SchedWiFi: An Innovative Approach to support Scheduled Traffic in Ad-hoc Industrial IEEE 802.11 networks

Gaetano Patti, Giuliana Alderisi, Lucia Lo Bello
Department of Electrical, Electronic and Computer Engineering
University of Catania
Catania, Italy
{gaetano.patti, giuliana.alderisi, lobello}@unict.it

Abstract

The spread use of IEEE 802.11 networks in industrial automation raised the need to support multiple traffic classes with different requirements. This paper proposes a novel approach, called SchedWiFi, that provides flexible support to the scheduled traffic class, i.e., a high priority traffic class that is transmitted according to a fixed schedule, over IEEE 802.11 ad-hoc industrial networks. SchedWiFi operates on the IEEE 802.11n physical layer, thus providing high datarate, and supports multiple traffic classes with different priorities. SchedWiFi modifies the EDCA mechanism allowing to transmit scheduled traffic without requiring any predefined superframe structure, or timeslots, thus allowing for more flexible schedule of non-ST traffic.

Index Terms

Real-time networks, Scheduled Traffic, Industrial Automation, IEEE 802.11, Ad-hoc networks.

1. Introduction and Motivation

Wireless industrial networks have to cope with the requirements that industrial applications impose on communication [1]. Real-time capabilities, i.e., bounded delays for the time-sensitive traffic classes, are one of the properties that these networks have to provide. One of the most commonly adopted wireless technologies for industrial automation applications is the IEEE 802.11 standard [2], that specifies mechanisms to achieve QoS requirements. Among them, the Enhanced Distributed Channel Access (EDCA), which introduces QoS support in the IEEE 802.11 standard providing traffic prioritization, the Hybrid Coordination Function Controlled Channel Access (HCCA), which offers parameterized QoS support, and the Mesh Coordination Function Controlled Channel Access (MCCA), that allows nodes to access the channel at selected times with low contention. However, these mechanisms are not suitable for real-time traffic support, as shown in [3], [4], [5], [6].

For this reason, several approaches were proposed in the literature to support real-time communications over wireless industrial networks. Most of them either operate in managed mode or are based on TDMA mechanisms. However, access points limit the node mobility as, in order to maintain connectivity, there must be an access point in the area a node is moving to, while TDMA-based approaches offer a limited flexibility for the transmission of dynamically added traffic flows (e.g., aperiodic flows).

This paper considers a particular class of real-time traffic, called Scheduled Traffic (ST), which is a high priority traffic class that is transmitted according to a fixed schedule. The paper proposes a novel approach, here called SchedWiFi, that provides a flexible support to the Scheduled Traffic over IEEE 802.11 ad-hoc networks. The aim of SchedWiFi is to provide ST flows with low and bounded end-to-end latency, very low jitter and low packet loss. SchedWiFi modifies the EDCA mechanism introducing the possibility to transmit ST flows in a way that prevents any interference from non-ST traffic thanks to both the introduction of the concept of ST Window and the use of a Time-Aware Shaper (TAS). The ST Window of an ST flow represents the time window in which the transmission of the ST flow is foreseen. The TAS blocks all the transmissions which could interfere with the ST Windows of any ST flow. Thanks to the TAS, the ST Windows of ST flows are temporally isolated, i.e., the medium access is granted to each ST flow in an exclusive way, i.e., no other ST or non-ST flows can transmit within the ST Window of the considered flow.

978-1-4673-7929-8/15/$31.00

One main feature of SchedWiFi is that it does not require any predefined superframe structure (as the time is not partitioned in superframes) or timeslots, thus allowing a more flexible transmission of the non-ST flows, which can be transmitted whenever they will not interfere with the ST Window of any ST flow. Moreover, ST Windows have different lengths for different ST flows, based on the flow payload, while when building a superframe the same timeslot size is chosen for all the flows, regardless of the different payloads of their frames. Moreover, SchedWiFi works in ad-hoc mode and does not need an access point or network coordinator, that represents a single point of failure for the network.

The paper is organized as follows. Sect. 2 outlines related works, while Sect. 3 recall the basics of the EDCA mechanism in the IEEE 802.11 standard. Sect. 4 introduces the SchedWiFi approach. Sect.5 addresses the simulation scenario and the traffic model, while Sect. 6 presents the simulation results obtained using the OMNeT++ simulation tool. Finally, Sect. 7 concludes the paper and outlines directions for future work.

2. Related Works

Many protocols in the literature propose methods to handle real-time communications over IEEE 802.11 networks. Most of them operate in managed mode, i.e., they require access points that manage the transmissions. For instance, in [7] and [8] a MAC protocol enabling for real-time communications in IEEE 802.11 networks, called IsoMAC, was proposed. The protocol provides a scheduled phase for process data, in which nodes transmit according to a TDMA mechanism, and a contention phase for best-effort and management traffic. In [9], a TDMA-based approach based on the EDCA was proposed. The protocol operates modifying the Contention Window of the EDCA for real-time traffic, that is transmitted according to a TDMA mechanism. In [10] a TDMA mechanism, called RT-WiFi, that adopts a centralized channel and time management to access the channels according to strict timing schedule, is described. RT-WiFi supports predictable, high-speed wireless control systems. However it cannot operate on ad-hoc networks. The paper in [11] presented a mechanism, called Group Sequential Communication (GSC) that uses a Publish/Subscribe paradigm and improves the HCCA. The mechanisms in [7]–[11] are able to support real-time transmissions, but require access points and are not suitable for ad-hoc networks. In [12] a wireless adaptation of the dominance protocol used in the CAN bus is proposed. However, it does not provide support to scheduled traffic, as the temporal isolation of the highest priority messages is not guaranteed. In [13] and in [14] two approaches for real-time communications based on polling mechanism were presented. Both approaches use an Earliest Deadline First (EDF) scheduler, work on top of the MAC IEEE 802.11 and allow for real-time communications without modifying the standard. However, they introduce a high overhead, due to the polling mechanism. Some approaches to handle real-time traffic on IEEE 802.11 networks in ad-hoc mode (i.e., without the need for access points) were also proposed. In particular, in [15] a Stream Reservation protocol is described. The network nodes use RTS/CTS messages to grant the medium access to a stream. In order to prevent collisions, each node maintains a reservation table that contains the schedule. However, such a protocol is specifically designed for voice traffic and does not address reliability issues. In [16] a distributed mechanism, in which the time is split into two phases, i.e., the scheduled phase and the contention phase, of a so-called virtual frame, is presented. Slots are assigned to the nodes in a distributed way, thus providing collision-free transmissions. The length of the virtual frame is adapted to the workload condition so as to have low delays. The protocol in [16] does not provide any real-time mechanism and may suffer from jitter. Another distributed protocol, called RT-WMP, for real-time communication was investigated in [17]. RT-WMP is specifically designed for mobile robot communications and provides a token passing mechanism. The results proved that RT-WMP works well with a low number of nodes, but it suffers from scalability problems. In [18], an adaptive TDMA protocol is proposed. Such a protocol partitions the time into cyclic temporal windows, in which nodes are allowed to transmit following a TDMA mechanism, and can be used in ad-hoc networks. In [19] a wireless version of the TTEthernet [20] standard that provides support to scheduled traffic is presented. The proposed approach is centralized and needs an access point node. In [21] the real-time behavior is achieved in a topology management protocol which provides bounded delays, while in [22] a load balancing mechanism was proposed to reduce the load in access points and achieve a better QoS. In [23] an overview of QoS protocols for ad-hoc networks is proposed and the importance of the QoS support in IEEE 802.11 network is proved. Finally, in [24] a wireless traffic smoother was proposed to enable soft real-time traffic over IEEE 802.11 wireless networks. The SchedWiFi approach proposed in this work aims to provide ST flows with low and bounded end-to-end latency, very low jitter and low packet loss and has several nice features. One is the temporal isolation property for the ST flows, which is enforced by the TAS. Second, the enhanced flexibility compared to a fixed superframe structure or timeslots, as the ST Windows have different lengths for different ST flows (based on the flow payload) while when building a superframe the same timeslot size is chosen for all the flows, regardless of the different frame payloads. Moreover, in SchedWiFi the ST flow transmissions follow the application-assigned schedule and therefore non-ST flows are transmitted whenever they do not interfere with the ST traffic transmissions. Conversely, in TDMA-based mechanisms transmissions are bound to their assigned slots and typically there is an over-provisioning of the number of slots (e.g., to enable the superframe to also accommodate aperiodic traffic and periodic flows with periods longer than the superframe length). This way, in TDMA-based mechanisms the network cycle time may be longer than needed. The third relevant property is that, unlike most of the existing approaches (e.g., IsoMAC, RT-WiFi), SchedWiFi works in ad-hoc mode without any access point or network coordinator. This results in higher flexibility, as access points typically are to be connected to a wired network, and higher reliability, as the single-point of failure of the centralized approaches is avoided.

3. Background of the EDCA mechanism

The SchedWiFi approach presented in this paper adopts the Enhanced Distributed Channel Access (EDCA) mechanism defined in the IEEE 802.11e amendment [25], which is part of the IEEE 802.11-2012 standard [2] and offers prioritized QoS support.

The EDCA mechanism provides the following priority classes, called Access Categories (ACs):

- AC_VO: Voice traffic (highest priority).
- AC_VI: Video traffic.
- AC_BE: Best Effort traffic.
- AC_BK: Background traffic (lowest priority).

Each AC has its own transmission queue and the highest priority queue (i.e., the AC_VO queue) is served first. When the AC_VO queue is empty, the AC_VI queue is served, and so on. A node that has data to transmit listens to the channel. If the channel is busy, the transmission will be deferred, otherwise the node listens to the channel for a time interval, called an Arbitration InterFrame Space (AIFS), as shown in Fig. 1.

Each AC has a different AIFS, that is calculated as

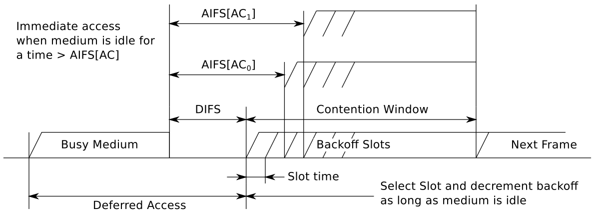

Figure 1. EDCA medium access timings [2].

in Formula (1)

$$AIFS[AC] = AIFSN[AC] \times SlotTime + SIFS \quad (1)$$

where AIFSN[AC] > 2 is defined according to the AC, while the SlotTime and SIFS are parameters defined in the standard [2] which depend on the Physical Layer adopted.

To avoid collisions of messages belonging to the same AC, a random backoff time is added to AIFSN. The backoff time is within a Contention Window (CW), that is bounded for each AC by the parameters CW_{min} and CW_{max}. In case of retransmissions, the CW is iteratively incremented until it reaches the CW_{max} value for a given AC. To allow the transmission of the highest priority messages before the lower priority ones, the CWs are set so as to avoid overlap between CWs (e.g., $CW_{max}[AC\_VO] \leq CW_{min}[AC\_VI]$). Finally, TXOP is the maximum time duration for a node to transmit after winning access to the channel.

The EDCA mechanism does not provide any guarantee that messages priorities are respected. When the workload increases, the performance of the protocol deteriorates due to the narrow range of backoff values [3]. Moreover, traffic prioritization is not sufficient to handle specific kinds of traffic, such as Scheduled Traffic (e.g., control traffic), which require real-time capabilities and collision-free transmissions. Unlike EDCA, the SchedWiFi approach is specifically designed to offer support to Scheduled Traffic, as it will be discussed in the following Section.

4. The SchedWiFi approach

4.1. Scheduled Traffic

The Scheduled Traffic in our model is a high priority periodic traffic that is transmitted according to a time schedule so as to ensure no interference from other traffic types. The characteristics of ST flows (period P, frame size L) are fixed and a priori known. The transmission sequence of ST flows is handled off-line by a scheduler which adopts offset-scheduling

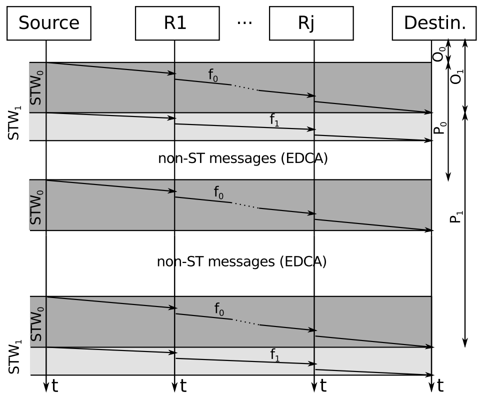

Figure 2. Example of ST-Windows.

techniques [26] to avoid collisions between different ST flows. The end-to-end transmission of an ST flow along its path from source to the final destination, that goes through one or more relay nodes when direct transmission is not possible, occurs within a temporal window, called an ST-Window (STW). During the STW only the messages of the ST flow assigned to the ST-Window can be transmitted from the source node to the destination node along the defined route. Moreover, the nodes that are not involved in the transmission of the ST-messages are not allowed to transmit during the STW, and so collisions are avoided. In Fig. 2 a scheduling example with two flows, f_0 and f_1, is shown. There are two STWs, one for each flow, i.e., the STW_0 (dark gray) and the STW_1 (light gray). The two considered flows have two different periods, P_0 and P_1. In our schedule two different offsets, O_0 and O_1, to avoid any possible overlap between the two STWs, are used.

All the network nodes are aware of the start time and the end time of each ST-Window, through an off-line configuration. This way, no collision may occur even in case of hidden nodes. In order to establish a common notion of time in the proposed approach, a clock synchronization mechanism is required. Several synchronization algorithms are proposed in literature, for example [27]–[30]. Here nodes are assumed to be synchronized according to the algorithm in [30], which provides a synchronization accuracy of 25 μs in a network with up to 100 nodes.

Each ST-Window has a fixed length and is sized to accommodate the end-to-end message transmissions from the source to the destination. The ST-Window is sized taking into account the message length, the number of hops, the maximum number of retransmissions allowed for each hop, and the synchronization accuracy. The ST-Window sizing will be discussed in Sect. 4.2.

As far as the routing policy is concerned, in this approach nodes maintain a static routing table in which the ST flows have fixed routes defined during the deployment phase. The main advantage of such a choice is that in this way the routing delays for the ST traffic are constant and predictable, and the ST-Windows can be easily sized.

To improve fault-tolerance, in SchedWiFi each node has to maintain a routing table with a backup path (with the same number of hops) for each flow. When a sender node does not receive the acknowledgement from the receiver one, the sender transmits the message on the backup path.

4.2. Sizing the ST-Window

In the SchedWiFi approach, the use of ST-Window isolates scheduled traffic from the interference of other traffic categories. During this time interval, nodes can only transmit scheduled traffic, hence the ST-Window for an ST flow has to be sized so as to accommodate the transmission of a message of the flow from the source to the final destination, including acknowledgments and retransmissions. The ST-Window size therefore depends on:

- The maximum clock difference between nodes (Δ), that is equal to $\Delta = max\_sync\_error + max\_skew$, where max_skew is the clock difference just before a synchronization round.
- The number of hops (H) from the source to the destination.
- The message size (L).
- The maximum number of retransmissions for each hop (R). Note that, in the SchedWiFi approach a message is acknowledged on each hop, hence the time for the ACK transmissions has to be considered too.

Hence, the ST-Window size is calculated as in Formula (2)

$$STW = 2\Delta + (TX_{data} + 2SIFS + TX_{ack})\,(1+R)H \tag{2}$$

where TX_{data} is the time required for the transmission of L bytes (in industrial contexts L, i.e., the size of ST messages, is small) and TX_{ack} is the time required for the transmission of an ACK message.

4.3. Non-ST traffic

The SchedWiFi approach foresees that the non-ST traffic is transmitted out of the ST-Windows and, according to the EDCA rules, after the contention phase, as explained in Sect. 3. As a result, delays for non-ST traffic are not predictable. However, in order to provide a different QoS to different flows, messages are prioritized according to their access category, as defined by EDCA protocol specified in the standard [2]. The SchedWiFi approach provides three non-ST traffic Categories:

- **Periodic High Priority (PHP):** The periodic traffic flows with the highest priority among the non-ST flows.
- **Periodic Low Priority (PLP):** The periodic traffic flows with the second highest priority among the non-ST flows.
- **Best Effort (BE):** The aperiodic traffic, with the lowest priority.

The mapping between the SchedWiFi Categories (SC) and the Access Categories (AC) defined by the EDCA protocol in the IEEE 802.11 standard is shown in Table 1.

The shorter the periods of the ST flows, the smaller the space between consecutive ST windows, and this could limit the probability to accommodate the transmission of non-ST flows between them. In order to avoid the resulting potential for starvation of the non-ST flows, the non-ST messages can be fragmented according to the fragmentation technique described in [2], thus enabling them to fit the space between two consecutive ST Windows.

4.4. Time-Aware Shaper (TAS)

In order to isolate the ST traffic from the non-ST traffic, SchedWiFi provides each node with a Time-Aware Shaper (TAS) that prevents the transmission of the traffic that could interfere with the transmission of ST messages. As shown in Fig. 3, the TAS function is implemented just before the frame transmission in the physical layer. The TAS checks whether the transmission duration and acknowledgement reception of a non-ST frame exceeds the start time of a ST-Window. If so, the TAS blocks the transmission and restores the MAC state to the one before the start of the CSMA/CA operations. The TAS also blocks the transmission of ST messages that would exceed their own STW and interfere with other ST transmissions. Note that it is recommendable that the ST windows are consecutive, in order to minimize the bandwidth waste due to the TAS mechanism.

Table 1. Mapping between SCs and ACs

SchedWiFi	ST	PHP	PLP	BE
802.11	-	AC _VO	AC_VI	AC_BE

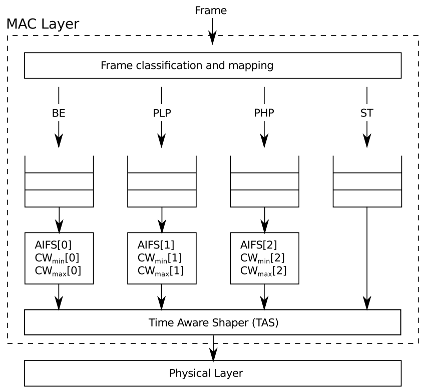

Figure 3. Frame flow diagram of SchedWiFi.

In this way, the ST messages can be transmitted without any interference from both other ST messages and non-ST messages.

5. Performance Evaluation

This section describes the scenario and the traffic model used in the OMNet++ simulations.

5.1. Simulation Scenario

In Fig. 4 the considered set of nodes is shown. The sensing area is 100x100 m^2 large and the SchedWiFI nodes within the area are pseudo-randomly located.

There are three kinds of SchedWiFI nodes in the network:

- The source nodes ($S_{1...6}$, in grey in the picture). They generate and send messages.
- The receiver nodes ($R_{1...3}$, in black in Fig. 4). They are the sinks that receive messages.
- The relay nodes (in white in Fig. 4). They do not generate messages at the application layer, but forward the messages received from other nodes

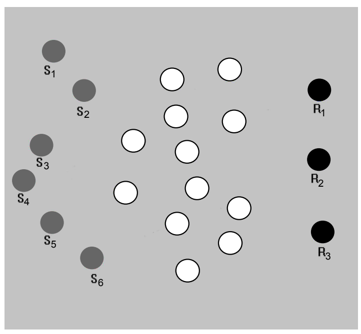

Figure 4. Network topology

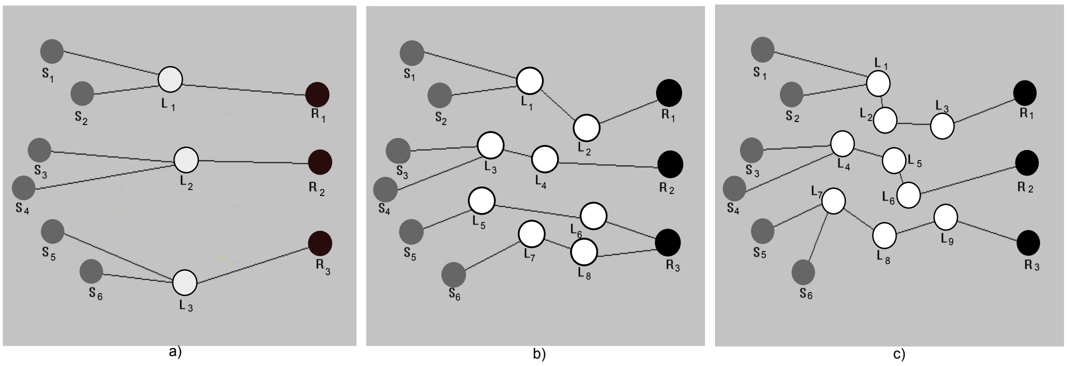

Figure 5. a) 2-hop configuration b) 3-hop configuration c) 4-hop configuration

(either source or relay nodes) according to their routing tables.

In the simulation, the start time of each flow is obtained through an exponential distribution with mean equal to 1 s.

The SchedWiFi approach, due to the ST-Windows, requires high bandwidth, as bandwidth is reserved in each node in the entire network for each transmission of ST messages and for the possible retransmissions. For this reason, in this scenario the IEEE 802.11n [31] physical layer was chosen, which provides a theoretical datarate up to 600 Mb/s. SchedWiFi is also suitable for working on higher datarate physical layers, e.g., the IEEE 802.11ac-2013, which provides a theoretical datarate up to 6.93 Gb/s, while lower datarates PHY types (e.g., the IEEE 802.11b/g) are not recommended for efficiency reasons.

The adopted propagation model is the Log Normal Shadowing [32], with alpha equal to 2.4 and standard deviation sigma equal to 6.7. These values were chosen so as to reflect the signal propagation conditions in realistic industrial environments. A static routing algorithm was chosen, as the network is configured so that the coverage does not allow the source nodes to directly reach the relevant receiver nodes and all nodes in the network do not move, but keep a fixed position assigned during the configuration phase.

Using the same scenario, three different configurations were created and named 2-hop, 3-hop and 4-hop depending on the number of hops that the messages must traverse to reach the receiver nodes. The given scenario includes 12 available relay nodes. The propagation delay due to the distance between the nodes is negligible and does not affect the transmission delay. For each configuration, the relay nodes that are actually used were randomly chosen among all those available in the network. For each configuration, five runs were performed by varying the random parameters, in order to prove that the accuracy of the results is not affected by the use of a probabilistic distribution.

In Fig. 5 all the topologies for the 2-hop, 3-hop and 4-hop configuration are shown. Fig. 5.a describes the topology for the 2-hop configuration. Three relay nodes were chosen as follows. L_1 forwards the messages from S_1 and S_2, L_2 forwards the messages from S_3 and S_4, while L_3 forwards the messages from S_5 and S_6.

Fig. 5.b shows the 3-hop configuration. Eight relay nodes were chosen as follows.L_1 and L_2 forward the messages sent by S_1 and S_2. L_3 and L_4 forward the messages sent by S_3 and S_4. L_5 and L_6 forward the messages sent by S_5. Finally, L_7 and L_8 forward the messages sent by S_6.

Fig. 5.c shows the 4-hop configuration.Nine relay nodes were chosen as follows.L_1, L_2 and L_3 forward the messages sent by S_1 and S_2. L_4, L_5 and L_6 forward the messages sent by S_3 and S_4. Finally, L_7, L_8 and L_9 forward the messages sent by S_5 and S_6.

5.2. Traffic Model and Evaluation Metrics

Tab. 2 shows the traffic characterization and each row describes a traffic flow. The first column shows the source nodes, the second column the traffic class, the third and fourth column the payload and the period, respectively, and the last column the receiver node for the relevant flow. The source nodes S_1 and S_2 every 5 ms generate 46 bytes long ST messages and send them to R_1. In parallel, S_1 and S_2, every 3 ms, send to R_1 periodic high priority (PHP) messages and, every 3.5 ms, periodic low priority (PLP) messages. Both the high and the low priority messages are 1200 bytes long. With the same timings, S_3 and S_4 send to R_2 PHP and PLP messages. Finally, S_5 and S_6 send 1200 bytes long aperiodic best effort messages to R_3.

Table 2. Traffic characterization

Source node	Priority	Payload (byte)	Period (ms)	Destination
S1 - S2	ST	46	5	R1
S1 - S2	Periodic-HP	1200	3	R1
S1 - S2	Periodic-LP	1200	3.5	R1
S3 - S4	Periodic-HP	1200	3	R2
S3 - S4	Periodic-LP	1200	3.5	R2
S5 - S6	Best Effort	1200	-	R3

Table 3. Workload

Traffic Class	Workload
ST	147.2 Kbps
PHP	6.40 Mbps
PLP	5.48 Mbps

Table 3 shows the workload value for each traffic class.

The considered performance metrics are:

- The end-to-end delay, defined as the time interval between the message creation time and the message delivery time at the destination node, measured at the application layer.
- The jitter, defined as the difference between two adjacent end-to-end delay values.
- The Packet Loss Ratio (PLR), defined as the ratio between the number of packets lost (due to collisions or bit errors) and the number of packets sent.

Tables 4 and 5 show the MAC and PHY parameters used in the simulations. In Table 5, the AIFS parameter represents the time a node has to wait, listening an idle channel, before making a transmission attempt. CWmin and CWmax are the minimum and maximum length, expressed in slots, of the Contention Windows (CW), i.e., the time window within which the channel contention takes place, for a given access category. If a retransmission occurs, the CW for the relevant access category is increased up to the CWmax value.

6. Results

In this section the results obtained for each configuration are presented and discussed. The results are shown with the relevant confidence interval.

Table 4. PHY parameters

Parameter	Layer	Value
Datarate	PHY	300Mbps
Transmitter Power	PHY	2mW
Thermal Noise	PHY	-110dbm
Carrier Frequency	PHY	2.4 GHz
Modulation	PHY	QAM

Table 5. MAC parameters

Access Category	CWmin	CWmax	AIFSN
PHP	7	15	2
PLP	15	31	2
BE	31	1023	3

6.1. 2-hop configuration

ST Flows delay. In the 2-hop configuration, the maximum end-to-end delay obtained by SchedWiFi for the ST traffic is equal to 48 μs, that is twice the transmission time of a 46 bytes long ST message (i.e., 24 μs). The end-to-end delay of ST messages is bounded by design and in this simulation was found fixed, as there was no need for retransmissions. This result confirms that the ST Windows mechanism works well for the 2-hop configuration, so predictable delays for the ST traffic can be guaranteed.

Non-ST Flows delay. Fig. 6 shows the distribution of the end-to-end delay for the PHP traffic class. Most the end-to-end delay results are in the range [0.2 ms, 0.4 ms]. The results show that most of the PHP messages arrive at destination with an end-to-end delay lower than 1 ms. However, there is also a non-negligible number of messages that experience end-to-end delay values higher than 1 ms. This higher delay is mainly due both to the delay that PHP messages experience in the queues and to the action performed by the TASs. Fig. 7 shows the distribution of the end-to-end delay for the PLP traffic class. All the messages are received with an end-to-end delay lower than 6 ms and the average end-to-end delay value is in the range [0.2 ms, 1 ms]. The PHP traffic class presents mean end-to-end delay values lower than those obtained for the PLP traffic class thanks to the traffic prioritization mechanism implemented in the queues, that allows to serve PHP messages with the highest priority among the non-ST flows.

Packet loss ratio and jitter. Table 6 shows the packet loss ratio for each traffic class and each configuration, while Table 7 shows the mean jitter values obtained for each traffic class and for each configuration, with the relevant confidence interval.

The number of collisions experienced by non-ST flows is very low, especially for the PHP and PLP traffic classes. The packet loss ratio for the ST class is low and is not related to the number of collisions, as it depends entirely on the channel bit error rate. The throughput of the ST traffic, equal to 147 Kbps, is very close to the ST workload and this confirms

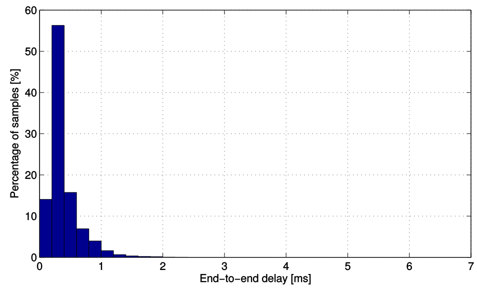

Figure 6. End-to-end delay for PHP traffic in the 2-hop configuration.

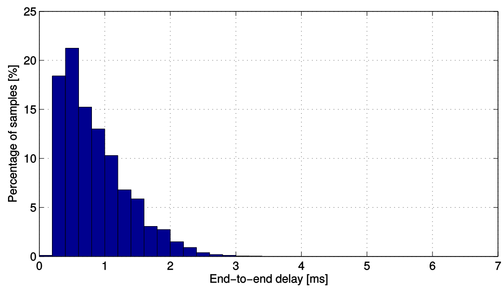

Figure 7. End-to-end delay for PLP traffic in the 2-hop configuration.

that the introduced channel noise does not significantly affect the transmissions of ST flows. The jitter value is negligible for the ST class, while for both the PHP and PLP traffic classes the jitter value is higher than 100 μs. This value is still acceptable, as most process automation applications can tolerate jitter values up to 1 ms [33].

6.2. 3-hop configuration

ST Flows delay. The end-to-end delay value for the ST class is equal to 72.51 μs, i.e., three times the

Table 6. Packet loss ratio for all the traffic classes

Traffic Class	2-hop	3-hop	4-hop
ST	0.19%	0.22%	0.31%
Periodic-HP	0.36%	0.98%	21.64%
Periodic-LP	0.57%	1.49%	25.77%
Best-Effort	1.34%	21.83%	97%

Table 7. Jitter values for all the traffic classes

Traffic Class	2-hop (μs)	3-hop (μs)	4-hop (μs)
ST	0.33 $\pm$ 0.003	0.47 $\pm$ 0.005	0.5 $\pm$ 0.006
Periodic-HP	111.74 $\pm$ 1.33	183.73 $\pm$ 1.79	1539 $\pm$ 44
Periodic-LP	159.85 $\pm$ 2.24	342.36 $\pm$ 2.95	1844 $\pm$ 67

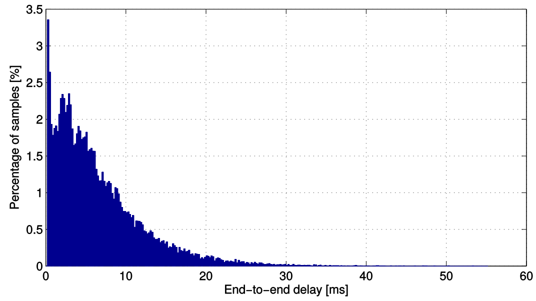

Figure 8. End-to-end delay for PHP traffic in the 3-hop configuration.

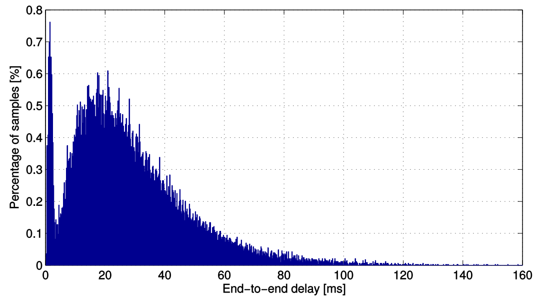

Figure 9. End-to-end delay for PLP traffic in the 3-hop configuration.

transmission delay of an ST message, which is 24 μs in the scenario here considered. This confirms that the SchedWifi mechanism works properly even in the 3-hop configuration, guaranteeing an upper bounded end-to-end delay for the ST class.

Non-ST Flows delay. In Figure 8 the distribution of the end-to-end delay values for the PHP traffic class in the 3-hop configuration is shown.

The end-to-end delay values for the PHP traffic that most frequently occur are in the range [1.5 ms, 10 ms]. The difference between these results and those obtained with the 2-hop configuration is mainly due to the increase of the queuing delay for PHP messages, which undergo an additional hop to reach the destination node.

Fig. 9 shows the end-to-end delay distribution for the PLP traffic class. More than the 80% of the PLP class messages obtain a value below 50 ms. Comparing these results with those of the 2-hop configuration, in the 3-hop configuration the delay values are in general higher. Moreover, the delay increases more significantly for the PLP traffic class than for the PHP traffic class. This result depends on the queuing time in the relay nodes, where the PLP messages have to wait for the transmission of the PHP messages before being transmitted.

Packet loss ratio and jitter. Table 7 shows the mean jitter results and the confidence intervals calculated at the 95%. Similarly to the 2-hop results, the jitter values obtained for the ST class are negligible, while those of both the PHP and PLP messages are lower than 1 ms, i.e., they are higher in this case than in the 2-hop configuration. As shown in Table 6, the packet loss ratio for the ST class is low and, as discussed in the 2-hop configuration, is not related to the number of collisions, as it depends on the channel bit error rate. The packet loss ratio for both PLP and PHP traffic classes is lower than 1.50%, while for the best-effort traffic is equal to 21.83%. This result, which exceeds the 3% threshold required in industrial automation applications [34], is due to both the queue overflow in the relay nodes and the high number of collisions experienced by the best-effort messages.

6.3. 4-hop configuration

ST Flows. In this configuration the end-to-end delay for the ST class is bounded and equal to 96 ms, i.e., four times the transmission time of a single ST message, and the jitter value is negligible, as shown in Table 7. The fourth column of Table 6 shows the packet loss ratio for the 4-hop configuration. The result for the ST class is comparable to those obtained for the 2-hop and 3-hop configurations. This proves that the ST messages, thanks to the ST Window mechanism, do not experience collisions when the hop number increases.

Non-ST Flows. Fig. 11 and 10 show the end-to-end delay distribution for the PHP and PLP traffic class, respectively. Comparing with the results obtained with the previous configurations, here the end-to-end delay values for the PHP traffic class are higher than in the previous configurations, up to 160 ms, while the 80% of the messages experienced an end-to-end delay lower than 50 ms. The end-to-end delay of the PLP traffic class is significantly increased and reaches quite high values, up to 650 ms, while the 80% of end-to-end delay results for the PLP class are lower than 210 ms. Jitter for non-ST flows also increases in this configuration.

Table 6 shows that the packet loss ratio for the PHP and PLP traffic class are higher, i.e., equal to 21.64% and 25.77%, respectively. This result is mainly due to the increased number of hops, which increases the probability of collisions for non-ST flows. The packet loss ratio obtained for the best effort traffic class is very high, i.e., close to 97%. This is because the best-effort traffic has the lowest priority, so best-effort flows in such an unfavourable configuration perform worse than in the previous configurations.

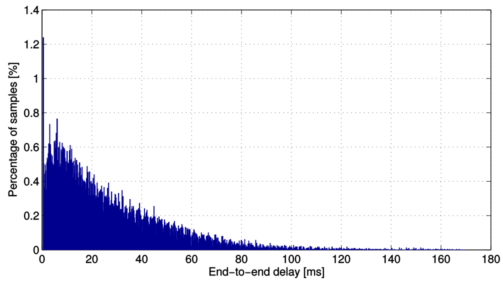

Figure 10. End-to-end delay for PHP traffic in the 4-hop configuration.

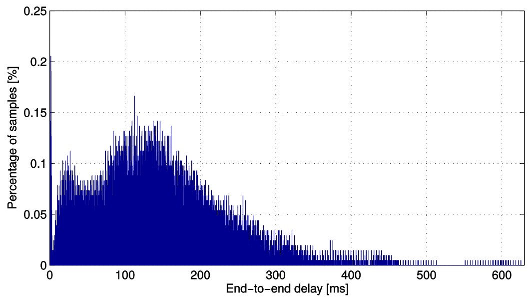

Figure 11. End-to-end delay for PLP traffic in the 4-hop configuration.

Table 8 summarizes the end-to-end results of the ST messages for each configuration.

Table 8. End-to-end Delay for ST Messages

2-hop	3-hop	4-hop
$48\mu s$	$72\mu s$	$96\mu s$

In Table 9 the average end-to-end delay and the confidence interval calculated at the 95% of confidence level for the simulations are presented. The results show that the confidence intervals are negligible if compared to the order of magnitude of values.

Table 9. Average end-to-end delay with confidence interval

Traffic Class	2-hop (ms)	3-hop (ms)	4-hop (ms)
PHP	0.365 ± 0.003	6.600 ± 0.057	26.804 ± 0.303
PLP	0.855 ± 0.005	28.394 ± 0.215	142.492 ± 1.182

Since the entire SchedWiFi approach is distributed and no polling mechanisms are foreseen, scalability depends on the number of ST flows in the network and the lengths of the ST Windows. In fact, the longer the ST Windows, the shorter the time available for non-ST transmissions. The results shown in this Section prove that with up to 17 nodes, 2 ST-flows, 10 non-ST flows and adopting paths long up to three hops, the packet loss ratio is always lower that 22%, even for the traffic with the lowest priority.

Nevertheless, the higher is the number of hops the larger are the ST Windows, which means less time for non-ST transmissions. In this case, the results proved that the traffic flows with lowest priority, i.e. the best effort traffic, could experience higher packet loss ratio and, eventually, starvation, as shown in Table 6. Anyway, it is likely that a large area will be covered by multiple clusters, each one with a small number of hops.

7. Conclusions and future works

This paper presented the SchedWiFi approach that provides support for scheduled traffic over IEEE 802.11 ad-hoc networks. In the paper, the SchedWiFi approach was presented and evaluated through simulations in three different configurations, in order to assess both the support provided to ST class and the scalability of the approach with an increasing number of relay nodes between the source and the destination.

Results proved that, as far as the ST class is concerned, in all the considered configurations SchedWiFi allows for low and predictable end-to-end delay values, negligible jitter and very low packet loss ratio. Compared to the MCCA mechanism defined in the IEEE 802.11 standard [2] and assessed in [6], in the SchedWiFi approach non real-time traffic does not affect the end-to-end delay of ST flows. On the other hand, the end-to-end delay values of all the non-ST classes is sensitive to the increase of the number of hops.

Future work will deal with mechanisms of bandwidth reservation for non-ST periodic traffic classes, and in particular for the PHP and PLP flows. In addition, the adoption of a dynamic routing protocol (such as controlled flooding) will be investigated. Comparative evaluations between SchedWiFi and other relevant approaches proposed in literature, such as [16] [17] [18] will be performed. Moreover, using the quality criteria and trade-offs described in [35], the security provided in SchedWiFi will be evaluated and discussed. Finally, as SchedWiFi requires only software modifications to the IEEE 802.11 MAC layer, the feasibility of implementing SchedWiFi on real devices will be investigated.

References

[1] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. of the IEEE*, vol. 93, no. 6, pp. 1130–1151, June 2005.

[2] *IEEE std. 802.11-2012 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, March 2012.

[3] S. Vittorio, G. A. Kaczynski, and L. Lo Bello, "Improving the real-time capabilities of IEEE 802.11e through a Contention Window Adapter," in *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2007, pp. 64–67.

[4] G. Cena, L. Seno, A. Valenzano, and C. Zunino, "On the Performance of IEEE 802.11e Wireless Infrastructures for Soft-Real-Time Industrial Applications," *IEEE Trans. on Industrial Informatics*, vol. 6, no. 3, 2010.

[5] A. Krasilov, A. Lyakhov, and A. Safonov, "Interference, Even with MCCA Channel Access Method in IEEE 802.11s Mesh Networks," in *Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Valencia, Spain, 17-22 Oct. 2011, pp. 752–757.

[6] C. M. D. Viegas, F. Vasques, P. Portugal, and R. Moraes, "Real-time communication in IEEE 802.11s mesh networks: simulation assessment considering the interference of non-real-time traffic sources," *EURASIP Journal on Wireless Comm. and Networking*, vol. 219, pp. 1–15, 2014.

[7] H. Trsek, L. Wisniewski, E. Toscano, and L. Lo Bello, "A flexible approach for real-time wireless communications in adaptable industrial automation systems," in *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Toulouse, France, 5-9 Sept. 2011.

[8] H. Trsek and J. Jasperneite, "An isochronous medium access for real-time wireless communications in industrial automation systems - A use case for wireless clock synchronization," in *Proc. of International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*, Munich, Sept. 2011, pp. 81–86.

[9] R. Costa, P. Portugal, F. Vasques, and R. Moraes, "A TDMA-based Mechanism for Real-Time Communication in IEEE 802.11e Networks," in *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Bilbao, 13-16 Sept. 2010.

[10] Y. H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications," in *Proc. of IEEE Real-Time Systems Symposium (RTSS)*, Vancouver, 3-6 Dec. 2013, pp. 140–149.

[11] Viegas, R. et al., "A new MAC scheme specifically suited for real-time industrial communication based on IEEE 802.11e," *Computers and Electrical Engineering*, vol. 39, 2012.

[12] N. Pereira, B. Andersson, and E. Tovar, "WiDom: a dominance protocol for wireless medium access," *Industrial Informatics, IEEE Transactions on*, vol. 3, pp. 120–130, 2007.

[13] E. Toscano and L. Lo Bello, "A Middleware for Reliable Soft Real Time Communication over IEEE 802.11 WLANs," in *Proc. of IEEE International Symposium on Industrial Embedded Systems (SIES)*, Vasteras, 15-17 June 2011, pp. 115–122.

[14] L. Seno, S. Vitturi, and F. Tramarin, "Tuning of IEEE 802.11 MAC for improving real-time in industrial wireless networks," in *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Krakow, Poland, 17-21 Sept. 2012.

[15] X. Tian, T. Ideguchi, T. Okuda, and N. Okazaki, "Improving throughput of WLANs by scheduling random access," in *Proc. of International Symposium on Wireless and Pervasive Computing (ISWPC)*, Hong Kong, 23-25 Feb. 2011.

[16] A. Arkoub, U. A. Khan, and J. Seitz, "Stream reservation MAC protocol for wireless ad-hoc networks," in *Proc. of IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Amman, Jordan, 3-5 Dec. 2013.

[17] Tardioli, D. et al., "A wireless multi-hop protocol for real-time applications," *Computer Communications*, vol. 55, 2015.

[18] F. Santos, L. Almeida, and L. S. Lopes, "Self-configuration of an adaptive TDMA wireless communication protocol for teams of mobile robots," in *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Hamburg, Germany, 15-18 Sept. 2008, pp. 1197–1204.

[19] P. Gutierrez Peon, H. Kopetz, and W. Steiner, "Towards a reliable and high-speed wireless complement to ttethernet," in *IEEE Emerging Technology and Factory Automation (ETFA)*, Sept 2014, pp. 1–4.

[20] W. Steiner, G. Bauer, B. Hall, and M. Paulitsch, "Ttethernet: Time-triggered ethernet," in *Itime Triggered Communication*, R. Obermaisser, Ed. CRC Press, 2011.

[21] E. Toscano and L. Lo Bello, "A topology management protocol with bounded delay for Wireless Sensor Networks," in *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Hamburg, Germany, 15-18 Sept. 2008, pp. 942–951.

[22] M. Collotta, L. Lo Bello, E. Toscano, and O. Mirabella, "Dynamic load balancing techniques for flexible wireless industrial networks," in *Proc. of the Annual Conference on IEEE Industrial Electronics Society (IECON)*, Glendale, AZ, USA, 7-10 Nov. 2010, pp. 1329–1334.

[23] M. Natkaniec, K. Kosek-Szott, S. Szott, and G. Bianchi, "A survey of medium access mechanisms for providing qos in ad-hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, 2013.

[24] L. Lo Bello, G. A. Kaczyński, F. Sgró, and O. Mirabella, "A wireless traffic smoother for soft real-time communications over IEEE 802.11 industrial networks," in *Proc. of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Prague, Czech Republic, 20-22 Sept. 2006, pp. 1073–1079.

[25] *IEEE 802.11e-2005 - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE, Nov. 2005.

[26] K. Tindell, "Adding Time-Offsets to Schedulability Analysis," Dept. of Computer Science, York University, Tech. Rep. YCS221, 1994.

[27] D. Zhou, L. Huang, and T. H. Lai, "On the scalability of IEEE 802.11 ad-hoc-mode timing synchronization function," *Wireless Networks*, vol. 14, no. 4, pp. 479–499, Aug. 2008.

[28] A. Mahmood, R. Exel, and T. Sauter, "Delay and Jitter Characterization for Software-Based Clock Synchronization Over WLAN Using PTP," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, 2014.

[29] J. P. Sheu, C. M. Chao, and C. W. Sun, "A clock synchronization algorithm for multi-hop wireless ad hoc networks," in *Proc. of the 24th International Conference on Distributed Computing Systems*, 2004.

[30] D. Zhou and T. H. Lai, "An Accurate and Scalable Clock Synchronization Protocol for IEEE 802.11-Based Multihop Ad Hoc Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1797–1808, Dec. 2007.

[31] *IEEE 802.11n-2009 - Amend. 5: Enhancements for Higher Throughput*.

[32] Studei P. and Alonso G., "Log-normal shadowing meets SINR: A numerical study of Capacity in Wireless Networks," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, San Diego, CA, 2007.

[33] G. Scheible, D. Dzung, J. Endresen, and J.-E. Frey, "Innovation and intellectual property rights," in *Industrial Communication Technology Handbook, Second Edition*, R. Zurawski, Ed. CRC Press, 2014.

[34] H. Zha and X. Chen and Y. Fang, , "A Call Admission and Rate Control Scheme for Multimedia Support over IEEE 802.11 Wireless LANs," in *Proceedings of the 1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE)*, Dallas, TX, 2004.

[35] E. Lisova, E. Uhlemann, J. Åkerberg, and M. Björkman, "Towards secure wireless ttethernet for industrial process automation applications," in *IEEE Emerging Technology and Factory Automation (ETFA)*, Sept 2014, pp. 1–4.