

Seamless Link-level Redundancy to Improve Reliability of Industrial Wi-Fi Networks

Gianluca Cena, *Senior Member, IEEE*, Stefano Scanzio, *Member, IEEE*, and Adriano Valenzano, *Senior Member, IEEE*

Abstract—The adoption of wireless communications and, in particular, Wi-Fi, at the lowest level of the factory automation hierarchy has not increased as fast as expected so far, mainly because of serious issues concerning determinism. Actually, besides the random access scheme, disturbance and interference prevent reliable communication over the air and, as a matter of fact, make wireless networks unable to support distributed real-time control applications properly. Several papers recently appeared in the literature suggest that diversity could be leveraged to overcome this limitation effectively.

In this paper a reference architecture is introduced, which describes how seamless link-level redundancy can be applied to Wi-Fi. The framework is general enough to serve as a basis for future protocol enhancements, and also includes two optimizations aimed at improving the quality of wireless communication by avoiding unnecessary replicated transmissions. Some relevant solutions have been analyzed by means of a thorough simulation campaign, in order to highlight their benefits when compared to conventional Wi-Fi. Results show that both packet losses and network latencies improve noticeably.

Index Terms—Wireless networks, Wi-Fi, IEEE 802.11, Seamless redundancy, Dependability, PRP

I. INTRODUCTION

Wireless communication technologies, and particularly industrial wireless sensor networks (IWSNs) [1], are used more and more in automation scenarios, including real-time control systems [2]. Two of the most popular solutions for process automation at the time of writing are WirelessHART and ISA100.11a [3]. Although they basically rely on IEEE 802.15.4 [4] for frame transmission over the air, mechanisms like channel hopping and channel blacklisting have been included in order to make communication more resilient to interference and disturbance, which often abound in industrial environments [5]. Instead, for factory automation (which has tighter timing constraints), commercial solutions like the wireless interface for sensors and actuators (WISA) and the industrial wireless LAN (IWLAN) were developed by ABB and Siemens, respectively. The former is based on IEEE 802.15.1 (Bluetooth), enhanced by a specific full-duplex access scheme, whereas the latter employs an extension to IEEE 802.11

This work was partially supported by the Ministry of Education, University, and Research of Italy (MIUR) in the framework of the Project Wi-Fact “Wireless FACTory and beyond” (DM53543). Copyright (c) 2016 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. The authors are with the National Research Council of Italy, Istituto di Elettronica e di Ingegneria dell’Informazione e delle Telecomunicazioni (CNR-IEIIT), I-10129 Torino, Italy (e-mail: gianluca.cena@ieiit.cnr.it, stefano.scanzio@ieiit.cnr.it, adriano.valenzano@ieiit.cnr.it).

[6], which exploits one or more centralized coordinators and foresees optimized multiple antennas or radiating cables to ensure reliable communication.

With the exception of IWLANs, IEEE 802.11 networks are usually deemed not reliable enough for real-time control applications [7]. Although *rate adaptation* techniques [8], customarily employed in commercial Wi-Fi equipment, allow tolerating a decrease in the signal-to-noise ratio (SNR) by switching to more robust modulations schemes based on lower transmission rates, they are able to mitigate the instability of wireless communication only in part.

The adoption of techniques like channel hopping and blacklisting in Wi-Fi is probably not a viable option either. Although a specific physical layer based on frequency hopping spread spectrum (FHSS) was foreseen since the initial IEEE 802.11 specification, it never becomes very popular and recently it was declared obsolete. In theory, a mechanism could be layered on top of orthogonal frequency-division multiplexing (OFDM) so that every wireless station (STA) synchronizes to beacons and hops—at faster pace—between channels. However, association of conventional STAs would be precluded, which poses coexistence issues with existing devices.

Ensuring high availability in industrial automation systems that include wireless links can be accomplished by exploiting *diversity* [9]. For example, in the case of Wi-Fi, simultaneous dual-band equipment such as access points (APs) and wireless bridges (WBs) can be used to this purpose. In the case a problem is experienced that blocks communication over a channel for a protracted period of time, the spanning tree protocol (STP) or its rapid variant (RSTP) take care of dynamically selecting a new path, which includes wireless links operating at a different frequency. Unfortunately, recovery times for these solutions can be as long as several seconds [10], which makes them unsuitable for the typical real-time applications found at the shop-floor level.

The extended channel switching (ECS) procedure defined in IEEE 802.11 allows to deal with situations where disturbance affecting the wireless channel grows excessively. This is obtained by letting the AP notify the STAs that the channel frequency is going to be changed. Although the transition is quite fast in this case, frames may be lost (or delayed unacceptably) before the decision to switch channel is made.

A different approach relies on a particular modulation and coding scheme (MCS), known as MCS 32, included in the IEEE 802.11n specification. This technique exploits channel bonding by sending the same frame at 6 Mbps on both halves of a single 40 MHz channel. MCS 32 is noticeably more robust

than other schemes, but frame transmissions require that both the bonded channels be available at the same time, and this may reduce network utilization and increase latency.

Seamless redundancy—as specified in IEC 62439-3 [11], also known as parallel redundancy protocol (PRP)—is probably the best approach to ensure both timeliness and high dependability in control systems based on real-time Ethernet (RTE) technology. In PRP, each frame is sent by transmitters on two distinct networks. The two copies reach their destination(s) through different paths and, possibly, at different times. The first copy that arrives to destination is retained and delivered to the application (or the upper protocol layers), whereas the second one (i.e., the *duplicate*) is discarded. This solution has two advantages: first, frames are lost only when they are dropped on both networks. This means that the likelihood that a packet is not delivered because of faults affecting network equipment and media can be lower (even noticeably lower) than for non-redundant solutions. Second, the transmission latency decreases as well, and corresponds to the minimum among the times taken by all the different copies of each frame to reach the destination through the relevant networks paths.

PRP is meant to both provide seamless *end-to-end* communication redundancy and grant a certain degree of fault tolerance to critical applications in highly-dependable systems. In [12] it was used, together with conventional Wi-Fi equipment, to increase the communication reliability of a wireless link between two nodes. In the following, such a solution will be denoted *PRP over Wi-Fi* (PoW). When coupled with proper medium access control (MAC) overlays [13], [14], [15], [16], PoW can enable quasi-deterministic behavior.

Unfortunately, this approach also exhibits a non-negligible drawback, originating from the transmission of every frame on both channels. This implies that the network load on each single channel is exactly the same as for non-redundant solutions. As a result, the overall traffic over the air is doubled, which could be a nuisance as bandwidth in unlicensed portions of the radio spectrum is a scarce resource.

Since the quality of communication may vary suddenly and unexpectedly on wireless channels, packet queues in intermediate relay devices (e.g., APs) could grow considerably. As a consequence, latency increases and devices may even run out of memory, hence causing frames to be dropped. In such conditions, preventing unnecessary transmissions over the redundant channels is the key to improve the overall network behavior. Incidentally, doing so also reduces power consumption, which can be quite appealing in mobile devices.

In the following, the *Wi-Fi Redundancy (Wi-Red)* solution is presented, which is aimed at providing seamless *link-level* redundancy in IEEE 802.11 networks. It also includes mechanisms that make it less bandwidth-hungry than PoW, by reducing the amount of duplicate frames sent on air. Unlike approaches like [17], *Wi-Red* relies on proven Wi-Fi technology, which makes it a natural solution to deploy wireless extensions of Ethernet, especially in industrial plants. Moreover, *Wi-Red* is conceived to grant perfect coexistence with conventional Wi-Fi.

The basic principles behind *Wi-Red* were first introduced in

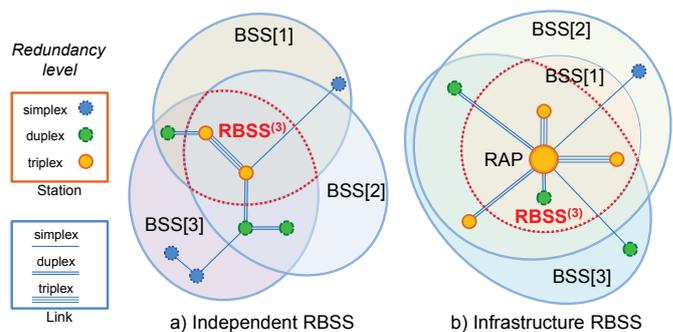


Fig. 1. RBSS and redundancy levels (station/link).

[18] and [19]. With respect to those preliminary papers, several relevant aspects have been taken into account here, including in particular the extension of redundancy mechanisms to multiple channels, their relationships with Quality of Service (QoS), and their effects on the order of received frames. Instead, frame aggregation and block acknowledgment are not considered in this paper, since possible benefits due to the inclusion of optimizations in redundant transmission are still questionable in these cases and need further investigation. Moreover, such mechanisms are useful when most frames spend non-negligible amounts of time queued for transmission, but this is unlikely to occur in the case of process data exchanged in typical real-time industrial applications.

The resulting reference architecture is suitable for describing the existing proposals and general enough to provide a solid basis for future enhancements, including the definition of additional heuristics and cross-layer integration with upper protocols (e.g., the MAC) in order to improve determinism further.

The paper is structured as follows: in Section II a general framework is introduced to deal with seamless redundancy in Wi-Fi, while in Section III the *Wi-Red* protocol is presented with focus on two mechanisms that prevent transmissions of useless duplicates. Section IV describes the testbed we used for experimental evaluation. Finally, Section V reports on the results of some simulation campaigns and performs a quick comparison of the different techniques.

II. WI-FI REDUNDANCY FRAMEWORK

Basically, *Wi-Red* is seamless redundancy—as defined by PRP—applied to Wi-Fi. Unlike PRP, which was conceived as an end-to-end solution to provide fault tolerance in Ethernet-based systems, *Wi-Red* operates on each single wireless link—either between a STA and the AP in an infrastructure basic service set (BSS) or between two STAs in an independent basic service set (IBSS). It aims at enhancing the quality of communication in Wi-Fi to a degree that, in theory, could resemble wired connections.

Another difference between PRP and *Wi-Red* is the way duplicate copies of the same frame are detected by receivers. In PRP a redundancy control trailer (RCT) is appended to every Ethernet frame for its unambiguous identification by end nodes. This is not needed in *Wi-Red*, at least in theory,

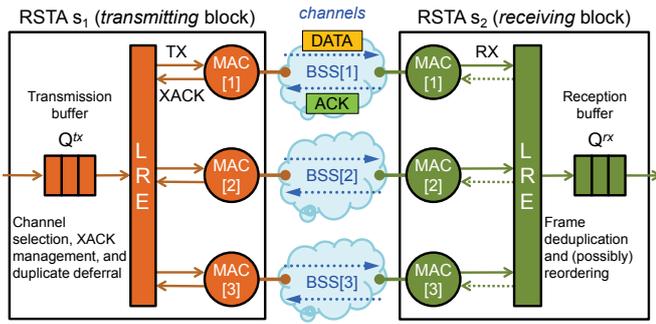


Fig. 2. RSTA conceptual block diagram (triplex redundancy).

since the *sequence control* field included in each IEEE 802.11 MAC protocol data unit (MPDU) can be used to this extent. In practice, keeping sequence numbers aligned across different MACs operating on distinct channels is hardly feasible with existing chipsets. Therefore, software implementations of *Wi-Red* may still make use of RCT. This can be obtained quite easily, since the maximum transmission unit (MTU) in IEEE 802.11 is larger than the Ethernet MTU.

A. Network Architecture

As shown in Fig. 1, a *Wi-Red* network consists of a number of *redundant stations* (RSTAs), each of which is joined to two (or more) distinct BSSs. Generically speaking, the intersection of these BSSs is referred to as *redundant BSS* (RBSS) and includes all the related RSTAs. In the following text, s_m is the m -th RSTA in the considered RBSS, while each BSS in the RBSS is denoted $BSS[k]$, $k \in [1 \dots R_N]$, where R_N is the *network redundancy level* (simplex, duplex, triplex, and so on).

BSSs overlap in space, but operate on different channels or, even better, distinct frequency bands (e.g., 2.4, 5, and 60 GHz). Besides RSTAs, each BSS may include conventional, non-redundant STAs as well. The RBSS coverage area comprises at least the places where RSTAs are located, and corresponds to the region where full-redundant communication is allowed. Non-redundant (or partially-redundant) communication between RSTAs may be possible in areas larger than the RBSS.

B. Redundant Stations

As depicted in Fig. 2, an RSTA includes several independent entities, or *substations* (sub-STAs), each one being a member of one BSS in the RBSS. Sub-STAs of RSTA s_m are referred to as $s_m[k]$, $k \in [1 \dots R_{s_m}]$, where R_{s_m} is the *station redundancy level* of s_m (i.e., the number of channels simultaneously supported by s_m). Each sub-STA implements the physical (PHY) and MAC protocol layers. Although both layers have to comply to IEEE 802.11 and full compatibility must be retained with Wi-Fi equipment, slight changes might be needed to the MAC in order to enhance performance.

Operations of the different sub-STAs in the same RSTA are co-ordered by a *link redundancy entity* (LRE). On the transmitter side, LRE takes care of dispatching frames, queued in the *transmission buffer* (Q^{tx}), to the MACs of the local sub-STAs. Suitable mechanisms should be introduced in order to prevent

useless transmission of duplicates. RSTA implementations can make use of either a single buffer Q^{tx} , shared among all the sub-STAs, or separate transmission buffers $Q^{tx}[k]$ for different sub-STAs. In the first case each buffer entry has to include also information about the transmission state of the relevant frame on every BSS (e.g., waiting, transmitting, retransmitting, aborting, transmitted, or discarded). On the receiver side, LRE removes duplicates from the *reception buffer* (Q^{rx}) and, possibly, reorders frames before they are delivered to the upper layers.

C. Redundant Networks

Real RBSSs are likely to be deployed as infrastructure networks. Infrastructure RBSSs are based on *redundant access points* (RAPs). The RAP (here denoted s_0) relays frames between either different RSTAs or any RSTA and the distribution system (DS), i.e., the Ethernet backbone. In this case, R_N is the same as the station redundancy level R_{s_0} of the RAP. Definitions are slightly different for redundant IBSSs, as communications take place between STAs directly. As this kind of networks is less significant from a practical point of view, it will not be considered in this paper.

The behavior of RSTAs, as observed on any BSS, is not noticeably different from conventional STAs. This means that each RSTA has to join each BSS in the RBSS separately. To enable this, the conventional Wi-Fi procedures for AP association must be extended, so as to allow each RSTA to discover all the redundant channels in the RBSS. For instance, this information can be included in beacons sent by the RAP and other management frames (association, probe, and so on), by extending their basic format. Doing so does not impair backward compatibility with legacy STAs.

To achieve *full redundancy*, every RSTA must be member (through distinct sub-STAs) of all the BSSs in the RBSS. This is possible provided that: a) all R_{s_m} values are the same as R_N , and b) the actual *link redundancy level* R_{l_m} of every RSTA s_m coincides with R_{s_m} (i.e., s_m is within the RBSS coverage area and all its sub-STAs can communicate on the related BSSs). In other words, $R_{l_m} = R_{s_m} = R_N$, $\forall s_m \in RBSS$.

Conversely, R_{l_m} is not requested to be the same as R_N for all RSTAs in order to support *partial redundancy*. For example, a duplex RSTA may associate with a triplex RAP, and R_{l_m} is only duplex in this case. The same happens when a triplex RSTA is located in a place where only two BSSs of a triplex RAP can be reached. In general $R_{l_m} \leq \min(R_{s_m}, R_{s_0})$, and inequality may hold when s_m is outside the coverage area of some BSS in the RBSS. Let $RBSS^{(r)}$ denote an r -level RBSS ($r \leq R_N$), defined as the set of all the RSTAs that are enabled to communicate over links whose redundancy level is at least r , i.e., $RBSS^{(r)} = \{s_m | R_{l_m} \geq r\}$. For these sets the condition $r_1 < r_2 \Rightarrow RBSS^{(r_1)} \supseteq RBSS^{(r_2)}$ holds.

Details about setting up a redundant link and negotiating its minimum allowed redundancy level are outside the scope of this paper. For the sake of simplicity and with no loss of generality, in the following we will assume full redundancy.

III. REDUNDANCY PROTOCOL

In *Wi-Red*, a source RSTA can send multiple copies of the same frame on different channels of a redundant link. Duplicates are then detected and discarded by the target RSTA(s), as it happens in PRP. However, in *Wi-Red* these operations are performed directly by the data-link layer (DL).

In the *basic* version of *Wi-Red*, distinct copies of every frame are concurrently sent on all channels. In this way, the traffic generated by an RSTA on each BSS is the same as a conventional, non-redundant STA in Wi-Fi. When the overall behavior is considered, this approach closely resembles the PoW solution in [12]—even though PoW actually operates above the DL layer. As pointed out in [18] and [19], bandwidth may be wasted and this is not a good option, since the wireless spectrum in unlicensed bands is already crammed.

Two mechanisms were introduced to cope with this drawback, which operate on the transmitter side and are implemented in the LRE. They are aimed at decreasing the overall traffic on air by preventing the transmission of inessential duplicates, while improving the quality of communication at the same time. The first mechanism *reactively* avoids the transmission of duplicates for frames that have already been delivered to the intended receiver. This conceptually simple approach does not exhibit any drawbacks and always brings benefits. The second mechanism, instead, works *proactively*, by deferring the transmission of duplicates for frames that are already in the process of being sent. However, while reducing network traffic, deferral may increase latency. Hence, it has to be managed carefully.

Although the adoption of these *duplicate avoidance* (DA) mechanisms is highly recommended in *Wi-Red*, their implementation is not mandatory. Correct communications shall be granted in the RBSS nevertheless, irrespective of the fact that any of the involved RSTAs be actually able to provide such a kind of support. Both DA mechanisms apply to unicast (*acknowledged*) transmissions only, while no optimization is possible for multicast (*unacknowledged*) communications. Consequently, multicast frames have to be sent on all channels according to the *basic Wi-Red* rules.

Unlike MCS 32, *Wi-Red* exploits *time* diversity besides *frequency* diversity. In fact, unless wireless channels remain idle for most time (that is quite a favorable condition), different copies of the same frame are typically transmitted at different instants. This means that *Wi-Red* may bring benefits for both narrow-band disturbance and short wide-band noise pulses.

In the following, we will take into account the behavior of a single unidirectional redundant link, but this is not a limiting assumption. Since acknowledged transmissions are dealt with by the MAC on a per-link basis, the *Wi-Red* redundancy scheme and the related optimizations are independently applied to each communication between RSTA pairs and, in particular, to each *uplink* and *downlink* between the RAP and an RSTA. Overall, each frame received by the RAP over any redundant uplink undergoes duplicate detection and removal before forwarding. Moreover, as soon as a valid copy of a frame addressed to an RSTA becomes available in the RAP (this holds also for frames received on Ethernet ports), it is

queued for transmission on the relevant redundant downlink.

A. Reactive Duplicate Avoidance

Transmissions of redundant copies *after* a frame has already been delivered successfully on another channel are completely useless and should be avoided. The sender can infer the occurrence of such a situation by exploiting the *acknowledgment* (ACK) mechanism, which is used in the distributed coordination function (DCF) of IEEE 802.11 to confirm the correct delivery of unicast data frames (DCF is the most basic MAC scheme in Wi-Fi, on which all the other coordination functions rely). DA approaches based on this principle are referred to as *reactive duplicate avoidance* (RDA). According to DCF, every time a STA transmits a frame, a timeout (*ACKTimeout*) is started. If an ACK frame is not received from the intended destination before *ACKTimeout* expires, a *retry counter* (RC) is increased and a retransmission is attempted. When RC reaches a predefined *retry limit* (RL), retransmissions end and the frame is discarded. *Wi-Red* optimizes transmissions by means of *cross acknowledge* (XACK) events. An XACK is generated inside RSTA s_m whenever one of its sub-STAs receives an ACK frame for a pending frame exchange. Such an event can typically be inferred from the interrupts raised by wireless adapters at the end of each frame transmission. Every XACK is tagged with the channel k where the ACK was received (that is, $BSS[k]$) and a unique identifier i of the relevant data frame (F_i), which can be easily derived from the MAC sequence control field. On reception of the $XACK(k, i)$ event, every sub-STA $s_m[g]$ in the RSTA, $g \neq k$ (that is, with the exception of $s_m[k]$), searches its transmission queue $Q_m^{tx}[g]$ for a corresponding data frame and, if a match occurs, the matching entry is removed. If the RSTA adopts a single transmission buffer Q_m^{tx} , then LRE is in charge of removing useless duplicates. We call the above mechanism RDA/Q: on most platforms, it can be implemented by introducing changes in the device drivers of wireless adapters.

RDA/Q can be improved by allowing sub-STAs to abort ongoing transmissions, too. To do so, when $XACK(k, i)$ is triggered by $s_m[k]$, the MACs of the other sub-STAs $s_m[g]$ in the RSTA are inspected to determine if frame F_i is currently in the process to be sent. When this occurs, the transmission (or retransmission) of F_i on $s_m[g]$ is aborted. A suitable *abortable* DCF (ADCF) technique was described in [18], which requires MAC modifications but retains full compatibility with DCF. It can be proven that frame transmissions can be safely aborted (that is, without impairing coexistence with legacy Wi-Fi STAs) only between retries. In practice, when an abort request is issued in a sub-STA, its MAC has to wait for the end of the current transmission attempt (including the ACK reception). If the ACK is received before *ACKTimeout* expiration, abortion is unnecessary. Otherwise, the MAC keeps on waiting until *ACKTimeout* expires, and only then it cancels the transmission and resets its *contention window* (CW) as if the frame were discarded. More details about ADCF can be found in [18]. We call RDA/M the adoption of the above MAC-level transmission abort mechanism in conjunction with RDA.

Deep changes are needed to the DCF protocol state machines to implement RDA/M. Instead, a much simpler so-

TABLE I
 MAIN FEATURES OF DUPLICATE AVOIDANCE MECHANISMS TO REDUCE NETWORK LOAD IN *Wi-Red* – FULL SUPPORT (●) / PARTIAL SUPPORT (○).

Operation	RDA/Q	RDA/R	RDA/M	RDA/Q+PDA	RDA/R+PDA	RDA/M+PDA
Removes duplicates from transmission queues upon XACKs	●	●	●	●	●	●
Aborts ongoing MAC transmissions of duplicates upon XACKs		○	●		○	●
Defers further MAC transmissions of duplicates upon heuristics				●	●	●

lution can be devised, we call RDA/R, which is based on a streamlined version of ADCF. In this case, in order to abort an ongoing frame transmission in the MAC of $s_m[g]$, its retry counter $RC_m[g]$ is set equal to the retry limit RL, as suggested in [20]. While RDA/R requires limited changes to DCF, this arrangement is sub-optimal since the check on RC for any given frame is carried out *after* each transmission attempt. This means that frames, which were already fed into the MAC and whose transmission has still to begin at the instant when the abort request is issued, nevertheless undergo a transmission attempt. Preliminary experiments, carried out on a partial implementation of RDA/M, showed that, although it is much more complex than RDA/R, it only has a slight performance edge. Because of its intrinsic simplicity, RDA/R is a satisfactory compromise between benefits and complexity. For this reason it has been considered in the following analysis in the place of RDA/M.

Summing up, upon XACK events RDA/Q is only able to remove duplicate frames queued in the transmission buffers of the sub-STAs, whereas RDA/R (and RDA/M) are also provided with the ability of aborting ongoing MAC transmissions. Conversely, XACKs are not dealt at all by *basic Wi-Red*, which transmits every single frame on all channels. Because of their different ability to prevent duplicate transmissions, we must expect increasing performance when moving from *basic Wi-Red* to RDA/Q and from RDA/Q to RDA/R.

B. Proactive Duplicate Avoidance

A second class of DA mechanisms for *Wi-Red* is meant to *complement* and improve RDA. They spring into action whenever the MAC of a sub-STA concludes a frame transfer and a new frame has to be selected for transmission from the RSTA queue (provided that at least one frame is present there). Let $s_m[k]$ be that sub-STA and F_h the frame in the first position of its queue (i.e., the oldest one), $F_h = \text{head}(Q_m^{tx}[k])$. According to DCF, F_h should be immediately fed into the MAC of $s_m[k]$ for transmission. Performance can be improved by checking whether F_h is already being sent by other sub-STAs in s_m . In this case, the transmission of a duplicate of F_h by $s_m[k]$ can be temporarily deferred and another pending frame in $Q_m^{tx}[k]$ can be selected and sent in the place of F_h .

We call this kind of mechanism *proactive duplicate avoidance* (PDA), as it aims at preventing transmissions that are likely unnecessary, in particular when disturbance and interference are at acceptable levels. In fact, if an XACK is received for a frame whose redundant copy has been deferred, a duplicate (and hence, useless) transmission is avoided. The higher the number of redundant channels, the larger the benefits that PDA techniques achieve. However, special attention must be

paid when dealing with time-critical communications, in which case the effect of deferral on transmission latency has to be carefully evaluated.

Using PDA without RDA, although in theory possible, is mostly pointless in practice. By disabling RDA, in fact, a frame would no longer be removed from all the sub-STAs in the sender following the reception of the related ACK on any channel. Because of deferrals, the frame transmission order may vary across channels, but the overall amount of network traffic remains the same as in *basic Wi-Red*. As a consequence, no noticeable improvements have to be expected from this arrangement over the much simpler PoW. For this reason, PDA should be regarded uniquely as a set of heuristics aimed at improving RDA behavior, by reducing further the likelihood that duplicate copies are sent over the air.

Different approaches can be adopted to decide whether a duplicate transmission has to be deferred. A very simple solution for duplex RSTAs is the *dynamic duplicate deferral* (DDD) technique [19]. DDD works as follows: the MAC in the sub-STA of s_m that works on channel $\bar{k} \neq k$, i.e., $s_m[\bar{k}]$, is checked to know if it is in the process of sending F_h . If the answer is affirmative, the relevant retry counter $RC_m[\bar{k}]$ is checked against a suitable *duplicate deferral threshold* (D_{th}). If $RC_m[\bar{k}] \geq D_{th}$ (that is, F_h has already experienced at least D_{th} transmission attempts on channel \bar{k}) a duplicate of F_h is also sent by $s_m[\bar{k}]$. Otherwise, the next frame F_{h+} , following F_h in $Q_m^{tx}[k]$, is extracted from that queue and fed into the MAC of $s_m[k]$. Conceptually, this corresponds to swapping frames F_h and F_{h+} at the head of $Q_m^{tx}[k]$. As shown in [19], setting $D_{th} = 0$ disables DDD and reverts the system to pure RDA.

DDD adoption increases the RSTA throughput, since redundant channels can be exploited to send different packets at the same time. As long as most transmissions are successful (this happens when channels suffer from mild interference and disturbance), the overall throughput is almost doubled. Conversely, when the quality of channels worsens and a higher number of retransmissions is experienced, which exceeds D_{th} , reliability is preserved by means of frequency/time diversity. Of course, leaving a channel idle is pointless when pending frames have real-time constraints, so deferral should not take place if F_h is the only frame in $Q_m^{tx}[k]$. In this case, the best option is sending a copy of F_h also on BSS[k].

The extension of DDD to triplex or higher redundancy ($R_N \geq 3$) is not difficult. A solution is requiring that queued frames be suitably flagged. In practice, each frame is set as *ready* at its insertion in the transmission queue (unless $D_{th} = 0$, which causes the frames to be marked *undeferrable*). When the frame is moved from the queue to a MAC, its state

is switched to *deferrable*. As soon as RC in the MAC of any sub-STA $s_m[g]$ reaches the threshold D_{th} (that is, when $RC_m[g] \geq D_{th}$), the frame being sent by that MAC is flagged as *undeferrable*. If the RSTA adopts separate transmission buffers, state changes have to be performed for all the frame copies stored in the different queues.

Whenever the MAC of any sub-STA in s_m becomes idle, the first *undeferrable* frame is selected for transmission. If there are no undeferrable frames, the first *ready* frame is taken. Otherwise, when all the queued frames are marked *deferrable*, the first one is transmitted anyway to reduce latency. Whatever the case, frames that were previously discarded by any MAC of s_m shall not be sent again on the same BSS. This has to be checked explicitly only if the RSTA relies on a single transmission buffer Q_m^{tx} . Conversely, frames discarded by $s_m[k]$ are removed automatically from $Q_m^{tx}[k]$ by the relevant MAC. A brief synoptic of the operations carried out by the different DA approaches is reported in Table I.

C. Quality of Service

Improving communication by means of seamless redundancy at the Wi-Fi physical layer certainly helps with ensuring true real-time behavior, but typically it is not sufficient on its own. Mechanisms have to be included in the MAC layer as well, so that STAs (or RSTAs) can coordinate their access to the shared medium, hence lowering interference among devices belonging to the same system as much as possible. To this extent, the point coordination function (PCF) was introduced in the very first IEEE 802.11 specification. Later, it was noticeably improved with the hybrid coordination function (HCF) and, particularly, the HCF controlled channel access (HCCA). A modified version of PCF is currently adopted by IWLAN whereas, to the best of our knowledge, HCCA is not supported by commercial devices yet. In both cases a coordinator takes care of explicitly querying STAs, so as to ensure controlled and collision-free network access.

While advisable, the joint use of *Wi-Red* and deterministic MAC mechanisms is not completely trivial. In fact, cross-layer optimizations (which complement those carried out by the RDA and PDA mechanisms) should be devised in order to maximize benefits. For this reason, in this paper we focus on a much simpler way to achieve QoS in Wi-Fi, that is, the enhanced distributed channel access (EDCA). EDCA was originally conceived to support multimedia traffic. Several solutions have been proposed in the recent literature, which foresee the use of the access categories (AC) defined by EDCA to improve communication determinism in soft real-time wireless control systems as well [13], [14], [21].

QoS can be easily added to *Wi-Red* if EDCA is used in place of DCF. This means that, in each sub-STA, traffic is split into four ACs, namely voice (AC_VO), video (AC_VI), best effort (AC_BE), and background (AC_BK), in decreasing priority order. Each AC is provided with its own transmission queue, and a suitable AC is selected for each QoS-frame, mainly depending on its timing constraints. For instance, the voice category is often envisaged to convey process data for which the transmission latency should be kept as short as possible,

whereas best-effort traffic could be employed for tasks with no particular requirements about time.

While the adoption of EDCA does not affect RDA, some changes may be needed to PDA. For instance, the D_{th} threshold in DDD can be selected on a per-AC basis. In this case, its values are denoted DTH[AC]. From a practical point of view DTH[AC_VO] can be set to a low value (e.g., 0 or 1), to cope with time-critical data. With this arrangement, parallel transmissions of multiple copies of the same frame are started as soon as possible on each channel. By contrast, a higher value (e.g., 3 or more) should be selected for DTH[AC_BK]. In this way, different frames will be sent (as long as possible) on different channels at the same time, thus maximizing the throughput for background traffic at the expenses of increased latency for those frames which experience retransmissions. So that, for any given AC, every RSTA in the RBSS receives the same QoS, DTH[AC] values must be selected on a network-wide basis. For instance, they can be included in beacons sent by the AP, in the same way as the other AC parameters, e.g., the arbitration interframe space number (AIFSN[AC]).

Besides DDD, other PDA schemes can be devised, which rely on more complex heuristics based on, e.g., queuing times. However, this requires different sets of parameters with respect to DTH[AC]. The definition of additional PDA mechanisms, as well as the negotiation of their parameters in the network, are very complex matters and deserve further investigation.

D. Frame reordering

In transmission schemes that exploit diversity (and particularly PDA), frames may be received by the target node(s) in a different order than they were generated on the sender. This behavior is not the same as conventional Wi-Fi, and may introduce problems in upper-layer protocols. LRE can deal with this issue on the receiver side, and reorder frames before they are relayed to the upper layers through the logical link control (LLC) interface. A single *receive buffer* (Q_m^{rx}) is then shared among all the sub-STAs of s_m . Moreover, in order to grant compatibility among different *Wi-Red* implementations, the receiver must not rely on the presence of any DA mechanism on the transmitter side.

The reordering procedure works as follow. Received frames are inserted in Q_m^{rx} in their unique identifier order (that is the order followed by the DL-user on the transmitter side). Frames received out of order introduce gaps in the buffer, and if a frame copy is already present in Q_m^{rx} a duplicate is not created. Q_m^{rx} operates according to the sliding window principle: let F_x be the last frame correctly delivered to the DL-user on s_m and F_{x+1} the frame that was queued on the transmitter side of the redundant link (by the DL-user on the source RSTA) immediately after F_x . When F_{x+1} becomes available in Q_m^{rx} , it is extracted, together with all the *subsequent* frames in Q_m^{rx} with contiguous identifiers (i.e., F_{x+2} , F_{x+3} , etc.), and orderly delivered to the local DL-user. This corresponds to moving the sliding window forward up to the next gap (or until the queue is empty).

If a frame is dropped on all channels (quite unlikely event) a permanent gap appears, which would prevent the

sliding window from advancing, thus blocking frame transfers. Timeouts can be used to circumvent this problem. In principle, a different timeout is started every time the *first* copy of a frame is received and stored in Q_m^{rx} . If the timeout for any frame F_y expires, that frame (F_y), all its *preceding* frames (i.e., frames F_z , $z < y$, which were queued for transmission before F_y), and all the frames following F_y with contiguous identifiers (i.e., F_{y+1} , F_{y+2} , etc.) are extracted from Q_m^{rx} and delivered to the DL-user on s_m . Frames preceding F_y which have not been received yet are definitively lost.

Many aspects of reordering resemble PRP—and, to some extent, the mechanism to deal with block acknowledgment in Wi-Fi—but implementation and optimization details are not discussed here. In practice, three basic policies can be adopted for the delivery of frames to the DL-user, that is:

- 1) *Ordered*: frames are reordered as explained above, and then delivered orderly to the upper layers. In this case, latency may increase appreciably.
- 2) *Unordered*: frames are delivered at once after duplicates have been removed. Implementation is very simple, but its behavior differs from conventional Wi-Fi.
- 3) *Not-unordered*: out-of-order late frames are discarded (this option is suitable for producer-consumer schemes only). Latency decreases, but the amount of dropped frames grows larger. Optimization is possible also on the sender side: for example, as soon as a new frame F_i is queued, all pending transmissions for earlier frames F_j , $j < i$, can be aborted. As this policy is rather peculiar, it will not be considered in the following.

E. Security and fault tolerance

Seamless redundancy protocols like PRP and *Wi-Red* may suffer from issues related to security aspects. For instance, an attacker may inject frames with wrong sequence numbers and, by exploiting the mechanism that discards duplicates in receivers, force them to throw away all the correct copies of the related packets. The net effect resembles a denial of service (DoS) attack. However, very effective security protocols exist for Wi-Fi, like the Wi-Fi protected access (WPA and WPA2), which are currently included in every commercial device. They use a message integrity check (MIC), which prevents the receiving RSTAs from accepting fake frames forged by attackers. If secure wireless connections are set up in *Wi-Red*, LREs in receivers are certain to operate on authentic messages.

Another aspect that is very important in several application fields—including industrial plants—is fault tolerance [22]. Ensuring fault-tolerant communications among devices in distributed control systems can be achieved by resorting to, e.g., PRP. This requires that two distinct, parallel networks are deployed. If the system comprises Wi-Fi links, the related wireless devices have to be replicated as well, and should operate on different channels as in PoW. A point that needs to be stressed about *Wi-Red* is that, unlike PRP, it is not meant to enhance system reliability against *permanent* faults in devices and cables, but to improve communication quality over IEEE 802.11 wireless links—in terms of packet losses and transmission latencies—in the presence of *temporary* channel

disturbance and interference. *Wi-Red* is noticeably cheaper than PRP, as it does not require full network duplication. All it is needed is upgrading Wi-Fi equipment to *Wi-Red*.

IV. PERFORMANCE EVALUATION

The performance of *Wi-Red* has been first evaluated through network simulation and compared to non-redundant Wi-Fi (i.e., plain DCF). Then, a comparison was made among DA mechanisms. In the following, duplex redundancy is considered since it is the most cost-effective solution for real systems.

A. Communication quality

Two important indexes for communication quality in industrial scenarios are the *packet loss ratio* (fraction of packets that never reach their destination) and the *transmission latency* (time elapsed between the instant when a packet is queued by the transmitting user and the instant it becomes available to the receiving user). Both indexes are measured at the application level: in the following, we will call “*packet*” the piece of information to be exchanged (as seen above the redundancy layer), while “*frame*” refers to what is actually sent on air.

Hard real-time applications require that delays experienced by packets never exceed predefined deadlines. Softer requirements, typical of non-time-critical systems, may refer to statistical properties of latency in *steady-state* conditions. In this case, system behavior is correct as long as constraints are not violated for statistical indexes. For instance, an upper bound can be defined for the *fraction* of packets that are allowed to exceed a given deadline. Equivalently, a specific latency *percentile* can be requested to be lower than such a deadline.

Typical statistical indexes for the transmission latency d include its *average* value (\bar{d}), *standard deviation* (σ_d), *minimum* (d_{min}) and *maximum* (d_{max}) values, as well as percentiles. In all experiments d_{min} was equal to the theoretical transmission time over the air (38 μ s). Conversely, d_{max} values were found to be little significant for our purposes, as they could vary appreciably also in similar experiments. For this reason we took into account the 95, 99, and 99.9 percentiles (d_{p95} , d_{p99} , and $d_{p99.9}$, respectively) in place of d_{max} . Obviously, $d_{p95} \leq d_{p99} \leq d_{p99.9} \leq d_{max}$.

Latency d_i experienced by any given packet can be expressed as the sum of 3 components: the *queuing time* (d_Q) spent in Q^{tx} before the frame transmission begins, the *transmission time* (d_T), which includes possible retransmissions, until the frame is delivered successfully to its destination, and the *reordering time* (d_R) possibly spent in Q^{rx} before the packet is relayed to the upper layers. The same contributions can be found in the corresponding average values.

With respect to packet delivery, the packet loss ratio P_{lost} takes into account packets that have been definitely lost (i.e., packets whose copies were eventually dropped on both channels because of transmission buffer overruns, timeouts in the reordering buffer, or the retry limit being exceeded). $P_{d=d_{min}}$ is the fraction of packets incurring in delays due only to propagation on air and processing by adapters, whereas $P_{d>d_{min}} = 1 - P_{d=d_{min}}$ concerns packets for which $d > d_{min}$ (i.e., packets which have been either lost or further delayed

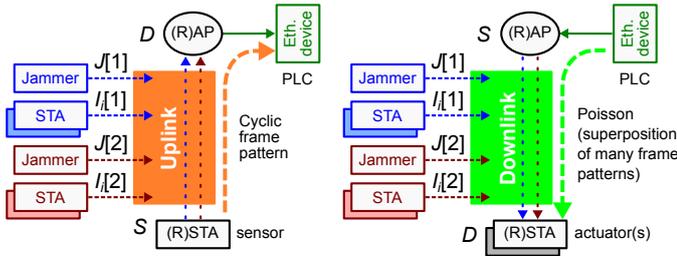


Fig. 3. Structure of the testbed used for simulations (uplink and downlink).

because of buffer queuing, retransmissions, or deferral by the MAC). Similarly, $P_{d>1}$, $P_{d>10}$, and $P_{d>100}$ represent *deadline miss ratios*, and refer to packets whose delivery took more than 1, 10, and 100 ms, respectively. Obviously $P_{d>d_{min}} \geq P_{d>1} \geq P_{d>10} \geq P_{d>100} \geq P_{lost}$.

B. Testbed structure

Simulation experiments were carried out by means of a popular network modeler, and the testbed sketched in Fig. 3 was used to evaluate the communication quality on a single (simplex or duplex) unidirectional wireless *link*. The experimental setup consists of:

- one *source* node (S), which generates unicast packets to be sent on the link according to predefined rules;
- one *destination* node (D), which receives packets and computes statistical indexes of interest;
- a number of *interfering* nodes (I_i), which produce concurrent bulk traffic on wireless channels;
- two *jammer* nodes (J), which inject noise pulses on channels according to the Gilbert-Elliot model.

In case of simplex DCF links, S and D (as well as interferers) were based on the conventional IEEE 802.11 simulator models. The same is true for *basic Wi-Red* (practically similar to PoW), whereas RSTA models were purposely developed for S and D to deal with DA mechanisms in *Wi-Red*.

DCF enables a set of STAs in an IBSS to share the network bandwidth fairly, but the most common setup in real applications are by far infrastructure networks. APs act as *hubs*, which collect and relay all frames among nodes (both wireless and wired). Thus, the amount of traffic sent by the AP is noticeably higher than for ordinary STAs. When the BSS experiences temporary congestion, the AP may easily become the network bottleneck. This is a well-known problem [21], and many improvements to basic DCF (i.e., transmission opportunities and frame aggregation [6]) were primarily conceived bearing APs in mind. From this point of view *Wi-Red* is no exception: the RAP is the node where frames may likely suffer from longer latencies and even get lost because of buffer overflows.

To cope with the RAP *asymmetric* behavior, two different situations were considered in the experiments, dealing with transmissions from an RSTA to the RAP (*uplink*) and viceversa (*downlink*).

1) *Uplink transmissions*: in this case experiments somehow resemble those presented in [19], with S and D acting as a generic RSTA and the RAP, respectively. In a typical scenario, S models a *sensor* that sends its measured data

to a *controller* (e.g., a PLC connected to a wired segment) through the RAP. The packet generation law is predictable and was selected to be cyclical. Since modern APs are frequently equipped with Gigabit Ethernet interfaces, outgoing frames from the RAP can be safely assumed not to be queued. Moreover, transmission times over wires can also be neglected. In theory, if the packet *arrival rate* λ on the $S \rightarrow D$ link is kept below the available network capacity, cyclic generation does not cause frame queuing. Practically, interfering traffic and disturbance have to reach quite high levels to affect the link communication quality in a tangible way. As shown in the following, transmission latency is due, for a non-negligible part, to the frame reordering procedure carried out by the RAP.

2) *Downlink transmissions*: In this network setup S acts as the RAP whereas D models *multiple* RSTAs (e.g., a set of remote *actuators*). It is worth noting that this is possible since all downlinks originating from S share the same DCF protocol automaton, and each link is granted the same average communication quality. The packet generation law for S consists of the superposition of several non-synchronized cyclical streams with different transmission periods. To produce general results as much as possible, this was approximated by a Poisson process with arrival rate λ , so that the whole $S \rightarrow D$ traffic can be characterized by means of a single parameter.

Differently from the uplink experiments, frame reordering was not taken into account. Actually, reordering is carried out separately for each downlink by the relevant target RSTA. If the number of RSTAs modeled by D is large enough, the order of the frames sent to any specific device is unlikely affected by DA mechanisms in a significant way. In practice, the contribution of reordering to latency is much smaller than in the uplink case, and hence it can be safely neglected.

C. Testbed configuration

In each experiment S was configured to send a large number of packets addressed to D , with a payload size equal to

TABLE II
SIMULATION PARAMETERS FOR THE SOURCE TRAFFIC ($S \rightarrow D$ LINK).

Parameter	$c(1)$	$e(1)$	$e(0.5)$
Direction	<i>uplink</i>	<i>downlink</i>	<i>downlink</i>
Packet payload size	50 B	50 B	50 B
Packet generation law	cyclic	exponential	exponential
Mean generation time ($\bar{T} = 1/\lambda$)	1 ms	1 ms	0.5 ms

TABLE III
SIMULATION PARAMETERS FOR THE SURROUNDING ENVIRONMENT.

Parameter	<i>Benign</i> (E_B)	<i>Hostile</i> (E_H)
Frames per burst	700	700
Frame payload size	1500 B	1500 B
Frame generation period	500 μ s	500 μ s
Gap between bursts	Exp(1 s)	Exp(1 s)
Number of interferers	2	4
Overall interfering traffic (Δ)	$\geq 36\%$	$\geq 72\%$
Step duration	1 μ s	1 μ s
Trans. prob. good \rightarrow bad (p_{gb})	$1.74 \cdot 10^{-4}$	$1.74 \cdot 10^{-4}$
Trans. prob. bad \rightarrow good (p_{bg})	$1.74 \cdot 10^{-2}$	$1.74 \cdot 10^{-3}$
Bit error prob. good state (p_g)	0	0
Bit error prob. bad state (p_b)	$7.5 \cdot 10^{-2}$	$7.5 \cdot 10^{-2}$

TABLE IV
 RESULTS OF THE STEADY-STATE ANALYSIS WITH DISTURBANCE AND INTERFERENCE (Wi-Fi, *basic Wi-Red*, AND RDA MECHANISMS).

Environment and $S \rightarrow D$ traffic		Redund. approach	Statistics about packet transmission latency [ms]						Percentages about transmission latencies and failures [%o]					Mean queue size		
			\bar{d}	σ_d	d_{p95}	d_{p99}	$d_{p99.9}$	d_{max}	$P_{d>d_{min}}$	$P_{d>1}$	$P_{d>10}$	$P_{d>100}$	P_{lost}	$\bar{q}[1]$	$\bar{q}[2]$	
Benign environment E_B	Uplink	$c(1)$	DCF	4.34	13.78	31.9	72.3	116.6	218	475	263	93.6	2.38	0.0001	4.62	-
			PoW	0.482	2.93	1.391	8.50	45.9	102.6	226	69.6	9.03	0.0013	0	4.67	4.60
			RDA/Q	0.141	0.394	0.649	1.575	4.69	31.9	202	24.1	0.1477	0.0	0	0.53	0.53
			RDA/R	0.113	0.237	0.524	1.114	2.41	20.3	194	13.22	0.0045	0.0	0	0.48	0.48
	Downlink	$e(1)$	DCF	5.16	14.89	34.0	78.8	128.6	271.3	636	355	105.4	3.64	0.0002	5.72	-
			PoW	0.806	3.43	3.41	12.14	51.1	113.9	451	136.4	12.49	0.0092	0	5.84	5.75
			RDA/Q	0.261	0.658	1.119	2.85	7.45	39.9	421	58.2	0.473	0.0	0	0.79	0.79
			RDA/R	0.212	0.413	0.909	1.986	4.13	21.3	416	42.6	0.025	0.0	0	0.72	0.72
Downlink	$e(0.5)$	DCF	105.3	115.6	342	466	589	750	894	717	670	408	21.73	197.0	-	
		PoW	46.6	69.2	192.2	293	427	713	812	543	473	183.2	0.622	194.9	196.4	
		RDA/Q	2.84	9.33	16.41	49.2	98.4	199.0	658	215	75.4	0.921	0	6.11	6.08	
		RDA/R	1.107	3.72	5.08	17.99	46.7	99.2	647	167.4	23.4	0.0	0	2.72	2.71	
Hostile environment E_H	Uplink	$c(1)$	DCF	54.7	88.3	237	396	624	894	788	664	478	205	0.1803	54.71	-
			PoW	15.53	36.6	93.4	173.0	295	587	626	446	232	44.0	0	55.65	55.52
			RDA/Q	1.126	4.39	4.31	20.0	58.6	145.8	520	177.3	23.2	0.1020	0	1.38	1.38
			RDA/R	0.435	1.091	1.721	4.54	13.37	53.2	500	109.8	2.06	0.0	0	0.69	0.69
	Downlink	$e(1)$	DCF	57.5	89.8	241	404	613	1203	856	723	509	216	0.208	57.97	-
			PoW	16.96	37.9	97.2	180.2	306	644	738	523	258	47.4	0	58.21	58.35
			RDA/Q	1.642	5.27	6.85	25.7	64.9	166.0	656	269	34.0	0.1893	0	2.15	2.14
			RDA/R	0.759	1.658	3.07	7.47	18.97	82.2	644	210	5.32	0.0	0	1.25	1.25
Downlink	$e(0.5)$	DCF	338	204	697	834	976	1242	999	735	724	662	245	469.6	-	
		PoW	263	180.7	600	752	911	1320	998	893	871	753	67.2	467.8	467.7	
		RDA/Q	48.9	71.7	201	312	429	713	987	671	529	184.9	1.845	95.61	95.48	
		RDA/R	21.9	39.5	107.1	182	276	407	985	585	369	58.1	0.0421	43.53	43.46	

50 B. The period between subsequent packets was set to the constant value 1 ms in uplink simulations, whereas packet inter-arrival times were exponentially distributed for downlink experiments and two sub-cases were considered, with expected value $\bar{T} = 1/\lambda$ equal to 0.5 ms and 1 ms, respectively. Overall, three kinds of $S \rightarrow D$ traffic were taken into account, by varying the generation law—either *cyclic* (c) or *exponential* (e)—and mean intertime (\bar{T}), as reported in Table II. In the following they are denoted $c(1)$, $e(1)$, and $e(0.5)$. Instead, two different conditions were considered for interference and disturbance of the surrounding environment, that is *benign* (E_B) and *hostile* (E_H). Parameters of interest are listed in Table III.

Interfering traffic consists of a number of STAs that repeatedly generate bursts of 700 frames. For each burst, frames with a 1500 B payload are sent every 500 μ s. Unlike [18] and [19], acknowledged transmissions were used, as they more faithfully model concurrent TCP/IP traffic over the air. The inter-burst gap is an exponentially distributed random variable with mean value 1 s. Traffic produced on BSS $[k]$ by each interfering node $I_i[k]$ (excluding retransmissions) is equal to $700 \cdot (T_{TX} + T_{SIFS} + T_{ACK} + T_{DIFS}) / (700 \cdot 500 \mu\text{s} + 1 \text{s}) \simeq 18\%$ of the available channel bandwidth, where T_{TX} and T_{ACK} are the transmission times of interfering and ACK frames (254 μ s and 34 μ s, respectively), while T_{SIFS} and T_{DIFS} are the short and distributed interframe space durations (10 μ s and 50 μ s, respectively, as nodes in the testbed were set for simplicity as part of an IBSS). We used two interfering nodes for each channel in E_B and four nodes in E_H . In this way, the overall interfering traffic Λ on each BSS was $\geq 36\%$ and $\geq 72\%$, respectively.

Disturbance was modeled independently on each channel according to the Gilbert-Elliott model [23], [24]. Indeed, other models exist [25], which describe the wireless spectrum more accurately when the environment is known to some

extent. However, the main purpose of this simulation campaign was comparing approaches based on seamless redundancy (including RDA and PDA techniques) with plain Wi-Fi in the same operating conditions. A preliminary measurement campaign was run on a prototype PoW setup made up of real devices, deployed in both a lab environment (high interfering traffic) and an industrial plant (with high-current welding guns). Results showed that the advantages of *Wi-Red* over Wi-Fi mainly depend on whether channels are reasonably independent (which is the case of most real-world systems), while the specific nature of the surrounding environment plays a marginal role. For these reasons, and in order to obtain generally applicable results, we opted for the simpler Gilbert-Elliott disturbance model.

As in [26], 5 parameters are needed to this purpose, that is the transition probability from good to bad state (p_{gb}) and viceversa (p_{bg}), the bit error probability for good (p_g) and bad state (p_b), and the step duration (1 μ s). Gilbert-Elliott parameters for E_B are similar to those reported in [26] and resemble a *typical* real-world Wi-Fi channel ($p_{gb} = 1.74 \cdot 10^{-4}$, $p_{bg} = 1.74 \cdot 10^{-2}$, $p_g = 0$, and $p_b = 7.5 \cdot 10^{-2}$). To stress the proposed DA techniques, disturbance in E_H was set about one order of magnitude higher than E_B (p_{bg} was decreased to $1.74 \cdot 10^{-3}$). In the simulator, disturbance was achieved using one jammer per channel ($J[1]$ and $J[2]$).

Much more cases were actually analyzed in our simulation campaign, besides those described above, by considering alternate sets of parameters. Only the most significant ones are presented in the following for space reasons.

V. RESULTS

RDA mechanisms were evaluated independently and compared to both plain DCF and *basic Wi-Red* (PoW). Then, a specific PDA technique (namely DDD) was considered and the impact of the deferral threshold value on latencies analyzed.

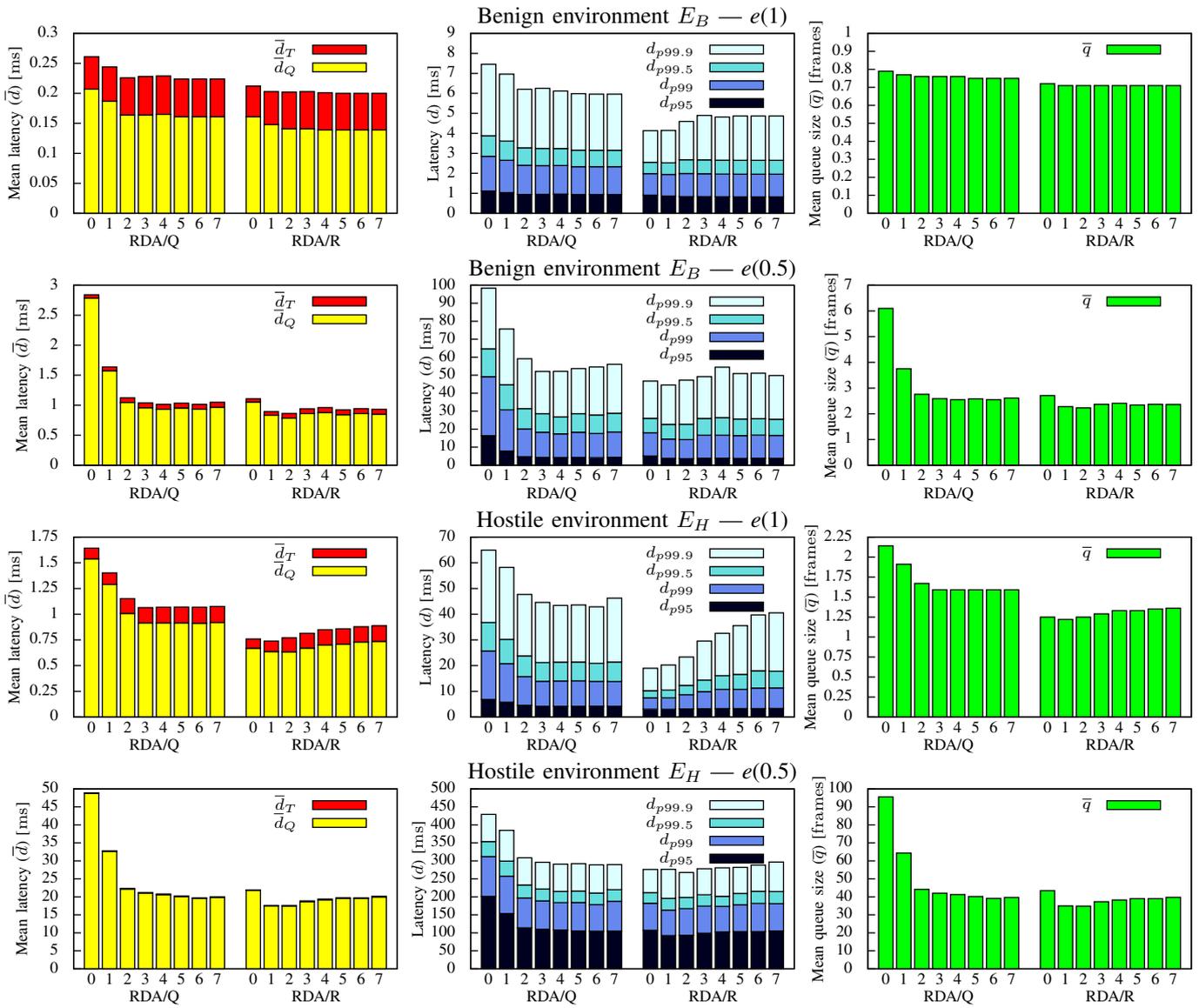


Fig. 4. Results for *Wi-Red* with PDA (DDD): downlink case (Poisson $S \rightarrow D$ traffic), RDA/R and RDA/Q (with and without ADCF).

A. RDA mechanisms

Table IV summarizes the statistical performance indexes for legacy (non-redundant) DCF, PoW (which, over a single redundant link, behaves the same as *basic Wi-Red*, without any DA), and *Wi-Red* (with either RDA/Q or RDA/R).

1) *Uplink transmissions*: In this case, the $S \rightarrow D$ generation law is $c(1)$. In benign environmental conditions (E_B) *basic Wi-Red* provided a tenfold reduction of the mean transmission latency \bar{d} over DCF, and the adoption of RDA improved its behavior further. In hostile conditions (E_H) *basic Wi-Red* was no longer able to grant such a huge improvement, whereas RDA boosted performance noticeably again. The reason is that, in hostile environments RDA prevents queues from growing out of control. The analysis of higher percentiles ($d_{p99.9}$) shows that RDA always outperformed both DCF and *basic Wi-Red*.

This means that very few packets were excessively delayed in the experiments.

By comparing RDA/Q and RDA/R, it is clear that the ability of the latter to abort transmissions at the MAC level brings significant advantages with respect to the simple removal of frames in Q^{tx} carried out by the former, in particular when the environment is hostile. This is due to the decrease in the overall network load when useless retransmissions are aborted.

2) *Downlink transmissions*: Two different values were used for the $S \rightarrow D$ generation rate λ , so that in $e(0.5)$ it was twice than in $e(1)$. Since RDA is aimed at preventing duplicate transmissions, the available network bandwidth virtually doubles with respect to *basic Wi-Red*. A comparison of the mean transmission latency \bar{d} for PoW in $e(1)$ and RDA/R in $e(0.5)$ (i.e., 0.806 ms vs. 1.107 ms in E_B and 16.96 ms vs. 21.9 ms in E_H) shows that RDA was able to improve the link throughput

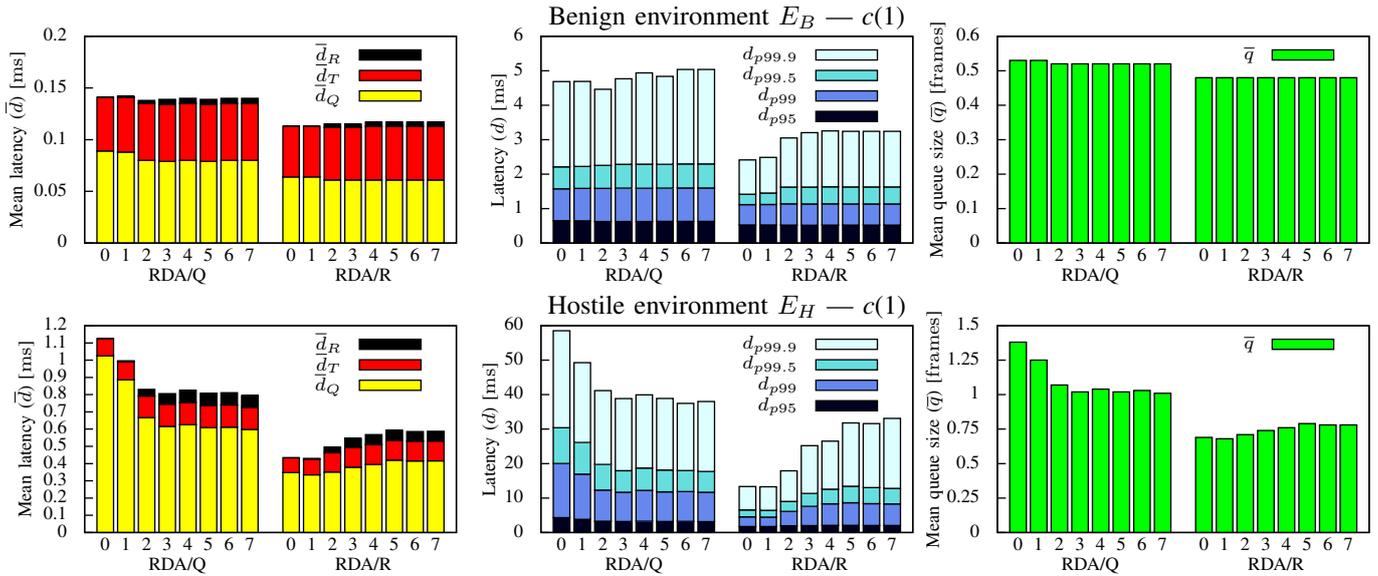


Fig. 5. Results for *Wi-Red* with PDA (DDD): *uplink* case (*cyclic S→D* traffic), RDA/R and RDA/Q (with and without ADFC).

by a factor close to two with respect to *basic Wi-Red*.

Poisson arrivals with rate λ close to the link throughput, as in $e(0.5)$, may cause Q^{tx} to fill up, and this behavior is even more evident in hostile environment (E_H). Consequently, both latency and the packet loss ratio increase. Experiments confirmed that RDA, and in particular RDA/R, can provide substantial benefits to the fraction of dropped packets. For instance, in the worst conditions—i.e., $e(0.5)$ and E_H —more than 24% of packets were lost in conventional Wi-Fi (DCF); basic redundant schemes like PoW reduced this fraction to less than 7%, while the adoption of DA mechanisms shrank it down to 0.18% (RDA/Q) and 0.004% (RDA/R).

B. PDA mechanisms

Figures 4 and 5 show the effect of adopting PDA along with RDA in *downlink* and *uplink* transmissions, respectively. DDD was explicitly considered and the threshold value D_{th} varied in the range from 0 (DDD disabled) to 7 (a duplicate is sent only after the first copy of a frame has been discarded by the MAC on the other channel).

TABLE V
 PACKET LOSS RATIO, UNORDERED (U) AND ORDERED (O) POLICIES.

PDA/ D_{th}	Packet Loss Ratio (P_{lost}) [$\cdot 10^{-6}$]							
	E_B				E_H			
	RDA/Q		RDA/R		RDA/Q		RDA/R	
	U	O	U	O	U	O	U	O
DDD/0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DDD/1	0.0	0.1	0.0	0.0	0.0	1.3	0.0	1.5
DDD/2	0.0	0.1	0.0	0.3	0.0	14.4	0.0	13.8
DDD/3	0.0	1.4	0.0	0.7	0.0	131.5	0.0	118.9
DDD/4	0.0	7.2	0.0	4.9	0.0	331.7	0.0	269.2
DDD/5	0.0	8.5	0.0	5.7	0.0	345.2	0.0	287.8
DDD/6	0.0	10.3	0.0	5.7	0.0	352.3	0.0	280.8
DDD/7	0.0	10.3	0.0	5.7	0.0	335.8	0.0	269.1

Three groups of diagrams are shown for each experiment. The leftmost vertical group in each figure concerns the mean latency \bar{d} : each bar highlights the different contributions due to queuing (\bar{d}_Q), transmission (\bar{d}_T) and, for $c(1)$ only, reordering (\bar{d}_R). The second group concerns percentiles, shown in terms of difference with respect to the predecessor. Finally, the rightmost group depicts the mean queue size \bar{q} (in redundant schemes, $\bar{q} = (\bar{q}[1] + \bar{q}[2])/2$). Two sets of bars are included in each diagram, which concern RDA/Q (on the left) and RDA/R (on the right), respectively.

1) *Downlink transmissions*: As can be seen in Fig. 4, the mean latency \bar{d} mainly depended on the time d_Q spent by frames in the transmission buffer Q^{tx} . As a consequence, any DA mechanism that helps reducing the number of queued frames improves the overall link quality. In all diagrams in Fig. 4, RDA/R always performs better than RDA/Q. The cost to be paid is that a modified MAC is needed, able to support some kind of ADFC.

The RDA/Q behavior can be effectively improved by selecting higher values for D_{th} , in particular when λ approaches the mean channel throughput. In fact, increasing D_{th} reduces the amount of duplicates sent on air. An optimal D_{th} value in this case is around 3. The adoption of DDD with RDA/R introduced slight improvements when λ was high, as in $e(0.5)$, but only when $D_{th} = 1$. This is because, with duplex redundancy, such a choice enables scattering frame transmissions on the two channels. When $D_{th} > 1$ the retry counter and backoff duration are allowed to grow, and this worsens latency.

2) *Uplink transmissions*: Similar considerations as above apply to results shown in Fig. 5, where the number of queued packets is lower on average because packet generation is cyclic. Consequently, benefits introduced by DDD are not as significant as in the case of Poisson arrivals. As can be seen in the leftmost graphs, latency is due in part to reordering in the reception buffer, especially in hostile environment (E_H),

where the order of the received frames may be altered tangibly because of redundant transmission.

3) *Packet loss rate*: Timeliness is useless if too many packets are dropped. Table V reports P_{lost} values for $c(1)$ experiments (uplink). The first row of the table shows that no packets were lost when RDA mechanisms were used alone ($D_{th} = 0$ means that DDD was disabled). The same held when the *unordered* delivery policy was adopted. Conversely, packets were dropped for higher D_{th} values in the case of *ordered* delivery. This occurs because of timeouts in the reordering block, which prevent packets from being stuck in Q^{rx} for more than 10 ms.

VI. CONCLUSIONS

A fundamental requirement in industrial applications is determinism. Consequently, underlying communication systems must grant that all packets are delivered timely and correctly. Less demanding real-time control systems also exist, where the above property has to hold for the vast majority of packets. Unfortunately, wireless networks—and in particular Wi-Fi, which is considered the wireless counterpart of Ethernet—are not reliable enough, not even for the latter kind of applications.

As the wireless transmission medium is not under complete control of system designers and managers, interference and disturbance can not be avoided. For this reason, providing some “quasi-deterministic” overlay—implemented, e.g., as a software protocol layer on top of Wi-Fi—is often necessary, but usually not sufficient. Seamless redundancy permits to face peculiarities of radio transmission and improves the wireless channel reliability. Besides plain use of PRP and redundant Wi-Fi equipment, the adoption of duplicate avoidance mechanisms, which prevent the bandwidth from being wasted, improves the system behavior further.

In this paper, a unified framework has been provided to describe and categorize the most recent proposals that apply seamless link-level redundancy to Wi-Fi. A thorough comparison of the available solutions has been carried out through simulation. Results are encouraging, and show that communication quality can be largely improved—up to orders of magnitude—by introducing small changes in existing wireless devices (no alteration is needed in the physical layer and only minor modifications are requested to the MAC). It is worth remarking that, at present, a large number of Wi-Fi devices are dual band, and many APs support simultaneous dual-band operations as well. This means, that most hardware resources needed to support seamless redundancy are already there.

While mainly targeted at industrial, building, and home automation systems, *Wi-Red* is very appealing also in other application scenarios, such as high-fidelity multimedia streaming. For this reason, we believe that it should be incorporated in future versions of IEEE 802.11 standards.

REFERENCES

[1] G. Hancke, V. Gungor, and G. Hancke, “Guest Editorial Special Section on Industrial Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 762–765, Feb 2014.

[2] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. Johansson, “System Architectures, Protocols and Algorithms for Aperiodic Wireless Control Systems,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, Feb 2014.

[3] S. Petersen and S. Carlsen, “WirelessHART Versus ISA100.11a: The Format War Hits the Factory Floor,” *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, Dec 2011.

[4] “IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs),” *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, Sept 2011.

[5] F. Rezha and S. Y. Shin, “Performance Analysis of ISA100.11a Under Interference From an IEEE 802.11b Wireless Network,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 919–927, May 2014.

[6] “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 - Redline,” *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007) - Redline*, pp. 1–5229, March 2012.

[7] S. Vitturi, F. Tramarin, and L. Seno, “Industrial Wireless Networks: The Significance of Timeliness in Communication Systems,” *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 40–51, June 2013.

[8] S. Vitturi, L. Seno, F. Tramarin, and M. Bertocco, “On the Rate Adaptation Techniques of IEEE 802.11 Networks for Industrial Applications,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 198–208, Feb 2013.

[9] N. Marchenko, T. Andre, G. Brandner, W. Masood, and C. Bettstetter, “An Experimental Study of Selective Cooperative Relaying in Industrial Wireless Sensor Networks,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1806–1816, Aug 2014.

[10] G. Prytz, “Network recovery time measurements of RSTP in an ethernet ring topology,” in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2007, pp. 1247–1253.

[11] IEC, “Industrial communication networks - High availability automation networks - Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR),” *IEC Std 62439-3 ed 2.0*, pp. 1–177, 2010.

[12] M. Rentschler and P. Laukemann, “Towards a reliable parallel redundant WLAN black channel,” in *IEEE International Workshop on Factory Communication Systems (WFCS)*, 2012, pp. 255–264.

[13] R. Moraes, F. Vasques, P. Portugal, and J. Fonseca, “VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks,” *IEEE Transactions on Industrial Informatics*, vol. 3, no. 3, pp. 215–224, 2007.

[14] R. Costa, P. Portugal, F. Vasques, R. Moraes, and R. Custodio, “A coordination layer to handle real-time communication in Wi-Fi networks with uncontrolled traffic sources,” in *IEEE Conference on Local Computer Networks (LCN)*, 2011, pp. 263–266.

[15] 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 *IEEE International Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*, Sept 2011, pp. 81–86.

[16] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, “Implementation and Evaluation of the Reference Broadcast Infrastructure Synchronization Protocol,” *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 801–811, June 2015.

[17] D.-K. Dang, A. Mifdaoui, and T. Gayraud, “Design and analysis of UWB-based network for reliable and timely communications in safety-critical avionics,” in *IEEE Workshop on Factory Communication Systems (WFCS)*, May 2014, pp. 1–10.

[18] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, “An Enhanced MAC to Increase Reliability in Redundant Wi-Fi Networks,” in *IEEE International Workshop on Factory Communication Systems (WFCS)*, 2014, pp. 1–10.

[19] —, “Dynamic duplicate deferral techniques for redundant Wi-Fi networks,” in *IEEE Conference on Emerging Technology and Factory Automation (ETFA)*, Sept 2014, pp. 1–8.

[20] J. Bai, E. Eyisi, Y. Xue, and X. Koutsoukos, “Dynamic Tuning Retransmission Limit of IEEE 802.11 MAC Protocol for Networked Control Systems,” in *IEEE/ACM Int. Conference on Cyber, Physical and Social Computing (CPSCom)*, 2010, pp. 666–671.

[21] 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 Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 425–437, 2010.

- [22] Z. Gao, S. Ding, and C. Cecati, "Real-time fault diagnosis and fault-tolerant control," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3752–3756, June 2015.
- [23] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.
- [24] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.
- [25] F. Heereman, W. Joseph, E. Tanghe, L. Verloock, and L. Martens, "Performance Degradation Due to Multipath Noise for Narrowband OFDM Systems: Channel-Based Analysis and Experimental Determination," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1396–1405, March 2015.
- [26] A. Willig, M. Kubisch, and A. Wolisz, "Measurements and stochastic modeling of a wireless link in an industrial environment," *Telecommunication Networks Group, Tech. Univ. Berlin*, 2001, [Online] Available: <http://timmurphy.org/2009/07/22/line-spacing-in-latex-documents/>.



Adriano Valenzano (SM'09) received the Laurea degree in electronic engineering from Politecnico di Torino, Torino, Italy, in 1980. He is Director of Research with the National Research Council of Italy (CNR). He is currently with the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), Torino, Italy, where he is responsible for research concerning distributed computer systems, local area networks, and communication protocols. He has coauthored approximately 200 refereed journal and conference papers in the area

of computer engineering.

Dr. Valenzano is the recipient of the 2013 IEEE IES and ABB Lifetime Contribution to Factory Automation Award. He also received, as a coauthor, the Best Paper Award presented at the Fifth and Eighth IEEE Workshops on Factory Communication Systems (WFCS 2004 and WFCS 2010).

He has served as a technical referee for several international journals and conferences, also taking part in the program committees of international events of primary importance. Since 2007, he has been serving as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.



Gianluca Cena (SM'09) received the Laurea degree in electronic engineering and the Ph.D. degree in information and system engineering from the Politecnico di Torino, Turin, Italy, in 1991 and 1996, respectively.

In 1995, he became an Assistant Professor in the Department of Computer Engineering, Politecnico di Torino. Since 2005 he has been a Director of Research with the National Research Council of Italy (CNR), and in particular with the Institute of Electronics, Computer, and Telecommunication

Engineering (IEIIT), Turin. His research interests include industrial communication systems and real-time networks. In these areas, he has coauthored more than 100 technical papers.

Dr. Cena served as Program Co-Chairman for the 2006 and 2008 editions of the IEEE Workshop on Factory Communication Systems, and has been an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS since 2009.



Stefano Scanzio (S'06-M'12) received the Laurea and Ph.D. degrees in computer science from Politecnico di Torino, Torino, Italy, in 2004 and 2008, respectively.

He was with the Department of Computer Engineering, Politecnico di Torino, from 2004 to 2009, where he was involved in research on speech recognition and, in particular, he has been active in classification methods and algorithms. Since 2009, he has been with the National Research Council of Italy (CNR), where he is a tenured technical Researcher with the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT), Turin.

Dr. Scanzio teaches several courses on computer science at Politecnico di Torino. He has authored and co-authored several papers in international journals and conferences in the area of industrial communication systems and real-time networks.

Dr. Scanzio teaches several courses on computer science at Politecnico di Torino. He has authored and co-authored several papers in international journals and conferences in the area of industrial communication systems and real-time networks.