# Quality-of-Service-Based Minimal Latency Routing for Wireless Networks

Fjolla Ademaj, *Member, IEEE*, and Hans-Peter Bernhard, *Senior Member, IEEE*

***Abstract*—Minimized and nearly deterministic end-to-end latency facilitates real-time data acquisition and actuator control. In addition, defined latency is an integral part of quality-oriented service in order to get closer to the reliability of wired networks and at the same time takes advantage of wireless networking. This article introduces a QoS routing protocol capable of balancing power consumption between wireless sensor and actuator nodes while minimizing end-to-end latency. We introduce a time-division multiple access scheme in the routed wireless network to enable defined latency, and in addition, it improves the energy efficiency by avoiding collisions, which eliminates time and energy consuming retries. Our novel routing method allows latency and round-trip times to be calculated in advance. We implement a demonstrator and show experimental results of a wireless sensor network with our proposed routing scheme.**

***Index Terms*—Deterministic latency, energy balance, multihop communication, quality of service (QoS), routing, wireless sensor networks (WSNs).**

## I. Introduction

THE DEVELOPMENT of wireless sensor networks (WSNs) and wireless sensor and actuator networks (WSANs) in industrial applications has grown extensively in recent years, leading, among others, to the emergence of industrial WSNs (IWSNs) and industrial WSANs (IWSANs) [1]. The low complexity, simple deployment, low cost, and flexibility are the main reasons why IWSN is a promising technology in industrial environments. In these networks, the main target is to achieve an autonomous operation that involves less human interaction and provides stable connectivity among various systems and subsystems within factories. However, the harsh conditions in industrial environments due to metallic structures, multipath propagation, equipment noise, and electromagnetic interference impose many challenges on such wireless networks. These imperfections in the wireless medium result in performance degradation and restrict the quality of service (QoS) for IWSN applications.

In comparison to the general purpose WSN, the IWSNs are generally planned network topologies with fixed location of nodes or low node mobility. For the different applications that find use in such scenarios, tighter requirements apply that necessitate optimizations of the specific network characteristics. In the past years, the IEEE 802.15.4 standard [2] has been adopted to meet the requirements of industrial applications. A number of different technologies have emerged that provide solutions for low-data-rate low-power IWSNs, such as WirelessHART, ISA SP100.11a, and WIA-PA. In particular, the focus has been on the design and development of medium access control (MAC) protocols that improve the network performance [3]. The main performance metrics for IWSNs are *data reliability*, *synchronicity* of data samples across different sensor nodes (sensor data must be sampled synchronously on all nodes and additionally the data must be transmitted synchronously), *determinism,* and *low latency*. Additionally, in many industrial applications, it is essential to design low-power sensors that do not need battery replacement over their lifetimes. This requires efficiency in terms of energy consumption when designing communication protocols. In more complex multihop networks, routing algorithms become crucial as they determine routes and channels for the transmission as well as the redundancy policy and, in this manner, directly affect the aforementioned performance metrics. Hence, a careful design of the routing protocols, especially in the case of stringent requirements that apply for IWSNs, is necessary. In this article, we introduce a new routing algorithm named QoS-based minimum latency routing (QMLR) to provide a defined routing under QoS constraints. The QMLR algorithm is designed for multihop networks in IWSN that are based on time-division multiple access (TDMA). It defines the routes depending on a given QoS per link, minimizes the latency and at the same time minimizes and balances the energy consumption at sensor nodes.

### *A. Related Work*

Differently from WSNs, the network structure in IWSNs is typically centralized. Centralized management is used to allow better control of the network operation and also to simplify

Manuscript received July 24, 2020; revised November 30, 2020, January 29, 2021, and March 23, 2021; accepted April 1, 2021. Date of publication April 7, 2021; date of current version December 6, 2021. This work was supported in part by the InSecTT project[1], by the ECSEL Joint Undertaking (JU) under Grant 876038, and by the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. Paper no. TII-20-3559. *(Corresponding author: Hans-Peter Bernhard.)*

Fjolla Ademaj is with Silicon Austria Labs GmbH, 8010 Graz, Austria (e-mail: fjolla.ademaj@silicon-austria.com).

Hans-Peter Bernhard is with Silicon Austria Labs GmbH, 8010 Graz, Austria, and also with the Institute for Communications Engineering and RF-Systems, Johannes Kepler University Linz, 4040 Linz, Austria (e-mail: h.p.bernhard@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3071596.

Digital Object Identifier 10.1109/TII.2021.3071596

[1] [Online]. Available: https://www.insectt.eu/

the hardware and software of the nodes. In such a centralized topology, the base station (BS) or gateway is the central device involved in most network activities. While there exist approaches of decentralized routing for IWSNs such as in [4] and [5], they perform poorly under latency, resulting in higher latency as compared to centralized approaches as the one in [6] and, therefore, are not suitable for low-latency communication. A common approach in centralized routing schemes is increase of reliability by introducing path redundancy. Such approach is implemented via graph routing. There are many graph routing algorithms developed in the past decade [7]–[9] targeting optimization of various network characteristics. Jindong *et al.* [7] select the shortest path within each hop using breadth first tree and uses received signal level to select neighbors. This algorithm provides a higher reliability compared to single-hop routing; however, it does not provide deterministic latency nor it is optimized to be energy efficient. Zhang *et al.* [9] present an algorithm that reduces the energy consumption of a single route and maximizes the network lifetime at the same time. However, reliability under link quality and deterministic latency are not in the scope of this article. Differently from the aforementioned approaches, in recent work [10]–[12], machine learning and reinforcement learning approaches have been introduced to find optimal routing. While these algorithms iteratively update the weighted cost in finding optimal routes in a graph, they show to be problematic in case of network fluctuations and failures due to the learning adaptation methodology, thus increasing energy consumption and reducing network lifetime significantly. Unlike the aforementioned algorithms that use specific criteria to find optimal routes, there exist flooding protocols that distribute messages to every part of the graph. Yu *et al.* [13] and Ferrari *et al.* [14] utilize a flooding algorithm in IWSNs, which enables a very short latency. However, a major drawback that comes from flooding is the increase of the overall communication load in a sensor network, resulting in a dramatic increase in the total power consumption. Another disadvantage of this algorithm is the vulnerability to packet insertion, replay, and denial-of-service attacks. These security issues exclude the algorithm from many applications. Another important aspect in multihop communications is traffic distribution and congestion control that is known as load balancing. Li *et al.* [15] present a load-balancing algorithm that can adjust the load over multihops. A threshold sharing algorithm is applied to split the packets into multiple segments that are delivered via multihops to the destination. While it is beneficial for increasing reliability and robustness in a network, it does not fulfill the requirements for a deterministic latency. In this context, Samara and Hosseini [16] have extended the packet header by additional energy budget information where the routing is packet based and not centrally managed. Other approaches like in [17] and [18] consider heuristic functions to balance the routing among the relaying nodes. A routing algorithm that minimizes the network latency and balances the energy consumption of the nodes has been introduced in [19]. In comparison to the approach of low-energy adaptive clustering hierarchy algorithm [20], [21], where the wireless network is clustered to achieve energy optimization with randomly selected cluster heads, Bernhard *et al.* [19] do not consider random cluster heads. The route metrics are compared from an energy perspective only within equal hop distances. Selecting the cluster heads randomly is not able to fulfill the requirement for a deterministic latency.

In contrast to most other methods, Bernhard *et al.* [19] introduce a routing algorithm that integrates the MAC layer as a part of the routing. Hence, it is possible to apply a synchronized TDMA scheme in more than one layer so that the overall synchronization is guaranteed. Therefore, the latency in the routed network is nearly deterministic for a given number of hops and TDMA timing. Although this algorithm considers the energy consumption and load balancing, it does not treat the QoS aspect of the routed links. The QoS aspect is crucial in providing a guaranteed end-to-end performance and it is especially demanded for real-time applications. While most of the routing algorithms focus on energy consumption and reliability, they ignore the real-time aspect [22] and assume that the traffic speed is sufficient to meet the required QoS.

## B. Contribution and Organization

In this article, we present the QMLR routing algorithm that first reduces the energy consumption caused by collisions, overhearing, control packet overhead, idle, and periodic listening, second, balances the load at nodes to avoid congestion where the energy state of individual nodes is used as control parameter to optimize energy consumption in order to avoid energy shortages caused by the network load, and finally, guarantees a minimum communication quality. We adapt the energy- and power-efficient synchronous sensor network (EPhESOS) protocol [23] to set up a collision-free routing domain to avoid the aforementioned shortcomings. The routing algorithm enables a defined latency concept motivated by multihop synchronization [24] and it links the layer-based synchronization to tree-structured routing. The proposed QMLR algorithm consists of two stages: 1) the network discovery phase and 2) the node communication phase. In the first phase, the network is organized in layers, where relay nodes that meet QoS criteria are determined. This layer-based structure is hierarchically constructed starting from the central BS to the furthest nodes in the network. In the second phase, after the network structure is known to the base station (BS), routing takes place. Here, the route metrics are compared from an energy perspective only within equal hop distances. In this way both the QoS perspective and the energy efficiency perspective are integrated. Additionally, we give an answer to the question that how many hops are acceptable or necessary to guarantee a defined level of QoS. Another aspect is the practical implementation and demonstration, which is often missing in the literature. We believe that this is crucial to understand better and further enhance routing protocols. To address these issues, we additionally focus on implementation aspects of the proposed routing algorithm. A full featured implementation in a sensor network is presented as proof of concept to validate the feasibility of our approach with commercial of-the-shelf devices. Moreover, we evaluate the network discovery and routing phase with various network settings and present one example for a chosen network topology.

The rest of this article is organized as follows. In Section II, we describe the MAC layer, versatile enough to allow QoS-oriented routing. The system model and the routing algorithm are presented in Section III. An overview of the behavior of the proposed routing algorithm on the energy consumption is given in Section IV. Section V analyzes various network parameters for the two respective routing phases by means of simulations. In Section VI, the behavior of the proposed algorithm is demonstrated by an actual implementation of a wireless sensor network, followed by a discussion on measurement results. Finally, Section VII concludes this article.

## II. Media Access of EPhESOS for QoS-Oriented Routing

Interoperability between layers is crucial for the design of routing algorithms. For example, tasks executed in higher layers, such as routing, depend on timing and organization of the MAC-layer. Since we are considering deterministic protocols, WirelessHART and ISA100.11a [25], [26] are candidates for the implementation of a deterministic routing. However, these protocols are not sufficiently flexible and energy efficient to address a QoS communication and routing concept. Therefore, in our work, we adopt the EPhESOS MAC layer protocol [23] and extend it with the routing concept. This protocol provides reliable synchronization to reduce transceiver uptime, in order to eliminate the need of listening to a beacon before data transmission by the node.

The MAC-layer of EPhESOS supports the standard logical link control layer introduced by the open systems interconnection model with handshaking and flow control. Additionally, it provides energy efficient and synchronized data flow from nodes to a centralized BS. This protocol is TDMA based and is optimized for low-power sensor network applications. A centralized clock is supplying the network for synchronizing communications, measurements, and triggering of events. Another advantage of EPhESOS is the energy-efficient handshaking, which allows very short duty cycles of nodes. In long-lasting standby or sleep periods of the nodes, energy-efficient measures like reducing the sleep clock frequency and switching OFFthe transceiver units can be applied. The time granularity of the wake-up mechanism is defined by the sleep clock. To overcome the resulting problems, concepts for synchronization and determining the wake-up interval are adapted according to the work in [27] and [28], which enables synchronicity of ±1 clock period. As for WirelessHART or ISA100.11a, superframes (SFs) are defined in a very simple way, where an SF begins with a beacon and lasts until the next consecutive beacon. In EPhESOS, after the beacon, a TDMA scheme begins, dividing the SF into $K$ equal time slots (TSs), as illustrated in Fig. 1. This seems inflexible, but if the protocol is deterministic, the maximum number of nodes must be constant for each SF to achieve a constant QoS service, i.e., bit rate, data throughput, packet loss, latency, and jitter of the communication delay. In general, the bit rate of the BS is constant and the duration is equal for the TSs assigned to individual nodes. This enables an equal and constant QoS only limited by the physical (PHY)-layer properties. Thus, the MAC layer and above can only influence the QoS parameters of data throughput per node and end-to-end latency to the BS. Many protocols for indoor wireless applications are based on a star topology and have no routing function. Even when considering the star topology, each node can have an individual time window length, which allows for an individual QoS based on the specific requirements of the node. Obviously, the total length of an SF is limited, but within this limit an individual QoS is possible. When considering network topologies with nodes that are capable to act as a relay between the BS and other nodes in the network (multihop routing), then the total length of an SF needs to be adapted. If we consider routing with equal TSs, the QoS of the routed nodes is limited and is indirectly proportional to the number of nodes routed by a node. This means that the QoS of nodes with more than one hop to the BS drops automatically. Therefore, it is necessary to extend the TS of relay nodes to allow a constant QoS for links via nodes acting as a relay. If nodes need a lower QoS, it is recommended to use smaller TSs in favor of a network-wide throughput. This means that the MAC layer must provide the ability to use a variable TS length according to the QoS required at particular nodes. We introduce an SF timing that is valid all over the routed network. Therefore, we can avoid collisions within the network. As depicted in Fig. 1, the routing nodes pass the beacon to further nodes or hops in their own TS. In this example, node $N_1$ acts as a relay for node $N_{16}$, where $N_1$ passes the information to $N_{16}$ and receives messages from $N_{16}$. In the following SF, the message of $N_{16}$ and $N_1$ is sent to $B$ in the TS of $N_1$. For more details on the timing, consider the work in [19].

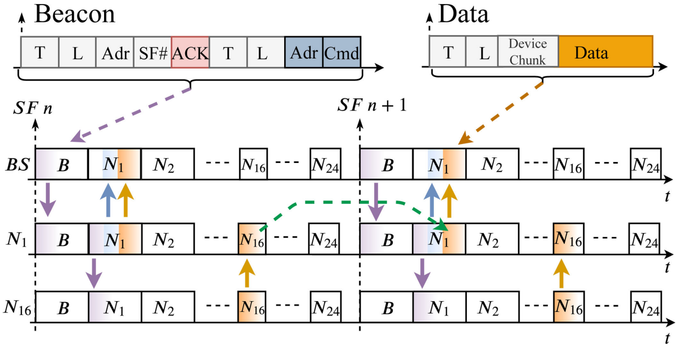


Fig. 1. SF structure in EPhESOS for TDMA.

To provide as much flexibility as possible, we employ a lightweight SF with a few unconditional requirements. Each SF has a counter value (SF#) generated by the BS and sent to the sensor network by the beacon. The beacon contains an address that identifies the sending node. Each listening node synchronizes the local clock with the BS. In addition, a bit field at the end of the beacon represents the confirmation information. Each acknowledge/not-acknowledge is represented by a bit assigned to a TS. In the employed EPhESOS protocol family, the energy consumption of individual nodes is minimized while the complicated processing steps are avoided and redundancies are eliminated. A highly effective, scalable datagram structure is used to encapsulate data and commands. The EPhESOS datagram introduces a highly flexible and versatile data model, where EPhESOS datagrams are byte sequences, so-called chunks, with a 2-B header representing a type, length, value structure [23].

## III. System Model

We consider a network comprising of a single BS and $N_k$ nodes that have a direct communication link with the BS. The network elements are a part of the set $\mathbf{R} = \{N_0, N_1, \ldots, N_K\}$, with BS assigned to element $N_0$. In this topology, the BS manages all other nodes. The main objective of our network design is to optimize routing with respect to various parameters by setting the following QoS criteria:

1) a distributed algorithm for the network discovery;
2) centrally planned routes;
3) optimized latency for topology or least number of hops;
4) defined PER for individual nodes and complete routes;
5) energy-balanced routing;
6) minimal queuing on the routing/relay nodes;
7) collision-free access scheme.

In order to optimize the routing, the routes can be balanced considering any of the aforementioned criteria as well as according to routes' individual QoS requirements. We develop a routing algorithm that is based on a structured network connection concept organized in layers and consists of two stages. In the first stage, the network is organized in layers depending on the quality of each link (the QoS criteria applied). Each layer then consists of nodes that are able to reroute to the next layer. We refer to these nodes as *relay nodes*. All calculations to find the routes are performed centrally at the BS, since, in practice, the BS is connected to the mains power supply. In the second stage, the network routes are determined based on the employed optimization criteria. In this way, the network topology follows a hierarchical tree structure, starting with the connection of BS and the first layer of *relay* nodes. Next, this hierarchical tree topology continues with a layer-to-layer connectivity considering only those nodes that meet a defined QoS criterion. In the following, we provide a detailed description of the two-stage routing algorithm.

### A. Network Discovery

We consider a single BS, $N_0$, located at the center of the operating area a set of nodes $\{N_1, \ldots, N_K\}$ distributed within the operating area. Since there is a high probability that not all nodes can have a direct link to the BS with a guaranteed QoS, the relaying concept is introduced. In the first stage of the routing, referred to as the network discovery stage, we define signal-to-noise ratio (SNR) as the quality measure and apply it to determine the nodes with the best link quality. These nodes then can serve as a relay between the BS and the rest of the nodes in the network. The discovery of the network starts from the central BS. As a QoS parameter, we select the SNR. By setting an SNR threshold, denoted with $\gamma$, we apply the following criteria to the network:

$$\langle \mathbf{R}_i \rangle = P\left(\mathrm{SNR} > \gamma\right). \tag{1}$$

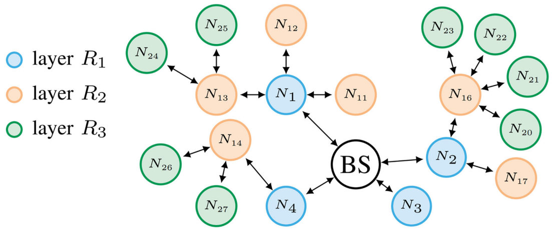


Fig. 2. Link-quality-based layers of network nodes.

**Algorithm 1:** Distributed Network Discovery at BS.

```
1: function GETSTRUCTURE$\mathbf{R}$
2:    $\mathbf{R}_f = \mathbf{R} \backslash N_0$
3:    $n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K \leftarrow$ FINDMAPS$\mathbf{R}_f, N_0, 0, \emptyset$
4:    return $n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K$
```

As a result, in the first step, the set of nodes $\mathbf{R}_i$ that meets this criterion compose the first layer of relay nodes. We denote these nodes as $\mathbf{R}_1 \subset \mathbf{R}$. All other nodes that are not directly reachable from the BS or have a very poor communication link quality are part of the set $\mathbf{R} \backslash \mathbf{R}_1$, i.e., $\mathbf{R}_2 \cup \mathbf{R}_3 \ldots \cup \mathbf{R}_n$.

In the next step, relay nodes of the first layer $\mathbf{R}_1$ act as relay and send out beacons to the rest of the network. By utilizing the same link quality criteria from (1), the second layer of nodes $\mathbf{R}_2$, which meet this criterion, is established. Nodes in $\mathbf{R}_2$ need one intermediate hop to reach the central BS. This procedure is repeated in a recursive fashion until the network discovery phase is completed. The aforementioned organization in layers can be expressed as

$$\mathbf{R}_{n+1} = \left(\mathbf{R} \backslash \cup_{m=1}^{n} \mathbf{R}_m\right) \text{ respond to } \quad \mathbf{R}_n \tag{2}$$

where *respond to* selects those nodes, contained in the set of the left-hand side, which respond with a certain SNR (1) to some or all node elements of the set $\mathbf{R}_n$. An example of a layered network structure is illustrated in Fig. 2. It can be noticed that the connections start from the central BS and follow a hierarchical tree structure. Each layer is distinguished by a different color, where nodes marked in blue color belong to the fist relay layer and serve as relay nodes for the nodes marked in orange color (layer 2). The mathematical description of (2) for analyzing the network is implemented as a recursive algorithm having two parts, the BS introduced in Algorithm 1, and the node part introduced in Algorithm 2. According to the previous description, the set of all available nodes $\mathbf{R}$ is split up into disjoint subsets $\mathbf{R}^p = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{n_{\max}}\}$, denoting the layers. For each layer $\mathbf{R}_n$, the subscript $n$ indicates the routing distance to the BS.

Algorithm 1 presents the starting function of the routing executed by the BS. The function GETSTRUCTURE is utilized with the parameter $\mathbf{R}$, representing the set of all available nodes. Initially, the BS is removed from the set $\mathbf{R}$ (see *step 2* in Algorithm 1). In this way, all nodes not assigned to a specific layer are part of the set $\mathbf{R}_f$. Next, the function FINDMAPS is executed. The purpose of this function is to discover the network, i.e., determine relay nodes in a layered structure, and is executed on the

**Algorithm 2:** Distributed Network Discovery at Sensor Node.

```
function FINDMAPS $\mathbf{R}_f, N_n, n, N_{n-1}$
  $N_n$.relay $\leftarrow$ $true$
  SENDTO ($\mathbf{R}_f$, RESPONDSTO$N_n$)
  for all $N_c \in \mathbf{R}_f$ do
    if $N_c$ respondto $N_n$ then
      $\mathbf{R}_{n+1} \leftarrow \mathbf{R}_{n+1} \cup N_c$
      $\mathbf{F}_{N_n} \leftarrow \mathbf{F}_{N_n} \cup N_c$
    $\mathbf{R}_f \leftarrow \mathbf{R}_f \backslash \mathbf{R}_{n+1}$
    $n_{\max} \leftarrow n$
    for all $N_c \in \mathbf{R}_{n+1} \wedge \neg(N_c$.relay) do
      SENDTO ($N_c$, FINDMAPS$\mathbf{R}_f, N_c, n+1, N_n$)
    for all $N_c \in \mathbf{R}_{n+1} \wedge \neg(N_c$.relay) do
      $n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K \cup \leftarrow$ RECEIVEDFROM ($N_c$)
    return $n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K$
function RESPONDSTO$N_{n-1}$
  $\mathbf{B}_{N_n} \leftarrow \mathbf{B}_{N_n} \cup N_{n-1}$
  return
```

node side. The stepwise procedure of FINDMAPS is presented in Algorithm 2.

In the following, we describe the distributed algorithm according to the call sequence. With the initialization of FINDMAPS, first the command SENDTO is executed. If the function is executed for the first time, a communication request is sent as a broadcast to all free nodes in the network. Let us consider that node $N_n$ is executing this command. In response to node $N_n$, all not connected nodes are sending a connect message via the SENDTO command and get the assigned TS and a synchronization time information, which is implemented in the EPhESOS protocol. It is important to note that a certain amount of time is defined as a waiting period for the responses from the reached nodes from $N_n$. This time is set to a minimum of two wake up intervals[2] (W), i.e., $\tau_{\min} = 2$wake up interval (W). In this way, actual node $N_n$ keeps track of the newly detected nodes if $N_c$ *respond to* $N_n$ results $true$, which adds this node to the set $\mathbf{R}_{n+1}$. Additionally, each detected node is added to the forward list of the node $N_n$, i.e., $\mathbf{F}_{N_n}$. Next, the set of free nodes is reduced by removing $\mathbf{R}_{n+1}$ from $\mathbf{R}_f$. Since the algorithm is recursive, the FINDMAPS command is sent to all nodes of $\mathbf{R}_{n+1}$ if those nodes are not a relay, and therefore, FINDMAPS has not already been processed.

As the algorithm is recursive and distributed, the node $N_k$ waits for all replies of $\mathbf{R}_{n+1}$ nodes. The replies are the vectors of $\mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K$ of the recursive path behind each answering node, denoting layers, forward nodes, and backward nodes, respectively. All received vectors are used to build a cumulative union of the found layers, forward and backward vectors. This result is sent to the calling node by the return command. The algorithm stops to do the recursion, if $\mathbf{R}_{n+1}$ is an empty set. It is important to emphasize that all nodes receiving the command RESPONDSTO add the sender $N_{n-1}$ to the backward link of the receiving node, denoted by $\mathbf{B}_{N_n}$. Fig. 3 illsutrates an example after the Algorithm 2 is executed with the set $\{n, \mathbf{R}, \mathbf{F}, \mathbf{B}\}$ for each node.

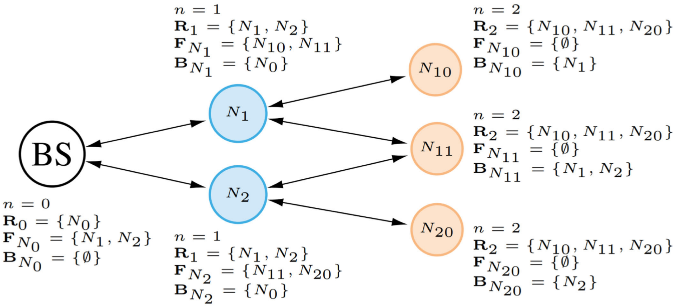

$n = 1$
$\mathbf{R}_1 = \{N_1, N_2\}$
$\mathbf{F}_{N_1} = \{N_{10}, N_{11}\}$
$\mathbf{B}_{N_1} = \{N_0\}$
$n = 2$
$\mathbf{R}_2 = \{N_{10}, N_{11}, N_{20}\}$
$\mathbf{F}_{N_{10}} = \{\emptyset\}$
$\mathbf{B}_{N_{10}} = \{N_1\}$
$N_{10}$
$N_1$
BS
$n = 2$
$\mathbf{R}_2 = \{N_{10}, N_{11}, N_{20}\}$
$\mathbf{F}_{N_{11}} = \{\emptyset\}$
$\mathbf{B}_{N_{11}} = \{N_1, N_2\}$
$N_{11}$
$N_2$
$n = 0$
$\mathbf{R}_0 = \{N_0\}$
$\mathbf{F}_{N_0} = \{N_1, N_2\}$
$\mathbf{B}_{N_0} = \{\emptyset\}$
$N_{20}$
$n = 1$
$\mathbf{R}_1 = \{N_1, N_2\}$
$\mathbf{F}_{N_2} = \{N_{11}, N_{20}\}$
$\mathbf{B}_{N_2} = \{N_0\}$
$n = 2$
$\mathbf{R}_2 = \{N_{10}, N_{11}, N_{20}\}$
$\mathbf{F}_{N_{20}} = \{\emptyset\}$
$\mathbf{B}_{N_{20}} = \{N_2\}$

Fig. 3. Result of Algorithm 2 with $\{n, \mathbf{R}, \mathbf{F}, \mathbf{B}\}$ for each node.

## B. Routing Algorithm

After the network structure is known to the BS, the routing is initiated. The algorithm provides the following key features:

1) centrally planned routing;
2) minimized latency while being nearly deterministic;
3) energy-balanced routing.

Since the BS collects all the data from the nodes, the routing is performed centrally from the BS. The latency is automatically minimized when the network discovery approach described in Section III-A is utilized. Obviously, the latency increases with the number of layers in the network; however, with the QoS threshold, it is ensured that within one hop, the latency is minimal, i.e., the higher the QoS threshold, the lower the latency. Additionally, with the SF structure presented in Section II, the round trip time is limited to one SF improving, thus, the latency within a hop. In such a setting, the maximum latency is known even before the routing is started. Additionally, the routing accounts for a balance in terms of energy. This is achieved by considering the link energy equivalent (LEE) that a relay node is capable to handle when serving other nodes. Algorithm 3 summarizes the steps for the routing phase considering the previously mentioned parameters latency, energy, and quality of connection.

The routing algorithm needs the following parameters: $n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K$ and is processed starting with layer $n_{\max} - 1$ and moving backward layer-by-layer up to the BS. The nodes belonging to one layer, i.e., $\mathbf{R}_f$, are ordered according to their LEE value. At this point, in order to ensure that every node gets a route, first the nodes that have only one possible link to the previous layer are handled. Afterward, nodes that have the least LEE value are processed. Let us consider a discovered network structure, as shown in Fig. 4(a). The numbers in black color denote the node, whereas the numbers in red indicate the LEE value of each node.

In this particular example, routing starts at $\mathbf{R}_3 = \{N_{20}, N_{21}, N_{22}, N_{23}\}$ since $n_{\max} = 4$. Then, from the nodes in $\mathbf{R}_3$, the ones that have connections to the nodes in $\mathbf{R}_4$ are handled first, e.g., in our example, $N_{31} \in \mathbf{R}_4$ has only one possible

[2] A wake up interval (W) denotes the period within which nodes wake up during the network discovery phase. In [23], it was shown that two wake up periods are sufficient to detect all reachable nodes in a network consisting of 100 nodes, which is a reasonable size for many scenarios, including industrial factory communication systems.

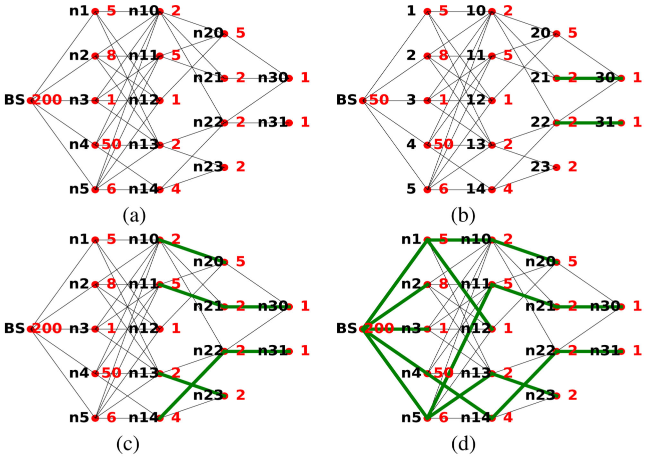


Fig. 4. Execution of the routing phase on a sample network consisting of four layers and 16 nodes. (a) Discovered, not routed network. (b) Resulting routes of $\mathbf{R}_4$. (c) Resulting routes of $\mathbf{R}_3$ and $\mathbf{R}_4$. (d) Resulting routes of the network.

connection with $N_{22}$; therefore, $N_{22}$ is processed first. This is indicated with *line* 4 in Algorithm 3.

Next, the $L$ value gets computed. Parameter $L$ determines how many nodes the current node $N_k$ can route. It is the minimum of $\#_E N_k$, which denotes the LEE value of the node $N_k$, or $\max_{\#_E} \mathbf{B}_{N_k}$, indicating the maximum LEE value of all backward nodes that $N_k$ has. Afterward, the forward nodes for $N_k$ get calculated. These are temporarily stored in the set $\mathbf{T}$. The forward nodes get chosen from a set consisting of $\{\mathbf{F}^u_{N_k}, \mathbf{F}_{N_k} \backslash \mathbf{F}^u_{N_k}\}$. This alignment ensures that we are dealing with the nodes first that do not have other possible routes into the current layer. Also, if $\mathbf{F}^u_{N_k}$ is empty, which means there are no nodes in layer $n+1$ that have a unique link into the $\mathbf{R}_n$ via the node $N_k$, no prioritization takes place. The nodes that $N_k$ can route are limited by the previously computed $L$ value. Node $N_k$ can only route a node $N_g$ if $L > \#_L \mathbf{T} + \#_L N_g$, where $\#_L \mathbf{T}$ means the total number of links the temporary set $\mathbf{T}$ has and $\#_L N_g$ describes the number of nodes $N_g$ routes. If this condition is fulfilled, the node $N_g$ gets added to the temporary set $\mathbf{T}$. After all nodes from the combined sets are handled, node $N_k$ forwards the nodes inside $\mathbf{T}$, and therefore, the routes to the node $N_k$ are finished. After the routing of node $N_k$ is finished, $\#_L N_k$ gets set to the total number of links, its forwarding nodes provide + 1 for the node $N_k$ itself. The term $\#_L N_k$ describes the total amount of nodes $N_k$ links.

The next step is to update the nodes $N_g \in \mathbf{R}_n \backslash N_k$. This includes to remove the already routed nodes $\mathbf{F}_{N_k}$ from the forward sets of $N_g$. Finally, the node $N_i \in \mathbf{B}_{N_g}$, which has the smallest LEE that still can manage the link count of $N_k$, gets removed. This ensures that there is no overbooking in the next layer. This process is performed on all nodes $N_k \in \mathbf{R}_n$ and produces a routing table $\mathbf{F}_{N_k}$. The resulting routes after dealing with the first layer can be found in Fig. 4(b), for the same network, as given in Fig. 4(a). Then, the algorithm moves one layer backward to $\mathbf{R}_{n-1}$ and starts over. This is done for all other layers in this network. The results of each layer routing are shown in Fig. 4(c) and (d), respectively.

**Algorithm 3:** Network Link Phase.

```
 1: function SETROUTES$n_{\max}, \mathbf{R}^p, \mathbf{F}^K, \mathbf{B}^K$
 2:   for $n = n_{\max} - 1; n > 0; n \leftarrow n - 1$ do
 3:     for all $N_k \in \mathbf{R}_n$ do
 4:       $\mathbf{F}^u_{N_k} \leftarrow \mathbf{F}_{N_k} \backslash (\bigcup_{N_g \in \mathbf{R}_n \backslash N_k} \mathbf{F}_{N_g})$
 5:       $\mathbf{T} \leftarrow \emptyset$
 6:       $L \leftarrow \min\{\max_{\#} \mathbf{B}_{N_k}, \#N_k\}$
 7:       for all $N_g \in \{\mathbf{F}^u_{N_k}, \mathbf{F}_{N_k} \backslash \mathbf{F}^u_{N_k}\}$ do
 8:         if $L > \#\mathbf{T} + \#N_g$ then
 9:           $\mathbf{T} \leftarrow \{\mathbf{T}, N_g\}$
10:       $\mathbf{F}_{N_k} \leftarrow \mathbf{T}$
11:       $\#N_k \leftarrow \#\mathbf{F}_{N_k} + 1$
12:       for all $N_g \in \mathbf{R}_n \backslash N_k$ do
13:         $\mathbf{F}_{N_g} \leftarrow \mathbf{F}_{N_g} \backslash \mathbf{F}_{N_k}$
14:       $\#\mathbf{B}_{N_k} \leftarrow \#\mathbf{B}_{N_k} - \min_{\#N_i}\{N_i \in \mathbf{B}_{N_k} \wedge \#N_i > \#\mathbf{T}\}$
15:   return $\mathbf{R}^p, \mathbf{F}^K$
```

While the network is operating, the nodes monitor their neighbor nodes and send the updated tables to the BS in the same way as described in Algorithm 2. At the BS the routing is recalculated anytime new and information is updated, or if it is required by a reported QoS drop at specific nodes. The new routing information is delivered to the individual nodes along with a time stamp defining the effective start. The periodicity of monitoring and rerouting is limited by the available communication and energy resources.

## IV. Energy Consumption

The QMLR algorithm calculates its routes according to the energy budget of individual nodes, given in LEE values. One LEE is defined as the energy required for the reception and transmission for one link or one packet, if we assume each link to have the same packet rate. The ramp up of the transceiver and similar operations are addressed separately. Therefore, the energy for an LEE is $E_{\text{LEE}} = E_{\text{TX}} + E_{\text{RX}}$ with $E_{\text{TX}}$ and $E_{\text{RX}}$ being the sent and received packet energy, respectively. Since our QMLR algorithm is divided in two stages: 1) the network discovery and 2) routing phase; we calculate separately the energy consumption for each phase. In addition, we count for the operational phase of the WSN.

In the network discovery phase, each routing node has to switch ON its transceiver with power consumption $P_{\text{RX}}$ for two wake-up periods (2W) and listen to its neighboring nodes [23]. All routing nodes answer to each detected node at least with one packet containing the assigned TS and synchronization information. Apart from discovery, the routing node offers its regular service by sending the beacon in one packet with a rate of $\frac{1}{T_{\text{SF}}}$. The necessary energy is given by the maximum number of answering nodes in the network that is $K - 2$, excluding the BS and the node itself. Thus, the energy of the discovery phase

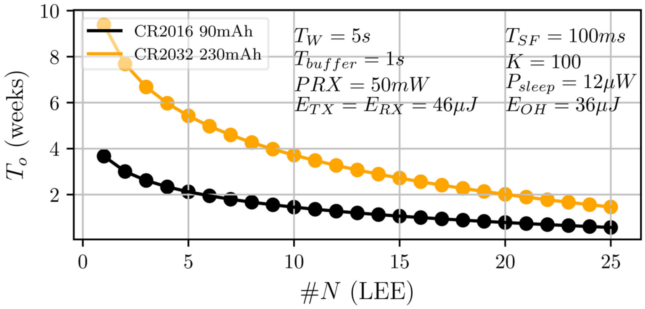


Fig. 5. Operation time $T_\mathrm{o}$ of a node.

amounts to

$$E_\mathrm{d} \leq 2T_\mathrm{W}P_\mathrm{RX} + (K-2)E_\mathrm{TX} + E_\mathrm{TX}\frac{2W}{T_\mathrm{SF}}. \tag{3}$$

As the routing calculations are done at the BS, the energy consumption of the nodes is defined by distributing the routing tables to all routing nodes. Therefore, its energy is proportional to LEE including the overhead for transceiver ramp up $E_\mathrm{OH}$. The number of links over one routing node is calculated by its routing table $\#N_k = \#\mathbf{F}_{N_k} + 1$, and therefore, $E_\mathrm{r} = \#N_k(E_\mathrm{LEE} + E_\mathrm{OH})$.

The energy consumption during the operation phase $T_\mathrm{o}$ is also proportional to $\#N_k$. Additionally, the beacon is sent every SF. Hence, it results in

$$E_\mathrm{o} = T_\mathrm{o}\left(\#N_k(E_\mathrm{LEE} + E_\mathrm{OH})\frac{1}{T_\mathrm{buffer}} + E_\mathrm{TX}\frac{1}{T_\mathrm{SF}}\right). \tag{4}$$

To save energy, the buffering time $T_\mathrm{buffer}$ is used by sending several sensor data values in one packet. Short packets are avoided, which at a high packet rate increases the overhead due to a higher number of transceiver ramp ups. The design parameter $T_\mathrm{buffer}$ has to be adapted to the application, which is reflected by the constraints time slot (TS) and maximum packet length. Finally, the total energy demand is given by the three energy values added up as $E_\mathrm{total} = E_\mathrm{d} + E_\mathrm{r} + E_\mathrm{o}$. Therefore, the total operation time $T_\mathrm{o}$ is a function of the number of relays. The results are shown in Fig. 5. In this case, we consider two different battery types (CR2016,CR2032) that are usable for the hardware implementation discussed in Section VI.

## V. Evaluation of the Network Discovery

We consider a network topology consisting of sensor nodes and a single BS in IWSNs such as factory halls. The BS is located in the center of an area 50 × 50 m, whereas the nodes are uniformly distributed within the area. The network is static, i.e., the nodes and the BS are not moving. Within the network, we implemented a TDMA scheme with nodes that are synchronized and, therefore, consider an interference-free network. The BS and nodes are equipped with an omnidirectional isotropic antenna. We consider the SNR as link quality measure, defined as

$$\mathrm{SNR}_{0,k} = \frac{P_0 L_{0,k}(d)\mathcal{X}}{S_0} \tag{5}$$

TABLE I
Simulation Parameters

| Parameter | Value |
|---|---|
| Frequency | 2.245 GHz |
| Antenna type | omnidirectional |
| Path loss exponent $a_1$ | -1.1 [29, Table 4-2] |
| Path loss exponent $a_2$ | -2.6 [29, Table 4-2] |
| Path gain $b_1$ | -46 [29, Table 4-2] |
| Path gain $b_2$ | -30 [29, Table 4-2] |
| Breakpoint distance $d_\mathrm{BP}$ | 11 [29, Table 4-2] |
| Shadow fading $\sigma_\mathrm{SF}$ | 1.7 [31, Table II] |

where the BS (node $N_0$) is located at $(x_0, y_0)$ and has transmission power $P_0$, whereas a sensor node is located at $(x_k, y_k)$. The parameter $L_{0,k}(d)$ denotes the distance dependent pathloss and the term $\mathcal{X} \sim \mathrm{Lognormal}(0, \sigma_{SF})$ represents the shadow fading random variable modeled by zero-mean lognormal distribution with standard deviation $\sigma_{SF}$. The parameter $S_0$ denotes the thermal noise floor.

To model the distance dependent propagation losses $L(d)$ between BS and sensor nodes, we consider the log-distance path loss model from [29]

$$\mathrm{L}(d) = \begin{cases} a_1 10\log_{10}(d) + b_1, & d \leq d_\mathrm{BP} \\ a_2 10\log_{10}(d) + b_2 + d_\mathrm{BP}\left(a_2 - a_1\right), & d > d_\mathrm{BP} \end{cases} \tag{6}$$

where $d$ denotes the distance in meters between two locations. Parameters $a_1$ and $a_2$ represent path loss exponents, $b_1$ and $b_2$ indicate the path gain at a certain distance from the transmitter, and $d_\mathrm{BP}$ is the breakpoint distance indicating the distance where the slope of the path loss changes. This path loss model characterizes explicitly industrial factory environments, and it is derived based on measurements performed in such environments. Since our focus in this work is on industrial factory scenarios, we select the simulation parameters accordingly to match the measurement setup, as shown in Table I. Since we consider only a static scenario and omnidirectional antennas, according to the work in [30], including small-scale propagation effects only provides a linear shift in terms of fading margin when compared to considering large-scale fading alone; therefore, we omit the small-scale fading part.

We implement the network discovery system model described in Section III-A and concentrate on evaluating the impact of network density and the SNR threshold for the link quality on the overall discovered structure with the proposed algorithm.

### A. Number of Layers Versus QoS Threshold

In this section, we focus on the impact of various design parameters on the performance of the routing algorithm. We first look at the impact of the SNR threshold on the network discovery phase and evaluate the number of layers that are necessary to maintain a given SNR threshold. With the simulation setup described earlier, considering a network size of 50 × 50 m and a node density of $\lambda = \{0.01, 0.1, 0.5\}$, we evaluate the averaged maximum number of layers $n_\mathrm{max}$ achieved in the network discovery phase. The SNR threshold is considered in the range $\gamma = [-15, 45]$ dB. Fig. 6 provides the simulation

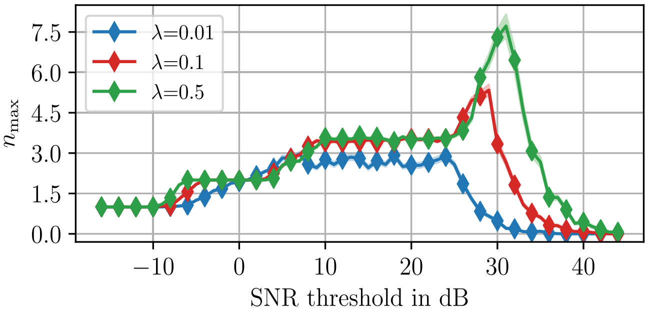
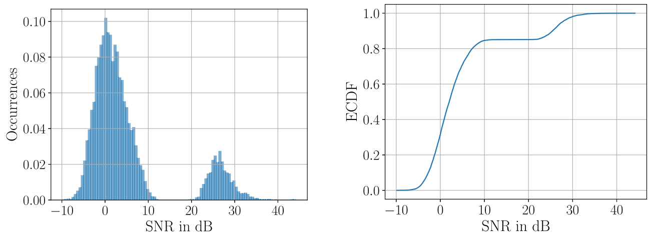


Fig. 6. Average maximum layer number over $\gamma$.



(a)

(b)

Fig. 7. Initial SNR distribution in the network. (a) Histogram of the basic SNR. (b) Initial ECDF of SNR.

results averaged over 1000 simulation realizations. The results indicate a specific range of the SNR threshold where the network discovery gives the highest number of relaying layers. With $\gamma < -10\,\text{dB}$, all nodes comprise a single layer. This comes from the fact that the SNR threshold is very low and all nodes achieve it. If the SNR threshold is increased, the average number of layers starts to increase. This increase varies with the node density. For the three depicted node densities, in the SNR range $8\,\text{dB} < \gamma < 23\,\text{dB}$, $n_{\max}$ is constant, achieving a maximum of four layers for $\lambda = \{0.1, 0.5\}$ and three layers for $\lambda = 0.01$. This behavior reflects the effects of the SNR distribution at nodes prior to the network discovery phase (see Fig. 7). In this case, the SNR distribution results to be bimodal [see Fig. 7(a)] and in the SNR range $\gamma' \in [10, 23]$ dB, there are no occurrences of the SNR. This behavior stems from the fact that the path loss model we utilize is a two-slope model that has a gap of about 17 dB of slope change at the breakpoint distance ($d_{\text{BP}} = 13\,\text{m}$). According to the work in [29], in their measurements performed in a factory hall of about the same size as ours, there are big boilers with metallic surface that contribute in a lot of reflections and blockages. Moreover, all the locations after the breakpoint distance are found to be in nonline-of-sight (NLOS). In other words, for scenarios in industrial factories, the change from line-of-sight (LOS) to NLOS increases the loss due to big metallic objects that are typical in such scenarios. In our scenario, this results in having a constant probability of $\mathcal{P}(x \leq \gamma') = 0.85$ that the specific range of SNR cannot be achieved, as depicted in Fig. 7(b). When increasing the SNR threshold to $\gamma > 23\,\text{dB}$, we notice differences in the achieved number of layers for various densities. For node density $\lambda = 0.01$, $n_{\max}$ starts to decrease, indicating that the number of nodes that can reach the given threshold is smaller. For higher node densities, we observe that $n_{\max}$ increases further until a maximal value is achieved, yielding at around five layers for $\gamma = 28\,\text{dB}$, and eight layers for $\gamma = 31\,\text{dB}$, at densities $\lambda = 0.1$ and $\lambda = 0.5$, respectively. The results indicate that the drop-off value in the number of layers depends on the node density and that, for higher densities, a higher number of layers is necessary in order to maintain a better link quality. Therefore, for not very dense networks, it is recommendable to consider networks with a maximum of four hops, which support a minimum SNR of up to 23 dB. It is important to note that a higher number of layers indicate a higher latency; therefore, it is recommended to select an SNR threshold that gives an acceptable link quality for the lowest number of layers.

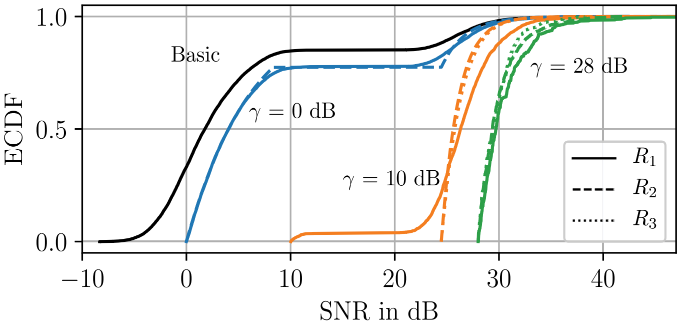


Fig. 8. SNR ECDF at different layers for SNR threshold $\gamma = \{0, 10, 28\}$ dB and node density $\lambda = 0.1$.

## B. Link Performance Evaluation

*1) Achieved SNR:* Next, we carry out a performance comparison in terms of the achieved SNR at specific layers in the network. We consider the same settings as above, a node density $\lambda = 0.1$ and depict three SNR threshold values $\gamma = \{0, 10, 28\}$ dB. Each of these values characterizes a certain behavior in the network, as conferred in Fig. 6: at $\gamma = 0\,\text{dB}$, the number of layers starts to increase, at $\gamma = 10\,\text{dB}$, it is noticed a flat behavior in terms of number of layers (the range of SNR that occurs with a very low probability, switching from LOS to NLOS), whereas at $\gamma = 28\,\text{dB}$, the maximum number of layers is obtained. Fig. 8 shows the results in terms of achieved SNR statistics. The basic setting represents the case without applying the node discovery algorithm. For $\gamma = 0\,\text{dB}$, only two layers of relay nodes are achieved in the network. An overall improvement of about $1 - 3\,\text{dB}$ in terms of achieved SNR can be noticed compared to the basic setting. For $\gamma = 10\,\text{dB}$, the network is resolved into three layers of relay nodes. The improvement in terms of SNR at each layer is substantially compared to the basic setting. As the results indicate, even though the SNR threshold is 10 dB, the network achieves an SNR higher than 23 dB. Only in the first layer, there are still nodes that operate in NLOS under the same threshold. With the proposed hierarchical structure of our algorithm, we are able to achieve very good link quality and bring the nodes closer to the BS. This enables that each layer of relay

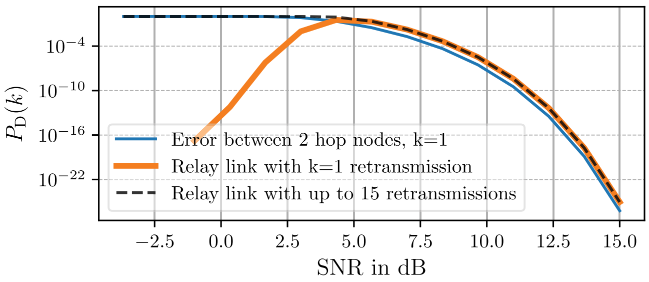


Fig. 9. Achievable $P_D(k)$ for an average packet length of 256 b defined by SNR threshold $\gamma$.

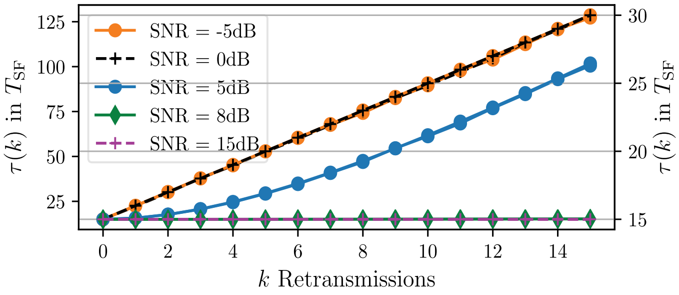


Fig. 10. Latency $\tau(k)$ for $k$ retransmissions with $P_D(k)$.

nodes is always in the higher SNR range, i.e., the second peak in the right-hand side of the histogram in Fig. 7(a). Similarly, for $\gamma = 28$ dB, the improvement follows the set threshold. Note that, for $\gamma = 28$ dB, the network achieves a larger number of layers, but we show only the first three layers in order not to overload the figure. Nevertheless, also for this case, it can be seen that with the network discovery phase, a guaranteed QoS is attained. Furthermore, it is observed that there is a gap between the first layer ($\mathbf{R}_1$) and the rest of relay layers, providing better performance for nodes in $\mathbf{R}_1$. This mainly stems from the fact that the nodes comprising $\mathbf{R}_1$ are in more favorable positions since they are directly communicating with the BS located at the center. This enables that nodes in $\mathbf{R}_1$ have always smaller distances to the BS than the distances that nodes in $\mathbf{R}_2$ have with their corresponding relays in $\mathbf{R}_1$.

*2) Latency:* We evaluate the latency as $\tau = n_{\max} T_{\mathrm{SF}}$ whereby time of flight $d/c \ll T_{\mathrm{SF}}$ is neglected. Therefore, the communication system can provide real-time conditions with defined latency $\tau$. Obviously, errors occur, which are described as bit error probability (BER) or packet error probability (PER). Based on the work in[32], the BER of QPSK with an additive white noise Gaussian channel model (also used in IEEE 802.15.4) is given by $P_{\mathrm{b}} = \frac{1}{2}\mathrm{erfc}(2\sqrt{\mathrm{SNR}})$. The PER, $P_{\mathrm{P}}$, for a packet length $L$ is, therefore, $P_{\mathrm{P}} = 1 - (1 - P_{\mathrm{b}})^L$ without considering any channel or block coding. The PER influences the defined latency, since corrupted packets must be retransmitted between hops. Each hop is a potential source of error. Hence, multiple combinations can occur and create $k$ additional delays up to $n_{\max} T_{\mathrm{SF}}$. The probability of additional $k$ delays is

$$P_{\mathrm{D}}(k) = \binom{n_{\max}}{k} P_{\mathrm{P}}^k (1 - P_{\mathrm{P}})^{(n_{\max}-k)}. \tag{7}$$

As shown in [33] and [34], increase of SNR allows us to decrease the PER dramatically for IEEE 802.15.4 communication links. Therefore, we apply the SNR threshold $\gamma$ to calculate the achievable $P_{\mathrm{D}}(k)$. According to III-A, the layer structure supports a network wide constant PER, which results in a latency guarantee of $1 - P_{\mathrm{D}}$. In Fig. 9, the $P_{\mathrm{D}}(k)$ dependence on SNR is shown for a single hop (blue line) and for any combination of $k$ packet errors. As can be seen from the figure, it is necessary to have at least 10- dB SNR to achieve a $P_{\mathrm{D}}$ of $10^{-6}$, given an average packet length of 256 b. Moreover, the probability of having only one retransmission is similar to $k$ retransmissions for SNR $> 5$ dB. Below that value, multiple corrupted packets are much more likely as single-error occurrences. As already mentioned, the PER can be reduced by applying channel or block coding, which is not scope of this article.

TABLE II
PROBABILITY OF ADDITIONAL DELAYS $kT_{\mathrm{SF}}$ (JITTER)

| SNR \ $k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 5 dB | 0.2 | 0.3 | 0.2 | 0.1 | 0.03 |
| 8 dB | 0.998 | 1e-03 | 1e-06 | 9e-10 | 3e-13 |

The average end-to-end latency $\tau$ of a relayed link with $n_{\max}$ hops and $k$ retransmissions with probability $P_{\mathrm{D}}(k)$ is

$$\tau(k) = \begin{cases} \left(n_{\max} + \sum_{r=1}^{k} \alpha(r) P_{\mathrm{D}}(k)\right) T_{\mathrm{SF}}, & \text{case I} \\ \left(n_{\max} + k P_{\mathrm{D}}(k)\right) T_{\mathrm{SF}}. & \text{case II} \end{cases}. \tag{8}$$

In case I, each failed transmission is restarted from source and the erroneous hop index is modeled by uniform distribution $\alpha(r) \sim \mathcal{U}(1, n_{\max})$. In case II, the erroneous single-hop transmission is repeated and not restarted from source, $\alpha(r) = 1$. Fig. 10 shows the average end-to-end latency depending on the number of retransmissions $k$, for various SNR values in a network with $n_{\max} = 15$ hops. Case I and case II are quantified in the left and right y-axis, respectively. The results indicate nearly deterministic latency for SNR $> 5$ dB and that error mitigation strategy case II outperforms case I clearly. To show the delay jitter, Table II presents the probability of additional $k$ delays for SNR of 5 and 8 dB.

## C. Relation to Existing Routing Algorithms

The proposed QMLR algorithm is designed for energy-efficient wireless networks aiming use cases requiring defined latency. Since the objective in this article is on deterministic WSNs, for a relation to other existing routing algorithm, we focus on WirelessHART as one of the most prominent example in deterministic industrial wireless communications. In the WirelessHART standard, the routing algorithm is not specified, leaving it open for individual implementations and strategies. To that end, there are many routing algorithms developed, most of which are based on the concept of graph routing. A graph routing involves path redundancy where intermediate nodes or relays can have multiple neighbors and thus more possibilities

TABLE III
QUALITATIVE COMPARISON OF ROUTING ALGORITHMS FOR DETERMINISTIC MAC

| Objective | ELHFR[7] | QLRR-WA[11] | QMLR |
|---|---|---|---|
| Energy reduction | ✓ | ✓ | ✓ |
| Energy balancing | ✗ | ✓ | ✓ |
| Load balancing per layer | ✗ | ✗ | ✓ |
| Average network lifetime | - | ✓ | ✗ |
| Network lifetime with all nodes | - | ✗ | ✓ |
| Greedy routing | ✓ | ✓ | ✗ |
| QoS-based network discovery | ✗ | ✓ | ✓ |
| Desired QoS level | ✗ | ✗ | ✓ |
| Nearly deterministic latency | - | ✗ | ✓ |
| Min. latency for topology | ✓ | ✓ | ✓ |
| Calculation complexity $\mathcal{O}(n)$ | ✗ | ✗ | ✓ |

to forward packets via more than one route. In this regard, also our approach can be considered to reflect a graph routing.

An interesting approach is that of authors in [7], known as the exhenced least-hop first routing (ELHFR) algorithm. The main concept in this routing is to define the network topology as a connected graph that generates subgraphs that include all shortest paths from a node to a given destination. In this way, ELHFR follows the breadth first tree approach to generate the spanning tree of the topology graph. At the end, the selection of the best route is based on the received signal strength information. Another approach that has been developed in the past few years is based on machine learning and reinforcement learning [10]–[12]. Machine learning provides the ability that a system learns and improves with experience, whereas with reinforcement learning, it learns which behavior it should follow to maximize its rewards. We consider the Q-learning reliable routing with a weighting agent (QLRR-WA) [11] as the most recent algorithm that merges machine learning and reinforcement learning and provide a comparison in terms of objectives and achievements of this approach as compared to the conventional graph routing ELHFR and our proposed QMLR algorithm.

Table III summarizes the objectives of the three aforementioned routing schemes. In terms of energy consumption reduction and energy balancing between different nodes in the network, QLRR-WA and QMLR satisfy both of these objectives, whereas ELHFR does not consider a balancing in terms of energy. Furthermore, only QMLR provides a load balancing per layer, meaning that within a layer, load at relay nodes is balanced. In terms of average network lifetime, the QLRR-WA applies a constraint on extending the network lifetime by an iterative agent learning approach; however, this algorithm allows for learning adaptations when there are fluctuations and failures in the network, thus increasing the energy consumption and reducing the averaged network lifetime significantly [11]. In contrast, the energy-related and optimized network lifetime for our QMLR algorithm is defined as long as all nodes operate to justify mission critical applications for every node. The construction of network routes is based on the QoS for QLRR-WA and QMLR; however, in QLRR-WA, the routes are chosen based on the received signal strength among possible neighboring nodes. In our QMLR, the layers are built by a desired threshold on SNR. A change in this threshold influences the topology of the network, as we have shown in this article. While all three algorithms consider minimum latency for the network topology, only QMLR provides a nearly deterministic latency. Both ELHFR and QLRR-WA utilize greedy routing, whereas the QMLR not. In a centralized routing, the network information is available centrally to rank the overall impact of routing via a specific node, which can be an advantage especially for industrial environments.

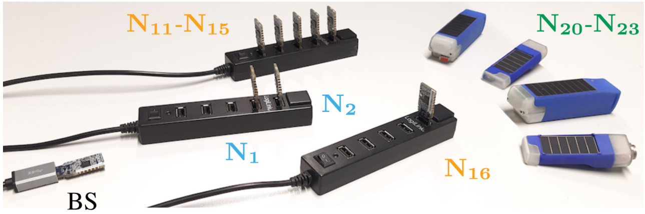


Fig. 11. Demonstrator consisting of NRF51 development boards running the QMLR algorithm with EPhESOS protocol.

## VI. DEMONSTRATOR

The demonstrator is implemented using standalone wireless nodes to experiment with various network settings. The nodes are equipped with a CortexM0 microcontroller and optionally with an attached PT100/PT1000 measurement circuit. The used Nordic NRF51822 CortexM0 has an on chip BLE transceiver, which supports energy-efficient wireless communication. In the demonstrator, the nodes are powered by either solar panels using ambient light or standard power supply. Furthermore, it is a full featured sensor network implementing the EPhESOS protocol for defined latency communication. We added the QMLR algorithm with the discovery and routing functionality to the EPhESOS implementation provided by authors in [23] and [19]. The communication protocol is not changed and so it is as energy efficient as it was designed. The protocol has a waking-up period $T_{\mathrm{W}} = 5$ s, which can be adapted to the energy saving and latency constraints. For the period $T_{\mathrm{W}}$, the nodes are not communicating and, therefore, all network activity must take this parameter into account.

Here, the demonstrator is used to support a proof of concept for the network discovery phase Algorithm 2 and the network establishing phase. The routing Algorithm 3 is evaluated by simulations as there is no difference between implementation and simulation. To generate repeatable experiments, we use predefined channel settings retrieved from the path loss model, which generates a selected network structure. Therefore, each individual realization has the same number of nodes fulfilling the link quality threshold (1) per layer. With these network realizations the discovery and routing algorithms are performed. Hence, different network topologies can be created and efficiently analyzed. Simple star networks and various multilayer networks are tested. Out of this set, we depict a representative three-layer network with the hardware arrangement as shown in Fig. 11, with first layer consisting of nodes $N_1$ and $N_2$, second layer with nodes $N_{11} - N_{15}$ connected to $N_1$ and node $N_{16}$ connected to $N_2$. The third layer ($N_{20} - N_{23}$) can be reached

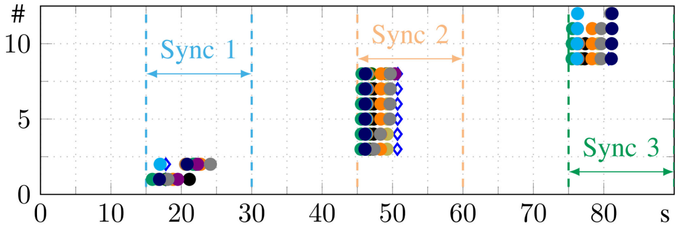


Fig. 12. Cumulative number of synchronized nodes per route as a function of network time.

from $N_{16}$. Using this topology, the discovery phase is recorded. We show here the consistency of the implementation with theory and simulation based on the use of 12 nodes. The experiments illustrate the layered discovery process required to build the forward and backward tables of the QMLR algorithm.

Algorithm 2 starts with a discovery phase at the BS from 0 to 15 s, as indicated in Fig. 12. In this phase, the nodes, which are discovered, are assigned for synchronization. Synchronization starts at 15 s and lasts until 30 s to allow all nodes in the layer ($N_1$, $N_2$) to synchronize. For the specification of the discovery duration, we use the results from [23] and set the sync duration to 3W. According to the previous procedure for layer 1, a 15-s discovery phase is applied, followed by 15-s synchronization of layer 2. As can be seen in Fig. 12, all six nodes in layer 2 are synchronized. The same procedure is repeated for layer 3 to discover and synchronize all 12 nodes in the network. Finally, based on the network discovery, the routing can be done according to Algorithm 3 resulting in a defined latency QoS-based routing.

## VII. Conclusion

With QMLR, we introduced a QoS-based routing algorithm that featured defined latency. The algorithm was supported by the power-efficient WSN protocol EPhESOS but was also applicable to other real-time wireless communication protocols. The QMLR algorithm used the SNR of the communication channel as a control parameter for the link quality and inserted a load balancing between nodes in terms of energy consumption when nodes were acting as relays. Our analysis showed that the network was adapting to the node density and communication channel based on the QoS criteria. Supported by the discovered structure of the network imposed by the QoS criteria, the algorithm guaranteed a packet delivery rate. We showed that, for a packet length of 256 b, a packet error rate of $10^{-6}$ was achievable for an SNR of at least 10 dB. Furthermore, when accounting for possible retransmissions over multiple hops, we showed that, for SNR $> 5$ dB, the probability of having $k$-retransmissions was similar to that of one-retransmission. This indicated the improvement in terms of achieving nearly deterministic latency if the SNR threshold was increased accordingly. Finally, we showed a hardware implementation and demonstrated the concept of QMLR and its usability in low-power wireless sensor networks. The network discovery process was presented with off-the-shelf sensor nodes and specially designed solar-powered sensor nodes to support wireless autarkic sensor technology in the real world.

## Acknowledgment

This article reflects only the authors' view, and the Commission is not responsible for any use that may be made of the information it contains.

## References

[1] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*. Orlando, FL, USA: Academic, 2008.

[2] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2015, Apr. 2016.

[3] S. Raza, M. Faheem, and M. Guenes, "Industrial wireless sensor and actuator networks in Industry 4.0: Exploring requirements, protocols, and challenges—A MAC survey," *Int. J. Commun. Syst.*, vol. 32, 2019, Art. no. e4074.

[4] V. P. Modekurthy, A. Saifullah, and S. Madria, "DistributedHART: A distributed real-time scheduling system for WirelessHART networks," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, 2019, pp. 216–227.

[5] J. Shi, M. Sha, and Z. Yang, "Distributed graph routing and scheduling for industrial wireless sensor-actuator networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1669–1682, Aug. 2019.

[6] F. Zhang and A. Burns, "Improvement to quick processor-demand analysis for EDF-scheduled real-time systems," in *Proc. 21st Euromicro Conf. Real-Time Syst.*, 2009, pp. 76–86.

[7] Z. Jindong, L. Zhenjun, and Z.Yaopei, "ELHFR: A graph routing in industrial wireless mesh network," in *Proc. Int. Conf. Inf. Automat.*, 2009, pp. 106–110.

[8] P. Guo, J. Cao, and X. Liu, "Lossless in-network processing in WSNs for domain-specific monitoring applications," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2130–2139, Oct. 2017.

[9] S. Zhang, A. Yan, and T. Ma, "Energy-balanced routing for maximizing network lifetime in WirelessHART," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 10, 2013, Art. no. 173185.

[10] X. Han, X. Ma, and D. Chen, "Energy-balancing routing algorithm for WirelessHART," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst.*, 2019, pp. 1–7.

[11] G. Künzel, L. S. Indrusiak, and C. E. Pereira, "Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5617–5625, Aug. 2020.

[12] H. Farag, M. Gidlund, and P. Österberg, "DeP-D: A decentralized primal-dual optimization algorithm for industrial wireless sensor networks," in *Proc. 15th IEEE Int. Workshop Factory Commun. Syst.*, 2019, pp. 1–5.

[13] K. Yu, M. Gidlund, J. Åkerberg, and M. Björkman, "Performance evaluations and measurements of the realflow routing protocol in wireless industrial networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1410–1420, Jun. 2017.

[14] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proc. 10th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2011, pp. 73–84.

[15] S. Li, S. Zhao, X. Wang, K. Zhang, and L. Li, "Adaptive and secure load-balancing routing protocol for service-oriented wireless sensor networks," *IEEE Syst. J.*, vol. 8, no. 3, pp. 858–867, Sep. 2014.

[16] K. Samara and H. Hosseini, "A routing protocol for wireless sensor networks with load balancing," in *Proc. IEEE Int. Conf. Electro Inf. Technol.*, 2016, pp. 156–161.

[17] T. T. Huynh, A. V. Dinh-Duc, and C. H. Tran, "Delay-constrained energy-efficient cluster-based multi-hop routing in wireless sensor networks," *J. Commun. Netw.*, vol. 18, no. 4, pp. 580–588, Aug. 2016.

[18] T. T. Huynh, T. N. Tran, C. H. Tran, and A. V. Dinh-Duc, "Delay constraint energy-efficient routing based on Lagrange relaxation in wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 7, no. 5, pp. 138–145, 2017.

[19] H. Bernhard, A. Springer, P. Priller, and L. B. Hörmann, "Energy balanced routing for latency minimized wireless sensor networks," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst.*, 2018, pp. 1–9.

[20] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, doi: 10.1109/HICSS.2000.926982.

[21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[22] K. R. Venugopal, S. P. T, and M. Kumaraswamy, *QoS Routing Algorithms for Wireless Sensor Networks*. Singapore: Springer, 2020.

[23] H.-P. Bernhard, A. Springer, A. Berger, and P. Priller, "Life cycle of wireless sensor nodes in industrial environments," in *Proc. 13th IEEE Int. Workshop Factory Commun. Syst.*, 2017, pp. 1–9.

[24] Y. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[25] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor," *IEEE Ind. Electron. Mag.*, vol. 5, no. 4, pp. 23–34, Dec. 2011.

[26] O. Khader and A. Willig, "An energy consumption analysis of the WirelessHART TDMA protocol," *Comput. Commun.*, vol. 36, no. 7, pp. 804–816, 2013.

[27] H. P. Bernhard, A. Berger, and A. Springer, "Timing synchronization of low power wireless sensor nodes with largely differing clock frequencies and variable synchronization intervals," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Automat.*, 2015, pp. 1–7.

[28] H.-P. Bernhard and A. Springer, "Linear complex iterative frequency estimation of sparse and non-sparse pulse and point processes," in *Proc. 25th Eur. Signal Process. Conf.*, 2017, pp. 1150–1154.

[29] " NIST channelsounder overview and channel measurements in manufacturing facilities," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. TN 1979, 2017.

[30] B. Singh, Z. Li, O. Tirkkonen, M. A. Uusitalo, and P. Mogensen, "Ultra-reliable communication in a factory environment for 5 G wireless networks: Link level and deployment study," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2016, pp. 1–5.

[31] S. Jaeckel *et al.*, "Industrial indoor measurements from 2–6 GHz for the 3GPP-NR and QuaDRiGa channel model," in *Proc. IEEE 90th Veh. Technol. Conf.*, 2019, pp. 1–7.

[32] S. Haykin, *Communication Systems*. Hoboken, NJ, USA: Wiley, 2001.

[33] S. Biswas, A. Chandra, and S. D. Roy, "Per reduction with relays for low energy short range 802.15.4 WPN," in *Proc. IEEE Int. WIE Conf. Elect. Comput. Eng.*, 2015, pp. 354–357.

[34] A. Lavric, V. Popa, I. Finis, A. Gaitan, and A. Petrariu, "Packet error rate analysis of IEEE 802.15.4 under 802.11 G and Bluetooth interferences," in *Proc. 9th Int. Conf. Commun.*, 2012, pp. 259–262.

**Fjolla Ademaj** (Member, IEEE) received the M.Sc. (Hons.) degree in electrical engineering from the Faculty of Electrical and Computer Engineering, University of Prishtina, Pristina, Kosovo, in 2014, and the Dr.Tech. (Hons.) degree in telecommunications engineering from Technische Universität (TU) Wien, Vienna, Austria, in 2019.

From 2014 to 2019, she was a Project Assistant with the Institute of Telecommunications, TU Wien, where she codeveloped the Vienna LTE-A and 5G system level simulators. Since 2019, she has been a Postdoctoral Researcher with Silicon Austria LABs GmbH Research Center, Graz, Austria. Her research interests include wireless communications, system level modeling and simulations, channel modeling, and signal processing.

**Hans-Peter Bernhard** (Senior Member, IEEE) received the master's degree in electrical engineering and the Ph.D. degree in technical sciences from Technische Universität (TU) Wien, Vienna, Austria, in 1991 and 1997, respectively.

He was an Assistant Professor with TU Wien until 1998 and joined the Johannes Kepler University (JKU), Linz, Austria, as a Lecturer, in 1999. In 2014, he started as a Senior Scientist with JKU and with Silicon Austria Labs in 2018 as a Principal Scientist and the Head of Research Unit Wireless Communications. He is currently a Visiting Researcher with the Prague Academy of Sciences, Prague, Czech Republic, and the University of Cambridge, Cambridge, U.K. His research interests include the design and analysis of time-sensitive communication systems with a focus to resilient solutions.

Dr. Bernhard has given several invited talks on various aspects of wireless factory communication and is active in organizing IEEE and ACM conferences and workshops in various chair roles. He is a Member of IEEE IES Technical Committee on Factory Automation and Technical Committee on Industrial Informatics.