International Journal On Engineering Technology and Sciences – IJETS™
ISSN(P): 2349-3968, ISSN (O): 2349-3976
Volume III, Issue IV, April- 2016

CLUSTER BASED LOAD BALANCING AND INTRUSION DETECTION SYSTEM FOR MANET USING PANEL PROTOCOL

A.Saroja
PG Scholar
Department Of CSE
Professional Group Of Institution,
Palladam.
E-mail – sssaro3@gmail.com.

R.Roopa– M.E.(CSE),
Assistant Professor,
Department of CSE,
Professional Group Of Institution,
Palladam.
E-mail – roopa.renuga@gmail.com.

Abstract- ***Advances in mobile ad-hoc network (MANET) technology have enabled small and low-cost sensors with the capability of sensing various types of physical and environmental conditions, data processing, delay sensitive applications and wireless communication. In the MANET, the sensor nodes have a limited transmission range and their processing and storage capabilities as well as their energy resources are limited. A triple umpiring system has already been proved for its better performance in MANETs. The clustering technique is effective in prolonging the lifetime of the MANET. In this paper, we introduce PANEL, a position-based aggregator node election protocol for wireless sensor networks. The novelty of PANEL with respect to other aggregator node election protocols is that it supports asynchronous sensor network applications where the sensor readings are fetched by the base stations after some delay. In particular, the motivation for the design of PANEL was to support reliable and persistent data storage applications. PANEL ensures load balancing, and it supports intra-and inter-cluster routing allowing sensor to aggregator, aggregator to aggregator, base station to aggregator, and aggregator to base station communications. PANEL creates more cohesive clusters than LEACH, and, on the other hand, that PANEL is more energy efficient than LEACH.***

Keywords-Sensor Networks, Network Protocols, Aggregator Node Election, Fault Tolerance, Security.

Introduction

Mobile Ad-hoc Networks consist of a multitude of tiny sensor nodes capable for wireless communications and a few powerful base stations. The sensor nodes usually perform some monitoring task (e.g., measure various environmental parameters). The base stations collect sensor readings a nd forward them for further processing to a service center.

Based on how the sensor readings reach the base stations, we can distinguish synchronous

And asynchronous sensor networks. In the synchronous case, the sensor readings are sent to the base stations in real-time using multi-hop wireless communications, where the sensor nodes cooperatively forward data packets on behalf of other sensor nodes towards the base stations. In the asynchronous case, the sensor readingsare fetched by the base stations after some delay (e.g., once every day or week). In this case, the base stations are often mobile, and they physically approach the sensors in order to fetch their data through a single wireless hop. Examples of synchronous sensor network applications include for est fire alarm systems and building automation systems where real-time operation is indispensable. Examples of asynchronous applications include habitat monitoring systems and agricultural applications such as vineyard monitoring where real-time operation is not an issue. As sensor nodes are often severely resource constrained, various techniques have been proposed to ensure the efficient operation of sensor networks. One of these techniques is called aggregation or in-network processing. The idea is that instead of forwarding (in case of synchronous applications) or storing (in case of asynchronous applications) raw sensor readings, data can be first processed, combined, and compressed by some distinguished sensor nodes, called aggregators.

While aggregation increases the overall efficiency of the sensor network, the aggregator nodes themselves use more resources than the regular sensor nodes. For this reason, it is desirable to change the aggregators from time to time, and thereby, to better balance the load on the sensor nodes. For this purpose, aggregator node election protocols can be used in the sensor network that allow dynamic re-assignment of the aggregator role.

In this paper, which is an enhanced version of our previously published conference paper [5], we introduce PANEL, a position-based aggregator node election protocol for wireless sensor networks. As its name indicates, PANEL uses the geographical position information of the nodes to determine which of them should be the aggregators. Like other aggregator node election protocols, PANEL also ensures load balancing in the sense that each node is elected aggregator nearly equally frequently. The salient feature of PANEL that makes it novel and different from other aggregator node election protocols is that besides synchronous applications, PANEL also supports asynchronous applications. In particular, the motivation for the design of PANEL was to support Tiny PEDS (Tiny Persistent En-crypted Data Storage) [13], and other similar asynchronous sensor network applications. In TinyPEDS, aggregator nodes collect and aggregate sensor readings from the clusters that they are responsible for, and then persistently store the aggregated values (in an encrypted form). In addition, in order to increase reliability, the aggregators replicate their stored data at the aggregators of some selected backup clusters. These backup aggregators (i.e., the aggregators in the backup clusters) must be chosen in such a way that they are farther away from the primary aggregator than a certain distance called the disaster radius. The rationale is that if there is a disaster in which the primary aggregator is destroyed, its data is still available at and can be retrieved from the backup aggregators. Being a position-based protocol, PANEL supports TinyPEDSand applications alike by providing assurances regarding the distance between the elected aggregator nodes. The organization of the paper is the following: In Section 2, we report on the related work. In Section 3, we introduce the general assumptions that we based the design of PANEL upon. In Section 4, we describe the operation of PANEL. In Section 5, we present our simulation-based comparison of the performance of PANEL with that of LEACH [42], an aggregator node election protocol well-known from the literature. In Section 6, we discuss some possible extensions of PANEL. And finally, in Section 7, we conclude the paper.

2. Related Work

One of the most well-known approach for aggregator node election is the LEACH protocol [6]. In LEACH, the clustering is based on random numbers: each node picks a random number and according to its value the node becomes a cluster-head (and in the same time aggregator) or remains cluster member. The cluster members join the cluster of the cluster-head with the highest energy advertisement. The advantage of LEACH is that it °atly balances the energy consumption of the network, but it uses one-hop communication between the cluster members and the elected cluster head, as well as between the cluster heads and the base station, which can waste energy.

Other clustering protocols in the literature can be classified on the basis of how they elect the aggregator nodes. For example, in [13], the communication cost and the remaining energy of the sensor nodes is considered, while in [3], a generalized *weight* is used for this purpose. Graph theoretical approaches can be found in [2] and in [8]. In [1], the authors propose heuristics to form clusters of nodes that are within *d* hops away from each other, while in [4], new clusters are created as the size of the overlapping areas of existing clusters becomes small. The SANE protocol [11] combines three random aggregator node election schemes while considering adversarial attacks.

The papers listed above are all related to the aggregator node election problem assuming clustering. However, none of the above methods are able to guarantee a minimum distance between certain aggregators. However, in our motivating application area (i.e., reliable and persistent distributed data storage), backup aggregators must be chosen in such a way that they reside farther away from the primary aggregator than a certain disaster range. PANEL can guarantee a minimum distance between aggregators, because in PANEL, the aggregator nodes reside within fixed size clusters. For instance, the minimum distance between two aggregators belonging to non-neighboring clusters is *dx*, where *x* is the number of clusters between the two aggregators, and *d* is the cluster size.

One of these papers is [43], which discusses the main issues of clustering in sensor networks and concludes that clustering is a useful tool for topology-management and for in-network data aggregation. In [32], the authors detail the different data aggregation techniques and highlight the trade-offs between energy efficiency, data accuracy, and latency. In [29], one can read about a comparison of homogeneous (i.e., all the nodes have same hardware capability) and heterogeneous (i.e., nodes have different hardware capabilities) sensor networks from clustering point of view, while in [28], the authors investigate the schemes of single-hop and multi-hop communication and their impact on clustering.

The papers that propose solutions for clustering can be classified based on their primary aim (however, most of these papers have multiple aims). The largest group according to this classification consists of papers that aim at lifetime maximization of the sensor network. This group of papers can be further divided based on the method that they use for the clustering: it can be either probabilistic or deterministic. In case of probabilistic solutions, the cluster heads are elected based on some randomness, while in case of deterministic solutions, some iterative or centralized strategies are deployed. The most known probabilistic cluster formation algorithm is the LEACH protocol [17, 18]. In LEACH, the clustering goes as follows: in every round each sensor node picks a random number , and if this random number is smaller than a threshold, the node becomes a cluster head. Next, it advertises itself with constant energy via radio communication and waits for cluster members. The cluster members are the non-cluster head nodes, and each of them joins that cluster head's cluster whose advertisement was received with the highest energy (in general, this is the nearest cluster head's cluster). The properties of LEACH are that it flatly balances the energy consumption of the network, however, it uses only one-hop communication, the remaining energy of the nodes is not a parameter by the election (but it should be because of the increased energy needs of a cluster head node), and the protocol requires every node to be able to reach the base station in one hop, which is not generally true in sensor networks. Other related probabilistic cluster formation solutions can be found in [25, 24, 3, 40, 34, 23]. In [25, 24], one can read about a data gathering scheme, called PEGASIS, that forms a chain of nodes and elects a leader node for energy efficient collection of the sensors' measurements. In [3], the authors propose a technique to build k -hop clusters, i.e., where the cluster members are at most k-hops away from the cluster head. In [40], a LEACH-like idea is exploited considering the residual energy of the nodes as well. In [34, 23], the problem of unequal energy dissipation is considered in case of equally sized clusters. Therefore, in these latter papers, the authors propose to form unequal size clusters: smaller ones close to the base station and larger ones further from it, as the cluster heads of closer lying clusters have more load due to the message forwarding task for further lying clusters. The most important probabilistic solution for our purposes is LEACH [42], which can also be considered as the generalization of [17] and [40]. In LEACH, the cluster formation algorithm is more sophisticated; it elects the cluster heads based on their remaining energy and on a secondary parameter that can control the cluster density, the load balancing, and the amount of intra-cluster communication. The LEACH protocol seems to be a good trade-off between termination speed and cluster head distribution by allowing some communication between neighboring nodes. The simulation results of LEACH show that it out performs LEACH in terms of network lifetime and in ratio of energy dissipated for clustering. The list of papers that aim at network lifetime maximization using a deterministic aggregator node election method is quite extensive as well. In [14, 15, 41], one can find solutions for the mentioned problem assuming that the sensor network is heterogeneous, i.e., there are less energy constrained gateway nodes among the usual constrained sensor nodes that can help in cluster formation and in the routing of sensor messages. The authors of [9, 19] approach the clustering problem from data gathering point of view, and propose a centralized solution for near optimal scheduling of the message sending. In [12] , the problem of lifetime maximization is handled by balancing the load of cluster heads and by minimizing the total distance between sensor nodes and cluster heads. [16] tackles the same problem using fuzzy logic with the variables of energy, node concentration and node centrality with respect to the entire cluster. There are some papers that do not primarily aim at network lifetime prolongation, but at quick cluster formation. In [31], the authors consider event-driven sensor networks with high degree of spatial-temporal correlation. The main focus of the paper is on cross-layered design of localized algorithms for performing quick data aggregation and quick hierarchy formation allowing prompt response to queries. In [10], one can read about a fast clustering algorithm suitable for large scale sensor networks by its property of locality, scalability, and self-healing in case of node failures and newly deployed nodes. The following group of papers collects those works that aim at enhanced network management. For example, the objective of [21] is to show that one can efficiently compute an asymptotically optimal clustering, even when collision resistant packet forwarding is not ensured. The authors of [4] propose a hierarchical clustering approach, where the cluster heads are clustered again at the higher layer of the hierarchy. Here, the clustering problem is defined in a graph theoretic framework, and a distributed solution is presented that results in size-bounded clusters. The ACE algorithm [7] aims at forming minimally overlapping clusters with the help of cluster migration. The algorithm ends in constant time regardless of the size of the network and uses only local communications between nodes. Security is rarely considered in the cluster formation or aggregator node election problem. An exception is [33], where the authors deal with the issue of non-manipulable aggregator node election. When an attacker node is able to manipulate the aggregator election process, it is able to influence the operation of the network, for example, by electing itself as aggregator and maliciously filtering the sensors' measurements. Three independent countermeasures are proposed against such an attacker in [33], all of them based on distributed random number generation. Another exception is [26], where the author proposed a technique for resilient cluster formation, which consists of a neighbor validation module based on wormhole detection, a priority-based selection of the cluster head nodes, and a centralized detection module that aims at detecting the nodes that have an abnormally large number of neighbors (as these are most likely compromised). The papers listed above are all related to the aggregator node election problem assuming clustering. However, none of the above methods are able to guarantee a minimum distance between certain aggregators. However, in our motivating application area (i.e., reliable and persistent distributed data storage), backup aggregators must be chosen in such a way that they reside farther away from the primary aggregator than a certain disaster range. In [1], the authors detail an approach for aggregator node election that is able to guarantee this minimum distance, however, the proposed solution is centralized, and thus, its applicability is limited. On the contrary, PANEL can guarantee a minimum distance between aggregators in a distributed manner, because in PANEL, the aggregator nodes reside within fixed size clusters and a re elected locally without the need of a central controller. For instance, the minimum distance between two aggregators belonging to non-neighboring clusters is dx, where x is the number of clusters between the two aggregators, and d is the physical size of the cluster.

3. Proposed model – PANEL

Our Assumption

We assume that the sensor network consists of homogeneous sensors and non Homogeneous sensors (in terms of resources). The sensor nodes are deployed in a bounded area, and this area is partitioned into geographical clusters. We aim at electing a single aggregator per cluster. The density of the network is large enough so that the nodes within each cluster are connected when they use maximum power for transmission. In other words, there exists a route between any pair of sensors of a given cluster that contains only sensors from that cluster. This assumption on the connectivity within a cluster is crucial to the correct operation of PANEL_GEAR, and it can be satisfied by appropriately choosing the cluster size (given the deployment density of the network and the maximum power range of the nodes).

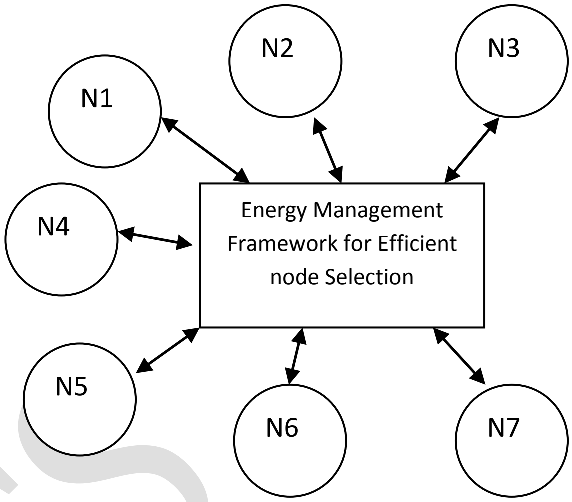


Figure 1: Energy Management of the Mobile ad-hoc network

We further assume the communication time between the sink and sensor nodes is negligible, as compared with the sink node's travelling time. Similarly, the delay due to multihop communications including transmission, propagation, and queuing delays is negligible with respect to the travelling time of the mobile sink in a given round. Each RP node has sufficient storage to buffer all sensed data. The mobile sink is aware of the location of each RP. All nodes are connected, and there are no isolated sensor nodes. Sensor nodes have a fixed data transmission range.

Definition 1 (Delay of data). The delay of data is defined as the time spent by the mobile sink moving from one sink site to the next sink site.

Definition 2 (Network lifetime (T)). The network lifetime (T) is defined as the elapsed time since the launch of this network till the instant that the first node dies.

Network Model and Assumption

The MANET has been modelled using a distributed routing protocol utilizing the diverse traffic handling by nodes through aware of their positions. Each node is supposed to be aware of its current node state and forwarding node state in order to route the data to the destination. Mobile ad-hoc networkdoes a packetization to transmit the data to a destination node through intermediate nodes.

Algorithm: PANEL Scheduling algorithm

Input: $\alpha = \{T=\sum t_k\}$

Output: $\alpha^{|} = \{T^{|}\}$

Divide Graph into connected sub graphs.

Apply the SSDR approach to each sub graphs and obtains the optimal sink path as well as corresponding routes.

Calculate the Source timings stayed in node and Travelling time from Source node to next node until destination node.

Choose the Longest network lifetime as best Data travelling Path

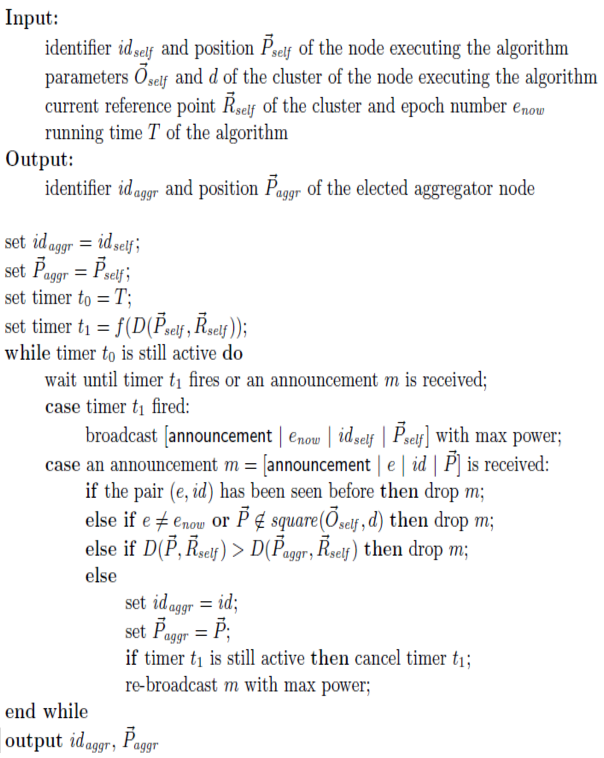
Calculate linear trajectory, Boundary trajectory and arbitrary trajectory values to prove the Optimal Sinks for Data Travelling. Begin

For k=1: k<m: k++ do

Where k = network life time

M is the mobility of the mobile Sink .

The pseudo-code of the aggregator election procedure of PANEL



```
Input:
    identifier $id_{self}$ and position $\vec{P}_{self}$ of the node executing the algorithm
    parameters $\vec{O}_{self}$ and $d$ of the cluster of the node executing the algorithm
    current reference point $\vec{R}_{self}$ of the cluster and epoch number $e_{now}$
    running time $T$ of the algorithm
Output:
    identifier $id_{aggr}$ and position $\vec{P}_{aggr}$ of the elected aggregator node

set $id_{aggr} = id_{self}$;
set $\vec{P}_{aggr} = \vec{P}_{self}$;
set timer $t_0 = T$;
set timer $t_1 = f(D(\vec{P}_{self}, \vec{R}_{self}))$;
while timer $t_0$ is still active do
    wait until timer $t_1$ fires or an announcement $m$ is received;
    case timer $t_1$ fired:
        broadcast [announcement | $e_{now}$ | $id_{self}$ | $\vec{P}_{self}$] with max power;
    case an announcement $m$ = [announcement | $e$ | $id$ | $\vec{P}$] is received:
        if the pair $(e, id)$ has been seen before then drop $m$;
        else if $e \neq e_{now}$ or $\vec{P} \notin square(\vec{O}_{self}, d)$ then drop $m$;
        else if $D(\vec{P}, \vec{R}_{self}) > D(\vec{P}_{aggr}, \vec{R}_{self})$ then drop $m$;
        else
            set $id_{aggr} = id$;
            set $\vec{P}_{aggr} = \vec{P}$;
            if timer $t_1$ is still active then cancel timer $t_1$;
            re-broadcast $m$ with max power;
end while
output $id_{aggr}$, $\vec{P}_{aggr}$
```

Clustering of the mobile Sink for Path Prediction

The intra-cluster routing protocol of PANEL can take advantage of the fact that the nodes within the cluster communicate during the aggregator election procedure. In particular, announcement messages containing the identifier and the position information of their sources are flooded in the cluster. This can be used to set up backward pointers towards the sources of the announcement messages in the routing tables of the nodes. More specifically, in PANEL_GEAR, every node that hears an announcement records the identifier and the position of the originator of the announcement as destination, it records the identifier of the node from which it received the first copy of the announcement as the next hop towards the recorded destination, and it computes and records the power level needed to transmit to this next hop node. The identifier of the next hop is obtained from the lower-layer (e.g., MAC) header of the message encapsulating the announcement. The computation of the required power level relies on the fact that the nodes

IJETS

transmit announcement messages with maximum power, and the receiving nodes can measure the power level with which they receive those messages.

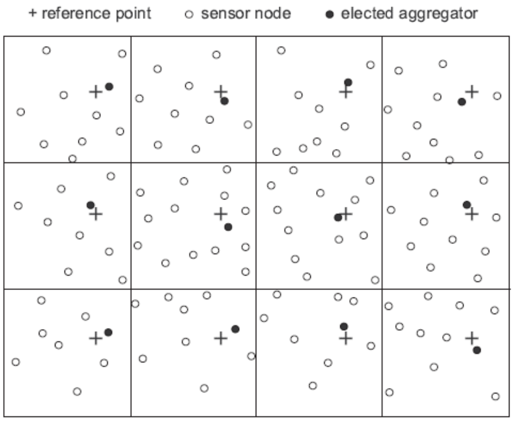


Figure 2: Illustration of the geographical clustering in PANEL

The aggregator node election procedure needs communications within the cluster. PANEL takes advantage of these communications and uses them to establish routing tables for intra-cluster routing. In particular, at the end of the aggregator node election procedure, the nodes also learn the next hop towards the aggregator elected for the current epoch.

PANEL also includes a position-based routing protocol that is used in inter-cluster communications. As the nodes are aware of their geographical position, this seems to be a natural choice that does not result in additional overhead. The position-based routing protocol is used for routing messages from a distant basestation or from a distant aggregator towards the reference point of a given cluster. Once the message enters the cluster, it is routed further towards the aggregator using the intra-cluster routing protocol based on the routing tables established during the aggregator node election procedure. Any position-based routing protocol can be integrated with PANEL; currently, we are experimenting with the Greedy Perimeter Stateless Routing (GPSR) protocol [7].

Finally, we want to point out that in PANEL, the reference points of the clusters are re-computed and the aggregator election procedure is re-executed in each epoch. This ensures load balancing in the sense that each node of the cluster can become aggregator with nearly equal probability. In addition, the nodes can accumulate information that they receive in the different epochs and use that for routing and intrusion detection purposes.

IJETS

The CH election procedure needs intra-cluster communications. PANEL communications to establish routing tables for intra-cluster routing. The intercluster routing protocol is used to route messages to and from a distant cluster. These messages can be queries from and responses to a distant base station, as well as backup messages destined to distant aggregators that contain replicated data. We recommend using a position-based routing protocol as the intercluster routing protocol for the following two reasons. First, PANEL already makes the assumption that the nodes are aware of their positions, and therefore, this position information can naturally be reused for routing purposes. Second, intercluster routing is concerned with messages that need to be routed (i) to the aggregator of a distant cluster or (ii) to a distant base station. Regarding case (i), in PANEL, the identifier of the aggregator node is not known explicitly outside the cluster, but, instead, one knows only the reference point to which the aggregator happens to be the closest node. Regarding case (ii), the query messages can contain the geographical position of the base station to which the responses should be sent back. Thus, in all cases, messages need to be routed towards a geographical position, and hence, position-based routing seems to fit best for inter-cluster routing in PANEL_GEAR. Apart from being a position-based routing protocol, we do not restrict the choice for inter-cluster routing in PANEL_GEAR.

4. Experimental Results

Implementation and Sensor Node Details:

The PANEL protocol was implemented and tested using construction of network model. The performance has been explained through the two important properties like node life time of networks.

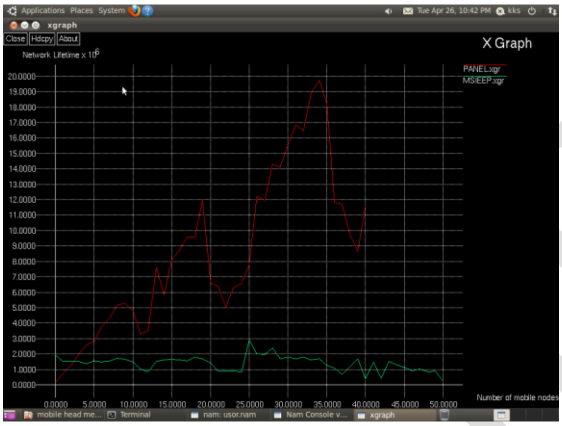


Figure: 3: Network Lifetime vs. No. of mobile Nodes against the Path Discovery

In Figure 3, Experimental results show the used no.of nodes and the throughput of the mobile sink in data covering. The performance of the algorithms is measured by the average convergence speed with respect to the number of sensors while retaining the long lifetime state and by the number of fitness evaluations. We presented a mathematical formulation that jointly considers different issues such as sink scheduling, data routing, bounded delay, and so on. The formulation is general and can be extended. However, this formulation is a MINLP and is time consuming to solve directly. As a result, the performance of the algorithm can be represented by the involving a mobile sink and the impact of network parameters (e.g., the number of sensors, the delay bound) on the network lifetime. The linear trajectory significantly outperforms the other two and would save a relatively long computational time.

Figure 4: End to End Delay of the Mobile Sensor nodes vs. Number of the node

In Figure 4, end to end delay and node life time has been calculated based on the transmission distance of the node. The protocol code is a simplified implementation of all the modules except the cluster head (CH). Due to physical limitations of sensor nodes and the difficulties in diverse area deployment, it is extremely difficult to perform as extensive evaluation as done in the simulation study. The aim of this experiment is to practically investigate the feasibility of the protocol. The motivation is neither to evaluate the scalability of the protocol nor to compare it with other protocols, which were already carried out in the previous section. An experimental network of 100 nodes was deployed with one source and two sinks, where one acts as primary and the other as secondary. We fixed the maximum transmission power to 2, resulting in a power range of a few tens of centimetres (less than 1 m). The

IJETS

source node generated a 20 bytes packet each second and transmitted it to the primary sink and possibly also to the secondary sink. This depends on the packet type, decided upon each transmission. 40 percent of the packets were regular, 20 percent were delay-sensitive, 20 percent were reliability-sensitive, and 20 percent were critical.

Conclusion

We described PANEL, a position-based aggregator node election protocol for wireless sensor networks. The novelty of PANEL with respect to other aggregator node election protocols is that it supports asynchronous sensor network applications where the sensor readings are fetched by the base stations after some delay. In particular, the motivation for the design of PANEL was to support reliable and persistent data storage applications, such as TinyPEDS. PANEL uses the position information of the nodes to determine which of them should become aggregator. PANEL ensures load balancing, meaning that each node has nearly the same chance to become aggregator, and it supports intra- and inter-cluster routing allowing sensor to aggregator, aggregator to aggregator, base station to aggregator, and aggregator to base station communications. Besides describing the operation of PANEL, we also evaluated its efficiency by means of simulations. In particular, we compared the cluster formation capabilities and the energy consumption of PANEL to that of LEACH, an aggregator node election protocol well-known from the literature. Our results show that PANEL behaves better than LEACH in both comparisons.

Reference

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] W. Song, R. Huang, M. Xu, B.A. Shirazi, and R. Lahusen, “Design and Deployment of Sensor Network for Real-Time High-Fidelity Volcano Monitoring,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1658-1674, Nov. 2010.

[3] C. Liu, K. Wu, and J. Pei, “An Energy-Efficient Data Collection Framework for Mobile Ad-hoc Networksby Exploiting Spatiotemporal Correlation,” IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 7, pp. 1010-1023, 2007.

[4] M. Perillo, Z. Cheng, and W. Heinzleman, “An Analysis of Strategies for Mitigating the Sensor Network Hot Spot Problem,” Proc. Second Int’l Conf. Mobile and Ubiquitous Systems (Mobiquitous ’05), July 2005.

[5] W. Wang, V. Srinivasan, and K. Chua, “Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks,” Proc. ACM MobiCom, Aug. 2005

[6] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and I. Robla, “A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends,” *Sensors*, vol. 9, no. 6, pp. 4728–4750, Jun. 2009.

[7] H. Chan and A. Perrig. ACE: An emergent algorithm for highly uniform cluster formation. In Proceedings of the First European Workshop on Sensor Networks, January 2004.

[8] Chipcon Corporation. CC1000 Datasheet. http://www.chipcon.com/files/CC1000 Data Sheet22.pdf.

[9] K. Dasgupta, K. Kalpakis, and P. Namjoshi. An efficient clustering–base d heuristic for data gathering and aggregation in sensor networks. In

Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2003.

[10] M. Demirbas, A. Arora, and V. Mittal. FLOC: A fast local clustering service for wireless sensor networks. In Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS)2004.

[11] S. Ganeriwal, S. Capkun, C.-C. Han, and M. B. Srivastava. Secure time synchronization service for sensor networks. In Proceedings of the 4th ACM workshop on Wireless security (WiSe), 2005.

IJETS