This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2017 conference.

Identification of Data Aggregators in Wireless Sensor Network

Ganesh Khade[1] and S. Mini[2]
Department of Computer Science and Engineering, National Institute of Technology Goa, Goa-403401, India
Email: [1]khadevikas41@gmail.com [2]mini2min2002@yahoo.co.in

***Abstract*—Data aggregation is one of the major challenges in wireless sensor networks. Several approaches have been proposed in the past. They can be broadly categorized into three types. The first is tree-based approach, where data aggregation is performed by generating minimum spanning tree. The second is In-Network approach, where data aggregation is done by performing data compression operation on data. The third one is cluster-based approach, where data aggregation is done by dividing whole network into several clusters. The comprehensive study on data aggregation that is carried out reveals potential scope for improvement in the existing data aggregation algorithms. In this work, we construct Connected Dominating Set based on 1-hop Neighbors (CDS-1HN). This is used to determine the virtual backbone of a given network. Data aggregators are identified by using the virtual backbone. The results reveal that considerable amount of energy can be saved by using the proposed method.**

I. Introduction

Wireless Sensor Network (WSN) consists of a collection of sensor nodes that have some computational power and sensing capability. They are used for large number of applications such as vehicle tracking, forest fire detection etc. The routing models in WSN are classified into two types: Address-Centric and Data Centric. In Address-Centric Routing, the source node sends data to the base station along the shortest path and in Data Centric Routing, sensor node sends data to the nearest intermediate node. After receiving data intermediate node performs data aggregation operation on collected data and forwards result to the base station [1]. WSN can be deployed for observing certain area and supervising the system where human interference is unattainable. In a large scale network there are many problems that may arise, such as, many nodes obstruct each other, presence of plenty possible routes and so on. These problems can be resolved by choosing a few sensor nodes as a virtual backbone (VB) for a network, where interior nodes are connected to each other and exterior nodes are connected to at least one node in VB [2].

Sensor nodes are generally battery powered, so energy aware data transfer is needed in the pursuance of increasing the network lifetime [3]. The sensors monitor the environment and sends the sensed data to the sink node where data aggregation operation on collecting information is done for end-user queries [4]. In large-scale WSN, sensor nodes generate large amount of data and transmits to the sink node, which performs data aggregation operation on received data and broadcasts result to a server for different applications such as patients health monitoring, military applications etc. [5], [6].

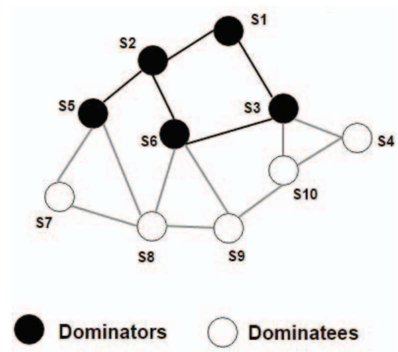

Fig. 1. Virtual backbone in WSN.

A VB is used to connect pieces of network. A Connected Dominating Set (CDS) is considered as VB which is beneficial for minimizing the traffic load on sensor node. For a given WSN, Dominating Set (DS) is defined as a subset of nodes where the node not in subset is neighboring to at least one member of subset. If the nodes in the DS are connected to each other then it is known as CDS [12]. Fig. 1 illustrates the sketch map of VB in WSN. The nodes within VB are known as dominators whereas the nodes outside VB are called as dominatees. The DS in Fig. 1 consists of $\{S_1, S_2, S_3, S_4, S_5, S_6\}$.

Jing He *et al.* [16] defines CDS as a VB which is used for effective routing and broadcasting in WSN. There are various applications of CDS in WSN such as minimizing routing overhead, energy-efficient routing, area coverage etc. [13], [15]. The sensor nodes in WSN communicate with each other over shared medium using one-hop or multi-hop relays. Since there is no physical backbone framework in WSN, a VB is constructed using DS. The identification of CDS in WSN using disk graph is done in [18]. In this paper, we construct Connected Dominating Set based on 1-Hop Neighbors (CDS-1HN) that determine a CDS from a given network. The CDS-1HN is divided into two phases. In the first phase, all sensor nodes are sorted in descending order based on one-hop neighbors degree. The construction of CDS is done in the second phase.

The rest of the paper is organized as follows. Section II describes the related work. Section III states the problem. A detailed illustration of CDS-1HN algorithm along with an example is provided in Section IV. The experimental results are described and analysed in Section V. The conclusion and future work are given in Section VI.

978-1-5090-4442-9/17/$31.00 ©2017 IEEE

II. Related Work

In WSN, it is difficult to transmit data directly to the interested user during event detection due to energy and other resource constraints. Instead, the nearest sensor node performs data aggregation operation on data locally in order to eliminate the redundancy. Data aggregation is the process of integrating data from different sources in pursuance of removing data redundancy, decreasing the number of transmissions. The data aggregation is beneficial for reducing communication cost and to improvise the energy efficiency. A tree-based approach is proposed in [1] by establishing an aggregation tree which spreads all the node as vertices. Tree-based approach is classified into three types. The first is Center at Nearest Source in which the source node which is nearest to the sink node is considered as data aggregator and all other source nodes send data directly to this source node which performs aggregation operation on receiving data and then sends result to the base station. The second is Shortest path tree in which every source node forwards its data directly to the base station along the shortest path between the two. The third is Greedy Incremental Tree. In this approach, aggregation tree is generated sequentially. At the first step, the tree consist of the shortest path between the base station and the node which is adjacent to the base station. In the next step, the node which is nearest to the tree is connected to the tree. The Load Balanced Connected Dominating Set (LBCDS) algorithm for identifying the CDS from a given WSN is proposed in [2]. S. Wan *et al.* [6] propose a tree based structured approach for data aggregation that reduces energy consumption during data transmission. The efficiency of data aggregation is increased by routing the structure of the WSN and it also focuses on how to designate superior routing metric on the basis of data quality to promote data aggregation. The problem of determining an optimal data aggregation tree that increases the network lifetime is NP-complete. Y. Wu. *et al.* [7] propose an approximation algorithm that generates suboptimal tree.

Different ways of data aggregation are described in [8]. One is by aggregating information received by sensor nodes into one packet. In another approach, the sink node performs data aggregation operation by using divisible functions such as sum, max, min, average etc. The last one is data compression where the data reported by sensor nodes is compressed. Two approximation algorithms are proposed that resolve the problem of establishing a data aggregation tree in order to minimize the energy utilization. A graph based model is proposed in [9] for data aggregation at the sink nodes within WSN.

In-Network data aggregation [10] is the comprehensive process of collecting and routing data over the network. The data is processed at intermediary node with the aim of minimizing energy consumption to maximize the network lifetime. In-Network data aggregation is classified into two types: with size reduction and without size reduction. The sink node performs the data compression operation during data aggregation in In-Network data aggregation with size reduction. For example, suppose two different source nodes send same reading which is the locally measured temperature to the sink node. Then the sink node takes average of these readings and sends result to the base station. The sink sends information to the base station without performing any compression operation in In-Network data aggregation without size reduction. If two different nodes send two different readings containing locally measured temperature, the sink node combines both the readings into a single packet without performing any data compression operation and forwards result to the base station. Cluster-based approach is discussed in [11], where the whole network is broken into several clusters. Among all sensor nodes, one node is elected as cluster head and all other nodes are considered as cluster members. Cluster head performs data aggregation operation. The cluster head is elected among cluster members. Y. Xiang *et al.* [14] propose two distributed algorithms for WSN. One is based on a growing tree whereas the second is based on the Maximal Independent Set (MIS). Genetic algorithm based construction of LBCDS for identifying CDS and balancing traffic load on each sensor node in a given network is proposed in [16]. Q. Tang *et al.* [17] propose an algorithm, CSCDS, which is used to identify the DS in WSN. The CSCDS is divided into two phases. In the first phase, the dominators are identified based on one hop white neighbor information. In the second phase, the CDS is constructed on the basis of connected subset. The virtual backbone scheduling (VBS) is used to dynamically turn off the radio of the sensor node in order to minimize energy consumption [18]. They also propose an algorithm which distributes network energy consumption uniformly among all sensor nodes and extends network lifetime.

In this paper we propose CDS-1HN. It is used for determining data aggregators using CDS in a given WSN. This work considers degree of 1-hop neighbors to identify data aggregators.

III. Problem Statement

A. Network Model

A WSN can be modelled as a connected undirected general graph $G = (V, E)$ where the sensor nodes are represented by V and the communication link between the sensor nodes is represented by E. Two nodes can communicate if and only if both are in each others communication range. The connected nodes can share information among each other. It is assumed that all nodes have identical communication range. The nearest dominator is connected to the base station.

B. Preliminary

1) Dominators: The nodes within CDS are considered as Dominators as well as data aggregators. They work as data forwarders as well as receivers. It is marked black.

2) Dominatees: The nodes outside CDS are known as Dominatees. They forward data to their connected dominators. It is marked white.

C. Problem Definition

For a graph $G = (V, E)$ which represents the WSN, the CCDS-1HN problem is defined to discover subset $V' \subseteq V$, where $V' = (S_1, S_2, \dots S_m)$ such that

1) $G' = (V', E')$ where, $E' = \{e/e = (u, v), u \in V', v \in V'\}$ is connected.
2) $\forall(u \in V)$ and $(u \notin V')$, $\exists(v \in V')$, such that $(u, v) \in E$.

IV. Construction of Connected Dominating Set Based on 1-Hop Neighbors (CDS-1HN)

A. Preliminary Terminologies

1) One-hop Neighbor: The nodes u and v are one-hop neighbors to each other iff $d(u, v) \leq r$ where r is the transmission range.

2) Node Degree: For every node i, node degree is the number of one-hop neighbors. It is denoted by (d_i)

3) Mean Degree:

$$\bar{d} = \frac{(\sum_{i=1}^{n} d_i)}{n} \tag{1}$$

where, $\bar{d}$ = mean degree of the graph
n = number of nodes in the graph.

B. Algorithm Description

The CDS-1HN algorithm is divided into two phases. In the first phase, all the sensor nodes are sorted in descending order on the basis of node degree. If there exists a tie among the sensor nodes then the sum of 1-hop neighbors degree of those sensor nodes are used to break the tie between them.

In the second phase, in accordance with sorted nodes, the construction of CDS is done. The nodes whose node degree is 1 cannot be considered as dominator and cannot be shifted into DS. If the sensor node has node degree greater than mean degree of the graph and all of its 1-hop neighbors are not dominated then that node is shifted into DS. In order to balance the load on each dominator in DS, we cannot add the sensor node into DS which has node degree less than mean degree of the graph. For that node, if there exist only one 1-hop neighbor node then that node is considered as dominator and shifted into DS. Otherwise, we need to calculate the sum of 1-hop neighbor degree for each node which has node degree greater than one and the node which has least sum is considered as dominator and shifted into DS. Algorithm 1 and 2 show the pseudo-code for CDS-1HN.

To illustrate the method, we consider the WSN shown in Fig. 2 to build CDS by using CDS-1HN. By using node degree of each sensor node (d), we can determine the mean degree of the graph $\bar{d} = 2.4444$.

In the first phase, for S_4 and S_7, there exists a tie. So, we calculate the sum of degree of 1 hop neighbors.

$$S_4 \Rightarrow d(S_3) + d(S_2) + d(S_7) + d(S_1) = 14$$
$$S_7 \Rightarrow d(S_6) + d(S_2) + d(S_4) + d(S_1) = 13.$$

Therefore we sort S_4 and S_7 as $\{S_4, S_7\}$.

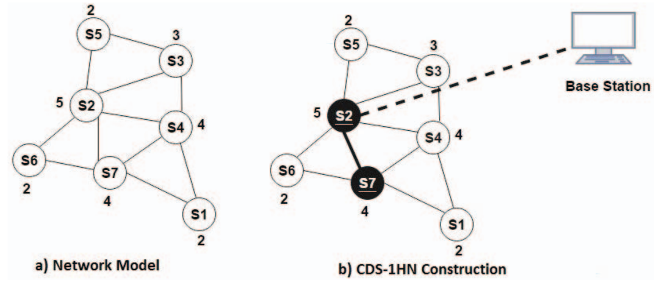

Fig. 2. Construction of CDS-1HN.

Algorithm 1 : CDS-1HN: Sorting Nodes

- **Instance**:
  - A graph $G = (V, E)$ represents WSN.
  - Node degree (d_i).
- **Algorithm**

```
1: Sort the n sensors in decreasing order on the basis (d_i), where 1 ≤ i ≤ n.
2: if there exists a tie among sensor nodes then
3:     goto 6
4: else
5:     goto 12
6:     while true do
7:         Check 1-hop neighbors of every node.
8:         Calculate the sum of all 1-hop neighbors degree of every node.
9:         Arrange the nodes in descending order based on sum.
10:    end while
11: end if
12: Store node IDs in the array denoted by A.
```

For S_1, S_5 and S_6 too there exists a tie. We carry out the same procedure to resolve the tie.

$$S_1 \Rightarrow d(S_4) + d(S_7) = 8$$
$$S_5 \Rightarrow d(S_2) + d(S_3) = 8$$
$$S_6 \Rightarrow d(S_2) + d(S_7) = 9.$$

Therefore we sort S_1, S_5 and S_6 as $\{S_6, S_1, S_5\}$.

Finally, sorted list of nodes is $\{S_2, S_4, S_7, S_3, S_6, S_1, S_5\}$.

In the second phase, S_2 has node degree greater than mean degree. So, S_2 is added to the DS and S_2 dominates $\{S_3, S_4, S_5, S_6, S_7\}$.

In the next round, we can see that S_1 is not dominated. We cannot add S_1 to the DS because S_1 has node degree less than mean degree.

Therefore, we check the 1-hop neighbors of S_1.

$$S_4 \Rightarrow d(S_3) + d(S_2) + d(S_7) + d(S_1) = 14$$
$$S_7 \Rightarrow d(S_6) + d(S_2) + d(S_4) + d(S_1) = 13.$$

Algorithm 2 : CDS-1HN: Construction of CDS

- **Instance**
  - A graph $G = (V, E)$ represents WSN.
  - Node degree (d_i).
  - Mean degree of the graph ($\bar{d}$).
  - An array 'A' which contains sorted node IDs.
- **Algorithm**

```
1:  for i = 1 to n do
2:      if A[i] ≥ d̄ then
3:          if sensor node represented as A_i and all of its 1-hop neighbors are not dominated then
4:              Mark that node black
5:          end if
6:      else
7:          goto 8
8:          while true do
9:              Check 1-hop neighbors of A[i].
10:             if there exists only one 1-hop neighbor then
11:                 Mark that node black
12:             else
13:                 if there exists more than one 1-hop neighbor and degree of each 1-hop neighbor is greater than one then
14:                     goto 17
15:                 else
16:                     goto 21
17:                     while true do
18:                         Calculate sum of degrees of 1-hop neighbors for each node.
19:                         Mark the node as black which has least sum.
20:                     end while
21:                     Calculate sum of degrees of all 1-hop neighbors except the node which has node degree one.
22:                     Mark the node as black which has least sum.
23:                 end if
24:             end if
25:         end while
26:     end if
27:     return black nodes
28: end for
```

We add S_7 to the DS as dominator for S_1 because S_7 has least sum.

Finally, CDS $= \{S_2, S_7\}$, dominators $= \{S_2, S_7\}$ and dominatees $= \{S_1, S_3, S_4, S_5, S_6, S_7\}$. The dominator node S_2 is connected to base station because it is the nearest node.

V. Results and Discussion

We consider a 500 × 500 m region. The number of sensor nodes are varied from 100 to 300 with an interval of 50. The communication range is varied from 50 m to 150 m with an interval of 25. It is also assumed that each sensor node has 100 units of energy. Transmission and reception of a packet consumes 1 unit energy each. Results are reported as an average of 100 instances. The experimental results show that CDS-1HN algorithm provides less CDS size as compared to an algorithm in [14] and LBCDS [2]. It is also found that the energy consumption of CDS-1HN is lesser than Algo 1 [14] and LBCDS. With increase in the number of nodes from 100 to 300, the proposed algorithm saves considerable amount of energy.

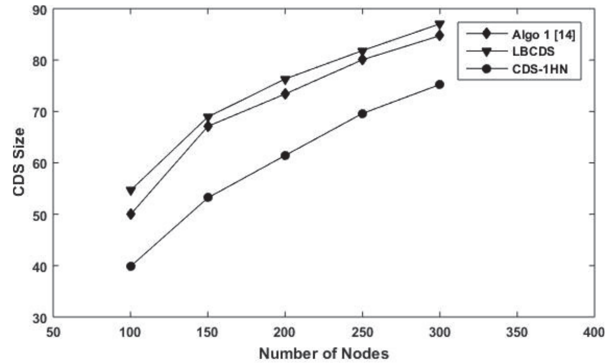

Fig. 3. Number of nodes vs. CDS size.

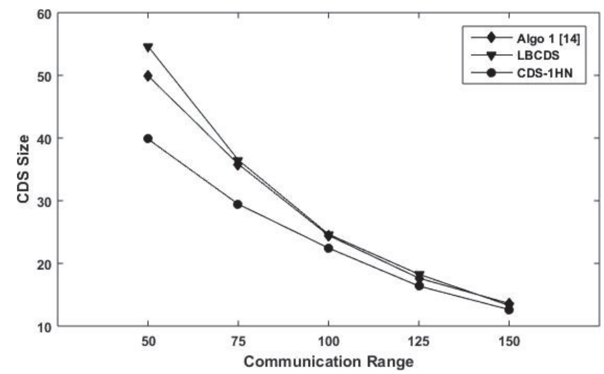

Fig. 4. Communication range vs. CDS size.

Fig. 3 illustrates the comparison of proposed method with Algo 1 [14] and LBCDS for CDS size. As the number of nodes are incremented, the number of dominators increases because when the number of nodes increases, we require extra number of dominators to construct CDS. For 100 nodes, CDS-1HN algorithm gives less CDS size compared to Algo 1 [14] and LBCDS. It is also noticed that with an increment in the number of nodes, the behavior of Algo 1 [14] and LBCDS is almost same and it shows nearly same CDS size. For similar observation CDS-1HN gives better result i.e. less CDS size. It is observed that CDS-1HN identifies lesser number of dominators compared to LBCDS and Algo 1 [14]. This will lead to energy preservation.

Fig. 4 shows the plot of communication range vs. CDS size for all the three algorithms. With increase in communication range, the number of dominators in all algorithms decrease.

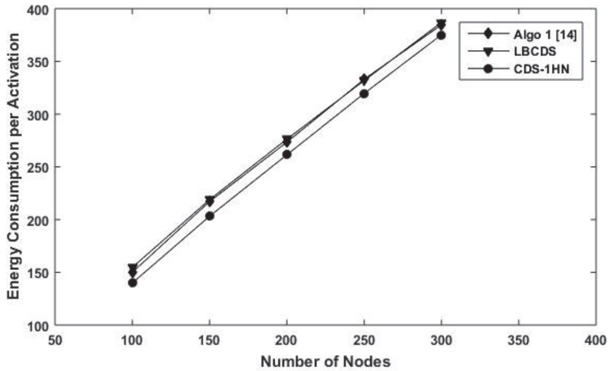

Fig. 5. Number of nodes vs. Energy consumption per activation.

This is because when communication range of dominators increases, it will cover more number of dominatees. When communication range is 50 m, CDS-1HN provides lesser CDS size compared to Algo 1 [14] and LBCDS. As communication range increases from 75 m to 125 m, Algo 1 [14] and LBCDS gives almost same CDS size whereas proposed method gives less CDS size. For communication range 150 m, all the three algorithms give same CDS size. It is also observed that CDS-1HN uses less number of dominators as compared to other algorithms for the experimented communication range. As a result, energy consumption is reduced. The graphical representation of energy consumption of network model using the three algorithms is shown in Fig. 5. As the number of nodes increases, the energy utilization also increases because with increase in number of nodes, the number of dominators also increases and they consume more energy. It is also observed that when the number of nodes are increased from 100 to 300 with communication range 50, the proposed algorithm consumes less energy with respect to other existing algorithms. For 100 nodes, the CDS-1HN saves approximately 10 units energy as compared to Algo 1 [14] and 15 unit compared to LBCDS. With increase in number of nodes from 150 to 300, CDS-1HN saves considerable amount of energy with respect to both algorithms. We notice that the energy utilization in the network by using CDS-1HN is lesser than both the existing algorithms.

VI. Conclusion and Future Work

The energy utilization of a network can be reduced by determining appropriate data aggregators from a given network. In this work, we proposed CDS-1HN, a method to determine CDS in a given network. CDS-1HN was compared with two existing algorithms. Experimental results show that the proposed algorithm saves more energy compared to the existing algorithms by identifying the appropriate dominators. To avoid the network failure, it is needed to balance the load on each data aggregator. An efficient algorithm to balance the traffic load on dominators is proposed to be designed in the future.

References

[1] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker, “Impact of data aggregation in wireless sensor networks,” in *22nd IEEE International Conference on Distributed Computing Systems*, July 2002, pp. 575–578.

[2] Jing (Selena) He, Shouling Ji, Yi Pan, and Yingshu Li, “Greedy construction of load-balanced virtual backbones in wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 14, no. 7, May 2014, pp. 673–688.

[3] Rajesh Kumar Yadav, Daya Gupta, and D. K. Lobiyal, “Energy efficient reactive protocol for data aggregation,” in *Wireless Sensor Network, 3rd International Conference on Computing for Sustainable Global Development*, Mar. 2016.

[4] Jing (Selena) He, Shouling Ji, Yi Pan, and Yingshu Li, “Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, July 2014.

[5] Shaohua Wan, Yudong Zhang, and Jia Chen, “On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks,” *IEEE Sensors Journal*, vol. 16, no. 20, Oct. 2016, pp. 7433–7440.

[6] Cheng Zhao, Wuxiong Zhang, Yang Yang, and Sha Yao, “Treelet-based clustered compressive data aggregation for wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, Sep. 2015, pp. 4257–4267.

[7] Y. Wu, S. Fahmy, and N. B. Shroff, “On the construction of a maximum-lifetime data gathering tree in sensor networks: np-completeness and approximation algorithm,” in *INFOCOM 2008. The 27th Conference on Computer Communications*. IEEE, Apr. 2008.

[8] Tung-Wei Kuo, Kate Ching-Ju Lin, and Ming-Jer Tsai, “On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: np-completeness and approximation algorithms,” *IEEE Transactions on Computers*, vol. 65, no. 10, Oct. 2016, pp. 3109–3121.

[9] S. J. Habib and P. N. Marimuthu, “Data aggregation at the gateways through sensors tasks scheduling in wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 1, no. 3, Sep. 2011, pp. 171–178.

[10] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” *IEEE Wireless Communication*, vol. 14, no. 2, Apr. 2007, pp. 70–87.

[11] K. Dasgupta, K. Kalpakis, and P. Namjoshi, “An efficient clustering-based heuristic for data gathering and aggregation in sensor networks,” *IEEE Wireless Communications and Networking*, vol. 3, Mar. 2003, pp. 1948–1953.

[12] Wu Di, Qu Yan, and Tong Ning, “Connected dominating set based hybrid routing algorithm in ad hoc networks with obstacles,” in *IEEE International Conference on Communications*, June 2006.

[13] E. H. Wassim, A. F. Ala, G. Mohsen, and H. H. Chen, “On efficient network planning and routing in large-scale MANETs,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, Sep. 2009, pp. 3796–3801.

[14] Y. Xiang, K. Xing, W. Cheng, E. K. Park, and S. Rotenstreich, “Distributed virtual backbone construction in sensor networks with asymmetric links,” *Wireless Communications and Mobile Computing*, vol. 11, Aug. 2011, pp. 1051–1060.

[15] Fei Dai and Jie Wu, “On constructing k-connected k-dominating set in wireless ad hoc and sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, July 2006, pp. 947–958.

[16] Jing (Selena) He, Shouling Ji, Mingyuan Yan, Yi Pan, and Yingshu Li, “Genetic-algorithm-based construction of load-balanced cdss in wireless sensor networks,” in *Military Communications Conference*, Nov. 2011.

[17] Qiang Tang, Yuan-Sheng Luo, Ming-Zhong Xie, and Ping Li, “Connected dominating set construction algorithm for wireless networks based on connected subset,” *Journal of Communications*, vol. 11, no. 1, Jan. 2016.

[18] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D. Z. Du, “Connected dominating sets in wireless networks with different transmission ranges,” *IEEE Transactions on Mobile Computing*, vol. 6, July 2007, pp. 721–730.