Distributed Detection of Critical Nodes in Wireless Sensor Networks Using Connected Dominating Set

Orhan Dagdeviren
International Computer Institute
Ege University
Izmir, Turkey
orhan.dagdeviren@ege.edu.tr

Vahid Khalilpour Akram
International Computer Institute
Ege University
Izmir, Turkey
vahid59@gmail.com

Bulent Tavli
Electrical and Electronics Engineering Department
TOBB-ETU
Ankara, Turkey
btavli@etu.edu.tr

Huseyin Ugur Yildiz
Electrical and Electronics Engineering Department
TED University
Ankara, Turkey
huguryildiz@ieee.org

Can Atilgan
Department of Computer Sciences Dokuz Eylul University
Izmir, Turkey
can.atilgan@deu.edu.tr

***Abstract*— A critical node (cut vertex) in a wireless sensor network (WSN) is a node which its failure divides the network to disconnected parts. Identification of such nodes is the first step in countering against the threats on WSN reliability. In this paper, we propose the Connected Dominating Set based CUT vertex detection (CDSCUT) approach. We define 5 new rules for deciding about the statuses of nodes. The proposed algorithm has O(Δ^2) local computation complexity and O($\log_2 n$) bit complexity per node where n is the node count and Δ is the maximum node degree. The simulation results show that our proposed approach achieves up to 30% improvement in percentage of status detection compared to the previous CDS based approach while consuming similar time and energy.**

***Keywords*— Wireless Sensor Networks, Critical Node Detection, Connected Dominating Set, Reliability, Connectivity.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are widely used in many applications including military services, healthcare, disaster relief, industrial automation, intelligent structures, etc. Generally, there is no communication infrastructure in WSNs. The nodes connect to each other in an ad hoc manner and collaborate to forward the sensed data or messages to the target nodes. We can model a WSN as a graph $G = (V, E)$ where V is the set of nodes (vertices) and E is the set of links (edges). G is connected if there is at least one path between each pair of nodes in V. Node $v \in V$ is critical if its removal increases the number of connected components in G [1-3]. For example in Fig. 1, black nodes are critical. If node 1 stops working, for example, the network is divided into three disconnected components. Detecting critical nodes provides useful information about the fault tolerance of the network. For example, identification of the critical nodes in a WSN, will enable the network management authority to take counter measures to reinforce these potential security risks.

A dominating set (DS) is a subset of nodes $D \subseteq V$ such that every node $v \in V$ is in D or has at least one neighbor in D. All nodes $u \in D$ are called dominators and each node $w \notin D$ is a ordinary node (dominatee). D is a connected DS (CDS) if nodes $u \in D$ remains connected after removing all ordinary nodes from G. In Fig. 1, for example, $\{2,8,5\}$ is a minimum dominating set, and $\{1,2,5,7,8\}$ is a minimum CDS. CDS is a very useful structure for routing in WSNs and usually is used as a backbone of the network to facilitate the reliable communication with remote nodes [4].

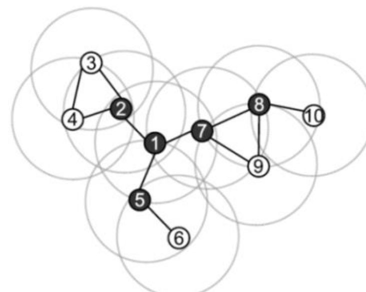

Fig. 1. The network model.

Various distributed depth first search (DFS) and other spanning tree based algorithms have been presented for finding the critical nodes [1-3]. In these algorithms, a spanning tree is implemented as an extra service with many message passings which increases the network traffic and total consumed energy. To the best of our knowledge, PArtition Detection and Recovery Algorithm (PADRA) [5] is the only distributed algorithm that detects critical nodes using CDS.

In this paper, we propose a distributed CDS based critical node detection algorithm with 5 new rules. Our algorithm is distributed and does not require a localization operation. Hence, it is suitable for large scale sensor network applications. We implemented our algorithm and PADRA and also a central algorithm (CENTRAL) on TinyOS-TOSSIM simulator and showed that our algorithm improves the detection percentage of PADRA up to 30%. Besides, the wall clock time and energy consumption of our algorithm are significantly better than the CENTRAL algorithm.

II. BACKGROUND

PADRA finds the critical nodes from CDS in a WSN without sending extra messages. The rules of the PADRA are as follows:

- PADRA Rule 1 (PR1): Ordinary nodes are not critical.
- PADRA Rule 2 (PR2): A dominator node is critical if it has a neighbor which is not connected to another node.

An example operation of the PADRA is given in Fig. 2. Nodes 3, 4, 6, 9 and 10 are ordinary nodes so PR1 identifies them as non-critical nodes. According to PR2, node 5 and node 8 are critical and nodes 1, 2 and 7 cannot be identified by the PADRA.

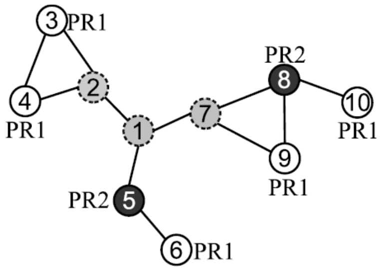

Fig. 2. An example operation of PADRA.

Finding the minimum CDS is an independent and interesting problem which is the subject of many studies. In our simulations, we have used the same marking based algorithm with the PADRA to find the CDS in the networks [5]. The rules of this algorithm are as follows:

- If node u has at least two neighbors that have no edge between themselves, then node u is a dominator.
- If node u has a dominator neighbor with a larger id which is connected to all neighbors of u then node u is not a dominator.
- If node u has two connected dominator neighbors v and z with larger ids, such that the union of their neighbors cover the neighbors of u then u is not dominator.

III. PROPOSED APPROACH

To detect the statuses of nodes, we proposed a CDS based critical node (CUT vertex) detection algorithm (CDSCUT) with 5 new rules which takes more situations into consideration than the PADRA. Let $\Gamma(u)$ shows the neighbor list of node u, $\Gamma_O(u)$ be the ordinary neighbors of u, $\Gamma_D(u)$ shows the dominator neighbors of node u and G/u be the graph without node u. We use $U \sim V$ to show that every node $u \in U$ has at least one path to every node $v \in V$.

Rule1 (R1): *A dominator node u is not critical if $\exists v \in \Gamma(u)$ such that $\Gamma(u) \subseteq \Gamma(v)$.*

According to R1, a dominator node u is not critical if it has a neighbor v which is directly connected to all other neighbors of u. The nodes in D form a connected subgraph of G such that $\forall v \notin D{:}\, \exists u \in D$ where $u \in \Gamma(v)$. Therefore, $\forall v \in V{:}\, \Gamma_D(v) \neq \emptyset \rightarrow v \sim V$ is true. If a dominator node u has a neighbor v such that $\Gamma(u) \subseteq \Gamma(v)$ then either $\Gamma(v) = V/v$ or $\exists w \in \Gamma(v)$ such that $w \neq u \wedge w \in D$, because u is a part of CDS. In the first case, node v keeps the connectivity of all other nodes in G/u which implies that u is not critical. For the second case in G/u we have $\Gamma_D(v) = \{w\} \rightarrow v \sim V \rightarrow \Gamma(v) \sim V \rightarrow \Gamma(u) \sim V$. So in G/u all nodes remain connected and u is not critical. For example, in Fig. 3.a, node 3 is not critical because $\Gamma(3) \subseteq \Gamma(1)$. Let G_u shows the two-hop local subgraph of node u. Getting a separated G_u/u is a required, but not sufficient, condition for any critical node u.

Rule2 (R2): *A dominator node u is not critical if G_u/u is a connected graph.*

In other words, a critical node u in G is also critical in G_u. If u is critical in G then $\exists (x, y) \in V$ such that $x \nsim y$ in G/u. So there is only one vertex disjoint path between x and y as $p(x, y) = \{x, .. i, u, j, .. y\}$ where $i, j \in \Gamma(u)$. If the neighbors of u remain connected in G/u then $x \nsim y \wedge i \sim j$ in G/u will be a contradiction because $i \sim j$ keeps the connectivity between x and y in $p(x, y)$. In Fig. 3.b, for example, node 4 is not critical because the $G_4/4$ is connected. However his rule is not valid for node 2.

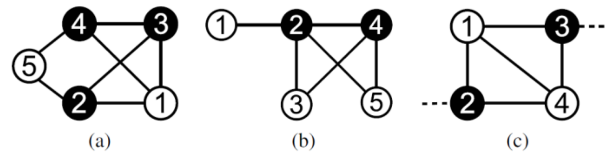

Fig. 3. (Black nodes are dominators) a) Node 3 is not critical (R1). b) Node 4 is not critical (R2). c)Nodes 2 and 3 are not critical (R3).

Rule3 (R3): *A dominator node u is not critical if in G_u/u we have $\forall v \in \Gamma_D(u){:}\, v \sim \Gamma_D(u)$ and $\forall v \in \Gamma_O(u){:}\, \Gamma_D(v)/u \neq \emptyset$.*

A dominator node u is not critical if $\Gamma_D(u)$ remain connected in G_u/u and all ordinary neighbors of u have at least one dominator neighbor in G_u/u. The connectivity of D is not violated if all dominator neighbors of u remain connected in G_u/u. So, if $\forall v \in \Gamma_D(u){:}\, v \sim \Gamma_D(u)$ in G_u/u then $\forall v \in D{:}\, v \sim D$ in G/u. Each ordinary node $v \notin D$ has at least one neighbor in D hence $\forall v \in \Gamma_O(u) : \Gamma_D(v)/u \neq \emptyset \rightarrow v \sim D \rightarrow v \sim V$ in G/u graph. Therefore, in G_u/u we have $\forall v \in \Gamma_D(u){:}\, v \sim \Gamma_D(u) \ \wedge\ \forall v \in \Gamma_O(u){:}\, \Gamma_D(v)/u \neq \emptyset \rightarrow \forall v \in \Gamma(u){:}\, v \sim V$ which indicates that u is not critical in G. For example, in Fig. 3.c, nodes 2 and 3 are not critical because their neighbors have another dominator neighbor and without node 2 or node 3 D remains connected.

Rule4 (R4): *A dominator node u which is critical in G_u is critical in G if $\exists v \in \Gamma(u){:}\, \Gamma(v) \subseteq \{\Gamma(u) \cup u\} \wedge v \nsim \Gamma(u)/\Gamma(v)$ in G_u/u.*

If u is a critical node in G_u, then $\exists x \in \Gamma(u){:}\, x \nsim v$ in G_u/u. If u is critical node in the entire network then we must have $x \nsim v$ in G/u. Suppose that $x \sim v$ in G/u. So $\exists\, y \in \Gamma(v) :\ x \sim y$. Without lose of generality, we can assume that y is a neighbor of x in G. We have $\forall w \in \Gamma(v){:}\, \Gamma(w) \subseteq \{\Gamma(v) \cup v\} \rightarrow \Gamma(y) \subseteq \{\Gamma(v) \cup v\} \rightarrow x \in \Gamma(v)$. So v must be a neighbor of node x which is a contradiction because if there is an edge between x and v then we always have $x \sim v$ in G_u/u. For example, in Fig 4.a, $G_4/4$ is not a connected graph and $\Gamma(1) \subseteq \Gamma(2)$. Hence, node 4 is critical.

Rule5 (R5): *A dominator node v is critical if it has a critical neighbor u such that $\Gamma_D(u) = \{v\}$ and $\forall w \in \Gamma(u)/v{:}\, \{\Gamma_D(w) = u \wedge \Gamma(w) \subseteq \Gamma(u) \cup u\}$.*

We have $\forall w \in \Gamma(u)/v{:}\, \{\Gamma_D(w) = u \wedge \Gamma(w) \subseteq \Gamma(u) \cup u\}$ so $\nexists\, w \in \Gamma(u)/v{:}\, w \sim v$ in G/u and $\nexists\, w \in \Gamma(u)/v{:}\, w \sim \Gamma(v)$ in G/u. Also $\forall w \in \Gamma(u)/v{:}\, \Gamma_D(w) = u$ so $\Gamma(u) \cap \Gamma(v) = \emptyset$. Consequently, without v there is no path to any node $w \in \Gamma(v)$ from u and $u \nsim \Gamma(v)/u$ in G/v which proves that v is critical. For example in Fig. 4.b, the critical node 4 is the only dominator neighbor of nodes 1 and 2 and the $\Gamma(1)$ and $\Gamma(2)$ are the subsets of $\Gamma(4) \cup 4$. So, node is critical.

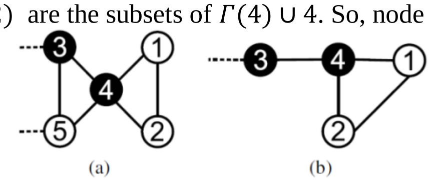

Fig. 4. a) Node 4 is critical (R4). b)Node 3 is critical (R5).

Since the rules run on two-hop neighbor lists of nodes, the total computational complexity of all proposed rules is $O(\Delta^2)$ where Δ is the maximum degree of a node in the network. The total message complexity is $O(1)$ per node (due to rule 5). The total bit complexity is $O(log_2 n)$ per node since the message size is limited by $O(log_2 n)$ where n is the node count.

IV. Performance Evaluations

To evaluate the performance of the proposed approach, we implemented PADRA, CDSCUT and a CENTRAL algorithm in TinyOS-TOSSIM simulator [6]. In CENTRAL algorithm, all nodes find their one-hop neighbor list and send it to a single (sink) node to find the cut vertices using a central DFS based algorithm. We generated 5 class of random geometric networks varying from 50 to 250 nodes by incrementing 50 nodes at each step. For n=50, the average detected cut vertices in CENTRAL algorithm (Fig. 5.a) is about 6 while this value for CDSCUT and PADRA are 5.4 and 4.3 respectively. For n=250, CDSCUT has found about 6 more cut vertices than the PADRA. The CENTRAL algorithm has found approximately 5 more cut vertices than CDSCUT and 11 more cut vertices than PADRA in average. The proposed algorithm has found at least 73% of all cut vertices (about 14.6 from 20 nodes for n=250) which is about 30 % higher than PADRA. PADRA determines the statuses (critical or non-critical) of about 60% of the nodes in average and this value slightly increases by adding more nodes to the network (Fig. 5.b). The CDSCUT algorithm determines the statuses of more than 80% of the nodes for n=50 and about 90% of nodes for n=250. The total consumed energy in CENTRAL algorithm is significantly higher than the other algorithms (Fig. 5.c). CENTRAL algorithm consumes about 180J energy in the networks with 250 nodes which is 8 times higher than CDSCUT and PADRA where the total consumed energies of CDSCUT and PADRA are similar.

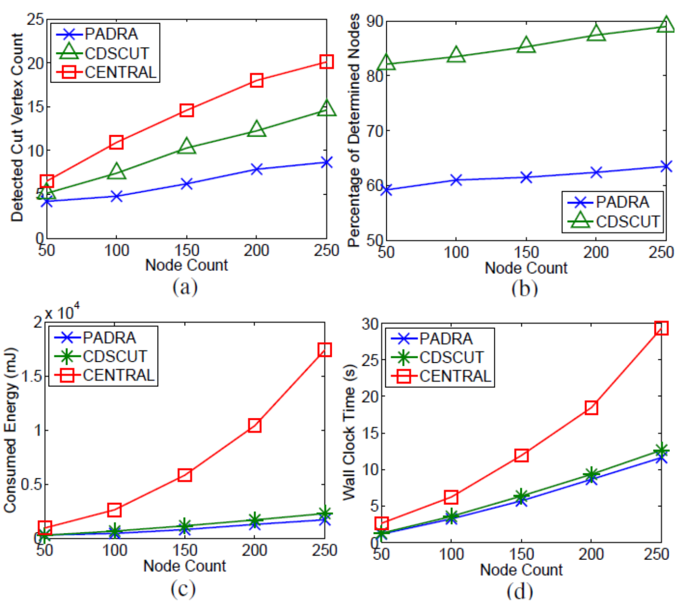

Fig. 5. Performance comparison of algorithms against node count a) Detected Cut Vertex Count b) Percentage of Determined Nodes c) Consumed Energy d) Wallclock Time

Collecting the entire topology information in a single node increases the wall clock time of CENTRAL algorithm dramatically when we increase the node count (Fig. 5.d). In contrast, the wall clock times of PADRA and CDSCUT show a linear function of node count. The wall clock time of PADRA is slightly lower than that of CDSCUT. However, the maximum difference is less than 2 s for all topologies. For n=250, the CENTRAL algorithm takes about 30 s to complete while this interval for PADRA and proposed algorithm are 11 s and 13 s respectively which means that for n=250, the proposed algorithm is 2.3 times and PADRA is 2.7 times faster than the CENTRAL algorithm.

V. Conclusion

In many WSN deployments reliability of the network is of utmost importance (e.g., critical infrastructure monitoring) and network partitioning should be avoided to be able to monitor the whole deployment area. As such, the capability to identify the critical nodes that can leave the network partitioned when incapacitated is imperative. In this paper, we propose the CDSCUT algorithm for deciding about the status (critical or non-critical) of the nodes in WSN from CDS information. The algorithm has 5 additional rules over the PADRA algorithm where these rules can be executed on two-hop neighbor list of nodes without applying a network wide spanning tree algorithm. The local computation overhead of the rules are bounded by $O(\Delta^2)$. Besides, the message complexity is $O(1)$ per node and the bit complexity is limited by $O(log_2(n))$ per node. The proposed approach has up to 30% improvement in finding the critical nodes on WSNs while it keeps the energy consumption and wall clock time close to the PADRA algorithm.

Acknowledgment

The authors would like to thank the TUBITAK (Scientific and Technical Research Council of Turkey) for financial support of the project 113E470 which this work belongs to it.

References

[1] O. Dagdeviren and V. K. Akram, “An energy-efficient distributed cut vertex detection algorithm for wireless sensor networks,” The Computer Journal, vol. 57, no. 12, pp. 1852–1869, 2014.

[2] V. Turau, “Computing bridges, articulations, and 2-connected components in wireless sensor networks,” in Algorithmic Aspects of Wireless Sensor Networks, pp. 164–175, Springer, 2006.

[3] S. Xiong and J. Li, “An efficient algorithm for cut vertex detection in wireless sensor networks,” in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, pp. 368–377, IEEE, 2010.

[4] J. Wu and H. Li, “On calculating connected dominating set for efficient routing in ad hoc wireless networks,” in Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications, pp. 7–14, ACM, 1999.

[5] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, “Distributed recovery of actor failures in wireless sensor and actor networks,” in Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE, pp. 2480–2485, IEEE, 2008.

[6] P. Levis, N. Lee, M. Welsh, and D. Culler, “Tossim: Accurate and scalable simulation of entire tinyos applications,” in Proceedings of the 1st international conference on Embedded networked sensor systems, pp. 126–137, ACM, 2003.