This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

A Dynamic Rate Selection Algorithm for IEEE 802.11 Industrial Wireless LAN

Federico Tramarin, *Member, IEEE,* Stefano Vitturi, *Senior Member, IEEE,* and Michele Luvisotto, *Student Member, IEEE,*

***Abstract*—The Multi–Rate Support feature has been introduced by the IEEE 802.11 standard to improve system performance, and has been widely exploited by means of Rate Adaptation (RA) strategies within general purpose Wireless LANs. These strategies revealed ineffective for real–time industrial communications, and alternative solutions, better tailored for such a specific field of application, were investigated. The preliminary outcomes of the analyses carried out were promising, even if they clearly indicated that further efforts were necessary. In this direction, this paper firstly proposes Rate Selection for Industrial Networks (RSIN), an innovative RA algorithm specifically conceived for the real–time industrial scenario with the goal of minimizing the transmission error probability, while taking into account the deadline imposed to packet delivery. Then, it describes the practical implementation of RSIN on commercial devices, along with that of other formerly introduced RA techniques. Finally, the paper presents a thorough performance analysis, carried out to investigate the behavior of the addressed RA schemes. Such an assessment was performed via both experimental campaigns and simulations. The obtained results, on the one hand, confirm the effectiveness of the RA techniques purposely designed for real–time industrial communication. On the other hand, they clearly indicate that RSIN outperforms all the other strategies.**

I. Introduction

Multirate Support (MRS) is a feature offered by the IEEE 802.11 Wireless LAN (WLAN) standard [1] to improve system performance. Basically, MRS allows a station to dynamically select the transmission rate for a forthcoming packet with the aim of increasing the chance of successful delivery, by better adapting to channel conditions. Indeed, MRS relies on the fact that lower rates adopt more robust modulations, and are hence able to ensure higher transmission success probabilities even under low signal–to–noise ratio (SNR) conditions.

The IEEE 802.11 standard does not define any rate selection algorithm, leaving the practical implementation to manufacturers of compliant devices. Therefore, this has led to the design of different Rate Adaptation (RA) strategies, currently implemented on most off–the–shelf available devices, mainly conceived for general purpose WLAN applications. These techniques revealed not effective in the industrial communication scenario, as assessed in [2], [3], since the design choices were mainly targeted at network throughput maximization, while performance indexes of prominent importance for real–time industrial communications, such as timeliness and reliability, were not addressed.

Manuscript received May 20, 2016. Accepted October 3, 2016.

The authors are with the National Research Council of Italy, CNR–IEIIT, and also with the Department of Information Engineering, University of Padova, Via Gradenigo 6/B, I–35131, Padova, Italy.
E-mails: {vitturi, tramarin, luvisott}@dei.unipd.it.

Therefore, two new RA schemes were proposed in [2], that proved to be promising for some specific traffic profiles of IEEE 802.11–based industrial communication systems. More importantly, the assessment carried out in [2] highlighted that significant benefits can be expected from RA techniques specifically designed for the industrial scenario and, consequently, suggested further studies in this direction.

The present paper collocates in this area of research. Its main contribution is the introduction of an innovative RA technique specifically conceived for real–time industrial communication systems, based on a completely different approach with respect to other known algorithms. As a further noteworthy aspect, this paper describes how the proposed RA algorithm has been implemented on commercially available devices, without requiring modifications neither to the IEEE 802.11 standard nor to the firmware of network cards. Moreover, this paper presents the results of a measurement campaign executed on a prototype network where the performance of the proposed RA scheme have been compared with those of state–of–the–art industrial and general purpose RA algorithms.

II. State of the Art and Contribution

Industrial applications, such as factory automation, control and monitoring, typically impose tight requirements to the underlying communication networks. For example, control systems may require sampling rates in the order of some kHz, characterized by a high accuracy, thus compelling the delivery of data messages within strict deadlines, that may result as low as some hundreds of microseconds [4], [5]. These requirements may be difficult to satisfy, especially when wireless networks are used. Indeed, such kind of communication systems are much more error prone than their wired counterparts and, moreover, they may introduce random delays in data transmission due to the backoff procedures that take place between two consecutive transmission attempts. Thus, particular care is necessary in the design of protocols for real–time industrial wireless networks.

A significant example in this direction is RT–WiFi, a real-time protocol, proposed in [6], implemented on top of the IEEE 802.11 physical layer. RT–WiFi is based on a Time Division Multiple Access (TDMA) technique, which ensures an ordered access to the transmission medium exploiting the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

standard IEEE 802.11 Timing Synchronization Function (TSF) to achieve nodes' synchronization. It is worth mentioning that similar techniques are used even by industrial wireless sensor/actuator networks [7], where sampling rates may be more relaxed, whereas the number of nodes may increase significantly [8].

In the IEEE 802.11 WLAN context, MRS constitutes an effective opportunity to improve network performance. In general communication systems, this has been largely interpreted in the maximization of the application level throughput. Indeed, the first RA algorithm published in literature and implemented on devices was Automatic Rate Fallback (ARF) [9], [10], which actually was designed to that purpose. Moreover, it is worth mentioning that some efforts have been spent to improve the performance of ARF, to take into account both short–term and long–term variations of the wireless channel behavior, thus further optimizing network throughput [11].

These efforts, however, proved to be of little significance for a real–time industrial scenario, where different performance indicators are typically considered [12]. Nonetheless, since the validity of the MRS approach is unquestionable, it can be exploited by designing new RA algorithms that point to the optimization of different performance indicators, specifically tailored for the given applications. To this regard, in the industrial scenario two algorithms derived from (ARF) have been proposed in [2]: Static retransmission rate ARF (SARF) and Fast rate reduction ARF (FARF). Both of them behave similarly to ARF, *i.e.* they select the next transmission speed based only on the outcomes of the very last transmission attempts. In practice, both algorithms increase their transmission speed toward the immediately faster one after a predefined number of consecutive successful transmissions. Conversely, these algorithms behave in a smarter way than ARF by acting on the speed at which to perform packet retransmissions. Specifically, if the initial packet delivery attempt is failed, SARF and FARF select the lowest transmission speed (among those available) to ensure the highest success probability in all the subsequent retransmission attempts. Moreover, they differentiate in the selection of the rate for the following packets, after the current one has been successfully delivered. Indeed, SARF returns back at the initial speed, *i.e.* that at which the first attempt failed, because it considers only the result of the first transmission attempt to update the initial rate. Conversely, FARF remains at the lowest value and starts increasing transmission speed from this value.

Both the proposed techniques allow to considerably reduce the number of transmission attempts for a single packet delivery and, hence, decrease the randomness caused by the IEEE 802.11 backoff mechanism, with a consequent improvement of the real–time performance. Nonetheless, they may exhibit a certain inefficiency in reaching the most adequate transmission rate for a given channel status. Indeed, supposing for example that the SNR varies suddenly, causing the channel to switch from a "good" to a "bad" state (a very common situation in industrial environments), then both SARF and FARF would very likely transmit the packet at the second attempt, executed at the lowest rate, hence introducing a limited randomness. At the same time, however, this may not represent the most appropriate choice, since the selection of a higher rate could have ensured a lower packet delivery time with a comparable success probability. Also, considering the subsequent transmissions of other packets, SARF may likely experience some failures before reaching a stable rate, whereas FARF needs several successful transmissions at low speeds before settling at the same rate. It may be concluded that, the choice made by both techniques of retransmitting at the lowest possible rate, which is dictated by the absence of knowledge about channel status, may unnecessarily penalize the packet delivery time.

A further general purpose RA algorithm, *Minstrel* [13], has been proposed in the past years and has rapidly become widespread. Once again, Minstrel relies on the previous communication history, even if its rationale is quite more sophisticated with respect to ARF/SARF/FARF. Indeed, Minstrel builds and maintains a table containing statistics about the successful/failed transmissions for each available rate, updated over a predefined temporal window. Each time a data packet has to be transmitted, Minstrel associates a "Retry Chain" to it, obtained from the collected statistics, which contains four pairs of elements, each one specifying a transmission rate and the number of attempts that can be carried out at that rate. The order in which rates are chosen is fixed, *i.e.* first the two highest throughput ones, then the rate with the highest success probability, and finally the lowest possible one. Thus, a station using Minstrel as RA algorithm initially transmits the packet using the rate indicated in the first pair of the "Retry Chain". Then, if the packet is not delivered within the specified number of attempts, it moves to the second pair, and so on.

The performance analysis carried out in [3] highlighted that the behavior of default Minstrel configurations is definitely not suitable for industrial communications. Indeed, similarly to ARF, it introduces a considerable randomness on packet delivery that is detrimental for real–time applications. However, in [3], an adequate tuning of the Minstrel parameters has been proposed, which showed better performance than both SARF and FARF for a specific traffic profile, such as that generated by real–time industrial multimedia applications [14].

The above analyses, along with the obtained results (that, it is worth remembering, are derived from theoretical models as well as from simulations), represent a valuable step in the context of RA strategies for IEEE 802.11 real–time industrial applications. However, further investigations can be envisaged toward more exhaustive achievements. Particularly, new RA algorithms need to be designed to ensure a prompt reaction to channel status variations without unnecessarily penalizing the transmission rate. Also, the practical feasibility of the different RA techniques has to be addressed and a comparative performance analysis needs to be carried out.

The main contribution of this paper is concerned with the above issues. In detail, we firstly propose the design of Rate Selection for Industrial Networks (RSIN), an innovative RA technique based on the following main features:

- dynamic identification of the channel status, exploiting device–measured SNR levels;
- rate selection based on a constrained minimization of the packet error rate;
- knowledge of the deadline on packet delivery time.

Subsequently, we provide some details about the practical implementation we carried out of RSIN, SARF and FARF, and analyze the results of an extensive set of measurements conducted on a prototype network. The experiments have been executed in a laboratory in which a controlled source of in–band interference allowed to emulate as far as possible industrial operational contexts. In the last part of the paper, we also provide the results of a simulation analysis carried out on a more complex industrial scenario that, due to practical reasons, could not have been built in practice.

Finally, for the sake of completeness, it is worth observing that some general purpose RA techniques based on the SNR knowledge were already proposed, as for example Receiver Based AutoRate (RBAR) [15]. However, with such strategies, the SNR value was obtained via the exchange of RTS/CTS frames, a procedure that increases the transmission overhead and, as such, negatively impacts on the behavior of real–time applications.

III. The RSIN Technique

RSIN has been conceived to target real-time industrial communication, and hence we will refer to configurations and protocols typical of such a scenario [16]. In this context, it is quite customary that a central controller is in charge of managing a set of sensors/actuators. During operations, the controller exchanges process data with each sensor/actuator, either on a cyclic basis or triggered by specific events, and the generated traffic is characterized by a prevalence of scheduled transmissions that have to be completed within tight deadlines. The most common wireless configuration that reflects such a scenario is an Infrastructure WLAN, where the central controller is connected to the Access Point (AP), and sensors/actuators are located on some wireless stations (STAs). Such kind of networks will be hence addressed in this paper. The proposed technique, however, has been designed to be in no way restricted to a given configuration, and then it could be seamlessly adopted by different networks such as those based on ad–hoc and mesh topologies.

From the design perspective, RSIN is based on two main assumptions. The first one specifies that in any data exchange between two stations, each packet has to contain an additional field in which the transmitting node inserts the perceived SNR relevant to the last received packet from the other node. This is feasible, since the SNR value can be evaluated by the Wireless Network Interface Cards (WNICs) adopted by the stations and then included in the payload of the exchanged frames, as briefly sketched in Fig. 1. Indeed, the SNR evaluation can be carried out on a per–packet basis by extracting the Received Signal Strength Indicator (RSSI) from the incoming frame, and then subtracting from this value the noise floor power. To this regard, it is worth recalling that both the RSSI and the noise floor level have to be measured by the WNIC to correctly perform frame decoding. Hence, the availability of the SNR value to RSIN does not depend on the physical features of the WNICs but, rather, by their device drivers that may or may not provide such an information. As will be described in the following, the devices we used in the

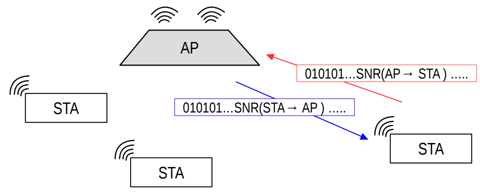

Fig. 1. Inclusion of the SNR value in the payload of exchanged frames.

experimental set–up made the SNR value available, so that the RSIN technique could be implemented. In any case, to ensure the correct operation of RSIN it may be assumed that the SNR information could be retrieved even in a different way, for example, via estimation.

It is also worth observing that the measured SNR value can be typically stored in one byte. As such, in the considered scenario, where usually small amounts of data are exchanged, adding a field with this information to the frame payload has a negligible impact on the overall frame size, as well as on its delivery time.

RSIN further assumes that any transmitting node is aware of the relationship between the Packet Error Rate (PER) and SNR for any possible transmission rate. Such an information can be actually derived from theoretical analyses [17], or through extensive experimental measurements campaigns, as the ones presented in [18].

A. Formal Description

The aforementioned assumptions allow to state that each node is provided with the map:

$$\mathcal{F} : S \times \mathcal{R} \rightarrow \mathcal{P} \tag{1}$$

where S represents the set of possible SNR levels, $\mathcal{R}$ is the set of the available transmission rates[1] and $\mathcal{P}$ is the set of probability values. Indeed, the outcome of the map is the probability P_e with which the next frame transmission fails, given a particular combination of R and S. Clearly, P_e is a real number which belongs to the range $[0, 1]$. It is worth noticing that the PER also depends on the transmitted frame size L, and hence the provided map $\mathcal{F}$ should scale accordingly.

The RSIN technique is defined as an optimization problem. Given a packet to be transmitted with a deadline D, and a specific transmitter–receiver pair, the problem can be formulated as to find the number of attempts and the relevant sequence of rates to be used for the transmission of that packet, with the twofold goal of minimizing the residual transmission error probability, while ensuring the packet is delivered within its deadline. This reflects in the solution of the following minimization problem

$$\min_{N \leq N_{max},\ r^{(i)} \in \mathcal{R}} \mathcal{L}\left(L, S, N, r^{(1)}, r^{(2)}, \ldots, r^{(N)}\right) \tag{2}$$

subject to the constraint

$$\max_{N \leq N_{max},\ r^{(i)} \in \mathcal{R}} \mathcal{D}\left(L, S, N, r^{(1)}, r^{(2)}, \ldots, r^{(N)}\right) \leq D \tag{3}$$

[1]For instance, $\mathcal{R}$ can be constituted by the 4 different IEEE 802.11b rates, the 8 IEEE 802.11g ones, or the various Modulation and Coding Schemes available for IEEE 802.11n/ac.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

In Eq. (2), $\mathcal{L}(\cdot)$ is a function that calculates the residual packet error probability for a packet with a payload of L bytes, transmitted to a receiver which perceives a SNR level of S dB, after N consecutive transmission attempts have been carried out at the rates $r^{(1)}, r^{(2)}, \ldots, r^{(N)}$ (where $r^{(i)}$ is the rate selected for the i–th attempt). Moreover, in both Eq. (2) and Eq. (3) the condition $N \leq N_{max}$ has to hold, where N_{max} is the default maximum number of attempts specified by the IEEE 802.11 standard, typically set to $N_{max} = 7$.

The constraint imposed to the minimization function is relevant to the frame delivery time, $\mathcal{D}$. This is defined as the time elapsed from the instant in which a packet starts to be transmitted to the instant in which the transmitter receives the correspondent acknowledgment (ACK) frame. In the considered real–time communication scenario, $\mathcal{D}$ has to be lower or equal to the deadline D.

RSIN is invoked to obtain, within all the possible combinations of N (number of transmission attempts) and the corresponding rates $r^{(1)}, \ldots, r^{(N)}$, the sequence that represents the optimal solution to the problem in Eq. (2), by considering the most updated level of SNR S perceived between the transmitter–receiver pair, the map $\mathcal{F}$ and the constraint of Eq. (3). This solution is constituted by the sequence of rates at which any single attempt of transmitting the packet has to be carried out.[2]

Considering that the minimization problem through the map $\mathcal{F}$ assumes a probabilistic behavior, the maximum value of $\mathcal{D}$ to be used in Eq. (3) is determined under the worst–case assumption that the first $N-1$ consecutive attempts are failed, whereas the N–th one is successful. Therefore, the expected maximum delivery time can be expressed as

$$\max \mathcal{D}\left(L, S, N, r^{(1)}, \ldots, r^{(N)}\right) = N \cdot t_{DIFS} + t_{data}(L, r^{(1)}) + \sum_{i=1}^{N-1} \left[t_{ACK\_TO}(r^{(i)}) + t_{slot} \cdot \max[I_{bo}(i)] + t_{data}(L, r^{(i+1)})\right] + t_{SIFS} + t_{ack}(r^{(N)}) \tag{4}$$

where t_{DIFS} and t_{SIFS} are the duration of the DIFS and SIFS periods, respectively, while the term $t_{data}(L, r^{(i)})$ represents the actual transmission time, at the i–th attempt, of a frame with payload of L bytes at rate $r^{(i)}$. Moreover, $t_{ACK\_TO}(r^{(i)})$ is the ACK timeout, *i.e.* the maximum time a node waits for the reception of an ACK frame before considering its transmission as failed. Then, $t_{ack}(r^{(i)})$ is the time to transmit the ACK frame given that the rate $r^{(i)}$ is used to transmit the originating data frame. Finally, the IEEE 802.11 CSMA/CA procedure introduces after each failed attempt a random backoff time, *i.e.* $I_{bo}(n)$, drawn from a uniform distribution. The maximum duration of such a time at the i-th transmission attempt is a multiple of the slot time t_{slot}, depends on the contention window (CW) length and can be expressed as $2^{i-1} \cdot (CW_{min} + 1) - 1$. The aforementioned values can all be retrieved from the IEEE 802.11 specifications, and clearly depend on the selected physical layer.

[2]It is worth noting that, in general, the same rate could be used for consecutive attempts.

B. Solution of the Optimization Problem

The solution of the problem formulated by Eq. (2) and Eq. (3) requires to deal with some issues that impact on the practical implementation of RSIN.

The first issue is concerned with the possible existence of more than one valid solution. Indeed, it is likely that a set of optimal rate sequences along with the corresponding number of attempts exist, each solution satisfying both Eq. (2) and Eq. (3). For instance, if the deadline is long enough, it may happen that several combinations, all constituted by a sequence of attempts at the minimum transmission rate, would easily solve the optimization problem.

However, since in such cases RSIN has to perform some further selection steps, we can define some suitable selection rules to further optimize the final solution. First, among the above set of solutions, RSIN selects the sequences with the minimum number of transmission attempts. Indeed, the lower the number N, the lower the jitter on frame delivery, since less backoff procedures are necessary to successfully complete the transmission. Actually, even after this step it is not ensured to obtain a single solution. Therefore, considering that the remaining solutions already ensure the minimization of the transmission error probability (this is a priori, by design of RSIN) and the minimum number of transmission attempts, RSIN will eventually choose the rate sequence that minimizes the delivery time $\mathcal{D}$, without affecting any of the previous constraints. This will also ensure to increase the Real–Time Throughput (RTT), a further meaningful performance indicator, since unnecessarily low–rate combinations are avoided.

Another important issue is concerned with the solution of Eq. (2) and Eq. (3). Indeed, such a solution actually represents a non–linear problem, that depends on a high number of variables. For this reason, the search for the exact optimal solution can be carried out through a "brute force" approach, which explores the whole set of possible combinations. This means that, trivially, this brute force algorithm has to iterate on each one of the $(N+1)$-tuple of type $N, r^{(1)}, \ldots, r^{(N)}$, check if the constraint in Eq. (3) is met, and calculate the residual error probability. Unfortunately, such an approach may lead to a considerable number of iterations and, possibly, to high processing times. Indeed, if R is the number of available MCSs and N that of the available transmission attempts, the resulting number of iterations is in the order of $\propto R^{N+1}$. Thus, for example, in the common case of R=8 and N=7, RSIN would have to complete ~19 million iterations.

It is hence necessary that adequate strategies to limit the number of iterations are undertaken. In this direction, the most immediate option is the reduction of the solution space size. Thus, a limit on the number of subsequent transmission attempts N_{max} may be imposed (this has the further benefit of limiting the jitter on the frame delivery time). However, the most important assumption introduced for the solution of the problem in Eq. (2) is the constraint that the resulting sequence of transmission rates has to contain only monotonically decreasing values:

$$r^{(1)} \geq r^{(2)} \geq \ldots r^{(N-1)} \geq r^{(N)} \tag{5}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

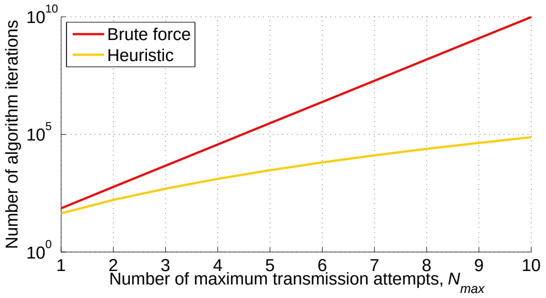

Fig. 2. Number of iterations carried out by the RSIN algorithm.

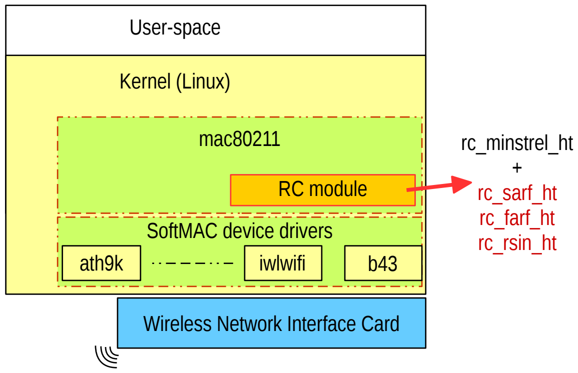

Fig. 3. Schematic representation of the internal Linux kernel structure relevant to the IEEE 802.11 stack.

This is highly reasonable, since progressively reducing the transmission rate of the subsequent attempts in an IEEE 802.11 network, as this assumption does, allows in general to decrease the residual packet transmission error probability, which represents the goal of the RSIN technique. With the above assumptions we obtain a "heuristic" solution of the problem formulated by Eq. (2), that allows to considerably decrease the number of iterations, with respect to the "brute force" approach, as can be observed in Fig. 2, while providing the same results to the constrained minimization task.

IV. Experimental Setup

The goal of this work is to provide an experimental validation and performance assessment of the proposed RSIN algorithm in a real–life industrial scenario. To this aim, it is required its actual implementation on real devices, as well as the deployment of an adequate prototype network that emulates a typical industrial communication scenario.

A. Implementation of the RA Strategies

We implemented the RSIN technique in some real devices based on commercial Wireless Network Interface Cards (WNICs), along with both SARF and FARF. Furthermore, we have taken into consideration the widespread Minstrel RA algorithm [19] which is commonly adopted by several general purpose WLAN devices. Since Minstrel was not designed for industrial applications, we adequately tuned it following the guidelines proposed in [3].

To provide an effective implementation, we leveraged on the IEEE 802.11 networking architecture provided by the Linux kernel, briefly depicted in Fig. 3. Within this framework, any implementation of RA techniques has to reside within the `mac80211` kernel module, as highlighted in the rightmost part of the figure. At the beginning of a packet transmission procedure, any RA algorithm has to provide the WNIC driver with the list of rates to be used for each subsequent transmission attempt (as it is done by the Retry Chain defined by Minstrel). In the case of RSIN, such a list is that obtained from the solution of the optimization problem described in Section III. Since the computational burden of the RSIN algorithm may impact on the performance of the stations that use it, we carried out a specific assessment to estimate the time it takes to complete the solution of the problem formulated by Eq. (2) and Eq. (3) on the personal computers used in the prototype network, that will be presented in the following. Under the assumption that the set of MCSs contains eight different transmission rates, and that a frame may eventually undergo up to four retransmissions, this processing delay is actually bounded to $50\,\mu$s. Clearly, the implementation of RSIN on different devices could require different times.

In our implementation, the RSIN algorithm is called by the WNIC immediately after the reception of a frame (that carries the measured SNR value) from the partner, and hence its execution has to be concluded before the next transmission. This always happened in all the tests we carried out. However, in case the execution of the algorithm is not completed before the next scheduled transmission, we adopted the choice of transmitting at the last selected rate. From a practical point of view, this reflects in a slightly reduced responsiveness of RSIN, that actually would employ more time to adapt to channel status variations.

The above described evaluation of the RSIN execution time clearly refers to the most general operational context in which the frames to be transmitted have different payloads and deadlines, so that each new transmission requires a new complete execution of the algorithm. However, for the case in which all frames share the same length and are subjected to the same deadline, an alternative, more convenient, solution could be devised. Indeed, given the map $\mathcal{F}$, the WNIC may initially (off–line) execute the RSIN algorithm to build a look–up table where the final rate sequence is stored for each possible value of the SNR. This is actually feasible, since in this case the unique variable left is the SNR value. Consequently, the selection of the suitable rate sequence to be used for the transmission of a packet simply reduces to a search procedure within the look–up table, with a considerable reduction of the computational burden.

As a concluding remark, in our implementation the map $\mathcal{F}$ exploited by RSIN has been retrieved through the extensive measurement campaign described in [18].

B. Prototype Network

The software modules that implement the aforementioned RA techniques have been introduced in some desktop workstations (Dell Optiplex PCs, models 745, 755 and 960), all running the Ubuntu 14.10 Operating System based on the

TABLE I
IEEE 802.11 PARAMETERS (2.4 GHz BAND)

Description	Value
MIMO configuration	2×2 STBC
Channel Bandwidth	40 MHz @ 2.4 GHz
Modulation and Coding Schemes (MCS)	0-7
Transmission rates	13.5, 27, 40.5, 54, 81, 108, 121.5, 135 Mbit/s
Slot Time	9 µs
Distributed Inter-Frame Space (DIFS)	28 µs
Short Inter-Frame Space (SIFS)	10 µs
Max number of MAC–layer retries, N_{max}	7
Payload size	50 Bytes

Linux kernel version 3.16.4. The adopted workstations were equipped with WNICs by TP-LINK (models TL-WN851ND and TL-WN881ND), each one compliant with the IEEE 802.11n standard, and allowing 2×2 MIMO operations. The cited WNICs exploit an Atheros AR9287 chip, so that they leverage the "SoftMAC" device principle, allowing a fine–grained control of the transmission path from the kernel–space device drivers. In this specific case, they are handled by the open–source `ath9k` module.

The above workstations have been deployed in a prototype network that comprises three nodes, as schematically represented in Fig. 4. Such a network allows an effective control of devices and wireless medium, without sacrificing the generality of the obtained results. The network is configured in infrastructure mode, where one workstation behaves like an Access Point (AP), while the other ones act as IEEE 802.11 stations (STAs) associated to the AP. The network is designed to emulate an industrial configuration, where a controller node (the AP) is in charge of polling the attached sensors/actuators (the STAs). To this aim, we developed a software that implements the desired exchange of packets between network nodes at the Data Link layer. Such an application is clearly based on a master–slave architecture, where the master generates polling requests on a periodic basis, to which each slave responds immediately.

All the experiments have been carried out on an IEEE 802.11n network. The configuration parameters for both the PHY and MAC layers were set in agreement with the analysis provided by [18]. As a consequence, we set up a 2×2 MIMO configuration on 40 MHz channels in the 2.4 GHz ISM band with the Space–Time Block Coding (STBC) option enabled to increase reliability. The main network parameters adopted in these tests are summarized in Table I.

The experimental measurements have all been carried out in a research laboratory where, unfortunately, a complete electromagnetic isolation was not achievable. However, we selected a channel which was not steadily used by other WLANs by monitoring the surrounding environment with a real–time spectrum analyzer.

As far as the physical channel is concerned, it is worth remembering that an industrial environment is typically affected by quite relevant fading effects, reflections from metallic surfaces, possibly long communication distances and multi–path interference. Most of these aspects have been taken

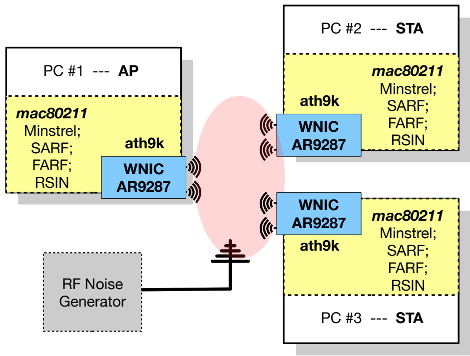

Fig. 4. A representative sketch of the measurement setup.

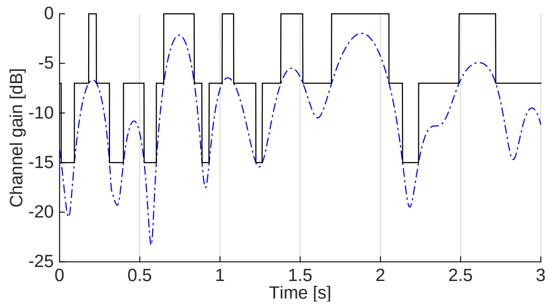

Fig. 5. A snapshot of the path gain obtained using IEEE 802.11 TGn channel model "F", and the relevant quantization with three levels used for synthesizing the injected noise.

into consideration by the IEEE 802.11 Task Group n (TGn) that developed adequate channel models, including model "F" specifically conceived to describe the industrial scenario [20]. A realization of such a channel is provided in Fig. 5 (dashed blue lines), in terms of channel gain behavior over time.

To emulate in practice such a channel, we used an RF signal generator (an Agilent E4433B) as a controlled artificial source of impairments. Specifically, the instrument was set to yield a wide–band AWGN–like noise, whose power level was modulated to mimic the fluctuations of the channel gain as described by the considered model "F". However, for feasibility reasons, the channel gain behavior was quantized using three levels and the resulting pattern, described by the solid black lines in Fig. 5, was adopted to modulate the RF generator power. This artificial disturbance, centered on the carrier frequency of the selected channel, was then injected on the medium through a directional antenna with the main lobe directed toward all the WNICs.

As can be inferred from Fig. 5, the synthesized channel will result in three significantly different SNR levels at the receivers. The lowest one has been calibrated to block any transmission but those at the lowest rate, which will be anyway impaired as well. The intermediate level will seriously impair only the highest Modulation and Coding Schemes (MCSs).[3] Finally, in the absence of injected noise, all transmission rates can be exploited without significant errors, even if occasionally some interference from co–located mobile devices may arise.

[3]This notation is used by IEEE 802.11n to identify the different physical layer options such as channel bandwidth, modulation, etc.

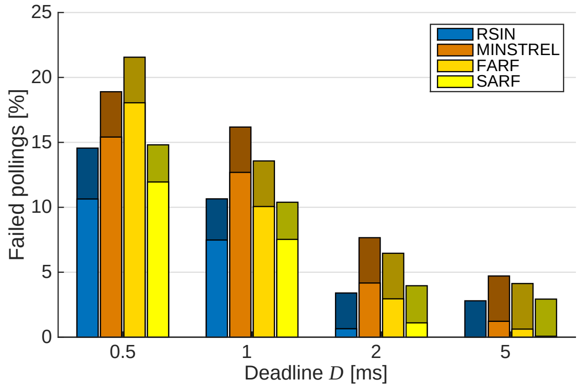

Fig. 6. Comparison among the different RA techniques with respect to their ability to satisfy a given transmission deadline.

V. Performance Evaluation

We present here the outcomes of the performance assessment carried out on the proposed RSIN as well as on the other RA techniques, to carefully characterize their behavior with respect to metrics of interest for industrial communication.

A. Experimental Assessment of RA Strategies

The first experiments we present are concerned with the ability of a station to successfully deliver a packet within a specific deadline.

In these experiments, the AP in Fig. 4 continuously polls the two STAs. For each query, the AP sends a request frame to the addressed STA which, consequently, answers with a response one. If a STA does not answer within a specific timeout (set to a value much higher than the deadline), the polling is considered as failed and the AP moves to the subsequent STA. Several experimental sessions have been carried out for different deadline values, each one comprising 10.000 network cycles. Then, we analyzed, among the received packets, the number of missed deadlines for each RA scheme. The obtained results are reported in Fig. 6.

It is worth considering that in the experiments a failure may also arise, so that a transmission results completely unsuccessful and the packet lost. These occurrences, have been considered separately from those relevant to a successful transmission but with a delivery time exceeding the deadline.

To this regard, Fig. 6 summarizes the aforementioned results for some significant deadline values. Each bar is subdivided in two parts, with the lower one relevant to the percentage of missed deadlines on the total number of delivery attempts, whereas the upper part reports the percentage of transmission failures. Focusing only on the percentage of missed deadlines, the figure highlights that the proposed RSIN algorithm outperforms all the other RA schemes.

A further performance indicator of interest in this assessment is represented by the statistics of the delivery time of the frames involved in polling operations, that are presented in Table II. The results are relevant to all the considered RA techniques. In particular, to provide a more exhaustive assessment, the behavior of RSIN has been analyzed for two different deadline values, namely 0.5 ms and 5 ms. Moreover,

TABLE II
Delivery time statistics

RA technique	Mean	Standard Deviation
RSIN, D=0.5 ms	502.7 µs	325.9 µs
RSIN, D=5 ms	477.2 µs	304.6 µs
Minstrel	635.2 µs	883.1 µs
FARF	630.0 µs	947.8 µs
SARF	516.7 µs	458.7 µs

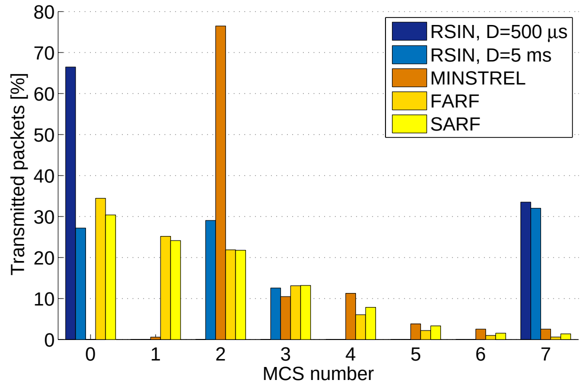

Fig. 7. Histogram of employed transmission rates with different RA algorithm

since in the experimental setup the channel behavior was the same for each packet transmission, the values provided in the table have been obtained taking into account all the data packets exchanged in the network cycles.

As can be seen, RSIN shows a better behavior than all other techniques, since it allows to achieve lower values of both mean and standard deviation of the delivery time. Particularly, with respect to SARF, which showed a comparable performance in terms of missed deadlines and lost packets, RSIN is able to provide a significantly lower standard deviation. This represents a meaningful result, since such a metric is closely related to the jitter on packet delivery.

As a final analysis, we present in Fig. 7 the distribution of the IEEE 802.11n MCSs as selected by the different RA techniques during network operations (the correspondent transmission rates can be inferred from Table I). As can be seen, the Minstrel technique typically settles around MCS 2-4, whereas both FARF and SARF tend to prefer lower MCSs to higher ones. Conversely, RSIN is able to select the lowest as well as the highest MCSs (in agreement with the channel status) confirming in this way its effectiveness.

B. Simulation Assessment

The experimental evaluation discussed so far could not deal effectively with the case of more complex networks, for practical reasons, since their implementation would have required a significantly higher number of nodes, as well as larger and different test environments.

As a consequence, we decided to perform this second part of the performance assessment through a simulation analysis. Therefore, we devised an effective simulation framework, that has been purposely designed and implemented in the Matlab® environment. In the simulator, we implemented all

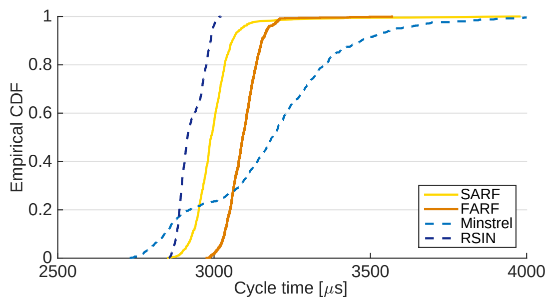

Fig. 8. ECDF of the cycle time, payload of 50 Bytes.

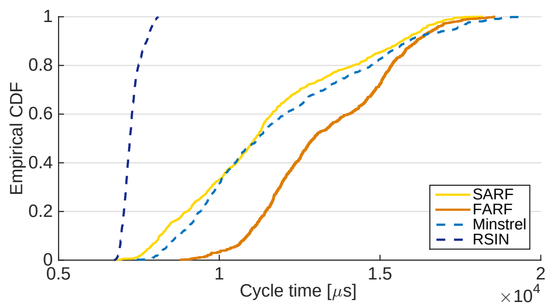

Fig. 9. ECDF of the cycle time, payload of 500 Bytes.

the relevant modules of the IEEE 802.11 protocol, with a special attention to model the behavior of the CSMA/CA procedure, the characteristics and the parameters of the IEEE 802.11n amendment. Also, in the simulation environment the full IEEE 802.11 TGn channel model "F", as depicted by the blue dashed lines in Fig. 5, has been exploited. Finally, we implemented all the RA algorithms discussed in this paper.

As far as the simulation setup is concerned, the network topology is actually identical to that discussed in Section IV-B. Nonetheless, the network can be now composed by n wireless sensors/actuators (typically, we set n=10), attached to the AP through IEEE 802.11n links. The distance between any node and the AP is randomly selected at each new simulation.

In this context, among several meaningful performance indicators, we concentrated our attention mainly on the network *cycle time*, defined as the time required at the controller to complete the polling procedure on all the attached slaves. Indeed, the communication protocol is the same one described in the previous subsection, and hence the controller sequentially polls all the slaves.

We leveraged on the simulation environment to analyze two representative traffic profiles, the first one characterized by typical industrial small–sized packets (the same 50 bytes payload adopted in the experimental measurements), while the second profile targeted at emulating multimedia real–time traffic, with payloads in the order of hundreds of bytes. Given a set of simulation parameters, 10.000 network cycles are performed, and the analysis is repeated ten times with different nodes placement within the environment.

A first set of outcomes is reported in Fig. 8, which shows the Empirical Cumulative Density Function (ECDF) of the cycle time. The deadline imposed for the delivery of each packet, set in the RSIN algorithm, is equal to 500 µs. Comparing the trend obtained for RSIN with the other ones, it is evident that, under the same channel and network conditions, this strategy is able to provide a cycle time considerably lower and more stable. SARF and FARF share a quite similar trend, even if with steadily higher values, and are also characterized by a long tail of cycles needing more than 3.2 ms to complete. Minstrel is sometimes able to reach lower cycle times, but at the expense of an increased jitter, and of a non–negligible percentage of cycles needing more than 3.5 ms to conclude.

In a second set of simulations, the configuration was modified so that the response packets from slaves have a payload length of 500 Bytes, to which a deadline of 1.5 ms is associated. The obtained outcomes are presented in Fig. 9, which highlights even more evidently the performance gain obtained by RSIN. Indeed, while the increased payload size clearly led to longer packet transmission times and also to an increased chance of delivery failure, the cycle time variability is again very limited when RSIN is adopted. Conversely, the other three considered RA schemes are clearly unable to provide satisfactory results, since the cycle time values are distributed between 7 ms and 20 ms. This behavior can be explained by the fact that RSIN performs an optimization step based on a more complete set of constraints with respect to the other schemes. In particular, it relies on a better estimation of the channel status and it is explicitly aware of the application level deadline. Moreover, Fig. 9 clearly puts in evidence that the choice of always retransmitting at the lowest available rate, as done by both SARF and FARF, leads to longer cycle times, especially for increased payloads.

A summary of the simulation outcomes is reported in Table III, which allows to draw some further considerations. RSIN, besides providing a lower average cycle time, is also always able to achieve a far lower standard deviation. This is observed both for the case of a payload of 50 Bytes, and when the payload is increased to 500 Bytes, where the standard deviation is almost an order of magnitude lower with respect to the other techniques. Another aspect that deserves attention is the achievable real–time throughput (RTT), that is, the net transfer speed of data bytes in the unit of time, relevant only to real–time data flows. Table III shows that RSIN provides always a higher RTT than the other RA schemes. However, while in the first scenario this index is inherently limited by the high network overhead compared to the small payloads, in the second scenario RSIN is much more able than the other algorithms to exploit the network resources, and as such delivers a higher quantity of real–time data. It is worth observing that the results of the simulations are not directly comparable with those of the practical experiments. Indeed, the values reported in Table II are considerably higher than those of Table III (normalized by the number of network nodes) since the former are necessarily affected by the internal delays of components.

As a final experiment, we performed an assessment of the dynamic rate selection carried out by the different RA techniques. To this regard we emulated the occurrence, on the network, of a short deep fade on the SNR value. Fig. 10 reports the behavior of the transmission attempts as they were

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

TABLE III
Cycle time and Real–Time Throughput for all RA algorithms

	RSIN	SARF	FARF	Minstrel
Metric	**L_{data}=50 Byte**			
Average cycle time [ms]	2.93	3.00	3.09	3.19
Standard deviation [µs]	45	130	77	333
RTT [Mbit/s]	2.46	2.41	2.33	2.29
	L_{data}=500 Byte			
Average cycle time [ms]	7.28	11.47	13.34	11.87
Standard deviation [µs]	364	2883	2440	3001
RTT [Mbit/s]	5.46	3.68	3.07	3.55

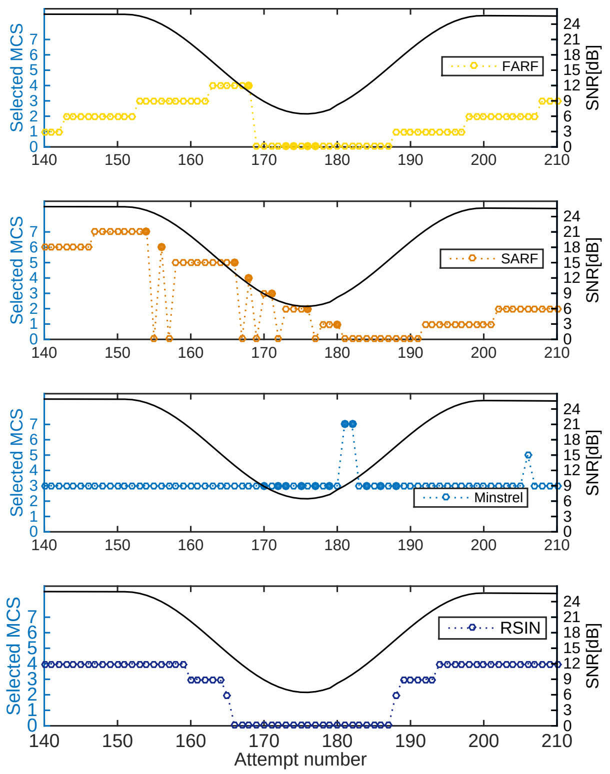

Fig. 10. Rate selection behavior for all the considered RA algorithms.

performed by a specific station on the network. The figure shows, on the x–axis, the attempt number, that is, the progressive number of transmitted frames taking into account also all the retransmissions. In particular, empty circles represent successfully delivered frames, while filled circles represent failed transmissions. The figure considers a small snapshot kept during network operations and highlights, in the solid black line the behavior of the SNR level. This situation, though quite severe to deal with, instantaneously modifies the probability that frames can be delivered successfully, and hence stresses the ability of a RA scheme to promptly address this change. As it can be observed from the lowest figure, RSIN is the only RA algorithm able to track and correspondingly adapt to the channel behavior, without the need of retransmissions. All the other techniques required a (variable) number of retransmissions in order to adapt.

VI. Final Remarks

In this paper we presented a new RA algorithm, named RSIN, particularly targeted at real–time industrial traffic profiles. RSIN leverages on the knowledge of the wireless channel status and, on this basis, selects the sequence of transmission rates to be used for packet transmission by solving a constrained optimization problem. The performance figures of the proposed RSIN algorithm have been discussed and compared with those of SARF, FARF and Minstrel, based on the outcomes of an extensive measurements campaign conducted on real devices, on which all the aforementioned RA techniques have been implemented. Moreover, we showed through simulations the scalability of RSIN on a typical, more complex, industrial network. The analysis has highlighted that RSIN is able to outperform all the previous RA algorithms in terms of both reliability and timeliness.

Some further interesting developments in the framework of MRS for real–time communications can be outlined. As a first activity, the behavior of RSIN in case a WNIC is unable to provide the SNR value should be investigated. In this situation, clearly, an accurate SNR estimation has to be carried out to ensure the appropriate behavior of RSIN. Moreover, the described techniques need to be implemented on different devices, specifically conceived for the industrial scenario, for instance, sensors, actuators and controllers. In this way, the computational burden of the implemented RA strategies could be adequately assessed.

Appendix A
Pseudocode for the RSIN algorithm

The parameters used by the pseudocode of the RSIN algorithm are: the frame length L, the SNR level, the maximum allowed number of transmission attempts N_{max}, the available set of transmission rates $\mathcal{R}$ and, finally, the deadline D.

The procedure returns the optimal number of transmission attempt N_{opt} to be carried out, and the relevant set R_{opt} of transmission rates to be used for each attempt.

The core of the pseudocode is constituted by rows 6-30. The parameter ϵ has been added for the calculation of the minimum number of attempts required by a particular combination (rows 16 and 18) to avoid that the algorithm returns with high probability the trivial solution given by the selection of MCS 0, for all the transmission attempts.

In the last section of the code (rows 32-38), the best solution (*i.e.* lowest residual error probability) is selected among those calculated in the previous main cycle.

1: **procedure** (N_{opt}, R_{opt})=RSIN$(L, SNR, N_{max}, \mathcal{R}, D)$
2:   **for** $i \leftarrow 1$ **to** N_{max} **do** ▷ Internal vector initialization
3:     $\mathbf{R}^{\mathbf{opt}} \leftarrow \mathbf{R}^{\mathbf{opt}} \cup \emptyset$
4:     $\mathbf{L} \leftarrow \mathbf{L} \cup 1$
5:   **end for**
6:   **for** $n \leftarrow 1$ **to** N_{max} **do** ▷ Main cycle
7:     **for each** $(r_1, \ldots, r_n) \in \mathcal{R}^n$ **do** ▷ Ordered satisfying Eq. (5)
8:       **if** $\mathcal{D}(L, SNR, n, r_1, \ldots, r_n) > D$ **then**
9:         **continue**
10:       **end if**
11:       $l \leftarrow \mathcal{L}(L, SNR, n, r_1, \ldots, r_n)$
12:       **if** $l < \mathbf{L}[n]$ **then**
13:         $\mathbf{R}^{\mathbf{opt}}[n] \leftarrow (r_1, \ldots, r_n)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616327, IEEE Transactions on Industrial Informatics

$\mathbf{L}[n] \leftarrow l$
else if $l == \mathbf{L}[n]$ **then**
$a \leftarrow \min_{m \leq n} \{\mathcal{L}(L, SNR, m, r_1, \ldots, r_m) < \epsilon\}$
$r^{opt} \leftarrow \mathbf{R^{opt}}[n]$
$a^{opt} \leftarrow \min_{m \leq n} \{\mathcal{L}(L, SNR, m, r^{opt}[1], \ldots, r^{opt}[m]) < \epsilon\}$
if $a < a^{opt}$ **then**
$\mathbf{R^{opt}}[n] \leftarrow (r_1, \ldots, r_n)$
$\mathbf{L}[n] \leftarrow l$
else if $a == a^{opt}$ **then**
if $\mathcal{D}(L, SNR, n, r_1, \ldots, r_n) < \mathcal{D}(L, SNR, n, r^{opt})$ **then**
$\mathbf{R^{opt}}[n] \leftarrow (r_1, \ldots, r_n)$
$\mathbf{L}[n] \leftarrow l$
end if
end if
end if
end for
end for
$l_{min} \leftarrow 1$
for $n \leftarrow 1$ **to** N_{max} **do**
if $\mathbf{L}[n] < l_{min}$ **then**
$l_{min} \leftarrow \mathbf{L}[n]$
$N_{opt} \leftarrow n$
$R_{opt} \leftarrow \mathbf{R^{opt}}[n]$
end if
end for
end procedure

References

[1] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std., March 2012.

[2] S. Vitturi, L. Seno, F. Tramarin *et al.*, "On the Rate Adaptation Techniques of IEEE 802.11 Networks for Industrial Applications," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 198–208, Feb 2013.

[3] F. Tramarin, S. Vitturi, and M. Luvisotto, "Improved Rate Adaptation Strategies for Real-Time Industrial IEEE 802.11n WLANs," in *Emerging Technologies Factory Automation (ETFA), 2015 IEEE 20th Conference on*, Sept 2015, pp. 1–8.

[4] S. Ding, P. Zhang, S. Yin, and E. Ding, "An Integrated Design Framework of Fault-Tolerant Wireless Networked Control Systems for Industrial Automatic Control Applications," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 462–471, Feb 2013.

[5] A. Ulusoy, O. Gurbuz, and A. Onat, "Wireless model-based predictive networked control system over cooperative wireless network," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 1, pp. 41–51, Feb 2011.

[6] Y.-H. Wei, Q. Leng, S. Han, A. Mok, W. Zhang, and M. Tomizuka, "RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications," in *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, Dec 2013, pp. 140–149.

[7] 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," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 824–835, Feb 2014.

[8] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Aspects," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.

[9] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, pp. 118–133, August 1997.

[10] K. Hong, S. Lee, and N. Golmie, "Throughput Study for Admission Control in IEEE 802.11 DCF with ARF," *Communications Letters, IEEE*, vol. 13, no. June, pp. 432–434, june 2009.

[11] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: a Practical Approach," in *Proceedings of the 7th ACM Intl. Symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2004, pp. 126–134.

[12] *IEC61784 International Standard: Digital data communications for measurement and control. Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems. Part 2: Additional profiles for ISO/IEC8802–3 based communication networks in real–time applications*, International Electrotechnical Commission Std., November 2007.

[13] D. Xia, J. Hart, and Q. Fu, "Evaluation of the Minstrel Rate Adaptation Algorithm in IEEE 802.11g WLANs," in *Communications (ICC), 2013 IEEE International Conference on*, June 2013, pp. 2223–2228.

[14] J. Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras, "Online QoS Management for Multimedia Real-Time Transmission in Industrial Networks," *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 3, pp. 1061–1071, March 2011.

[15] G. Holland, N. H. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proceedings of the ACM SIGMOBILE 7/01*, 2001, pp. 236–251.

[16] T. Sauter, "The Three Generations of Field-Level Networks–Evolution and Compatibility Issues," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3585–3595, November 2010.

[17] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[18] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the Use of IEEE 802.11n for Industrial Communications," *Industrial Informatics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[19] (2015, January) Minstrel specification. [Online]. Available: http://wireless.kernel.org/en/developers/Documentation/mac80211/RateControl/minstrel

[20] V. Erceg, "TGn channel models," *IEEE 802.11 document 03/940r4*, 2004.

Federico Tramarin (S '07, M '12) is a senior Post-Doctoral researcher with the Institute of Electronics, Computer and Telecommunication Engineering of the National Research Council of Italy (IEIIT-CNR) since 2013. He got his PhD in Information Engineering in 2012 with the Electronic Measurement research group of the University of Padova, Italy, which he joined in 2008, after his master degree in Electronic Engineer. His research activities, started in the field of distributed measurements, are recently focused on industrial communication systems, performance characterization and protocol optimization for real-time (wired and wireless) networks. He is a member of the IEEE IES Technical Committee of Factory Automation, and of the IEEE P61158 Working Group.

Stefano Vitturi (M '13, SM '16) received the Laurea degree in electronics engineering from the University of Padova, Padova, Italy, in 1984. He has been a Senior Researcher with the Institute of Electronics and Computer and Telecommunications, National Research Council of Italy, Padova, since January 2002. From 1985 to 2001, he worked at the control and data acquisition system of RFX, a nuclear fusion experiment located in Padova, where he was the head of the Automation and Informatics group. His research interests include industrial automation systems and real-time industrial communication networks.

Michele Luvisotto (S '15) received the Masters degree in automation engineering from the University of Padova, Padova, Italy, in 2014. He is a PhD student with the Department of Information Engineering, University of Padova, Padova, Italy. He has been a visiting researcher at ABB Corporate Research Center in Västerås (Sweden) from March to September 2016. His research interests include wireless networks and real-time industrial communication.