A Priority-Aware Multichannel Adaptive Framework for the IEEE 802.15.4e-LLDN

Gaetano Patti, Lucia Lo Bello

***Abstract*—The Low Latency Deterministic Network (LLDN) protocol defined in the IEEE 802.15.4e amendment is intended for factory automation applications that require very low latency and large networks, such as automotive manufacturing. However, LLDN does not provide priority support to properly deal with real-time traffic or dynamic channel configuration capabilities to cope with unreliable channels. Moreover, it offers a limited scalability, as the cycle time grows linearly with the number of network nodes. This paper proposes the Priority-aware Multichannel Adaptive (PriMulA) framework, which introduces in the LLDN priority-aware scheduling, multichannel communication, adaptive channel selection, and channel blacklisting. PriMulA supports a higher number of network nodes than the LLDN protocol while keeping short cycle times. In addition, PriMuLA avoids deadline miss and improves the network reliability. It maintains the interoperability with LLDN standard nodes and can be implemented on Commercial Off-The-Shelf (COTS) devices. The paper presents the framework, a schedulability analysis, comparative simulations, and a proof-of-concept implementation.**

***Index Terms*—Real-time networks, Industrial networks, Real-time scheduling, IEEE 802.15.4e, Low Latency Deterministic Network.**

I. Introduction and motivation

INDUSTRIAL wireless sensor networks (IWSNs) require bounded message latency [1]–[3], high reliability [4]–[6], scalability [7]–[9] and, in some cases, very short cycle times. As such requirements were not adequately addressed by the IEEE Std 802.15.4-2011, the IEEE 802.15.4e amendment [10] was proposed, which defines three Media Access Control (MAC) protocols, i.e., the Low Latency Deterministic Network (LLDN), the Deterministic and Synchronous Multi-channel Extension (DSME) and the Time Slotted Channel Hopping (TSCH). These protocols are quite different, as they target different industrial application domains. DSME fits well general industrial applications that do not need low latency, but require deterministic latency, flexibility, high reliability, efficiency, scalability, and robustness. The TSCH instead meets the requirements of process control applications, while the typical applications addressed by the LLDN are those found in factory automation (such as, automotive manufacturing), in which a large number of devices observe and control the production. These applications, which require very low latency and large networks with many sensors (even more than 100), could not be supported by DSME or TSCH. In fact, the DSME protocol proved to be able to achieve low latency (i.e., in the order of hundreds of milliseconds) by using the Ultra Wide Band physical layer, which operates at 850 kbps [11]. However, with the most common IEEE 802.15.4 physical layer operating at 250 kbps, low latency is difficult to obtain with the DSME for two reasons. First, the DSME MAC frames have a high overhead (as typically they are the same as the original IEEE 802.15.4 protocol). Second, the DSME provides a constrained superframe structure composed of sixteen timeslots, one for the beacon and the others for the Contention Access and the Contention Free Period, respectively. If more timeslots are needed, an entire superframe has to be added in the multi-superframe and this increases latency. As far as the TSCH is concerned, some assessments were recently proposed in the literature [12], [13]. In particular, TSCH proved to be not suitable to support applications that require low latency (or round-trip time) and, more in general, applications with fast dynamics. In fact, the TSCH uses the classical IEEE 802.15.4 MAC frames, which entail a high overhead and therefore a high cycle time [12]. Conversely, in the LLDN low latency transmissions are achieved introducing MAC frames that provide a minimum MAC overhead of 3 bytes (1 byte for the header, 2 bytes for the FCS).

However, the LLDN protocol has some limitations. First, it does not provide any message prioritization mechanism. Second, it suffers from scalability problems, as the cycle time (i.e., the time in which all nodes can transmit once) grows linearly with the number of network nodes. Finally, LLDN does not provide mechanisms for dynamic channel configuration, so if the current channel becomes unreliable, there is no way to dynamically switch to another channel. To overcome these limitations, this paper proposes the Priority-aware Multichannel Adaptive (PriMulA) framework, which introduces in LLDN priority-aware scheduling, multichannel communication and dynamic channel configuration. Thanks to the multichannel communication, PriMulA allows for an increase in the number of nodes in the LLDN network while minimizing the superframe augmentation. PriMulA therefore enables the deployment of larger LLDN networks for applications requiring low latency (e.g., round-trip time in the order of tens of milliseconds), such as robotic and automotive manufacturing [10]. PriMulA improves the network reliability providing a mechanism for adaptive channel selection and blacklisting and, thanks to the introduction of message priority, PriMulA avoids deadline miss. Finally, PriMulA maintains the interoperability with LLDN standard nodes and can be implemented on COTS devices.

The paper is organized as follows. Sect. II presents related work, while Sect. III overviews the LLDN protocol. Sect. IV

Gaetano Patti and Lucia Lo Bello are with the Department of Electrical, Electronics and Computer Engineering, University of Catania, 95125, Catania, Italy (e-mail: gaetano.patti@unict.it, lobello@unict.it)

describes the PriMulA framework. Sect. V presents a schedulability analysis for real-time messages. Sect. VI provides comparative assessments of LLDN, PriMulA and another approach, obtained through OMNeT simulations. Sect. VII presents an implementation of PriMulA on off-the-shelf devices. Finally, Sect. VIII concludes the paper and gives hints for future work.

II. Related Work

Several works in the literature present solutions to address the requirements of IWSNs. Shen at al. [3] propose a MAC protocol to support high priority traffic over IWSNs, while Yan et al. [2] investigate the planning of the superframe structure of slotted MAC protocols. To improve scalability and delays, Tung et al. [9] suggest multiple transceivers in a single node. This approach reduces delays, as multiple nodes can transmit at the same time, but increases complexity and costs. Toscano and Lo Bello [14], [8] propose a multi-channel approach for the original IEEE 802.15.4 beacon-enabled protocol with the twofold aim of avoiding beacon collisions and allowing multiple devices to communicate at the same time in networks with multiple nodes. Mechanisms to enhance reliability in IWSN are proposed in the literature [15]–[17]. Dobslaw et al. [6] present and assess a low-complexity scheduling algorithm to guarantee bounded end-to-end reliability, while Willig et al. [18] address relaying and the problem of scheduling relayers and retransmission slots for periodic flows in a Time Division Multiple Access (TDMA)-based network. To improve slot efficiency, Yang et al. [19] present a shared slot scheduling for retransmission in IWSNs. Unlike such works, PriMulA addresses three objectives, i.e., the ability to cope with message deadlines, reliability, and scalability, with the aim of achieving all of them in an integrated way. As far as LLDN is concerned, Dariz et al. [20] propose an enhancement of the LLDN to reduce the cycle time and handle different traffic classes, which provides for different timeslot lengths according to the data to be transmitted. However, the approach has the drawback of introducing heavy modifications in the LLDN standard. In the IEEE 802.15.4k [21] standard a mechanism for the transmission of critical event messages, called a Priority Channel Access (PCA), was specified. The PCA allows nodes to transmit critical event messages during the Contention Access Period using a modified Carrier Sense Multiple Access (CSMA) mechanism, which, on average, reduces the backoff duration compared to that of the plain CSMA. However, unlike PriMulA, the PCA does not provide guarantees on the maximum message delay. The MC-LLDN proposed by Patti et al. [22] provides a two-level network, in which a Higher Level Network (HLN), made up of nodes that are directly connected to the Personal Area Network (PAN)-Coordinator, and several sub-networks operate at the same time on different channels. Such an approach improves the cycle time up to 44% in a network with 100 nodes. However, the MC-LLDN cannot cope with message deadlines, as there is no way to prioritize messages based on their different temporal constraints. In addition, in MC-LLDN the channels for the HLN and the sub-networks are statically allocated. Consequently, if one such channel becomes permanently unreliable (e.g., due to interference), the relevant sub-network would be unreachable or, if the affected channel is the HLN one, the entire network would be disrupted. The PriMulA framework is the solution for addressing all these aspects in a unified way.

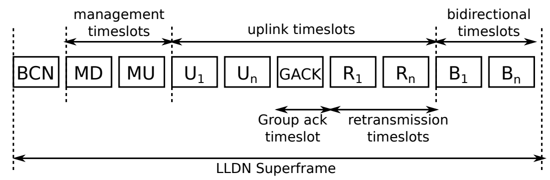

Fig. 1. The LLDN Superframe (redrawn version from [10])

III. LLDN: an overview

In the LLDN protocol, the network time is organized in cyclically scheduled superframes made up of equally sized timeslots, as shown in Fig. 1. The LLDN superframe starts with the transmission of an LL-Beacon frame (BCN) from the PAN-Coordinator. The beacon is used for synchronization purposes and contains information about the superframe structure and the network state (i.e., number of timeslots, acknowledgement information, timeslot direction, etc.). The LLDN global state can change at any time and this information is contained in the beacon frame, hence, if the beacon is lost, the transmission of the end nodes is compromised for the entire cycle. The beacon is optionally followed by two management timeslots, for downlink (MD) and uplink (MU) transmission. The MU timeslot is contended among the end-nodes through a CSMA mechanism. The remaining part of the superframe contains uplink (U) and bidirectional (B) timeslots for data transmission. The uplink ones are used by the end-nodes to transmit data to the PAN-Coordinator (some timeslots can also be reserved for retransmissions (R)), while in the bidirectional ones both the PAN-Coordinator and the end-nodes can transmit. The separated group ack timeslot, which provides a bitmap to acknowledge the successful transmissions of the current cycle, is used. Retransmission timeslots come at the end of the uplink timeslots, as shown in Fig. 1, and are not preassigned to nodes, as their assignment is regulated by a distributed algorithm defined in the standard, which depends on the cumulative ack transmitted in each cycle. On each cycle, a node can retransmit the unacknowledged messages at most once. The duration of a timeslot (T_{ts}) is defined in [10] as a function of the maximum expected data payload which can be transmitted by a node plus the overhead introduced by the physical and the MAC layer, as in

$$T_{ts} = \frac{(o+n)sb + \mathsf{IFSpace}}{symbolRate}, \tag{1}$$

where o is the overhead (in bytes) introduced by the physical and MAC layer, n is the maximum expected data payload, sb is the number of symbols per byte, and $\mathsf{IFSpace}$ is the inter-frame space (i.e., 12 symbols for frames shorter than 18 bytes, 40 symbols otherwise).

The number of timeslots in a superframe (N) multiplied by the timeslot duration gives the cycle time (T_s)

$$T_s = N \times T_{ts}. \tag{2}$$

As in the LLDN each node has at least one timeslot assigned, the cycle time grows linearly with the number of nodes. This increases the cycle times in networks with a very large number of nodes. For this reason, in [22] the MC-LLDN was presented. In the MC-LLDN all the nodes in the sub-networks send data to the relevant sub-coordinator. Each sub-coordinator uses a single timeslot to transmit all the data collected from all the nodes belonging to the sub-network, therefore the timeslot size has to be large enough to accommodate in one timeslot all the messages generated within a sub-network. Conversely, in PriMulA the timeslot size can be smaller, as the framework provides the nodes with the ability to embed a configurable number (Ω) of PriMulA messages in a single LL-Data frame (i.e., the standard data frame transmitted from the MAC layer). Combining this feature with message priorities, the timeslot size in PriMulA is set large enough to allow for the transmission of the Ω highest priority messages only. The reduction of the timeslot size in PriMulA shortens the cycle time, which depends on both the number of timeslots and their size, as shown in (2). As a result, when compared with LLDN and MC-LLDN, PriMulA can support applications with shorter message generation periods.

IV. The Priority-based Multichannel-LLDN

The PriMulA framework here proposed, comparing with the standard LLDN, aims to provide several novel features, i.e.,

- Support for meeting the message deadlines, through priority-aware scheduling;
- Shorter cycle times and improved scalability, through a combination of hierarchical topology, multichannel communication and message priority;
- Increased network reliability, through dynamic channel selection.

A. Priority-aware scheduling

In PriMulA, each node maintains at the MAC level a queue of outgoing messages ordered by priority. To efficiently support priority scheduling, PriMulA introduces a novel message, called a PriMulA message, that is embedded in the payload of a standard LL-Data frame. This message consists of the message priority and payload. The PRIO field is encoded in one byte to achieve a good tradeoff between the number of priorities and the overhead of each message, so up to 256 different priorities can be handled. However, a larger PRIO field would not affect the effectiveness of the proposed approach.

In PriMulA, messages are periodically generated by the application, with fixed period (P). For each message, the application defines a lifetime, called a relative deadline (D). D is the maximum time interval, measured at the application layer, within which a generated message has to be consumed. Both P and D of a message depend on the supported application. The relative deadline can be shorter than or equal to period, or greater than the period. PriMuLa adopts a fixed priority assignment, which assigns priorities to messages according to their relative deadlines, i.e., the shorter the relative deadline, the higher the priority.

B. Multichannel communication

PriMulA provides for a hierarchical topology, as shown in Fig. 2, where an HLN with three sub-networks is drawn. Node 0 is the PAN-Coordinator and operates on channel Y, while Nodes 1, 2 and 3 are sub-coordinators and operate on channels W, G and Z, respectively. The sub-networks work in parallel and the sub-coordinators switch between the HLN and their sub-network in fixed timeslots. As shown in Fig. 3, PriMulA provides multiple superframes, one for the HLN and one for each sub-network. All the superframes contain the same number of timeslots and all the timeslots have the same size. In timeslot 1, the PAN-Coordinator transmits its beacon frame to synchronize all nodes in the HLN and Nodes 1, 2 and 3 receive it on channel Y. In timeslot 2, the sub-coordinators switch to their own channels and transmit their beacon. In the next timeslots, the nodes transmit their data to the relevant sub-coordinator. Each node is allowed to transmit up to Ω messages in priority order. Ω is heuristically calculated as a tradeoff between the cycle time and the estimated delay of the messages. Ω calculation is described in detail afterwards in this Section. On timeslot 4, the sub-coordinator 3 switches back to the HLN channel to transmit to the PAN-Coordinator the Ω PriMulA messages with the highest priority gathered from its sub-network nodes. On timeslots 5 and 6 the sub-coordinators 2 and 1, respectively, do the same.

In PriMulA, the minimum number of timeslots (N_{min}) that must be provided in each superframe to allow the periodical transmission of all nodes but the PAN-Coordinator can be calculated as in

$$N_{min} = max_{i=0...C}\left(E_{\mathsf{HLN}} + 2, E_i + 2\right), \tag{3}$$

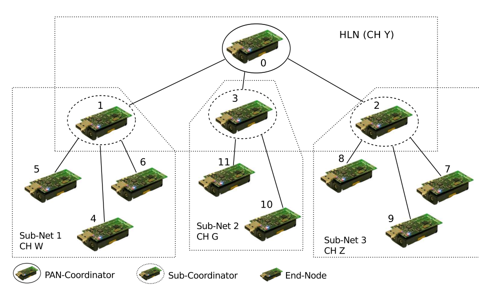

Fig. 2. Example of logical topology.

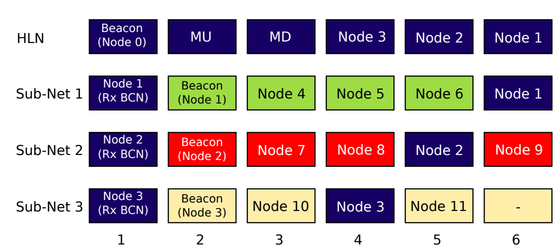

Fig. 3. Example of superframes in the PriMulA network.

where C is the number of sub-coordinators (hence, the number of sub-networks), E_{HLN} represents the number of nodes in the HLN (excluding the PAN-Coordinator), while E_i is the number of nodes in the ith sub-network (including the sub-coordinator). Therefore, there is one timeslot for each node in the HLN, one timeslot for each node in the ith subnetwork and in both cases two additional timeslots are needed, one for the beacon transmission from the PAN-Coordinator and one for the beacon transmission from sub-coordinators, respectively. Such a formula assumes the minimal configuration, i.e., with no management timeslots. The number of subnetworks (C) is heuristically set as a tradeoff between reliability and cycle time. In fact, a high C value limits the number of available channels to be used if the current channel becomes unreliable, while a small C value entails a high number of nodes per subnetwork, a high number of messages to be handled by a sub-coordinator and, consequently, high delays. The number of subnetworks must be in the range [1,15], as the IEEE 802.15.4 Physical Layer [23] provides only 16 channels and one of them is used by the HLN. The number of nodes per subnetwork (E) is also heuristically set. As all the superframes must have the same length, the nodes should be evenly distributed over the subnetworks whenever possible. The timeslot size (in seconds) is determined in the same way as in (1), calculating the maximum expected payload for an LL-Data frame (n) as in

$$n = (\mathsf{PriMulAHdr} + \phi)\,\Omega, \tag{4}$$

where $\mathsf{PriMulAHdr}$ is the PriMulA message header length, ϕ is the maximum expected length of the PriMulA message payload (that depends on the supported application) and Ω is the maximum number of PriMulA messages which can be embedded in a single LL-Data frame. The Ω value is chosen as a tradeoff between cycle time and message delays. On one hand, Equation 4 shows that the timeslot length grows with Ω, so a larger Ω entails a longer cycle time. On the other hand, a large Ω value allows to transmit more messages in one timeslot, thus reducing the waiting times of the messages queued in the nodes. As the standard IEEE 802.15.4 physical layer provides for a maximum payload of 127 bytes, Ω has to be set so as to not exceed the maximum payload allowed. Hence, condition (5) must hold

$$\Omega \in \aleph : 1 \leq \Omega \leq \left\lfloor \frac{127 - o}{(\mathsf{PriMulAHdr} + \phi)} \right\rfloor, \tag{5}$$

where $o = 3$ is the MAC overhead (in bytes) and the right-hand side indicates the maximum number of PriMulA messages that can be transmitted without exceeding the maximum allowed payload. Once the message generation patterns of the nodes are known, and the number of subnetworks as well as the total number of nodes for each subnetwork are set, Ω is chosen as the largest value that maximizes the throughput of the subnetwork with the heaviest load. The Ω value has then to be fed into the timing analysis presented in Sect. V. If the message set is found schedulable, the Ω value is confirmed, otherwise a lower value is chosen (as previously explained) and the timing analysis is run again.

C. Dynamic channel configuration and blacklisting

In PriMulA, during the network configuration, the PAN-Coordinator assigns to each sub-coordinator the channel number of the relevant sub-network. This operation complies with the standard [10], that foresees a generic *Configuration Parameters* field in the command data frame [10] that is exchanged during the configuration phase. To avoid cross-interference [24] between contiguous channels, the PAN-Coordinator first assigns to each sub-network, starting from the HLN, odd-numbered channels in ascending order, and then assigns the even-numbered channels in descending order. These steps maintain the HLN channel far enough from the other sub-network channels to reduce the chance for cross-interference. After the configuration, as foreseen in the LLDN standard [10], the network switches to the *Online State*, in which nodes start data transmission. During this phase, if the channel of a sub-network becomes unreliable, the sub-coordinator changes its state to *Configuration State* and transmits to the PAN-Coordinator, in the uplink management timeslot, a *Configuration status frame*, i.e., a standard command frame indicating that a node requires to be reconfigured. Note that to detect an unreliable channel and activate the Dynamic Channel Configuration (DCC) mechanism, any approach provided in the literature, such as [25], can be adopted. The PAN-Coordinator is aware that the node requiring configuration is a sub-coordinator, therefore it assigns a new channel (chosen from the available ones) to the sub-coordinator that required it and blacklists the unreliable channel. If there are no more available channels, the PAN-Coordinator picks up from the blacklist the least recently blacklisted channel and assigns it to the sub-coordinator. As during the channel switching phase the end-devices of the sub-network that is subject to interference do not receive any beacon, they switch to the *discovery state*. Once the sub-coordinator is assigned a new channel, it also switches to the *discovery state* to find and configure its sub-network nodes. The DCC mechanism here proposed only involves the sub-coordinators and the PAN-Coordinator, so it is transparent to the end-nodes and therefore is compatible with the normal operation of standard LLDN end-nodes.

V. Schedulability analysis

In PriMulA, in the general case, each message is sent twice, i.e., from the end-node to the sub-coordinator and from the sub-coordinator to the PAN-Coordinator. Each node can generate multiple messages belonging to different flows with different (or equal) priorities. The messages within the node are scheduled according to their priority, which depends on the flow relative deadline. If two messages have the same priority (i.e., the same relative deadline), they are transmitted in *First-In First-Out* (FIFO) order. The maximum time RT_i taken by a message of the ith PriMulA flow sent from an end-node to arrive to the PAN-Coordinator is calculated as

$$RT_i = T_{q1_i} + T_{q2_i} + 2T_{x_i}, \tag{6}$$

where T_{q1_i} and T_{q2_i} are the maximum queuing delays experienced by the message in the end-node and in the sub-coordinator, respectively, while T_{x_i} is the message transmission time. In the case where an end-node is directly connected to the PAN-Coordinator, RT_i is given by $RT_i = T_{q1_i} + T_{x_i}$.

The analysis here presented assumes no retransmissions, but it can be extended to consider retransmissions using the separated group ack defined by the standard [10] and described in Sect III. The separated group ack allows messages to be retransmitted once within the same superframe. To calculate the WCRT in this case, a retransmission timeslot for each uplink timeslot has to be assumed. Moreover, in the worst case all the messages are retransmitted, following the same order as the transmissions. Under these hypotheses, the analysis here presented can be applied assuming that messages will be transmitted only in the uplink retransmissions timeslots. Table I summarizes the notations used in the paper. The ith periodic flow is schedulable (i.e., all the messages of the ith flow will arrive to the PAN-Coordinator within their deadline) if condition (7) holds

$$RT_i \leq D_i, \tag{7}$$

where D_i is the ith flow relative deadline. As it was explained in Sect. IV-A, the relative deadlines are constraints imposed by the application timing. In the following, a timing analysis for calculating T_{q1_i}, T_{q2_i}, and T_{x_i} is presented.

TABLE I
PriMulA notation

Symbol	Definition
RT_i	The maximum response time of a PriMulA message of the ith flow.
T_{q1_i}	The maximum time that a PriMulA message of the ith flow waits to be transmitted in the end-node queue.
T_{q2_i}	The maximum time that a PriMulA message of the ith flow waits to be transmitted in the sub-coordinator queue.
T_{x_i}	The transmission time of the ith message.
D_i	The relative deadline of a PriMulA message in the ith flow.
T_{ts}	The timeslot duration.
$s_j(t)$	The maximum number of messages that the jth node can transmit in the $[0, t)$ interval.
T_s	The superframe duration or cycle time.
N	The number of timeslots within a superframe.
$z^{(j)}$	The zth timeslot assigned to the jth node.
$p_z^{(j)}$	The index of the zth timeslot assigned to the jth node.
$z_{worst}^{(j)}$	The zth timeslot assigned to the jth node, which provides the worst case response time for a message if the message arrives shortly after the beginning of this timeslot.
$p_{z_{worst}}^{(j)}$	The index of the z_{worst} timeslot assigned to the jth node.
Γ_j	The number of timeslots assigned to the jth node in a superframe.
Ω	The maximum number of PriMulA messages that a node can transmit within a timeslot.
p_{Γ_j}	The index of the last timeslot assigned to the jth node in the superframe.
$I_i(t)$	The interference experienced by the ith flow due to high-priority flows at the time t.
P_i	The period of a PriMulA message of the ith flow.
X_i	The largest number of consecutive timeslots needed to transmit a message of the ith flow.
$w(X)$	The longest time a node has to wait to see X consecutive timeslots.
C	The number of sub-coordinators.
E	The number of nodes belonging to a sub-network.
p_m	The index of the timeslot assigned to the mth sub-coordinator.
LM	The number of late messages.
RxM	The number of received messages by the PAN-Coordinator.
GM	The overall number of generated messages.

A. Response-time analysis

First, we assume that T_{x_i} is equal to the timeslot duration (T_{ts}), as the latter is configured to allow the complete transmission plus the inter-frame space. Under this assumption, the worst case response time for a message of the ith flow is given by

$$RT_i = T_{q1_i} + T_{q2_i} + 2T_{ts}. \tag{8}$$

To calculate T_{q1} and T_{q2} both the position within the superframe of the timeslots assigned to the sender end-node and the interference of high-priority messages have to be considered.

Calculating T_{q1}: To determine T_{q_1}, the worst case response-time (WCRT) analysis is made calculating both the resource availability and the worst case for the resource request. In PriMulA the resource availability for the jth node is given by the number of timeslots in a super-frame in which the node can transmit, while the resource request is the number of timeslots needed to transmit the messages generated by the node.

The analysis here presented, which applies to fixed priority non-preemptive scheduling, is based on the worst case response-time (WCRT) analysis that was proposed by Joseph and Pandya [26] and extended by Lehoczky with the "busy period" approach [27], and on the findings in Davis et al. [28]. Consequently, the response times of all messages of a flow within a busy period have to be examined. The busy period is defined as the maximum interval during which any message of priority lower than the priority of the ith flow is unable to start transmission. The busy period starts at the time t^s in which a message with a priority higher than or equal to the one of the ith flow is enqueued and there are no messages with priority higher than or equal to the ith flow waiting to be transmitted that were queued strictly before t^s. The busy period ends at the earliest time t^e in which there are no messages of priority equal or greater than the ith flow waiting to be transmitted that were queued strictly before time t^e. With this definition all the messages with the same or higher priority than the ith flow that were enqueued before the end of the busy period are transmitted during the busy period [28].

In PriMulA the resource availability in the interval [0,t) is calculated as the maximum number of messages ($s_j(t)$) that the jth node can transmit in the interval $[0, t)$. In Fig. 4 an example of the $s(t)$ function for a generic Node A with $\Omega = 2$ is shown. In the Figure, at $t_0 = 0$ a superframe starts. From this instant, the time that the Node A has to wait for the arrival of its first timeslot ($z = 1$), i.e., the time interval between the start of the superframe and the time at which the first timeslot assigned to node A starts, is $(p_1^A - 1)T_{ts}$, where p_1^A is the position of the first timeslot assigned to the Node A (i.e., $p_1^A = 2$ in Fig. 4). At this time the Node A can start transmitting up to Ω messages. Then Node A waits for its second timeslot ($z = 2$) that, in the case of Fig. 4, from the beginning of the same superframe starts at $(p_2^A - 1)T_{ts} = (4-1)T_{ts}$. Hence, Node A can transmit other Ω messages. This pattern periodically repeats with the superframe period T_s. Hence, $s_j(t)$ is given by

$$s_j(t) = \sum_{z=1}^{\Gamma_j} \left\lfloor \frac{t + T_s - (p_z^{(j)} - 1)T_{ts}}{T_s} \right\rfloor \Omega, \tag{9}$$

where $(p_z^{(j)} - 1)T_{ts}$ is the offset of the zth timeslot from the beginning of the superframe. Therefore $[t + T_s - (p_z^{(j)} - 1)T_{ts}]/T_s$ is the number of timeslots assigned to the jth node available in the time interval $[0, t)$. The floor operator is intended to make the function stepwise, as any transmission can only start at the beginning of the corresponding timeslot. In fact, $s_j(t)$ is increased by Ω (in Fig. 4, $\Omega = 2$) only at the beginning of each timeslot. Finally, the summation is required to consider all the timeslots assigned to the jth node within a superframe. As node transmissions do not collide with each other, thanks to the TDMA mechanism, for the T_{q1} calculation only the interference of the messages generated within the same node has to be analyzed. The worst case arrival time for a message of the jth node occurs when the message arrives shortly after the beginning of the timeslot ($p^{(j)}_{z_{worst}}$) that, among those assigned to node j, is the furthest one from the next timeslot for the same node.

To calculate T_{q_1}, the longest time ($w_j(X)$) the jth node has to wait to see X consecutive timeslots is calculated as

$$w_j(X) = QT_s + \left(p^{(j)}_{\lceil \frac{(R+1)}{\Omega} \rceil} - p^{(j)}_{z_{worst}} \right) T_{ts}, \tag{10}$$

where Q is the quotient of the Euclidean division of $X - 1 + (z_{worst} \cdot \Omega)$ by $\Gamma_j \Omega$ and R is the reminder, i.e., $[X - 1 + (z_{worst} \cdot \Omega)] = Q\Gamma_j\Omega + R$.

For instance, in the case of Fig. 4, Node A has $\Gamma_A = 2$, $\Omega = 2$ and $p^{(A)}_{z_{worst}} = 2$ (i.e., $z^{(A)}_{worst} = 1$), so for $X = 5$, we have $Q = 1$ and $R = 3$. Therefore, Node A has to wait for $w(5) = T_s + (p_2^{(A)} - p_1^{(A)})T_{ts} = T_s + (4-2)T_{ts} = T_s + 2T_{ts}$. In fact, looking at the example in Fig. 4, if the first message arrives at the beginning of timeslot 2, it will be transmitted in timeslot 4, thus the time at which Node A can start transmitting its 5th message is $t = T_s + 2T_{ts}$.

To apply the busy period approach. First, we calculate the number of messages of the ith flow generated in any interval of length t, i.e.,

$$I_i(t) = \left\lceil \frac{t}{P_i} \right\rceil. \tag{11}$$

Then, we calculate the largest number (X_i) of timeslots that are needed to transmit the ith message, which is given by the smallest value of X_i that satisfies

$$X_i = 1 + \sum_{\mathrm{pri}(h) > \mathrm{pri}(i)} I_h(w_j(X_i)) \quad + \sum_{\substack{\mathrm{pri}(h) = \mathrm{pri}(i) \\ h \neq i}} 1, \tag{12}$$

where 1 is the timeslot required to transmit the message, while the parameter (X_i) encompasses the following sources of interference:

- The hth higher priority messages (i.e., $\mathrm{pri}(h) > \mathrm{pri}(i)$).
- The messages with the same priority as the ith message that compete with it in the end-node queue (in this case a FIFO policy is adopted).

Equation (12) has not simple solution, as the X_i term appears both in the Left-hand-side (LHS) and in the Right-hand-side (RHS) of the equation under the ceiling operator. Thus, the calculation of X_i is performed by the following iterations

$$\begin{cases} X_i^{(0)} & = 1 \\ X_i^{(k+1)} & = 1 + \sum_{\mathrm{pri}(h) > \mathrm{pri}(i)} I_h(w_j(X_i^{(k)})) \\ & \quad + \sum_{\substack{\mathrm{pri}(h) = \mathrm{pri}(i) \\ h \neq i}} 1 \end{cases}. \tag{13}$$

Iteration starts with $X_i^{(k)} = 1$, with $k = 0$, as the message to be transmitted requires one timeslot. Then $X_i^{(k+1)}$ is iteratively calculated until the LHS is equal to the RHS, i.e., until the interference stops growing. Equation (13) is proved to converge if the timeslots required to transmit every message of the jth node is lower than or equal to the number of timeslots available for the jth node [26]. This way, T_{q1_i} for the ith message is calculated as $T_{q1_i} = w_i(X_i^{(k)})$.

Calculating T_{q2}: To compute T_{q2}, all the messages that a sub-coordinator receives from the nodes belonging to its sub-network plus the messages generated by the sub-coordinator itself are considered. The $s_m(t)$, $I_i(t)$ and $w_m(X)$ functions can be calculated using Equations (9), (11), and (10), but taking into account, instead of p_j, the timeslot index (p_m) of the mth sub-coordinator in the HLN superframe, as the messages in the sub-coordinator queue are generated by all the sub-network nodes.

Applying the busy period approach, the largest number (X_i) of timeslots that are needed to transmit a message of the ith flow is calculated as in

$$\begin{cases} X_i^{(0)} & = 1 \\ X_i^{(k+1)} & = 1 + \sum_{\mathrm{pri}(h) > \mathrm{pri}(i)} I_h(w_j(X_i^{(k)})) \\ & \quad + \sum_{\substack{\mathrm{pri}(h) = \mathrm{pri}(i) \\ t_{start}(h) < t_{start}(i)}} I_h(w_j(X_i^{(k)})) \\ & \quad + \sum_{\substack{\mathrm{pri}(h) = \mathrm{pri}(i) \\ h \neq i}} 1 \end{cases}. \tag{14}$$

The ($X_i^{(k+1)}$) parameter encompasses the following sources of interference:

- The messages with a higher priority than the ith message (i.e., $\mathrm{pri}(h) > \mathrm{pri}(i)$).

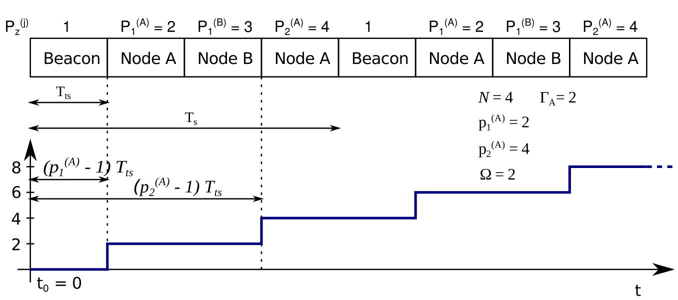

Fig. 4. Example of $s_j(t)$ function: the $s_1(t)$.

- The messages with the same priority as the ith message that are either transmitted to the sub-coordinator in the timeslots preceding the one in which the ith message is transmitted or that are transmitted in the same timeslot as the ith message, but before it (ties are broken in a FIFO way) within.
- The messages with the same priority as the ith message generated by the sub-coordinator.

This way, T_{q2_i} for the ith message is calculated as $T_{q2_i} = w_m(X_i^{(k)})$.

According to [26], Equation (14) is proved to converge if the number of timeslots needed to transmit all the sub-network messages is lower than or equal to the number of timeslots available for the relevant sub-coordinator.

TABLE II
DATA MESSAGES CONFIGURATION

Message ID	Period P (ms)
1	100ms
2	250ms
3	450ms

TABLE III
LLDN CONFIGURATION

# of Nodes	T_{ts} (ms)	N	T_s (ms)	Ω	Workload (kb/s)	DMR
20	2.656	21	55.776	3	46.72	0
30	2.656	31	82.336	3	70.08	0
40	2.080	41	85.280	2	93.44	0.35%
(NSP) 45	2.080	51	95.680	2	105.12	0.67%

VI. SIMULATION SCENARIO

This section presents a comparative assessment, obtained through simulations, of the LLDN, the MC-LLDN proposed in [22] and PriMulA. A simulation model was developed using the OMNeT++ framework and the inetmanet-2.0 library. The aim is to evaluate the network scalability and reliability, while taking into account the message deadlines. The simulated PHY layer uses the DSSS O-QPSK modulation and operates at 250 kbps in the 2.4 GHz frequency band.

To assess scalability two metrics are defined. The first one is the network saturation point, which represents the maximum number of nodes for which condition (15) is met, under the assumption that all the nodes generate the same traffic

$$\sum_{i=1}^{NumFlows} \frac{1}{P_i} < \sum_{j=1}^{E} \frac{\Gamma_j \Omega}{T_s}, \forall \text{ sub-network.} \tag{15}$$

In condition (15) the LHS is the maximum achievable workload generated in a sub-network and the RHS is the overall sub-network throughput, which takes into account that each end-node or sub-coordinator can transmit up to $\Gamma_j \Omega$ messages every cycle time.

The second metric is the Deadline Miss Ratio, i.e., the ratio of the number of late messages (LM) to the overall number of messages received (RxM) by the PAN-Coordinator, as in

$$DMR = \frac{LM}{RxM}. \tag{16}$$

Two other metrics are used for the reliability assessment. The first one is the Packet Loss Ratio (PLR), defined as in

$$PLR = 1 - \frac{RxM}{GM}, \tag{17}$$

where RxM is the number of messages correctly received from the PAN-Coordinator and GM is the number of messages transmitted to the PAN-Coordinator. The second is the reconfiguration time, which is the time the network takes to reconfigure itself after a DCC run. A realistic networked control system scenario was set up, in which each node generates and periodically transmits 18-byte data messages. The message periods are shown in Table II. In all the simulations the relative deadline of a message is considered equal to the message period. For the sake of generality, the message periods are not multiple of each other and belong to the range [100,500]ms, to comply with the data sampling periods that are typically found in realistic networked control system applications [29].

A. Scalability assessment

In this scenario, each node periodically transmits to the PAN-Coordinator three kinds of messages. In each run the number of nodes is increased to approach the situation in which the network workload is higher than the maximum achievable throughput. In this assessment no retransmissions and errors are considered, as the aim here is to compare the scalability of the three different approaches, while their reliability is dealt with in Sect. VI-B. In the LLDN network configuration, the superframe provides one timeslot for each node and messages are transmitted in a FIFO order. In both the MC-LLDN and PriMulA networks, instead, there are multiple superframes. The aim of the simulations is to provide some insight on the number of nodes that can be supported without reaching saturation even in the case of deadline miss. For this reason and for the sake of scalability, in all the assessed protocols the network configuration is not based on the worst case analysis, but is heuristically chosen so as to maximize the network throughput while maintaining the lowest possible cycle times and, consequently, to obtain the lowest end-to-end delays.

In the simulated scenario PriMulA is schedulable with up to 40 nodes. When the number of nodes is 50, the lowest priority flows miss their deadline, as their WCRT is equal to 0.5208s, while their relative deadline is 0.45s.

Table III shows the LLDN configuration parameters and results. As the messages transmitted by each node in every configuration are the same, in terms of period and length, the timeslot duration depends on the Ω parameter only. We refer to the notation in Table I. Table IV shows the MC-LLDN and PriMulA network configuration parameters and results. For the configurations in which the remainder of the division of *Number of nodes* by C is equal to 1, the spare node is assigned directly to the HLN (e.g., this occurs in the PriMulA configuration with 50 nodes). Moreover, in the PriMulA configuration the unused timeslots in a sub-network are assigned as a second timeslot to the nodes belonging to the sub-network (starting from the end-node which is assigned the first uplink timeslot) to minimize the message

TABLE IV
MC-LLDN AND PRIMULA CONFIGURATIONS AND RESULTS

MC-LLDN configuration							
# of Nodes	C	E	N	T_s (ms)	Ω	**Workload (kb/s)**	**DMR**
20	5	4	7	23.520	4	46.72	0
30	6	5	8	31.744	5	70.08	0.002%
40	8	5	10	39.680	5	93.44	0.025%
50	10	5	12	47.616	5	116.80	1.61%
60	10	6	12	54.912	6	140.16	7.43%
(NSP) 67	11	6	13	59.488	6	156.51	16.89%
PriMulA configuration							
# of Nodes	C	E	N	T_s (ms)	Ω	**Workload (kb/s)**	**DMR**
20	5	4	7	10.752	1	46.72	0
30	5	6	7	15.008	2	70.08	0
40	7	6	9	24.768	3	93.44	0
50	7	7	9	30.240	4	116.80	0
57	8	7	10	45.760	6	133.15	0
64	9	7	11	50.336	6	149.50	0.03%
(NSP) 70	14	5	16	73.216	6	163.52	1.79%

waiting time in the node queue. On the contrary, in the MC-LLDN approach the unused timeslots are not assigned, as the sub-coordinators within a superframe have to transmit one message for each timeslot in the sub-network. Note that the $max(\Omega) = 6$, as higher values would exceed the maximum payload that can be transmitted by the physical layer (i.e., 127 bytes). The configurations in Tables III and IV already provide some insights on the scalability of the three networks. The network saturation point (NSP) of the addressed approaches significantly varies. It goes from 45 nodes for the LLDN up to 67 and 70 nodes for the MC-LLDN and PriMulA, respectively. Tables III and IV show that PriMulA outperforms both the LLDN and the MC-LLDN in terms of cycle times (T_s), as shorter sampling periods and end-to-end delays (i.e., from the sensor to the PAN-Coordinator) are achieved.

As far as the DMR is concerned, results in Tables III and IV show that PriMulA outperforms the LLDN in terms of scalability. In fact, PriMulA allows up to 57 nodes in the overall network without deadline miss and 70 nodes with a deadline miss of 1.79%, while the LLDN allows up to 30 nodes without deadline miss and up to 45 nodes with a deadline miss of 0.67%. The MC-LLDN approach supports a high number of nodes before reaching saturation, but only 20 nodes without deadline miss (Table IV). The reason for this is that the MC-LLDN does not provide any prioritization mechanism, hence there is no way to prioritize the messages with shorter relative deadlines. Moreover, Table IV shows that when the number of nodes is less than 60 the MC-LLDN cycle times are higher than those of PriMulA. This is because in the MC-LLDN, in each cycle, the sub-coordinators have to transmit at least one message for each sub-network node (i.e., $\Omega = E$), hence timeslots are larger.

B. Reliability assessment

Here the standard LLDN protocol and PriMulA are comparatively assessed in a scenario characterized by packet loss and interference. MC-LLDN is not evaluated, as in [22] retransmissions were not addressed.

Packet Loss and Deadline Miss Ratio: Simulations refer to a scenario with 20 nodes randomly placed in a sensing area of 100m x 100m, with the PAN-Coordinator in the center. Simulations were performed with and without retransmissions. The PriMulA sub-coordinators are chosen among the nodes that are placed halfway between their sub-network and the PAN-Coordinator. The result of this choice is that the distances between the PAN-Coordinator and the sub-coordinators are always lower than 20m. Retransmissions are configured using the *separated group ack* defined by the standard [10]. The retransmission timeslots are placed after the uplink ones, so the corrupted messages can be retransmitted within the same superframe. The LLDN protocol retransmissions are scheduled according to the algorithm defined for this purpose in the standard [10]. In this scenario, each node runs an application that generates two messages, m_1 and m_2, with a 16-byte payload and periods $P_1 = 100$ms and $P_2 = 250$ms, respectively. The superframe configurations for both approaches are shown in Table V. In the case of retransmissions enabled, the PriMulA network consists of seven sub-networks, each one composed of a maximum of three nodes. With $\Omega = 3$, all the highest priority messages will be transmitted by the sub-coordinator within one superframe. For this configuration of PriMulA, the schedulability analysis presented in Sect. V gives a worst case response time of 94.72ms for the messages with 100ms period and of 186.88ms for the messages with 250ms period. Packet loss was obtained applying the Log-normal shadowing channel model with $n = 2.04$ and $\sigma = 6.7$, as these values are realistic for the industrial context [30]. This model was chosen as it is adopted in large scale indoor industrial environments [30]. Simulations were repeated six times varying only the seed used for random numbers generation. The simulated time was set equal to 300s to collect statistics on more than 10^5 messages, while the other parameters were unchanged. At the end of simulations, the queues in all the nodes were empty, so all the generated messages were transmitted. Fig. 5 shows the simulation results in terms of PLR with and without retransmissions. Results show that with retransmissions both the LLDN and PriMulA obtain PLR values lower than 1%, with negligible differences between them. Without retransmissions the PLR for both approaches increases and PriMulA provides a lower PLR than the LLDN although the messages are transmitted twice. In PriMulA the PLR measured on the HLN (i.e., on the links from the sub-coordinators to the PAN-Coordinator) in the case of no retransmissions is very close to zero (i.e., 0.05%), as the sub-coordinators are placed nearby the PAN-Coordinator. Conversely, the PLR of the links between the end-nodes and the sub-coordinators is higher than 2%, as the end-nodes are placed at a longer distance from the sub-coordinators. The

TABLE V
SUPERFRAME CONFIGURATIONS

Retransmissions not enabled								
Network	Mngt.	Uplink	Retr.	N	C	$max(E)$	Ω	T_s (ms)
PriMulA	2	5	0	7	4	5	1	10.30
LLDN	0	20	0	21	-	-	1	30.24
Retransmissions enabled								
Network	Mngt.	Uplink	Retr.	N	C	$max(E)$	Ω	T_s (ms)
PriMulA	2	15	7	18	7	3	3	46.08
LLDN	0	41	20	42	-	-	2	81.98

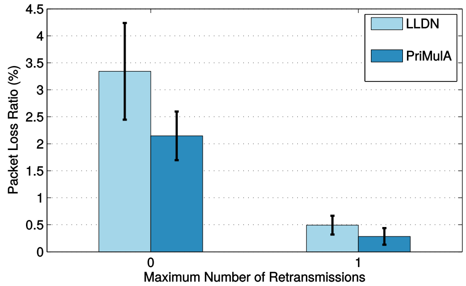

Fig. 5. Packet Loss Ratio.

TABLE VI
DEADLINE MISS RATIO

Retransmissions not enabled				
Network	**Avg. DMR**	**Min. DMR**	**Max. DMR**	**Std. Dev.**
PriMulA	0%	0%	0%	0%
LLDN	0.015%	0.003%	0.028%	0.010%
Retransmissions enabled				
Network	**Avg. DMR**	**Min. DMR**	**Max. DMR**	**Std. Dev.**
PriMulA	0.42%	0.24%	0.65%	0.17%
LLDN	3.38%	2.22%	4.77%	0.94%

distances between transmitters and receiver in the PriMulA scenario, on average, are lower than in the LLDN scenario, as in PriMulA the sub-coordinators are chosen among the nodes that are placed halfway between the sub-network nodes and the PAN-Coordinator, while in the LLDN nodes directly transmit to the PAN-Coordinator. Hence it is the distance between the nodes the parameter that has the major influence on the PLR in the simulation. Different channel models than the Log-normal shadowing might lead to different results, however, the comparison between different channel models is out of the scope of this paper.

As far as the DMR is concerned, the results in Table VI show that when retransmissions are not enabled no deadline miss was experienced by PriMulA and the DMR for LLDN is negligible. However, when retransmissions are enabled, PriMulA outperforms the standard LLDN. In fact the DMR for the PriMulA network is always lower than 1%, while in the LLDN it reaches 4.77% in the worst case and 2.22% in the best one. Such a result is due to the higher cycle time of the LLDN. In fact, when a beacon is lost, all the messages have to wait in the node queue for the next available timeslot in the next cycle, thus the waiting time for the standard LLDN is increased by a cycle time (i.e., 81.98ms) and the messages arrive after their deadline. Conversely, in PriMulA the network cycle time is 46.08ms and consequently a message has to wait a shorter time than in the LLDN.

Network reconfiguration time: For the sake of comparison PriMulA is compared with an improved LLDN that implements the DCC mechanism here proposed. To assess the DCC mechanism, an interferer node was added and the simulations were repeated in the same scenario. The interferer node, after 90s of simulation time, starts to transmit data with an exponentially distributed interarrival time between consecutive transmissions. A simple online algorithm based on the packet loss observed is used to activate the DCC mechanism, but any approach in the literature able to detect an unreliable channel can be adopted. Both the PAN-Coordinator and the sub-coordinators collect the number (RX) of LL-Data frames that arrived within a given number of superframes (*DCC_SfN*), chosen according to the channel quality variability. The DCC procedure is triggered if this number is below a given fraction (η) of the average RX_i, as in

$$RX_i < \eta * avg(RX_{n=1..(i-1)}), \quad (18)$$

where i indicates the ith superframe, while $\eta \in (0, 1]$. In the PriMulA scenario, the DCC was assessed in two cases, i.e., with the interferer on the HLN channel and on a sub-network channel, respectively. In the standard LLDN scenario the interferer affects the channel on which the network operates. Simulation parameters are summarized in Table VII. The assessed metric is the reconfiguration time, which is given by the sum of the durations of the discovery and of the configuration phase. Such a value depends on the discovery time defined as a parameter in the IEEE 802.15.4e amendment [10]. In the simulations, the discovery time is set to the minimum value that allows the discovery of all the subnetwork nodes. Such a value indicates the duration of the discovery state. Table VIII shows that the reconfiguration time is dominated by the discovery phase (whose duration is shown in Table VII), while the time for the configuration phase is negligible. PriMulA results show that if the interferer affects a sub-network (SN in Table VIII), the reconfiguration time is close to the discovery time of the sub-network, while if the interferer is on the HLN channel, the entire network is reconfigured, so the discovery time is close to the sum of the discovery time in the HLN plus the discovery time in the sub-network. In the LLDN with DCC, the reconfiguration time is comparable with that of a PriMulA sub-network.

TABLE VII
DCC ASSESSMENT PARAMETERS

Network	Interf. Ch.	η	*DCC_SfN*	Discovery Time
PriMulA	11	0.905	11 (HLN)	PanCoord: 16s, SubCoord: 32s
PriMulA	17	0.667	11	PanCoord: 16s, SubCoord: 32s
LLDN	11	0.667	6	32s

VII. IMPLEMENTATION OF PRIMULA ON REAL DEVICES

A proof-of-concept PriMulA implementation was realized to prove the PriMulA feasibility on COTS devices. The devices are the TelosB motes equipped with a CC2420 radio transceiver, which provides an IEEE 802.15.4-compliant physical layer that adopts the 2.4 GHz DSSS O-QPSK PHY with a datarate of 250 kbps. The implementation is realized

TABLE VIII
DCC RECONFIGURATION TIMES

Network	Avg. Reconf. Time	Conf. Interval 95%
PriMulA (HLN)	48.52s	±0.99s
PriMulA (SN)	33.76s	±2.21s
LLDN	34.50s	±0.002s

using TinyOS. The considered scenario comprises one PAN-Coordinator, three end-devices and two sub-coordinators, in charge of two sub-networks. The traffic patterns are taken from a realistic industrial scenario [31], N is equal to 6 and the timeslot (T_{ts}) is equal to 1.3 ms, a value very close to the theoretical one (1.2ms). All the statistics were collected every 30s. Under a workload of 3.8 kbps, an average throughput of 3.6 kbps was achieved in a noisy and error-prone environment. The time for the sub-coordinator to commute between the HLN and the sub-network channel was measured equal to 213.6μs and is comparable with the one in the standard (192μs). In the HLN, end-devices and sub-coordinators co-exist and the multichannel approach allows for multiple sub-networks operating in parallel without interferences.

VIII. Conclusions

PriMulA improves the LLDN in several respects. Comparative simulations in realistic scenarios show that the number of nodes that the PriMulA network can support without reaching saturation is increased by 56% compared to the LLDN. The introduction of message priorities not only reduces the cycle time compared to the LLDN and to the MC-LLDN, but it also provides a lower deadline miss ratio. The proof-of-concept implementation on the TelosB motes proves the implementation feasibility without any hardware modification. Future work will address the PriMulA implementation on more complex testbeds, the stochastic analysis [32] and the support for event-driven transmissions.

References

[1] H. Le, J. Van Eck, and M. Takizawa, “An Efficient Hybrid Medium Access Control Technique for Digital Ecosystems,” *IEEE Trans. Ind. Electron.*, vol. 60, no. 3, pp. 1070–1076, Mar. 2013.

[2] H. Yan, Y. Zhang, Z. Pang, and L. D. Xu, “Superframe Planning and Access Latency of Slotted MAC for Industrial WSN in IoT Environment,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1242–1251, May. 2014.

[3] W. Shen, T. Zhang, F. Barac, and M. Gidlund, “PriorityMAC: A Priority-Enhanced MAC Protocol for Critical Traffic in Industrial Wireless Sensor and Actuator Networks,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 824–835, Feb. 2014.

[4] V. C. Gungor, B. Lu, and G. P. Hancke, “Opportunities and Challenges of Wireless Sensor Networks in Smart Grid,” *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3557–3564, Oct. 2010.

[5] O. Kreibich, J. Neuzil, and R. Smid, “Quality-Based Multiple-Sensor Fusion in an Industrial Wireless Sensor Network for MCM,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4903–4911, Sept. 2014.

[6] F. Dobslaw, T. Zhang, and M. Gidlund, “End-to-End Reliability-Aware Scheduling for Wireless Sensor Networks,” *IEEE Trans. Ind. Informat.*, vol. PP, no. 99, Dec. 2014.

[7] V. C. Gungor and G. P. Hancke, “Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, 2009.

[8] E. Toscano and L. Lo Bello, “Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks,” *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 337–350, May. 2012.

[9] H. Y. Tung, K. F. Tsang, K. T. Chui, H. C. Tung, H. R. Chi, G. P. Hancke, and K. F. Man, “The generic design of a high-traffic advanced metering infrastructure using ZigBee,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 836–844, 2014.

[10] *IEEE Std. 802.15.4e-2012, ¨Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 1: MAC sublayer¨*, 2012.

[11] T. Paso, J. Haapola, and J. Iinatti, “Feasibility Study of IEEE 802.15.4e DSME Utilizing IR-UWB and S-Aloha,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, London, UK, Sept. 2013, pp. 1863–1867.

[12] G. Alderisi, G. Patti, O. Mirabella, and L. Lo Bello, “Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios,” in *IEEE Int. Conf. on Industrial Informatics (INDIN)*, Cambridge, UK, Jul. 2015.

[13] M. Palattella, N. Accettura, L. Grieco, G. Boggia, M. Dohler, and T. Engel, “On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH,” *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.

[14] E. Toscano and L. Lo Bello, “A multichannel approach to avoid beacon collisions in IEEE 802.15.4 cluster-tree industrial networks,” in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Mallorca, Spain, Sept. 2009.

[15] T. Semprebom, R. Moraes, C. Montez, P. Portugal, and F. Vasques, “Quality of Service Provision Assessment for DDBP Approach in IEEE 802.15.4 Networks,” in *IEEE Int. Conf. on Industrial Informatics (INDIN)*, Porto Alegre, Brazil, Jul. 2014, pp. 118–123.

[16] G. A. Shah, V. C. Gungor, and O. B. Akan, “A Cross-Layer QoS-Aware Communication Framework in Cognitive Radio Sensor Networks for Smart Grid Applications,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 3, pp. 1477–1485, Aug. 2013.

[17] M. Collotta, A. Lo Cascio, G. Pau, and G. Scata, “A fuzzy controller to improve CSMA/CA performance in IEEE 802.15.4 industrial wireless sensor networks,” in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Cagliari, Italy, Sept. 2013.

[18] A. Willig and E. Uhlemann, “Deadline-aware scheduling of cooperative relayers in TDMA-based wireless industrial networks,” *Wireless Networks*, vol. 20, no. 1, pp. 73–88, Jan. 2014.

[19] D. Yang, Y. Xu, H. Wang, T. Zheng, H. Zhang, H. Zhang, and M. Gidlund, “Assignment of Segmented Slots Enabling Reliable Real-Time Transmission in Industrial Wireless Sensor Networks,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3966–3977, Jun. 2015.

[20] L. Dariz and M. Malaguti, G. Ruggeri, “Performance analysis of IEEE 802.15.4 real-time enhancement,” in *IEEE Int. Symp. on Industrial Electronics (ISIE)*, Istanbul, Jun. 2014, pp. 1475–1480.

[21] *IEEE Std. 802.15.4k-2013, ¨Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), Amendment 5: Physical Layer Specifications for Low Energy, Critical Infrastructure Monitoring Networks¨*, 2013.

[22] G. Patti, G. Alderisi, and L. Lo Bello, “Introducing multi-level communication in the IEEE 802.15.4e protocol: the MultiChannel-LLDN,” in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Barcelona, Spain, Sept. 2014, pp. 1–8.

[23] *IEEE Std. 802.15.4-2011 - IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks*.

[24] L. Lo Bello and E. Toscano, “Coexistence Issues of Multiple Co-Located IEEE 802.15.4/ZigBee Networks Running on Adjacent Radio Channels in Industrial Environments,” *IEEE Trans. Ind. Informat.*, vol. 5, no. 2, pp. 157–167, May. 2009.

[25] D. Yang, Y. Xu, and M. Gidlund, “Coexistence of IEEE802.15.4 based networks: A survey,” in *Conf. of IEEE Industrial Electronics Society (IECON)*, Glendale, USA, Nov. 2010, 2107-2113.

[26] M. Joseph and P. K. Pandya, “Finding Response Times in a Real-Time System,” *The Computer Journal*, vol. 29, no. 5, pp. 390–395, Oct. 1986.

[27] J. Lehoczky, “Fixed Priority Scheduling of Periodic Task Sets With Arbitrary Deadlines,” in *IEEE Real-Time Systems Symposium (RTSS)*, Lake Buena Vista, FL, Dec. 1990, pp. 201–209.

[28] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, “Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised,” *Real-Time Systems*, vol. 35, p. 239–272, Jan. 2007.

[29] M. García-Rivera and A. Barreiro, “Analysis of networked control systems with drops and variable delays,” *Automatica*, vol. 43, no. 12, pp. 2054–2059, 2007.

[30] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme, “The Industrial Indoor Channel: Large-Scale and Temporal Fading at 900, 2400, and 5200 MHz,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 7, pp. 2740–2751, Jul. 2008.

[31] J. Akerberg, M. Gidlund, and M. Bjorkman, “Future research challenges in wireless sensor and actuator networks targeting industrial automation,” in *IEEE Int. Conf. on Ind. Informat. (INDIN)*, Jul. 2011.

[32] G. Kaczynski, L. Lo Bello, and T. Nolte, “Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems,” in *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Patras, Greece, Sept. 2007, pp. 101–110.

Lucia Lo Bello (M'02-SM'09) is an Associate Professor at the University of Catania, Italy. She received the M.S. degree in electronic engineering and the Ph.D. degree in computer engineering in 1994 and 1998, respectively. She authored or coauthored more than 140 technical papers in the area of real-time embedded systems, industrial automation networks and wireless sensor networks.

Gaetano Patti (M'13) received the M.S. degree (*summa cum laude*) in computer engineering from the University of Catania in 2013. He is currently working towards the Ph.D. degree in Computer Engineering. His research interests include real-time industrial networks, Wireless Sensor Actuators Networks (WSANs), powerline communications and networks for mobile robots applications.