

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



An Algorithmic Approach to the Node Selection Problem in Industrial Wireless Sensor Networks

Veeramani Sonai and Indira Bharathi

Abstract

Industrial Wireless Sensor Networks (IWSN) are the special class of WSN where it faces many challenges like improving process efficiency and meet the financial requirement of the industry. Most of the IWSNs contains a large number of sensor nodes over the deployment field. Due to lack of predetermined network infrastructure demands, IWSNs to deploy a minimum number of sink nodes and maintain network connectivity with other sensor nodes. Capacitated Sink Node Placement Problem (CSNPP) finds its application in the Industrial wireless sensor network (IWSN), for the appropriate placement of sink nodes. The problem of placing a minimum number of sink nodes in a weighted topology such that each sink node should have a maximum number of sensor nodes within the given capacity is known as Capacitated Sink Node Placement Problem. This chapter proposes a heuristic based approach to solve Capacitated Sink Node Placement Problem.

Keywords: industrial wireless sensor networks, sink node, capacitated sink node, node placement, wighted topology

1. Introduction

Traditionally, the industries are automating their system using wired communications. The wired communication requires quite expensive communication channels and also it needs to be monitored regularly. Due to the high cost incurred in maintaining the wired system, most of the industries not implementing it. Therefore, cost-effective and alternative to wired system needs to be implemented in the industries. Due to recent growth of wireless sensor network (WSN), demands improvement of product development and service provisioning process for the industrial applications. Due to recent enhancement in the wireless sensor network (WSN), the realization of industrial automation can be feasible using WSN.

Figure 1 shows a simple structure of wireless sensor network. In WSN, a small sensor unit is attached to every industrial equipment to monitor the environmental parameters like humidity, temperature, pressure etc., Since all the nodes are not actively participating in the data collection process only some the nodes will be used to gather the data from different sources and these nodes are known as *sink node*. This measured information is transmitted to the sink node and the to the common

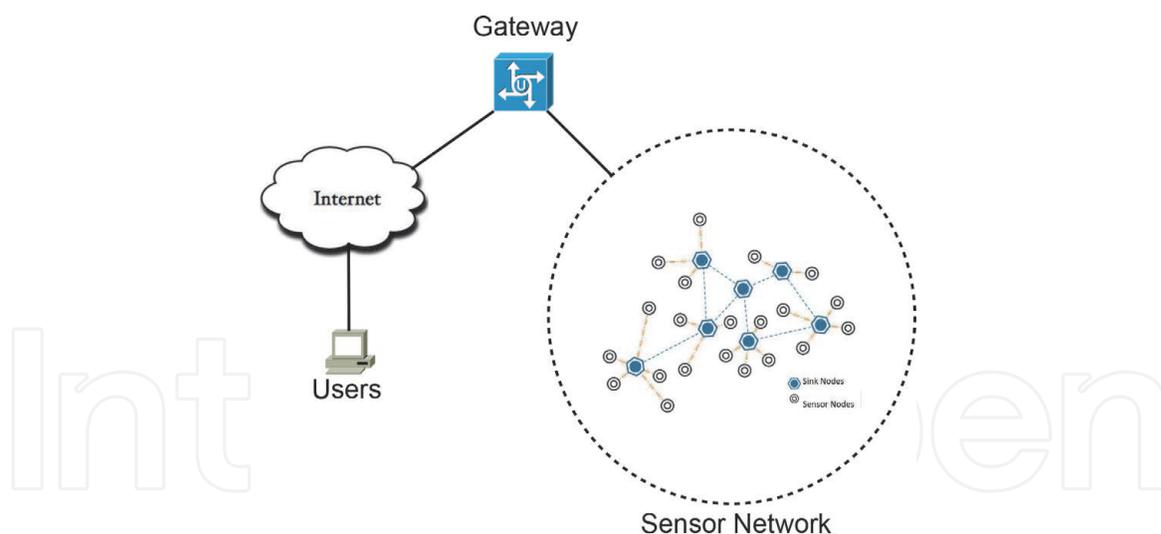


Figure 1.
Simple wireless sensor network.

gateway. Finally, the data will be sent the users through Internet where gateway is connected. In this regard, WSN is widely used to create industrial wireless sensor network (IWSN) that rapidly responses to the real-time events.

Recently, there are many research activities mainly focusing on developing a method to address the challenges like Quality-of-Service (QoS [1, 2]), data redundancy, resource constraints, security, large-scale deployment in the IWSN. In a large-scale network, having single sink node has some disadvantages related to performance and scalability. In such case, the amount of sink nodes in the network has to be increased to overwhelm sink node failure. Hence a better solution for this problem is the usage of multiple sink nodes. Most of the researches are also not focused on the task of multiple sink nodes in the sensor network. So in this chapter, we are concentrated on deploying and handling multiple node in the sensor network.

The sink node deployment problem impacts in flow *set-up time*, *fault tolerance* and *reliability*. These issues include deployment of sink nodes in the specified topology and decide the minimum number of sink nodes required for the same topology. Each node in the topology is connected with weight value known as capacity and each edge is said to be connected with weight value called processing request to the sensor. Deployment of least number of sink node in a specified network topology *will reduce the sink nodes and interaction time between the sensor nodes* become an NP-Hard problem [3]. Thus defining the optimal number of sink nodes is a challenging problem in the Industrial Wireless Sensor Networks (IWSN) because this affects many metrics in the network. In early, sink node placement problems use interaction time between the sensor node and the sink node as metric for each sink node. Those approaches do not consider the situation when the sink node failure happens. So, each sink node has been assigned a maximum (request processing) load apart from considering interaction time. The load applied to process the real-time actions determine the availability as well as the efficiency of the network. Hence the sink node offer the satisfactory resources to implement such actions. The load for the sink node is considered based on the number of actions arrived. Due to insufficient capacity of sink node, to manage resources like processor, bandwidth and memory. In this chapter, the sink node is defined as capacitated which denotes the maximum number of actions are to be processed by the sink node. While choosing the sink node to control different sensor node, the communication time among sensor node and sink node should be reduced in case, if the

nodes are said to be near to each other. If both sensor and sink node are far away, then it takes more time to interact with one another. The core objectives of load consideration for every sink node are as follows:

- Every sensor node has a restriction on memory and processing of each action, only limited number of requests are processed by the sink node.
- As the load of the sink node increases, it fails to handle the request more than its maximum capacity. This will lead to increase of sink node and sensor interaction time.
- If single sink node fails then it may affect another sink node.

The sink node and sensor node communication time plays an essential parameter in the deployment of sink node. A simple approach is, the sensor node is said to be connected with the nearest selected sink node. But this method fails to work properly due to the limitation of load on each sink node. To overcome this issue, the sensor node connecting to the sink node should be stopped as long as its maximum load is reached. The main objective here is to place a minimum number of sink nodes in the network. This work also focuses on reducing the interaction time between sink nodes and sensor nodes. This work proposes an efficient heuristic method to resolve the Capacitated Sink Node Placement Problem.

2. Literature survey

IWSNs bring several advantages over wired industrial automation like monitoring and control [4–6]. In order to utilize the benefits of IWSNs, an effective communication system needs to be maintained. Such a system should address the unique challenges in IWSNs. In this regard, many researchers are working hard to find the solution to the IWSNs challenges, especially establishing and maintaining network connectivity. In a large-scale network, identifying the minimum number of sink nodes and their appropriate placement becomes an NP-Hard problem [3, 7–12].

3. Problem statement

3.1 Prefaces

Assume a simple connected topology T with non-empty nodes of set $V(T)$ and the edge set $E(T) = \{ \{u, v\} \mid u, v \in V(T) \text{ and } u \text{ is adjacent to } v \text{ in } T \text{ and } u \neq v \}$. The neighborhood of a node v of T , $N_T(v)$, is the set of vertices adjacent to v in T . The degree of the vertex v is $d_T(v) = |N_T(v)|$. $\Delta(T)$ denotes the maximum degree of a topology T .

3.2 Problem description

Capacitated Sink Node Placement Problem (CSNPP) is defined as follows:

Given a weighted topology T with weighted nodes $V(T) = \{v_1, \dots, v_n\}$, and the weighted edges $E(T)$. Here, the objective is to find $S \subseteq \{v_1, \dots, v_n\}$ such that $|S|$ is minimum and for each node in S , say x , covers the maximum subset $R \subseteq N_T(x)$ subject to the constraint given in Eq. (1),

$$\sum_{y \in R} \text{weight}(x, y) \leq \text{weight}(x) \quad (1)$$

where, $\text{weight}(x)$ denotes the weight of the node x and $\text{weight}(x, y)$ denotes the weight of the edge between x and y .

3.3 The solution for the capacitated sink node placement problem (CSNPP)

The node placement problem in the network becomes NP-hard [3]. An efficient heuristic method is applied to provide a solution to this problem. The proposed method implements a novel algorithm with the time complexity of $O(n \log n + n\Delta \log \Delta)$, where n is the number of nodes and Δ is the maximum degree of the topology T . This method chooses the node that had maximum node weight value and started assigning the neighborhood nodes with its associated edge weight. Once the sum of edge weight value is larger than the node weight value, then the node is terminated. Furthermore, this method does not consider the maximum degree of a node. The sorted nodes are found first and unmarked the rest of the nodes in topology T . Then, for every unmarked node x in the sorted node list, this algorithm requests one more algorithm 1, which allocates a sensor node for node x . Next process, the node is marked as x and all nodes assigned to it, and finally node x is added to the necessary set of sink nodes. This step is repeated until all the nodes are marked. The performance measures to the Node_Assignment(W, S, N), where W denotes the maximum capacity of the node, S denotes the set of weighted edge value incident on a node, and N represents the set of nodes connected with those nodes. The new node assignment procedure is described in Algorithm 1.

Algorithm 1 Node_Assignment(W, S, N).

Require: Consider a set N consists of m weighted nodes, say n_1, n_2, \dots, n_m and a set S of m weighted edges, say e_1, e_2, \dots, e_m and the maximum capacity W .

Ensure: The set $M \subseteq N$ such that $|M|$ is maximum subject to the constraint

$$\sum_{n_i \in M} e_i \leq W$$

Let $M = \emptyset$.

Sort the edge weights in increasing order, say $e_1 \leq e_2, \dots, \leq e_m$.

for $i = 1$ to m **do**.

if $\sum_{j=1}^i e_j \leq W$ **when**.

$M = M \cup \{n_i\}$

end if.

end for.

return M .

Algorithm 2 Placing_Sink_Nodes.

Require: The weighted topology T with weighted nodes.

Ensure: Subset of $V(T)$ to be placed as sink node.

Let T has n weighted nodes.

Sort the weights of nodes in decreasing order, say $w_1 \geq \dots \geq w_n$. Let v_1, \dots, v_n be the corresponding labels of the nodes.

Let e_{ij} denotes the weight of the edge $\{v_i, v_j\}$.

Unmark all the nodes $v_i, 1 \leq v_i \leq n$.

for $i = 0$ to n **do**.

if v_i is unmarked **then**.

 Let $S = \{e_{ij} | v_j \in N_T(v_i) \text{ and } v_j \text{ is unmarked}\}$. Let N be the set of nodes associated to the v_j 's in S .

 Node_Assignment(w_i, S, N).

 Mark v_i and mark the nodes assigned to v_i .

end if.

end for.

4. Illustration of the proposed capacitated sink node placement problem

Algorithm 2 is traced as follows by using the topology T as shown in **Figure 2**. The node with weight value 21 has the highest degree so it is chosen for locating the initial sink node. This first node then occupy its neighborhood nodes with weighted values 19, 11, 13 since sum of its edge weight value is fewer than or equal to node with value 21 ($15 < 21$) as given in **Figure 3**. Likewise, the next node with higher weight value is 20 and it can occupy it neighborhood nodes 16, 8, 9 since the summation of edge value is smaller ($14 < 20$) as shown in **Figure 4**, which illustrates

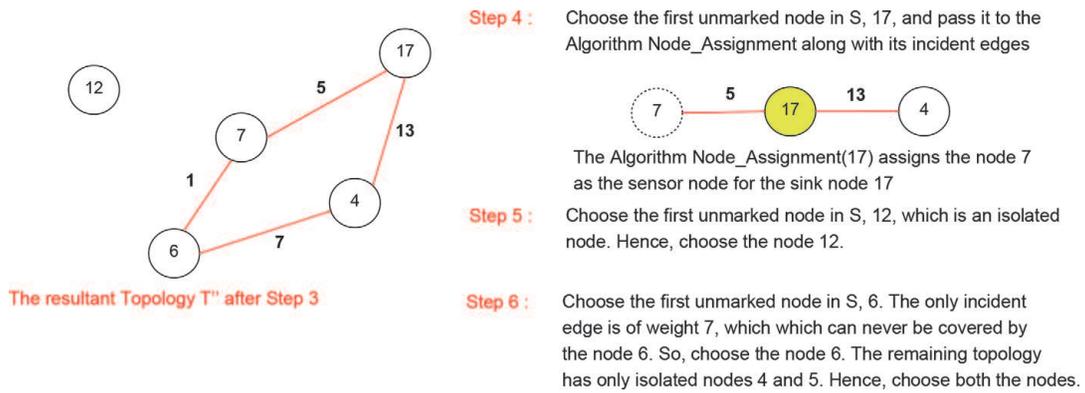


Figure 2.
 The topology T' after placing sink node with node weight 20.

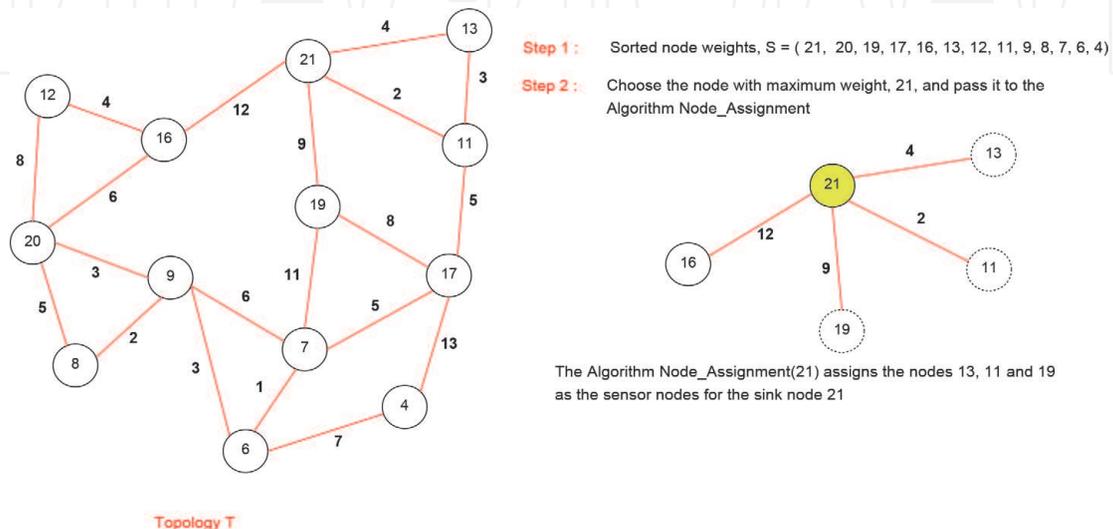


Figure 3.
 Given topology T with weights.

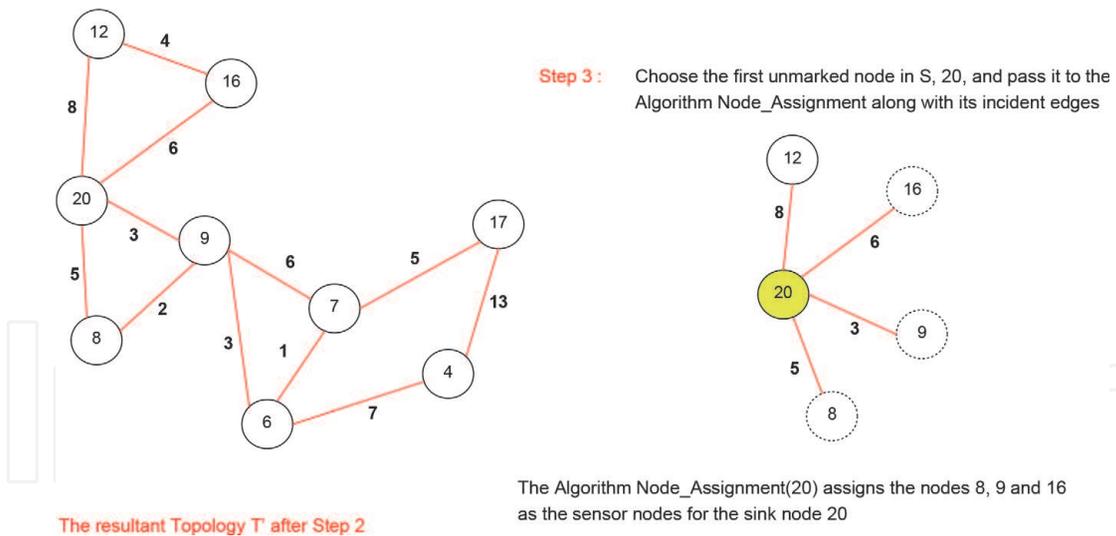


Figure 4.
The topology T' after placing sink node with node weight 21.

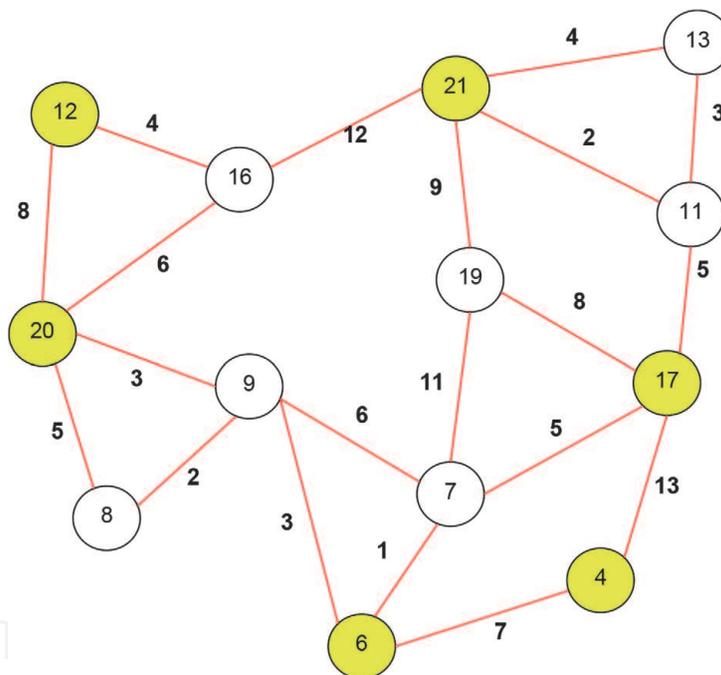


Figure 5.
Final topology after placing all the sink nodes.

the result after 20 is chosen as a sink node and the final topology is shown in **Figure 5**. The node spotted in the final topology represents the position of the sink nodes in topology T .

5. Theoretical analysis of proposed approach

For any given output sink node, the maximum capacity is satisfied according to the constraint (max. capacity) defined and also all the nodes in the given topology are covered completely.

Theorem 1.1 The solution to the Algorithm 2 is feasible.

Proof: The algorithm is proceeds by marking a node as sink node and also label all the nodes covered by the marked sink node. If unmarked nodes are found, then the algorithm is terminated. Henceforth, in a given topology all nodes are covered completely. Furthermore, the proposed solution 1, does not allow the node capacity beyond its maximum defined capacity. Hence, it is proved that the proposed algorithm is feasible.

The proposed approach uses greedy strategy and gives the approximate solution. For a given instance of proposed approach $Node_Assignment()$, it always provides the optimal solution.

Theorem 1.2 The solution to the Algorithm 1 is optimum.

Proof: Let x_1, x_2, \dots, x_s be the set of nodes of the sorted weights w_1, w_2, \dots, w_s in the given topology T and let the maximum capacity be W . Thus, $w_1 \leq w_2 \leq \dots \leq w_s$.

Let $A = (a_1, \dots, a_s)$ be the output of the algorithm and $B = (b_1, \dots, b_s)$ be any feasible solution, $a_i, b_i \in \{0, 1\}$, $1 \leq i \leq s$. Note that, $a_i = 1$ (as well as $b_i = 1$) denotes the presence of w_i in the solution and $a_i = 0$ (as well as $b_i = 0$) denotes the absence of w_i in the solution.

Claim 1: If B is *OPTIMUM*, then $\sum b_i \geq \sum a_i$.

The algorithm outputs a feasible solution since the number of 1's in *OPTIMUM* \geq number of 1's in any feasible solution. Thus the above claim is true.

Observation 1: There exists a $k \in \{1, \dots, n\}$ such that $a_i = 1, 1 \leq i \leq k - 1$ and $a_j = 0, k \leq j \leq n$.

Thus, to show that the algorithm is *OPTIMUM*, it is required to establish $\sum a_i \geq \sum b_i$. Towards this end, it is proved that the following sub claim.

Claim 2: If B is any feasible solution, then $\sum a_i \geq \sum b_i$.

Proof for claim 2 is by mathematical induction on m bits, where m is the number of bits in which A and B differ.

Base Case: $m = 0$. Clearly, $\sum a_i \geq \sum b_i$.

Induction Hypothesis: Consider, the claim is true if they differ in less than m -places, $m \geq 0$.

Induction Step: Let A and B differ in $(m + 1)$ -places, $m \geq 0$.

Observation 2: Since, A and B differ in $(m + 1)$ -places there exists $j, 1 \leq j \leq k - 1$ such that $a_j = 1$ and $b_j = 0$.

Find the least $j \in \{1, \dots, k - 1\}$ such that $a_j = 1$ and $b_j = 0$. Clearly, it can not be the case that $j \in \{k, \dots, n\}$. If so, then B is not a feasible solution.

Now, set $b_j = 1$ (For example, if $A = (1, 1, 1, 1, 0, 0, \dots, 0)$ and

$B = (1, 1, 0, 0, 0, 1, \dots, 0)$, then the modified B is $C = (1, 1, 1, 0, 0, 1, \dots, 0)$). By doing this, it is concluded that either the modified B , say C , is feasible or C is not feasible.

Case 1: C is feasible.

Number of positions in which A and C differs is m . By hypothesis, $\sum a_i \geq \sum c_i$. Note that, $\sum c_i \geq \sum b_i$. Thus, $\sum a_i \geq \sum b_i$. Hence, the sub claim is true if C is feasible.

Case 2: C is not feasible.

Since C is not feasible, there exists a $j \in \{k, \dots, n\}$ such that $a_j = 0$ and $c_j = 1$. Now, set $c_j = 0$. This implies, A and the modified C , say D , differs at $(m - 1)$ -places and by the induction hypothesis, $\sum a_i \geq \sum d_i$. Observe that $\sum d_i = \sum b_i$. Thus, $\sum a_i \geq \sum b_i$.

From the above said claims, it is concluded that $\sum a_i = \sum b_i$. Hence, the Algorithm 1 provides an optimal solution.

Theorem 1.3 The Algorithm 2 runs in $O(n \log n) + O(n\Delta \log(\Delta))$, where n is the number of nodes in the topology T .

Proof. According to the degree of nodes, the nodes are arranged in decreasing order and it takes $O(n \log n)$ as the worst case complexity. The first maximum degree node is passed as input to the Algorithm 1. The same way the algorithm proceeds with the next maximum degree which is not marked and so on which takes the complexity of $O(\Delta \log (\Delta))$. The Algorithm 1 follows by taking local minimum value and adds to the global sum which satisfies the constraints. The algorithm terminates when it breaks the constraints. Thus, the algorithm produce an optimal solution. So the proposed solution takes the complexity of $O(n \log n) + O(n\Delta \log (\Delta))$ for identifying and assigning the sink node.

6. Results and discussion

The simulation is done with test data taken from the Internet Topology Zoo Archive [13]. There are 261 topologies are available in the data set which indicates a large diversity of network structure. These data sets are gathered from different vendors of networking services. We have made a self written python script to convert Graphical Mark-up Language into adjacency list. The data set contains edge and node informations for the given topology. For a given network, capacity is considered as node value and load is considered as edge weight values. Based on the proposed method, the sink is selected in such a way that the summation of edge weight loads fewer than the total capacity of sink node. The simulation of proposed approach gives us total number of sink nodes required for given topology as shown in the **Table 1**. It is observed that the average number of sink nodes required to manage the given topology is 5 – 6.

Name of the Dataset	Number sink nodes
Abilene	4
BelNet	5
UniNett	6
AtmNet	7
Sprint	3
Bell Canada	4
Garr	3
ArpaNet1990	3
Airtel	7

Table 1.
Number of sink nodes vs topology.

7. Conclusion and future work

This chapter proposes an efficient algorithm to place the sink node for the given topological network. This algorithm reduces cost by reducing the number of the sink node being used in the network. Moreover, the placed sink node can serve a maximum number of sensor nodes. The future work is to enhance this approach when a sink node fails in the given topology.

IntechOpen

IntechOpen

Author details

Veeramani Sonai* and Indira Bharathi
Department of Computer Science and Engineering, Vignan's Foundation for
Science, Technology and Research, Guntur, Andhra Pradesh, India

*Address all correspondence to: veeramanicse@gmail.com

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Gungor VC, Hancke GP, Industrial wireless sensor networks: Challenges, design principles, and technical approaches, *IEEE Transactions on industrial electronics*, Issue 56, Vol. 10, pp. 4258–65, 2009.
- [2] Howitt I, Manges WW, Kuruganti PT, Allgood G, Gutierrez JA, Conrad JM, Wireless industrial sensor networks: Framework for QoS assessment and QoS management, *ISA transactions*, Issue 45, Vol. 3, pp. 347–59, 2006.
- [3] Garey MR, Johnson DS, *A Guide to the Theory of NP-Completeness*, WH Freeman, New York, 1979.
- [4] Korber HJ, Wattar H, Scholl G, Modular wireless real-time sensor/ actuator network for factory automation applications, *IEEE Transactions on Industrial Informatics*, Issue 3, Vol. 2, pp. 111–9, 2007.
- [5] Low KS, Win WN, Er MJ, Wireless sensor networks for industrial environments, In *IEEE Computational Intelligence for Modeling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vol. 2, pp. 271–276, 2005.
- [6] Shen X, Wang Z, Sun Y, Wireless sensor networks for industrial applications, In *IEEE Intelligent Control and Automation*, Vol. 4, pp. 3636–3640, 2004.
- [7] Khuller S, Sussmann YJ, The capacitated k-center problem, *SIAM Journal on Discrete Mathematics*, Vol.13, Issue 3, pp.403–18, 2000.
- [8] Veeramani S, Noor Mahammad Sk, A Heuristic Approach for the CCLP Problem in Software Defined Network (SDN), *Internetworking Indonesia Journal*, Vol. 10 No. 1, 2018.
- [9] Hochbaum DS, Shmoys DB, A best possible heuristic for the k-center problem, *Mathematics of operations research*, Vol. 10, Issue 2, pp.180–4, 1985.
- [10] Cygan M, Hajiaghayi M, Khuller S, LP rounding for k-centers with non-uniform hard capacities, In *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 273–282, 2012.
- [11] Plesnik J, A heuristic for the p-center problems in graphs, *Discrete Applied Mathematics*, Vol.17, Issue 3, pp.263–8, 1987.
- [12] Hancke GP, Allen B, Ultrawideband as an industrial wireless solution, *IEEE Pervasive Computing*, Issue 5, Vol. 4, pp. 78–85, 2006.
- [13] <http://www.topology-zoo.org/dataset.html>