Energy Balancing in the Self-configuring Sensor Networks

[#] Kazi Muheymin Sakib [1], Zahir Tari [2] , Ibrahim Khalil [3]
[1,2,3]*School of Computer Science and*
Information Technology, RMIT University
Melbourne, Victoria 3001, {*sakib,zahirt,ibrahimk*}*@cs.rmit.edu.au*

Abstract

Like other distributed systems, unattended sensor networks need to be balanced not only by load but also in terms of energy. Sensor networks are expected to live long. Lifetime can be maximized if and only if a balanced network can be formed. In this research work we first identify the tasks, that a sensor node accomplishes in its life time. Based on those, we define the sensor node life-cycle. Then we characterize the nodes based on their residual energy levels as, SEN - nodes having sufficient energy to perform additional responsibilities other than its own sensing task, or NEN - nodes having only necessary energy to perform its own task. Every node starts as SEN, and at some point it becomes NEN based on a predefined threshold value. Thus the nodes are protected from the early exhaustion. Finally, we developed a virtual clustering technique which is self-configurable and scalable. We apply our approach to a simulation environment and the simulation results justify our assertion.

1. Introduction

Each of the activities in a sensor network directly related to the consumption of energy. In such a distributed system, distributing processing and communications activity evenly across a network is highly required so that no single sensor node would be died early. Load balancing is especially important for sensor networks where it is difficult to predict the number of queries for sensing that will be issued. Sensor networks are often unattended. A sensor network has limited memory, limited bandwidth, and limited computational capabilities [1]. So well defined characterization of sensor nodes and better management of node life-cycle is needed. Despite of its energy constraint nature a sensor network is required to live long. The technique also should be scalable.

We suggest that system designers can address these challenging issues in a demand initiated node organization. Sensor Nodes should contribute to the network backbone only if it is needed to do so. Otherwise nodes can be inactive to save its precious energy for future usage. In a distributed dynamic system where the energy is a constraint, techniques should be developed where the recurrent transmission of dynamic state information is avoided. Moreover, to extend the network life an energy balancing mechanism should also be incorporated.

The aim of this research is on developing a self-configurable energy balanced topology for the sensor networks which will live longer. This will be achieved by organizing sensor nodes in a virtual cluster, similar to ASCENT [2], and by balancing energy as well as tasks over the network, similiar to the distributed load balancing approach [3].

Energy scarcity makes the sensor networks unique compared to the other distributed networks. To deal with this unique nature network decisions should be made based on residual energy. However, without characterising the sensor nodes based on their energy levles, it is impossible to take any energy based decision. Surprisingly, we did not find many works classifying sensor nodes.

In their paper [4], Vivek et al. proposed two types of node deployment. Type 0 nodes are simple sensing nodes. Type 1 nodes are the cluster heads that perform long-range transmissions, data aggregation and routing within the clusters etc. We argue that, if nodes are pre-characterized, perhaps from the factory, a precise node deployment is essential. The position of nodes need to be exact, otherwise some of them will be disconnected. This limits the network coverage. Yong et al. [5] identifies three types of sensor nodes. They denote sensors according to their resources as SRC - Small Resource Capacity, MRC - Medium Resource Capacity and LRC - Large Resource Capacity. They assumed that, LRC nodes are directly connected with the main power supply. This kind of nodes can only be found in an in-home sensor network. However, in a practical sensor field, the residual energy of sensor nodes is decreasing continuously.

In this research, we classify the nodes into two types, SEN, nodes having sufficient energy to carry others information as well as sense own task and NEN, nodes having only necessary energy to sense its own task. By doing this, we can give protection to the dying nodes which in turn increases the network lifetime. Other than that it also helps us to build a balanced network.

Hierarchichal organization of sensor nodes is used in [6], [7], however hierarchy may not always be scalable or self-configurable. Clustering, a special type of hierarchichal organization, is particularly useful for application that require scalability [8]. In a cluster, nodes send their data to the cluster head and the cluster head forwards the data to other cluster heads to get closer to the destination node [9]. Clustering enables bandwidth reuse, better resource utilization and power control [10]. However, conventional clusters rely on a fixed infrastructure, more precisely, on a fixed area. Conventional clustering algorithms require all of the participating nodes to advertise cluster-dependent information repeatedly [11]. Some existing technique even require special types of nodes like energy-limitless sensors [5], as cluster heads.

Instead of conventional clustering Kwon et al.: presented a passive clustering technique [11]. In passive clustering the clusters are allowed to be overlapped. However, every time the topology changes there will be at least two types of selection or election procedures: one election procedure for cluster head and several others for identifying common nodes agreed by the neighboring cluster heads. In ASCENT [2], Cerpa et al. selects the active nodes to reinforce the topology. Whenever they select a node as an active node it stays awake all its' lifetime and performs multihop routing. The area of interest for sensing may change randomly, an active node may no longer be needed as active. There may be more than the sufficient number of active nodes to carry the data.

The main contribution of this paper, apart from the node characterization (Section 3), is to develop a virtual clustering technique instead of its conventional counterpart which is described in Section 4. Instead of fixed architecture, here we developed a dynamic, mutually overlapped clustering mechanism. Members of a cluster are attached to the gateway instead of a cluster head, and the connection is not rigid. Any node can be attached to multiple gateways. Thus any node can be member of multiple clusters. We argue that the responsibilities of each sensor node might be changed over time. Based on current demand, nodes should be selected for specific task. To identify the tasks, we first define the node lifecycle in Section 2. Finally in Section 5 we show our simulation methods and results, which leads to draw the conclusion in Section 6.

2. Sensor node life-cycle

Sensor nodes are resource constrained, specially in terms of energy. To manage the sensor nodes efficiently, a well defined node life cycle is needed. Node lifecycle will help us to understand and identify the tasks that are involved.

By observing the activities of sensor nodes, a specific pattern is found over their lifetime. Each node starts with initialization and then based on the current available metrics such as energy level, communication environments and query requests it enters into the execution phase.

According to their activities, we divide the sensor node lives into three categories -

- Initialization phase
- Decision making phase and
- Execution phase

The phases are shown in Figure 1. The initialization phase only occurs once when a sensor node starts itself. In our diagram the Start state represents the initialization phase. During their life span, sensor nodes have to make various decisions, for example in which execution state it should be. This is the

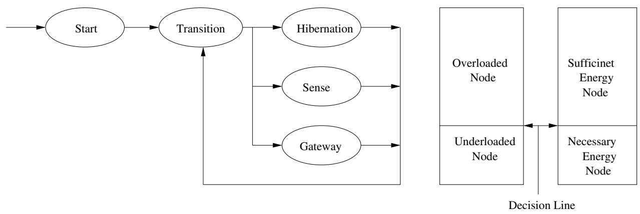

Fig. 1: Sensor Node Life-cycle with Distributed and Sensor Node Characterization.

decision making phase represented by the 'Transition' state in the diagram. Apart from initialization and decision-making, each node is either in the non-active state or in the active state. In the non-active state hibernating nodes do nothing other than listening. Active states can be divided into two types - sense and gateway. In the sense state nodes sense its own territory based on the prescheduled request or query from the sink (ie. the destination node). On the other hand, in the gateway state, nodes perform data communication, data aggregation in addition to its own sensing.

3. Node classification

Without characterizing the nodes on their energy levels, it is impossible to create an energy balanced network. At any given time, the residual energy level of each node may not be the same. This is understandable because besides sensing, some of the nodes have additional responsibilities, like transmitting other node's information and data aggregation. To make the network live long, these additional responsibilities should be redistributed among other nodes periodically. Periodical task redistribution will ensure an energy-balanced environment in the network. Distribution of responsibilities must be fair to ensure that a node having minimum energy to sense its region and transmit the sensed data to its nearby gateway node, should not be considered for any additional burdens.

Here we assumed that all the nodes are architecturally equal, that is, all other resources - processing power, memory capacity are equal in all the sensor nodes. It is only the energy, which is variable. It is actually decreasing over time. This unique nature of sensor nodes leads us to characterize them based on their residual energy. In traditional distributed systems, processors are characterized by their process loads [3]. For example, processors having loads below a certain threshold are called under-loaded processors as shown in Figure 1. While others with a load above the predefined threshold are called overloaded processor. Similar concept is adopted here. Sensor nodes will be divided into two groups at a given time-

1) Sufficient Energy Node (SEN)
2) Necessary Energy Node (NEN)

The current (residual) energy level will be normalized to the maximum battery capacity and scaled to 100 [12]. Based on the normalized residual energy; a minimum level of energy is drawn, which will be at least needed to sense the region and to transmit the sensed data to nearby gateway station. Nodes having residual energy above that level are called Sufficient Energy Node (SEN), these nodes have sufficient energy to take additional responsibilities. Nodes having residual energy less than or equal to that level are called Necessary Energy Node (NEN), these nodes only have energy to perform their own sensing tasks. Now the nodes are characterized into two types, we propose, a node only need to inform others when it changes from SEN to NEN. This technique reduces communication overhead substantially.

A. Determining node types (SEN/NEN)

For simplicity we assumed that -

- Comparing with the data transmission or data reception, energy consumption in sensing is negligible [13].
- A node only transmits whenever a data packet is ready, and packet generation is directly proportional to sensing.
- We assume either nodes receive requests for sensing from sink nodes or there is a prescheduled query task for every node. Based on the query, nodes sense and generate data packets. The query of sensing may be random, non-uniform, so we can assume that, each sensing node has packet streams with Poisson distribution.

Then, energy consumption or transmission cost over time t of a non-gateway node is -

$$E_{i(NG)}(t) = E_{tx} \times \lambda_i \times t \tag{1}$$

Where, E_{tx} - is the energy cost for transmitting a single packet. λ_i - is the packet generation rate at node *i*.

A gateway receives data packets from its neighbors then transmits those to the next gateway. Since all the neighbors have independent packet streams, the packet stream for a gateway node is still a Poission process. Let, gateway *j* itself generates packets at a rate λ_j and has n_j neighbors. Then the packet arrival rate at gateway *j* is -

$$\lambda_j = \lambda_j + \sum_{k=1}^{n_j} \lambda_k \tag{2}$$

Where, λ_k is the packet generation rate at the *kth* neighbor.

The Expectation value of total packets at gateway *j* for duration *t* is -

$$\overline{X_j}(t) = \lambda_{j0} \times t + \sum_{k=1}^{n_j} \lambda_k \times t \tag{3}$$

For simplicity, let us assume that, energy to transmit (E_{tx}) and receive (E_{rx}) a single data packet is the same and is constant, ie. $E_{tx} = E_{rx} = C$,

Total energy consumption at gateway *j* - total energy spent to receive packets from the neighbors and transmit $\overline{X_j}(t)$ packets -

$$E_{i(G)}(t) = E_{rx} \times \sum_{k=1}^{n_j} \lambda_k \times t + E_{tx}\overline{X_j}(t) \tag{4}$$

$$\Rightarrow E_{i(G)}(t) = C\left(\sum_{k=1}^{n_j} \lambda_k \times t + \overline{X_j}(t)\right) \tag{5}$$

Now, if node *i*, spends t_1 time as a non-gateway node and t_2 time as a gateway, total energy consumption of node *i* over time *t* (where, $t = t_1 + t_2$) can be found by equation 1 and equation 5, as -

$$E_i(t) = E_{i(NG)}(t_1) + E_{i(G)}(t_2) \tag{6}$$

If total amount of initial energy in a sensor node, is *E* and the energy decision level is E_{th} (Energy Threshold) then,

$$\delta = \frac{E - E_i(t)}{E} = \begin{cases} \delta > E_{th} & Type = SEN \\ \delta < E_{th} & Type = NEN \\ \delta = 0 & Type = Exhausted \end{cases} \tag{7}$$

4. Proposed Topology Formation Technique Description

Our technique deploys nodes according to their abilities and responsibilities. The technique selects gateways, normal sensing nodes or hibernating nodes with the aim of creating a balanced, lifetime maximized network. Based on the current demand nodes can move from one state to another.

A. Virtual clustering

Here we propose a novel virtual clustering technique. On top of that other networking issues, e.g. routing, data aggregation can be implemented. In virtual clustering, there is no fixed area for any clusters. Cluster areas can overlap with each other's. Members of any virtual cluster can be associated with multiple gateways. Members of any virtual cluster can even initiate to form another virtual cluster as overlapping is allowed. Virtual clustering is demand initiated, adaptive and self-configuring.

We build the virtual clusters around the gateway nodes. The gateway selection algorithm, described later, choose the appropriate gateways. To maintain the network connectivity each gateway is formed within the range of at least another gateway. It will also ensure that nodes within the cluster will have redundant connections to their gateways and have minimized packet loss probability as well. The region of a virtual cluster is the region of the gateway node, so that clustering is nothing but selecting the gateway nodes then associating the neighbors with the gateway.

Whenever a gateway node is selected, it informs its neighbors through gateway confirmation message. Each node receives the gateway confirmation message, updates the gateway table. Though the gateway node does not maintain any member list, the nodes received the gateway confirmation message become connected with that gateway. To be able to transmit sensed data, every node needs to be connected with at least one gateway. Thus the gateway will form a loosely coupled virtual cluster with the surrounding neighbors, which is shown in the Figure 2.

According to the definition of a gateway, there must have at least one gateway node in the neighbors. We assumed that there are some nodes called initial gateways. Initial gateways never change their states. Sink nodes are considered as all time gateway nodes. Based on these initial gateway nodes, the topology will be formed all over the sensor field. This assumption was necessary to bootstrap.

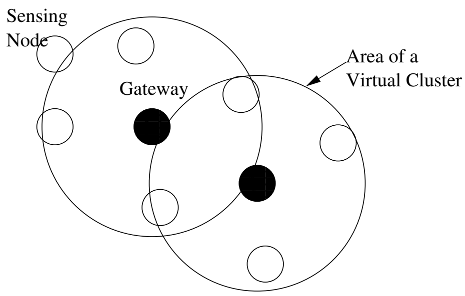

Fig. 2: Virtual Clustering.

B. Gateway selection Algorithm

After receiving a gateway request message, a node first checks its energy level and connection to other gateways. If the energy level of that node is above the energy threshold and the node is connected with at least another gateway node, it is considered as an eligible candidate to become a gateway. Each eligible node then checks the number of gateway nodes in its communication range. If the number of gateway nodes equals the upper bound of a threshold (*Gth*), it refrains itself from becoming a gateway node.

If the node is a SEN, connected to at least one other gateway node and the number of connected gateways has not exceeded the gateway threshold *Gth*, it enters into the transition state to participate in the gateway formation procedure. The node then sends its willingness message to its neighbors and waits for a predefined amount of time (t_p) to hear from other nodes. If there are no other aspirant nodes from the neighbors, it becomes a gateway node, and informs the neighbors to update their respective tables. If the node receives willingness messages from others, that is, there is more than one eligible node; the node will wait for a random amount of time (t_s). Within this period of time, if any confirmation of gateway message is received from any other nodes, it exits from the gateway procedure and reverts to its previous state. Otherwise, after waiting t_s time, it sends its confirmation of gateway message to its neighbors and enters into the gateway state. The gateway selection algorithm is given in the Algorithm 1.

1) Network connectivity and coverage: As gateway selections are demand initiated, there could be isolated clusters. However, our algorithm carefully considers that important issue. Here we mathematically proved that, the algorithm ensures the network connectivity and network coverage.

Assume that,

- n_i is the set of neighboring nodes of gateway *i*.
- *N* is the set of all the sensor nodes in the network.
- $f(n)$ is the gateway selection algorithm.

The algorithm starts from the initial gateway nodes, and it selects gateways based on descent directions. If the communication range of a sensor node is *r*. average distance of the next gateway node will be $\frac{r}{2}$. From the initial gateway, the algorithm constructs a gateway sequence according to the following recursion rule.

Assume:
Each node has a gateway information table.
Let :
NType: type of node GNo: no of gateways
Gth: gateway threshold State: state of the node
Msg: Control message Table: gateway info
t_p: predefined wait time t_s: random wait time

When gateway request message received;
if *NType* $\neq$ *NEN AND Gno* $\neq 0$ *AND Gno* $\leq$ *Gth* **then**
  $State_{prev} \leftarrow State_{current}$;
  $State_{current} \leftarrow State_{transition}$;
  Broadcast($Msg_{willingness}$);
  Wait(t_p);
  if $Msg_{received}$ == $Msg_{willingness}$ **then**
    Wait(t_s);
    if $Msg_{received}$ == $Msg_{confirmation}$ **then**
      Update(Table);
      $State_{current} \leftarrow State_{prev}$;
    end
    else
      Broadcast($Msg_{confirmation}$);
      $State_{current} \leftarrow State_{gateway}$;
    end
  end
  else
    Broadcast($Msg_{confirmation}$);
    $State_{current} \leftarrow State_{gateway}$;
  end
end

Algorithm 1: Gateway Selcetion Algorithm

If, from the initial point (ie. sink), distances of a gateway *i* and its neighboring gateway *i+1*, are h_i and h_{i+1} respectively, then -

$$|h_{i+1} - h_i| \leq r \quad (8)$$
$$\Rightarrow h_{i+1} \leq h_i + r \quad (9)$$

According to the definition of a gateway, there must be another gateway ($GNo \neq 0$) within the communication range of that node. So that -

$$f(n_i) = \{x_i | x_i \in n_i \ AND \ x_i \in n_{i-1}\} \quad (10)$$
$$f(n_{i+1}) = \{x_{i+1} | x_{i+1} \in n_{i+1} \ AND \ x_{i+1} \in n_i\} \quad (11)$$

Which implies that -

$$n_i \cap n_{i+1} \neq \phi \quad (12)$$

Now, if equation 9 and equation 12 holds and if there are no nodes which form partition in the network, then -

$$n_0 \cup n_1 \cup n_2 \cup .... \cup n_i = N \quad (13)$$

Equation 12 ensures that a virtual cluster is connected to at least one other cluster. That ensures the connectivity among the network. Equation 13 ensures that the algorithm covers the entire network.

5. Simulation

We used OMNeT++ as the simulation tool. We assumed that the sink node is a stand alone machine connected to the main power supply. Otherthan that, all the sensor nodes have the same initial memory, same energy and same processing power. We also assumed that the nodes have the same communication range. Any node within the communication range can communicate with others bidirectionally. We have created our own energy model as described in the section III. The sink node is positioned at the top right corner of the simulated area, we then deploy the sensor nodes. Initially, we created a two hop network by putting the nodes as far as possible, that is, at the opposite end of the communication range. After that, we gradually increase the node density.

Number of generated data packets are directly proportional to the sensing query. The sensing query is also random and exponentially distributed. For each data communication there will be involved three types of energy consumption - sensing, transmit data to the gateway, then receive and retransmit that packet among the gateways. We assume that the energy consumed while there is no query negligible.

Our goal is to implement our algorithm in the simulation environment to compare with others. This serves the important purpose of validating our theoretical assumptions. The metrics that have been chosen to analyze the performances are : Network Lifetime - which is measured by two ways. The time taken before the first node of the network dies. A similar metric has been used in [14]. The other one is the time taken to exhaust 50% of the nodes, similar to the metric used in [2]. To show energy balancing we used here the standard deviation of residual energies. Standard deviation is calculated when the residual energy of at least one of the nodes reaches to zero.

On top of our topology we used shortest path algorithm. The transmission layer implements the simple CSMA MAC, which is provided by the OMNeT++. We used fixed parametric value for the Gateway threshold ≤ 3, Decision line of Energy levels =20% Gateway formation waiting time is twice the roundtrip. We will justify these values later.

A. Comparative Study

We compare the energy balancing and average lifetime of a sensor network where our proposed technique is applied, with the all-active case [the all-active case denotes the simple deployment of sensor nodes, where no topology formation algorithm was applied.], and ASCENT [2].

In Figure 3 we plot the standard deviation of remaining energy levels when the first node reaches to zero. This metric shows us how balanced the network is. For an ideal condition, where the tasks are exactly distributed, the standard deviation of remaining energy levels should be zero. The figure shows that the deviation is about 50% in our technique, while others are around 90%

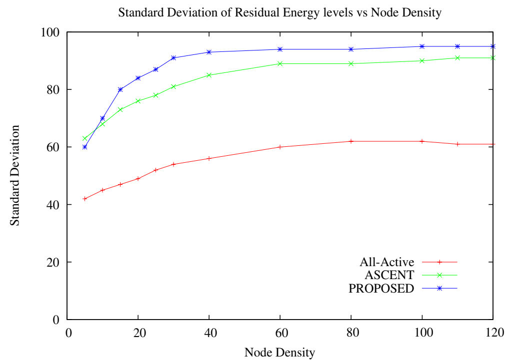

Fig. 3: Density vs. Standard Deviation of remaining energies : when first node dies.

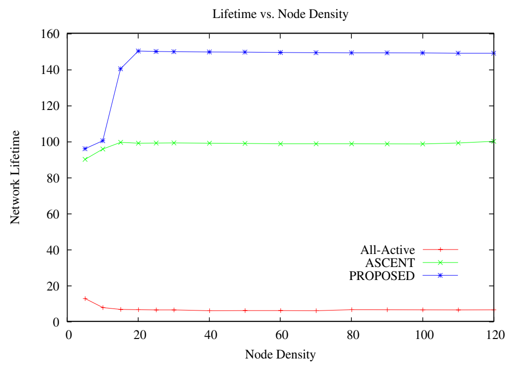

Fig. 4: Density vs. Lifetime : time taken to die the first node.

Figure 4 shows the time taken for the first node to die as a function of the node density. At a lower density our proposed technique performs similar to the ASCENT. This is because, when the density is too low, there will be excess gateway nodes to carry the data. The number of gateways is restricted by the parameter, gateway threshold. However, when we increased the node density, there were less gateway nodes comparing to the non-gateway nodes, the task distribution was more even and there were more nodes to carry the duty of an out going gateway. In the proposed technique, if any node crosses the decision line of the energy level, it is considered as NEN and they are forced to leave the gateway state. NEN nodes are alive and only can perform their own sensing. This kind of essential classifications were absent in other methods, which strongly effects their network lifetime. The result clearly shows that the proposed technique outperforms the others under moderate to high node densities. Another significance of this result is the scalability. It rapidly reaches around 150 time unit with 15 nodes, and enters into a saturation region.

Figure 5 shows the time taken to die at least 50% of the nodes while we varied the node density in our test bed. Though the gaps between the other techniques are reduced, however, we argue that, the network might be partitioned well before the time it takes to exhaust the 50% nodes. The figure shows that the proposed technique is still better than others.

In Figure 6 we show the number of events, ie. the number

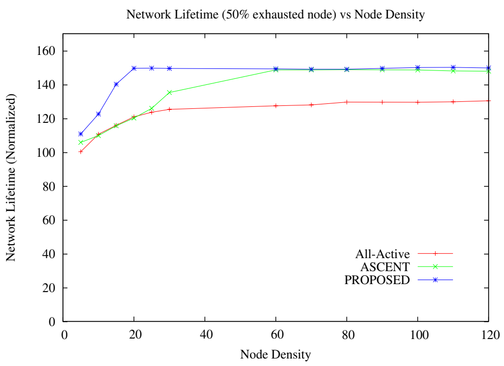

Fig. 5: Density vs. Lifetime : time taken to die 50 percent of the nodes.

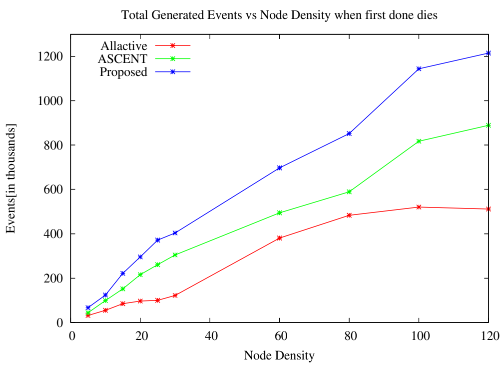

Fig. 6: Density vs. Total number of Generated events (in thousands) : upto exhaustion of first node.

of generated events upto the time when first node dies. Our proposed technique detects almost same number of events in a lesser dense condition comparing to other two techniques. However, when we increase the density it detects almost double than the ASCENT and tripple than the Allactive techniques detects the events. The result commensurate with our previous results (Figure 4 and 5) that our technique increases the life time.

6. Conclusion and Future Work

Sensor nodes are characterized as NEN or SEN (Figure 1). By doing this, we able to reduce the overhead transmission by a large margin as nodes now need to inform their state once in a lifetime only when it reduces from SEN to NEN. Energy consumption is directly related to the task that is performed by a node. Clear understanding of node lifecycle (Figure 1) helps us to distribute the tasks evenly. Which is important to make the network balanced.

Each of the gateway nodes forms an overlapping cluster as shown in Figure 2. We call these clusters as virtual clusters. Because the members and area of any particular cluster are not mutually exclusive. The organization ensures possibly redundant transmission paths from source to the sink. Virtual clusters are dynamic and adaptive. Whenever a node faces a high packet loss, or a node moves out of communication range from its gateway, it can initiate the gateway formation procedure. As a result, a new virtual cluster is formed. The topology, that formed, is self-organizing and scalable.

This research does not consider the network partition. Network partition can happen due to dying of sensor nodes. Sensor nodes can die early because of over activity or external events like stomping, natural calamities. Detecting network partitions and handling that partition will be our future work. We also do not consider the reduntant node coverage. However, we do believe scheduling for node coverage may enhance the network life.

Acknowledgments

This work is supported by Australian Research Council (ARC), linkage scheme, no. LP0347217.

References

[1] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[2] A. Cerpa and D. Estrin, "Ascent: Adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, July 2004.

[3] P. K. Sinha, *Distributed Operating Systems, Concepts and Design*. IEEE Press, 1997.

[4] V. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–15, Jan. 2005.

[5] Y. Ma and J. H. Aylor, "System lifetime optimization for heterogeneous sensor networks with a hub-spoke topology," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 286–294, July 2004.

[6] S. J. Baek, G. D. Veciana, and X. Su, "Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation," *IEEE Journal on Selected Areas in Communication*, vol. 22, no. 6, pp. 1130–1140, Aug. 2004.

[7] X. Cheng, B. Narahari, R. Simha, M. Cheng, and D. Liu, "Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 248–256, July 2003.

[8] H. Edgar and J. Callaway, *Wireless sensor networks: architectures and protocols*. Auerbach Publications, Boca Raton, FL, USA, 2003.

[9] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct. 2004.

[10] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *5th ACM/IEEE Mobicom Conference (MobiCom '99)*, Seattle, WA, USA, Aug. 1999.

[11] K. J. Taek, M. Gerla, V. Varma, M. Barton, and T. Hsing, "Efficient flooding with passive clustering-an overhead-free selective forward mechanism for ad hoc/sensor networks," *IEEE*, vol. 91, no. 8, pp. 1210–1220, Aug. 2003.

[12] X. Hong, M. Gerla, and H. Wang, "Load-balanced, energy-aware communications for mars sensor-networks," in *IEEE Aerospace Conference*, Bigsky, MI, USA, Mar. 2002, pp. 3–1109 – 3–1115.

[13] R. Min, M. Bhardwai, N. Ickes, A. Wang, and A. Chandrakasan, "The hardware and the network: Total-system strategies for power aware wireless microsensors," in *IEEE CAS Workshop on Wireless Communications and Networking Power Efficient Wireless Ad Hoc Networks*, Pasadena, California, USA, Sept. 2002.

[14] J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transsactions on Networking*, vol. 12, no. 4, pp. 609–619, Aug. 2004.