A3: A Topology Construction Algorithm for Wireless Sensor Networks

Pedro M. Wightman[1] and Miguel A. Labrador
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida 33620
{pedrow,labrador}@cse.usf.edu

Abstract— **Topology control is a well-known strategy to save energy and extend the lifetime of wireless sensor networks. This paper introduces the A3 (a tree) algorithm, a simple, distributed, and energy-efficient topology construction mechanism that finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and providing complete communication coverage. A3 utilizes a weighted distance-energy-based metric that permits the network operator to trade off the lengths of the branches (distance) for the robustness and durability of the tree (energy). Comparisons with other well-known topology construction mechanisms show the superiority of the proposed scheme in terms of the number of active nodes and energy efficiency. Simulation experiments show that to achieve complete communication coverage, A3 needs only 6% and 41% of the nodes active in dense and sparse scenarios, versus 8% and 43% and 5% and 43% of the EECDS and CDS-Rule-K algorithms, respectively. More importantly, the proposed protocol presents a low linearly bounded worst-case amount of messages per node that limits the overhead and the energy usage compared to a non-linear increase of the EECDS and CDS-Rule-K algorithms.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) continue to be very popular given the large number of application domains where they can provide useful information about. Nevertheless, WSNs also continue to be very limited in terms of computational, communication, storage, and energy resources and capabilities. Of all these constraints it is well-known that energy conservation is the most important aspect and that communications is the most expensive function in terms of energy consumption [2]. Therefore, the design of energy-efficient protocols and algorithms continues to be of utmost importance.

One known strategy to save energy in WSNs is that of Topology Control (TC). TC consists of two separate components: the *topology construction* mechanism, which finds a reduced topology while preserving important network properties, such as network connectivity and coverage, and the *topology maintenance* mechanism, which changes the reduced topology when it can't provide the requested service any longer. It is expected that these two mechanisms will work in a iterative manner until the network energy is depleted, and together will increase the network lifetime compared with a continuously run WSN without topology control.

[1]Professor on leave from Universidad del Norte, Barranquilla, Colombia. www.uninorte.edu.co

This paper introduces A3 (a tree), a topology construction protocol that addresses the problem of finding a reduced topology. A3 finds a sub-optimal Connected Dominating Set (CDS) to turn unnecessary nodes off while keeping the network connected and covered. The algorithm is based on a growing-tree technique and uses a selection metric based on the remaining energy on the nodes and distance among them. The selection metric allows the network operator to choose between a more reliable short-live network with a larger number of nodes, and a less reliable longer-lasting tree with fewer nodes. In addition, the A3 algorithm presents the following advantages: a) A3 is very scalable, as it only needs local information and operates in a completely distributed manner; b) A3 does not need location information; no GPS or any localization mechanism is necessary; c) A3 requires no synchronization scheme thanks to the ordered sequence of the tree creation; d) A3 is simple, and presents low computational complexity; and e) A3 is very energy-efficient. First, the CDS tree building process is done in only one phase and the node selection process avoids the use of node contest or two-hop information queries, which reduces the amount of overhead. Second, the small number of active nodes after the topology construction reduces the number of collisions considerably. Finally, A3 has a low and linearly bounded message complexity, which allows for the algorithm to be run many times as part of the topology control iterative cycle with very low energy cost.

The A3 algorithm is evaluated using simulations and compared with the Energy Efficient CDS (EECDS) [14] and the CDS-Rule-K [11] algorithms. The results demonstrate that in terms of the number of active nodes needed to build the reduced topology, the A3 protocol performs better than the EECDS and CDS-Rule-K mechanisms. A3 only needs 6% and 41% of the nodes active in dense and sparse scenarios while preserving network connectivity and communication coverage, versus 8% and 45% and 9% and 49% of the EECDS and CDS-Rule-K algorithms, respectively. More impressive is the significant difference in the amount of messages required for the creation of the CDS tree, where A3 presents a bounded, low, and linear message complexity compared with an non-linear increase of the other two mechanisms. This low message complexity is essential to realize energy savings in topology controlled networks compared with WSNs that run continuously without topology control.

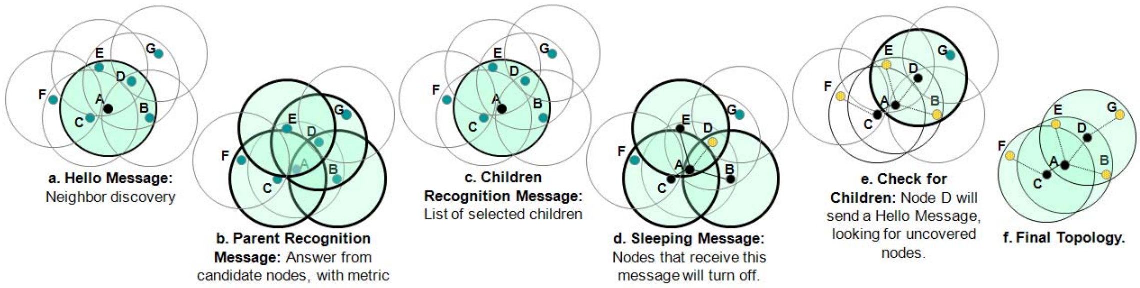


Fig. 1. The A3 Algorithm.

The rest of the paper is organized as follows. Section II includes the related work. In Section III the A3 algorithm is described in detail. Section IV presents the performance evaluation. Finally, Section V concludes the paper.

II. Related Work

Topology construction can be exercised by reducing the transmission range of all nodes by the same minimum amount, or the minimum transmission range for each node [8]. Other techniques are based on the assumption that nodes have information about their own positions and the position of their neighbors [5], or that they have directional antennas that are used to determine the orientation of the nodes [4], [6]. Although both assumptions are valid, they are costly and not easy to implement. Other topology control methods, such as the one considered in this paper, are based on the Connected Dominating Set (CDS) paradigm. Here, 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.

The CDS approach has been utilized in several papers [1]–[4], [11]–[14]. Most CDS-based mechanisms work in two phases: In phase one they create a preliminary version of the CDS, and in phase two they add or remove nodes from it to obtain a better approximation to the optimal CDS. Two relevant CDS-based mechanisms are the Energy Efficient CDS (EECDS) [14] and the CDS-Rule-K [11] algorithms.

The EECDS algorithm builds a CDS tree creating Maximal Independent Sets (MIS), which are clusters with non-connected clusterheads, and then selects gateway nodes to connect the clusterheads of the independent sets. The EECDS algorithm proceeds in two phases. The first phase begins with an initiator node that elects itself as a clusterhead and announces it to its neighborhood. This set of nodes is now "covered". The now "covered" nodes will pass the message to its uncovered neighbors, 2-hop away from the initiator, which start competing to become clusterheads. Once there is a new clusterhead, the process repeats with the 4-hop away nodes from the initiator, until there are no more uncovered nodes. On the second phase the covered non-clusterhead nodes compete to become gateways between the clusterheads.

The CDS-Rule-K algorithm utilizes the marking algorithm proposed in [13] and the pruning rule included in [12]. The idea is to start from a big set of nodes that accomplishes a minimum criterion and prune it according to a specific rule. In the first phase, the nodes will exchange their neighbor lists. A node will remain active if there is at least one pair of unconnected neighbors. In the second phase, a node decides to unmark itself if it determines that all its neighbors are covered by marked nodes with higher priority, which is given by the level of the node in the tree: lower level, higher priority. The final tree is a pruned version of the initial one with all redundant nodes with higher or equal priority removed.

III. The A3 Algorithm

The A3 algorithm produces an approximate solution to the Minimal Connected Dominating Set problem, which is proved to be NP-Hard in [9]. 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 and Second Opportunity.

*1) **Neighborhood Discovery**:* The CDS building process is started by a predefined node that might be the sink, right after the nodes are deployed. The sink, node A in Figure 1a, starts the protocol by sending an initial *Hello Message*. This message will allow the neighbors of A to know its "'parent"' node. In Figure 1a, nodes B, C, D, and E will receive the message. Node F is out of reach from node A. If the node that receives the message has not been covered by another node, it sets its state as covered, adopts the sender as its "parent node", and answers back with a *Parent Recognition Message*, as shown in Figure 1b. This message also includes a selection metric (explained later) that is calculated based on the signal strength of the received *Hello Message* and the remaining energy in the node. The metric will be used later by the parent node to sort the candidates. If the receiver has been already covered by another node, it ignores the *Hello Message*. The A3 algorithm uses four types of messages: *Hello Message*, *Parent Recognition Message*, *Children Recognition Message*, and *Sleeping Message*. If a parent node does not receive any

Parent Recognition Messages from its neighbors, it also turns off, such as the case of nodes E and B in the final topology, as shown in Figure 1f, given that they have no children.

*2) **Children Selection**:* The parent node sets a timeout for a certain amount of time to receive the answers from its neighbors. Each metric is stored in a list of candidates. Once this timeout expires, the parent node sorts the list in decreasing order according to the selection metric. The parent node then broadcasts a *Children Recognition Message* that includes the complete sorted list to all its candidates. In Figure 1c, node A sends the sorted list to nodes B, C, D, and E. Once the candidate nodes receive the list, they set a timeout period proportional to their position on the candidate list. During that timeout nodes wait for *Sleeping Messages* from their brothers. If a node receives a *Sleeping Message* during the timeout period, it turns itself off, meaning that one of its brothers is better qualified to become part of the tree. Based on this scheme, the best node according to the metric will send a *Sleeping Message* first, blocking any other node in its range. Therefore, only the other candidate nodes outside its area of coverage have the opportunity to start their own generation process. For example, in Figure 1d, node D received a *Sleeping Message* from E before its timer expired, so it turned off. Otherwise, it sends a *Sleeping Message* to turns its brothers off. At that time, this particular node becomes a new "parent node" and starts its own process of looking for candidates.

*3) **Second Opportunity**:* Although this methodology works very well, there are some cases in which a node sent to sleep is a bottleneck access to a section of the network. In order to avoid this situation, every node sets a timer once it receives the Sleeping Message to send a *Hello Message* and starts its own building process. As shown in Figure 1e, node D will wake up, send a *Hello Message* and will find node G uncovered, so node D will become active. This operation increases the overhead of the algorithm, but guarantees total coverage of the nodes in the graph, as proved in Lemma 1.

LEMMA 1. IF THE INITIAL GRAPH IS CONNECTED, ALL NODES WILL FINISH IN ANY OF THE FINAL STATES: ACTIVE OR SLEEPING.

Proof: Let us assume that a connected original graph is given. If at the final stage of the algorithm there exists at least one node that is not covered, it is because it did not receive any *Hello Message*. Given that every covered node is forced to send a *Hello Message*, each covered node will explore all its edges from the original graph looking for uncovered nodes. The algorithm will not stop until all nodes are covered in the total area of coverage of the tree, exploring all edges from this set of nodes. This means that the final tree is a connected subgraph itself because it has no edge to uncovered nodes. If there exists an uncovered node it means that there is no edge between any of the covered nodes and the uncovered node. Then the initial graph cannot be connected because it has at least two non connected sets of nodes, which contradicts the initial assumption of a connected graph.

It is worth emphasizing that A3 is completely distributed and needs no synchronization scheme. The process finishes when the last node finishes its own creation process. Each node is responsible for its own process and needs no information about the status of the over all process. Actually, nodes can start their application-related tasks as soon as they are selected as part of the CDS tree.

The computational complexity of the A3 algorithm can be easily calculated based on the fact that the sorting function executed by the parent nodes is the most expensive operation. Therefore, the complexity of the algorithm is given by the complexity of sorting routine, which can be bounded as $O(nlog(n))$. The message complexity is bounded by the worst case of 4 messages in the case of a node that becomes a parent node in the first opportunity: *Hello*, *Parent Recognition*, *Children Recognition*, and *Sleeping* messages. However, the number of messages is still defined by $O(n)$ for a network with n nodes, with a worst case scenario of $4n$ messages.

A. The selection metric

As explained before, nodes receiving a *Hello Message* calculate the RSSI of the signal from the parent node, and then calculate a metric that is sent back to the parent in the *Parent Recognition Message*. Upon receiving the metrics from all its children, the parent node creates and sends a sorted list that, at the end, determines which nodes will be part of the tree. The A3 algorithm calculates the benefit of including a new node in the tree using a metric that is proportional to the remaining energy of the node and the distance from the parent node, as defined in Equation 1:

$$Mx, y = W_E \cdot \frac{E_x}{E_{max}} + W_D \cdot \left(\frac{RSSI_y}{RSSI^*}\right) \tag{1}$$

where x is the candidate node, y is its parent node, W_E is the weight for the remaining energy in the node, E_x is the remaining energy in node x, E_{max} is the maximum initial energy, W_D is the weight for the distance from the parent node, $RSSI_y$ is the received signal strength from the parent node, and $RSSI^*$ is the minimum RSSI to ensure connectivity, which is given by the sensitivity of the receiver. Equation 1 produces a value between 0 and 1 that is assigned to each neighbor in the process of selecting the new nodes in the tree; the higher the value of the metric, the higher the priority.

As it can be seen from Equation 1, the selection metric gives priority to those nodes with higher energy and which are farther away from the parent node. The final effect of this choice is to have a tree with fewer nodes and better coverage. However, proper weight manipulation can satisfy different criteria, as needed by the network operator. If communication coverage is to be optimized and the average height of the node in the tree (number of hops) needs to be reduced, the distance metric must be weighted more heavily. The downside is that low energy nodes may be included in the tree, which may introduce early failures of nodes, and therefore reduce the lifetime of the tree. On the other hand, if reliability of the tree is desired, energy must be weighted more. The downside is that the tree may present more active nodes. In this paper, a balanced average that compromises these two aspects is used.

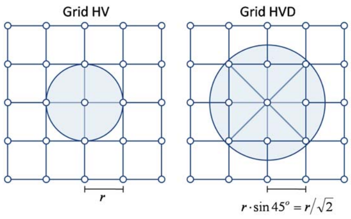


Fig. 2. The ideal grid scenarios: Grid HV and Grid HVD.

The study of the effects of these weights is part of our current investigation, whoever in preliminary results, we have seen that its main influence when the network have been working for some time and the topology maintenance protocol needs be executed.

IV. Performance Evaluation

In this section the results of the performance evaluation comparing the A3 algorithm with the EECDS and the CDS-Rule-K algorithms are presented. In the performance evaluation of the algorithms, the following assumptions were made:

- All nodes are located in a two dimensional space and have a perfect communication coverage disk.
- Nodes have no information about their position, orientation, or neighbors.
- The initial graph, the one formed right after the deployment, is connected.
- Distances can be calculated as a metric perfectly proportional to the Received Signal Strength Indicator (RSSI).
- There is no packet loss at the Data Link Layer.

Three sets of experiments are included. The first set maintains the number of nodes fixed and increases the node degree by changing the communication range of the nodes. The second set, on the other hand, varies the network density by changing the number of nodes while maintaining a fixed communication range. In these two experiments, the nodes are uniformly distributed in the area of deployment. The third set of experiments includes an additional theoretical comparison considering an ideal grid topology in which all nodes have the same number of neighbors. Two different grid topologies are used: the Grid HV topology, in which each node can listen to its horizontal and vertical neighbors; and the Grid HVD topology, in which each node can listen to its horizontal, vertical, and diagonal neighbors. Therefore, the number of neighbors in those topologies is at most 4 and 8, respectively. Figure 2 shows these scenarios. All the experiments, each result represents the average value of 50 random scenarios.

The critical transmission range, CTR, is used to vary the communication range of the nodes. By definition, the $CTR(n)$ is the minimum communication range that *w.h.p.* will produce a connected topology. The theoretical formula for the CTR presented in [7] is given by Equation 2:

$$CTR\left(n\right) = \sqrt{\frac{ln\left(n\right) + f(n)}{n\pi}} \tag{2}$$

TABLE I

Simulation Parameters.

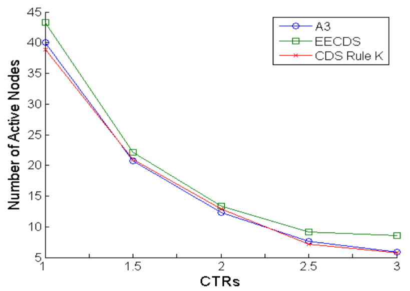
	Experiment 1	Experiment 2	Experiment 3
Deployment area	200mx200m		
Number of nodes	100	10,20,40,60,80 and 100	36 and 64
Transmission Range - Distances based on RSSI	1, 1.5, 2 and 3xCTR(100) equivalent to: 28m, 42m, 56m, 70m and 84m (Eqn. 2)	63m equivalent to 1xCTR(10)	40m
Node Distribution	Uniform (200,200)	Uniform (200,200)	Grid HV and Grid HVD
Instances per topology	50 instances and 3 replicates each		
E_{max}	1 Joule		
A3 Weights	$W_E = 0.5, W_D = 0.5$		
Energy Consumption	Eelec = $50nJ/bit$; Eamp = $10pJ/bit/m^2$ Short Messages = $25Bytes$ *Hello, Parent Recognition and Sleeping Messages* Long Messages = $100Bytes$ *Children Recognition and Data messages* Idle state energy consumption assumed negligible		

where $f(n)$ is an arbitrary function such that $lim_{n\to\infty}f\left(n\right) = +\infty$. However, Equation 2 only applies to uniformly distributed deployments and, more importantly, dense networks with very large values of n. Based on experimentation, it has been seen that this CTR provides an average number of neighbors between 4 and 8, even for small values of n. As in [8], we made $f(n) = log(log(n))$.

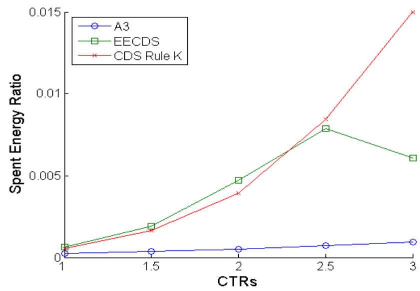
Three main performance metrics are utilized to assess the performance of the topology construction algorithms: 1) number of active nodes; 2) number of messages used in the CDS building process; and 3) amount of energy used in the process. The first metric shows how the topology construction mechanism can effectively reduce the amount of active nodes while preserving network connectivity and coverage. The other two metrics show how efficient the algorithm is in terms of overhead and energy consumption. The algorithms are evaluated in scenarios with sparse, medium dense, and dense topologies. The node degree and the density of the network are modified by increasing the communication range of the nodes and the number of nodes in the network. The three algorithms were implemented in a Java-based simulation tool called Atarraya [10], designed with the purpose of testing TC algorithms. Table I presents a summary of the simulation parameters used in the experiments.

A. Experiment 1: Changing the node degree

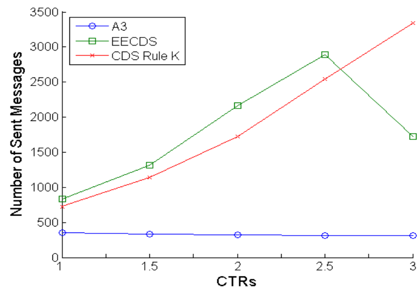
The main goal of this experiment is to compare the algorithms when the node degree of the network is changed by increasing the transmission range of the nodes. Given that these algorithms work based on information from neighbors, it is important to measure their performance with neighborhoods of different sizes. As it can be seen from Figure 3(a), the three algorithms produce trees with almost similar number of nodes; however, A3 generates fewer nodes in all scenarios. Also, all the algorithms tend to decrease the number of active



(a) Number of actives nodes.

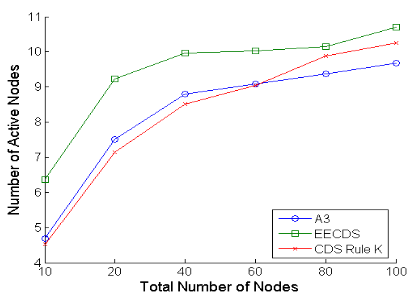


(b) Energy used in the CDS creation.

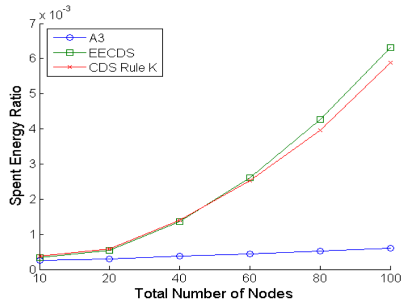


(c) Number of messages sent.

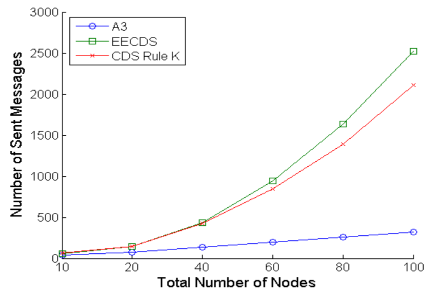
Fig. 3. Experiment 1 results: Changing the node degree.



(a) Number of actives nodes.



(b) Energy used in the CDS creation.



(c) Number of messages sent.

Fig. 4. Experiment 2 results: Changing the node density.

nodes with the node degree, as expected. Figures 3(b) and 3(c) show two important metrics: the total energy and number of messages used to build the CDS trees, respectively. In this case, the A3 mechanism shows its superior performance. A3 presents an almost constant energy consumption and number of messages compared with the CDS-Rule-K and EECDS algorithms, which show a non-linear increase trend. These results can be easily explained. The non-linear behavior of the CDS-Rule-K mechanism is explained by its pruning process in which every node must update nodes two hops away when it is unmarked. This overhead increases with the number of neighbors because more nodes will retransmit the message. Also, when the node degree increases, more nodes get unmarked and will produce this extra overhead. In the case of the EECDS algorithm, the factor that increases the amount of messages (and energy, consequently) is related to the competition used in both phases of the algorithm. This is due to the fact that with a higher communication range, more nodes are covered, and the tree has fewer nodes in higher levels. This, at the same time, reduces the amount of nodes competing to become part of the tree in the outer regions of the topology. The linearity of A3 is a consequence of the bounded number of messages that each node needs to transmit, which remains almost identical and never goes over $4n$ in ideal conditions.

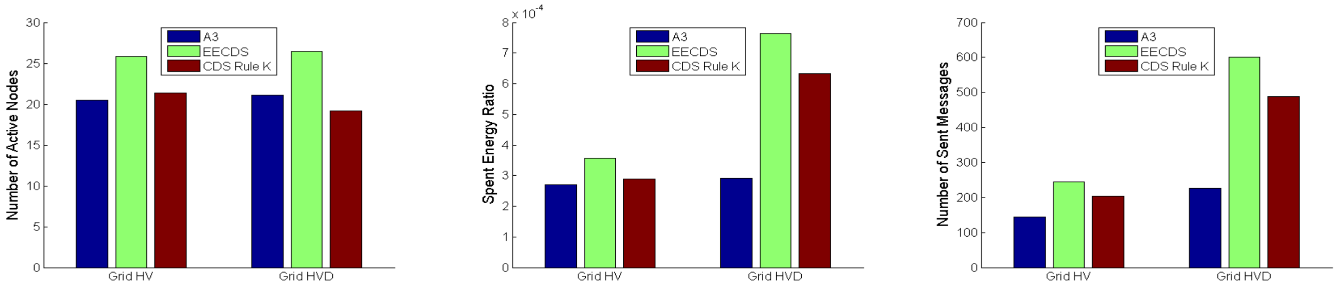
B. Experiment 2: Changing the node density

The main goal of this experiment is to compare the results produced by the algorithms when the network density is changed by varying the number of nodes in the area while keeping a fixed communication range of 63, equivalent in this experiment to $1xCTR(10)$. This experiment is important to show how scalable the algorithms are in dense topologies and how the resource usage depends on the number of nodes. The results, shown in Figure 4, are similar to the ones shown in Experiment 1. Figure 4(a) shows that all algorithms need a similar amount of active nodes, although before 60, CDS-Rule-K shows a small advantage over A3, after 60 both EECDS and CDS-Rule-K algorithm go above A3.

In terms of the message complexity and energy efficiency, the trends are similar. The CDS-Rule-K and the EECDS algorithms present a non-linear increase, while the A3 algorithm shows a low and linearly bounded number of messages and energy consumption (Figures 4(b) and 4(c)). *This shows that the A3 algorithm is scalable and is not highly affected by the number of nodes deployed.*

C. Experiment 3: Ideal grid topologies

The third experiment considers the ideal grid scenario with its two variants: HV and HVD, as shown in Figure 2. This experiment shows the performance of the algorithms in a perfectly homogeneous topology, with ideal condition of density and node degree, which could be considered a predefined scenario such as an office building sensor network. From Figure 5, it can be seen that the A3 algorithm shows similar or better results in the number of active nodes metrics, including 57% of the nodes in the grid HV and 33% in the grid HVD scenarios, versus 72% and 41% from EECDS, and 59% and 30% from CDS-Rule-K algorithms. The other two



(a) Number of actives nodes.

(b) Energy used in the CDS creation.

(c) Number of messages sent.

Fig. 5. Experiment 3 results: Performance using ideal grid topologies.

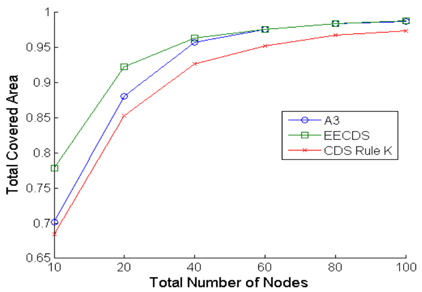


Fig. 6. Total area of communication coverage.

metrics show an increasing trend for EECDS and CDS-Rule-K while A3 still shows a bounded cost in overhead and energy.

D. *Area of communication coverage*

After the execution of the algorithms, the active nodes determine the communication coverage. This area is expected to cover as much of the deployment area as possible. Figure 6 shows the average communication area covered by the algorithms using the scenarios from Experiment 2. As it can be seen from this figure, and based on the results of the other experiments presented before, we can say that although all algorithms produce an almost similar coverage with the selected active nodes, A3 is still better; A3 covers the same or more area but using fewer resources than the other algorithms.

V. Conclusion

This paper presents the A3 topology construction algorithm to save energy consumption in wireless sensor networks. A3, a Connected Dominating Set-based algorithm, turns unnecessary nodes off while preserving the network connectivity and the communication coverage. Simulation results show the superiority of the A3 algorithm compared with the known EECDS and CDS-Rule-K algorithms in terms of number of active nodes needed, message complexity, and energy efficiency. The results show that A3 only needs 6% and 41% of the nodes active in dense and sparse scenarios while preserving network connectivity and communication coverage, versus 8% and 43% and 5% and 43% of the EECDS and CDS-Rule-K algorithms. More importantly, A3 provides a low linearly bounded number of messages and energy usage, compared with non-linear increasing trends shown by the CDS-Rule-K and EECDS algorithms. This last aspect is extremely important in order to use this algorithm in a complete topology control solution where the CDS tree will have to be changed many times. As part of our future work is to determine an approximation ratio to the optimal solution, as a performance metric of the algorithm.

References

[1] S. Butenko, X. Cheng, C. Oliveira, and P. M. Pardalos. *A New Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Networks*, pages 61–73. Kluwer Academic, 2004.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[3] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[4] V. Kumar, T. Arunan, and N. Balakrishnan. E-span: Enhanced-span with directional antenna. In *Proceedings of IEEE Conference on Convergent Technologies for Asia-Pacific Region*, volume 2, pages 675–679, 2002.

[5] L. Li, J. Y. Halpern, P. Bahl, Y-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, pages 264–273, 2001.

[6] N. Li, J. C. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. In *Proceesdings of IEEE INFOCOM*, volume 3, pages 1702–1712, 2003.

[7] M. Penrose. The longest edge of a random minimal spanning tree. *The Annals of Applied Probability*, 7(2):340–361, 1997.

[8] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, September 2005.

[9] P. J. Wan, K. Alzoubi, and O. Frieder. Distributed well connected dominating set in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM*, 2002.

[10] P. M. Wightman. Atarraya: A topology control simulator. http://www.cse.usf.edu/~pedrow/atarraya/.

[11] J. Wu, M. Cardei, F. Dai, and S. Yang. Extended dominating set and its applications in ad hoc networks using cooperative communication. *IEEE Trans. on Parallel and Distributed Systems*, 17(8):851–864, 2006.

[12] J. Wu and F. Dai. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 15(10):908–920, 2004.

[13] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd ACM international Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 7–14, 1999.

[14] Z. Yuanyuan, X. Jia, and H. Yanxiang. Energy efficient distributed connected dominating sets construction in wireless sensor networks. In *Proceeding of the 2006 ACM International Conference on Communications and Mobile Computing*, pages 797–802, 2006.