Autonomic Faulty Node Replacement in UAV-Assisted Wireless Sensor Networks: a Test-bed

Published Version:
Autonomic Faulty Node Replacement in UAV-Assisted Wireless Sensor Networks: a Test-bed / Montecchiari, Leonardo; Trotta, Angelo; Bononi, Luciano; Di Felice, Marco; Natalizio, Enrico. - ELETTRONICO. - (2023), pp. 1155-1158. (Intervento presentato al convegno 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC) tenutosi a Las Vegas, NV, USA nel 08-11 January 2023) [10.1109/ccnc51644.2023.10059801].

Availability:
This version is available at: https://hdl.handle.net/11585/964281 since: 2024-02-29

Published:
DOI: http://doi.org/10.1109/ccnc51644.2023.10059801

Autonomic Faulty Node Replacement in UAV-Assisted Wireless Sensor Networks: a Test-bed

Leonardo Montecchiari[*‡], Angelo Trotta[*], Luciano Bononi[*], Marco Di Felice[*†], Enrico Natalizio[‡],
[*] Department of Computer Science and Engineering, University of Bologna, Italy
[†] Advanced Research Center on Electronic Systems "Ercole De Castro", University of Bologna, Italy
[‡] Technology Innovation Institute (TII), United Arab Emirates
Emails: {leonardo.montecchiari, angelo.trotta5, luciano.bononi, marco.difelice3}@unibo.it {enrico.natalizio}@tii.ae

***Abstract*—Several use-cases of the Internet of Things (IoT) rely on the development of large-scale Wireless Sensor Networks (WSNs) in harsh environments characterized by limited Internet connectivity and battery-powered operations. In such scenarios, the failure of a single node due to energy depletion or hardware issues may cause network partitions and disrupt partially or completely the system operations until the intervention of a human operator. In this paper, we investigate the usage of Unmanned Aerial Networks (UAVs) to enable sensory data collection and support resilient communications in presence of faulty sensor nodes. More specifically, we study the possibility of replacing the ground devices with UAVs which are able to temporarily restore the multi-hop communication towards the WSN sink. To this aim, we extended the *Uhura* framework, a platform for robotic networking, with novel features for automatic network partition detection and UAV-sink coordination. Then, we created a small test-bed composed of a Bluetooth Mesh WSN and one drone, and characterized the performance of the UAV-assisted WSN system in terms of packet delivery ratio of the end-to-end data flows.**

***Index Terms*—Unmanned Aerial Vehicles (UAVs), Wireless Sensor Networks (WSNs), Resilient communications, Test-bed**

I. Introduction

Several use-cases of the Internet of Things (IoT) involve the deployment of large-scale Wireless Sensor Networks (WSNs) for scenario monitoring [1]. The sensory data are filtered locally on the edge, and, in most cases, transferred towards a remote cloud for storage and analytics tasks. The data transfer process may be challenging in harsh environments such as remote or post-disaster areas where the coverage of ground communication systems is absent or temporarily disrupted. For this reason, some recent works investigate the application of Unmanned Aerial Vehicles (UAVs) as mobile sinks able to collect data from ground devices. UAV-assisted WSNs have been applied to delay-tolerant IoT applications related e.g. to smart agriculture, structural health monitoring, or smart cities [2]. Thanks to the flexible maneuverability of the UAVs, the performance of WSNs can be optimized in several ways, both in terms of energy efficiency and throughput on the air-to-ground communication links [3]. Indeed, a significant number of studies addressed the design of novel clustering protocols to determine the optimal WSN topology [4] [5]; similarly, path planning strategies have been proposed to maximize the throughput of the data collection process while taking into account the limited flight autonomy of the UAVs [6] [7].

In this paper, we investigate the deployment of robust UAV-assisted WSNs where the aerial segment is not only in charge of collecting sensory data but also providing backup connectivity in case of failures of one or multiple ground nodes. Indeed, in multi-hop WSNs, events like energy depletion or hardware failures of a node may cause network partitions and affect the operations of the IoT system. Traditional recovery approaches leverage extra relay nodes and multi-path routing strategies; however, their effectiveness depends on the number of failures to handle as well as on the topology of the WSN. As an alternative, UAVs can be used as mobile relays to re-establish the connectivity within the WSN until the intervention of a network operator [8]. Thanks to their autonomous mobility, the UAVs can place themselves in order to minimize the number of relays needed and to cope with dynamic channel conditions. Different node placement algorithms have been proposed in the literature [9] [10]. However, few real experiments of UAV-assisted WSNs have been reported so far.

In this paper, we attempt to fill such a gap by describing the design, implementation, and validation of UAV-assisted WSNs with faulty node replacement capabilities. Our setup is based on the *Uhura* framework [11], which enables the creation of overlay robotic networks composed of ground/aerial segments by abstracting from the specific Machine to Machine (M2M) communication technology in use on each link. We extended the *Uhura* software with a novel module for the network recovery strategy, in execution on the WSN sink node. In case a network partition event is detected, the module is in charge of notifying the UAV swarm by providing the location where a mobile router is needed; the target UAV will land at that position and re-broadcast each received *Uhura* message on each adapter. We demonstrated the feasibility of our approach through a small scale test-bed developed at the drone arena of the TII; the experimental setup involved one drone and a ground multi-hop WSN based on the Bluetooth mesh technology [12]. We measured the Packet Delivery Ratio (PDR) to show the effectiveness of the proposed system to restore the end-to-end connectivity within the WSN, in a completely autonomic way.

The rest of the paper is structured as follows. Section II reviews the state of the art of UAV-assisted WSNs. Section III introduces the *Uhura* Framework and its architecture. Section IV describes the novel *Uhura* module. The test-bed and experimental results are presented in Section V. Conclusions and future works can be found in Section VI.

II. Related Works

UAV-aided WSNs have been progressively investigated for different purposes: (*i*) saving energy on battery-powered WSN nodes, or enabling power transfer on the ground-aerial link [6]; (*ii*) enhancing the system performance by reducing the length of the last mile; (*iii*) enabling data collection in harsh/remote environments characterized by the absence or the temporal disruption of the ground infrastructures. At the same time, the deployment of such systems requires addressing complex optimization problems involving both the ground and the aerial network segments. On the one side, novel MAC protocols have been proposed to ensure that the ground sensor nodes are able to offload their data during the transit of the UAVs [5]. Cluster-based mechanisms [13] are effective approaches to reduce channel contention and the number of concurrent transmissions towards the mobile sink. On the other side, several studies focused on the optimization of the UAV trajectories in order to maximize the throughput of the data collection while taking into account the limited flight autonomy of the drones. Trajectory planning can be addressed with non Machine Learning (ML) techniques, including both constraint programming and heuristics, or ML technique. As an example of the first category, in [7], the authors address the join optimization on the trajectory, altitude, velocity, and link scheduling so that the mission time in UAV-aided data collection is minimized. Regarding the second category, the work in [4] aims to minimize the energy consumption of a WSN-UAV system by computing the visiting order of the WSN cluster heads through a Deep Reinforcement Learning (DRL) algorithm. In the studies described so far, the UAVs are employed as mobile sinks hence they are supposed to fly over the WSN and then return back to the ground station. In other applications, the authors investigate the usage of UAVs as mobile routers hovering over the scenario in order to restore connectivity among isolated ground devices. In [9] we proposed a distributed swarm management technique computing the position of the UAVs so that the maximum number of ground devices are covered while the connectivity of the aerial mesh is preserved. In [8], the authors propose a centralized algorithm to detect possible network partitions through the usage of UAVs. Our paper addresses a similar problem however it introduces the following novel contributions: (*i*) it is agnostic from the M2M communication technology in use on ground/aerial links thanks to the *Uhura* framework; (*ii*) it executes partition detection directly on the WSN sink node by exploiting the presence of the *Uhura* overlay; (*iii*) it provides experimental evidence of the possibility of using UAVs as mobile routers in emergency situations.

III. The Uhura Framework

Uhura is a novel software platform enabling the deployment of integrated robotic systems composed of UAV, mobile and static ground devices, in a communication agnostic way [11]. The main goals of the tool are to overcome the fragmentation problem caused by the presence on the market of multiple M2M communication standards and, at the same time, to exploit the multi-radio capabilities of most IoT devices. To this aim, *Uhura* works by creating an overlay network among heterogeneous devices (called *Uhura* nodes), each provided with a unique identifier: different links of the overlay network can be mapped to different M2M stacks. The link-layer communication operations using a specific M2M stack are managed through *Uhura* adapters, which are designed to be highly modular and interchangeable thanks to the usage of a common API. In addition, the *Uhura* framework allows to monitor the network performance of each active adapter; in [11] we demonstrated the possibility of dynamically switching from one adapter to another in order to meet the Quality of Service (QoS) requirement of the upper-layer application.

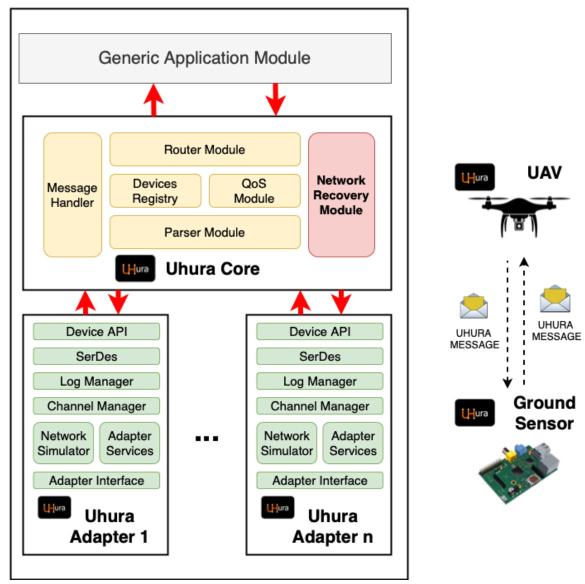

Fig. 1. The architecture of the *Uhura* framework.

Figure 1 shows the architecture of the software stack deployed in each node. The generic application module refers to the IoT application which needs to exchange data from/to other devices and is external to *Uhura*; however, it communicates with it via a publish/subscribe interface. The *Uhura* Core handles all the requests from the upper-layer application and sends them as generic messages on the overlay. In addition, it is in charge of creating the distributed network overlay through periodic `hello` messages, which are re-broadcasted in a multi-hop way; as a result, each *Uhura* node has full knowledge of the overlay graph. The *Uhura* Core can send a message using a specific Adapter or select the optimal one in a dynamic way (such feature is not exploited in this paper). The *Uhura* Adapter wraps the device API of a specific antenna module. In this paper, we extended the *Uhura* Core with a novel Network Recovery module which is in charge of detecting possible network partitions and, in the case of a WSN sink node, activating the node replacement procedure with the UAV swarm. Details of the network recovery algorithm are discussed in the Section below.

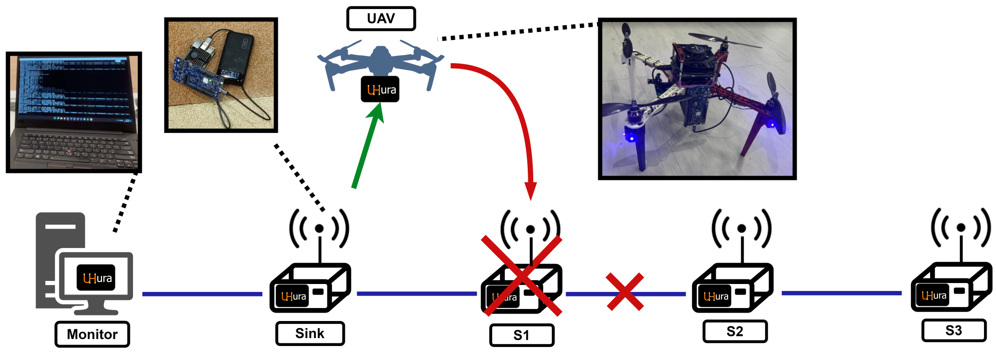

Fig. 2. The scenario of our test-bed included: a multi-hop Bluetooth mesh WSN composed of three nodes and a ground sink, one drone serving as mobile relay, and one laptop collecting the network logs.

IV. Network Recovery Algorithm

The main envisioned scenario for UAV-assisted WSNs is where the sensors data is collected by an elected cluster head (statically or dynamically selected among the network nodes) and then offloaded to one or more UAVs physically transporting them to a remote server not directly connected to the WSN [3]. In this work, we assume that the *Uhura* framework is installed on every device and that a sink node - statically selected among the ground devices - is continuously collecting the sensory data from the WSN. Periodically, a UAV fleet, composed of one or more UAVs, visits the WSN to gather the data from the sink node. During the WSN lifetime, some sensors may run out of energy or just disconnect from the other peers due to hardware failures. In such cases, multiple regions of the WSN could be disconnected due to the multi-hop topology. Our proposal envisages that the sink node is in charge of detecting the occurrence of a network partition and activating a recovery procedure; at the end of it, faulty sensors are replaced with UAVs working as mobile routers. In the following, we detail the recovery procedure running on the Recovery Module within the *Uhura* Core (Figure 1).

Let $G_0 = \{S_0, E_0\}$ be the overlay graph at deployment time, where $S_0 = \{s_{\text{sink}}, s_1, s_2, \ldots, s_N\}$ is the set of $N+1$ deployed sensors, with s_{sink} defined as the sink node, and $E_0 = \{w_0^{i,j} | s_i, s_j \in S_0\}$ is the set of weighted edges connecting the sensors. The edge weight $w_0^{i,j}$, with $0 < w_0^{i,j} \leq 1$, defines the quality of the network connection based on the Packet Delivery Ratio (PDR). Such metric is periodically computed by the QoS Module of the *Uhura* Core. We assume that the graph G_0 is calculated, and then frozen, after the initial deployment of the WSN.

We assume that at time t, a UAV fleet $U = \{u_1, u_2, \ldots, u_M\}$ of M UAVs, establishes a connection with the sink node s_1 to download the sensory data. After completing the data transfer, the sink node activates the recovery procedure from the Recovery Module in the *Uhura* Core. The procedure includes the following steps (see Figure 3(a)):

1) Calculate $G_t = \{S_t, E_t\}$ as the actual overlay graph at time t, with $S_t \subseteq S_0$. Let $G_F = \{S_F, E_F\}$ be the inactive graph where S_F is the set of disconnected sensors, with $S_F \cup S_t = S_0$, $S_F \cap S_t = \emptyset$, and $E_F \subset E_0$ are the arcs connecting only the inactive nodes. If $S_F = \emptyset$ then ends the recovery procedure.
2) Calculate $S_C \subseteq S_t$ as the set of candidate inactive sensors to be replaced by a UAV: $s_j \in S_C$ only if $\exists s_i \in S_t$ such that $w_0^{i,j} \in E_0$, i.e., if the active sensor s_i at time t were directly connected to the inactive sensors s_j at deploy time. Here, let $w^j_{0\leftrightarrow t}$ be the maximum edge weight from E_0 connecting the inactive node $s_j \in S_F$ with any sensor in S_t.
3) For each $s_j \in S_C$ calculate in G_F the connected components set CC_j to which s_j belongs to. We use a Breadth-First Search (BFS) graph visit starting from s_j to determine the connected components.
4) Select $\hat{s}_j \in S_C$ as the chosen sensor to be replaced by the UAV, where $\hat{s}_j = \arg\max_{s_j \in S_C} |\text{CC}_j| + w^j_{0\leftrightarrow t}$. In case multiple sensors have the same connected components set size, we choose the sensor with the maximum value of $w^j_{0\leftrightarrow t}$.
5) Choose randomly one available UAV from the UAV fleet and send it to the $\hat{s}_j$ coordinates in order to replace the operations of the inactive sensor. The target UAV will land at that position and re-broadcast each received *Uhura* message on each adapter.
6) If at least one UAV is left, then iterate the procedure starting from point 1.

V. Performance Evaluation

We evaluated the proposed network recovery strategy in a small scale test-bed deployed at the drone arena of the TII. The test-bed includes one ground WSN and one drone. The first consists of four Raspberry PI 4 devices, each provided with a Nordic nRF52840 BLE interface with the Bluetooth mesh stack. The ground nodes create a multi-hop Bluetooth mesh WSN as depicted in Figure 2: nodes $S1$, $S2$ and $S3$ transmit periodic measurements towards the sink, with nodes $S1$ and $S2$ working as mesh relay nodes. The drone is a DJI f450 frame with 4 rotors, a Pixhawk4 autopilot, and an Intel NUC companion board. The sink node is connected via cable to a laptop device which is used to collect the network logs

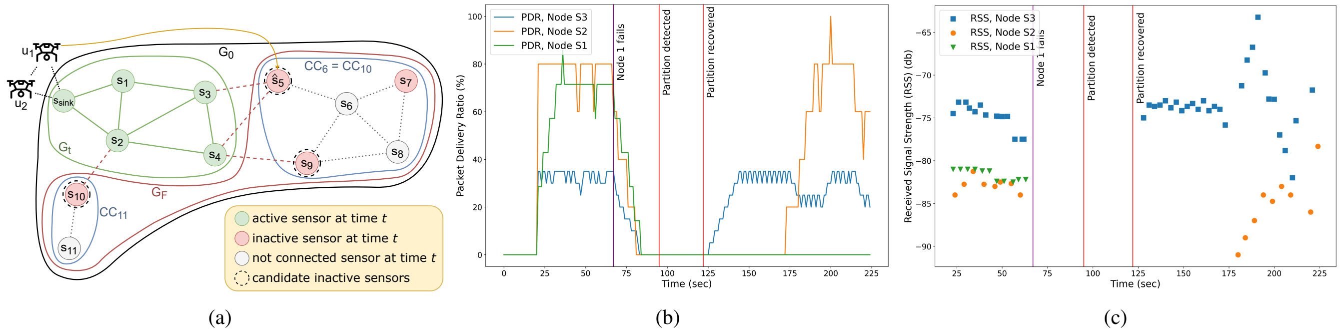

Fig. 3. Figure 3(a) shows the WSN scenario and the setup of the proposed network recovery strategy. Figure 3(b) shows the PDR for each data flow between nodes S1, S2 and S3 and the ground sink before and after the recovery procedure. Similarly, Figure 3(c) shows the RSS between each node and the UAV.

and compute the performance metrics. Both the drone and all the ground devices run a local instance of *Uhura* with a BLE adapter[1]. Balena Cloud[2] has been used to ease software deployment. In our experiments, we forced the failure of node $S1$, which at time t=70 seconds disconnects from the WSN and stops sending its own and relayed messages towards the sink. As a result, the recovery procedure is started, with the sink detecting the network partition at time t=95 seconds and issuing a command to the UAV; the latter lands close to node $S2$ at time t=125 seconds (see Figure 3(b)). The landing coordinates are hardcoded in the control software of the drone, leaving the optimization of the mobile sink placement as future work. Figure 3(b) shows the PDR computed at the sink node for the three different data flows originated from nodes $S1$, $S2$ and $S3$. The PDR is affected by the source node position: for this reason, $S3$ experiences the lowest PDR before node $S1$ failure due to the two hop distance from the sink. After node $S1$ failure (t=70 seconds), the data collection is completely disrupted for around 50 seconds; after $t > 125$ seconds, it is easy to notice that the data flows of nodes $S2$ and $S3$ are fully re-established and the PDR metric achieves similar results than before the node failure. However, the message relay operations are now operated by the UAV landed on the ground rather than by node $S1$. Figure 3(c) shows the Received Signal Strenght (RSS) of the wireless link between nodes $S1$, $S2$, $S3$ and the UAV. It is easy to notice that the RSS values before and after the recovery procedure are similar, but with higher fluctuations when the UAV relay is active. This is due to the different UAV antenna as well as the varied channel conditions.

VI. Conclusions and Future Works

In this paper, we addressed the design and evaluation of UAV-assisted WSNs in which the aerial devices work both as mobile data mules and as mobile relays able to replace the operations of faulty ground relay nodes. To this aim, a network recovery algorithm was proposed and implemented in the *Uhura* framework. The effectiveness and robustness of the UAV-aided network recovery solution were demonstrated through a small scale test-bed. We believe in the potential of such UAV-assisted WSNs for many different IoT applications and for this reason we are considering possible extensions related to: (*i*) optimization of the network recovery algorithm and specifically of the placement of the relay UAVs; (*ii*) replacement of sensing functionalities through the UAVs; (*iii*) additional experiments and measurements, taking into account also the lifetime of the UAV-WSN system.

[1]https://github.com/patonz/uhura/tree/main/src/lib/adapters/blesh

[2]https://www.balena.io/

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] Z. Hu, Z. Bai, Y. Yang, Z. Zheng, K. Bian, and L. Song, “Uav aided aerial-ground iot for air quality sensing in smart city: Architecture, technologies, and implementation,” *IEEE Network*, vol. 33, no. 2, pp. 14–22, 2019.

[3] W. Xie and X. Bai, “Research on data collection mechanism of wireless sensor network based on uav,” in *2021 2nd International Conference on Computing, Networks and Internet of Things*, ser. CNIOT ’21. New York, NY, USA: Association for Computing Machinery, 2021.

[4] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, “Uav trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9540–9554, 2021.

[5] S. Poudel and S. Moh, “Medium access control protocols for unmanned aerial vehicle-aided wireless sensor networks: A survey,” *IEEE Access*, vol. 7, pp. 65 728–65 744, 2019.

[6] A. Trotta, M. D. Felice, L. Perilli, E. F. Scarselli, and T. S. Cinotti, “Bee-drones: Ultra low-power monitoring systems based on unmanned aerial vehicles and wake-up radio ground sensors,” *Computer Networks*, vol. 180, p. 107425, 2020.

[7] J. Li, H. Zhao, H. Wang, F. Gu, J. Wei, H. Yin, and B. Ren, “Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in uav-aided data collection,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1464–1475, 2020.

[8] A. Zear and V. Ranga, “Uavs assisted network partition detection and connectivity restoration in wireless sensor and actor networks,” *Ad Hoc Networks*, vol. 130, p. 102823, 2022.

[9] M. Di Felice, A. Trotta, L. Bedogni, K. R. Chowdhury, and L. Bononi, “Self-organizing aerial mesh networks for emergency communication,” in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, 2014, pp. 1631–1636.

[10] Z. Kang, C. You, and R. Zhang, “Placement learning for multi-uav relaying: A gibbs sampling approach,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[11] L. Montecchiari, D. Albani, A. Trotta, M. Di Felice, and E. Natalizio, “Uhura: a software framework for swarm management in multi-radio robotic networks,” in *4th International Workshop on Wireless Sensors and Drones in Internet of Things (IEEE Wi-DroIT 2022)*, 2022.

[12] “Bluetooth sig. mesh profile specification version 1.0.1, revision date: 2019-01-21.”

[13] R. Sugihara and R. K. Gupta, “Optimizing energy-latency trade-off in sensor networks with controlled mobility,” in *IEEE INFOCOM 2009*, 2009, pp. 2566–2570.