Transactions Letters

Clustering Algorithm in Initialization of Multi-Hop Wireless Sensor Networks

Peng Guo, Tao Jiang, Kui Zhang, and Hsiao-Hwa Chen

***Abstract*—In most application scenarios of wireless sensor networks (WSN), sensor nodes are usually deployed randomly and do not have any knowledge about the network environment or even their ID's at the initial stage of their operations. In this paper, we address the clustering problems with a newly deployed multi-hop WSN where most existing clustering algorithms can hardly be used due to the absence of MAC link connections among the nodes. We propose an effective clustering algorithm based on a random contention model without the prior knowledge of the network and the ID's of nodes. Computer simulations have been used to show the effectiveness of the algorithm with a relatively low complexity if compared with existing schemes.**

***Index Terms*—Wireless sensor network, multi-hop network, clustering algorithm, contention channel.**

I. INTRODUCTION

MANY protocols have been developed for their applications in wireless sensor networks (WSNs) to improve their overall system performance. However, it is noted that most of them may not work well in particular at the initial stage of a newly deployed WSN, because a WSN has not been adapted to the operational environment and the sensor nodes may not have enough knowledge about the whole network, including the number of their neighbor nodes, the network topology, or even the ID's of the nodes. In such a circumstance, it is normally very hard for a WSN to provide reliable point-to-point connections among the nodes, making it impossible to enable any higher layer protocols. Therefore, the initialization (such as clustering initialization, etc.) should be fulfilled successfully before the protocols in different layers can work properly in a WSN.

It is well known that one of the major issues in the node/network initialization process is to establish an initial infrastructure as quickly as possible such that the network can provide effective and reliable radio connection links to the nodes. Therefore, as far as initialization in multi-hop wireless sensor networks is concerned, one of the most challenging issues is to select the cluster leaders among numerous newly deployed sensor nodes.

Many clustering algorithms for WSNs have been reported [1]–[6]. Some of them aimed to preserve and balance the energy consumption of the whole network using a cluster-based architecture to prolong the network lifetime, while the others aimed to provide an efficient aggregation data rate according to different applications. However, none of these clustering algorithms were designed primarily for the initialization process of sensor networks. It was always assumed in the literatures that there have been reliable communication links between every pair of adjacent nodes, and thus the MAC layer functionalities could have been fully established. However, under the circumstance of initialization stage, the sensor nodes have very little knowledge about the whole network, and they may even have no assigned ID's. Without the coordination of an MAC layer, it is very difficult for them to successfully transmit a message or acknowledge a message. Hence, it can be seen that an effective clustering algorithm for initialization period should coordinate both cluster leader election and packet transmission in the absence of MAC layer of the network. For this reason, most of the existing clustering algorithms may not work properly in the initialization process.

Recently, some works have been reported as an effort to study the initialization processes for different networks in the literature [7]–[10]. However, most of them addressed the issue only for single-hop wireless networks, in which the channel state (i.e., busy or idle status) is kept the same for all sensor nodes. Therefore, the results obtained from these works can not be applied directly to a multi-hop wireless network due to the problems related to the "hidden nodes" effect. There are some other works which studied in particular the initialization process in a multi-hop WSN [11]. However, they were focused mainly on the initial transmission range adjustment and ID assignment.

To our best knowledge, there are very few works particularly addressing the problem of clustering in initialization of multi-hop WSNs, except [12] which assumed that the sensor nodes have three independent communication channels

---

Manuscript received January 11, 2008; revised November 19, 2008, February 26, 2009, and April 14, 2009; accepted August 18, 2009. The associate editor coordinating the review of this letter and approving it for publication was Q. Zhang.

P. Guo, T. Jiang, and K. Zhang are with Wuhan National Laboratory for Optoelectronics, Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China (e-mail: guopeng@mail.hust.edu.cn; Tao.Jiang@ieee.org; kylelxx@gmail.com).

H.-H. Chen is with the Department of Engineering Science, National Cheng Kung University, Taiwan (e-mail: hshwchen@ieee.org).

The work presented in this paper was supported in part by the NSF of China under the Grants 60903171 and 60872008, the Program for New Century Excellent Talents in University of China with Grant NCET-08-0217, and Taiwan NSC Grant NSC 98-2219-E-006-011.

Digital Object Identifier 10.1109/TWC.2009.12.080042.

(one for contention, and the other two for broadcasting), and thus the sensor nodes can send packets through three different channels with different probabilities according to their different states. In [12], an equivalent scheme was also suggested for a sensor network with a single communication channel. Another clustering algorithm for initialization with a single communication channel was proposed in [13]. However, the time complexity of the algorithm suggested in [13] is much higher than that of the scheme reported in [12]. Therefore, in this paper, we will focus on the comparisons between the performances of our proposed scheme and that proposed in [12] for the sake of fairness.

In the algorithm presented in this paper, we use a special data communication model to emulate the packet transmissions in initialization stage where no MAC layer is available and nodes have little knowledge about the newly deployed network. In the model, each node sends packets randomly and the duration needed by a node to successfully receive a packet is calculated even if the density of local deployment is dynamic. Our algorithm works without the knowledge about the whole network, except for the average node deployment density which can be determined by the requirements of different applications. The computer simulations have been conducted and validated, showing that our proposed clustering algorithm can provide a desirable steady-state performance with a relatively low complexity if compared with that reported in [12].

II. Proposed Clustering Algorithm

To study the clustering initialization problem for a multi-hop wireless sensor network, we should make some assumptions first. Let us consider N sensor nodes that are initially scattered randomly in a square area of X. Then, it is assumed that 1) the network coverage is modeled as an unit disk graph [14], and a transmission will be successfully performed at a sensor node if and only if one of its neighbors sends packets; 2) sensor nodes do not have the collision detection capability; 3) we further assume that the time is slotted [7], [9], [11], just for analytical simplicity; 4) each sensor node knows the information about the average deployment density which may be determined by the requirements based on a particular application; 5) the newly deployed sensor nodes do not have ID's which need to be assigned according to the scale of an actually deployed network, as the length of a perpetual and unique ID could be too long and the cost could be too high if compared with the small amount of data conveyed in the packets.

According to [15], the distribution of sensor nodes converges to a two dimensional Poisson point process in X. Therefore, a sensor node can get the maximal number of its neighbors Δ according to the average node deployment density with an accuracy not lower than $1 - \varepsilon$ [16], where $\varepsilon > 0$ and $\varepsilon \to 0$. Hence, we can proceed to propose the clustering initialization algorithm as follows.

A. Proposed Algorithm

The primary goal of the proposed cluster initialization algorithm discussed in this paper is to elect cluster leader nodes, and the left-over non-leader nodes are denoted as slave nodes. Therefore, for any one sensor node present in a WSN, it should become either a leader node or a slave node after the execution of the algorithm, and there should not be any other leader nodes within a certain transmission range of a given leader node.

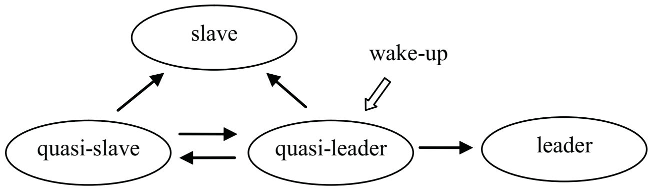

Fig. 1. The relationship between four sensor node states.

In the proposed algorithm, we assume that each sensor node has a timer. Moreover, we define four states of the nodes as follows: leader, quasi-leader, quasi-slave, and slave, whose relationships is illustrated in Figure 1. If a sensor node is an either leader or quasi-leader, it can send packets with probability p during every time slot. Otherwise, if a sensor node is a quasi-slave or slave node, it can only receive packets. Assume that all sensor nodes are quasi-leaders at the beginning of the network deployment stage, and each node has a timer which increments by one during each time slot. Then, they should run the proposed algorithm described in Algorithm 1 as follows. It can be seen that the algorithm ends when all nodes turn to leaders or slaves.

B. Theoretical Analysis

To get a deep insight into the effectiveness of the proposed clustering algorithm, we would like to give the detailed theoretical analysis including its validity and integrality with two Lemmas deduced as follows.

Lemma 1: Assume that each sensor node in a WSN sends a packet with a probability $p = 1 - \sqrt[1-\Delta]{\Delta}$ in every time slot. Then, any sensor node can correctly receive at least one packet from its neighbors with a probability not less than $1 - N^{-3}$ after $T_1 = \frac{3 \ln N}{p}$ time slots.

Proof: Consider a sensor node X_i $(i = 1, 2, \cdots, N)$ with D_i $(1 \le D_i \le \Delta)$ neighbors in the WSN. Let P_{suc} be the probability that X_i correctly receives a packet in a time slot, and P_{no} denote the probability that X_i does not receive packets in the subsequential T time slots. Since X_i correctly receives a packet in a time slot only if one of its neighbors sends packet, we have

$$\begin{cases} P_{suc} = C_{D_i}^{1} p \left(1 - p\right)^{D_i - 1} = D_i p \left(1 - p\right)^{D_i - 1} \\ P_{no} = \left(1 - P_{suc}\right)^{T} \le \exp(-T P_{suc}) = N^{-3} \end{cases} \tag{1}$$

Thus, we can choose $T = \frac{3 \ln N}{P_{suc}}$.

To calculate the upper bound of T for any value of $D_i \in (1, \Delta)$, we should discuss the monotonicity of P_{suc} with D_i. It can be found that P_{suc} always monotonically increases when $D_i \in (1, -\frac{1}{\ln(1-p)})$, whereas it monotonically decreases when $D_i \in (-\frac{1}{\ln(1-p)}, \Delta)$. Therefore, the minimum of P_{suc}

Algorithm 1 Cluster leader election process.

```
Fun_send(): Send packet with probability $p$ in the current time slot;
Fun_recv(): Receive packet in the current time slot; and return the state of the sender;
upon wake-up do:
state of node $i$: $S_i = quasi-leader$;
while (1) do
    $timer++$;
    if $S_i = quasi-leader$ then
        Fun_send();
        if Fun_recv() = $quasi-leader$ then
            $S_i = quasi-slave$; $timer = 0$;
        else if Fun_recv() = $leader$ then
            $S_i = slave$; break;
        end if
        if $timer = 3\Delta \ln N$ then
            $S_i = leader$; $timer = 0$;
        end if
    end if
    if $S_i = quasi-slave$ then
        if Fun_recv() = $quasi-leader$ then
            $timer = 0$;
        else if Fun_recv() = $leader$ then
            $S_i = slave$; break;
        end if
        if $timer = 3\Delta \ln N$ then
            $S_i = quasi-leader$; $timer = 0$;
        end if
    end if
    if $S_i = leader$ then
        Fun_send();
        if $timer = 3e\Delta \ln N$ then
            break;
        end if
    end if
end while
```

is obtained when $D_i = 1$ or $D_i = \Delta$, and T gets its maximum accordingly. Hence, we have

$$\begin{cases} T\mid_{\max} = \dfrac{\ln \frac{1}{\varepsilon}}{p}, \text{for } p < 1 - \sqrt[1-\Delta]{\Delta} \\ T\mid_{\max} = \dfrac{\ln \frac{1}{\varepsilon}}{\Delta p}\left(1-p\right)^{1-\Delta}, \text{otherwise} \end{cases} \tag{2}$$

Therefore, when we have

$$T_1 = \max\left\{\frac{3\ln N}{\Delta p}(1-p)^{1-\Delta},\ \frac{3\ln N}{p}\right\},$$

Lemma 1 can be true. Furthermore, it is easy to see that T_1 gets its minimum value $\frac{3\ln N}{p}$ when $p = 1 - \sqrt[1-\Delta]{\Delta}$. ■

From the proof of Lemma 1, some conclusions can be drawn as follows. 1) No matter how many node's neighbors may exist, varying between 1 to Δ during the T_1 time slots, Lemma 1 always keeps true. 2) If a node (such as a quasi-leader node in the algorithm) does not receive any packet after T_1 time slots, then the number of its neighbors sending packet must have decreased to zero in the T_1 time slots. That is to say, there are no quasi-leaders left around this node.

Lemma 2: Assume that each sensor node in a newly deployed sensor network sends packet with a probability $p = \frac{1}{\Delta}$ in every time slot. Then, every neighbor sensor node of X_i can correctly receive at least one packet from X_i with a probability not lower than $1 - N^{-3}$ after $T_2 = 3\Delta \ln N(1 + \frac{1}{\Delta-1})^{\Delta-1}$ time slots.

Proof: Let P'_{suc} be the probability that any one of the neighbors correctly receives one packet from X_i in a time slot, and P'_{no} denote the probability that the neighbor node does not receive any packet in the subsequential T' time slots. Then, we have

$$\begin{cases} P'_{suc} = p\left(1-p\right)^{D_i-1} \geq p\left(1-p\right)^{\Delta-1} \\ P'_{no} \leq \left[1 - p\left(1-p\right)^{\Delta-1}\right]^{T'} \\ \qquad < \exp\left\{-T'p\left(1-p\right)^{\Delta-1}\right\} = N^{-3} \end{cases} \tag{3}$$

Therefore, when we have

$$T_2 = T' = \frac{3\ln N}{p\left(1-p\right)^{\Delta-1}} = \frac{3\ln N}{p}\left(1-p\right)^{1-\Delta},$$

Lemma 2 can be true. Furthermore, it is easy to see that T_2 gets its minimum value $3\Delta \ln N(1 + \frac{1}{\Delta-1})^{\Delta-1}$ when $p = \frac{1}{\Delta}$. ■

Considering the monotonicity of T_2, we can also draw a conclusion that Lemma 2 will still hold when the number of neighbor nodes sending packets decreases during T_2 time slots.

Since $T_2 \geq T_1$ and $p = 1 - \sqrt[1-\Delta]{\Delta} \approx \frac{1}{\Delta}$, we select $p = \frac{1}{\Delta}$ for the two Lemmas. Therefore, $T_1 = 3\Delta \ln N$ is selected in the proposed algorithm. Furthermore, since we have

$$3\Delta \ln N(1 + \frac{1}{\Delta - 1})^{\Delta-1} \leq 3e\Delta \ln N,$$

which will be simplified, $T_2 = 3e\Delta \ln N$ is selected in the algorithm. With the help of these two Lemmas and the conclusions, we can give the detailed proof of the proposed algorithm as follows.

Validity: Among all leader node's neighbors, there should be no other leader nodes. Among all slave node's neighbors, there is at least one leader node.

Proof: Suppose that there is a new leader node B, which is also covered by the transmission range of another leader node A. Then, before B turns to be a leader node, there will be two possible situations for the sensor node A: 1) If leader node A has stopped sending packets, B must have received a packet from A according to Lemma 2 and it turns to be a slave; 2) If A has just turned to be a leader, then it has already sent packets during $3\Delta \ln N$ time slots (being a quasi-leader) according to the algorithm. Hence, B will turn to be a quasi-slave according to Lemma 1 and can not turn back to a quasi-leader. Therefore, B can not be a leader node. In addition, according to the algorithm, we know the unique reason that a node turns to be a slave is that it has received a packet from a leader. Therefore, there is surely at least one leader within the slave node's transmission range. Hence, the validity of the algorithm has been proved. ■

Integrality: The proposed algorithm is convergent. In other words, it can ensure that each sensor node turns to be a slave or a leader within a limited time.

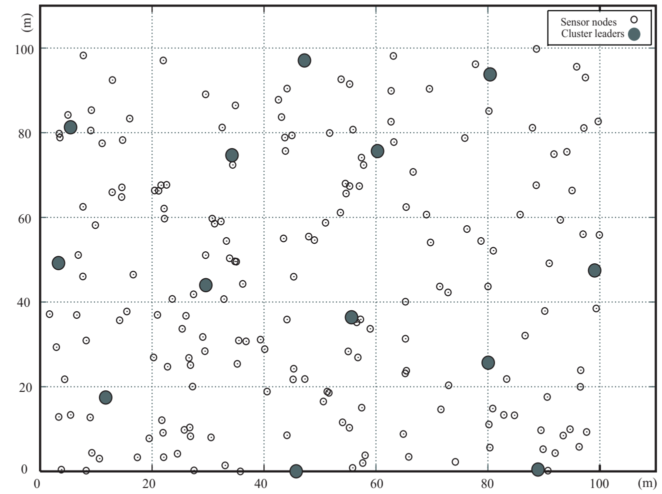

Fig. 2. The final node distribution after using the proposed algorithm.

Proof: According to above analysis, there is no more than one node that can turn to be a leader node in any local 1-hop area after the first round of leader competition. However, in view of the whole network, it should be true that there must be at least one quasi-leader node that may set others to be quasi-slaves nodes previously and none of the others can set it. Hence, it will certainly becomes a leader node. Therefore, for every $3\Delta \ln N$ time slots, there must be at least one quasi-leader turned to be a leader in the whole networks. Hence, the algorithm is convergent. ■

III. SIMULATION RESULTS

To validate the performance of the proposed cluster initialization algorithm, computer simulations have been conducted and their results are given in this section. Moreover, we will also compare the performance of the proposed algorithm with that proposed in [12]. In the computer simulations, 200 sensor nodes have been deployed randomly and uniformly in an area of 100×100 square meters. The transmission range of each sensor node is assumed to be 20 meters. In this paper, the value of ε has been set to 10^{-3}. Therefore, it is easy for us to obtain $\Delta = 60$, $p = 0.0263$, $T_1 = 954$, and $T_2 = 2592$. For comparison convenience, we also have set the other parameters to $\alpha = 752.5$ and $\eta = 0.0078$, which were also used in [12].

Figures 2 shows the computer simulation results for the election of sensor cluster headers in a newly employed WSN according to the proposed algorithm. It can be seen that the cluster leader nodes are very uniformly distributed in the WSN and all the other nodes are in the transmission range of the leader nodes.

Figure 3 illustrates the results of the computational complexity (i.e., the running time in terms of the number of time slots) using the proposed algorithm and that given in [12]. From Figure 3, the proposed algorithm terminates at about 4000 time slots; whereas the algorithm proposed in [12] terminates after more than 9000 time slots. From Figure 3, it also can be found that during the first 900 time slots, the number of leader nodes using the algorithm proposed in [12] increases rapidly; whereas the number of leader nodes in the proposed algorithm is almost zero. During the follow-up time slots, the number of leader nodes in the proposed algorithm increases very fast until achieving its maximum at about 1800 time slots. However, the number of leader nodes in the algorithm presented in [12] reaches to its maximum at almost 5000 time slots.

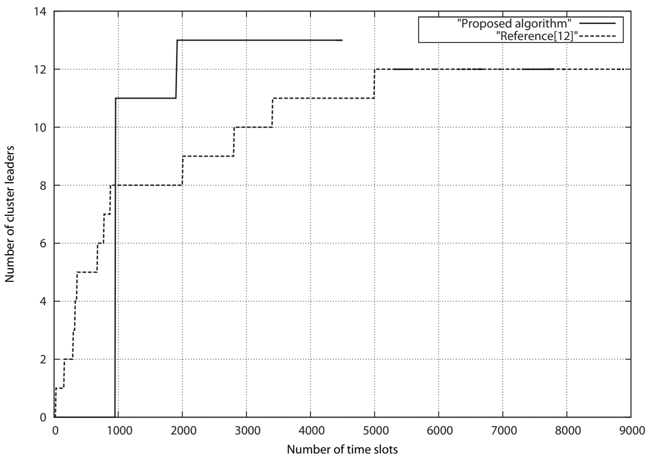

Fig. 3. Comparison of the leader election processes using the proposed algorithm and that presented in [12].

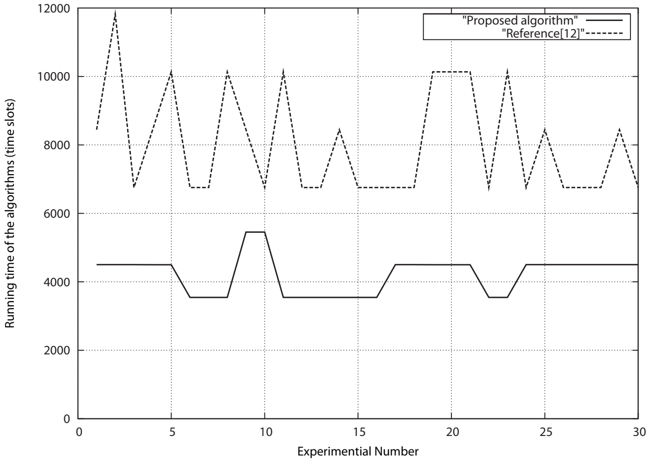

Fig. 4. Comparison of the running times using the proposed algorithm and that presented in [12].

Figure 4 illustrates that the running times of the two algorithms in a set of experiments in the same initial WSN. From Figure 4, it is easy to observe that the proposed algorithm runs steadily and it often ends at about 4000 time slots; whereas the running time of the algorithm given in [12] varies within a wide range from 6700 time slots to 12000 time slots. It also can be seen that, the average of the running time using the algorithm proposed in [12] is about 8000 time slots, which is twice of that for the proposed algorithm. Therefore, the computational complexity of the proposed algorithm is much lower than that of the algorithm given in [12].

Figure 5 shows the comparisons of the running times between the proposed algorithm and that presented in [12] with different scales of WSNs. The numbers of the sensor nodes in the WSN are different, whereas the sensor node deployment densities of the WSNs are kept the same. We run 30 times

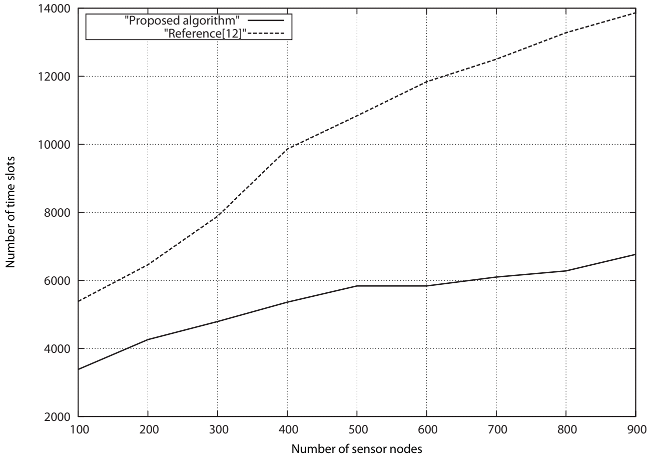

Fig. 5. Comparison of the running times using the two algorithms with different networks scales.

of experiments for each WSN using the two algorithms, and corresponding average results have been shown in Figure 5. From Figure 5, we can see that the running time of the proposed algorithm increases slowly with the increment of network scale, demonstrating the distributive characteristics of the proposed algorithm proposed in this paper. However, the running time of the algorithm suggested in [12] increases very fast with the network scale.

IV. CONCLUSION

In the initial stage of a newly deployed wireless sensor network, most of existing communication protocols may not work properly. In this paper, we have discussed the issue on cluster leader election process in the initialization stage of a multi-hop wireless sensor network, and we proposed a clustering algorithm based on a single communication channel. Compared to the previously reported schemes, our proposed algorithm can offer a much better performance with a relatively low computational complexity. The effectiveness of the proposed algorithm has been demonstrated by both theoretical analysis and computer simulations.

REFERENCES

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishman, “An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, 2002, pp. 660-670.

[2] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Topology management for sensor networks: exploiting latency and density," in *Proc 3rd ACM Int'l Symp. Mobile Ad Hoc Networding Computing*, Lausanne, Switzerland, June 2002, pp. 135-145.

[3] H. Huang and J. Wu, “A probabilistic clustering algorithm in wireless sensor networks," in *Proc. IEEE 62nd Veh. Technol. Conf.*, Texas, USA, vol. 3, Sept. 2005, pp. 1796-1798.

[4] C.-H. Lin and M.-J. Tsai, “A comment on “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks"," *IEEE Trans. Mobile Computing*, vol. 5, no. 10, Oct. 2006, pp. 1471-1472.

[5] C. Ali and P. Samuel, “Power-efficient clustering in wireless sensor networks under coverage constraint," in *Proc. 4th IEEE International Conf. Wireless Mobile Computing, Networking Commun.*, Avignon, France, Oct. 2008, pp. 460-465.

[6] F. Tashtarian, M. T. Honary, M. Mazinani, A. T. Haghighat, and J. Chitizadeh, “A new level based clustering scheme for wireless sensor networks," in *Proc. IEEE/ACS International Conf. Computer Systems Applications*, Doha, Qatar, Apr. 2008, pp. 284-290.

[7] T. Hayashi, K. Nakano, and S. Olariu, “Randomized initialization protocols for packet radio networks," in *Proc. 13th International 10th Symposium Parallel Distributed Processing*, Puerto Rico, Apr. 1999, pp. 544-548.

[8] R. S. Bhuvaneswaran, J. L. Bordim, J. Cui, N. Ishii, and K. Nakano, “An energy-efficient initialization protocol for wireless sensor networks," in *Proc. 30th International Conference on Parallel Processing*, Valencia, Spain, Sep. 2001, pp. 423-428.

[9] Z. Cai, M. Lu, and X. Wang, “Distributed initialization algorithms for single-hop ad hoc networks with minislotted carrier sensing," *IEEE Trans. Parallel Distributed Syst.*, vol. 14, no. 5, pp. 516-528, May 2003.

[10] J. Cichon, M. Kutylowski, and M. Zawada, “Initialization for ad hoc radio networks with carrier sensing and collision detection," in *Proc. 5th International Conf. AD-HOC Networks Wireless*, Ottawa, Canada, Aug. 2006, pp. 308-320.

[11] V. Ravelomanana, “Optimal initialization and gossiping algorithms for random radio networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 18, no. 1, pp. 17-28, Jan. 2007.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhoffer, “Initializing newly deployed ad hoc and sensor networks," in *Proc. 10th Annual International Conf. Mobile Computing Networking (MOBICOM)*, Sep. 2004, pp. 260-274.

[13] L. Jin, L. Fang, and H. Zailu, “Initialization algorithm based on snir model for wireless sensor networks," in *Proc. 2th International Conf. Wireless Commun., Networking Mobile Computing*, China, Sep. 2006, pp. 1-4.

[14] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit disk graphs," *Discrete Mathematics*, 1990, pp. 165-177.

[15] B. Liu and D. Towsley, “A study of the coverage of large-scale sensor networks," in *Proc. 1st IEEE International Conf. Mobile Ad-hoc Sensor Syst.*, Fort Lauderdale, FL, USA, Oct. 2004, pp. 475-483.

[16] S. Iyer and D. Manjunath, “Topological properties of random wireless networks," in *Sadhana: Proc. Indian Academy Sciences*, vol. 31, no. 2, Apr. 2006, pp. 117-139.