

Article

A Novel MAC Protocol for Low Datarate Cooperative Mobile Robot Teams

Gaetano Patti *  and Luca Leonardi  and Lucia Lo Bello 

Department of Electrical, Electronics and Computer Engineering, University of Catania, Viale Andrea Doria 6, 95125 Catania, Italy; luca.leonardi@unict.it (L.L.); lobello@unict.it (L.L.B.)

* Correspondence: gaetano.patti@unict.it; Tel.: +39-095-738-2386

Received: 14 December 2019; Accepted: 26 January 2020; Published: 1 February 2020



Abstract: Cooperative mobile robot applications enable robots to perform tasks that are more complex than those that each single robot can perform alone. In this application context, communication networks play a very important role, as they have to cope with strict requirements (e.g., in terms of mobility, reliability, and bounded latencies). Recent cooperative robot applications foresee the support of low datarate communication technologies, that provide, among other benefits, lower energy consumption and easy integration with Wireless Sensor Networks (WSNs). Unfortunately, the state-of-the-art solutions either entail high costs and complexity or are not suitable for low data rate communications. Consequently, novel solutions for cooperating robots are required. For this reason, this paper presents RoboMAC, a new MAC protocol for mobile cooperating robots that enables the integration of robots with WSNs, supports mobility and real-time communications, and provides high scalability. The paper also presents a proof-of-concept implementation that proves the feasibility of the RoboMAC protocol on COTS devices.

Keywords: cooperative robots; mobile robots; mobile wireless sensor networks; MAC protocols; Sub-GHz wireless communications

1. Introduction

Modern automation applications provide smart devices, e.g., sensors and actuators, with the capability to process data in a distributed way, enabling the possibility of more complex industrial applications. In this context of cooperation among multiple smart devices, the introduction of teams of cooperating robots is envisaged. The cooperation of multiple robots overcomes some limitations in the robotics field, as it allows us to perform more complex tasks [1] than those that one robot can perform alone. As a consequence, there is growing interest from both industry and academia towards robotic applications. In such applications, the communication system plays a fundamental role, as robots need to exchange data among them to perform a task. In this application field, the communication protocol must support reliable and real-time communications to meet the requirements provided by the applications. The heterogeneity of the supported applications and their requirements imposes the investigation of innovative communication systems able to fulfill all the application needs. For instance, the integration of multiple mobile robots in an automation network also requires mobility support together with other features, such as real-time behavior and reliability.

In the broad spectrum of automation applications, the flexibility and adaptivity of the communication systems have also gained remarkable interest [2,3]. For instance, Germany started the so-called fourth industrial revolution (Industrie 4.0 [4]), which foresees the Internet connectivity of industrial assets [5,6]. According to the “Industrie 4.0” vision, industry shall develop networks integrating machinery, sensors and plants with nodes able to autonomously exchange information, trigger actions, and cooperate with each other without human intervention. This not only

applies to field devices or machines, but also to the mobile robots that, for instance, automatically transport various types of goods on the factory floor or handle materials in automated manufacturing systems.

The typical requirements of industrial automation communications [7] are:

- **Reliability:** automation networks have to provide a suitable reliability level for the transmissions, i.e., a low error probability. To cope with this requirement, some of the mechanisms adopted are retransmissions (with or without acknowledgment), relaying (i.e., the transmission of a message over multiple paths from the source to the destination), and Automatic Repeat reQuests (ARQs) to detect errors.
- **Fault-tolerance:** the node services have to be always available, as a stop of some hours entails high costs for the industry. For this reason, fault-tolerance mechanisms (such as node redundancy and recovery, replication of functions) have to be provided.
- **Real-Time:** networks have to meet the timing constraints of the supported applications. This means that the end-to-end delays of messages, i.e., the delivery times from the source to the destination, have to be bounded and predictable. Hence, suitable deadline-aware Medium Access Control (MAC) mechanisms to avoid unpredictability in communications and scheduling policies have to be investigated.

The above-mentioned requirements have a general significance in automation networks. However, different applications impose different constraints. In particular, cooperative multi-robot applications require networks able to support additional constraints, e.g.,

- **Low message latency:** due to the fast dynamics of this kind of systems, messages not only have to be delivered within a certain deadline (real-time requirement), but also need low message latencies (i.e., from hundreds of microseconds to hundreds of milliseconds).
- **Support to multiple traffic classes:** the integration of the Internet technologies in robotics applications and also in industrial networks requires the support for the transmission of multiple traffic types [8,9], with different timing constraints, in an efficient way and without compromising the transmission of the control traffic (that is generally the most critical one).
- **Mobility:** the introduction of mobile robots in the industrial scenario entails additional requirements on the networks. In particular, the standard wireless communication technologies in general do not specifically support node mobility. For these reasons, several mechanisms that modify the standard specifications were proposed [10,11].
- **Scalability:** cooperative multi-robot networks are typically realized with a high number of nodes. However, some of the most adopted medium access mechanisms in industrial automation are TDMA-based. Such a mechanism is well known to be not very scalable, as a high number of nodes entails a high number of timeslots in the communication cycle, thus increasing the message latency. In order to cope with this issue, several approaches were proposed [12,13].

Nowadays, cooperative robotic applications are typically based on wireless technologies [14], as they can provide support for mobile nodes. Several works [15–17] proposed cooperating robots as mobile nodes in low datarate Wireless Sensor Network (WSN). Such nodes are commonly based on the IEEE 802.11 technology [18,19]. However, IEEE 802.11 is not suitable for low datarate devices. Other works proposed robots equipped with multiple different communication technologies [15] in order to cope with diverse application requirements. Unfortunately, such a solution entails high costs and complexity. For this reason, this paper presents RoboMAC, a new MAC protocol for mobile cooperating robots. RoboMAC enables the integration of robots with WSNs, supports real-time communications and mobility, and provides high scalability.

This paper builds upon the approach proposed in [20], which outlines the design of a low datarate protocol for cooperating mobile robot applications requiring. However, in [20] the approach is merely sketched and leaves some open design issues, such as the node synchronization.

Conversely, in this paper, a node synchronization mechanism based on the expected arrival time of messages is designed. Moreover, the work in [20] does not address the implementation of the proposed approach on Commercial-Off-The-Shelves (COTS) devices. Conversely, this paper describes in detail how RoboMAC works and presents its implementation on the STMicroelectronics SPIRIT1 Sub-GHz devices, which operate on less crowded frequencies than the other Industrial, Scientific and Medical (ISM) ones and provide a higher radio coverage. The contribution of this paper consists of the RoboMAC protocol itself, a novel protocol for mobile robot applications that supports clusterization, message priorities, and multichannel communications over a TDMA-based medium access scheme. The paper presents a proof-of-concept implementation on real COTS devices to demonstrate the protocol feasibility on mobile robot applications and provides some experimental and analytical assessments.

The paper is organized as follows. Section 2 presents related works. Section 3 describes the main concepts and approaches underpinning the protocol here envisaged and presents possible solutions, while Section 4 presents the RoboMAC protocol, discussing the different implementation choices and motivating the adopted solutions. Section 5 presents an experimental assessment of the proof-of-concept implementation and some analytical results. Finally, Section 6 gives conclusions and hints for future works.

2. Related Work

Mobile robots play a vital role in WSNs and huge research efforts have been devoted on mobile robots applications [21,22] during the last decades. In [23] a review of a wide range of techniques related to mobile robots in WSNs is provided. The paper in [24] presents an overview of the research works carried out in the field of multi-robot coordination and communication.

Most of the protocols for mobile low datarate networks refer to WSNs applications. In particular, Mobile Wireless Sensor Networks (MWSNs), i.e., a growing popular class of WSN that involves mobile nodes, are becoming an important area of research for the WSN community [25,26]. For instance, several works address the routing issues of mobile robots networks [27,28].

The MobiSense protocol is presented in [29]. It is a TDMA-based protocol that operates over a hybrid network architecture made up of fixed and mobile nodes organized in a cluster-tree topology. In each cluster, nodes communicate on different channels. MobiSense provides high throughput thanks to multi-channel communication and the adoption of a fast cluster selection mechanism. However, MobiSense requires fixed nodes, thus it is not suitable for cooperative robot communications.

The work in [30] proposed M-TDMA, i.e., a TDMA-based protocol that supports mobile nodes. In M-TDMA network the nodes are organized into clusters. Each cluster is coordinated by a cluster-head node. Since a node that joins a cluster requests a slot to the cluster-head, a number of slots have to be reserved for this purpose. Other slots have to be reserved to allow inter-cluster communication. As a consequence, this reservation mechanism entails high message latencies.

Some MAC protocols defined for MWSNs are presented in [31,32]. Although some of them provide bounded delays, none of them are able to cope with all the application requirements discussed in Section 1. In [33,34] two approaches that exploit multi-channel communications to allow multiple transmission at the same time were proposed, but they do not offer mobility support. Several other protocols were proposed to support communications among mobile robots [18,19,35]. However, most of them operate on IEEE 802.11 networks, thus they are not suitable for low datarate communications. In [36,37] IsoMAC, an isochronous medium access on top of the IEEE 802.11 physical layer, was presented. The IsoMAC protocol meets the real-time, reliability and mobility requirements of cooperative robots networks, however, it needs access points to manage communications. Since nodes may lose the network connectivity when they move away from the access points coverage area, this aspect represents a disadvantage in cooperative robot networks.

The TDMA-based approach proposed in [38,39] introduced a multihop real-time protocol for Bluetooth Low Energy networks, in which it is possible to identify several subnetworks, i.e., one for each master device. However, the network is not organized in clusters. Moreover, in such an

approach, that exploits channel hopping, mobile nodes are not supported and the routing is static and configured offline. Instead, RoboMAC works on low datarate technologies. It provides support for mobile nodes and exploits a TDMA-based mechanism combined with multichannel transmissions and clustering.

This paper investigates an innovative protocol for cooperative mobile robot applications suitable for low datarate communications and able to cope with all the requirements of the considered applications.

3. Design Choices

This section presents the requirements behind the RoboMAC design choices. The first consideration refers to the network topology. Figure 1 shows an example of a cooperative mobile robot network. As it can be seen, some mobile nodes, such as nodes 6 and 7, can transmit/receive messages only to/from nodes 3 and 5, respectively. Consequently, nodes 6 and 7 will lose the connection with the entire network if they move away from nodes 3 and 5, respectively.

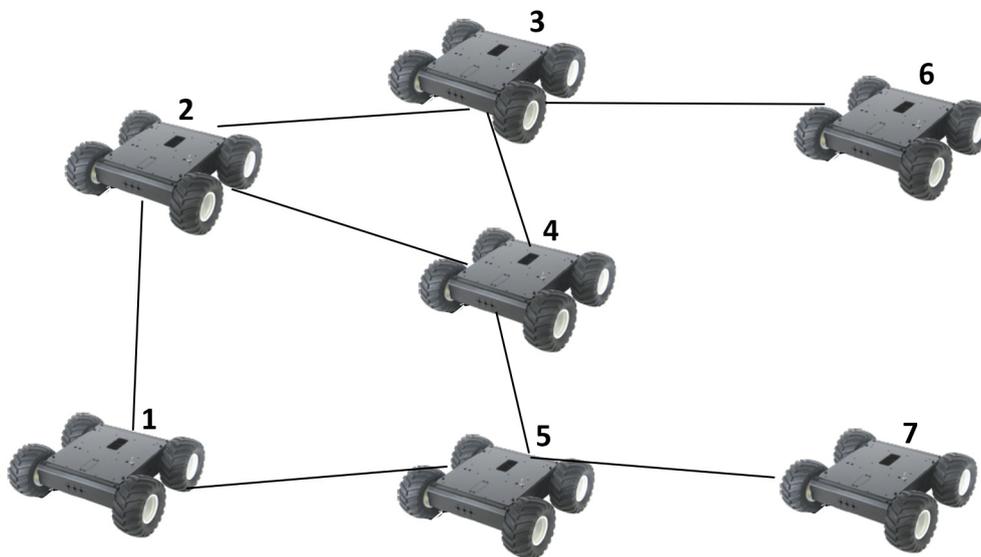


Figure 1. Cooperating robots—an example of network topology.

To ensure connectivity in the scenario shown in Figure 1, a distributed network management approach is suggested. In addition, in order to guarantee bounded delays, a TDMA-based medium access mechanism is advisable, since other approaches, such as the one proposed in [19], introduce an overhead due to network management messages, thus requiring high datarates. For these reasons, the RoboMAC protocol proposed in this paper is distributed and TDMA-based.

3.1. Bounded Delays

In RoboMAC the network time is organized into superframes that cyclically repeat. Each superframe is composed of slots. Each node has assigned one or more slots in which it is allowed to transmit one frame. The message scheduling in RoboMAC nodes is priority-based. In this paper, we assume a static priority assignment that depends on the application.

The works in [30,35] proposed the use of dedicated slots for the transmission of network management messages. Conversely, in RoboMAC we propose to transmit control information, together with data, in the slot assigned for data transmission. Control data is always transmitted (even when a node does not have data to transmit) in order to check the neighbor availability and to allow for node synchronization. This way, the number of slots in the superframe is considerably reduced, thus allowing for shorter cycle times and, hence, lower message latency.

3.2. Scalability

RoboMAC organizes the nodes in cluster depending on their position in the network (i.e., the nodes that are close to each other belong to the same cluster). As a result, RoboMAC efficiently support large networks. During the network initialization, the position of network nodes is estimated by exchanging a matrix, called Network Link Matrix, containing the Received Signal Strength Indicators (RSSI), which holds the link relations between the nodes. This way the nodes are aware of the network topology. RoboMAC uses two transmission channels, i.e., one for intracluster communications and one for intercluster ones. Intercluster communications are allowed between the nodes of two clusters when the relevant clusters do not have on-going intracluster communications.

Looking at the RoboMAC network example shown in Figure 2, during the slots assigned to the nodes of the cluster 2 (i.e., the nodes 3, 4, and 5), the nodes of the clusters 0, 1 and 3 can communicate on a different channel, provided that the message destination does not belong to cluster 2. Intercluster slots are assigned to the nodes in turn.

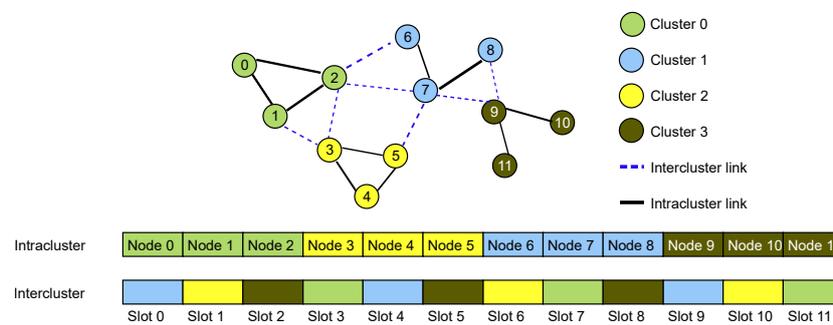


Figure 2. RoboMAC network example.

Note that all the network nodes are aware of the intracluster and intercluster superframe structures. This way, each node is able to know which clusters are able to receive intercluster messages on each slot.

3.3. Node Mobility

Cooperative mobile robots typically provide relative or absolute localization mechanisms (e.g., the ones based on the RSSI [40], Time-of-Flight [41], etc). The robot position is shared with the other robots and such information is exploited by several network management/routing tasks. For instance, it can be exploited by a routing approach based on the node position. This way, the routing protocol avoids the network overhead introduced by a network topology discovery mechanism. Conversely, if robots do not have a localization mechanism, the envisaged routing approach foresees that robots share a bitmap, here called the Network Link Matrix (NLM), that holds the link relations between the nodes. This way, the nodes are aware of the network topology.

RoboMAC provides dynamic clusters, as the nodes belonging to each cluster can vary over time. In fact, since mobile nodes are supported, the distances between the nodes can change, and so the RSSI. For this reason, RoboMAC provides a distributed topology management strategy that is based on the RSSI regularly acquired during the communications. Mobility issues, like the unpredictability of the network topology, are solved by transmitting either intercluster or intracluster topology information within the header of each exchanged frame.

As far as message routing is concerned, a position-based routing algorithm can be adopted. For instance, if the nodes share their position, geographic routing [42], i.e., a modified version of the Greedy Perimeter Stateless Routing (GPSR) [43], can be used. Otherwise, if the nodes share the NLM, RoboMAC can adopt a routing approach based on the lowest number of hops. In both cases, a RoboMAC-based network does not require additional routing messages, while supporting message transmissions in a predictable time.

RoboMAC adopts a flooding-based protocol in both intracluster and intercluster communications. Every time a node receives a message, it tests if the message is for a node of its cluster. If this is the case, it sends the message on the intracluster channel, otherwise, the node transmits the message on the intercluster channel. Mobility is supported thanks to the flooding algorithm. In fact, flooding allows messages to reach all the nodes of the network or of the cluster. To avoid message retransmission loops due to the flooding approach, the routing algorithm takes into account the message sequence number and the maximum number of hops. Each message received by a node on the intracluster channel or on the intercluster one is retransmitted once to each neighbor until it reaches the destination. Moreover, every node belonging to a cluster periodically shares a matrix containing the RSSI for the messages transmitted in the cluster, so the nodes are aware of the physical topology of the cluster. This also allows a message to be routed directly to the destination, thus avoiding flooding. The intracluster and intercluster communications allow for a reduction of the number of nodes involved in the flooding algorithm, thus reducing the number of retransmissions and, consequently, the transmission overhead generated in the network.

3.4. Advantages of the Proposed Approach

The proposed approach offers several advantages to cooperative mobile robot applications, i.e.,

- Predictable and low delays. The TDMA mechanism provides deterministic medium access time. It also allows the use of real-time algorithms to prioritize the messages using either static or dynamic priorities.
- Seamless mobility. The distributed topology management allows that messages can be transmitted by mobile robots regardless of whether they are moving or not, as such an approach enables the knowledge of the link relations among nodes.
- Improved scalability. Scalability is a known limitation of TDMA-based approaches, as the number of slots grows with the number of nodes, therefore the message delays also grow. To solve this issue, the TDMA-based mechanism, combined with multichannel transmissions and clustering, allows for shorter superframe and simultaneous transmissions on multiple channels, thus supporting tens of nodes while maintaining bounded message delays in the order of hundreds of milliseconds.

4. The RoboMAC Protocol

Figure 3 shows the architecture of a RoboMAC node. The MAC layer provides two sublayers, i.e., the medium access and synchronization sublayer and the clustering and routing one. The upper sublayer communicates with the lower one through two prioritized queues (i.e., the IntraCluster and the InterCluster PrioQueue) for the upcoming frame, and one FIFO queue for the incoming frames.

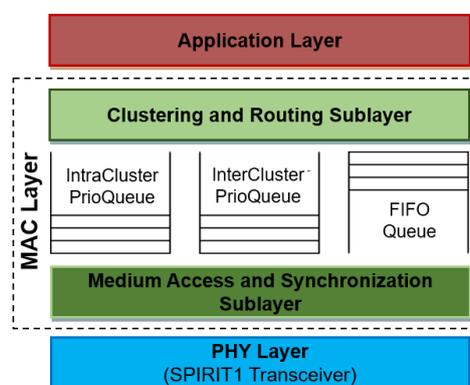


Figure 3. RoboMAC node architecture.

Each data frame arrived from the application layer is sent to the clustering and routing layer, which processes the data frame, encapsulates it in a routing frame and finally inserts the frame in the intracenter queue, if the message is for intracenter communications, or in the intercenter queue otherwise. Both the intracenter and intercenter queues are prioritized. The medium access and synchronization layer extracts the frame from the relevant queue and transmits it to the transceiver. In order to implement the medium access mechanism, as shown in Figure 2, in each superframe the clustering and routing layer modifies at runtime the superframe configuration of the medium access and synchronization layer, setting for each slot the slot owner, the channel and the relevant associated queue. This way, the medium access and synchronization layer knows the actions to undertake at the beginning of each slot.

4.1. Physical Layer

The RoboMAC protocol was specifically devised to work on low datarate devices. The STMicroelectronics SPIRIT1 [44] Sub-GHz device was chosen, as it works on frequencies that are less crowded than the 2.4 GHz ones and allows for a higher radio coverage. The adopted development board is the commercially available STMicroelectronics STEVAL-IKR002V5 [44] (Figure 4). The board is composed of a motherboard equipped with a STM32L1 family microcontroller (MCU) and the SPIRIT1 transceiver, that operates at 915 MHz and provides a datarate of 250 kbps.

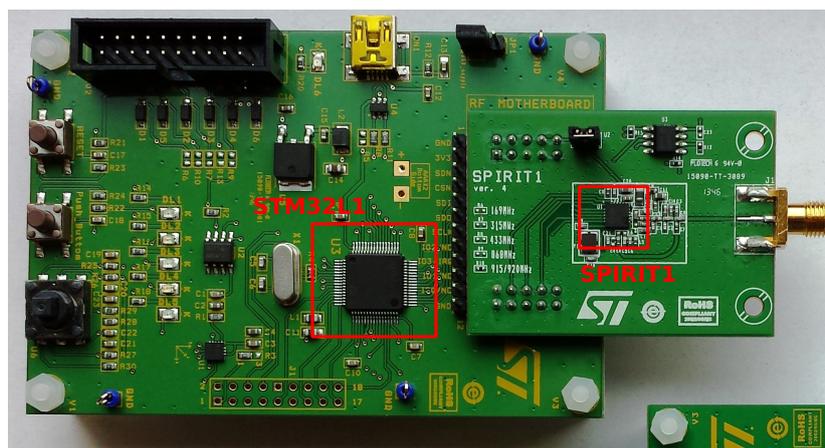


Figure 4. STMicroelectronics STEVAL-IKR002V5.

The SPIRIT1 transceiver maintains two FIFO queues, called TX-FIFO and RX-FIFO, 96 bytes long, that contain, respectively, the frames to be transmitted and the incoming ones. Moreover, multichannel communications are allowed, as the transceiver provides a channel switch function, but the channel switch time is not negligible, as the oscillator should be recalibrated every time. Note that, recalibration is not mandatory, as the calibration values can be measured and set offline. However, the oscillator recalibration leads to a better configuration, thus reducing the communication errors. Some measurements were performed to determine the channel switch time plus the recalibration time. The results showed that the time interval is no higher than 1.92 ms. The development kit provides an API set to manage the communication between the MCU and the transceiver. Due to the SPI communication, the time delay for each operation between the transceiver and the MCU is non-negligible. For this reason, the se communication delays were measured. Results are reported in Table 1.

Table 1. STEVAL-IKR002V5 measured timings.

Payload (Bytes)	T_{ptx} (μ s)	T_{tx} (μ s)	T_{rx} (μ s)	Tot. (μ s)
87	900	3400	800	5100
75	800	3100	700	4600
63	700	2700	700	4100
51	600	2300	600	3500
39	500	1900	500	2900
27	400	1500	400	2300
15	300	1100	300	1700
3	200	800	200	1200

In Table 1 the T_{ptx} represents the time taken to transmit a frame from the microcontroller to the TX-FIFO of the transceiver, T_{tx} is the transmission time and T_{rx} is the time required for receiving a frame from the RX-FIFO to the microcontroller.

4.2. Medium Access and Synchronization Layer

As mentioned before, RoboMAC provides a TDMA mechanism in which the time is divided in superframes that, in turn, are divided into slots. Each node can transmit in its assigned slots. A triplet {slot owner, transmission channel, transmission queue} is assigned to each slot of the superframe. This way the transmission algorithm knows when the node has to transmit, the channel to switch to and the queue from which the frame has to be extracted.

To enable node synchronization, each node transmits the frame at a fixed time (T_g) after the start of the slot and inserts the slot start time in the header of the MAC frame. When a node receives the sync word of the physical frame (i.e., after the first 6 bytes) an interrupt is raised. In the Interrupt Service Routine (ISR) the receiver node calculates the offset of its clock using Equation (1)

$$Offset = CurrentTime - (SlotStartTime + T_g + \frac{6 \times 8}{data\ rate}), \tag{1}$$

where SlotStartTime is the timestamp written in the header of the MAC frame by the transmitter node, $(6 \times 8) / data\ rate$ is the time required to transmit the first 6 bytes and CurrentTime is the current clock value of the receiver node. To reduce the errors caused by babbling idiot nodes that may transmit in a random time within a slot, the offsets calculated from the received packets are collected only if their values are within a time range.

As shown in Figure 5, in RoboMAC slots are sized taking into account:

- The maximum time required for channel switching (T_{ch}), which is equal to 1.92 ms;
- The maximum between the time required to cope with the synchronization accuracy (T_{sync}) and the maximum time required to transmit the frame from the microcontroller to the transceiver (i.e., $T_g = \max(T_{ptx}, T_{sync})$);
- The time needed for data transmission (T_{tx});
- The Interframe Space (IfS).

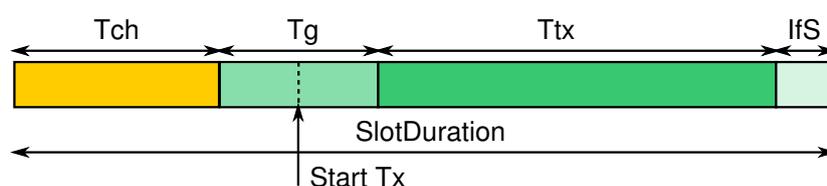


Figure 5. RoboMAC slot timings.

Hence, slots are sized according to the Equation (2)

$$\text{SlotDuration} = T_{ch} + T_g + T_{tx} + IfS. \quad (2)$$

This way the transmission completion within a slot is guaranteed.

4.3. Clustering and Routing Layer

The clustering and routing layer provides the functionalities of network and topology management. Five functional states are foreseen for this layer, as depicted in Figure 6.

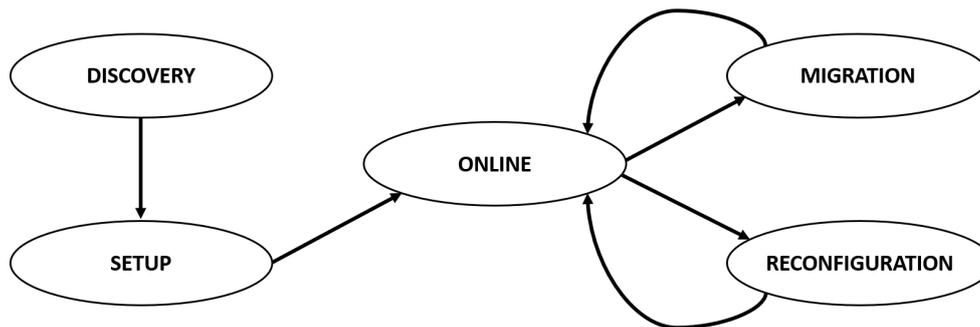


Figure 6. Clustering routing layer functional states.

- **Discovery state.** This is the initial state in which each node discovers its neighbor nodes. This state has a fixed duration, that is configured offline. The discovery state duration varies according to the number of network nodes.
- **Setup state.** When the discovery period is elapsed, nodes enter the Setup state. In this phase all the nodes exchange and update the Network Link Matrix (NLM). At the end of the setup state, all the network nodes are aware of the network topology and execute the clustering algorithm, which assigns each node to a cluster. The clustering algorithm runs in each node without message transmissions and takes into account the connectivity relations of nodes provided in the NLM. The result of the clustering algorithm is a vector containing the cluster identifier associated to each node (here called the cluster member vector).
- **Online state.** In the online state, the clustering and routing layer enables the operations of cluster maintenance and routing. Cluster maintenance is needed to update the information of the cluster member vector and the cluster topology. In the Online state, the nodes transmit in the header of the intracluster messages the NLM of the nodes belonging to a cluster. Such an information is used to discover topology changes and to start the migration or the reconfiguration phase.
- **Migration state.** A node enters this state when it moves far away from the nodes of its cluster. In this case the node starts the procedure to join another cluster.
- **Reconfiguration state.** This state is reached when there are not the minimum conditions to maintain a cluster and a new clusterization is required, for instance, when a cluster is composed of a single node. The reconfiguration state involves only two clusters, i.e., the one to which the node requested the reconfiguration and the one chosen by the node based on its position and the number of nodes in the cluster.

Both the migration and the reconfiguration states involve a restricted set of network nodes and operate within a bounded time.

5. RoboMAC Assessment

The main reason for implementing the RoboMAC protocol on Commercial-Off-The-Shelves (COTS) devices was the need to demonstrate its feasibility through a simple proof-of-concept application. Once the implemented application was available, we also used it as a testbed for some experiments

on RoboMAC. In particular, here we report the results obtained in terms of (i) Packet Loss Ratio (PLR), that are discussed in Section 5.1; (ii) the discovery and setup times, presented in Section 5.2; (iii) the protocol ability to cope with the maximum delay constraints, that is discussed in Section 5.3.

In addition to the experimental assessment of the proof-of-concept implementation, Section 5.4 also provides an analytical assessment of the RoboMAC scalability with a varying number of mobile nodes.

5.1. Packet Loss Ratio Evaluation

The testbed for the measurement was composed of five STEVAL-IKR002V5 and one PC. In each experiment the network nodes were placed in a daisy-chain topology, as shown in Figure 7, without obstacles between them. This topology is the worst configuration for the RoboMAC protocol, as each node has only two neighbors and the delay of a message from the source to the destination is the highest, due to the fact that a message has to cross all the network nodes to reach the destination.

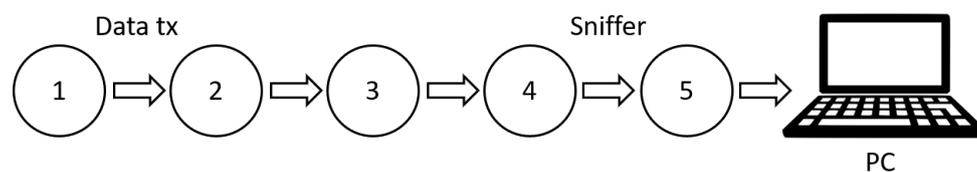


Figure 7. Testbed topology for Packet Loss Ratio (PLR) measurements.

In this assessment, the sender node generates $N = 1000$ messages 86-byte long. The payload of each message contains the overall number of generated messages. Each time the destination node receives a message, it calculates the PLR according to Equation (3) and waits for another message. Measurements were repeated varying the number of nodes and thus the number of hops (i.e., from 1 to 3 hops).

The PLR is calculated as the number of lost messages over the number of transmitted messages, i.e.,

$$PLR = 1 - \frac{NumRxMessages}{NumTxMessages}. \quad (3)$$

where NumRxMessages is the number of messages correctly received at the application of the destination node, and the NumTxMessages is the number of messages transmitted from the application of the source node.

Figure 8 shows the results of the PLR measurements. The graph shows that, in the case of 1 and 2 hops, the PLR is comparable and is lower than 3%. Instead, in the case of 3-hop communication, the PLR grows and is close to 7%. However, no consecutive packet losses were experienced.

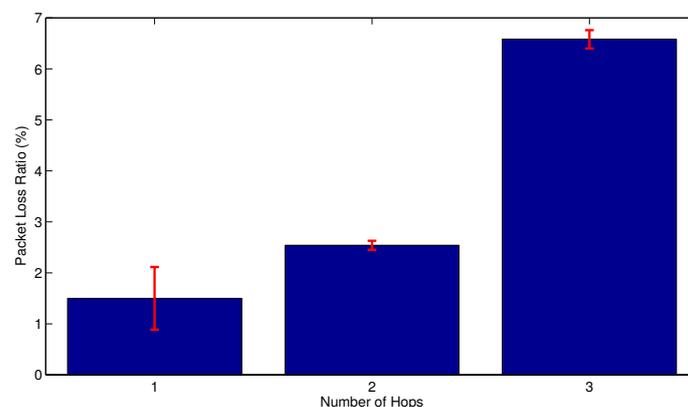


Figure 8. PLR results.

In cooperating robots, applications nodes typically share their state adopting the “last-is-best” approach (i.e., no retransmissions are needed as the last transmitted state is the best). Due to the high PLR experienced, the status of the nodes has to be transmitted more times than it is really required. This way, at least one message containing the updated status of a robot arrives with no error within a bounded interval.

5.2. SetupTime Evaluation with a Varying Number of Nodes

The Discovery and Setup times are defined as the times required by all the network nodes to complete the discovery and the setup phases, respectively. The lower their values, the shorter the network startup. The Discovery time is constant and is offline configured. In this case, we set it equal to 10s, as previous experimental measurements proved that such a value is appropriate to allow the discovery of all the network nodes. The Setup times were instead measured under a varying number of nodes (from 3 to 5). Table 2 shows the configuration parameters used in this experiment.

Table 2. Configuration parameters.

Number of Nodes	Number of Clusters	Superfram Config.	Slot Duration
3	3	[0, 1, 2]	9.6 ms
4	4	[0, 1, 2, 3]	9.6 ms
5	5	[0, 1, 2, 3, 4]	9.6 ms

In Table 2 the first column shows the number of nodes involved in the experiment, the second column is the number of configured clusters, the third column is the parameter that represents the superframe structure, and the fourth column is the slot duration. For this assessment, each experiment was repeated 10 times. The adopted testbed is the same proof-of-concept implementation presented in Section 5.1.

The average time taken by all the network nodes to complete the discovery and the setup phases with an increasing number of nodes is shown in Figure 9. The setup phase requires that nodes exchange the Link Network Matrix to run the clusterization algorithm. In the assessed scenario, the obtained setup times are 181 ± 22 ms in the case with three nodes, 155 ± 1 ms in the case with four nodes, and 286 ± 38 ms in the case of five nodes. Setup times are very low and negligible compared to the Discovery time. Here we recall that, when many nodes move far away from each other, thus creating the need for reshaping all the clusters, only the setup phase is performed. In this case, this experiment demonstrates that in the assessed scenario it takes only a few hundreds of milliseconds to reshape all the clusters. This value is totally in line with the timings of cooperative multi-robot applications.

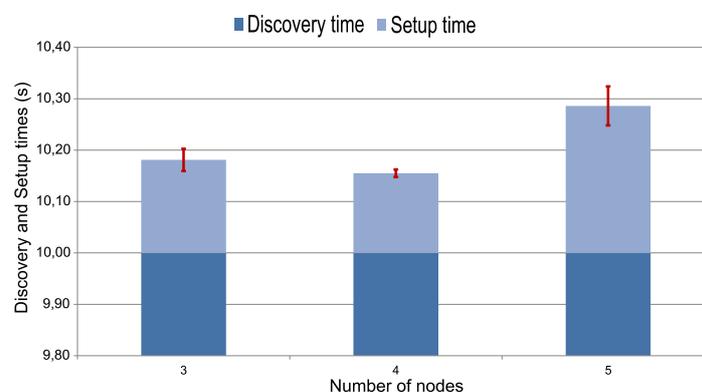


Figure 9. Discovery and setup times assessment results.

5.3. RoboMAC Test on a Proof-of-Concept Application

RoboMAC was tested in a simple proof-of-concept cooperative search application in which two robots (controlled by a centralized PC) cooperate to search and reach a radio target randomly placed in the environment. No optimized techniques were used for cooperative search, as the purpose of the proof-of-concept is to prove the RoboMAC feasibility, and the considered application is just an example. The assessed scenario is presented in Figure 10.

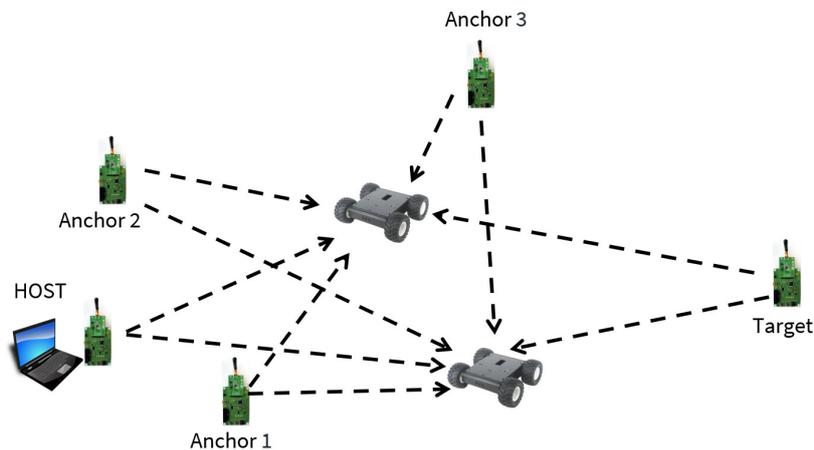


Figure 10. Assessed scenario.

The scenario is composed of 7 nodes:

- 2 mobile robots equipped with a STEVAL-IKR002V5 communication device, one IMU ST iNemo-M1 (Accelerometer, Gyroscope, Magnetometer), 3 ultrasonic sensors and a STM32F4 microcontroller.
- 3 STEVAL-IKR002V5 nodes, which act as anchors and are placed in defined and fixed positions. Each anchor periodically transmits a frame to the robots, to allow them to retrieve the RSSI and estimate the distance to the anchor.
- 1 Notebook connected to one STEVAL-IKR002V5 that acts as a host controller for the robots.
- 1 STEVAL-IKR002V5 device randomly placed into the test area. Such a device acts as a target for the robots. It periodically transmits a beacon frame to the robots so as to allow them to estimate their distance to the target.

The robots collect all the RSSI values received from the anchors and from the target. Then, each robot transmits such information plus its status (i.e., motors status and sensors sampled values) to the host that, in turn, estimates the position of both the robots and the target and transmits the relevant motors command to the robots.

As the host works only with the data collected from the two robots, the classical trilateration method to estimate the position of the target does not provide a single point, but two points, as shown in Figure 11.

As it is possible to see in Figure 11, one of the two estimated positions is the one in which the real target is placed, so the host sends one robot to one estimated position and the other one to the second estimated position.

All of these operations are required within 200 ms, that in this case is the minimum cycle time, i.e., the time within which all nodes in the network have to exchange their information/status at least once.

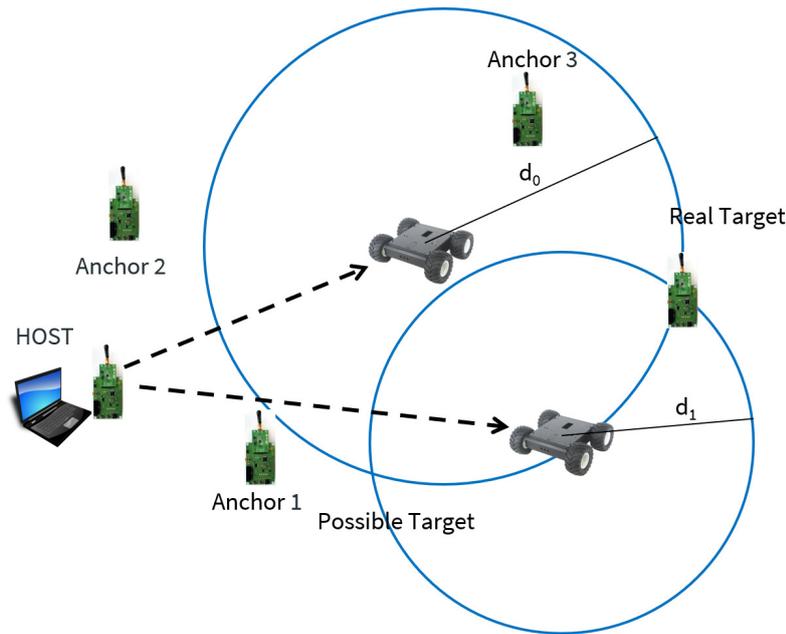


Figure 11. Estimated target positions.

In this scenario the superframe is made up of eight slots. Each node is assigned one slot, with the exception of the host that has assigned two slots for communicating with both the robots. The slot length was configured to be 10 ms, so the entire superframe length is of 80 ms. This means that the application cannot tolerate two consecutive message losses, as the message relative deadline is 200 ms. In order to test that the real-time constraints are met, if no commands arrive to a robot for more than 200 ms, the entire application terminates.

Note that the commands from the controller to the robots are very simple (e.g., “move forward with speed Y ” or “turn right with speed Z ”, etc.).

The experiment was repeated more than 10 times and robots never stopped moving until they reached the target. Such a result confirms that all messages always arrived to the destination within the deadline (i.e., within 200 ms) regardless of the high PLR experienced by the devices. This result confirms that, in the proof-of-concept application, the RoboMAC protocol was able to cope with the real-time constraints imposed by multi-robot applications while supporting node mobility.

5.4. Scalability Assessment

Two main factors affect the scalability of the RoboMac protocol, i.e., the overall workload and the superframe length.

The first one is the workload generated by the application, in terms of the number of flows handled by each node, their periods and intended destinations. In general, the higher the workload, the longer is the queuing time of a message in the transmission queue. In this work message priorities, clusterization and multichannel communication are the solutions proposed to deal with the application workload. The idea behind the clusterization scheme is that cooperating robots are typically located close to each other [45]. Hence, if the robots belonging to the same cluster cooperate in the same task, most of the exchanged messages remain within the cluster. Conversely, robots exchange a lower number of messages with the robots belonging to external clusters, for operations such as task assignment and event-driven messages, and the delay constraints are longer. Some example of these applications are presented in [46,47], where drones and/or multiple mobile ground robots cooperate to perform specific tasks. As intercluster communications are typically used to transmit management messages or event-driven messages that have more relaxed temporal constraints than intracluster communications, the scalability assessment here described focuses on the intracluster communications, that features the most stringent time constraints.

The second factor that affects scalability is the superframe length, which depends on the number of network nodes. In fact, each node requires at least one slot to transmit messages and this entails a longer superframe, a longer cycle time and, consequently, higher delays. For example, in an application of cooperative search, like the one presented in Section 5.3, each network node transmits one intracluster message per cycle, with the exception of the node in charge of running the control algorithm, which transmits one message per cycle to each mobile robot within the cluster. The total amount of slots required to transmit all intercluster messages is given by the summation of the overall number of fixed nodes (N) plus twice the number of mobile nodes (M), i.e.,

$$SF = N + 2M. \quad (4)$$

We hereby provide an analytical calculation of scalability in a scenario in which the number of mobile nodes increases from 2 to 50. In this scenario, three anchor nodes are placed in the same coverage area, so that all the nodes are able to communicate with each other. This choice is to realize the worst-case for intracluster communications (as nodes cannot communicate at the same time on the same channel). The slot size is set to 7 ms, as this is the realistic value achieved with the STMicroelectronics STEVAL-IKR002V5 devices during the experiments presented in Section 5.3. The assessed metric is the cycle time (CT), defined as the time taken to exchange the input/output data between the controller, i.e., the host in our case, and all the networked devices once, calculated as

$$CT = SF \times T_s, \quad (5)$$

where, SF is the number of slots in the superframe, calculated as in Equation (4), and T_s is the slot duration. Figure 12 shows the cycle time calculated with Equation (5) varying the number of mobile nodes.

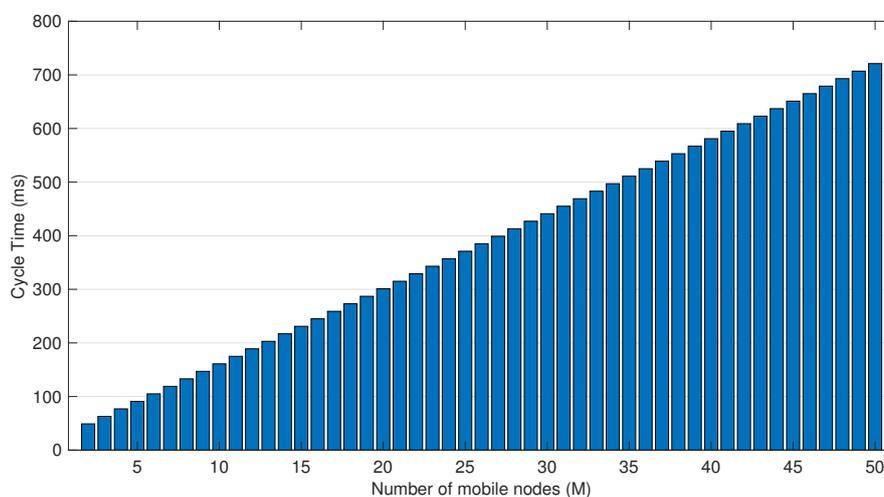


Figure 12. Scalability assessment results.

The analytical results show that, as it was expected, the cycle time grows proportionally with the number of nodes. It is interesting that, in the assessed scenario, in the case of 50 nodes, the cycle time is lower than 800 ms.

6. Conclusions

This paper addressed an innovative MAC protocol, called RoboMAC, that is specifically devised for low datarate networks of cooperating mobile robots. RoboMAC supports mobility thanks to flooding and a distributed topology discovery mechanism that enables each node to join a cluster without the need for any centralized coordinator. Moreover, the combination of clustering with flooding

transmissions solves the problem of link unpredictability due to the node mobility. RoboMAC also achieves bounded delays through a TDMA-based access mechanism. A proof-of-concept implementation on the STMicroelectronics Spirit1 Sub-GHz devices proved the feasibility of RoboMAC on COTS devices and was also used to perform some experimental assessments. The results obtained confirmed the protocol ability to provide bounded delays, thus making it suitable for cooperating mobile robot networks. The protocol is also able to provide short cluster reconfiguration times, thus proving to be adequate for the needs of industrial robotics applications. Future work will deal with innovative retransmission policies to improve reliability. Moreover, as robots are typically battery-powered, energy savings policies will be also addressed. Finally, an improvement of the routing algorithm that operates on the RoboMAC network layer will be investigated.

Author Contributions: Conceptualization, methodology, investigation, writing—original draft preparation, and writing—review and editing: G.P., L.L., L.L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yan, Z.; Jouandeau, N.; Cherif, A.A. A Survey and Analysis of Multi-Robot Coordination. *Int. J. Adv. Robotic Syst.* **2013**, *10*, 1–18. [CrossRef]
2. Leonardi, L.; Ashjaei, M.; Fotouhi, H.; Lo Bello, L. A Proposal Towards Software-Defined Management of Heterogeneous Virtualized Industrial Networks. In Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN 2019), Espoo, Finland, 23–25 July 2019.
3. Iannizzotto, G.; La Rosa, F.; Costanzo, C.; Lanzafame, P. A multimodal perceptual user interface for collaborative environments. In *Lecture Notes in Computer Science*; 3617 LNCS; Springer: Berlin, Germany, 2005; pp. 115–122.
4. Germany Federal Ministry for Economic Affairs and Energy (BMWi). *Industrie 4.0—Aspects of the Research Roadmap in Application Scenarios*. 2016. Available online: <http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.html> (accessed on 10 December 2019)
5. Battaglia, F.; Iannizzotto, G.; Lo Bello, L. JxActinium: A runtime manager for secure REST-ful CoAP applications working over JXTA. In Proceedings of the ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016; pp. 1611–1618.
6. Fazio, M.; Puliafito, A. Cloud4sens: A cloud-based architecture for sensor controlling and monitoring. *IEEE Commun. Mag.* **2015**, *53*, 41–47. [CrossRef]
7. Vitturi, S.; Zunino, C.; Sauter, T. Industrial Communication Systems and Their Future Challenges: Next-Generation Ethernet, IIoT, and 5G. *Proc. IEEE* **2019**, *107*, 944–961. [CrossRef]
8. Leonardi, L.; Battaglia, F.; Lo Bello, L. RT-LoRa: A Medium Access Strategy to support Real-time flows over LoRa-based networks for Industrial IoT applications. *IEEE Internet Things J.* **2019**, *6*. [CrossRef]
9. Leonardi, L.; Battaglia, F.; Patti, G.; Lo Bello, L. Industrial LoRa: A Novel Medium Access Strategy for LoRa in Industry 4.0 Applications. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 4141–4146. [CrossRef]
10. Al-Nidawi, Y.; Yahya, H.; Kemp, A. Tackling Mobility in Low Latency Deterministic Multihop IEEE 802.15.4e Sensor Network. *IEEE Sens. J.* **2016**, *16*, 1412–1427. [CrossRef]
11. Sauter, T.; Jasperneite, J.; Lo Bello, L. Towards new hybrid networks for industrial automation. In Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA), Torino, Italy, 4–7 September 2009.
12. Battaglia, F.; Collotta, M.; Leonardi, L.; Lo Bello, L.; Patti, G. Novel Extensions to Enhance Scalability and Reliability of the IEEE 802.15.4-DSME Protocol. *Electronics* **2020**, *9*, 126. [CrossRef]
13. Battaglia, F.; Collotta, M.; Leonardi, L.; Lo Bello, L.; Patti, G. A scalable approach for periodic traffic scheduling in IEEE 802.15.4-DSME networks. In Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN 2019), Espoo, Finland, 23–25 July 2019.

14. Alizadeh Jarchlo, E.; Haxhibeqiri, J.; Moerman, I.; Hoebeke, J. To Mesh or not to Mesh: Flexible Wireless Indoor Communication Among Mobile Robots in Industrial Environments. In Proceedings of the 15th International Conference, ADHOC-NOW 2016, Lille, France, 4–6 July 2016; Volume 9724, pp. 325–338. [[CrossRef](#)]
15. Derbas, A.M.; Al-Aubidy, K.M.; Ali, M.M.; Al-Mutairi, A.W. Multi-Robot System for Real-Time Sensing and Monitoring. In Proceedings of the International Workshop on Research and Education in Mechatronics (REM), El Gouna, Red Sea, Egypt, 9–11 September 2014.
16. Wichmann, A.; Okkalioglu, B.D.; Korkmaz, T. The integration of mobile (tele) robotics and wireless sensor networks: A survey. *Comput. Commun.* **2014**, *51*, 21–35. [[CrossRef](#)]
17. Koubaa, A.; Abdelmajid, K. (Eds.) *Cooperative Robots and Sensor Networks 2014*; Springer: Berlin, Germany, 2014.
18. Santos, F.; Almeida, L.; Lopes, L.S. Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2008; pp. 1197–1204.
19. Tardioli, D.; Sicignano, D.; Villarroel, J.L. A wireless multi-hop protocol for real-time applications. *Comput. Commun.* **2015**, *55*, 4–21. [[CrossRef](#)]
20. Patti, G.; Muscato, G.; Abbate, N.; Lo Bello, L. Towards low-datarate communications for cooperative mobile robots. In Proceedings of the 2015 IEEE World Conference on Factory Communication Systems (WFCS), Palma de Mallorca, Spain, 27–29 May 2015; pp. 1–4. [[CrossRef](#)]
21. Kazala, R.; Taneva, A.; Petrov, M.; Penkov, S. Wireless Network for Mobile Robot Applications. *IFAC-PapersOnLine* **2015**, *48*, 231–236. [[CrossRef](#)]
22. Dandashy, T.; Al-Mouhamed, M.; Khan, I. A reliable peer-to-peer protocol for multi-robot operating in mobile ad-hoc wireless networks. *Int. Arab J. Inf. Technol.* **2019**, *16*, 72–79.
23. Huang, H.; Savkin, A.V.; Ding, M.; Huang, C. Mobile robots in wireless sensor networks: A survey on tasks. *Compute. Netw.* **2019**, *148*, 1–19. [[CrossRef](#)]
24. Doriya, R.; Mishra, S.; Gupta, S. A brief survey and analysis of multi-robot communication and coordination. In Proceedings of the International Conference on Computing, Communication Automation, Greater Noida, India, 15–16 May 2015; pp. 1014–1021. [[CrossRef](#)]
25. Darif, A.; Hasna, C.; Rachid, S. *An Energy-Efficient MAC Protocol for Mobile Wireless Sensor Network Based on IR-UWB*; Springer: Berlin, Germany, 2018; pp. 1–12. [[CrossRef](#)]
26. Zhang, W.; Yang, X.; Yu, L.; Liu, S. Sequential Fusion Estimation for RSS-Based Mobile Robots Localization With Event-Driven WSNs. *IEEE Trans. Ind. Inf.* **2016**, *12*, 1519–1528. [[CrossRef](#)]
27. Maxon, S.; Zhang, F. Extending a routing protocol for mobile robot mesh networking. In Proceedings of the 2017 IEEE Conference on Control Technology and Applications (CCTA), Mauna Lani Resort, HI, USA, 27–30 August 2017; pp. 37–42. [[CrossRef](#)]
28. Sharma, H.; Rajesh, M. Design and simulation of WSN for ZigBee based communication in multi-robot system. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, Tamil Nadu, India, 1–2 October 2017; pp. 1353–1355. [[CrossRef](#)]
29. Gong, A.; Landsiedel, O.; Johansson, M. MobiSense: Power-efficient micro-mobility in wireless sensor networks. In Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, Spain 27–29 June 2011.
30. Jhumka, A.; Kulkarni, S. On the Design of Mobility-tolerant TDMA-based Media Access Control (MAC) Protocol for Mobile Sensor Networks. In Proceedings of the 4th International Conference on Distributed Computing and Internet Technology (ICDCIT), Bangalore, India, 17–20 December 2007; pp. 42–53.
31. Sudheendran, S.; Bouachir, O.; Moussa, S.; Dahmane, A.O. Review—Challenges of mobility aware MAC protocols in WSN, year=2018. In Proceedings of the 2018 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, UAE, 6–7 February 2018; pp. 1–6. [[CrossRef](#)]
32. Dong, Q.; Dargie, W. A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 88–100. [[CrossRef](#)]
33. Toscano, E.; Lo Bello, L. A multichannel approach to avoid beacon collisions in IEEE 802.15.4 cluster-tree industrial networks. In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, (ETFA), Palma de Mallorca, Spain, 22–25 September 2009.

34. Patti, G.; Alderisi, G.; Lo Bello, L. Introducing multi-level communication in the IEEE 802.15.4e protocol: The MultiChannel-LLDN. In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; pp. 1–8.
35. Facchinetti, T.; Almeida, L.; Buttazzo, G.C.; Marchini, C. Real-Time Resource Reservation Protocol for Wireless Mobile Ad Hoc Networks. In Proceedings of the IEEE International Real-Time Systems Symposium (RTSS), Lisbon, Portugal, 5–8 December 2004; pp. 382–391.
36. Trsek, H.; Jasperneite, J. An isochronous medium access for real-time wireless communications in industrial automation systems—A use case for wireless clock synchronization. In Proceedings of the International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS), Munich, Germany, 12–16 September 2011; pp. 81–86.
37. Trsek, H.; Wisniewski, L.; Toscano, E.; Lo Bello, L. A flexible approach for real-time wireless communications in adaptable industrial automation systems. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2011.
38. Leonardi, L.; Patti, G.; Lo Bello, L. Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access* **2018**, *6*, 26505–26519. [[CrossRef](#)]
39. Patti, G.; Leonardi, L.; Lo Bello, L. A Bluetooth Low Energy real-time protocol for Industrial Wireless mesh Networks. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 4627–4632. [[CrossRef](#)]
40. An, P. An Algorithm of Mobile Robot Node Location Based on Wireless Sensor Network. *Int. J. Online Eng.* **2017**, *13*, 4. [[CrossRef](#)]
41. Liu, W.; Lv, M.; Luo, Y. Sequential Time-of-Flight: Localization of mobile robots with single receiver. *J. Harbin Instit. Technol.* **2016**, *23*, 37–43. [[CrossRef](#)]
42. Cadger, F.; Curran, K.; Santos, J.; Moffet, S. A Survey of Geographical Routing in Wireless Ad-Hoc Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 621–653. [[CrossRef](#)]
43. Karp, B.; Kung, H.T. Gpsr: Greedy perimeter stateless routing for wireless networks. In Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom), Atlanta, GA, USA, 23–28 September 2000; ACM: New York, NY, USA, 2000; p. 243–254.
44. STEVAL-IKR002V5: SPIRIT1—Low Data Rate Transceiver—915 MHz—Full Kit. 2014. Available online: https://www.st.com/resource/en/data_brief/steval-ikr002v5.pdf (accessed on 10 December 2019)
45. Rizk, Y.; Awad, M.; Tunstel, E.W. Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Comput. Surv.* **2019**, *52*, 1–31. [[CrossRef](#)]
46. Khan, A.; Rinner, B.; Cavallaro, A. Cooperative Robots to Observe Moving Targets: Review. *IEEE Trans. Cybern.* **2018**, *48*, 187–198. [[CrossRef](#)] [[PubMed](#)]
47. Rosa, L.; Cognetti, M.; Nicastro, A.; Alvarez, P.; Oriolo, G. Multi-task Cooperative Control in a Heterogeneous Ground-Air Robot Team. *IFAC-PapersOnLine* **2015**, *48*, 53–58. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).