional networking issues. For a more detailed survey on the various computational research aspects of Wireless Sensor Networks, see the survey papers [3, 5, 14, 17, 16].

In this paper we focus on the problem of self-organizing these networks into clusters [2]. The use of clustering based protocols is important in WSNs since clustering allows the sensor nodes to reduce their energy consumption significantly. Nodes organize themselves into local clusters with each cluster having one node designated as the cluster-head. Nodes then communicate to their local cluster-head which in turn transmits data from this cluster to the base station. This results in considerable energy savings since the transmission distance to the cluster-head is very small when compared to the distance to the base station. However, clustering does impose a significant drain on the battery of the cluster-head. Hence, protocols like LEACH (Low-Energy Adaptive Clustering Hierarchy) [13] rotate the cluster-head responsibility randomly between nodes so as to ensure fairness. Additionally, clustering allows the cluster-head to aggregate data before transmission, effectively reducing the amount of data being transmitted to the base station.

Our work in this paper focuses on making such clustering algorithms fault-tolerant. Since most clustering algorithms operate in rounds, where a cluster-head is elected and then used for the duration of the round, the failure of a cluster-head results in the nodes in that cluster being disconnected from base station for the remainder of the duration of this round. Failure is of particular concern in the context of WSNs since they are often deployed in harsh terrains resulting in a high likelihood of node failure. Our work attempts to devise models for the design of fault-tolerant clustering by using general principles that can be applied to any clustering protocol. We introduce two basic models - *Intra-Cluster* Recovery and *Inter-Cluster* Recovery. We then modify LEACH to test these models in Section 5 and examine the performance of these recovery algorithms for improved coverage.

The remainder of this paper is organized as follows. In Section 2 we briefly cover related work on the problem of clustering and fault-tolerance in Wireless Sensor Networks. In Section 3 we discuss modifications made to LEACH in order to allow nodes to detect the failure of a cluster-head. Section 4 presents two models for building fault-tolerance into a clustering algorithm. Section 5 studies the performance of these models by comparing them to standard LEACH. Finally, we present a summary of future work in Section 6 and conclude in Section 7.

2 Related Work

A key application of wireless sensor networks is the collection of data for reporting in various scenarios. Considerable research has gone into various areas of research with Wireless Sensor Networks. A common theme in current research has been to view these networks as large-scale distributed systems and study various problems with a perspective of examining common design techniques that emerge. For example, in our prior work [6, 7] we have presented a model for designing distributed solutions to the coverage problems in these networks that can also be applied to a host of other problems.

In Wireless Sensor Networks, there are two types of data reporting scenarios: event-driven and on-demand [4]. Event-driven reporting occurs when one or more sensor nodes detect an event and report it to the sink. In on-demand reporting, the sink initiates a query and the nodes respond with data to this query.

Clustering has been seen as an important means to conserve energy while routing data in WSNs. Several clustering algorithms have been proposed in the literature including randomized clustering of which LEACH [13] is an example, lowest cluster-ID clustering [11] and highest degree of connectivity [8] clustering. In practice, randomized clustering coupled with cluster-head rotation has been shown to be an effective approach to clustering [13]. See [2] for a survey on clustering algorithms for Wireless Sensor Networks.

Fault Tolerance has been extensively studied in the broader context of distributing computing [15], yet little work has been done on the specific problem of fault-tolerant clustering in Wireless Sensor Networks. [12] examines fault tolerance in clustering, but only looks at heterogeneous sensor networks where clustering is performed by special high-energy gateway nodes that are much more powerful than regular sensor nodes. We now turn our attention to the LEACH protocol and examine it in more detail.

LEACH (Low-Energy Adaptive Clustering Hierarchy) [13] is a self-organizing, adaptive clustering protocol based on randomization. The randomization allows LEACH to distribute the overhead of being a cluster-head evenly amongst the nodes in the sensor network. Additionally, LEACH uses data fusion to compress the data being sent to the base station, thereby reducing the energy wasted in transmitting redundant data. In LEACH, sensors elect themselves to be cluster-heads with a certain probability. This phase in known as the Advertisement Phase. These cluster-heads then broadcast their status to the other nodes. Nodes determine which cluster-head they belong to by looking at the strength of the signals received from the various cluster-heads.

Once all nodes are organized into clusters, each cluster-heads creates a Time Division Multiplexed (TDMA) schedule for each cluster. This allows the sensor nodes in the cluster to turn off their transceiver except at the time slot allocated to them, thereby saving considerable energy. A round is completed when every node has gone through its transmission. At this point, the cluster-head aggregates the data collected from the cluster and transmits it to the base station. The round is considered completed after this and the advertisement phase starts over again.

LEACH has emerged as the primary clustering protocol in WSNs and has been shown to be effective and energy-efficient. Hence, we chose to implement our extensions over LEACH.

3 Failure Detection

A key first step in devising a model for fault-tolerant clustering is that of detecting cluster-head failure. In this step, the nodes in a given cluster must be able to detect the failure of their cluster-head so that they can initiate the Failure Recovery phase outlined in Section 4.

Based on the existing setup of LEACH [13], a cluster-head only transmits data to its subordinate nodes during the advertisement phase (also known as the selection phase). During a round, once selection has been performed, the only nodes transmitting data are the subordinate nodes. As explained in Section 2, once the nodes organize themselves into clusters, each cluster-head creates a schedule for the nodes in its cluster. Therefore, in the current state of the LEACH protocol, a node in the cluster has no idea as to whether its cluster-head has failed, since it will never hear from the cluster-head again. Thus, the failure of a cluster head effectively disconnects the entire cluster for the remainder of the round. Also, all transmissions made by nodes post the cluster-heads failure are lost since they never get sent to the base-station.

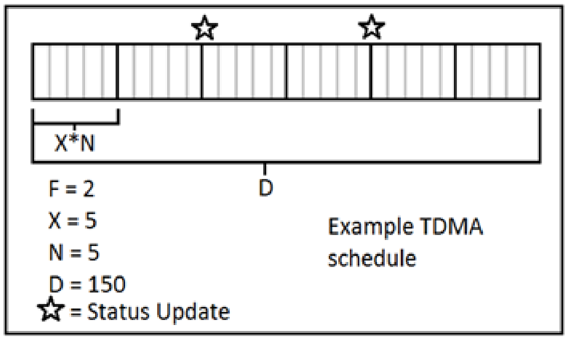

Figure 1. Failure Detection messages added to the TDM schedule

In order to implement detection of failure, we modify LEACH to add some communication from the cluster-head to the nodes in the cluster. Consider the following situation - let a round begin at time t_1. Let the next round begin at time $t_1 + d$ where, d is the duration of a single round. Note that so far, this representation is exactly how a round works in LEACH. Now, suppose that for the cluster in question, there are n nodes that are a part of this cluster, each of which transmits for x seconds in the schedule for this cluster. Then, for this cluster there are $s = \frac{\frac{d}{x}}{n}$, sets of transmissions between the nodes and the cluster-head. We implement detection by having the cluster-head pick a random number f between 1 and s, where s is as defined above. After every f set of transmissions (i.e. at time $f * x * n$), all nodes turn their radio receiver back on and the cluster-head sends a small ping message indicating that it is up and running. If such a transmission is not received, the nodes can assume that the cluster-head has failed and employ the recovery algorithm.

Figure 1 illustrates the status update messages into an existing schedule for the given f, x, n and d values.

4 Failure Recovery

In this section we present two models for recovery from cluster-head failure. We call these models *Inter-Cluster* Recovery and *Intra-Cluster* Recovery.

4.1 Inter-Cluster Recovery

The basis of the Inter-Cluster Recovery model lies in the theory that if k-fault tolerance is desired (i.e., the ability to handle the failure of up to k nodes is needed), then the model should build this into the advertisement phase. In the basic form of LEACH, each node attaches itself to a single cluster-head during the selection phase of each round. Each node makes this determination by examining the signal-strength of the received signal from the various cluster-heads, and picking the strongest single, since this is the cluster-head closest to a given node. In this scenario, if the cluster-head to which a node is attached fails, the nodes in the cluster are left transmitting to no one until the next round begins with a whole new selection process.

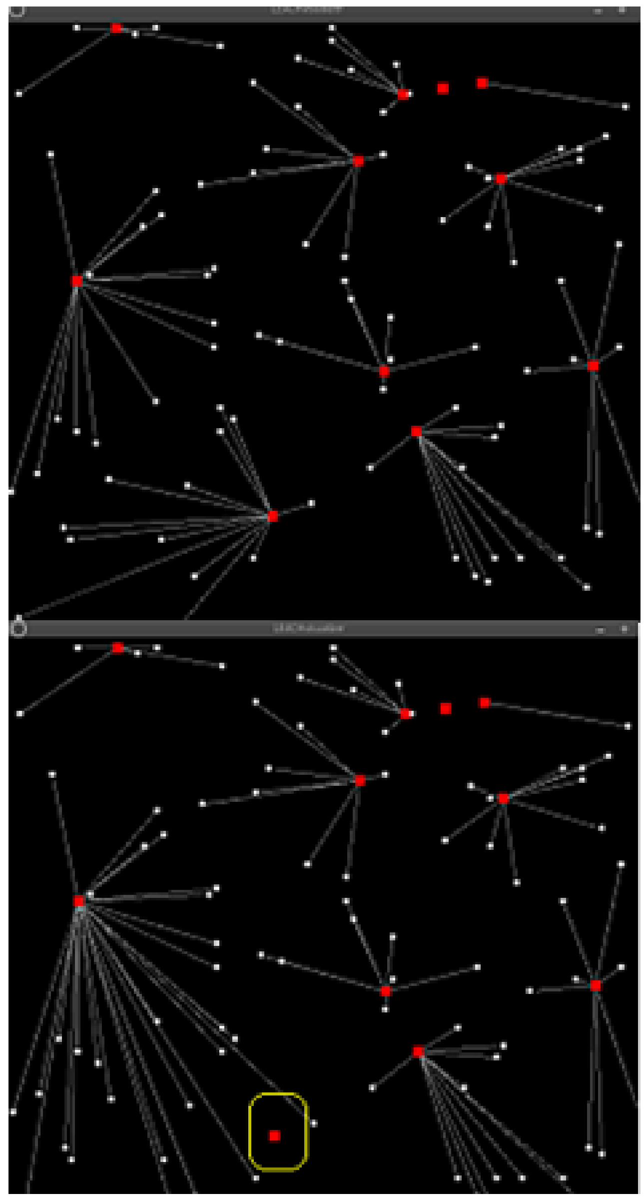

Figure 2. Inter-Cluster Recovery

In the Inter-Cluster Recovery model, we have k-backup cluster heads assigned to each node, where k represents the degree of fault-tolerance needed. The selection round is modified to have each node select k cluster-heads by order of their signal strength. A node still uses its closest cluster-head but can move to a backup cluster-head in the event of failure. However, simply picking k alternative cluster-heads to serve as a backup is not sufficient since these nodes must now accommodate room in their TDMA schedules for the nodes they are serving backup to. Without this, when a mid-round failure detected, the nodes making this detection cannot switch to their backups since those nodes will already be in the middle of a round.

Hence, during the selection phase each node transmits its list of k cluster-heads. The cluster-heads in turn allocate slots in their TDMA schedule for each node. Since each cluster-head uses different frequencies to prevent interference from adjacent clusters, each node keeps a table of the transmission time in the schedule and the frequency for each of its k backup cluster-heads. When the transmission phase of a round begins, a node transmits according the schedule of its closest cluster-head. Upon detection of the failure of this cluster-head, a node changes its transmission time slot and frequency to that of the cluster-head with the second strongest signal amongst the remaining $k-1$ cluster-heads. Successive failures can be handled in this manner until a node runs out of backup cluster-heads.

In this model, there is a trade-off for the increased fault tolerance. Since every cluster-head serving as a backup makes room for all the nodes it serves in its schedule, there are a number of unused bandwidth slots in each round. However, this costs minimal additional energy since all nodes except for cluster-heads have their radios switched off when it is not their turn to transmit. Hence, there is a small increase in the energy cost for a cluster-head since it now has a longer round to deal with. Also, if a node is forced to move to a backup cluster-head it must now spend more energy communicating with this cluster-head since it is further away than the failed cluster-head. We show in Section 5 that this is an effective model for small values of k.

4.2 Intra-Cluster Recovery

The Intra-Cluster Recovery protocol is much simpler than the Inter-Cluster protocol. The idea here is to have a failed cluster-head seamlessly replaced by the next high energy node in the same cluster. This prevents the need for a network-wide re-clustering round and instead contains the changes to the cluster in which failure has occurred. In our implementation of this model, when a cluster-head is picked for a given cluster, up to k nodes in the cluster randomly volunteer as backup cluster heads. These nodes are then sorted by remaining battery life and assigned a rank. When the cluster-head creates a transmission schedule for the cluster, this schedule is shared with all nodes. Hence, there is no overhead in transmissions needed to inform the backup nodes of the schedule. When a cluster-head failure is detected by means of a missed ping message, the node with the next highest rank takes over as the cluster-head. The assumption here is that all nodes in the same cluster are within communication range. If this assumption is true, nodes not serving as a backup are not impacted by the failure since they broadcast their data at the same point in the TDMA schedule as before.

Note that in this case, there is no overhead of wasted bandwidth since there are no unused slots reserved in the TDMA schedule. However, any data collected at the original cluster-head will be lost unless both the cluster-head and its backup are aggregating information through out the duration of the round. This model makes use of the broadcast nature of the wireless medium to achieve its goals with minimum overheads. As expected, we see in Section 5 that for the same value of k, there is a smaller energy overhead of Intra-Cluster Recovery when compared to Inter-Cluster Recovery.

5 Simulations

In order to study the performance of the propose fault-tolerance models, we have conducted preliminary simulations in $C++$. We implement LEACH as described in [13] and then modify this implementation to include both Inter-Cluster Recovery and Intra-Cluster Recovery as described in Section 4. Additionally, we implemented an visualizer to capture the clustering patterns of the network and represent these in a 2-d plane.

For our simulation setup, we experiment with networks of 100 sensors randomly placed in $100x100$ area. We repeat all simulations presented in this section over 30 iterations of randomly positioned sensor nodes. Since one of the goals of this simulation study was to examine the improved uptime of the network in the face of cluster-head failures, we place 10 targets randomly in this space and measure as a percentage of the lifetime, how long these targets are covered.

For the purpose of this simulation, we considered homogeneous nodes with identical batteries. The simulations also employ a linear power model where the energy required to transmit over a given distance is a linear function of the distance. In every round, 5% of the nodes volunteer to be cluster-heads. Node failure is injected randomly into the network. We limit each round to a maximum of 5 transmissions and we limit the simulation to a maximum of a 150 rounds. The energy costs are modeled along those of real sensors with idle energy costs being comparable to receiving costs and transmission costs being approximately double that of reception [9, 10].

We now examine the results of our simulations. Note that we implement fault tolerance for $k = 3$ in this study. In all instances of the modified Inter-Cluster and Intra-Cluster models, the failure of cluster-heads was detected and the network reorganized immediately upon this detection. We show in Figure 2, an example of the operation of the Inter-Cluster Recovery Model before and after node failure. The top frame shows the network before failure. Cluster-heads are shown in red while other sensor nodes are shown in white. An edge connects a given sensor node to its cluster-head. In this network, the cluster-head circled in yellow in the bottom frame fails. As can be seen from the figure, every node in the cluster joins the cluster-head above upon detection of failure.

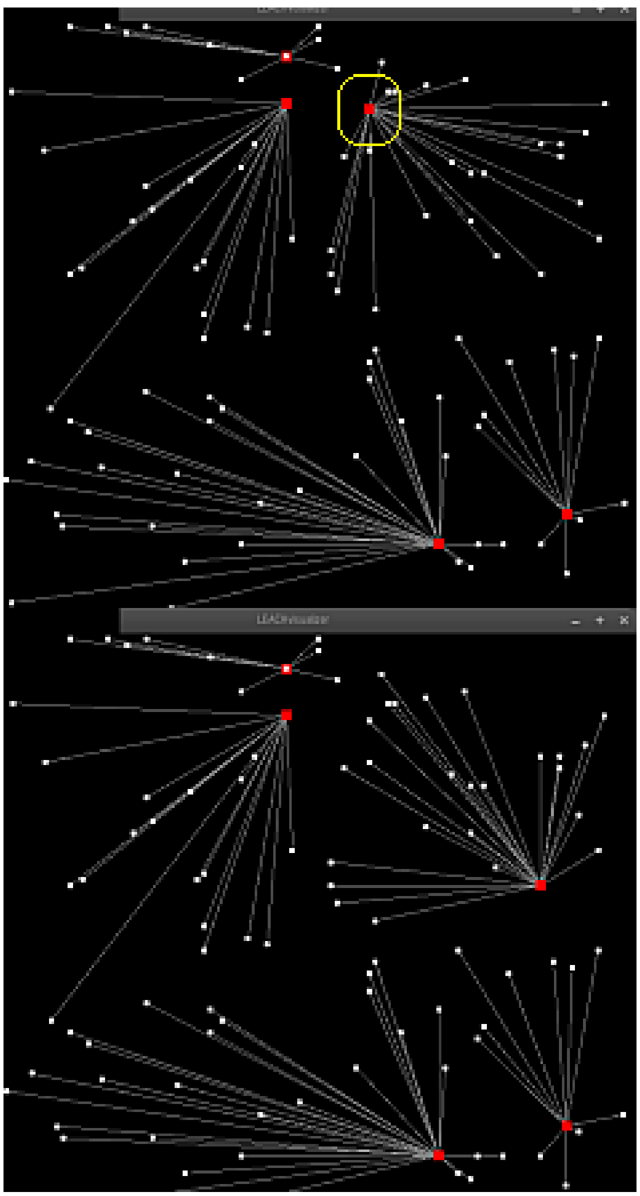

Figure 3. Intra-Cluster Recovery

In Figure 3, we see an example of the operation of the Intra-Cluster Recovery model. In this case, the cluster-head circled in yellow in the top-frame fails. Since the cluster-head will be replaced by another sensor-node in the same cluster, upon detection, the protocol picks at random a different node (shown in red in the frame below) and all nodes in the cluster now start using this new node as their cluster-head. The TDM schedule is left unmodified from before.

We next examine the number of rounds as a measure of the lifetime of the network. For a given network, we compare the number of operational rounds achieved by LEACH and compare this to LEACH with Inter-Cluster and Intra-Cluster Recovery. This enables us to quantify the trade-off in terms of the lifetime reduction caused by the implementation of fault-tolerance. We plot this data for the thirty randomly generated network iterations in Figure 4.

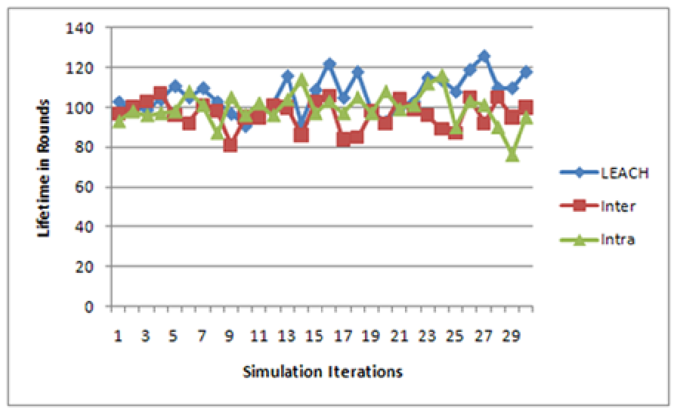

Figure 4. Lifetime measured in rounds

As can be seen from the figure, both Inter-Cluster Recovery and Intra-Cluster Recovery perform well with a very small reduction in lifetime. On an average, Inter-Cluster Recovery has a reduction in lifetime of about 9% and Intra-Cluster Recovery has a reduction of lifetime of 6% when compared to standard LEACH. The additional cost for Inter-Cluster Recovery is due to the additional uptime for cluster-heads caused by longer schedules. Hence, we see that fault-tolerance is achieved with minimal loss in lifetime.

When the coverage of the 10 randomly scattered targets is measured across the three protocols, Inter and Intra-Cluster Recovery achieve on an average a 5% increase in coverage. It is worth pointing out that standard LEACH in the face of failures does not provide constant coverage. It spends more time with less coverage sporadically resulting in significant lapses of coverage in some cases (based on which nodes failed).

6 Future Work

These preliminary simulation results show that increased coverage and fault-tolerance can be achieved at the cost of a small reduction in the lifetime of the network. In applications where coverage is critical, this is a viable trade-off to make. As part of our future work, we would like to conduct more extensive simulations. For example, we would like to repeat these simulations with a varying number of targets to be covered being taken into consideration. We would also like to study the lifetime against varying percentages of node failures (as opposed to the random failure model considered here). Additionally, we would like to experiment with variations in the random time that the ping message is sent by the cluster-head for failure detection as described in Section 3.

Also, moving our implementation to a network simulator like $ns2$ [1] would allow us to experiment with varying energy models and keep track of more detailed node and traffic data.

7 Conclusions

In this paper, we examined two models for recovery from cluster-head failure in Wireless Sensor Networks. The models are applicable to any clustering algorithm with minor modifications, even though they are presented in this paper for the LEACH clustering protocol. We show through simulations that these models achieve significant improvement in coverage and fault-tolerance with a minimal trade-off in terms of reduced network lifetime.

References

[1] The network simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[2] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.*, 30:2826–2841, October 2007.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Commun. Mag.*, pages 102–114, 2002.

[4] J. Carle and D. Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *Computer*, 37(2):40–46, 2004.

[5] C.-Y. Chong and S. Kumar. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug. 2003.

[6] A. Dhawan and S. K. Prasad. A distributed algorithmic framework for coverage problems in wireless sensor networks. *Procs. Intl. Parallel and Dist. Processing Symp. Workshops (IPDPS), Workshop on Advances in Parallel and Distributed Computational Models (APDCM)*, pages 1–8, 2008.

[7] A. Dhawan and S. K. Prasad. A distributed algorithmic framework for coverage problems in wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems(IJPEDS)*, 2009.

[8] J. B. Dugan and K. S. Trivedi. Coverage modeling for dependability analysis of fault-tolerant systems. *IEEE Trans. Comput.*, 38:775–787, June 1989.

[9] L. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1548–1557 vol.3, 2001.

[10] L. M. Feeney. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mob. Netw. Appl.*, 6(3):239–249, 2001.

[11] M. Gerla and J. Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless Networks*, 1:255–265, 1995. 10.1007/BF01200845.

[12] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. *IEEE Wireless Communications and Networking, 2003*, pages 1579–1584, 2003.

[13] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.

[14] S. S. Iyengar and R. Brooks. Computing and communications in distributed sensor networks. *Special Issue, Jr. of Parallel and Distributed Computing*, 64(7), 2004.

[15] D. K. Pradhan, editor. *Fault-tolerant computer system design*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[16] K. Römer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.

[17] S. Schmid and R. Wattenhofer. Algorithmic models for sensor networks. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 11 pp.–, April 2006.