IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 14 May 2024, accepted 18 June 2024, date of publication 27 June 2024, date of current version 8 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3419813

RESEARCH ARTICLE

Fog-Based Bootstrapping and Failure Repair Protocols for Wake-Up Receiver (WuRx)-Based Wireless Sensor Networks

MAXIMILIAN WEBER[1], **GHOFRANE FERSI**[2], **AND FAOUZI DERBEL**[1], **(Senior Member, IEEE)**
[1]Smart Diagnostic and Online Monitoring, Leipzig University of Applied Sciences, 04277 Leipzig, Germany
[2]Research Laboratory of Development and Control of Distributed Applications (ReDCAD), Department of Computer Science and Applied Mathematics, National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia

Corresponding author: Maximilian Weber (maximilian.weber@htwk-leipzig.de)

This work was supported by Leipzig University of Applied Sciences through Sächsisches Staatsministerium für Wissenschaft, Kultur und Tourismus.

ABSTRACT The Wake-up Receiver (WuRx) presents a promising solution for energy-efficient, low-latency, and reliable wireless sensor networks (WSN) for numerous applications. With the use of an always-on energy-efficient WuRx in addition to the main radio that is initially kept in sleeping mode, the energy of the sensor is preserved and the main radio is only woken up by the WuRx when it is required. While WuRx-based sensor networks offer numerous advantages, they encounter significant challenges during the bootstrapping and failure repair phases due to the sleeping state of the sensor's main radio. This limits its responsiveness and ability to effectively initiate or restore network functionalities. Also, the constrained range of WuRx exacerbates communication issues within the network, impeding seamless coordination and data transmission during critical phases. This paper aims to overcome these challenges by taking profit from the fog computing. We propose a novel fog-based bootstrapping protocol for indoor WuRx-based WSN. In our protocol, the network is divided into clusters each of which is managed by a fog node that orchestrates the bootstrapping process according to the nodes' rank. Furthermore, we present a novel fog-based failure repair protocol in which the fog node detects nodes and links failures and replaces in a time and energy-efficient manner the failed nodes and/or links with other alternatives to ensure continuity and reliability of communication within the network. We propose as well another failure repair strategy based on acknowledgment and specify what are the pros and cons of each approach. The performance evaluation of our bootstrapping and failure repair protocols has shown their time and energy efficiency and their adequacy in time-critical applications.

INDEX TERMS Wireless sensor network, wake-up based sensors, clusters, setup phase, failure repair, latency.

I. INTRODUCTION

In recent years, there has been a notable increase in research studies on WSNs, driven by their diverse applications across various sectors [1], [2] [3], [4] [5], [6] [7], [8], [9]. Each sensor in a WSN is equipped with a radio transceiver, a microprocessor, an electronic circuit for communication, and a typically battery-powered energy supply. These sensor nodes find widespread use in areas such as health monitoring, localization [10] environmental surveillance, military tracking, and animal detection and monitoring [11]. As the production of new and increasingly miniaturized devices, with more components are integrated within confined spaces for improved efficiency, it is necessary to involve a variety of different technical developments, to meet performance or reliability specifications. The development

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed M. A. Moustafa.

of semiconductors, which are essential for the evolution of wireless sensor networks, is also worthy of particular attention as shown in [12] and [13]. Enhancing the capabilities of sensor nodes is a fundamental requirement for advancing applications in WSNs. A balance between ultra-low power consumption to extend the lifespan of the network and latency-minimizing communication is crucial.

A promising solution to address these challenges involves the utilization of energy-autonomous, on-demand communication hardware known as wake-up receiver (WuRx) [14], [15]. These devices consume power in the order of microwatts, while conventional main radios consume power in the milliwatt range. To conserve energy, the main radio of sensor nodes can remain in sleep mode until the WuRx receives a signal to awaken the node [16].

WuRx-based sensors often have limited resources, making the initialization process challenging. Efficiently managing tasks like configuring communication parameters, and building routing tables while minimizing resource usage is a critical challenge. Furthermore, WuRx-based sensor networks often operate in dynamic environments, leading to changes in network topology. Adapting energy-efficient and rapid failure repair mechanisms to the dynamic nature of the network is a challenging issue.

There are a lot of papers in literature that have addressed energy efficiency by using WuRx based sensors. Even if the routing strategies are well described and the advantages of the individual strategies are shown, the occurrence of broken routes and malfunctioning sensor nodes is only poorly considered. Despite the use of the energy efficient WuRx and routing approaches, the published network setup and failure repair strategies are consuming a lot of energy and time because of the protocol behaviours.

We are addressing in this paper the aforementioned limits, and we propose time and energy-efficient bootstrapping and failure repair protocols. Our proposed protocols are made up of two layers: the sensor layer and the fog computing layer. The sensor layer consists of battery-powered sensors equipped with WuRx devices, while the fog computing layer comprises mains-powered fog nodes. These fog nodes are strategically positioned near the sensor nodes and are significantly more powerful than sensor nodes. Each fog node is responsible for managing a sensor cluster, building the routes in the bootstrapping phase, controlling the routing of data packets within the network, and repairing failures. Fog nodes maintain in their routing table, the source routes consisting of the intermediate nodes required to route a data packet from a specific node to its corresponding fog node. It is worth noting, that due to the important energy of the fog node, it is able to send packets to any node in its cluster without any intermediary. Whereas, a distant node in a given cluster is compelled to use intermediate nodes to reach its corresponding fog node.

In the bootstrapping phase, each fog node constructs the source routes that will be used by each sensor node in its cluster to reach their corresponding fog node. To this end, the fog node strategically wakes up the nodes recursively according to their distance from it. Then, the fog node employs as a first step a low Wake-up packet (WuPt) transmitting power, to wake up only the nodes that can communicate with it directly, add them to its routing table with the rank 1. After that, the fog node increases its transmitting power to wake up the second rank nodes that will construct their routes using rank 1 nodes as relays, and so on until all the nodes have joined the cluster and all source routes are constructed.

In the case of node failure, we propose a fog-based failure repair strategy, in which the entire error detection and correction is operated by the fog node as a central anchor point. This entails the utilization of a timer armed by the fog node after each data request, and additional Hello-messages to detect the failure. This allows the fog node to subsequently update its routes and guarantee proper communication functionality. We have as well proposed the acknowledgment (ACK)-failure repair, in which the nodes send an ACK of receipt after receiving a data packet. If the so-called ACK is not received by the sender, it will avoid the faulty node, and wakes up alternative nodes and use them as an alternative route to inform the fog node about the inoperative node.

We implemented our proposed protocols in a real-world scenario, utilizing the WuRx developed at Leipzig University of Applied Sciences. The experimental results have shown that our fog-based bootstrapping and failure repair protocols offer the best trade-off between time and energy efficiency. The results have proven also that the fog-based failure repair is better than the ACK-based failure repair in the case of a steady network witnessing a reasonable number of failed nodes. However, in the case of an extremely faulty network, the ACK-based approach presents better results.

The remaining part of the paper is organized as follows: Section II presents a summary of previously published work related to Wake-up radio assisted protocols, describing the setup phase and failure repair. In Section III, we elaborate on our proposed approaches. Section IV presents the performance evaluation of our proposed protocols, thereby concluding the paper in section V.

II. RELATED WORK

In the existing body of literature, various scientific studies have delved into addressing the bootstrapping and failure repair. As an illustration, researchers describe the initialization phase in [17], employing flooding to ascertain routes from the source to the destination while considering energy costs. It constructs routing tables, assigning probabilities to sub optimal paths based on energy metrics. The destination node initiates the connection by flooding towards the source, setting the "Cost" to zero. Intermediate nodes forward requests to neighbours closer to the source. Upon reception, the energy metric is calculated and added to the total path cost. Paths with high costs are excluded, and low-cost paths are added to the forwarding table. Nodes have multiple low-cost neighbours for packet routing. A benefit worth noting in this work is its ability to facilitate swift responses to both incoming and outgoing nodes within the network. Minimal route maintenance occurs, with infrequent localized flooding for path preservation. Although using non-optimal paths has the advantage of avoiding node failures due to exhaustion, transmission delays can occur. Even if localized flooding is performed at short distances from the destination to the source to keep all paths alive in the maintenance process, this leads to increased energy consumption of the battery-powered sensor nodes, even if there is no limitation on the network's functionality.

The network exploration outlined in [18], based on the wake-up signal, involves transmitting node IDs and neighborhood relations to the sink. Each node generates a broadcast packet with this information and additional metadata about locally buffered measurements. The data sink collects status information from all nodes. WRTA then optimizes topology and computes route paths. After a predefined timeout, it analyzes collected data and calculates an optimized communication tree. The root is the data sink, and the calculation routine utilizes all possible first-level links. A fitness function ensures a balanced communication infrastructure, considering node metadata for energy-efficient operation. An important aspect of this presented approach is its capability to incorporate various network and cross-layer parameters to optimize route paths, including considerations such as battery status, bandwidth constraints, and quality of service (QoS) parameters. In case of broken links, the data sink detects issues, analyzes routing problems, and transmits a route configuration update with alternative paths to the topology subset. The authors mention, that a critical situation with packet loss may occur based on the increased data volume, due to the limited network bandwidth of the 802.15.4 interfaces. It was also noted that the suggested routing protocol encounters elevated packet loss in a network with a depth of 3 hops for the route configuration process. The data sink identifies broken links during data transmission, then starting route configuration update, including alternative paths sent to the relevant topology subset.

CTP-WuR [19] functions as a distance-vector tree-based collection protocol, establishing and managing a minimum-cost tree rooted at the sink. Each node maintains a cost estimate for its route to the sink, with the sink having a cost of zero. Nodes calculate their cost by summing the cost of their next hop (parent) and the link cost to the parent. Routing information is exchanged through broadcasted control beacons containing the transmitter's local cost estimate. The advantage to note is that it reduces end-to-end latency by extending the achievable wake-up range. This allows for the reduction of both latency and energy consumption by bypassing the relay of the data packet through intermediate relay nodes. Adaptive beaconing reduces beacon frequency. data path validation is used for topology maintenance. In case of potential loops or inconsistencies, the transmitter's cost is compared to its next hop's cost for topology repair. In case of node failure and broken links, nodes attempt direct transmission multiple times. Unsuccessful attempts designate the parent of the parent as unreachable, relying only on the immediate parent for data forwarding. However, there is no further testing and propagation of information about existing errors to update the neighbouring nodes routes. The drawbacks of CTP-WuR, is that it requires additional latency due to the relaying of wake-up messages. This makes this strategy unsuitable for time-restricted applications. Moreover, packets are sent to the parent of the parent of the source node without being sure that the range is suitable. Extra time and energy will be wasted if the two nodes are far apart and are unable to communicate in a direct manner. This will lead to an increase in the packet error rate.

The existing solutions consume an increased amount of energy and are time-consuming because they inform the entire network by flooding or put a large number of nodes into the active state by broadcasting messages. In addition, faulty routes are not consistently analyzed and their routes are not restored in a timely manner to guarantee the best possible efficiency in aspects of energy consumption and latency.

To overcome the aforementioned limits, in the setup process, unlike the mentioned approaches, we offer an energy and time efficient strategy by waking up only the required nodes and integrating nodes in a broadcast manner. In addition, in the event of failed communication, we give the possibility to restore broken links or nonfunctional nodes with in a timely manner and with little energy expense. By immediately checking whether it is a broken link or an inoperative sensor node, the network is restored to its optimum state instantaneously.

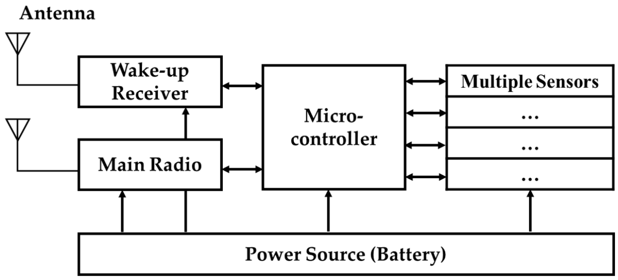

FIGURE 1. A wireless sensor node equipped with a WuRx, according to [21].

III. FOG-BASED BOOTSTRAPPING AND FAILURE REPAIR

A. WuRx-BASED SENSOR NODE DESCRIPTION

The wireless nodes employed in this study utilize readily available commercial off-the-shelf (COTS) components, known for their cost-effectiveness in terms of setup, maintenance, expansion, and development. The sensor node is built out of antennas, WuRx, main radio, sensors, energy source, and microcontroller, as shown in Fig. 1. To conserve energy, the WuRx, powered by a 3.0 V battery, incorporates passive components for continuous radio channel monitoring. The LF

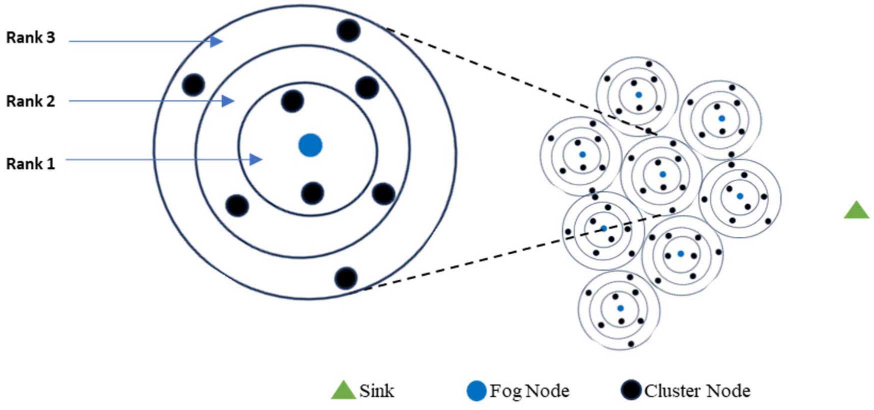

FIGURE 2. Fog-based clustered network with three ranks.

WuRx chip AS3933 is a low-power, 3-channel ASK receiver with a current consumption of roughly 3 μA during listening mode. The AS3933 [20] is engineered for carrier frequencies ranging from 15 kHz to 150 kHz, utilizing On-Off-Keying (OOK) modulation. The modulated OOK signal at 18.7 kHz is converted to a carrier signal at 868 MHz.

When received by a sensor node, it is subsequently reverted to the kHz band using a passive envelope detector. The AS3933 correlates the incoming signal with the node-specific address and triggers an interrupt when both addresses match. For communication, the radio module employed is the SPIRIT1 [22] by STMicroelectronics. When transmitting at +12 dBm output power at 868 MHz, it has a current consumption of 21 mA and consumes approximately 9 mA while receiving, with an estimated sensitivity of around −118 dBm.

The microcontroller (MCU) featured on these boards is the 16-bit MSP430G2553 [23], produced by Texas Instruments, operating at 8 MHz. The MSP430 offers the advantage of multiple low-power modes, with a power consumption of 2.55 μW in Low-Power Mode 3.

B. NETWORK DESCRIPTION

Our proposed system is made up of clusters. In each cluster, there is one fog node and multiple cluster nodes. Fog nodes are robust nodes, characterized by their ample energy reserves and high transmission power. Cluster nodes are energy-sensitive nodes equipped with WuRx, functioning as members within these clusters. These sensor nodes remain dormant until they receive a specific wake-up packet (WuPt) to activate them. When the sensor node is not in use, the entire sensor node remains in sleep mode, effectively minimizing power consumption. Each fog node operates on a primary power source using highest transmission power level, enabling to directly communicate with all nodes in its designated cluster. Conversely, the cluster members, powered by batteries, utilize energy saving low transmission power levels, with intermediate nodes acting as relays. This configuration results in an asymmetric link between each fog node and its corresponding cluster members.

To delve further into the detailed configuration of an individual cluster, we have presented in Fig. 2, the structure of a given cluster. It's crucial to emphasize that each cluster is under the jurisdiction of a sole fog node, with several cluster nodes designated to it. The fog node stands as the exclusive network entity responsible for facilitating communication between the central sink and the individual sensor nodes within its cluster. However, within a cluster, data exchange is made possible through a multi-hop approach. In other words, the individual sensor nodes can interact with other cluster nodes and transmit packets to the fog node, utilizing cluster nodes as intermediary relays.

Our work offers an approach for setting up a wireless sensor network and the possibility of restoring the network functionality in the event of failure. The considered clustered network is based on fog nodes and WuRx-based cluster nodes, but is neither application-bound nor hardware-bound to special specifications of a WuRx.

C. FOG-BASED BOOTSTRAPPING PROTOCOL

The individual clusters are divided into various ranks as depicted in Fig. 2. The differentiation between these ranks is achieved by the fog node waking up the individual nodes varying transmission power for the wake-up signal. Nodes addressed and awakened with minimal transmission power are assigned to rank 1 and are able to communicate directly with the fog node. Upon receiving this WuPt, the sensor nodes transition to the data transmission mode and acquire the so-called RouteRequest packet (RREQ). This packet contains the precise ID, which is the address of the fog node, and the rank counting, which is 0. Cluster nodes that receive the RREQ, store information indicating that the fog node is at a 1-hop distance, enabling direct communication. Then responding with a RouteReply packet (RREP), transmitting their own ID to the fog node, and increasing the rank counting, signaling the fog node that they belong to rank 1.

Algorithm 1 Setup Phase at Fog Node Side

```
Procedure ConstructSourceRoutes
if Fog node receives the signal for network setup phase then
    Wake-upPower=MinPower    ▷ Wake-upPower is the transmitting power employed by the fog node when sending the wake-up packet and MinPower is the minimal transmitting power that will be employed by the fog node to wake up the first rank of cluster sensors
    while Wake-upPower<=MaxPower do
        Fog node broadcasts wake-up packets with Wake-upPower
        Wake-upPower=Wake-upPower+IncreasedPower    ▷ IncreasedPower is the amount of power that is increased by the fog node to reach the following rank of cluster sensors
        if RREP received then
            Store source route included in RREP    ▷ the fog node extracts the source route from RREP and stores it in its routing table
            Send ACK to RREP source node
        else
            Stay in receive mode
        end if
    end while
end if
```

The fog node confirms the reception of the RREP with an ACK, signaling to the awakened nodes that their RREP has been received, and that the nodes can initiate an RREQ to add additional nodes to the network. This starts, after the fog node has sent a new WuPt using a higher transmission power to wake up nodes, that will be assigned to rank 2.

After the nodes in the second wake-up round have transitioned to the reception mode, they now receive an RREQ packet from the nodes awakened in the first wake-up round belonging to rank 1. Consequently, the recently activated nodes retain the data conveyed by nodes from rank 1 upon receiving the RREQ. They store information indicating their placement in the subsequent rank, which is rank 2, and identify the IDs of their next-hop nodes. After they have increased the rank counting, it is now at 2, the RREP is sent back to the sender of the first received RREQ. The sender now forwards the data packet to the fog node. Here, the fog node stores all the RREPs from rank 2 IDs to construct the source routes and subsequently sends another ACK to the rank 2 nodes. This process is repeated until all the nodes join the network.

Cluster nodes possess knowledge about routes to their direct neighbours, while the fog node, as the communication centre of the cluster, has all the information about all cluster nodes. Let's revisit the process of storing source routes in the routing table of the fog node and how this table encompasses all the alternative routes. The fog node gathers data from cluster nodes during different rounds of the setup phase and records it in its routing table, consisting of $n + 2$ columns, where n is the number of ranks in the cluster. In our example, $n = 2$. The first column stores information about the rank, the subsequent column holds the node's address, and the remaining two columns contain the nodes functioning as relays for data communication. Since nodes can utilize multiple routes or alternative paths formed during the setup phase, these are also stored in the fog node table.

Algorithm 2 Setup Phase at Cluster Nodes Side

```
Procedure ParticipateInSourceRoutes
if WuPt received then
    Wake-up
    if RREQ(ID,rank,Fog) received then
        Store ID and rank in myneighboursSet
        myrank=rank+1
        send RREP (myID,ID,Fog)    ▷ The current node will send an RREP to the RREQ source node, by putting its identifier myID in the RREP's source route field
    end if
    Enter reception mode
    while RREP received do    ▷ When the RREP is received by a relay node, it should add its identifier myID to the RREP's source route and forward it to its next Hop myNextHop. The process is repeated until reaching the fog node
        Add (myID, RREP)    ▷ Adds the current node's identifier to the source route's field of the RREP
        forward (RREP,myNextHop)
    end while
    if cluster node receives ACK then    ▷ The reception of the ACK means that the current cluster node is ready to start the network exploration by broadcasting RREQ
        Cluster Node broadcasts RREQ
    else
        Sleep mode
    end if
end if
```

To enhance comprehension of the process, we give two algorithms elaborating further on the specific steps for both the fog node and the cluster nodes. Algorithms 1 and 2 explain the setup phase respectively in the fog side and in the cluster nodes side.

D. WuRx-BASED ROUTING PROTOCOL

In a WuRx-based Wireless Sensor Network (WSN), the sensor nodes remain in a dormant state, necessitating their awakening for data transmission. The most straightforward approach involves each node transmitting a wake-up signal to the subsequent relay node until the data reaches its destination. However, this method proves time-consuming, as it entails waiting for each intermediate node to awaken and become ready for packet transmission at every step. Consequently, even slight delays of a few milliseconds can significantly impact time-restricted applications. Our

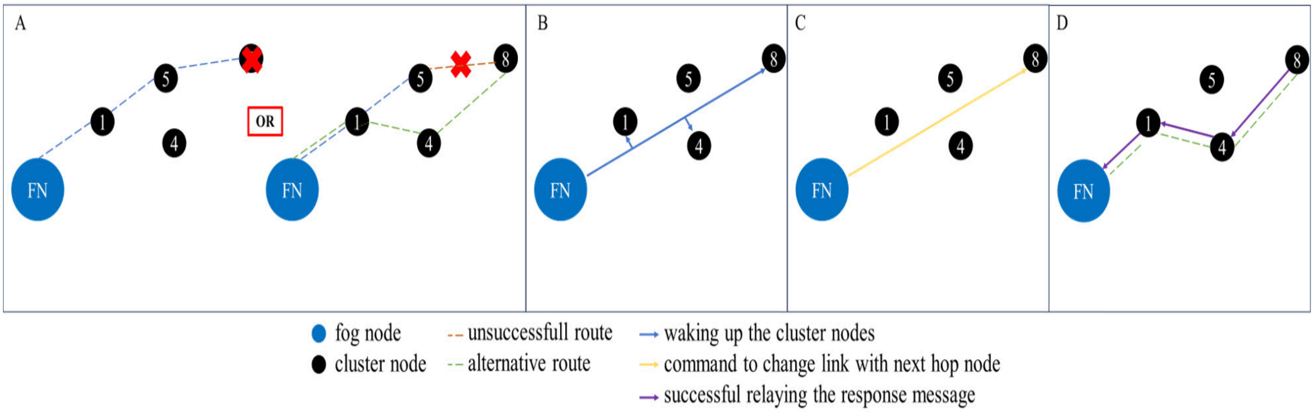

FIGURE 3. **A: Node failure and link failure. B-C: Testing of failure type. D: Establishing new route.**

protocol CWM [23] is designed to surpass this constraint. In CWM, the fog node picks from its routing table the source route that will be employed by the data packet from the data source to the fog node and wakes them up priorly before the data packets routing using a multicast wake-up packet. This guarantees a prompt activation of nodes that will serve as relays from the source to the fog node.

We stress here the fact that our bootstrapping and failure repair strategies are not restricted to CWM and can be applied in any source route routing protocol employing WuRx-based sensors like [24].

E. FAILURE REPAIR STRATEGIES

WSNs are powered by batteries due to their physical limitations, which makes them energy inefficient. If node A runs out of energy, the routes using node A become inaccessible, resulting in inefficient routing. In addition, the presence of physical obstacles degrades link quality and can even lead to packet loss. To ensure efficient and error-free network operation, it is therefore crucial to deal with node and link failures. For this reason, we need mechanisms to detect faulty nodes and links and enable alternative routes for data transmission. To achieve this, we propose troubleshooting protocols, which are described in more detail in the following subsection.

Two main problems can occur in the event of faulty transmission. It is possible that a sensor node is completely inoperable and therefore data transmission is no longer possible. This may be because the sensor node is no longer sufficiently supplied by the battery, or because it has simply been destroyed or damaged due to physical impact and is now no longer functional. It is also possible that the link between two nodes, i.e. the connection between them, has been interrupted. This can be temporary, but also permanent, if the route has been permanently disrupted by structural changes to the network environment.

In the event of temporary loss of routing routes which are disrupted or even completely blocked by physical objects, for example, the link between two nodes can be re-established. It is therefore advisable to send live messages every 24 hours to check the performance and connections between the nodes in order to keep the network up to date.

For performance reasons, however, the network should return to its initial best and most effective state as quickly as possible. It is therefore necessary to determine the two error types described and update the network accordingly. As already described, it can be a complete failure of the node, as shown on the left in Fig. 3 under sub-item A, or an interrupted connection between two nodes, as shown on the right. In order to analyze this error case, it is essential to determine whether it is a node or link failure.

We propose in the following two distinct failure repair strategies, one fog-based failure repair strategy and one acknowledgment-based failure repair strategy.

1) FOG-BASED FAILURE REPAIR PROTOCOL (REACTIVE FAILURE REPAIR)

The data transmission in each cluster is coordinated entirely by the fog node. In the event of a node failure, which is either a node meant to transmit data or a node acting as a relay, it is necessary to identify the source of the issue first. For this purpose, the fog node initiates the fault detection mechanism and individually checks the communication capability of each node involved in data transmission. When the fog node sends a data request to a specific node, it activates a timer. If the timer expires without receiving the expected data, the fog node identifies that the route contains at least one malfunctioning node or a failed link.

When data is required from a rank 1 node A, the fog node wakes up this node with a WuPt and sends a ‘Hello’ sequence. If the fog node does not receive confirmation within a specific time frame, it is determined that the corresponding node is no longer functional or the link between it and this node is failed. To verify which is the case, the fog node wakes up another physical neighbor B to node A with the least rank and sends a packet to node A, and informs it to use node B as an intermediate node to send the response. It arms a timer, if the fog node receives the response before the timer expiration, the fog node concludes that node A is still functioning and the link between it and node A is failed. Hence, it replaces the direct link between it and node A, with the link between node A and node B and the link between node B and it.

If the timer is expired without getting a response from node A, the fog node is sure that node A is failed. Subsequently, it removes all entries in its routing table that involve the non-functional node and informs the nodes having the failed node as the next hop, to change it with another alternative node. The fog node then informs the sink that the requested data can no longer be retrieved because this cluster node is no longer functional.

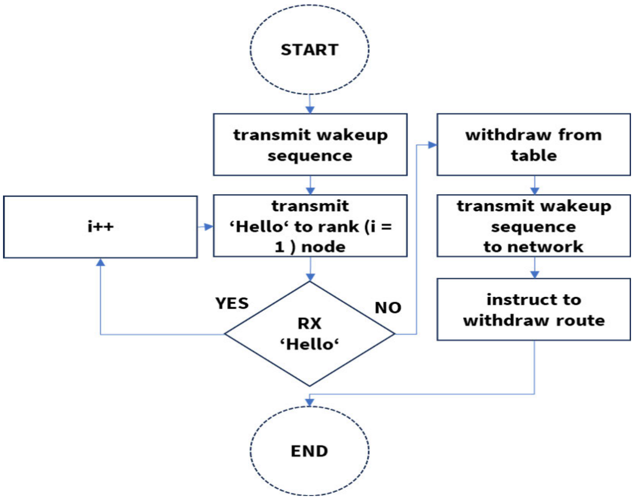

FIGURE 4. Fog-based orientated approach.

When data is requested from the next higher ranks, necessitating multi-hop communication, all nodes serving as relays are tested. Fig. 4 gives more insight to this process. The test begins from the lowest rank node to the highest rank node. Since, as mentioned earlier, individual nodes are in sleep mode, the fog node first sends a WuPt, followed by the 'Hello' sequence to its physical neighbour node A that is used in the route having the failed node. If the first rank node responds, the fog node addresses the second rank node B that is indicated in the failed route and sends it 'Hello' message. Evidently, the node B will use the node A as an intermediary to respond to the fog node.

When the fog node does not get a response, it verifies the nodes forming the routes one by one until identifying the node causing the problem. Once the problem is recognized, the fog node checks whether it is a node failure or a link failure. To test this, the fog node selects a source route from the non-responding node from its routing table, using an alternative node as the next hop (to change the connection) as shown in Fig. 3 at part B. To do this, the fog node wakes up the nodes and instructs the corresponding node to change its next hop node link and respond to the verification message, as shown in section C. If the fog node does receive a response, the cluster node is operable and the new link is to be used for further communication as shown in section D. If there is no response from the addressed sensor, it is clear that the node is no longer functional and can no longer be used for data transmission.

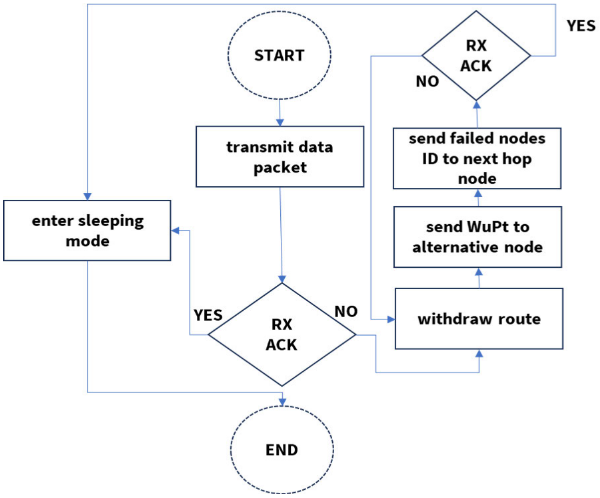

FIGURE 5. Acknowledgement-based failure repair.

2) ACKNOWLEDGEMENT-BASED FAILURE REPAIR (PROACTIVE FAILURE REPAIR)

The approach presented in this section and shown in Fig. 5 differs from the previously presented one since it does not rely on the fog node, but instead on that the individual cluster nodes verify the success of data transmission. They achieve this by sending an acknowledgment packet for every data relay, indicating whether the data was successfully received by the next node. If the sending node does not receive an ACK for the data packet during a period T, the sending node retransmits the data packets and waits for the ACK during the period T. After that period, if the sending node does not receive the ACK, the addressed node is considered failed and an alternative route must be chosen. To detect the source node failure, the fog node when it sends a data request, it arms a timer. If the timer is expired without getting a response, it considers the source node as failed. The fog node then informs the Sink that this node cannot be queried for data and replaces it with another appropriate alternative sensor node.

If a relay or its corresponding link is defective and does not respond to the sender with an ACK, this sender S awakens alternative nodes that are indicated in the alternative route and use them as temporary relays to deliver the packet to the fog node with indicating the probably failed node identifier in a specific field in the data packet.

When the fog node receives the data packet, it identifies the faulty issue, the fog node should distinguish between a node failure and a link failure. As illustrated in Fig. 3 at part B, the fog node tests this by choosing a source route from the non-responding node from its routing table and using an alternate node as the next hop (to change the connection). In order to accomplish this, as mentioned in section C, the fog node awakens the nodes and gives the corresponding node instructions to modify its next hop node link and reply to the verification message. The cluster node is operational and the new link should be used for further communication, as indicated in section D, if the fog node receives a response. It is evident that if the addressed sensor does not respond, it is no longer operational and cannot be used. If this is the case, the fog node updates the source routes.

IV. PERFORMANCE EVALUATION

The experiments have been conducted, considering a single cluster using 8 battery powered WuRx-based sensor nodes and one mains-powered fog node. The characteristics of these nodes are depicted in Table 1.

TABLE 1. Parameter values used in this experimental measurement.

Parameter	Value
Hardware	AS3933 (WuRx), SPIRIT1 (Transceiver), MSP430 (MCU), Raspberry Pi 4
Frequency Band	868.0 MHz
Bit-Rate	AS3933 (1.16 kbit/s), Spirit (38.4 kbit/s), MSP430 (8 Mbit/s)
Modulation	AS3933 (OOK), SPIRIT1 (FSK)
WuPt	Manchert Bits (8 Carrier Burst, 6.5 Preamble, 16 Address)
'Hello'	size 18 Byte (8 Byte Preamble, 4 Byte Sync, 1 Byte Length, 1 Byte Address, 3 Byte Payload, 1 Byte CRC)
ACK	size 18 Byte (8 Byte Preamble, 4 Byte Sync, 1 Byte Length, 1 Byte Address, 3 Byte Payload, 1 Byte CRC)
Transmission power sink and fog node	(+11 dBm)
Transmission power cluster nodes	WuPt (+11 dBm), 'Hello' and ACK (−34 dBm)

To experimentally collect measurement data and evaluate the results, an oscilloscope was employed to measure the energy consumption in milliampere-seconds (mAs) and the time in active mode in milliseconds (ms) for all sensor nodes. As shown in Fig. 6, the technical setting for the analyses also includes a shunt resistor of $12\ \Omega$ and a low noise amplifier to amplify the signal by a factor of 10. To gain a deeper understanding of the energy consumption measurement results for each sensor node, an individual measurement of their active times was conducted. The obtained measurement results detail the energy consumption and time duration for each node when sending, waiting, or receiving a Request (REQ), Wake-up Packet (WuPt), or Acknowledgement (ACK). The measurements were conducted using the described setup, and the subsequent section provides the outcomes.

A. BOOTSTRAPPING PROTOCOL

We have measured the time and energy required to reach the steady state. The proposed bootstrapping protocol requires only 347.46 ms to reach the steady state and consumes only 1.79 mAs in the setup phase. This is thanks to the rank-based joining process employed by the fog node that avoids to activate all the nodes in the same time and rather organize wisely their joining process by activating nodes sequentially from the nearest to the farthest ones.

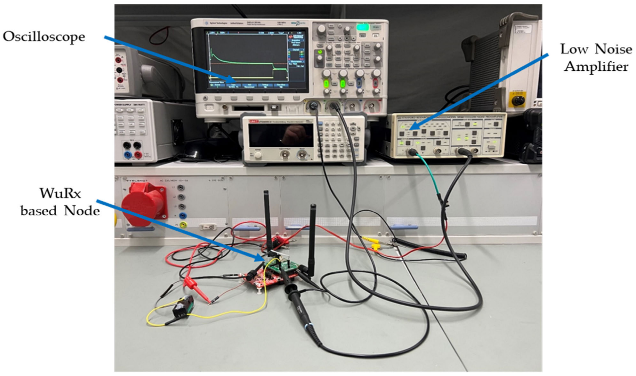

FIGURE 6. Set up used for measurements, according to [23].

B. FAILURE REPAIR PROTOCOLS

In the analysis of different scenarios, the results depicted in Fig. 7 reveal the following insights. The time required to detect the failure in transmission and to update routes in the network was considered. It is evident that, with the ACK approach, varying durations are required to detect and rectify errors depending on the rank of the failed node. In contrast, the fog-based approach demonstrates that nearly every error demands a consistent duration for detection and resolution. In summary, regardless of the rank at which the error occurs, the fog-based strategy necessitates a longer delay because it starts the failure repair only when the fog node does not receive the required data packet after a given delay. Furthermore, the fog node does not know which is the failed node in the set of nodes that are used in the routing process. It is compelled to verify these nodes one by one, which requires additional time. Whereas, the ACK-based approach can detect directly the failed node and replace it instantly, which saves time. Relevant scenarios are shown in Table 2.

TABLE 2. Different scenarios of failed nodes.

Scenario	Explanation
A	Data from rank 1 node - addressed node is inoperable
B	Data from rank 2 node - rank 1 relay node is inoperable
C	Data from rank 2 node - addressed node is inoperable
D	Data from rank 3 node - rank 1 node is inoperable
E	Data from rank 3 node - rank 2 node is inoperable
F	Data from rank 3 node - addressed node is inoperable

When considering the energy consumption in Fig. 8, in scenario A, where data is requested from a rank 1 node, it is observed that both the fog node (FN) and the ACK-based approach exhibit the same energy consumption. As only the energy consumption of the battery-operated cluster nodes is considered, this value represents the energy required to inform the network that the addressed node is non-functional, and hence, not available as a relay for the corresponding cluster nodes. Subsequently, the relevant nodes delete this inoperable node from their routing table and replace it with other appropriate ones.

In the next scenario involving data retrieval from a rank 2 node, where the node in the first rank is also non-functional, the energy consumption of the ACK approach outweighs

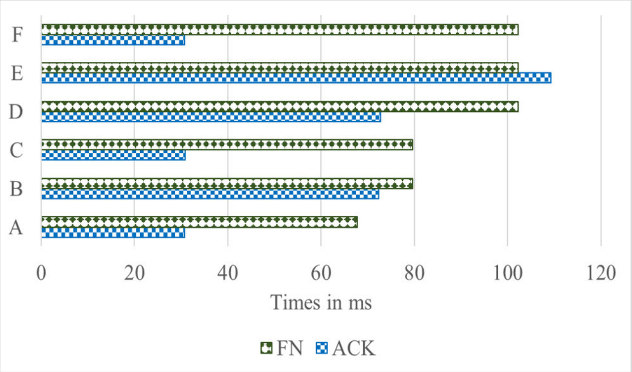

FIGURE 7. Time that is needed to detect one node's failure and update the network using the fog-based and ACK-based failure repair protocols.

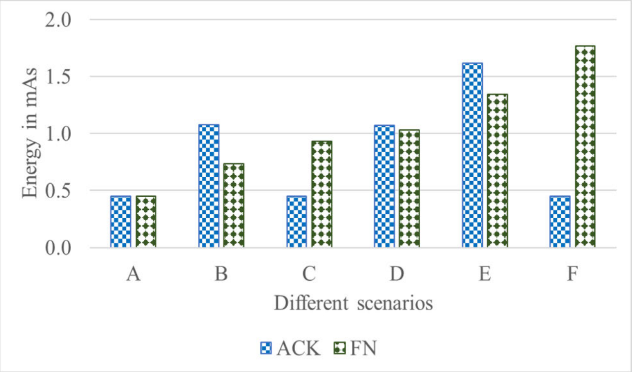

FIGURE 8. Energy that is consumed by the battery-powered cluster nodes after the failure detection and update of the network.

that of the FN approach. This is because an alternative route through another rank 1 node is necessary for data forwarding in the ACK approach, leading to increased energy consumption due to the transmission of the wake-up signal by the cluster nodes. In this error configuration, the fog-based approach prevails, as the battery-operated fog node handles the transmission of the energy-consuming wake-up packet.

Considering scenario C, the ACK-based approach dominates, as no data is being sent. After waking up the cluster nodes, the fog node sends the command to drop down the faulty nodes from the routing table. In contrast, in the fog-based approach, the fog node first tests the functionality of the rank 1 nodes, only registering afterward that the addressed node in rank 2 is non-functional. Hence, the fog node sends the WuPt and the drop-down command, resulting in increased energy consumption due to the testing and associated responses from rank 1 nodes.

In the scenario D, the energy consumption trends are reversed. Here, a rank 1 node is non-functional, but since the data packet is requested from rank 3, the ACK approach necessitates using an alternative relay in rank 1 to send the information to the fog node, resulting in slightly higher energy consumption. The fog-based approach has a slight advantage in this case, as although the rank 2 and rank 3 nodes are awakened, they return to sleep mode without being queried due to the detection of the error in rank 1.

The results for scenario E measurements show that in the case of a rank 2 error when querying data from a rank 3 node, the ACK approach is inferior. Examining the energy consumption of the cluster nodes explains this, as two relays need to be individually awakened by the cluster nodes to forward the information, resulting in increased energy consumption. In this scenario, the FN approach has a better balance, as the cluster nodes do not need to individually consume energy to wake up the relays.

In the last scenario, where the error lies in rank 3, the ACK approach clearly has lower energy consumption. In this case, only network update occurs without data reception. In the reactive FN-oriented approach, the rank 1 and rank 2 nodes are initially awakened and tested, leading to significant energy requirements. Thus, the FN approach exhibits its greatest weakness in the case of a rank 3 node failure.

Upon synthesizing the gathered insights, the determination of when to use each of the two approaches remains open for consideration. Examining the provided graph in Fig. 9 clarifies this aspect. A scenario involving the analysis of 1.000 data packet transmissions, with varying rates of transmission errors, was considered. The objective was to identify the threshold at which each strategy demonstrates its advantages. Based on the results, it can be concluded that the ACK approach becomes advantageous at approximately 314 transmission errors. Beyond this threshold, the additional transmission of acknowledgment messages no longer outweighs the energy consumption of the ACK approach, as the waiting, sending, receiving, and forwarding of test messages initiated by the fog-based approach in error scenarios becomes more energy-intensive. Contrarily, when the network is not too faulty (transmission error less than 314), the sending of additional ACK packet by each node in the routing process in ACK-based failure repair is much more energy-consuming than the energy consumed by the failure repair process in the fog-oriented approach since this latter will use its additional messages only a few times.

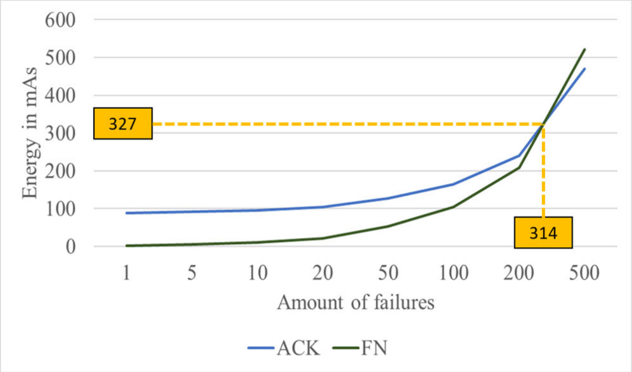

FIGURE 9. Energy consumption depending on the number of failures per 1.000 transmissions.

When examining the timeline as shown in Fig. 10, a distinct shift in the extra time needed becomes evident. The proactive ACK approach outperforms the reactive fog-based approach from a 74 errors onward, as sending and receiving ACK packets introduces additional time, affecting the fog-based approach's efficiency. Beyond this threshold, the fog-based approach requires more time for error detection and network updates.

Regarding Fig. 9, it is clear that the energy advantage of the fog-based approach comes into play below an error rate of 31.4%. However, if the error probability increases, the advantage of the ACK-based approach shifts only slightly. The advantage changes significantly in terms of the time required. Fig.10 shows, that the advantage of the ACK-based approach already predominates at a probability of occurrence of 7.4%. It becomes significantly clearer from a percentage of occurrence of 20%, i.e. as can be seen in Fig. 10 from an occurrence of 200 errors in 1.000 data transmissions.

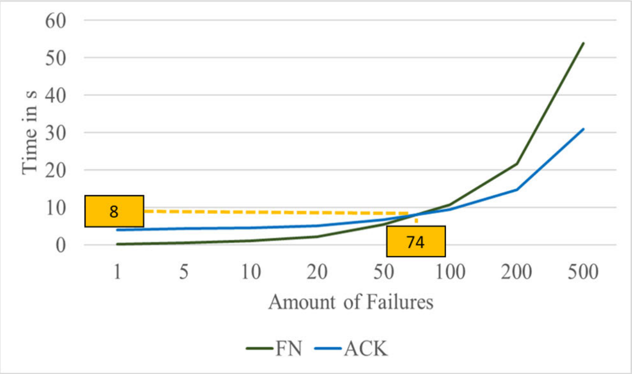

FIGURE 10. Required failure detection and repair time depending on the number of failures per 1.000 transmissions.

Summing up the insights reveals a constant trade-off between energy consumption and time efficiency in network strategies. Achieving an optimal balance is crucial for effective performance. In the absence of a faulty network, we opt for the reactive fog-based approach, as the proactive ACK approach consumes energy and wastes time for ACKs. If failures are infrequent and unlikely to occur, the fog-based approach is more suitable. The proactive ACK-based approach prevails when there is high probability of encountering many node failures in time-sensitive applications.

In practice, we can opt for both strategies in the same network. We start with the fog-based failure repair. The nodes' error rate is computed continually. Once the error rate reaches 30%, the failure repair strategy switches to the ACK-based approach.

V. CONCLUSION

In this paper, we have proposed a fog-based bootstrapping and failure repair protocols, in addition to another ACK-based failure repair protocol for clustered and heterogeneous Wake-up Receiver based WSNs. In our proposed contribution, the sink and fog nodes have more capabilities in terms of transmission power and energy budget than the individual cluster nodes. Each sensor node is equipped with a WuRx and battery powered. In our proposed approach, when establishing a network, a cluster is made up of a fog node and a set of WuRx-based sensor nodes having different ranks depending on their distance in terms of the number of hops separating them from their corresponding fog node.

The schemes we have presented for setting up and troubleshooting a WSN are not necessarily tied to a specific use case. Even if we have considered an indoor network in this work, the approaches can be used in a variety of different scenarios. Potential areas include the monitoring of processes in the industrial fabrication sector. Sending the measurement data to the control system enables prompt implementation of corrective actions in real-time when critical deviations occur. Another potential application example in the medical environment would be the monitoring of elderly or sick people at home.

In the bootstrapping phase, the fog node constructs its cluster by varying the transmitting power of its WuPt in a way that enables the sensor nodes receiving it to wake up and join the network in the order of their ranks. Upon completion of the setup phase, the fog node, being the sole interface of the cluster, possesses all the routes between the individual cluster nodes.

Based on this network, strategies were developed to enable the restoration of network functionality in the event of node's failure. A reactive fog-based approach, where communication coordination and failure repairs originate from the fog node, was compared with a proactive approach based on receiving acknowledgments. In the acknowledgment-based approach, cluster nodes send a sequence to confirm the receipt of the data packet. The comparison between both protocols considered the duration required to detect the error and update network routes. Additionally, the energy consumption incurred when one of the strategies for data communication error recovery is employed was examined.

In summary, experimental results have shown that there is a trade-off between key attributes of latency and energy efficiency. The fog-oriented approach takes generally slightly longer compared to ACK-based approach, but saves more energy when the error transmission rate does not exceed 31%. In conclusion, if the network is error-prone, the proactive ACK-based approach is recommended as it quickly and efficiently restores the network. However, if rare errors are expected, it is advisable to choose the fog-oriented approach.

Further work based on the results and the strategies presented here could consist of further improving the energy efficiency of the network. This could be done, for example, by linking the WuRx with energy harvesting solutions. Machine learning also offers a future starting point for further studies. Machine learning algorithms for fault detection could be used, for example, to predict errors or problems in data transmission. In addition to the battery status, parameters as bandwidth constraints and quality of service (QoS) can also be used to analyse the performance in order to make the network more efficient. In addition, the application can be expanded for further studies. This aspects include the expansion of cluster members and the possibility of mobile sensor nodes.

REFERENCES

[1] D. S. Ibrahim, A. F. Mahdi, and Q. M. Yas, "Challenges and issues for wireless sensor networks: A survey," *J. Glob. Sci. Res.*, vol. 6, no. 1, pp. 1079–1097, 2021.

[2] A. Bagwari, G. S. Tomar, J. Bagwari, J. L. V. Barbosa, and M. K. Sastry, *Advanced Wireless Communication and Sensor Networks: Applications and Simulations*. Boca Raton, FL, USA: CRC Press, 2023.

[3] R. Chéour, M. W. Jmal, A. Lay-Ekuakille, F. Derbel, O. Kanoun, and M. Abid, "Choice of efficient simulator tool for wireless sensor networks," in *Proc. IEEE Int. Workshop Meas. Netw. (M&N)*, Oct. 2013, pp. 210–213.

[4] I. Ketata, S. Ouerghemmi, A. Fakhfakh, and F. Derbel, "Design and implementation of low noise amplifier operating at 868 MHz for duty cycled wake-up receiver front-end," *Electronics*, vol. 11, no. 19, p. 3235, Oct. 2022.

[5] R. Chéour, F. Derbel, O. Kanoun, and M. Abid, "Wireless sensor networks with power management for low energy consumption," in *Proc. 10th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2013, pp. 1–6.

[6] S. Khriji, Y. Benbelgacem, R. Chéour, D. E. Houssaini, and O. Kanoun, "Design and implementation of a cloud-based event-driven architecture for real-time data processing in wireless sensor networks," *J. Supercomput.*, vol. 78, no. 3, pp. 3374–3401, Feb. 2022.

[7] O. Kanoun, S. Khriji, S. Naifar, S. Bradai, G. Bouattour, A. Bouhamed, D. El Houssaini, and C. Viehweger, "Prospects of wireless energy-aware sensors for smart factories in the Industry 4.0 era," *Electronics*, vol. 10, no. 23, p. 2929, Nov. 2021.

[8] R. Chéour, M. W. Jmal, S. Khriji, D. El Houssaini, C. Trigona, M. Abid, and O. Kanoun, "Towards hybrid energy-efficient power management in wireless sensor networks," *Sensors*, vol. 22, no. 1, p. 301, Dec. 2021.

[9] O. Kanoun, T. Keutel, C. Viehweger, X. Zhao, S. Bradai, S. Naifar, C. Trigona, B. Kallel, I. Chaour, G. Bouattour, S. Khriji, D. E. Houssaini, and M. Götz, "Next generation wireless energy aware sensors for Internet of Things: A review," in *Proc. 15th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2018, pp. 1–6.

[10] A. Guidara and F. Derbel, "A real-time indoor localization platform based on wireless sensor networks," in *Proc. IEEE 12th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2015, pp. 1–8.

[11] E. Zaraket, N. M. Murad, S. S. Yazdani, L. Rajaoarisoa, and B. Ravelo, "An overview on low energy wake-up radio technology: Active and passive circuits associated with MAC and routing protocols," *J. Netw. Comput. Appl.*, vol. 190, Sep. 2021, Art. no. 103140.

[12] A. K. W. Chee, "Quantitative dopant profiling by energy filtering in the scanning electron microscope," *IEEE Trans. Device Mater. Rel.*, vol. 16, no. 2, pp. 138–148, Jun. 2016.

[13] A. K. W. Chee, "Principles of high-resolution dopant profiling in the scanning helium ion microscope, image widths, and surface band bending," *IEEE Trans. Electron Devices*, vol. 66, no. 11, pp. 4883–4887, Nov. 2019.

[14] J. Aranda, M. Schölzel, D. Mendez, and H. Carrillo, "Multimodal wireless sensor networks based on wake-up radio receivers: An analytical model for energy consumption," *Revista Facultad de Ingeniería Universidad de Antioquia*, vol. 91, pp. 113–124, Jun. 2019.

[15] J. Parkash, S. Cheerla, P. Shanmugaraja, M. Manohara, and P. Madan, "Receiver-based data communication for wireless sensor networks," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, May 2023, pp. 1–7.

[16] R. Fromm, L. Schott, and F. Derbel, "An efficient low-power wake-up receiver architecture for power saving for transmitter and receiver communications," in *Proc. 10th Int. Conf. Sensor Netw.*, Feb. 2021, pp. 61–68.

[17] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Record. (WCNC)*, Mar. 2002, pp. 350–355.

[18] M. Vodel, M. Lippmann, and W. Hardt, "WRTA–wake-up-receiver optimised routing and topology optimisation approach," in *Proc. 12th Int. Conf. ITS Telecommun.*, Nov. 2012, pp. 329–334.

[19] S. Basagni, C. Petrioli, and D. Spenza, "CTP-WUR: The collection tree protocol in wake-up radio WSNs for critical applications," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2016, pp. 1–6.

[20] Texas Instruments. (2013). *MSP430G2253: Datasheet: Mixed Signal Microcontroller*. Accessed: Jan. 2024. [Online]. Available: https://www.ti.com/lit/ds/symlink/msp430g2553.pdf

[21] G. U. Gamm, M. Sippel, M. Kostic, and L. M. Reindl, "Low power wake-up receiver for wireless sensor nodes," in *Proc. 6th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, Dec. 2010, pp. 121–126.

[22] STMicroelectronics. (2021). *SPSGRFC-868: Datasheet: Sub-GHz-Module Sub-1 GHz*. Accessed: Jan. 2024. [Online]. Available: https://www.st.com/resource/en/datasheet/spirit1.pdf

[23] M. Weber, G. Fersi, R. Fromm, and F. Derbel, "Wake-up receiver-based routing for clustered multihop wireless sensor networks," *Sensors*, vol. 22, no. 9, p. 3254, Apr. 2022.

[24] M. Weber, G. Fersi, and F. Derbel, "Wake-up receiver based routing protocol for indoor wireless sensor networks," in *Proc. 18th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2022, pp. 289–295.

MAXIMILIAN WEBER was born in Heidelberg. He received the B.S. degree in medical engineering from the Faculty of Information Technology, HS Mannheim, Germany, in 2015, and the M.S. degree in electrical engineering and management from the Faculty of Engineering, HTWK Leipzig, Germany, in 2019. He is currently pursuing the Ph.D. degree in electrical engineering under the supervision of Prof. Faouzi Derbel. His current research interest includes the study of routing and communication protocols for wake-up receiver based wireless sensor networks.

GHOFRANE FERSI received the degree in computer science engineering and the M.S. degree (DEA) in systems of information and dedicated new technologies (NTSID) from the National School of Engineers of Sfax (ENIS), Tunisia, in 2008 and 2009, respectively, and the Ph.D. degree in computer science from the Research Laboratory of Development and Control of Distributed Applications (ReDCAD), ENIS, University of Sfax, in 2014. Since 2013, she has been an Assistant Professor with the Department of Computer Science, University of Sousse, Tunisia. Since 2016, she has been an Associate Professor. She is currently an Assistant Professor with ENIS, University of Sfax. Her current research interests include fog computing, security, the Internet of Things protocols, architectures, and applications.

FAOUZI DERBEL (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from the Technical University of Munich, Germany, in 1995, and the Ph.D. degree from the University of the Bundeswehr at Munich, Munich, in 2001. From 2000 to 2012, he was in different positions in the industrial area, such as the Strategic Product Manager and a Systems Engineer, responsible for wireless and future technologies with Siemens Building Technologies, Munich, Germany; and the Head of the Research and Development, Siemens Building Technologies, Mühlhausen, Germany, and QUNDIS Advanced Measuring Solutions, Germany. He is currently a Professor of smart diagnostics and online monitoring with Leipzig University of Applied Sciences. His research interests include power aware design of wireless sensor networks and sensor systems with smart signal processing and state estimation of machines and smart grids. He is a member of ETSI ERM TG28 dealing with short range wireless devices within European Telecommunication and Standardization Institute, and the Technical Group Smart Cities, "Deutsche Kommission Elektrotechnik, Elektronik und Informationstechnik im DIN und VDE." In 2006, he won together with his team the "Product Innovation Award" within company Siemens AG on worldwide level; and the "Product Success Story Award" of Texas Instruments, in 2009. He is the Editor-in-Chief of the *Advances in Systems, Signals and Devices* (ASSD), Special Issue on Communication and signal processing and power systems and smart energies, De Gruyter Verlag, Germany, and holds many awards and patents.

• • •