Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems pp. 145–149 DOI: 10.15439/2021F102 ISSN 2300-5963 ACSIS, Vol. 26

Connectivity Maintenance in IoT-based Mobile Networks: Approaches and Challenges

Vahid Khalilpour Akram
International Computer Institute
Ege University
Izmir, Bornova
vahid.akram@ege.edu.tr

Moharram Challenger
University of Antwerp
and Flanders Make
Flanders, Belgium
moharram.challenger@uantwerpen.be

***Abstract*—Connectivity is an important requirement in almost all IoT-based wireless networks. The multi-hop networks use intermediate nodes to create a communication path between other nodes. Hence losing some nodes may cut off all communication paths between other active nodes. Generally, the connectivity of a partitioned network can be restored by adding new or activating redundant nodes, moving available nodes to the new location, and increasing the wireless communication range of nodes. The restoration problem may have many constraints and sub-problems. The network may initially be disconnected, the nodes may be heterogeneous, reliable connections may be required between the nodes, we may have unreachable locations in the network area to put the new nodes or move exciting nodes, more than one node may fail at the same time and the expected coverage area may complicate the connectivity restoration problem. In this paper, we study the main challenges and methods of connectivity restoration in IoT-based wireless networks.**

***Index Terms*—Internet of Things, Connectivity, Multi-hop Wireless Network, Mobile Networks.**

I. Introduction

INTERNET of Things (IoT) is one of the fastest-growing and promising technologies that already formed a revolution in daily human life. In recent years, the new generation of smart buildings, structures, vehicles, clothes and almost all types of objects that every day are used by people benefit from IoT technologies [1], [2]. Technically, IoT is a set of small, low-energy electronic devices that can connect to the Internet over wired or wireless communication platforms [3], [4]. These devices may have different types of capabilities such as processing, sensing, and data storage. Recent advances in electronic and hardware technologies allow the generation of a wide range of tiny, low-cost, low-energy devices that support local processing, sensing, and various communication methods. The diversity and capabilities of IoT devices grow exponentially day by day which allows people to use them in different application areas. Tracking the status and location of patients and health care devices in hospitals [5], automation of activities and increasing the quality and efficiency of products in agriculture [6], tracking a mobile object in indoor or outdoor environments [7], controlling the objects in smart homes [8], automation of fabrication in factories [9], fast and efficient rescue systems [10], real-time monitoring systems of critical infrastructures [11], and providing ad-hoc or mobile communication platforms [12] are a few samples of IoT applications.

Connectivity is a critical necessity in all sorts of networks, including wired local area networks, wireless ad-hoc networks, mobile networks, and the Internet of Things. Ideally, all available devices in a network should be able to communicate with other devices in the network. In other words, the network must keep the connectivity between all available devices. In some types of networks, such as wired local area networks, preserving the connectivity between the nodes is almost straightforward. As long as the routers, switches, and cables work properly, any connected device may communicate with other devices under predefined security policies. In these networks, the status of endpoint devices has no effect on the connectivity of the network. For example, if a device stop working, the connectivity of other nodes will not be affected. However, preserving the connectivity in ad-hoc wireless networks may be much more complicated. In a wireless ad-hoc network, the nodes communicate with other remote nodes over multi-hop links. Using the ad-hoc routing protocols, each node forwards the received message to its neighbors which allows the nodes to remote nodes which are outside of their communication range. Therefore, the connectivity of nodes relies on the proper working of available intermediate nodes in the network. Consequently, if a node stop working, we may lose the connectivity between other working nodes. The problem will be much more complicated if the nodes are mobile. If a node changes its initial location, the connectivity between some other nodes may be completely destroyed. In a vehicle or drone network, if a mobile node changes its location, the communication paths between its neighbors will be changed. In the worst case, if there is no other redundant path, moving or losing a node may cut the communication paths to a large set of working nodes and waste many active resources.

The diversity of device and communication technologies allows establishing ad-hoc networks almost everywhere even in harsh environments such as mountains, sea-bed, and forests. In these networks, the nodes may use hybrid communication technologies such as Bluetooth, WiFi, GSM, LTE, LoRa, and Zigbee. Also, some nodes may be static with a fixed location and some other nodes may be mobile. For example, for real-time monitoring of an environment, we may distribute some sensor nodes in the environment and collect their sensing data over multi-hop links, mobile drones, or mobile vehicles (Fig. 1).

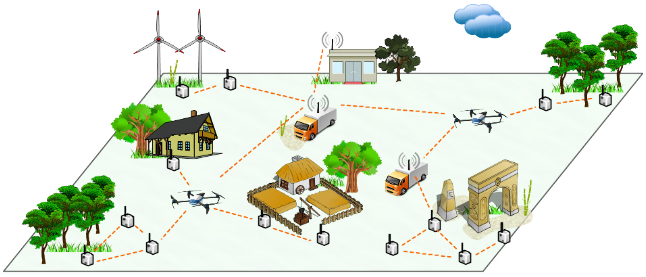


Fig. 1: Sample network for collecting sensed data from environment

A wide range of hardware and sensors are available for establishing a network similar to Fig. 1. For example, an ESP32 device support both WiFi, low energy Bluetooth communication technologies and have enough memory and processing power for most of monitoring applications. This device may be equipped with different types of sensors to gather various data from the environment. The new generation of drones [13] have more than one hour fly time and wide communication range which allows them to reach far locations miles away from the base station. However, preserving continues and reliable connectivity in wireless ad-hoc networks still is a challenging problem. In this paper we, focus on the applications and different challenges of connectivity maintenance in IoT based mobile ad-hoc networks. The remaining parts of this paper has been organized as follow; Section II provides a formal definition for connectivity problem and its different variants. Section III focuses on the open challenges and research problems on the efficient connectivity maintenance in mobile networks. Finally, Section IV provides the conclusion and future works.

II. Problem Formulation

We can model an ad-hoc network as graph $G(V, E)$ where V is the set of nodes and E is the set of edges between the nodes. For example, Fig. 2a shows a sample mobile ad-hoc network with 4 mobile nodes and 15 static nodes. Fig. 2b shows the graph model of this network where $V = \{0, 1, ..18\}$ and $E = \{(0, 7), (1, 3), (1, 7), ...\}$ is the set of links between the nodes. In Fig 2b triangles show the mobile nodes and circles show the static nodes. We assume that node 0, (the filled black node) is the base station of the network. The dashed big circles in Fig. 2b shows the communication range of the node which may differ based on the node types.

Generally, a network is called connected if there is at least a communication path between every pair of nodes. Connectivity is one of the most important requirements in all networks. In wireless ad-hoc networks, where the network connectivity relies on the proper working of nodes, different strategies have been developed to increase connectivity robustness. Placing redundant nodes, creating alternate paths between the nodes, and increasing the radio range of nodes are some of these strategies which have their own advantages and disadvantages. Placing redundant nodes in the environment is a simple and feasible approach but increases the network cost. Increasing the radio power of node allows them to connect more nodes but at the same time increase the energy consumption of nodes which are not desirable in the battery-powered networks. Creating and maintaining alternate paths between the nodes needs complex algorithms and real-time topology control which may be hard to implement.

Formally, a network is called k-connected if there is at least k path between every pair of nodes. Therefore in a 1-connected, there is at least one path and in a 3-connected network, there are at least 3 disjoint paths between every pair of nodes. Higher k values increase the reliability of the network but need precise nodes deployment and restoration strategies. Generally, challenges and problems on network connectivity can be classified into 2 groups as connectivity detection and connectivity restoration problems which are discussed in more detail in the following subsections.

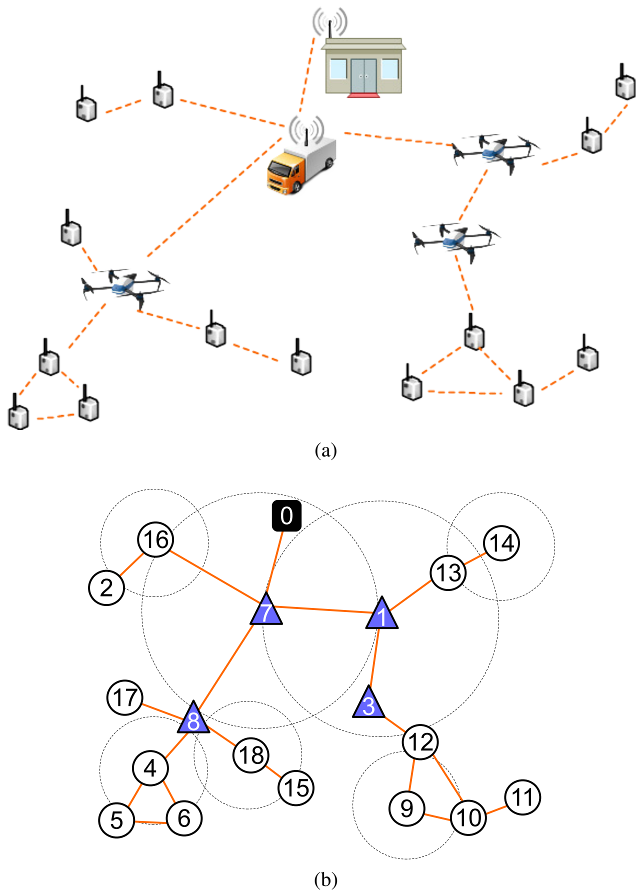
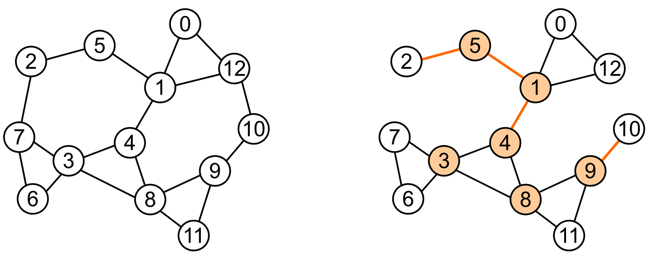


Fig. 2: a) Sample mobile network, b) Graph model of the network.

A. *Connectivity Detection*

Connectivity detection is the problem of finding the connectivity status and reliability of connections between the nodes. In the simplest case of the connectivity detection problem, the aim is to determine whether all nodes in the network are connected. In most applications, we need to ensure that all nodes have at least one communication path to each other which leads to the simplest form of connectivity detection problem. There are many central and distributed algorithms for the connectivity detection problem [14]. The central connectivity detection algorithms may use different methods such as depth-first search, network flow, path traversal, and matching to find the connectivity of the network.

Existing of a communication path between all nodes is a required condition in most applications, but in most cases is not enough. In wireless ad-hoc networks, 1-connectivity usually is considered unreliable because losing some nodes or links may disconnect a large number of nodes from the others. For example, Fig. 3a shows a sample 2-connected network that can tolerate any node or links failure without losing its connectivity. In contrast, Fig. 3b shows a 1-connected network with many critical links (orange color) and nodes (filled with orange) that losing each one destroy the network connectivity. A node whose failure destroys the network connectivity is called a critical node. Similarly, a link whose failure destroys the network connectivity is called a critical link or bridge. Detecting the critical nodes and bridges may help to improve connectivity reliability. For example, Fig. 3a and Fig. 3b show that adding only two links to the graph can resolve all critical nodes and links.



(a) (b)

Fig. 3: a) a sample 2-connected network, b) a sample 1-connected network with critical nodes and critical bridges.

Besides the bridges and critical nodes, we may find the minimum cut edges and minimum cut vertex of a network to measure its connectivity reliability. The minimum edge cut of a network is the smallest set of edges whose removal, destroys the connectivity of the network. For example, in Fig. 3a a minimum edge cut of the network is $\{(1,5)(2,7)\}$ which their removal disconnects node $\{2,5\}$ from the other nodes. Similarly, a minimum vertex cut of presented network in Fig. 3a is $\{1,9\}$. A network may have more than one minimum edge or minimum vertex cut. Finding the minimum vertex and edge cuts reveals the weak points and connectivity robustness of the network.

B. *Connectivity Restoration*

Network connectivity restoration is the process of increasing the reliability of network connectivity by reconnecting the disconnected nodes [15]. In some applications, the connectivity restoration is started after failure in some nodes that disconnect some working nodes from the others. However, some applications require continuous and reliable connectivity. In these applications, the connectivity restoration process must be started before complete disconnection to reinforce the unreliable connections. So, the connectivity restoration strategies can be classified into proactive and reactive groups. The proactive methods start after each node or links failure and reinforce the connectivity if required. For example, in the k-connectivity restoration methods [16], if a node failure reduces the k value, the restoration algorithm tries to increase the k value by moving other nodes or activating redundant nodes. The reactive methods start after network disconnection and try to reconnect the disconnected parts.

The connectivity restoration algorithms usually rely on the connectivity detection algorithms to determine the current connectivity status and decide about the required actions. Generally, the main approaches for connectivity restoration are moving the available mobile nodes to the new locations, activating or placing new nodes in the network environment, and increasing the radio communication of the nodes. Each approach has its own advantages and disadvantages. The movement-based methods use available resources in the network but require mobile nodes which are not feasible in some applications. Also moving the nodes from their initial location may disconnect some other links which complicate the connectivity restoration process.

Placing new nodes or activating redundant nodes simplifies the connectivity restoration process but requires additional resources. Also placing new nodes in the desired locations may not be possible in some harsh environments. Increasing the radio communication range of reaming nodes is another solution that may reconnect the disconnected parts. But increasing the radio communication range increases the energy consumption of nodes and may reduce the network lifetime. Besides these issues and constraints, the connectivity restoration problem has some other difficulties and challenges which are discussed in the next section.

III. CHALLENGES

In this section we discuss about the main challenges of connectivty restoration in mobile ad-hoc networks.

A. *Initial Connectivity*

A network can be initially connected or it can be disconnected after deployment. For example, after distributing a large set of sensor nodes to a forest using an airplane, with a high probability the resulting network will be disconnected. Some researchers assume that the network is initially connected and the connectivity restoration may start after failure or moving of nodes. This assumption simplifies the restoration problem as we ensure that restoring the disconnected links is enough for establishing the network connection. Connecting all nodes in a network that is initially disconnected is a hard problem because the set of possible solutions is very large. In the movement-based restoration, selecting the candidate nodes for moving, selecting the direction of movement, and calculating the movement distance is a hard problem because usually, the optimal solution needs a combination of different movements. For example, Fig. 4a shows the movement-based connectivity restoration in a network that is initially connected and Fig 4b shows another network that is initially disconnected. Similarly, connectivity restoration by placing new nodes or activating redundant nodes is much harder in the networks which are initially disconnected.

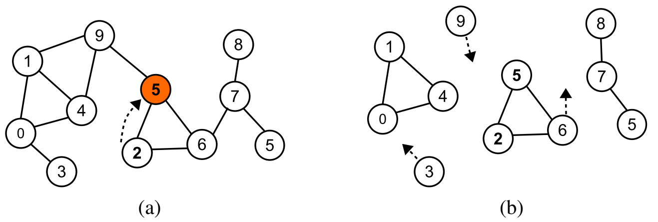


Fig. 4: a) Connectivity restoration when network is initially connected, b) Connectivity restoration when network is initially disconnected.

B. Heterogeneity

An IoT network may include a set of similar nodes with the same hardware and software properties. In such a homogeneous network all nodes have almost the same communication range, processing power, memory capacity, moving capability, etc. In contrast, in a heterogeneous network, the nodes may have different hardware and communication ranges. When the nodes have different communication ranges, some nodes may connect to a large number of nodes and some nodes may only have a limited set of neighbors. Also in a heterogeneous network, we may have uni-directed links which only allow one-way communications. Connectivity restoration in heterogeneous networks is much harder than homogeneous networks because the communication range of each node and the direction of links should be considered in graph model [17]. Most of the existing researches in connectivity restoration assume that all nodes have the same communication range.

C. k-connectivity

The aim of k-connectivity restoration is preserving the k value of a given network [14]. For example, in a 3-connected network, we want to preserve the 3-connectivity after losing some nodes. For $k = 1$ the problem is converted to the traditional connectivity restoration but for higher k values the problem will be much more complicated because moving every node in the network may affect the k value. In a 1-connected network, moving most of the nodes have no effect on the connectivity. For example in Fig. 4a moving each of the nodes $\{1, 2, 4\}$ does not affect the connectivity. However, in a k-connected network the set of candidate nodes that can leave their position without affecting k is limited, and finding these nodes needs some computation.

D. Target Positions

In the movement or deployment-based methods, we may assume that any position in the network area can be selected as a target position for moving the nodes or placing new nodes. Most of the existing research assumes that all nodes can move to their desired location or we may put the redundant or new nodes to the desired location. However, this assumption is not true for most real-world applications. Due to environmental conditions and obstacles, the nodes may not move to some location or we may not put the new nodes in the desired locations. To simplifies the restoration problem, some researchers assume that the new nodes can be only added to the location of exciting nodes or the nodes can only move to the location of existing nodes. This assumption simplifies the problem and converts it to a polynomial-time problem.

E. Single vs. Multiple Failure

Restoring the connectivity after a single node failure is generally simpler than the multiple nodes failure. After the failure of a single node its neighbor nodes may change their location to restore the connectivity because all of them may know the exact location of the failed node. However, in multiple nodes failure, a node and it's all neighbors may stop working at the same time. In this case, some of the failed nodes may be undetectable, or moving multiple nodes is impossible. Despite that the multiple node failures can happen in most real-world application, the researches that consider this case is limited and the number of proposed solutions is restricted [18].

F. Coverage

In some applications, the IoT nodes collect various data from enshrinement using different sensors. Losing a node in an IoT-based network or moving a node to a new location may lead to some coverage lost in the network. The coverage lost is not acceptable in some applications hence during the connectivity restoration we should preserve the maximal coverage. Restoring the connectivity and preserving the maximal coverage at the same time complicate the restoration process [19]. Especially in movement-based connectivity restoration, the nodes which have the minimal effect of total coverage area should be selected for movement. Generally, the coverage-aware connectivity restoration methods try to find the nodes which their covered are is also covered by the other nodes.

IV. CONCLUSION

Connectivity is one of the most important properties in most IoT-based wireless networks and robust connectivity is a vital requirement in most applications. In multi-hop networks, the connectivity of the network relies on the proper working of the nodes, and losing some nodes may destroy the connectivity.

In this paper, we surveyed the main challenges and methods of connectivity restoration in IoT-based wireless networks. Generally, the connectivity of a partitioned network can be restored by adding new or activating redundant nodes, moving available nodes to new locations, and increasing the wireless communication range of nodes. The restoration problem may have many constraints and sub-problems. Restoring the connectivity of a network that is initially connected is much simpler than connectivity all nodes in a network that is initially disconnected.

In a homogeneous network in which all nodes have the same hardware and software capabilities, the connectivity restoration is simpler than a heterogeneous network. In a heterogeneous network, the communication range and moving capabilities of each node may be different from the other nodes which complicate the restoration process. While the 1-connectivity allows the nodes to communicate with each other, the 1-connected networks are usually considered unreliable because losing a single node may destroy the connectivity. The k-connectivity restoration process tries to preserve k disjoint paths between every pair of nodes.

In some applications, the nodes in the network may go to every desired location or we may add new nodes to the desired location. However, in some other networks, the environmental conditions do not allow to put the new nodes or move the existing nodes to the desired locations. The connectivity restoration after a single failure can be simpler than the connectivity restoration after multiple failures because losing a node and its neighbors may complicate the restoration process. Finally losing a node in the network may lead to some coverage loss which may be not acceptable in some applications. Hence coverage-aware connectivity restoration algorithm tries to reconnect the connectivity while preserving the maximal coverage.

As future works, we will focus on the discussed challenges of the restoration problem to find efficient approaches that consider more than one criteria at the same time. For example, proposing a comprehensive approach that can handle multiple failures, maximize the coverage, preserve the k-connectivity, support heterogeneous nodes, and allow flexible target position selection can be very useful in many real-world applications. Also developing platform-specific languages and frameworks to support the deployment and connectivity restoration of different mobile and flying nodes under the discussed constraints can simplify the development and maintaining of complex IoT-based applications [20], [21].

REFERENCES

[1] S. Arslan, M. Challenger, and O. Dagdeviren, "Wireless Sensor Network based Fire Detection System for Libraries," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 271–276.

[2] L. Özgür, V. K. Akram, M. Challenger, and O. Dağdeviren, "An IoT based Smart Thermostat," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*. IEEE, 2018, pp. 252–256.

[3] B. Karaduman, T. Aşıcı, M. Challenger, and R. Eslampanah, "A cloud and Contiki based Fire Detection System using Multi-hop Wireless Sensor Networks," in *Proceedings of the Fourth International Conference on Engineering & MIS 2018*, 2018, pp. 1–5.

[4] B. Karaduman, M. Challenger, and R. Eslampanah, "ContikiOS based Library Fire Detection System," in *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*. IEEE, 2018, pp. 247–251.

[5] N. Karimpour, B. Karaduman, A. Ural, M. Challenger, and O. Dagdeviren, "IoT based Hand Hygiene Compliance Monitoring," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2019, pp. 1–6.

[6] M. S. Mekala and P. Viswanathan, "A survey: Smart agriculture iot with cloud computing," in *2017 international conference on microelectronic devices, circuits and systems (ICMDCS)*. IEEE, 2017, pp. 1–7.

[7] S. Shao, A. Khreishah, and I. Khalil, "Enabling real-time indoor tracking of iot devices through visible light retroreflection," *IEEE Transactions on Mobile Computing*, vol. 19, no. 4, pp. 836–851, 2019.

[8] Y. Jie, J. Y. Pei, L. Jun, G. Yun, and X. Wei, "Smart home system based on iot technologies," in *2013 International conference on computational and information sciences*. IEEE, 2013, pp. 1789–1791.

[9] I. E. Etim and J. Lota, "Power control in cognitive radios, internet-of things (iot) for factories and industrial automation," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 4701–4705.

[10] T. Ahn, J. Seok, I. Lee, and J. Han, "Reliable flying iot networks for uav disaster rescue operations," *Mobile Information Systems*, vol. 2018, 2018.

[11] S. L. Ullo and G. Sinha, "Advances in smart environment monitoring systems using iot and sensors," *Sensors*, vol. 20, no. 11, p. 3113, 2020.

[12] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[13] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, 2017.

[14] V. K. Akram and O. Dagdeviren, "Deck: A distributed, asynchronous and exact k-connectivity detection algorithm for wireless sensor networks," *Computer Communications*, vol. 116, pp. 9–20, 2018.

[15] Y. Zhang, J. Wang, and G. Hao, "An autonomous connectivity restoration algorithm based on finite state machine for wireless sensor-actor networks," *Sensors*, vol. 18, no. 1, p. 153, 2018.

[16] V. K. Akram and O. DAĞDEVİREN, "Tapu: Test and pick up-based k-connectivity restoration algorithm for wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 2, pp. 985–997, 2019.

[17] Y. Zeng, L. Xu, and Z. Chen, "Fault-tolerant algorithms for connectivity restoration in wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 3, 2016.

[18] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 844–856, 2012.

[19] N. Tamboli and M. Younis, "Coverage-aware connectivity restoration in mobile sensor networks," *Journal of network and computer applications*, vol. 33, no. 4, pp. 363–374, 2010.

[20] H. M. Marah, R. Eslampanah, and M. Challenger, "DSML4TinyOS: Code Generation for Wireless Devices," in *2nd International Workshop on Model-Driven Engineering for the Internet-of-Things (MDE4IoT), 21st International Conference on Model Driven Engineering Languages and Systems (MODELS2018)*. Copenhagen, Denmark, 2018.

[21] T. Z. Asici, B. Karaduman, R. Eslampanah, M. Challenger, J. Denil, and H. Vangheluwe, "Applying Model Driven Engineering Techniques to the Development of Contiki-based IoT Systems," in *2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT)*. IEEE, 2019, pp. 25–32.