HAL
open science

Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach

Carlos A Oroza, Jairo A Giraldo, Masood Parvania, Thomas Watteyne

To cite this version:

Carlos A Oroza, Jairo A Giraldo, Masood Parvania, Thomas Watteyne. Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach. IEEE Internet of Things Journal, 2021, 8 (24), pp.17429 - 17435. 10.1109/jiot.2021.3082168 . hal-03538244

HAL Id: hal-03538244

https://inria.hal.science/hal-03538244

Submitted on 20 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach

Carlos A. Oroza , *Member, IEEE*, Jairo A. Giraldo , *Member, IEEE*,
Masood Parvania , *Senior Member, IEEE*, and Thomas Watteyne , *Senior Member, IEEE*

***Abstract*—Existing methods for wireless-sensor network (WSN) topology optimization employ simplifying assumptions of a fixed communication radius between network nodes, which is ill-suited for IoT networks deployed in complex terrain. This article proposes a data-driven approach to WSN topology optimization, employing a Bayesian link classifier trained on LIDAR-derived terrain characteristics and an *in-situ* survey of link quality. The classifier is trained to predict where good network links (packet-delivery ratio, PDR>0.5) are likely to form in a region given complex terrain attributes. Then, given numerous candidate wireless node placements throughout the domain, the classifier is used to construct an undirected weighted graph of the potential connectivity across the domain. Edge weights in the connectivity graph are proportional to the probability of forming a good link between the nodes. A novel modified cycle-union (MCyU) algorithm for generating a 2-vertex-connected, Steiner minimal network is then applied to the undirected weighted graph of potential network element placements. This ensures a survivable network design, while maximizing the probability of good links within the final network. The total number and spatial distribution of network elements produced by the algorithm is compared to an existing WSN, deployed for environmental monitoring in remote regions. In addition, the MCyU algorithm has been evaluated in three graph test cases to compare with state-of-the-art solutions, where MCyU outperforms in terms of weight minimization and computation time.**

***Index Terms*—Bayesian approach, experimental validation, machine learning, topology optimization, wireless sensor networks (WSNs).**

I. Introduction

ADVANCES in low-power wireless communication are enabling large-scale deployments of wireless sensor networks (WSNs) in increasingly complex and remote regions [1]–[10]. An example of such deployments is large-scale wireless mesh networks designed to monitor water resources from snow in mountain regions [11]–[13]. The remote nature of these deployments prevents network maintenance for much of the year, necessitating a mesh network topology that is robust to node failure. Moreover, it is necessary to estimate the required number of network elements prior to field deployment as accessing remote regions repeatedly is cost-prohibitive.

Existing methods for optimizing mesh network topology typically rely on assumptions of a fixed isometric transmission range among network elements [14], [15]. This assumption is violated for wireless deployments in complex terrain, as terrain and vegetation characteristics result in significant variability in the path length of the network links between each node.

Consequently, such networks must be structured using a heuristic process in which field teams iteratively deploy and adjust network elements, aiming to connect the sensor nodes to a base station with a satellite or cellular uplink [13]. This process is labor-intensive and can produce wireless mesh topologies in which a single-node failure can result in a disconnected graph, resulting in loss of real-time data [13]. An improved methodology is needed which does not rely on simplified assumptions of node connectivity, in order to reliably construct survivable WSNs in complex terrain.

The design of survivable networks has gained attention in the last two decades due to the potential consequences of communication disruptions from natural or man-made causes [16]. Survivable networks are constructed by selecting links and nodes that minimize a cost function given certain requirements in terms of the subset of nodes that need to be connected and the maximum number of simultaneous node or link failures that the network can withstand [17]. In particular, when the objective is to design low-cost networks that survive a single-node failure, the problem of constructing the minimum 2-vertex connected Steiner (2VCS) network arises. In general, finding the minimum 2VCS networks is an MAXNP-hard combinatorial problem. Few algorithms that find approximations of the optimal solution have been developed [18]–[20]. Coullard *et al.* [18] proposed a linear-time algorithm for two special types of graphs: 1) the W4-free graph and 2) the Halin graph. Chimani *et al.* [19] transformed the 2VCS problem into a related one for directed graphs that share the same solution and proposed an integer linear programming (ILP) approach. Shen and Guo [20] also used a directed-graph transformation and derived a greedy approach to add disjoint paths that iteratively improve the solution.

Although the above approaches offer theoretical computation guarantees, implementing several of the subroutines in real applications with a large number of nodes (>5000) becomes computationally intractable.

Carlos A. Oroza is with the Civil and Environmental Engineering Department, University of Utah, Salt Lake City, UT 84112 USA (e-mail: carlos.oroza@utah.edu).

Jairo A. Giraldo and Masood Parvania are with the Electrical and Computer Engineering Department, University of Utah, Salt Lake City, UT 84112 USA.

Thomas Watteyne is with the EVA Team, Inria, 75012 Paris, France.
Digital Object Identifier 10.1109/JIOT.2021.3082168

This article proposes and evaluates an efficient, structured method for optimizing wireless mesh networks in complex terrain, which does not rely on fixed path-length assumptions in order to optimize the network topology. Instead, the proposed method leverages LIDAR remote-sensing data of terrain characteristics and employs a Bayesian link-classification approach, in which a link classifier is trained on *in-situ* observations of link quality and remote sensing of terrain characteristics. The classifier is used to evaluate the potential connectivity across the region in which the wireless mesh network will be deployed.

In order to efficiently compute the minimal survivable network from the potential connectivity graph, we develop a modified cycle-union (MCyU) algorithm which expands the ear-decomposition methodology [21] (also known as a simple cycle-union (SCU) algorithm) and offers low implementation complexity when compared to previous approaches. The algorithm finds a 2VCS from the connectivity graph while prioritizing the expected link quality [i.e., packet-delivery ratio (PDR)]. The novelty lies in penalizing the links between wireless nodes such that, at each iteration, they are added only if necessary and then employing a simple edge removal procedure that removes unnecessary connections. If all Steiner nodes are evaluated but some of them have not been added, then the threshold will increase until all Steiner nodes are part of the network. Finally, a simple routine removes possible redundant links.

The proposed method is applied to an existing network [13], and the number and distribution of nodes is compared to those determined manually. The proposed method consists of four steps overall:

1) LIDAR and *in-situ* PDR data collection;
2) Bayesian link classifier training;
3) connectivity graph construction;
4) 2-vertex Steiner minimal graph construction.

The remainder of this article is organized as follows. Section II describes the LIDAR and *in-situ* PDR data collection methods. Section III introduces the Bayesian link classification algorithm. Section IV explains the connectivity graph construction. Section V details the approximation algorithm used to generate a 2-vertex-connected graph. Section VI compares the number and spatial distribution of network elements against an existing network. Finally, Section VII summarizes the present study and discusses limitations and opportunities for future research.

II. Methods

A. Study Location

This study focuses on an existing WSN designed to monitor remote water resources. It is located in the Southern Sierra Nevada, California, and is part of the NSF Critical Zone Observatory [13]. First deployed in September 2009, the network consists of 23 sensor nodes, 34 repeater nodes, and a base station containing a cellular data uplink.

Each sensor node consists of a suite of sensors that are designed to provide representative measurements of key horologic variables across the landscape:

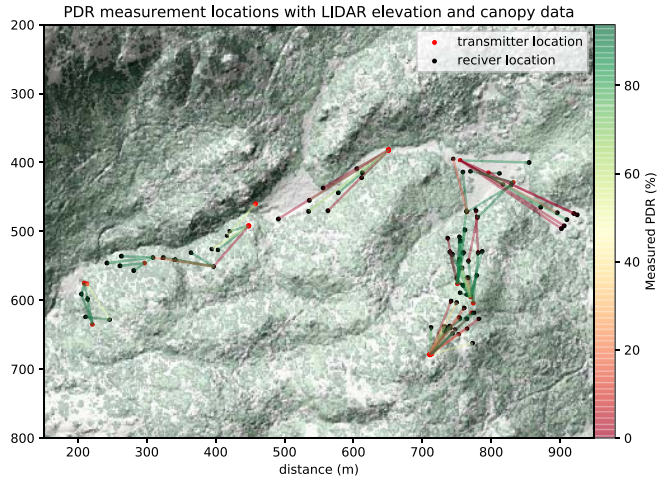

Fig. 1. *In-situ* PDR measurements overlain on the 1-m^2 LIDAR-derived terrain and canopy data used to train the Bayesian link classifier described in Section III.

1) snow depth (ultrasonic distance sensor);
2) temperature;
3) relative humidity;
4) soil moisture;
5) soil matric potential;
6) radiation.

The locations of the sensor nodes are determined using expert field surveys (designed to capture spatial heterogeneity of the above variables), and are therefore at fixed locations with the domain. The base station location is also fixed, constrained to where there is high signal strength for cellular Internet connectivity. Real-time data from the sensor stations are transmitted from the sensor stations to the base stations through an IEEE 802.15.4e low-power mesh network. For a more detailed description of the existing network (see [13]).

With the sensor stations and base station in fixed locations, the objective of the topology optimization algorithm is to determine both the required number and optimal locations for signal repeaters such that the resulting wireless mesh topology is robust to the failure of a single node, and that the expected link quality of the final mesh network is maximized.

B. LIDAR Data

LIDAR terrain and vegetation data (Fig. 1) were collected for the region around the existing WSN using a Gemini 06SEN/CON195 and 08DIG017 digitizer system installed on a PA-31 aircraft (see [22] for additional details). These data were collected at a point density of 11.65 pts/m^2, and used to make 1-m^2 digital elevation and canopy models. The LIDAR survey was conducted from August 5, 2010 to August 15, 2010. Point-cloud data were processed and classified by the National Center for Airborne Laser Mapping (NCALM), which is hosted on the NSF OpenTopography server.[1]

The study area is clipped from the point-cloud data set for calculating the above 1-m vegetation density. The point clouds are first grouped as ground returns or vegetation

[1] https://www.opentopography.org/

returns. Second, the elevation difference between each point is calculated by iterating through all vegetation data points and measuring the height to the nearest neighbor among all the ground returns. Vegetation data are filtered such that only the points above 1 m are retained (creating a binary feature that indicates the presence of vegetation in a grid cell, ignoring low-lying brush).

C. Packet-Delivery Ratio Data

PDR measurements (Fig. 1) are manually collected at points near the existing network. Antennas (L-COM HGV-2404U) with Metronome Systems repeater modules are placed on lightweight poles that are measured to be the same height as the poles in the existing network (3 m). The data are collected on four channels (2.405, 2.425, 2.445, and 2.465 GHz) using the `radiotest` module for Analog Devices SmartMesh IP products.[2]

The transmitter mote is set to continuously send a 40-B frame every 5 ms. The transmitter and receiver are placed in strategic locations throughout the observatory, with the intention of sampling across a representative range of path distances, vegetation characteristics, and angles. The maximum number of packets received is 973, which is taken as 100% PDR. Each transmitter and receiver location are measured with a handheld GPS unit. Fig. 1 shows the transmitter and receiver locations (red and black points, respectively), along with the average PDR along each path (line color).

Line of sight between the two antennas results in high PDR on all channels, even for long links (>100 m). For eaxmple, in Fig. 1, the link on the far right receives [639, 876, 929, 881] packets on each channel for the 2.405, 2.425, 2.445, and 2.465-GHz channels, respectively. Removing line-of-sight conditions at large distances strongly affected PDR: moving the receiver by 5 m (thereby introducing more vegetation along the path, see center right of Fig. 1) causes the packets received to go to zero on all channels. PDR is found to be variable when there is significant vegetation along the path: while long links are occasionally observed, high PDR measurements are rarely seen for links longer than 50 m when dense vegetation is present on the path.

III. Bayesian Link Classifier

A Bayesian link classifier is trained using the *in-situ* PDR measurements and LIDAR data. The LIDAR data are used to compute path properties of each measured point based on the GPS locations of the transmitter and receiver (see Table I). Vegetation, distance, and terrain intersection are used as the primary path properties affecting the path quality.

First, the 3-D Euclidean distance between the two points is computed (i.e., the line-of-sight distance between the two points based on the coordinates and digital elevation model). Then, Bresenham's line algorithm [23] is used to compute all the grid cells between the transmitter and the receiver. The amount of canopy is integrated along the path to calculate the total amount of vegetation on the path. Terrain intersection is computed by comparing the straight line of sight of the transmitter and receiver against the DEM profile between them. If the two intersected within a 0.5-m tolerance, the link was set to zero PDR. For computational efficiency, only links shorter than 200 m are considered.

The PDR classifier is created using a Gaussian Naïve Bayes classifier. Given an input feature space x (the LIDAR-derived path features), the Gaussian Naïve Bayes classification rule is given in

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y). \tag{1}$$

The model is implemented in Scikit Learn version 0.22 using the `GaussianNB` module.

IV. Connectivity Graph Construction

The Bayesian link classifier is then used together with the LIDAR data to construct a connectivity graph in the region around the existing WSN (Fig. 2). Sensor nodes and the base-station uplink—whose locations are predetermined based on hydrological considerations—are shown in green and black, respectively.

TABLE I
Subset of LIDAR-Derived Path Data Used to Train the Bayesian Link Classifier

Path Distance (m)	Path Vegetation	Good Link
84.356033	31.0	False
84.495324	38.0	False
88.111382	16.0	False
5.993607	1.0	True
45.090517	8.0	True
29.298013	16.0	False
69.004588	22.0	True
57.596499	10.0	True
47.540821	18.0	True
68.725011	23.0	True
79.302963	28.0	True
74.946648	10.0	True
79.568585	22.0	False
55.867959	8.0	True
95.800791	24.0	True

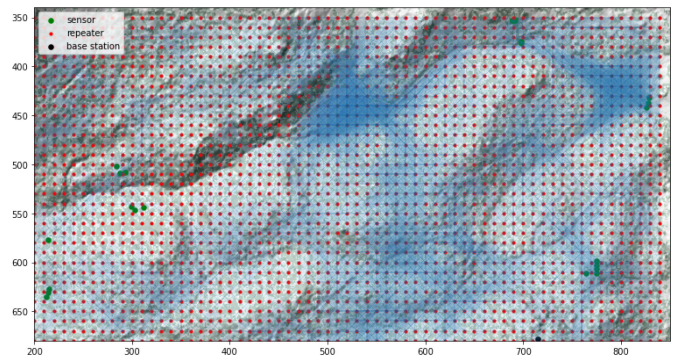

Fig. 2. Connectivity graph overlain on 1-m^2 LIDAR terrain and vegetation data. Sensor and base station (fixed) are shown as green and black dots, respectively. Potential repeater placements shown as red dots. Good network links shown as blue lines. Note high connectivity in flat, unvegetated regions (e.g., upper right); low connectivity in vegetated and/or regions with complex terrain (e.g., lower left).

[2]https://www.analog.com/en/applications/technology/smartmesh-pavilion-home.html

Potential signal repeater locations are then placed at even spatial intervals throughout the domain and links between each element are computed based on the Bayesian link classifier. To minimize the computational burden in the 2-vertex connectivity algorithm, we consider only links with a high probability of forming a good link (greater than 90%). Intuitively, there are dense, long-range links between elements with an unobstructed line of sight (e.g., upper right region of Fig. 2). Links are more sparse where terrain features obstruct line of sight between nodes (e.g., upper left region).

For clarity, Fig. 2 shows the results of this process for the 10-m interval for the potential repeater placements, whereas the interval used in subsequent steps is 5 m. The connectivity graph is undirected and edge weights are proportional to the probability of forming a good link between each element.

V. 2-Vertex Connectivity Algorithm

After obtaining the weighted graph with all possible interconnections between transmitters and receivers, it is necessary to find the optimal location of repeaters that guarantee specific properties of the network. This problem corresponds to the minimum 2VCS graph, which consists of finding the subgraph 2VSN of minimum cost (i.e., minimum weight) that guarantees that a group of nodes remain connected even when one node is removed. In addition, we also want to enforce that each node has at least three neighbors. This type of network offers several advantages for our IoT scenario: 1) we can find the minimum number of repeaters; 2) the network is able to *survive* the removal of any node (one at a time); and 3) we can guarantee that the links with higher PDR are connected.

Let $\mathcal{G}(V, \mathcal{E}, W)$ denote an undirected graph with vertex set V, edges $\mathcal{E}$, and weight matrix W. Let $S \subseteq V$ be the set of terminals, which corresponds to the minimum set of nodes that are part of the 2-vertex Steiner Network 2VSN. Let $\mathcal{P}_{s,u} = \{P^1_{s,u}, P^2_{s,u}, \ldots\}$ denote the set of all possible paths from s to u, and let $f(P^i_{s,u})$ denote the aggregated weight of path $P^i_{s,u}$. We say that path pair $P^i_{s,u}$ and $P^j_{s,u}$ are vertex disjoint if they do not have any common vertex, except by s and u. Similarly, we say that the path pair $P^i_{s,u}$ and $P^j_{s,v}$ are vertex disjoint if they only share the vertex s.

Following the notation in [24], for vertex s, we denote by $\text{VSPP}_{\mathcal{G}}(s, u)$ the shortest vertex-disjoint path pair in graph $\mathcal{G}$ from s to u, and by $\text{VSPP}_{\mathcal{G}}(s; u, v)$ the shortest vertex-disjoint path pair from s to u and from s to v. The term "shortest" refers to the pair of paths with minimum aggregated weight, i.e.,

$$\text{VSPP}_{\mathcal{G}}(s, u) = \min_{P^i_{s,u}, P^j_{s,u} \in \mathcal{P}_{s,u}} \left\{ f\left(P^i_{s,u}\right) + f\left(P^j_{s,u}\right) \right\}.$$

Similarly, we have that

$$\text{VSPP}_{\mathcal{G}}(s; u, v) = \min_{P^i_{s,u} \in \mathcal{P}_{s,u}, P^j_{s,v} \in \mathcal{P}_{s,v}} \left\{ f\left(P^i_{s,u}\right) + f\left(P^j_{s,v}\right) \right\}.$$

Our proposed algorithm to compute 2VSN is based on the "cycle-union algorithm," which consists of repeatedly adding a cycle containing at least two terminals until all terminals are at least 2-vertex connected. In the first iteration, we compute the function $\text{VSPP}_{\mathcal{G}}(s, u)$ between the pair of terminals $s, u \in S$ that results in the minimum weight. Then, the following iterations consist of connecting a terminal s outside of 2VSN with two distinct vertices of 2VSN, u, v, where u, v corresponds to the vertices that result in the minimum weight.

Algorithm 1: MCyU Algorithm

Input: Weighted graph $\mathcal{G}(V, \mathcal{E}, W)$, terminal set $S \subseteq V$, non-terminal extra weight ω, path threshold T, and path threshold increment ΔT.

Output: 2-Vertex connected Steiner network 2VSN

```
Add extra weight ω to all edges connected to
non-terminal nodes in G
P_0 ← min_{s,v}{VSPP_G(s, v) | s, v ∈ S, s ≠ v}
2VSN ← P_0
i=0
Remove extra weights in G associated to P_0
while not all terminals are in 2VSN do
    N = set of nodes in 2VSN
    i+ = 1
    if i ≥ |S| then
        i = 0, T = T + ΔT
    if s_i not in N then
        P_i ← min_{u,v}{VSPP_G(s_i; u, v) | u, v ∈ N, u ≠ v}
        if f(P_i) ≤ T then
            2VSN ← P_i
            Remove extra weights in G associated to P_i
N = set of nodes in 2VSN
for all x ∈ N do
    N_x = ordered set of neighbors of x
    for all y ∈ N_x do
        2VSN_Temp = 2VSN with no edge x to y
        if VSPP_{2VSN_Temp}(x, y) ≠ ∅ then
            Remove edge x, y from 2VSN
```

In order to guarantee that the algorithm focuses on creating paths that contain as less nonterminal nodes as possible, we add several modifications. First, we add an extra weight $\omega \geq 0$ to all edges that connect at least one nonterminal node. As a consequence, the algorithm prioritizes existing nodes and terminal nodes instead of adding many nonterminal nodes. Second, we add a threshold T that filters *heavy* paths, i.e., it only allows paths that satisfy $f(P_i) \leq T$. Therefore, if the minimum path pair associated with s_i has a weight greater than T, the path is not added and s_i is not connected in the current iteration. When all terminal nodes are evaluated, the iterations restart, the threshold T increases, and the terminal nodes that were not added before are evaluated again. This allows to find disjoint paths using the set of nodes and edges added in previous iterations, preventing the addition of nonnecessary nodes. The resulting graph 2VSN has a vertex and edge sets $\mathcal{V}_{\text{2VSN}}$ and $\mathcal{E}_{\text{2VSN}}$, respectively. Once a 2-vertex Steiner connected graph 2VSN is found, nonnecessary edges and vertices are removed by evaluating for each edge if its removal does not affect the 2-vertex connectivity. The MCyU algorithm is summarized in Algorithm 1. Finally, to guarantee that each node has at least three neighbors (different to 3-edge connectivity),

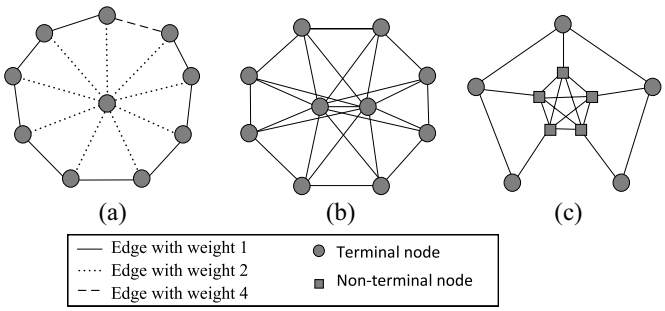

Fig. 3. Test graphs introduced in [20]. (a) Test graph with a single central node. (b) Test graph with two central nodes. (c) Test graph with non-terminal nodes.

TABLE II
COMPARISON OF THE TOTAL WEIGHT OBTAINED BY DIFFERENT 2-VERTEX STEINER CONNECTED ALGORITHMS

	Total Weight f_T			
	SCU	**Alg. in [20]**	**MCyU**	**Optimal**
Graph (a)	23	12	12	12
Graph (b)	15	15	11	11
Graph (c)	13	9	9	8

new paths from the original graph are added, giving priority to those paths formed by the nodes already existing in 2VSN.

The benefits of Algorithm 1 lie in the simplicity of its formulation and in the fact that, by adding weights to those edges connected to nonterminal nodes, it is possible to increase the efficiency of the computation of the pair u, v that minimizes $\text{VSPP}_{\mathcal{G}}(s_i; u, v)$. It is worth noting that selecting a small threshold T may incur additional computation time, but leads to a smaller set of nonterminal nodes in 2VSN. Therefore, there is a tradeoff between the computation time caused by T and the number of nodes of 2VSN.

In order to illustrate the viability of the MCyU algorithm, we compare its performance with the SCU algorithm, with the efficient algorithm (EFA) presented in [20], and with the optimal solution. Three different test graphs depicted in Fig. 3 are utilized. Recall that $\mathcal{E}_{VSN} \subseteq \mathcal{E}$ denotes the set of edges of the resulting graph for each algorithm, and for each edge, there is an associated weight W_e. Then, $f_T = \sum_{e \in \mathcal{E}_{VSN}} W_e$ denotes the total weight of the 2-vertex Steiner connected graph. The total weight of the resulting graphs is summarized in Table II. Note that the graph obtained with the SCU algorithm always contains a significant amount of unnecessary nodes. However, our proposed MCyU algorithm in all three cases leads to a lower number of edges, very close to the optimal solution. Particularly for Graph (b), our solution outperformed EFA by having 36% less nodes (see Fig. 4). On the other hand, the average runtime in an Intel Core i7 computer of the SCU algorithm for the three graphs is 1.25 s, whereas for the MCyU algorithm is 0.15 s, showing that the proposed MCyU is about 8× faster than the SCU. Clearly, MCyU not only leads to a lower number of nodes and edges but also to low computational time.

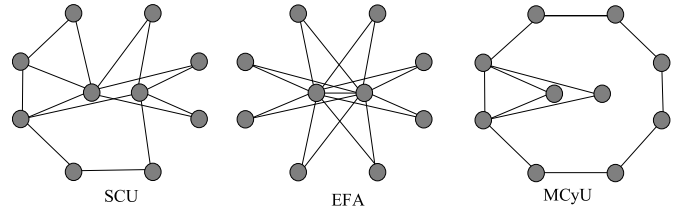

Fig. 4. Solution of different 2VCS algorithms for Graph (b).

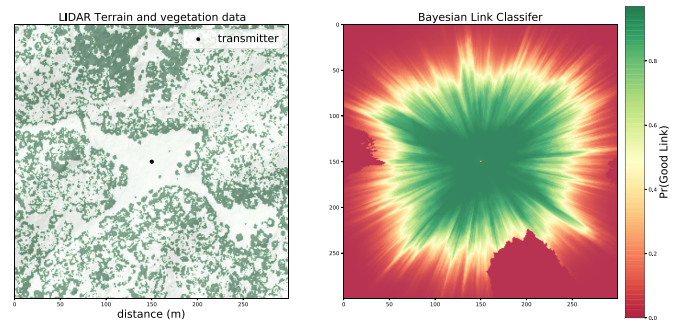

Fig. 5. Left: Transmitter location and LIDAR data. Right: Predicted link quality from the machine-learning algorithm at every 1-m^2 grid cell.

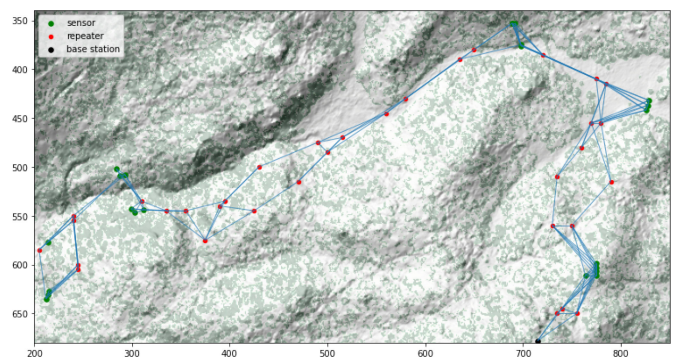

Fig. 6. Output of the proposed algorithm: sensor nodes (green points) and base station (black point) are fixed. The algorithm determines the signal repeater placements (red points) that ensure a minimal 2-vertex connected mesh network.

VI. STUDY RESULTS

A. Link Classifier

Fig. 5 shows the probability output of the trained `GaussianNB` classifier for every 1-m^2 grid cell in the domain, given a transmitter (black dot) at the center of the image. Good links (dark green regions) can exist at long distances when the canopy does not intersect the path. If the significant canopy intersects the path, the PDR is not likely to be high at links greater than approximately 50–75 m (light-green to red regions), where the path intersects the terrain, the good link probability is set to zero (dark-red regions). The accuracy of the Bayesian link classifier is determined to be 74% using tenfold cross-validation within the 81 PDR measurements (Fig. 1).

B. Network Topology

Fig. 6 shows the output of the proposed algorithm for the WSN in the study region. The algorithm identified

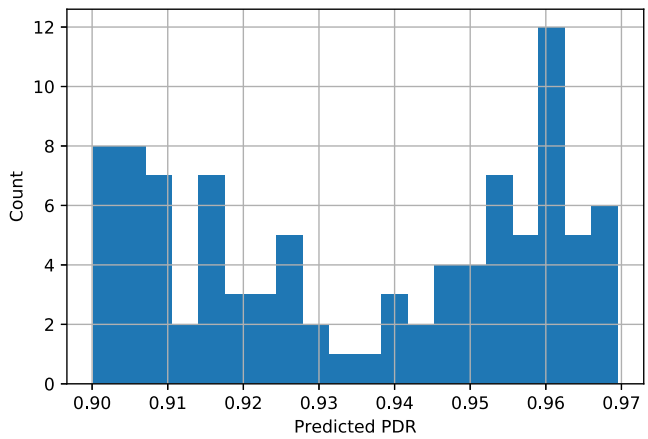

Fig. 7. Distribution of edge weights in the final network (i.e., probability of forming a good link between each node).

34 repeater locations (red points), which generate a 2-vertex connected graph of the sensor nodes (green points) to the base station (black point). Edges are shown as blue lines.

Note that the algorithm prioritizes line-of-sight connections (upper right and center regions of Fig. 6) to minimize the number of repeaters in the final network. An alternative topology could be constructed by building the network across the lower boundary of the region to the farthest sensor nodes. Although the distance between the base station and nodes is shorter along this path, the region has denser vegetation and fewer line-of-sight pathways, resulting in lower predicted PDR and shorter possible links.

The number of required repeater elements, as well as the overall spatial structure of the network, are similar to the same network structured manually (see [13, Fig. 9]). The distribution of edge weights in the final network is shown in Fig. 7.

VII. Conclusion

This study develops and assesses a method for optimizing WSN topology in complex terrain based on LIDAR data, limited *in-situ* observations of link quality, a Bayesian classification algorithm, and a 2-vertex connectivity algorithm, MCyU. The proposed MCyU algorithm can potentially outperform state-of-the-art solutions in terms of weight minimization and computation time, as it was demonstrated with three different test graphs.

Our research indicates that remote IoT networks may be able to be designed in an automated manner based on remote sensing and data-driven predictions of link quality, rather than heuristically in the field. Future work could consider: 1) placing repeaters in the proposed locations and evaluating the long-term topological evolution of the network; 2) improving the Bayesian link prediction algorithm with more extensive observations of link quality, particularly ones that capture temporal variability; and 3) employing more sophisticated machine-learning algorithms (e.g., tree-based classification models, such as random forest or neural networks) to improve the accuracy of the link classifier (though this would likely require more data and increase the computational burden).

Acknowledgment

The authors wish to thank Erin Stacy, Zeshi Zheng, Branko Kerkez, Matt Meadows, Keoma Brun, and the entire team at the NSF Southern Sierra Critical Zone Observatory.

References

[1] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in *Proc. IEEE Military Commun. Conf.*, 2009, pp. 1–7.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[3] C. O. Mathuna, T. O'Donnell, R. V. Martinez-Catala, J. Rohan, and B. O'Flynn, "Energy scavenging for long-term deployable wireless sensor networks," *Talanta*, vol. 75, no. 3, pp. 613–623, 2008.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. 1st ACM Int. Workshop Wireless Sens. Netw. Appl.*, 2002, pp. 88–97.

[5] T. Harms, S. Sedigh, and F. Bastianini, "Structural health monitoring of bridges using wireless sensor networks," *IEEE Instrum. Meas. Mag.*, vol. 13, no. 6, pp. 14–18, Dec. 2010.

[6] P. Jiang, H. Xia, Z. He, and Z. Wang, "Design of a water environment monitoring system based on wireless sensor networks," *Sensors*, vol. 9, no. 8, pp. 6411–6434, 2009.

[7] V. Henriques and R. Malekian, "Mine safety system using wireless sensor network," *IEEE Access*, vol. 4, pp. 3511–3521, 2016.

[8] D. K. Yadav, G. Karthik, S. Jayanthu, and S. K. Das, "Design of real-time slope monitoring system using time-domain reflectometry with wireless sensor network," *IEEE Sens. Lett.*, vol. 3, no. 2, pp. 1–4, Feb. 2019.

[9] M. V. Ramesh, "Real-time wireless sensor network for landslide detection," in *Proc. 3rd Int. Conf. Sens. Technol. Appl.*, 2009, pp. 405–409.

[10] M. J. Whelan, M. V. Gangone, and K. D. Janoyan, "Highway bridge assessment using an adaptive real-time wireless sensor network," *IEEE Sensors J.*, vol. 9, no. 11, pp. 1405–1413, Nov. 2009.

[11] Z. Zhang, S. Glaser, T. Watteyne, and S. Malek, "Long-term monitoring of the sierra nevada snowpack using wireless sensor networks," *IEEE Internet Things J.*, early access, Jan. 31, 2020, doi: 10.1109/JIOT.2020.2970596.

[12] S. Malek *et al.*, "Real-time alpine measurement system using wireless sensor networks," *Sensors*, vol. 17, no. 11, p. 2583, 2017.

[13] B. Kerkez, S. D. Glaser, R. C. Bales, and M. W. Meadows, "Design and performance of a wireless sensor network for catchment-scale snow and soil moisture measurements," *Water Resour. Res.*, vol. 48, no. 9, 2012, Art. no. W09515.

[14] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless mesh networks design—A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 299–310, 2nd Quart., 2012.

[15] S. Misra, S. D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks: Formulation and approximations," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 434–447, Apr. 2010.

[16] M. Grötschel, C. L. Monma, and M. Stoer, "Design of survivable networks," in *Handbooks in Operations Research and Management Science*, vol. 7. Amsterdam, The Netherlands: North-Holland, 1995, pp. 617–672.

[17] H. Kerivin and A. R. Mahjoub, "Design of survivable networks: A survey," *Netw. Int. J.*, vol. 46, no. 1, pp. 1–21, 2005.

[18] C. R. Coullard, A. Rais, R. L. Rardin, and D. K. Wagner, "Linear-time algorithms for the 2-connected steiner subgraph problem on special classes of graphs," *Networks*, vol. 23, no. 3, pp. 195–206, 1993.

[19] M. Chimani, M. Kandyba, I. Ljubić, and P. Mutzel, "Strong formulations for 2-node-connected steiner network problems," in *Proc. Int. Conf. Comb. Optim. Appl.*, 2008, pp. 190–200.

[20] H. Shen and L. Guo, "Efficient 2-approximation algorithms for computing 2-connected steiner minimal networks," *IEEE Trans. Comput.*, vol. 61, no. 7, pp. 954–968, Jul. 2012.

[21] K. Bernhard and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 3rd ed. Heidelberg, Germany: Springer-Verlag, 2006.
[22] G. Qinghua and R. Bales, *Snow-On and Snow-Off Lidar Point Cloud Data and Digital Elevation Models for Study of Topography, Snow, Ecosystems and Environmental Change at Southern Sierra Critical Zone Observatory, California: Southern Sierra CZO*, Univ. California Merced, Merced, CA, USA, 2012.
[23] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
[24] R. Bhandari, "Optimal physical diversity algorithms and survivable networks," in *Proc. 2nd IEEE Symp. Comput. Commun.*, 1997, pp. 433–441.

Carlos A. Oroza (Member, IEEE) received the M.S. and Ph.D. degrees in civil systems engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2012 and 2017, respectively.

He is currently an Assistant Professor of Civil and Environmental Engineering with the University of Utah, Salt Lake City, UT, USA.

Jairo A. Giraldo (Member, IEEE) received the B.S. degree in electronic engineering from the National University of Colombia, Manizales, Colombia, in 2010, and the M.S. and Ph.D. degrees from the Universidad de los Andes, Bogotá, Colombia, in 2012 and 2015, respectively.

He is currently a Research Assistant Professor with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT, USA. His research interests include security and privacy in cyber–physical systems, multiagent systems, and distributed control in the smart grid.

Masood Parvania (Senior Member, IEEE) is the Director of Utah Smart Energy Laboratory and an Assistant Professor of Electrical and Computer Engineering with the University of Utah, Salt Lake City, UT, USA. His research interests include the operation, economics, and resilience of power and energy systems, and modeling and operation of interdependent critical infrastructures.

Dr. Parvania serves as an Associate Editor for the IEEE TRANSACTIONS ON SMART GRID, the IEEE POWER ENGINEERING LETTERS, and the IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY. He is the Chair of IEEE Power and Energy Society (PES) Utah Chapter, the Chair of the IEEE PES Bulk Power Systems Operation Subcommittee, and the Vice-Chair of IEEE PES Risk, Reliability, and Probability Applications Subcommittee.

Thomas Watteyne (Senior Member, IEEE) received the M.Sc. degree in networking, the M.Eng. degree in telecommunications, and the Ph.D. degree in computer science from INSA Lyon, Villeurbanne, France, in 2005, 2005, and 2008, respectively.

He is an Insatiable Enthusiast of low-power wireless mesh technologies. He is a Researcher with Inria, Paris, France, in the new EVA Research Team, where he designs, models, and builds networking solutions based on a variety of Internet of Things (IoT) standards. He is a Senior Networking Design Engineer of Linear Technology with the Dust Networks Product Group, the Undisputed Leader in supplying low-power wireless mesh networks for demanding industrial process automation applications. Since 2013, he has been Co-Chair of the IETF 6TiSCH Working Group, which standardizes how to use IEEE802.15.4e TSCH in IPv6-enabled mesh networks, and recently joined the IETF Internet-of-Things Directorate. Prior to that, he was a Postdoctoral Research Lead with the Prof. Kristofer Pister's Team, University of California at Berkeley, Berkeley, CA, USA. He founded and co-leads Berkeley's OpenWSN Project, an open-source initiative to promote the use of fully standards-based protocol stacks for the IoT. From 2005 to 2008, he was a Research Engineer with France Telecom, Orange Labs, Paris. He is fluent in four languages.