FTHC: Fault Tolerance in Hierarchical Clustering Environment for WSN

Swati Sharma
Computer Science and Engineering Department
Maharishi Markandeshwar University, Mullana, Ambala

Rohit Vaid
Computer Science and Engineering Department
Maharishi Markandeshwar University, Mullana, Ambala

ABSTRACT

Wireless Sensor Networks (WSNs) are an important focus of research due to their many envisioned applications. They are formed by small, inexpensive and resource limited devices that can interact with the environment and communicate in a wireless manner with other devices. For energy conservation, the clustering technique is used where 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. Here we present two models for adding fault-tolerance to clustering algorithms with a hierarchy maintained among various levels of cluster heads from base station. 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. The proposed model 'FTHC: Fault Tolerance in Hierarchical Clustering Environment for WSN' is used for both Inter and Intra clustering environment. We evaluate the proposed model and compare it with protocol MECH in terms of network lifetime when the cluster head fail.

Keywords

Wireless sensor networks, radio model hierarchical clustering, cluster head failure, fault tolerance.

1. INTRODUCTION

A wireless sensor network (WSN) consists of a number of autonomous sensors to monitor physical or environmental conditions [9], such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. The development of wireless sensor networks was motivated by military applications [10, 11] such as battlefield surveillance; today such networks are used in many industrial and consumer applications also. Recently, research from the networking community as well as advances in micro-fabrication technology have brought about the realization of practical commercial wireless sensor networks, as can be seen in [3, 5, 6, 7, 8]. Since the nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attack, and so on as in [14, 15]. The nodes in sensor networks have very limited energy and their batteries cannot usually be recharged or replaced due to hostile or hazardous environments. So, one important characteristic of sensor networks is the power budget of wireless sensor nodes. Two components of a sensor node, sensing unit and wireless transceiver, usually directly interact with the environment which is subject to variety of physical, chemical, and biological factors. It results in low reliability of performance of sensor nodes. Even if condition of the hardware is good, the communication between sensor nodes are affected by many factors, such as signal strength, antenna angle, obstacles, weather conditions, interference. Fault tolerance is the ability of a system to deliver a desired level of functionality in the presence of faults as in [10, 12, 13]. Since the sensor nodes are prone to failure, fault tolerance should be seriously considered in many sensor network applications. Actually, extensive work has been done on fault tolerance and it has been one of the most important topics in WSNs.

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. Our work focuses on two models for adapting to the failure of a cluster-head. In the Intra-Cluster Recovery model, the failure of the cluster head replaces a failed cluster head with another node in the same cluster. In the Inter cluster, the new cluster head takes the responsibilities and the whole network works in the same manner without any interruption due to cluster head failure. We implement our models for cluster-head failure detection and recovery over the clustering protocol MECH and then modify MECH to test these models 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 our proposed system model for building fault tolerance into a clustering algorithm. Finally, we present conclusion in Section 4.

2. RELATED WORK

W. Heinzelman, A. Chandrakasan, and H. Balakrishnan in [1] proposes a self-organizing, adaptive clustering protocol that uses randomization to distribute the energy load evenly among the sensors in the network as in [9]. In [1], the nodes organize themselves into local clusters, with one node acting as the local base station or cluster-head. If the cluster heads were chosen a priori and fixed throughout the system lifetime, as in conventional clustering algorithms, it is easy to see that the unlucky sensors chosen to be cluster-heads would die quickly, ending the useful lifetime of all nodes belonging to those clusters. Thus it includes randomized rotation of the high-energy cluster-head position such that it rotates among the various sensors in order to not drain the battery of a single sensor. In addition, it performs local data fusion to "compress" the amount of data being sent from the clusters to the base station, further reducing energy dissipation and enhancing system lifetime. Sensors elect themselves to be local cluster-heads at any given time with a certain probability. These cluster head nodes broadcast their status to the other sensors in the network. Each sensor node determines to which cluster it wants to belong by choosing the cluster-head that requires the minimum communication energy. Once all the nodes are organized into clusters, each cluster-head creates a schedule for the nodes in its cluster. This allows the radio components of each non-cluster-head node to be turned off at all times except during its transmit time, thus minimizing the energy dissipated in the individual sensors. Once the cluster head has all the data from the nodes in its cluster, the cluster-head node aggregates the data and then transmits the compressed data to the base station. Since the base station is far away in the scenario we are examining, this is a high energy transmission. However, since there are only a few cluster heads, this only affects a small number of nodes.

S. Lindesy and C. Raghavendra in [2] proposes an idea for each node to receive from and transmit to close neighbours and take turns being the leader for transmission to the BS. This approach will distribute the energy load evenly among the sensor nodes in the network. We initially place the nodes randomly in the play field, and therefore, the i –th node is at a random location. The nodes will be organized to form a chain, which can either be accomplished by the sensor nodes themselves using a greedy algorithm starting from some node. Alternatively, the Base station can compute this chain and broadcast it to all the sensor nodes. To construct the chain, we start with the furthest node from the BS. We begin with this node in order to make sure that nodes farther from the BS have close neighbors, as in the greedy algorithm the neighbor distances will increase gradually since nodes already on the chain cannot be revisited. When a node dies, the chain is reconstructed in the same manner to bypass the dead node. For gathering data in each round, each node receives data from one neighbor, fuses with its own data, and transmits to the other neighbor on the chain. Nodes take turns transmitting to the BS, and we will use node number *i* mod N (N represents the number of nodes) to transmit to the BS in round *i*. Thus, the leader in each round of communication will be at a random position on the chain, which is important for nodes to die at random locations. The cost is very small since the token size is very small.

A. Manjeshwar and D. Agrawal in [3] give protocol that is targeted at reactive networks and is the first protocol developed for reactive networks. In this scheme (Threshold sensitive Energy Efficient sensor Network protocol), at every cluster change time, in addition to the attributes, the cluster-head broadcasts to its members, hard threshold (HT): This is a threshold value for the sensed attribute. It is the absolute value of the attribute beyond which, the node sensing this value must switch on its transmitter and report to its cluster head. Soft Threshold (ST): This is a small change in the value of the sensed attribute which triggers the node to switch on its transmitter and transmit. The nodes sense their environment continuously. The first time a parameter from the attribute set reaches its hard threshold value, the node switches on its transmitter and sends the sensed data. The sensed value is stored in an internal variable in the node, called the sensed value (SV). The nodes will next transmit data in the current cluster period, only when both the following conditions are true:

- The current value of the sensed attribute is greater than the hard threshold.
- The current value of the sensed attribute differs from SV by an amount equal to or greater than the soft threshold.

Next scheme in [4] is based on radio model like LEACH is based on. The approach is divided into rounds and each round consists of three phases: set-up phase, steady phase and forwarding phase. However, it has some drawbacks for sensor networks. First, it didn't consider the distribution of nodes. The number of nodes in each cluster is distributed unequally. After cluster-head gathers all the data from cluster members, it forwards the data to the base station directly. For cluster-heads that are far away from the base station, clearly direct transmission is bad for energy conservation. To improve the above drawbacks in LEACH, the cluster constructing method has to be changed. The communication between cluster-heads and base station also needs to be modified. MECH is mainly aimed at these two drawbacks. Cluster constructing method avoids the uneven member distribution for clusters. The hierarchical routing scheme avoids the long range direct communication between cluster-heads and the base station. Ruay-Shiung Chang and Chia-Jou Kuo in [4] use two parameters to establish the hierarchical relation among clusters: hop_count and energy level. However, there are a few drawbacks also. First, control messages are more than those in [3], because to get more information to construct more evenly distributed topology and the hierarchical routing tree. The cluster-head in protocol [3] transmits data to base station directly. The cluster heads in MECH use hierarchical routing to forward the data to the base station. Second, the synchronization mechanism may be expensive in hardware equipments for sensor nodes.

3. PROPOSED SYSTEM MODEL

3.1 Fault Tolerance in Hierarchical Clustering Environment

FTHC is based on protocol [4]. Our approach is divided into rounds and each round consists of five phases: set-up phase, steady phase, forwarding phase, CH failure, and recovery phase as shown in Fig. 1. We construct clusters before each round. Every node broadcasts a hello message to its neighbors. The Time to live of this hello message is set to one since we only need to gather neighbors of one hop. The radio range is also set to a certain transmission range. Each node records the number of neighbors. When the number of neighbors of a node reaches N, the node will broadcast an advertisement to its one hop neighbors informing it's the head of cluster. All the nodes who receive this advertisement record it. Furthermore, such nodes never broadcast the advertisement even if their number of neighbors reaches N. In order to synchronize, each cluster-head counts the total number of its cluster members and broadcasts the information to the base station. The base station computes the maximum number of time slots needed and broadcasts this information back to each cluster-head.

Setup Phase	Steady Phase	Forwarding Phase	Failure detection Phase	Recovery Phase

Fig. 1 Working of the Scheme FTHC

3.1.1 Setup phase and Steady phase

Because the maximum number of cluster members is bounded by N, the cluster-head can schedule the TDMA time slot for each cluster member at each round. In the set-up phase, every node will turn on the receiver such as in [4]. Then, the cluster-head will broadcast an advertisement that contains the TDMA time slot information. Each cluster member will know the time slot which belongs to it. Thus, the cluster member will keep the transceiver off until its time slot and will transmit the sensed data in its allotted time slot in the steady phase. The algorithm followed is:

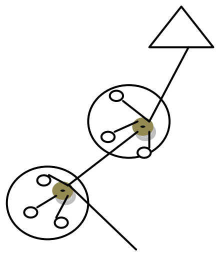
```
Step 1    Start
Step 2    For each node $N_1$ , Send “Hello” message
          to all nodes $N_2$....$N_n$ .
Step 3    For each node $N_2$....$N_n$ ,Send reply to $N_1$
          with distance.
Step 4    If neighbour count=CN then $_1$ sends
          message “I m head” to $N_2$....$N_n$.
Step 5    All nodes $N_2$....$N_n$ who receive message,
          Sends reply to CH.
Step 6    For each node in the cluster, CH receives
          data (sensing data + battery left).
Step 7    Chooses node with max left battery for
          next setup round.
Step 8    For each node N in cluster, Send data to
          CH in respective time slots.
Step 9    Stop
```

3.1.2 Forwarding phase

FTHC uses two parameters to establish the hierarchical relation among clusters: hop_num and Energy_msg. The following steps describe how the FTHC implements the hierarchy.

```
Step 1    Start
Step 2    Initialise hop_num=0,Energy_msg=100
Step 3    BS sends message to all CHs containing
          hop_num & Energy_msg
          1.  For each CH receiving msg
              hop_num=hop_num+1,
          2.  If Energy _msg > Energy_CH, then
              Energy_CH=Energy_msg
Step 4    For each $CH_y$ receiving msg from $CH_x$ ,
          $CH_x$ becomes forwarding CH
Step 5    For each $CH_x$ receiving msg from $CH_y$
          1.  If $hop\_num_{old}$ < $hop\_num_{new}$, do
              nothing
          2.  If $hop\_num_{old}$ > $hop\_num_{new,}$ replace
              forwarding CH
          3.  If $hop\_num_{old\,=}$ $hop\_num_{new}$, then if
              $Energy_{old}$ <= $Energy_{new}$ replace
              forwarding CH,else do nothing.
Step 6    Stop
```

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 as in Fig. 2. During a round, once selection has been performed, the only nodes transmitting data are the subordinate nodes. 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 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 as can be seen in Fig. 3. Also, all transmissions made by nodes post the cluster-heads failure are lost since they never get sent to the base-station.



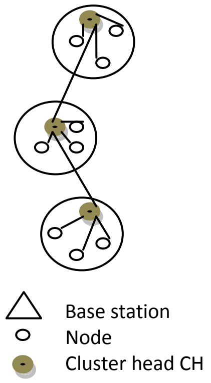


Fig. 2 Hierarchical clustering in FTHC

3.1.3 Failure detection phase

After every fixed set of transmissions 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

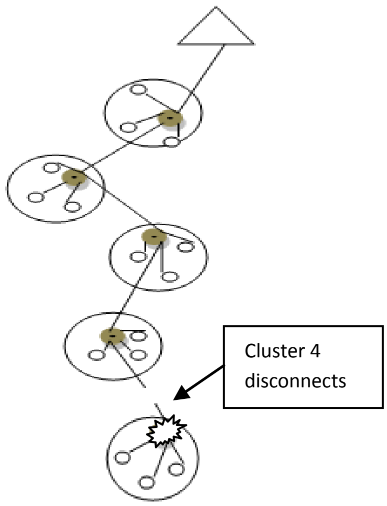


Fig. 3. CH failure

3.1.4 Failure recovery phase

We present two models for recovery from cluster-head failure in FTHC. We call these models Inter-Cluster Recovery and Intra-Cluster Recovery. In the basic form of FTHC, each node attaches itself to a single cluster-head during the selection phase of each round. 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. In the Inter-Cluster Recovery model, the nodes in the cluster with their CH failed become members of the CH of the upper level cluster as in Fig. 4. However, simply picking alternative cluster-heads to serve in case of lower CH failure is not sufficient since these nodes must now accommodate room in their TDMA schedules for the nodes they are serving in failure. 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. When the transmission phase of a round begins, a node transmits according the schedule of 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 backup cluster head.

Successive failures can be handled in this manner until a node runs out of backup cluster-head.

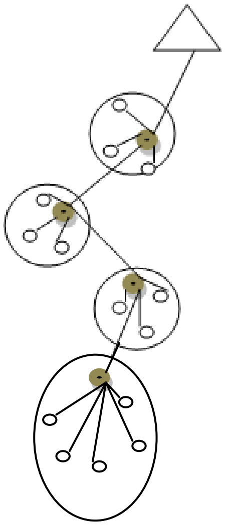


Fig. 4 Inter cluster recovery

The Intra-Cluster Recovery scheme of recovery is much simpler than the Inter-Cluster recovery. The idea here is to have a failed cluster-head seamlessly replaced by the next high energy node in the same cluster as in Fig. 5. 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 failure is detected by means of a missed ping message, the node with the next highest battery left takes over as the cluster-head.

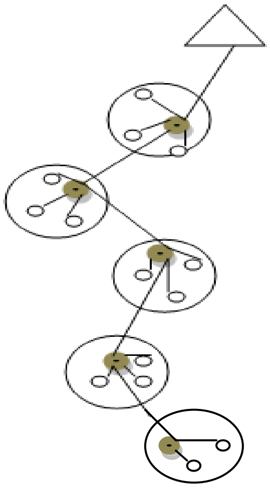


Fig. 5 Intra cluster recovery

3.2 Results and discussion

For our simulation setup, we experiment with networks of 20 sensors randomly placed in 100x100 area. 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, for the purpose of this simulation we considered homogeneous nodes with identical batteries (100%). The simulations also employ a linear power model as in [1] where the energy required to transmit over a given distance is a linear function of the distance. In Fig. 6 and Fig. 7, the graphs have been plotted to show the lifetime of the clusters along the different number of rounds. In MECH, a node dies when its battery life becomes 0. The first node dies in round 44. All the nodes of the cluster 0 die in round 73. Cluster 1, 2, 3 and 4 die in round 80. While in FTHCE, the first node dies accidently, not due to zero energy level which dies in round 23 and the recovery algorithm is applied. Hence the lifetime of network increases and first cluster dies in round 78 due to energy level reaching 0 in both FTHC intra and FTHC inter cluster recovery schemes. Clusters 1, 2, 3 and 4 die in round 80 in FTHC intra. In FTHC inter Cluster 1 and 2 die in round 80, while Cluster 3 and 4 die in round 86. These results show that the life time of overall network increases in FTHC as compared to MECH.

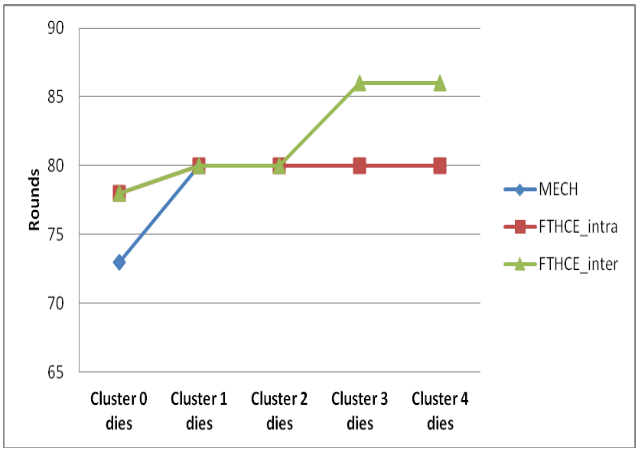


Fig. 6 Network Lifetime Comparison

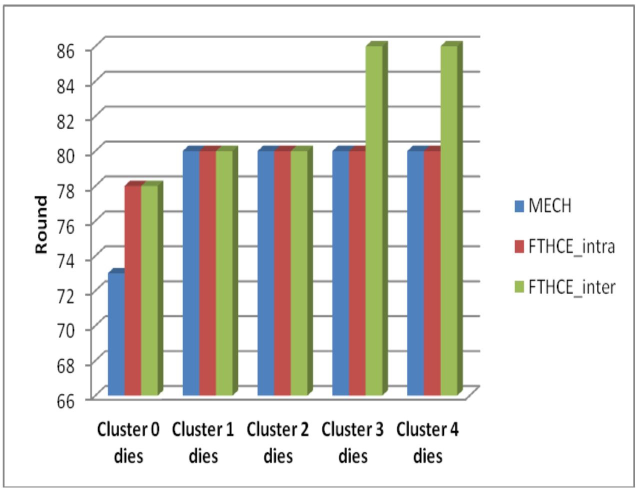


Fig. 7 Cluster Lifetime

Fig. 8 shows comparison between MECH, FTHC inter and FTHCE intra in terms of the number of rounds when first and last node dies.

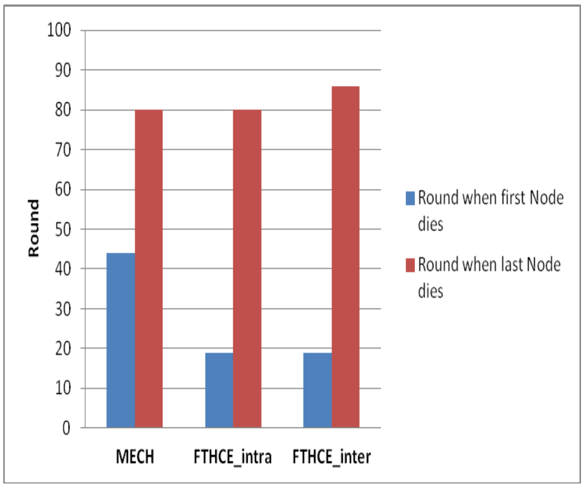


Fig. 8 Failure time for first and last node

Fig. 9 gives an idea of how the failure of cluster heads at various levels leads to the disconnection of clusters from other clusters/ Base station in MECH, FTHC intra and FTHC inter respectively.

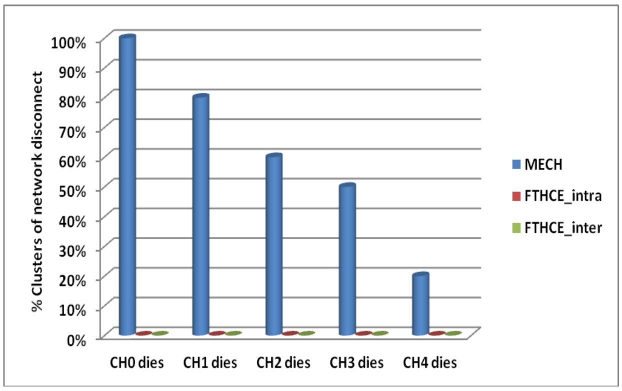


Fig. 9 Network Lifetime after the cluster head failure

In MECH, when the cluster head at level 4 (CH4) dies results in the disconnection of cluster 4 from the rest of the network. Similarly, the failure of cluster heads at levels 3, 2, 1, 0 i.e. CH3, CH2, CH1, and CH0 respectively results in the disconnection of the clusters 3, 2, 1 and 0 respectively from the rest of the network. The percentage of clusters disconnected has been shown better understanding of the results. While in case of FTHC inter and FTHC intra, whenever cluster head at any of the levels fails accidently, the recovery algorithm is called up and new cluster head takes the responsibility for the rest of the round. Therefore the disconnection of the cluster at that level from the rest of the network does not take place. Though these better results in the FTHC has been achieved at the cost of increased overhead of messages between nodes and/ or between nodes and base station. Also additional cost for Inter-Cluster recovery is due to the additional uptime for cluster heads caused by longer schedules.

4. CONCLUSION

In this paper we have discussed about the problem of network disconnectivity due to cluster head failures in wireless sensor networks and we have tried to find a solution for that. We have proposed fault tolerance in wireless sensor networks with hierarchical clustering environment. 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 new cluster head in case failure of its own cluster head, it must now spend more energy communicating with this cluster-head since it is further away than the failed cluster-head. The preliminary simulation results also show that increased network connectivity and fault-tolerance can be achieved at the cost of increased overhead of messages and longer schedules. In applications where connectivity is critical, this is a viable trade-off to make.

5. REFERENCES

[1] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks (LEACH). Proc. of the 33rd Hawaii International Conference on Systems Science-Volume 8, pp. 3005-3014, 2000.

[2] S. Lindesy and C. Raghavendra. PEGASIS: Power-Efficient Gathering in Sensor Information System. Proc. of 2002 IEEE Aerospace Conference, pp. 1-6, 2002.

[3] Manjeshwar and D. Agrawal. TEEN: A Routing Protocol for Enhanced Efficient in Wireless Sensor Networks. Proc. of the 15th International Parallel and Distributed Processing Symposium, pp. 2009-2015, 2001.

[4] R.S. Chang and C.J. Kuo. An Energy Efficient Routing Mechanism for Wireless Sensor Networks. Proc. of the 20th International Conference on Advanced Information Networking and Applications, 2006.

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

[6] M. Chu, H. Haussecker, and F. Zhao. Scalable Information- Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks. The International Journal of High Performance Computing Applications, Vol. 16, No. 3, pp. 293-313, 2002.

[7] S. Schmid and R. Wattenhofer. Algorithmic models for sensor networks. In Parallel and Distributed Processing Symposium, 2006.

[8] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. Wireless Networks, Vol. 8, pp. 169-185, 2002.

[9] K. Sohrabi, J. Pottie. Protocols for self-organization of a wireless sensor network. IEEE Personal Communications Vol. 7, Issue 5, pp. 16-27, 2000.

[10] M. Ding, D. Chen, K. Xing, and X. Cheng. Localized fault-tolerant event boundary detection in sensor networks. In INFOCOM 2005.

[11] C. Frank and K. R¨omer. Algorithms for Generic Role Assignment in Wireless Sensor Networks. In Proc. of the 3rd international conference on Embedded networked sensor systems, pp. 230–242, 2005.

[12] R. Guerraoui and A. Schiper. Fault-Tolerance by Replication in Distributed Systems. In Proc. of the 1996 Ada-Europe International Conference on Reliable Software Technologies, pp. 38–57, 1996.

[13] G. Gupta and M. Younis. Fault-Tolerant Clustering of Wireless Sensor Networks. Wireless Communications and Networking, 3:1579–1584, 2003.

[14] B. Krishnamachari and S. Iyengar. Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks. IEEE Transactions on Computers, 53:241–250, 2004.

[15] P. Levis and D. Culler. Mat´e: A Tiny Virtual Machine for Sensor Networks. In ASPLOS-X: Proc. of the 10th international conference on Architectural support for programming languages and operating systems, pp. 85–95, New York, NY, USA, 2002.