RELAY DEPLOYMENT ALGORITHMS IN WIRELESS SENSOR NETWORKS: A SURVEY

Lanny Sitanayah
Informatics Engineering Department; Faculty of Engineering
Universitas Katolik De La Salle Manado; Kombos – Kairagi I Manado, Telp:(0431) 871957
e-mail: lsitanayah@unikadelasalle.ac.id

Abstract-Wireless Sensor Networks (WSNs) are subject to failures. Even though reliable routing protocols for WSNs exist and are well-understood, the physical network topology must ensure that alternate routes with an acceptable length to the sinks are in fact available when failures occur. This requires a sensor network deployment to be planned with an objective of ensuring some measure of robustness in the topology, so that when failures do occur the protocols can continue to offer reliable delivery. To ensure that sensor nodes have sufficient paths, it may be necessary to add a number of additional relay nodes, which do not sense, but only forward data from other nodes. In this paper, we review a range of existing algorithms to deploy relay nodes for fault-tolerance. We classify the state-of-the-art relay placement algorithms based on routing structures, connectivity requirements, deployment locations, and fault-tolerant requirements.

Keyword-wireless sensor networks, network deployment planning, relay placement.

I. INTRODUCTION

To be able to offer reliable delivery when failures occur, a communication protocol depends on a physical network topology that guarantees alternative routes to the sink are in fact available. Therefore, one key objective in the topology planning of a Wireless Sensor Network (WSN) is to ensure some measure of robustness. In particular, one standard criterion is to make sure routes to the sink are available for all remaining sensor nodes after the failures of some sensor nodes or radio links. In addition, since there are sometimes data latency requirements, there may be a limit to the path length from sensor to sink. This can be achieved by planning the deployment so that every sensor node in the initial design has disjoint paths with a length constraint to the sink. To ensure that the sensors have sufficient paths, it may be necessary to add a number of additional relay nodes, which do not sense, but only forward data from other nodes.

In network topology planning, sensor nodes, relays and sinks are represented by vertices, and the radio links between them by edges. Two paths are *vertex-disjoint* (respectively *edge-disjoint*) if both of them do not share any vertices (respectively any edges), except the source and the sink. Vertex-disjoint paths are more resilient to failures than edge-disjoint paths [1], because if a source node has *k* vertex-disjoint paths, it is guaranteed to have a path to the sink after the failure of up to either *k*-1 nodes or k-1 radio links. On the other hand, edge-disjoint paths only protect against link failures. Figure 1(a) illustrates a network where the source node *s* has 2 vertex-disjoint paths to the sink *t*, while the example in Figure 1(b) shows a network where *s* has 2 edge-disjoint paths to *t*, but the paths are not vertex-disjoint. Since we are only interested in vertex-disjoint paths, we will use the term disjoint paths for short throughout this paper, unless we want to differentiate it from the edge-disjoint ones.

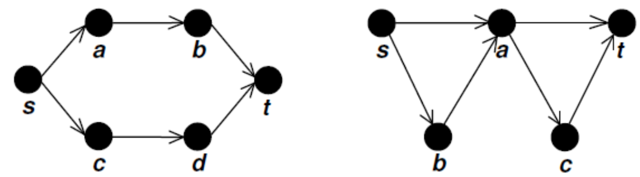

Figure 1. Examples of (a) vertex-disjoint and (b) edge-disjoint paths from the source *s* to the sink *t*

Finding several disjoint paths between a source and a sink is motivated by the following advantages [2]:

1. *Improving network reliability and survivability*. The network can use the alternative paths on demand to deliver messages if a path fails or becomes congested and cannot provide the required quality of service. The availability of *k* disjoint paths is able to tolerate failure of up to *k*-1 nodes.
2. *Providing multi-path routing capability*. Multi-path routing protocols can use all routes simultaneously to minimise latency or to provide redundancy in data transmission. Multi-path routing makes failure much less likely as all disjoint paths must become disconnected to interrupt the transmission.

Installing additional relay nodes, to ensure that sensor nodes have sufficient paths, comes at a cost that includes not just the hardware purchase but more significantly the installation and ongoing maintenance, thus motivating solutions that minimize the number of additional relay nodes. The relay placement problem for WSNs is concerned with deploying a minimum number of relay nodes into the networks to guarantee certain connectivity and survivability requirements. A classification scheme for relay placement problems according to [3] and [4] is shown in Figure 2.

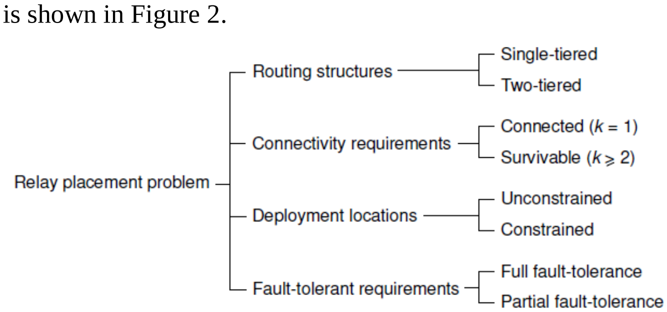

Figure 2. Relay placement problem classification [3][4]

Misra *et al*. [3] classify the relay placement problems based on the routing structures, the connectivity requirements and the deployment locations. Based on the routing structures, relay placement problems are categorised into *single-tiered* and *two-tiered*. In single-tiered, a sensor node also becomes a relay node to forward packets received from other nodes. The two-tiered network is a cluster-based network, where a sensor node only forwards its own data to a cluster head. Based on connectivity requirements, the problems are categorised into *connected* and *survivable*. In the connected relay placement, a small number of relay nodes is deployed to guarantee that the sensor nodes and the sinks or base stations are connected. In survivable relay placement, the relay nodes are placed to guarantee *k*-connectivity, where $k \geq 2$. Based on the deployment locations, the problems are divided into *unconstrained* and *constrained*. In the unconstrained relay placement, relay nodes can be placed anywhere. However, in practice, there are some limitations on the possible locations to deploy relay nodes. For example, relays cannot be placed at physical obstacles. In the constrained relay placement, relay nodes can only be deployed at a subset of candidate locations.

The relay placement problem is also classified based on the fault-tolerant requirements, *i.e. full fault-tolerance* and *partial fault-tolerance* [4][5]. Full fault-tolerance aims to deploy relay nodes in a network to establish *k*-connectivity between every pair of sensor nodes (original nodes) and relay nodes (additional nodes). Partial fault-tolerance aims to deploy relay nodes to establish *k*-connectivity only between every pair of sensor nodes as the original nodes. Full fault-tolerance has two properties [6]:

1. the network requires *k* node failures to disconnect it, and
2. there exist at least *k* vertex-disjoint paths between every pair of nodes in the network, not just between every node to a dedicated sink.

However, in some cases, partial fault-tolerance is preferable [4], because:

1. only the original nodes serve a useful purpose, the additional nodes merely provide additional connectivity,
2. partial *k*-connectivity is more economical than the full *k*-connectivity, because it requires fewer deployed relays.

The relay node deployment problem has long been acknowledged as significant. In this paper, we discuss the range of existing algorithms to deploy relay nodes for fault-tolerance. We categorise the reviewed algorithms based on the routing structures, *i.e.* single-tiered and two-tiered relay placement problems. Recall that in the two-tiered cases, sensor nodes are only within one hop from the relays that serve as cluster heads. Hence, the objective of the partial fault-tolerance is not to provide alternative paths for sensor nodes, but for relay nodes.

II. SINGLE-TIERED RELAY PLACEMENT PROBLEM

A. *k-Connectivity-Repair*

Bredin *et al*. [6] develop *k*-Connectivity-Repair as a centralised greedy algorithm and its distributed version for the single-tiered unconstrained full fault-tolerant relay placement problem to guarantee vertex *k*-connectivity. They assume that relay nodes have the same transmission range as sensor nodes and the range is normalized to one. The algorithm firstly computes a weighted complete graph, where the weight of an edge is one less than the Euclidean distance between a pair of sensors. The edge's weight represents the number of additional relays required to connect two sensors by a straight path. After that, this algorithm finds an approximate minimum-weight vertex *k*-connected subgraph by repeatedly adding edges in increasing order of weight until the subgraph is *k*-connected. If the subgraph is already *k*-connected, it repeatedly attempts to remove edges in decreasing order of weight, but putting the edge back if it is important for *k*-connectivity. Finally, it places clusters of *k* relays along each edge every one unit distance and *k*-1 relays at both endpoints of the edge.

The simulation results show that the distributed version of the algorithm nearly achieves the same number of required additional relays as the centralised greedy version. Moreover, compared to the random repair algorithm, where relays are scattered randomly until the *k*-connectivity is achieved, the two versions of *k*-Connectivity-Repair only require one seventh of the random repair cost to restore graph 3-connectivity.

B. Partial *k*-Connectivity-Repair

Pu *et al*. [4] propose Partial *k*-Connectivity-Repair by modifying the *k*-Connectivity-Repair algorithm by Bredin *et al*. [6] to guarantee only partial fault-tolerance. Partial *k*-Connectivity-Repair follows the same procedure as *k*-Connectivity-Repair to compute a weighted complete graph and to find a minimum-weight vertex *k*-connected subgraph. After that, instead of placing clusters of *k* relays along each edge every one unit distance and *k*-1 relays at both endpoints of the edge for full fault-tolerance, the proposed modification for partial fault-tolerance only deploys one relay every transmission range distance and none at the endpoints of each edge.

C. Connectivity-First

Han *et al*. [5] develop algorithms for the single-tiered unconstrained partial and full fault-tolerant relay placement problem for $k \geq 1$. They assume heterogenous WSNs, where sensors have different transmission radii, while relays use the same transmission radius. This asymmetric communication links together with the level of desired fault-tolerance divide the problem into four categories: one-way and two-way partial fault-tolerant, and one-way and two-way full fault-tolerant relay placement. The algorithms firstly calculate the weight of additional edges between each pair of sensors in a complete graph. The weight determines how many relays needed along a straight line between two sensors. It is calculated by dividing the Euclidean distance of the two sensors by the relay's transmission radius.

A greedy heuristic algorithm called Connectivity-First is then proposed to find the minimum *k*-connected spanning graph. It adds edges that can best help improving the connectivity until the graph becomes *k*-connected. An additional edge is selected because it has the highest contribution to the connectivity and has the least weight, *i.e.* the number of relays. The connectivity is checked using a maximum network-flow-based checking algorithm [7] as is used in [8]. When the graph is *k*-connected, the algorithm tries to remove redundant edges in decreasing order of weight as long as the removal does not break the *k*-connectivity. Finally, a number of relays is deployed along the selected additional edges. The results show that the algorithm by Bredin *et al*. [6] is more efficient for partial fault-tolerance, while Connectivity-First is more efficient for full fault-tolerance in terms of the number of relays that needs to be added to the network.

D. Redundant Router Placement

Ahlberg *et al*. [9] study the problem of single-tiered unconstrained partial fault-tolerant relay placement for $k = 1$ and $k \geq 2$. In the non-redundant relay placement ($k = 1$), they propose three algorithms:

1. Trivial Router Placement simply deploys relays on a straight line from each and every sensor to the sink.
2. Trivial Placement Reusing Routers sorts the sensors according to their distances to the sinks, connects the closest sensor to its sink by deploying relays on a straight line, and then connects the next closest sensor to the closest deployed relays or to the sink.
3. Cluster Router Placement groups nearby and connected sensors into a cluster and uses the Trivial Router Placement algorithm to connect clusters, instead of connecting each sensor separately.

For the redundant relay placement ($k \geq 2$), firstly the algorithm counts the number of available paths from each sensor to the sink using the Ford-Fulkerson maximum flow algorithm. If the number of available paths is not sufficient, the algorithm places redundant relays start from the furthest sensor from the sink.

Further, to reduce the number of deployed relays, Ahlberg *et al*. suggest two optimization techniques:

1. For the non-redundant placement ($k = 1$), all sensors are reconnected to the relay that has the shortest path to the sink. Relays with only connection to another relay are removed.
2. For the redundant placement ($k \geq 2$), each relay is temporarily removed and the number of available paths are recalculated, but placing it back if necessary.

E. Single-tiered and Two-tiered Fault-Tolerant Relay Placement (1tFTP and 2tFTP)

Zhang *et al*. [10] study the single-tiered and two-tiered unconstrained partial fault-tolerant relay placement problem for *k*-connectivity, where $k = 2$. Relay nodes are assumed to have larger transmission range than sensor nodes. The network may also have base stations. The proposed algorithms are:

1. *Single-tiered Fault-Tolerant Relay Placement* (1tFTP). It constructs a complete graph, computes a 2-connected spanning subgraph and steinerises the edges of the subgraph. The steinerisation process calculates edges' weight by dividing the Euclidean distance of any two vertices by the relay's transmission radius. For each edge, a number of relays is deployed along the straight line.
2. *Two-tiered Fault-Tolerant Relay Placement* (2tFTP). It uses the Two-tiered Relay Node Placement (2tRNP) algorithm that is developed for 1-connectivity proposed by Lloyd and Xue [11]. 2tRNP finds the minimum number of relays that can cover all sensors into one-hop clusters. Relays in all clusters are then connected by paths of additional relays. For this, 2tRNP finds the Steiner minimum tree with minimum number of Steiner points. 2tFTP then duplicates each of the relays found by 2tRNP.

1tFTP and 2tFTP are compared to two heuristics, 1tTSP and 2tTSP, that may produce close to optimal solutions. 1tTSP and 2tTSP compute a Traveling Salesman (TSP) tour of the graph and steinerise the edges of the tour to deploy relays. The simulation results show that in all cases with varied network density, the numbers of relays required by 1tFTP and 2tFTP are no more than 1.5 times the numbers of relays required by 1tTSP and 2tTSP.

F. Connected and Survivable Relay Node Placement (RNP_C and RNP_S)

Misra *et al*. [3] study the single-tiered constrained partial fault-tolerant relay placement problem for both the connectivity ($k = 1$) and the survivability ($k = 2$) requirements. They assume that the transmission range of sensor nodes is smaller than the transmission range of relay nodes. Misra *et al*. propose:

1. *Connected Relay Node Placement* (RNP_C) for $k = 1$. It firstly constructs the communication graph for sensors, base stations and relays' candidate locations. Then, it assigns edges' weight as the number of candidate relays they are incident with. Finally, the low weight tree subgraph is computed, from which the locations to place relays are identified. The unconstrained version of RNP_C is Single-tiered Relay Node Placement (1tRNP) studied by Lloyd and Xue [11], where there is no restriction on the locations of the relays.
2. *Survivable Relay Node Placement* (RNP_S) for $k = 2$. The algorithm constructs the communication graph and assigns edges' weight too. It then assigns connectivity requirements between every pair of vertices in the following way: $c(v, w) = 2$ if neither *v*, nor *w* is the candidate for relay. Otherwise, $c(v, w) = 0$. Then, the low weight 2-connected subgraph that meets the connectivity requirements is computed. Relays' candidate locations that appear in the subgraph are the positions to deploy additional relays.

In the simulation, RNP_C and RNP_S are compared to simulated annealing. The results show that they are able to produce almost the same numbers of relays as the results obtained by simulated annealing. Simulated annealing has 10 times longer running time, but only finds slightly better solutions in a few cases.

G. Greedy Randomised Adaptive Search Procedure for Additional Relay Placement (GRASP-ARP)

Sitanayah *et al*. [12][13] define the single-tiered constrained partial fault-tolerant relay placement problem for *k* disjoint paths with a length constraint for WSNs with data sinks, where each sensor node in the initial design has $k \geq 2$ length-bounded disjoint paths to one or more sinks. They present two centralised algorithms to be run during the initial topology planning:

1. Counting-Paths is a heuristic algorithm that counts the number of disjoint paths from each sensor node and finds the shortest disjoint paths to sinks. The basic Counting-Paths algorithm uses a maximum flow algorithm, such as Ford-Fulkerson [14] to find the actual *k* shortest disjoint paths, while the dynamic programming variant of Counting-Paths only finds *k* shortest disjoint paths to *k* neighbours that already have *k* disjoint paths.

2. Greedy Randomised Adaptive Search Procedure for Additional Relay Placement (GRASP-ARP) is a local search algorithm that uses Counting-Paths to minimise the number of relays that need to be deployed.

GRASP-ARP is compared to Partial *k*-Connectivity-Repair [4], which has been modified to work in constrained deployment locations. The simulation results show that GRASP-ARP outperforms *k*-Connectivity-Repair with fewer relay nodes and that GRASP-ARP with the dynamic programming variant of Counting-Paths are significantly faster for solving larger problems.

H. Greedy Randomised Adaptive Search Procedure for Additional Backup Placement (GRASP-ABP)

Another approach to *k*-connectivity and partial *k*-connectivity is to consider the relative importance of each node for delivering data to the sink from other nodes. If the failure of a node would disconnect many other nodes, or cause traffic from many other nodes to be delivered late, then the node is important, and we should ensure alternative paths around that node. The importance of a node in network analysis is called its *centrality* [15][16]. In [17], Sitanayah *et al*. introduce novel definitions of centrality which measure a node's impact on connectivity and path length for the rest of the network. Then, they use the centrality measure as a priority order for providing alternative paths. Thus, if resources resources are limited, only nodes with high centrality are addressed, with the intention of being robust to the most significant failures; in cases where more resources are available, nodes with lower centrality can also be addressed, and provide robustness against more failures.

Specifically, they define Length-constrained Connectivity and Rerouting Centrality (*l*-CRC), a new centrality index for WSNs with sinks. This centrality index has a pair of values. The first value measures the importance with respect to network connectivity under a path length constraint, while the second value measures the additional length of shortest paths that would be required after a node fails. The two values are later combined as Failure Centrality in [18].

Sitanayah *et al*. [17] study the single-tiered constrained partial fault-tolerant additional backup placement problem for *k* disjoint paths with a length constraint for WSNs with data sinks, where $k = 2$. They use the centrality index to determine the most critical nodes, and to assess the quality of positions for the relays to provide alternative paths around the nodes with high centrality. To decide whether a node is critical or not, they use a threshold. A node is critical if its centrality index is above the threshold. Rising the threshold can trade-off deployment cost for robustness. They introduce Greedy Randomised Adaptive Search Procedure for Additional Backup Placement (GRASP-ABP), a local search algorithm that searches for the smallest number of additional relays to ensure all sensor nodes have centrality measures below the threshold.

Simulation results show that GRASP-ABP deploys fewer additional relays with faster runtime compared to Partial *k*-Connectivity-Repair [4] and GRASP-ARP [12].

I. Greedy and Greedy Randomised Adaptive Search Procedure for Multiple Sink and Relay Placement (Greedy-MSRP and GRASP-MSRP)

To be robust to sink failure, multiple sinks are deployed in a network such that each sensor node is *double-covered*, *i.e*. it has length-bounded paths to two sinks. In [19][20], the relay placement problem is extended to include sinks, which are deployed together with relays for fault-tolerant multi-hop networks with a path length constraint. For the Multiple Sink and Relay Placement (MSRP) problem, Sitanayah *et al*. two algorithms, namely Greedy-MSRP and GRASP-MSRP. Both algorithms employ the concept of Length-constrained Connectivity and Rerouting Centrality (*l*-CRC) introduced in [17] to identify every *critical* node, *i.e*. a sensor node which if fails can cause other nodes to lose their length-bounded paths to sinks.

Greedy-MSRP deploys sinks and relays separately. Since the assumption is that the cost of a sink is more expensive than a relay node because it is usually powered, has large storage capacity and has WiFi/ethernet backhaul, Greedy-MSRP tries to trade some sinks for relays to minimise the total deployment cost but ensures that the network is still double-covered and non-critical. Unlike Greedy-MSRP, GRASP-MSRP minimises the number of uncovered and critical nodes simultaneously in its every local search move. Simulation results show that GRASP-MSRP runs faster than Greedy-MSRP and the solutions produced by GRASP-MSRP are over 30% less costly than those of Greedy-MSRP.

J. Constraint Programming (CP) Approach to the Additional Relay Placement Problem

A WSN is *k*-robust if an alternate length-constrained route to a sink is available for each surviving node after the failure of up to *k*-1 nodes. A WSN is *strongly* *k*-robust if there are *k* disjoint length-constrained routes to a sink for each node. Determining whether a network is *k*-robust is polynomial. However, determining whether a network is strongly *k*-robust is an NP-complete problem. Quesada *et al*. [21][22] develop a Constraint Programming (CP) approach for deciding strongly *k*-robustness that outperforms a Mixed-Integer Programming (MIP) model on larger problems. The CP model can solve problems in less time than it takes to generate the MIP models. To find qualifying paths for all sensors in the same network, the CP solution is able to solve the problems in reasonable time, but that the MIP model does not scale.

A network can be made (strongly) robust by deploying extra relay nodes. The CP approach is extended to an optimisation approach by using QuickXplain to search for a minimal set of relays, and compare it to GRASP-ARP [12]. The simulation results show that the approximate CP solution is competitive in time with the local search method on the larger problems, although with lower quality solutions. When the robustness requirement is relaxed by enforcing biconnectivity instead of disjointness, the QuickXplain based approach provides solutions that are not only competitive in time but also in quality.

III. TWO-TIERED RELAY PLACEMENT PROBLEM

A. 2-Connected Relay Node Double Cover (2CRNDC)

Hao *et al*. [23] propose an algorithm to solve the two-tiered constrained partial fault-tolerant relay placement problem for 2-

connectivity. Under an assumption that the distributed sensor nodes are already 2-connected, they want each sensor node to be able to communicate with at least two relay nodes and the network of the relays is 2-connected. They also assume that the relay nodes' transmission range is at least twice the transmission range of sensor nodes. In each iteration, the algorithm selects one relay from the set of candidate positions that can best cover as many sensor nodes, which are not covered by two relays, as possible. Then, it selects some relays from the set of candidate positions that can make the previously selected relay have two disjoint paths and become 2-connected. The algorithm proceeds until all sensors in the network are covered by at least two relays.

B. Connected Relay Node Single Cover (CRNSC) and 2-Connected Relay Node Double Cover (2CRNDC)

Tang *et al*. [24] study the problem of two-tiered unconstrained partial fault-tolerant relay placement for k = 1 and k = 2. Under an assumption that the transmission range of relay nodes is four times the range of sensor nodes, they propose:

1. *Connected Relay Node Single Cover* (CRNSC). It requires that each sensor node to be covered by at least one relay node, and that the set of relay nodes is connected.
2. 2-*Connected Relay Node Double Cover* (2CRNDC). It requires that each sensor node to be covered by at least two relay nodes and that the network induced by the relay nodes is 2-connected.

The main ideas of the algorithms are:

1. Divide the region into small cells of size l.2r, where l is an integer partition factor and r is the transmission range of sensor nodes.
2. For each cell, find all possible positions to deploy relays. Possible positions are the intersections of sensors' transmission circles of radius r. If a possible position is outside of a cell, it is replaced with the closest point on the border of the cell.
3. Without considering the connectivity, find the optimal solution to cover (k = 1) or double cover (k = 2) the sensor nodes within each cell using exhaustive search.
4. Make the network of relays connected (k = 1) or 2-connected (k = 2) by adding extra relays at some specific locations if necessary.

C. Minimum Relay-Node Placement for 1 and 2-Connectivity (MRP-1 and MRP-2)

Liu *et al*. [25] address the two-tiered unconstrained full fault-tolerant relay placement problem for k = 1 and k = 2. In the hierarchical network, relay nodes act as cluster heads and are connected with each other to perform data forwarding task. The proposed algorithms are:

1. *Minimum Relay-Node Placement for 1-connectivity* (MRP-1). The first step is finding the minimum number of relay nodes that can cover all sensor nodes. The network of relays may not be connected if the distance between them is larger than the transmission range. Therefore, more relays are needed. The second step is constructing Steiner tree to connect the relays such that the number of Steiner points, in this case the additional relays, is minimised.
2. *Minimum Relay-Node Placement for 2-connectivity* (MRP-2). To achieve 2-connectivity, MRP-2 adds three additional relay nodes in the transmission range's circle of each relay found in MRP-1.

These two algorithms can be utilised to the cases where the transmission ranges of sensor nodes and relay nodes are either the same or different.

D. k-Vertex Connectivity

Kashyap *et al*. [26] give algorithms for the two-tiered unconstrained and constrained partial fault-tolerant relay placement problem for edge and vertex k-connectivity, where $k \geq 2$. They assume a hierarchical network, where sensors forward data to cluster heads. Therefore, the network should have vertex-disjoint (or edge-disjoint) paths between each pair of cluster heads. Relay nodes are assumed to have the same communication capabilities as the cluster heads and the range is normalised to one. The algorithm for vertex k-connectivity starts by constructing a complete graph of cluster heads and calculating the edges' weight as the number of relays needed. The weight is calculated from the edge's length minus one. Then, the minimum cost vertex k-connected spanning subgraph is sought. After that, relays are placed along the additional edges of the resulting subgraph. Finally, the algorithm tries to remove relays, which are sorted arbitrarily, one by one by still preserving the vertex k-connectivity. The resulting graph is vertex k-connected.

IV. DISCUSSION

We show the comparisons of the reviewed relay placement algorithms for WSNs in Table 1. We compare the algorithms based on the connectivity requirements (k), the assumption made on the transmission ranges, *i.e*. R and r denote the transmission ranges of relay nodes and sensor nodes, respectively, the routing structures, the deployment locations, and the fault-tolerant requirements. Recall that for k = 1, the algorithm only guarantees that the network is connected. If $k \geq 2$, it guarantees survivability. Relay nodes can only be placed at a subset of candidate locations in the constrained deployment, but can be placed anywhere if the deployment locations are unconstrained.

In this paper, we assume that an initial WSN topology is connected and additional relays may be required for fault-tolerance. Even though relays may die during the network operation, we may only want to protect the network against sensor node failures because relays only provide additional connectivity to improve the network reliability and survivability. Based on the state-of-the-art relay placement algorithms that we reviewed in this paper, only GRASP-ARP [12][13], GRASP-ABP [17], Greedy-MSRP and GRASP-MSRP [19][20], and CP [21][22] take into account a path length constraint, which is an important factor in topology design as sometimes WSN applications have data latency requirements.

Table 1.
Comparison of existing relay placement algorithms

Algorithms	***k***	***R* vs *r***	**Routing**	**Deployment Locations**	**Fault-Tolerance**
<u>Single-tiered</u>					
k-Connectivity-Repair [6]	≥ 1	$R = r$	1-tiered	unconstrained	full
Partial *k*-Connectivity-Repair [4]	≥ 1	$R = r$	1-tiered	unconstrained	partial
Connectivity-First [5]	≥ 1	$R \geq r$	1-tiered	unconstrained	full/partial
Redundant Router Placement [9]	≥ 1	$R \geq 2r$	1-tiered	unconstrained	partial
1tFTP and 2tFTP [10]	2	$R \geq r$	1/2-tiered	unconstrained	partial
RNP_C and RNP_S [13]	1, 2	$R \geq r$	1-tiered	constrained	partial
GRASP-ARP [12][13]	≥ 2	$R = r$	1-tiered	constrained	partial
GRASP-ABP [17]	2	$R = r$	1-tiered	constrained	partial
Greedy-MSRP and GRASP-MSRP [19][20]	2	$R = r$	1-tiered	constrained	partial
CP [21][22]	2	$R = r$	1-tiered	constrained	partial
<u>Two-tiered</u>					
2CRNDC [23]	2	$R \geq 2r$	2-tiered	constrained	partial
CRNSC and 2CRNDC [24]	1, 2	$R \geq 4r$	2-tiered	unconstrained	partial
MRP-1 and MRP-2 [25]	1, 2	$R = r$, $R \neq r$	2-tiered	unconstrained	full
k-Vertex Connectivity [26]	≥ 2	$R = r$	2-tiered	un/constrained	partial

REFERENCES

[1] A. Srinivas and E. Modiano. Minimum Energy Disjoint Path Routing in Wireless Ad-hoc Networks. In *Proc. 9th Ann. Int'l Conf. Mobile Computing and Networking (MobiCom'03)*, pages 122-133, Sept. 2003.

[2] D. Torrieri. Algorithms for Finding an Optimal Set of Short Disjoint Paths in a Communication Network. *IEEE Transactions on Communications*, Volume 40, Number 11, pages 1698-1702, Nov. 1992.

[3] S. Misra, S.D. Hong, G. Xue and J. Tang. Constrained Relay Node Placement in Wireless Sensor Networks to Meet Connectivity and Survivability Requirements. In *Proc. 27th Ann. IEEE Conf. Computer Communications (INFOCOM'08)*, pages 281-285, Apr. 2008.

[4] J. Pu, Z. Xiong and X. Lu. Fault-Tolerant Deployment with *k*-connectivity and Partial *k*-connectivity in Sensor Networks. *Wireless Communications and Mobile Computing*, Volume 9, Number 7, pages 909-919, May 2008.

[5] X. Han, X. Cao, E.L. Lloyd and C.C. Shen. Fault-tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, Volume 9, Number 5, pages 643-656, May 2010.

[6] J.L. Bredin, E.D. Demaine, M. Hajiaghayi and D. Rus. Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance. In *Proc. 6th ACM Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc'05)*, pages 309-319, May 2005.

[7] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.

[8] R. Ravi and D.P. Williamson. An Approximation Algorithm for Minimum-Cost Vertex-Connectivity Problems. *Algorithmica*, Volume 18, Number 1, pages 21-43, 1997.

[9] M. Ahlberg, V. Vlassov and T. Yasui. Router Placement in Wireless Sensor Network. Technical Report KTH/ICT/ECS, Royal Institute of Technology (KTH), Stockholm, Sweden, 2006.

[10] W. Zhang, G. Xue and S. Misra. Fault-Tolerant Relay Node Placement in Wireless Sensor Networks: Problems and Algorithms. In *Proc. 26th Ann. IEEE Conf. Computer Communications (INFOCOM'07)*, pages 1649-1657, May 2007.

[11] E.L. Lloyd and G. Xue. Relay Node Placement in Wireless Sensor Networks. *IEEE Transactions on Computers*, Volume 56, Number 1, pages 134-138, Jan. 2007.

[12] L. Sitanayah, K.N. Brown, and C.J. Sreenan. Fault-Tolerant Relay Deployment for *k* Node-Disjoint Paths in Wireless Sensor Networks. In *Proc. 4th Int'l Conf. IFIP Wireless Days (WD'11)*, pages 1-6, Oct. 2011.

[13] L. Sitanayah, K.N. Brown, and C.J. Sreenan. A Fault-Tolerant Relay Placement Algorithm for Ensuring *k* Vertex-Disjoint Shortest Paths in Wireless Sensor Networks. *Ad Hoc Networks*, Volume 23, pages 145-162, Dec. 2014.

[14] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton, University Press, 1962.

[15] L.C. Freeman. Centrality in Social Networks Conceptual Clari_cation. *Social Networks*, Volume 1, Number 3, pages 215-239, 1979.

[16] U. Brandes. On Variants of Shortest-Path Betweenness Centrality and Their Generic Computation. *Social Networks*, Volume 30, Number 2, pages 136-145, May 2008.

[17] L. Sitanayah, K.N. Brown, and C.J. Sreenan. Fault-Tolerant Relay Deployment Based on Length-Constrained Connectivity and Rerouting Centrality in Wireless Sensor Networks. In *Proc. 9th European Conference on Wireless Sensor Networks (EWSN'12)*, Volume 7158 LNCS, pages 115-130, Feb. 2012.

[18] L. Sitanayah. Robust Sensor Network Deployment with Priority Based on Failure Centrality. In *Proc. 10th Int'l. Conf. Information Technology and Electrical Engineering (ICITEE'18),* pages 32-37, Jul. 2018.

[19] L. Sitanayah, K.N. Brown, and C.J. Sreenan. Multiple Sink and Relay Placement in Wireless Sensor Networks. In *Proc. 1st Workshop Artificial Intelligence for Telecommunications and Sensor Networks (WAITS'12), 20th European Conf. Artificial Intelligence (ECAI'12)*, pages 18-23, Aug. 2012.

[20] L. Sitanayah, K.N. Brown, and C.J. Sreenan. Planning the Deployment of Multiple Sinks and Relays in Wireless Sensor Networks. *Journal of Heuristics, special issue on Heuristics for Reliable and Efficient Wireless Sensor Networks Deployments*, Volume 21, Number 2, pages 197-232, Apr. 2015.

[21] L. Quesada, K.N. Brown, B. O'Sullivan, L. Sitanayah, and C.J. Sreenan. A Constraint Programming Approach to the Additional Relay Placement Problem in Wireless Sensor Networks. In *Proc. IEEE Int'l Conf. Tools with Artificial Intelligence (ICTAI'13)*, pages 1052-1059, Nov. 2013.

[22] L. Quesada, L. Sitanayah, K.N. Brown, B. O'Sullivan, and C.J. Sreenan. A Constraint Programming Approach to the Additional Relay Placement Problem in Wireless Sensor Networks. *Constraints*, Volume 20, Number 4, pages 433-451, Oct. 2015.

[23] B. Hao, J. Tang and G. Xue. Fault-Tolerant Relay Node Placement in Wireless Sensor Networks: Formulation and Approximation. In *Proc. Workshop High Performance Switching and Routing (HPSR'04)*, pages 246-250, Apr. 2004.

[24] J. Tang, B. Hao and A. Sen. Relay Node Placement in Large Scale Wireless Sensor Networks. *Computer Communications*, Volume 29, Number 4, pages 490-501, Feb. 2006.

[25] H. Liu, P. Wan and X. Jia. On Optimal Placement of Relay Nodes for Reliable Connectivity in Wireless Sensor Networks. *Combinatorial Optimization*, Volume 11, Number 2, pages 249-260, Mar. 2006.

[26] A. Kashyap, S. Khuller and M. Shayman. Relay Placement for Fault Tolerance in Wireless Networks in Higher Dimensions. *Computational Geometry: Theory and Applications*, Volume 44, Number 4, pages 206-215, May 2011.