

A Survey on Topology Control and Maintenance in Wireless Sensor Networks

Manish Singh

Department of Electronics and Communication Engineering
Invertis University, Bareilly, (UP) INDIA
e-mail: manishsingh.msingh@gmail.com

Abstract

Wireless Sensor Networks (WSNs) consist of devices equipped with radio transceivers that cooperate to form and maintain a fully connected network of sensor nodes. WSNs do not have a fixed infrastructure and do not use centralized methods for organization. This flexibility enables them to be used whenever a fixed infrastructure is unfeasible or inconvenient, hence making them attractive for numerous applications ranging from military, civil, industrial or health. Because of their unique structure, and limited energy storage, computational and memory resources, many of the existing protocols and algorithms designed for wired or wireless ad hoc networks cannot be directly used in WSNs. Beside this, they offer a flexible low cost solution to the problem of event monitoring, especially in places with limited accessibility or that represent danger to humans. Applications of large scale WSNs are becoming a reality example are being a Smart Grid, Machine to Machine communication networks and smart environment. It is expected that a topology control techniques will play an important role in managing the complexity of such highly complicated and distributed systems through self-organization capabilities. WSNs are made of resource constrained wireless devices, which require energy efficient mechanisms, algorithm/protocol. Control on topology is very important for efficient utilization of networks and is composed of two mechanisms, Topology Construction (TC) and Topology Maintenance (TM). By using these mechanism various protocols/algorithm have come into existence, like: A3, A3-Coverage (A3-Cov), Simple Tree, Just Tree, etc. This paper provides a full view of the studies of above mentioned algorithms and also provides an analysis of their merits and demerits.

Keywords: WSNs, A3, A3-Cov, Simple Tree, Just Tree

1. Introduction

WSNs have become an emerging technology that has a wide range of potential applications including environment monitoring, object tracking, scientific observing and forecasting, traffic control, etc [1] [2]. It normally consists of a large number of distributed nodes that organize themselves into a multi-hop wireless network and typically these nodes co-ordinate to perform a common task [3].

For efficient use of WSNs i.e. it functions properly its topology should be control and maintain time to time. For the first method, TC mainly focuses on constructing a connected topology while minimizing energy consumption of nodes to extend the network lifetime. Second i.e. TM it maintains the existing topology, when the existing can't provide the requested service any longer.

Topology construction is important to control the required topology. It is an important technique used in WSNs to achieve energy conservation and extend network lifetime without affecting important network performance such as connectivity and throughput [6]. Topology provides information about a set of nodes and connectivity (links) between a pair of nodes in the set. To construct a network topology, each sensor node discovers its neighbors and relative links using its maximum transmission power. Based on the information gathered the node can make decisions to build a network.

The rest of this paper is organized as follows: Section 2, introduces the related work. Section 3, compares merits and de-merits between A3 and A3-Cov, Simple Tree and Just Tree. Section 4, concludes this paper and also highlights future scope of the work.

2. Related Work

Apart from constructing a topology by reducing the transmission range, other techniques is also present based on the assumption that nodes have information about their own positions and the position of their neighbors or they have directional antennas that are used to determine the orientation of the nodes.

Other method is based on the Connected Dominating Set (CDS) [1] paradigm. The idea is not to change the transmission range of the nodes but to turn unnecessary nodes off while preserving important network properties, such as connectivity and communication coverage. Based on above mentioned topology construction methods some algorithms are discussed in this section.

2.1. A3

The A3 algorithm [4] [5] produces an approximate solution to the minimal CDS [1] problem. The A3 algorithm assumes no prior knowledge about the position or orientation of the nodes; therefore, the nodes do not have an exact geometric view of the topology. However, nodes can determine how far a node is based on the strength of the signal received, and this information is enough to select a close-to-optimal CDS tree, based on the belief that farther nodes will offer better area of communication coverage. The A3 algorithm is executed in 2 moments: Neighborhood discovery, children selection.

All nodes start with the unvisited state, except the stating node, which starts with the "Active" candidate state. An active candidate node sends a "Hello Message" to all its neighbours. The first one that sends this message is the sink node. In addition, this node sets a timer to wait for replies from unvisited neighbour nodes. All the neighbors send back a "Parent Recognition" message that includes their ID and their own selection metric, which is a convex combination of the ratio of remaining energy in the receiver, and the ratio of distance over the maximum transmission range. Also they adopt the sender as their "Parent nodes" and change their state to child.

After a period of time, the active candidate node stops listening for messages, sorts the list of "children" nodes (neighbors who answered) in a decreasing order, and sends this sorted list back to its children. If the active candidate node has received at least one answer, it will change its state to active; otherwise, it will change its state to "sleeping" and will turn off its components until the next topology maintenance routine is executed.

The children nodes find themselves in the list and wait for a period of time proportional to their position on the list. When the timer in a node expires, and it has not received any "Sleeping" messages, the node will send a "Sleeping" message, change its state to active candidate and if the node receives a "Sleeping" message while in the timer set, it will change its state to "Sleeping Candidate", and will turn off its component for a period of time. After this timer expires, the node will change its state to active candidate.

2.2. A3-Cov

A3-Cov algorithm works [4] very similar to A3 protocol, but presents important changes in some portion: like, if there are any nodes that have not received any "Parent Recognition" message, it means that there are no nodes that depend on it for communication purposes; however, they may still be useful in order to extend the network's sensing coverage. In order to do this, A3-Cov defines a new variable in the nodes called "sensing covered" i.e. node "x" is sensing covered by node "y" if "x" is inside the sensing range of "y" and "y" is an active node.

In A3 algorithm after the timer expire in node to receive "Sleep" message. If the node has been "Sensing Covered" by any other node (including its parent node), it sets a short timer to wait for "Sensing Covered" message from its active neighbor.

If the timer expires and the node is not "Sensing Covered" yet, it will turn itself on, changes its state to active and send a "Sensing Covered" message and a "Sleeping" message. If any node in its range receives the "Sensing Covered" message, it will evaluate if it has been covered by sender, in which case it will update the value of the "Sensing Covered" variable.

If the node received a "Sensing Covered" message from any other node, it will stop the timer changes its state to "Sleeping" and turn its component off until the next topology maintenance routine.

A3Cov expands considerably the coverage area as compare to A3.

2.3. Just Tree

For the homogeneous network number of nodes, the deployment area, sink node and the Virtual Network Interface (VNI) [7], play an important role. The just tree algorithm assumes one sink node responsible for message/ information broadcast. The sink nodes are capable of sending or receiving messages from other neighboring sensor nodes. If CDS rule-k is applied and the topology is constructed, this CDS rule-k is needed to run for a quite number of times, a lot of energy will be spent to maintain a particular topology if a sink node prefers to broadcast or send. The concept of spanning tree is considered in most of the cases. This concept can be employed for number of nodes starting from 50-1000 nodes or even more, but in order to achieve acceptable results the node number is increased in multiple of 100. As far as just tree protocol/algorithm is considered, less energy is spent with a reduced queue size, if number of nodes is increased. The message or number of events are propagated within the network using the same concept of parent node and child node, the parent node initiates the message and transfer this message to other sensing nodes acting as child node. Keeping itself in a dominant position i.e. if CDS rule-k is taken into consideration the parent node has maximum in build energy which gradually reduces as the number of nodes increases and message transfer takes place i.e. if the size of the tree is large (more number of nodes) the total energy spent will ultimately decreases till the last child node is covered in a general prospective, if the tree is giant it means that it will cover a larger deployment area and will have large number of children nodes connected with the parent node, which will be the main source of energy with some threshold value i.e. such topology will require atleast.

- (i) Homogenous network
- (ii) Flexible deployment area
- (iii) Parent node that initiates a "HELLO" message with same reasonable amount of threshold in terms of energy in order to support varying Queue size if number of children nodes are also varied.
- (iv) The recognition of the initiated "HELLO" message must be acknowledge by children nodes in order to estimate queue size, energy consumption, number of messages transfer during simulation, performed for different number of nodes for different time periods. The concept of just tree ensures that as the deployment area will increases or if the deployment area is constant the number of nodes if increased will denote the increase in the size of the tree in order to efficiently cover a flexible or constant deployment area.

2.4 Simple Tree

Simple Tree is a derivative form of one or more derivative of spanning tree derived from the just tree algorithm which considers only one CDS per one just tree. According to this, if this algorithm is further splitted into more than one CDS rule-k [1] the load on single parent node can be slightly reduced, however, the total energy spent may substantially increased and it may also affect the queue size. But such algorithm will also require more simulation time because there exist more number of subsets in the same deployment area for the increased node density i.e. such algorithm are not that much simple as far as their name highlights, but are complex requires a greater degree of simulation efforts, are hard to model, are dependent on large queue size and lastly at the out-set shows high value of energy i.e. spent energy. The only advantage of these type of algorithm lies in the aspect that they ensure complete message distribution within their individual CDS and further, if area of the topology is very sparse it can easily recovered by introducing a new CDS in the form of a simple tree rather than raising or disturbing the pre-existing just tree that have their individual CDS. Simple tree algorithm can also be modified to operate for heterogeneous network, if the CDS functions as a closed loop and even if the topology is homogenous and if the CDS functions as a close loop the number of

event or messages floated within the complete network will be less as compared to the just tree algorithm.

3. Comparative Analysis

Based on above discussed algorithm, by analyzing them, some important observations can be make-out between them. The important observations in support of the above discussed algorithm are listed in Table 1 and Table 2 respectively.

Table 1. Comparison between A3 and A3-Cov

Algorithm	Merits	Demerits
A3	<ol style="list-style-type: none"> 1. A3 does not need location information: no GPS or any Location mechanism is necessary. 2. A3 is simple and present Low computational complexity. 3. A3 requires no synchronization scheme. 4. No effect on topology, as if number of nodes increases its response time is almost same. 	<ol style="list-style-type: none"> 1. Number of message or data transfer rate is low in A3 protocol. 2. Coverage area in A3 is Less compared to A3-Cov protocol.
A3-Cov	<ol style="list-style-type: none"> 1. Most of the merits are same as like A3 protocol except: 2. Coverage area is more than A3protocol. 3. Data transfer rate is more than A3 protocol. 	<ol style="list-style-type: none"> 1. As number of nodes increases in A3-Cov response time varies acc. to nodes. 2. Energy consumption is more compared to A3 protocol

Table 2. Comparison between Simple Tree and Just Tree

Algorithm	Merits	Demerits
Simple Tree	<ol style="list-style-type: none"> 1. Applicable for heterogeneous as well as homogeneous close loop networks. 2. Complete distribution of message takes place. 3. Covers larger deployment area. 4. Energy spent on individual node increases. 	<ol style="list-style-type: none"> 1. Difficult to simulate. 2. Network complexity increases (as more than one just tree exist). 3. Response time is high.
Just Tree	<ol style="list-style-type: none"> 1. Reduced queue size compared to simple tree. 2. Energy spent is less as compared to simple tree. 3. Response time is low as compared to simple tree. 	<ol style="list-style-type: none"> 1. Covers less area as compare to simple tree. 2. Message broadcast depends entirely on the single parent node. 3. Suits only homogeneous networks.

4. Conclusion

The affect of existing algorithm for topology maintenance and topology construction have been discussed in this paper. This paper addresses: A3, A3-Cov, Simple Tree, Just Tree and provide an initial review of their performance in TM and TC for sparsely deployed WSNs. The work presented in this paper is a basic attempt towards analyzing and highlighting the performance metrics of the discussed algorithm.

In future these individual algorithms can be simulated to have a deeper insight for further optimizing their utility in varied type of WSNs.

References

- [1] A Karthikeyan, T Shankar, V Srividhya, Siva Charan Reddy V, Sandeep Kommineni. "Topology Control Algorithm for Better Sensing Coverage with Connectivity in Wireless Sensor Networks". *Journal of Theoretical and Applied Information Technology*. 2013; 52(3).
- [2] Degan Zhang, Guang Li, Ke Zheng, Xuechao Ming and Zhao-Hua Pan. "An Energy-Balanced Routing Method Based on Forward-Aware Factor for Wireless Sensor Networks". *IEEE Transactions on Industrial Informatics*. 2014; 10(1).
- [3] Mo Li and Baijian Yang. "A Survey on Topology issues in Wireless Sensor Network".

- [4] Pedro Mario Wightman R, Miguel A. Labrador. "Reducing the communication range or turning nodes off? An initial evaluation of topology control strategies for wireless sensor networks". *Ingeniería & Desarrollo*. Universidad del Norte. 2010; 28: 66-88.
- [5] Pedro M. Wightman and Miguel A. Labrador. "A3: A Topology Construction Algorithm for Wireless Sensor Networks". *IEEE*. 2008.
- [6] Azrina Abd Aziz, Y Ahmet Şekercioğlu, Paul Fitzpatrick, and Milosh Ivanovich. "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks". *IEEE Communications Surveys & Tutorials*. 2013;15(1).
- [7] Pedro M Wightman, Pedro M Wightman. "Atarraya: A Simulation Tool to Teach and Research Topology Control Algorithms for Wireless Sensor Networks". *ACM* ISBN 978-963-9799-45-5, Simulation tool. 2009