JOURNAL OF COMPUTING TECHNOLOGIES

ISSN 2278 – 3814

Available online at www.jctjournals.com

Volume 1 Issue 2, June 2012

Lifetime and Coverage Guarantees through Distributed life Time Maximization and Distributed Sensor Cover Algorithms in Wireless sensor networks

J.Jerophine Jeni* PG Student, D. Siva Senthil**,G.Kharmega Sundararaj#

, Department of Computer Science and Engineering

PSN College of Engineering and Technology, Melathediyoor, Tirunelveli, INDIA

*jeni2306@yahoo.com, **sivasenthil86dss@gmail.com

megaminindia@gmail.com,

Abstract— **In wireless sensor networks(WSNs),a large number of sensors perform distributed sensing of a target field. A sensor cover is a subset of the set of all sensors that covers the target field. The lifetime of the network is the time from the point the network starts operation until the set of all sensors with nonzero remaining energy does not constitute a sensor cover any more. An important goal in sensor networks is to design a schedule—that is, a sequence of sensor covers to activate in every time slot—so as to maximize the lifetime of the network. In this paper, we design a polynomial-time distributed algorithm for maximizing the lifetime of the network and prove that its lifetime is at most a factor O(log n * log nB) lower than the maximum possible lifetime, where n is the number of sensors and B is an upper bound on the initial energy of each sensor. Our algorithm does not require knowledge of the locations of nodes or directional information, which is difficult to obtain in Sensor networks. Each sensor only needs to know the distances between adjacent nodes in its transmission range and their sensing radii. In every slot, the algorithm first assigns a weight to each node that is exponential in the fraction of its initial energy that has been used up so far. Then, in a distributed manner, it finds an O(log n) approximate minimum weight sensor cover, which it activates in the slot.**

Keywords—***Approximation algorithms coordinate-free coverage, distributed algorithms, network lifetime, wireless sensor networks (WSNs)***

1. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions. The development of such networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control [1-2]. As depicted in Fig. 1, data collected by sensors is transmitted to a special node equipped with higher energy and processing capabilities called –Base Station‖ (BS) or –sink‖. The BS collects filters and aggregates data sent by sensors in order to extract useful information. WSNs have the potential to become the dominant sensing technology in many civilian and military applications, such as intrusion detection, environmental monitoring, object tracking, traffic control, and inventory management. In many of these applications, WSNs need to monitor the target field for detecting events of interest, e.g., the entrance of an intruder in intrusion detection applications. Widespread deployment of WSNs in target field monitoring is being deterred by the energy consumed in the monitoring process. The challenge is compounded by the fact that the sensors are battery-powered and owing to size limitations the sensors are battery-powered and owing to size limitations the sensors can only be deployed with low-lifetime batteries, most of which are not rechargeable. Thus a sensor cases to function (e.g., monitor)once its battery expires, and oftentimes, sensors whose batteries have expired cannot be easily replaced owing to logistics issues such as remoteness or inaccessibility of distribution areas. Thus, the success of the WSN technology is contingent upon developing strategies for intelligently using the available sensors so as to maximize the duration for which the entire target field is monitored by sensors. This duration, referred to as the network lifetime, is an important performance metric for the network as the coverage of the entire target field is essential for reliable detection of events of interest. Owing to large-scale availability of low-cost sensors, sensors are often deployed with some redundancy, that is, several locations in the target field can be monitored by multiple sensor . Lifetime of the WSNs can be substantially enhanced by intelligently activating the sensors that monitor the target field at any given time. We seek to maximize the lifetime of sensor networks by designing algorithms that dynamically activate sensors based on their residual energy content. The algorithm we develop is completely distributed, does not need to know the coordinates of any sensor, and provides provable guarantees on the attained lifetimes.

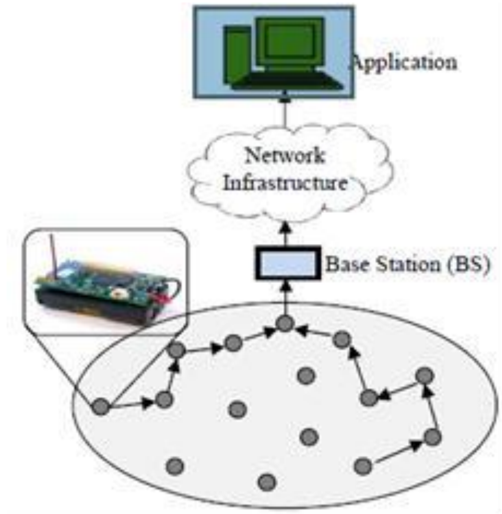

Fig-1 Wireless sensor networks

1.1. RELATED LITERATURE

Coverage, connectivity, and lifetime maximization for WSNs have received considerable attention in the last few years. Comprehensive surveys can be found in [1] and [2].Most of the existing papers focus on the coverage and connectivity aspects [3],[4],[5],and typically propose computational geometry-based approaches for discovering coverage holes and ensuring connectivity. An interesting connectivity property has been proven in [6] and [7], which shows that if the transmission radius of each node is at least twice of its sensing radius then coverage implies connectivity of the sensor network. We make the same assumption and, therefore, seek to maximize lifetime while guaranteeing coverage without explicitly considering connectivity.

We now summarize the papers that propose topology control solutions that maximize the network lifetime by scheduling the active periods of the sensors while preserving coverage and connectivity requirements. In [9], Cardei et al. addressed the problem of lifetime maximization when only a given set of targets need to be covered. They showed that the problem is NP-hard and provided heuristic sensor activation algorithms based on linear programming relaxations. They also proposed a greedy heuristic activation scheme that, at each round, seeks the minimal set of sensors that covers all the targets. They evaluated the lifetimes attained by the heuristic solutions using simulations, but did not provide provable guarantees on the lifetime of these schemes.

Wang et al. [7] showed that the monitoring area is covered if all intersection points between sensing borders of sensors and those between sensing borders of sensors and the monitoring area are covered .They also provided a distributed algorithm to activate a minimum set of sensors while ensuring coverage and connectivity. However, the algorithm in [7] assumes knowledge of coordinates of nodes and does not provide provable guarantees on the network lifetime.
The scheme proposed by Berman et al. [10] provides provable guarantees on the network lifetime while ensuring coverage of the target field. They have provided a centralized algorithm that attains a network lifetime that is within O(log n) of the maximum possible lifetime, where n is the number of sensors. This algorithm determines how to activate sensors based on an approximation solution of a linear program that requires complete knowledge of network topology, coordinates of sensor locations and initial energy of sensors.

Such linear programs can clearly be solved only by a central entity that knows all of the above, which is hard to realize in practice. Also, the sensors rarely know their precise locations since WSNs usually do not have access to global positioning systems(GPSs). Several sensor positioning systems [11], [12] have been proposed in the literature for learning the locations, without manual configuration or the use of GPS receivers. However they provide only coarse location estimations in practical settings [13]. Note that several coverage verification algorithms that do not assume knowledge regarding the locations of the sensors exist [4],[14],,but these papers do not provide a distributed coordinate-free sensor activation scheme that provides provable guarantees on the network lifetime

Finally, W_u et al.[8]considered a different notion of lifetime in a recent paper: the maximum time until which all nodes in the data aggregation tree of choice remain operational (a node in this case consumes energy only during communication). Since we focus on the energy consumed in sensing, our notion of lifetime, the problem formulation, and solution techniques differ substantially.

2. ALGORITHM OVERVIEW

We now describe the Distributed Lifetime Maximization (DLM) algorithm that we propose. In this section, we present a brief overview of the individual building blocks in DLM.

Our algorithm consists of an initialization phase and an activation phase. The initialization phase is executed once at the beginning of the network operation and informs the nodes of some network parameters. Every node executes the activation phase at the beginning of each subsequent time slot and decides whether to activate itself in the slot based only on the state information in its neighborhood. We now describe these phases, and introduce some new terminologies toward that end.
Consider a sensor cover C, and let sensor u have weight Wu, a positive real number. The weight of the sensor cover C is the sum of the weights of the sensor cover C is the sum of the weights of the sensors in C.

Definition (Minimum Weight Sensor Cover):A minimum weight sensor cover is a sensor cover that has the minimum weight among all sensor covers.

2.1. .INITIALIZATION PHASE

An initialization phase is executed at the beginning of the network operation , i.e., at time t=0. During the initialization phase, each sensor u acquires the following local information: 1) the set P_u of intersection points that it covers; 2) the identities of the sensors in T_u; and 3) the intersection points in P_u that are covered by each sensor in T_u.

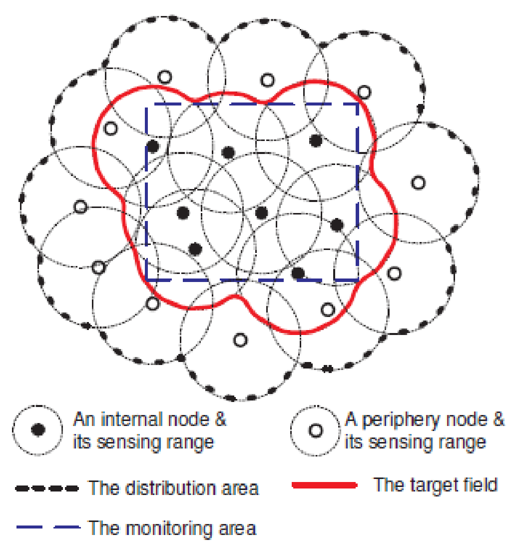

Figure 2: An example of a small WSN and its target field

Each sensor u learns this information in a distributed manner by merely communicating with its neighbors and using only localized distance information. In addition each sensor learns the following global network parameters: 1)n ,the total number of sensors; 2)the maximum amount B of the initial energy of any sensor.

2.2. ACTIVATION PHASE

The activation phase is executed at the beginning of each slot .We describe the computations in slot j.

1) Weight Assignment: Let $b_u(j)$ be the amount of energy of energy of sensor u that has been consumed in slots 1,….j-1 then at the beginning of slot j, sensor u has already consumed $lu(j) = bu(j)/Bu$ .fraction of its energy. If $b_u(j) > B_u$ -1, i.e., sensor u does not have enough energy to monitored its sensing range throughout slot j, then it assigns itself a weight of the beginning of slot j.

2) Sensor Activation: Sensors that have infinite weights at the beginning of slot j do not activate themselves in slot j. Among the rest, sensors are activated. so that the subset of activated sensors, S(j),constitutes an O(logn)-approximate minimum weight sensor cover .The sensors that do not activate themselves in slot j sleep in slot j.

Intuitively, DLM has been designed so that the sensors are activated so as to cover the target field whenever possible and the sensors that have large residual energy are preferentially selected. We will later prove that the lifetime of DLM is at least $1/O(\log n)(\log nB)$times that of the maximum lifetime of the network

When there does not exist a sensor cover any more such that each sensor in the cover has non zero energy ,the network lifetime is considered terminated .After the network lifetime termination, we cannot provide any guarantee on the target field coverage, although the sensors with finite weights continue to execute the algorithm and cover their sensing ranges.

Note that each sensor can determine its weight based only on local information. In the next section, we show how each sensor can execute the activation phase using distributed computations based only on local information obtained from its neighbors.

3. DISTRIBUTED SENSOR ACTIVATION

We now describe an algorithm, which we call the Distributed Sensor Cover algorithm using which sensors can determine, using simple distributed computations, whether to activate themselves in each slot. Clearly, we need to design a sensor cover with guarantees on its weight using distributed computations. Note that a sensor cover is an instance of a set cover, and centralized algorithms that attain an O(log n)-approximate set cover are well known. We instead accomplish the same goal using distributed computations only, extending the design technique developed by subhadrabandhu et al [15]. For the dominating set problem. We next describe our approach.

The sensor cover in each slot j is iteratively computed in an asynchronous manner. At the beginning of the activation phase in each slot, all the sensors with finite weights are contending for staying active in the slot. At any time during the activation phase, each contending sensor u determines the number of intersection points in P_u that have not yet been k-covered by the set of activated sensors and computes its activation preference ratio as the ratio between its weight in slot j, $W_u(j)$, and the above number. Each contending sensors u communicates its activation preference to the sensors in T_u at the beginning of the activation phase and each time that its value changes. Note that the letter occurs only when one of u's neighbors T_u becomes active. A contending sensor u activates itself once it detects that it has a lower activation preference then all contending sensors in T_u. Each sensor u that activates itself informs other sensors in T_u accordingly. Once a sensor u detects that all the intersection points P_u in its sensing range k-covered by the already active sensors in T_u, it updates its neighbors and enters a sleep mode. The activation process, in each slot terminates after each sensor decides whether to stay active or entire a sleep mode.

Clearly, each sensor can execute the aforementioned computations based only on locally available information and the information it acquires in the Initialization phase. Recall that a sensor u enters a sleep mode only after all the intersection points P_u in its sensing range are already k-covered. Thus, according to Theorem 1,during the lifetime of the network the subset of sensors activated at the end of the activation phase in each slot j,S(j), induces a sensor cover for the network. Moreover, we will later prove that S(j) constitutes an O(log n)-approximate minimum weighted sensor cover.

As mentioned, well-known centralized algorithms such as the one in can also be used to find an O(log n)-approximate sensor cover. We now compare our DLM algorithm, which runs a distributed sensor cover selection algorithm in each slot, to two natural implementations of any given centralized sensor cover computation algorithm. In the first implementation, at the beginning of the network operation, a central controller such as a base station: 1) collects the required information from each sensor; 2) computes a complete schedule of the set of sensors to activate in each slot ; and 3) distributes this schedule to the sensors. Note that, in practice, sensors are prone to failure due to hardware malfunction or damage from the environment. Since the centralized algorithms computes a complete schedule only at the beginning of the network operation, sensor failure during network operation may cause coverage holes to form, which could persist for a long time. On the other hand, the DLM algorithm is more robust to sensor failure: it selects a sensor cover at the beginning for every slot from among the operating sensors at that point in time so even if sensors fail in a slot, the resulting coverage holes will last only until the end of that slot.

The following alternative centralized implementation is more robust to sensor failure then the above centralized implementation. At the beginning of every slot, each operating sensors sends a message to the base station then selects a sensor cover and informs each sensor whether to be active or not in that slot. However , in each slot, several network-wide message exchanges are required, which taxes the network resources. This overhead is not incurred under our DLM algorithm since only a single network-wide broadcast is required at the beginning of the network operation and, subsequently, messages only need to be exchanged locally in each slot.

4. INITIALIZATION PHASE

During the initialization phase, each sensor u gains the knowledge of: a) the set P_u of intersection points that its covers b)the identities of the sensors in $T_{u,}$ which share intersection points with node u; and c) the set P_u,v of the intersection points in P_u that are covered by each sensor v in Tu. We show that u can determine the above using localized computations based on simple geometric properties.

4.1. OVERVIEW

We assume that during the system activation every sensor u initially evaluates its distances to each one of its neighbors in N_u. A major challenge in the initialization process is determining a unique identification for each intersection points are unknown and cannot be used as identifiers. To overcome this difficulty, every intersection point of any pair u ,v of intersecting sensors is identified by a triplet IP(u, v, i), where u is the sensor with lower id, v is the sensor with higher id, and I denotes the point index.

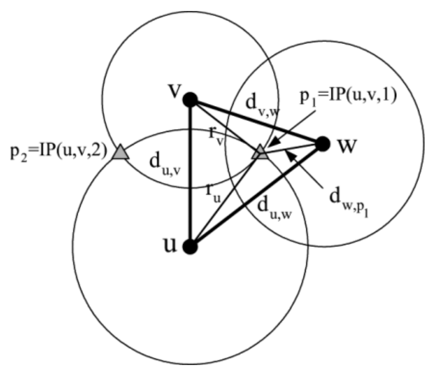

Fig.3. Pair of intersecting nodes and their intersection points

Since every pair u, v of intersecting sensors has two common intersection points, the node with the lower id, says u, arbitrarily determines the index I of each point. In addition, u also calculates the set Spi for both points pi=IP(u, v, i), i=1, 2, and communicates these sets to its neighbors, including node v. This ensures that each calculated set Sp corresponds to a single intersection point that is uniquely defined. We describe the calculation of such sets Sp .

5. SYNCHRONIZATION

We now discuss some synchronization-related aspects of our algorithm. At the beginning of the network operation, the sensors synchronize their clocks using a distributed algorithms; for a survey of synchronization algorithms for wireless sensor networks. Then, each sensor exchanges distances information with its neighbors and carries out the initialization base.

Subsequently, the sensors run the DSC algorithms in every slot.

Note that the only synchronization requirement in the DSC algorithm used for sensor cover computation in each slot is that sensors need to have synchronized clocks at the beginning of each slot. Thereafter, within the slot, the operation can be completely asynchronous. If the clocks of different sensors are accurately synchronized, then the DSC algorithm in works correctly; otherwise, the following problem occurs. Recall that at the beginning of the DSC algorithm, a sensor u sends its initial ap to each neighbor and then waits for the initial ap's of its neighbors. If u does not receive an initial-ap message from a neighbor w within a given time period then it considers w to be inactive. Now, if the clock of a neighbor w lags behind u's.

We now describe a minor modification, with which the DSC algorithm works correctly even in the presence of discrepancies between the clocks of different sensors .when a sensor u finds that a slot has started according to its own clock, it listens to the channel for a duration to. Then it sends out its initial ap and again listens to the channel for a duration to. If it does not receive an initial-ap message from a neighbor w in any of the two intervals, it assumes that w is inactive. Thus each node receives the initial ap of every other contending node at the end of the second interval.

6. DETECTION OF LIFETIME TERMINATION

We now augment our scheme with a simple distributed mechanism for detecting the termination of the network lifetime. By definition, the network lifetime terminates when there no longer exists a sensor cover such that every sensor in the cover has nonzero energy. Thus, from theorem 1, the network lifetime ends once one of the intersection points in the IP set P cannot be k-covered by the sensors that still have nonzero energy. Note that every point p is included in the closure of the sensing range of at least one internal node. Thus once an internal node u detects that u itself and all its neighbor in Nu have already declared that they are either active or in sleep mode. Node u informs the administrators about the coverage hole once this test fails.

7. SCHEME ANALYSIS

We now prove correctness and performance guarantees for the DLM algorithm. We prove the guarantees for DSC which DLM invokes.

7.1. DSC ALGORITHM ANALYSIS

We prove that DSC computes an O(log n)-approximation minimum weight sensor cover. Note that all the proofs allow for arbitrary but finite transit times of status update messages transmitted by nodes to their neighbors.

Theorem 2: At every activation phase: 1)DSC computes a sensor cover if there is no coverage hole; 2) DSC terminates in at most 2nV time if V is an upper bound on the transit delay of status update messages between the neighbors; and 3) DSC terminates in finite time if the transit delays are finite but cannot be upper-bounded

We omit the proof due to space constraints.

Remark 1: note that if, in a particular execution, DSC finds a sensor cover with n sensors, then the bound in 2) in theorem 2 can be improved to show that DSC terminated in at most 2nV time. Also, note that the bound of 2nV is not tight because sensors do not activate themselves serially, but sets of sensors activate themselves in parallel. In practice, DSC converges in a time that is much lower than this bound.

Now, recall that finding a minimum weight sensor cover is an instance of the minimum weight set cover problem. We now briefly describe the well-known greedy Centralized Set Cover(CSC) algorithm that computes an O(log n)-approximate minimum weight set cover. At each iteration, it selects the sensor that has the lowest ap among all the sensors, where ap is defined in the same way as for DSC, and then updates the ap's of the unselected sensors. This process continues until the set of selected sensors constitutes aq sensor cover.

The Distributed Sensor Cover (DSC) algorithm of sensor u

Definitions:

- Let $UC_u \subseteq P_u$ be the set of intersection points that have not yet been k-covered by the set of activated sensors.
- Let $CT_u \subseteq T_u$ be the set of contending neighbors of sensor u.

Begin

```
if wu(j) = ∞ or Pu = ∅ then
   mode = sleep
   Return mode
else
   mode = contending
   UCu = Pu
   CTu = Tu
   Send My-Init-AP(apu) message to every sensor w ∈ Tu
   Receive My-Init-AP(apw) message from every sensor w ∈ Tu
      // If My-Init-AP message not received from a sensor w ∈ Tu
      // within a given time period, then w is considered inactive
      // and it is removed from CTu.
   if (CTu == ∅ or apu < apw for every w ∈ CTu) then
      mode = active
      Send an I-am-Active message to every sensor w ∈ CTu.
   end if
   while mode == contending and upon reception of a message
   M from sensor v ∈ CTu do
      if the received message M is I-Am-Active then
         CTu = CTu − {v}
         // Let NCu ⊆ UCu ∩ Pu,v be the set of intersection
         // points that are k-covered (after
      end if
   end while
   Return mode
```

7.2. DLM ALGORITHM ANALYSIS

We now prove an approximation ratio for the lifetime attained by the DLM algorithm .Our analysis is similar to the ones used by Aspnes et al. [8] for online machine scheduling and virtual circuit routing problems, and Awerbuch et al[3], [4] for the online virtual circuit routing problem.

A sensor that is active in a slot consumes one unit of energy, and a sensor in sleep mode consumes one unit of energy, and a sensor in sleep mode consumes no energy. Finally, for proving the approximation ratio, we additionally assumes that the initial energy of each sensor is large enough. A modified version of DLM that will be used to prove an approximation ratio for DLM.

Note that DLM-T differs from DLM in the following ways:
1) the criterion it uses to declare the network as dead;2)it does not use a weight for a sensor u width 0 remaining energy, 3) it considers all nodes in the sensor-cover selection process, whereas DLM considers only those that have at least one unit of energy remaining. It is therefore not clear whether DLM-T selects nodes that have at least one unit of energy left. The next lemma, however, shows that this is indeed the case.

8. SIMULATION

We now evaluate the performance of DLM using simulations. We consider a WSN with n sensors, each with an initial energy of B units, sensing and transmission radii of 10 and 22 units respectively, deployed uniformly at random in a 50×50 units^2 target field, and examine the lifetime attained by DLM as functions of n and B. Each time slot was 1 unit long.We compared the lifetimes of the network under three algorithms: the DLM algorithm (Fig. 2), the Garg-Konemann (GK) algorithm [11] and a heuristic proposed in [13, 21] that we denote by Min-Num. At every slot, Min-Num finds a sensor cover with the minimum number of nodes (up to an $O(\log n)$ factor) and activates it. GK [11] generates a sequence of sets of weights to assign to the sensors and finds minimum weight sensor covers for each set of weights. When the initial energy of each sensor is the same, each sensor cover selected by GK is activated for an equal amount of time, which is a monotonically increasing function of an input parameter s. Thus, the number of sensor cover computations per slot, and hence the computation time required for GK, increases as s decreases

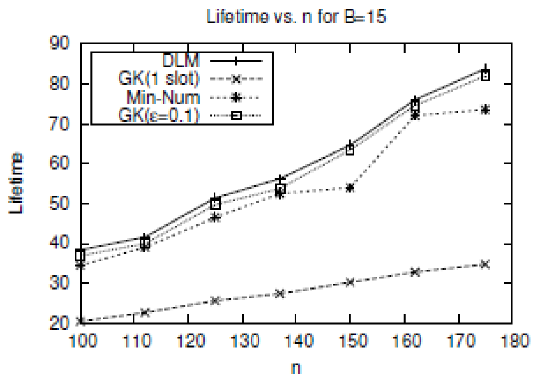

Figure 4: Plot of lifetimes of DLM, GK (1 slot), Min-Num and GK $(s = 0.1)$ vs. n for $B = 15$ units

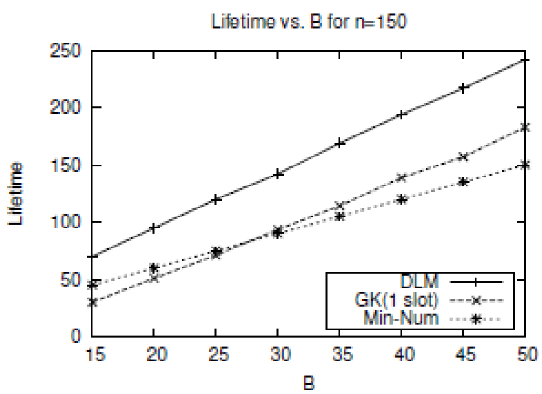

Figure 5: Plot of lifetimes of DLM, GK(1 slot) and Min-Num vs. B for $n = 150$

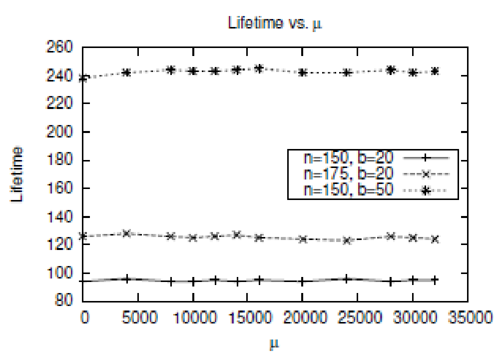

Figure 6: Plot of lifetimes of the DLM algorithm vs. μ for $(n = 150, B = 20)$, $(n = 175, B = 20)$ and $(n = 150, B = 50)$

9.CONCLUSION

We designed a distributed coordinate-free algorithm for attaining high lifetimes in sensor networks, subject to ensuring the k-coverage of the target field during the network lifetime. We proved that the lifetime attained by our algorithm approximates the maximum possible lifetime within a logarithmic approximation factor.

REFERENCES

[1] M. Cardei and J. Wu, Coverage in Wireless Sensor Networks, Hand-book of Sensor Networks. Boca Raton, FL:CRC Press,2004.
[2] L. Wang and Y.Xiao, –A survey of energy-efficient scheduling mechanisms in sensor networks,‖ Mobile networks application vol. 11, pp. 723-740,2006.
[3] C-F. Huang and Y.-C. Tseng, –The coverage problem in a wireless sensor network‖, in Proc. ACM WSNA, Sep. 2003, pp. 115-121.
[4] G. Kasbekar, Y. Bejerano, and S. Sarkar, –Generic coverage verification without location information using dimension reduction,‖ in Proc. WiOPT,Seoul,Korea, Jun. 2009, pp. 1-10.
[5] C.Zhang, Y.Zhang, and Y.Fang, –Detecting coverage boundary nodes in wireless sensor networks,‖ in Proc. IEEE ICNSC, Apr. 2006, pp.868-873.
[6] H. Zhang and J. C. Hou, –Maintaining sensing coverage and connectivity in large sensor networks,‖ Int. J. Wireless Ad hoc sensor Networks vol. 1, no. 1-2, pp. 89-123, Jan.2005.
[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless. And C. Gill, –Integrated coverage and connectivity configuration in wireless sensor networks,‖ in Proc. ACM SenSys, Los Angeles, CA, Nov. 2003, pp. 28-39
[8] Y. Wu, S. Fahmy, and N. Shroff, –On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm,‖ in Proc. IEEE INFOCOM, Phoenix, AZ, Apr. 2008, pp.356-360.
[9] M. Cardei, M.T. Thai, Y.Li, and W.Wu, –Energy-efficient target coverage in wireless sensor networks ,‖ in Proc. IEEE INFOCOM,Miami, FL, Mar. 2005, vol. 3, pp.1976-1984.
[10] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, –Efficient energy management in sensor networks,‖ in Ad Hoc and sensor Networks, Y. Xiao and Y. Pan, Eds. Commack, NY:Nova, 2005.
[11] D. Niculescu, –Positioning in ad hoc sensor networks,‖ IEEE Netw.vol. 18, no. 4, pp. 24–29, Jul.–Aug. 2004.
[12] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, –Locating the nodes: Cooperative localization in wireless sensor networks,‖ IEEE Signal Process. Mag., vol. 22, no. 4, pp. 54–69, Jul. 2005.
[13] Q. Shi, S. Kyperountas, N. S. Correal, and F. Niu, –Performance analysis of relative location estimation for multihop wireless sensor networks,‖ IEEE J. Sel. Areas Commun., vol. 23, no. 4, pp. 830–838, Apr.2005
[14] Y. Bejerano, –Simple and efficient k-coverage verification without location information,‖ in Proc. IEEE INFOCOM, Phoenix, AZ, Apr. 2008, pp. 291–295.
[15] D. Subhadrabandhu, F. Anjum, S. Kannan, and S. Sarkar, –Domination and coverage guarantees through distributed computation,‖ in Proc. 43rd Annu. Allerton Conf. Commun., Control Comput.,Monticello, IL, Sep. 28–30, 2005, pp. 205–217.