Bandwidth Management for Soft Real-Time Control Applications in Industrial Wireless Networks

Lucia Seno, *Member IEEE*, Gianluca Cena, *Senior Member IEEE*, Adriano Valenzano, *Senior Member IEEE* and Claudio Zunino
CNR-IEIIT, c.so Duca degli Abruzzi 24, I-10129 Torino, Italy
{lucia.seno, gianluca.cena, adriano.valenzano, claudio.zunino}@ieiit.cnr.it

L. Seno, G. Cena, A. Valenzano, and C. Zunino, "Bandwidth Management for Soft Real-Time Control Applications in Industrial Wireless Networks", *IEEE Transactions on Industrial Informatics*, Vol. 13, No. 5 (October 2017), pp. 2484-2495, IEEE Industrial Electronics Society (USA). DOI: 10.1109/TII.2017.2720638

available at https://doi.org/10.1109/TII.2017.2720638.

Bandwidth Management for Soft Real-time Control Applications in Industrial Wireless Networks

Lucia Seno, *Member IEEE*, Gianluca Cena, *Senior Member IEEE*, Adriano Valenzano, *Senior Member IEEE*, and Claudio Zunino

***Abstract*—Industrial distributed control systems would greatly benefit from the adoption of wireless communication technologies, if only guarantees could be provided on timing of time-critical data delivery over the ether. This paper presents solutions to handle single-hop deadline-constrained periodic traffic, which combine centralized transmission scheduling according to EDF and ARQ. For each solution, an admission control test is provided, which guarantees a configurable number of retries to each data instance within its deadline, statically addressing both timeliness and reliability on a per-instance basis. Dynamic bandwidth management strategies are also introduced that use runtime information about unperformed guaranteed retries and reassign them, as extra retries, to failed instances whose deadline has not yet expired. Simulation results show that significant benefits can be obtained, in terms of both determinism and improved performance, by a careful static/dynamic management of the available communication resources.**

***Index Terms*—Industrial wireless networks, traffic scheduling, EDF, retransmission strategies, soft real-time control systems.**

I. INTRODUCTION

Nowadays, wireless communication technologies are pervading a large number of application domains and have also become very attractive for industrial distributed control systems (IDCS), which are at the basis of cyber-physical production systems [1]–[3]. Unfortunately, real-time requirements on data delivery typical of IDCS often prevent the adoption of wireless networks, intrinsically characterized by non-deterministic behavior [4], as communication infrastructure. This is surely true for hard real-time applications [5], where deadlines on data delivery shall not be exceeded and any delay or loss may have severe consequences. IDCS characterized by soft real-time requirements [5], however, are becoming popular, thanks to the advances in networked control theory [6] which have increased robustness of control algorithms with respect to non-idealities (i.e., delays and losses) introduced by real networks. Such applications can tolerate deadline misses, as long as their impact is a priori guaranteed to be below some *functional threshold*, even though this may affect, to some extent, control quality and system accuracy. Often, the functional threshold is expressed in probabilistic terms as a lower bound on the likelihood of correctly delivering single data instances within their deadline. Indeed, as control quality also depends on the distribution of missed deadlines over time, per-instance requirements, unlike aggregate ones (e.g., on average), decrease the likelihood of failures concentrating in short time intervals. In these applications, a priori guaranteeing that functional requirements are satisfied is mandatory, while any performance improvement beyond the functional level is highly desirable, as it reflects in a more accurate system behavior.

This paper targets soft-real time applications and focuses on two issues that have to be jointly addressed to satisfy functional requirements. On one hand, *timeliness* (i.e., performing time-critical data transmissions within deadlines), which in case of multiple time-critical data has to consider *individual* deadlines but also *concurrency* among data (originating at either one or a plurality of nodes) for the shared medium. On the other, *reliability* (i.e., even if transmissions are performed on time, time-critical data may still miss their deadlines due to transmission errors), which is particularly relevant in wireless networks given the error-prone nature of the ether.

Timeliness is usually dealt with at the medium access control (MAC) sublayer, where deadline-aware mechanisms can be integrated with those allowing nodes to share the transmission support. Available MAC solutions can be roughly divided in *exclusive assignment* and *random access* protocols. Exclusive assignment protocols, e.g., those based on time, frequency, code, or space division multiple access (TDMA, FDMA, CDMA, SDMA), prevent collisions by a priori dividing the available communication resources among nodes, so that each can exclusively access its share. Conversely, with random access protocols, e.g., carrier sense multiple access (CSMA), medium access is uncoordinated and random elements (e.g., backoff) are used to decrease collision likelihood. Exclusive assignment protocols are the best choice when pursuing determinism, as resources can be statically allocated so as to meet both individual and concurrency requirements. This way, bounded MAC delays and no intra-network collisions are ensured at the expenses of higher coordination overheads and, in case of variable traffic characteristics, of scalability. When dealing with time-critical applications, traffic is typically known in advance. However, if error correction (e.g., Automatic Repeat reQuest, ARQ) is adopted, retransmissions are determined, at runtime, by channel errors, introducing variability in the traffic pattern.

Among exclusive assignment protocols, TDMA is the most widespread as it does not require ad-hoc hardware, nor complex signal processing. In TDMA-based protocols, time is typically divided in cycles of fixed duration, each comprising a number of time slots assigned to nodes for transmission. Examples are guaranteed time slots (GTS) in IEEE 802.15.4-

---

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: {lucia.seno, gianluca.cena, adriano.valenzano, claudio.zunino}@ieiit.cnr.it.

based networks (e.g., ZigBee), and WirelessHART. Solutions have also been proposed to enhance determinism of standard CSMA protocols by implementing TDMA on top of them, e.g., in [7] this approach is applied to the IEEE 802.11 (Wi-Fi) Distributed Coordination Function (DCF). All these examples mimic cyclic scheduling, whose flexibility is rather poor as any variation in the traffic characteristics may require to change the schedule entirely or, at best, to tolerate suboptimal bandwidth usage (e.g., oversizing slots to accommodate retries may cause unnecessarily high MAC delays). Moreover, cyclic scheduling cannot easily handle asynchronous traffic or overload conditions [5].

Better ability to adapt to varying traffic and channel conditions is provided by online resource reservation mechanisms [8] and, among them, by MAC protocols that rely on transmission scheduling according to the policies devised for operating systems (OS), where individual and concurrency aspects of timeliness are statically addressed through feasibility analysis (*admission control*) [9], [10]. Unlike wired networks like CAN, efficient distributed mechanisms for enforcement of scheduling policies are not available in wireless networks, and centralized polling is often employed to let the nodes know when they shall transmit, causing higher overhead.

Concerning reliability, techniques exist exploiting diversity (in time, frequency, or space), which increase the likelihood of successful data delivery on either *i*) a *single* transmission, e.g., Forward Error Correction (FEC) and robust modulation schemes, or *ii*) *multiple* (sequential or simultaneous) transmissions, like ARQ, cooperative relaying [11], and spatial (multiple-input and multiple-output, MIMO) or frequency (seamless) redundancy [12]. Real wireless technologies customarily rely on a combination of (part of) these techniques. Importantly, most approaches trade timeliness for reliability [3]. Often, runtime information on channel conditions is used to achieve optimal trade-offs: e.g., rate adaptation (RA) mechanisms rely on signal-to-noise ratio (SNR) estimates to select, for each data transmission, the highest bit rate providing adequate probability of successful delivery [13].

In this paper, we investigate solutions for supporting periodic deadline-constrained traffic in single-hop wireless networks, combining centralized transmission scheduling according to the Earliest Deadline First (EDF) policy and a truncated stop-and-wait ARQ scheme. The aim is to provide a general approach, which can be applied to different wireless technologies, tackling both timeliness and reliability on a per-instance basis. The hypotheses our work is based on fit in with factory automation cells, which typically consist of sensors and actuators distributed over a plant portion, and one or more controllers implementing monitoring/supervising/control applications. Within a cell, data must be periodically sent from the field to the controllers and vice-versa, as well as among controllers, with timings dictated by the dynamics of the system under control. Unexpected delays or losses, e.g., in sensor samples, force controllers to rely on obsolete values, affecting plant accuracy. Multi-hop networks are mainly used to cover much larger areas and when timing constraints are less strict, so they will not be considered here.

In the following, admission control tests are provided to guarantee each data instance a (configurable) number of *planned* retries within its deadline, under two distinct retransmission strategies. Including retransmissions in feasibility analysis (*static* bandwidth management) allows to jointly address individual and concurrency aspects of timeliness, as well as reliability. As a downside, it leads to pessimistic decisions, as data are admitted only when the amount of bandwidth for all planned retries can be statically allocated, even if at runtime many will reveal unnecessary. To compensate pessimism, *dynamic* bandwidth management strategies are also introduced that exploit runtime information about unused preallocated bandwidth and reassign it to failed data instances without compromising planned retries. Preliminary results obtained with this approach have been presented in [14], where a simple dynamic bandwidth management strategy was introduced and evaluated for a specific retransmission strategy, while in [15] an experimental campaign was carried out on a prototype running on Linux PCs and based on Wi-Fi, also exploiting seamless channel redundancy. Issues related to the software implementation on commercial devices were discussed in [16].

This paper provides a comprehensive theoretical overview of the approach, examining combinations of different strategies for dealing with static and dynamic bandwidth management, and evaluating performance via simulation. The paper is organized as follows: Sect. II discusses relevant works appeared in the literature, Sect. III recaps the basics of the approach, the admission control test for error-free channel, and changes required to cope with retransmissions in real channel conditions. Sect. IV presents two dynamic bandwidth management strategies, Sect. V comments on simulation results, and, finally, Sect. VI draws some conclusions.

II. RELATED WORKS

Several papers exist in the literature that deal with mechanisms for supporting deadline-constrained traffic over wireless networks. Some proposals combine ARQ schemes and techniques for improving reliability of single retries to increase the likelihood of successful data delivery within an intended deadline. Among them, [17], [18] deal with Deadline-Dependent Coding (DDC) and combine ARQ and FEC. In [17] RS-coded data failing a number of retries and a voting procedure are retransmitted using a higher-redundancy RS-code, while in [18] concatenated codes and iterative decoding are used at each retry. Results are provided for error probability vs. transmission time, allowing designers to find the appropriate trade-off between reliability and timeliness. Analogously, in [19] an algorithm is proposed for Wi-Fi which selects for each data the number of retries and their bit rate to minimize the resulting error probability while not exceeding the deadline. The above solutions consider individual requirements of deadline-constrained traffic but, unlike our approach, they do not address concurrency explicitly.

Concerning proposals that explicitly take concurrency into account, [20] presents a solution, based on EDF scheduling, to minimize energy consumption in wireless sensor networks (WSN) supporting deadline-constrained data exchanges. Although the goal is finding a trade-off between timeliness and energy consumption (modulation schemes with a lower number of bits per symbol are more energy-efficient but imply lower bit rates), the framework is quite similar to ours. However, mechanisms for guaranteeing a required level of reliability are not considered there.

Some works [21]–[23] address both timeliness and reliability under concurrent medium access by relying on centralized transmission scheduling implemented over conventional TDMA protocols. Although these solutions do not rely on polling, they partially suffer from the same limitations as cyclic scheduling, and do not allow any dynamic bandwidth management.

Centralized access is also foreseen by standard solutions like the IEEE 802.11 Hybrid Coordination Function (HCF). In particular, the HCF controlled channel access (HCCA) is based on online resource reservation and polling, and relies on a hybrid coordinator (HC) that schedules transmissions according to their traffic specifications (TSPEC), possibly coupled with transmission opportunities (TXOP), in order to allow dynamic bandwidth allocation to nodes. As HCCA only specifies a rather basic bandwidth allocation algorithm, enhanced solutions have been proposed, tailored to different scenarios. As an example, in [24], [25] two feedback-based dynamic schedulers (FBDS) and an admission control mechanism are presented for handling bursty traffic. FBDSs dynamically divide available bandwidth among nodes based on runtime information about their message queue length, and are shown to offer tangible improvements, in terms of latency, with respect to cases in which allocation is statically performed based on nominal traffic rates. However, most of the proposed enhancements for HCCA, including [24], [25], are conceived for multimedia traffic and focus on providing guarantees on average performance, but are not able to ensure a fine-grained (per data instance) control on timeliness, as demanded by many real-time industrial control applications.

Holistic solutions based on centralized EDF scheduling and ARQ for periodic deadline-constrained data delivery over unreliable networks were proposed in [9], [10]. In [9] a predefined number of consecutive retries (logical real-time channels) is ensured per-instance through admission control. Moreover, a bandwidth portion is statically allocated for providing additional retries to failed instances, by including virtual periodic time-critical data (retransmission channels) in feasibility analysis. While this approach resembles ours, as it provides guarantees on a per-instance basis, no mechanism is foreseen to recover preallocated unused bandwidth, leading to suboptimal usage. The proposal in [10] is grounded on the same principles and provides a more optimistic admission control test by exploiting a priori information about transmission error probabilities. Authors consider channels with known transmission error probabilities, constant over time, and perform feasibility analysis on a time-critical data set where periods of tasks modeling retries are set equal to their average values. This means that, as in most realizations errors are not distributed exactly as assumed in feasibility analysis, the likelihood to have unexpectedly distributed deadline misses grows. The solution in [10] allows for looser admission control, but does not offer any guarantees on a per-instance basis.

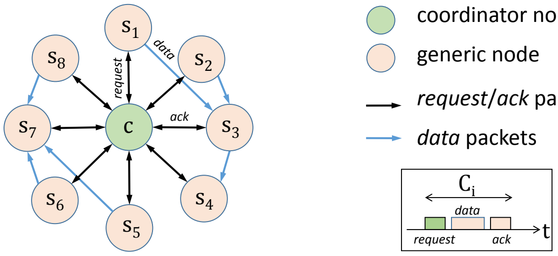

Fig. 1: Example of network configuration with 9 nodes.

It is worth remarking that, when random elements (e.g., channel errors and collisions) may affect the outcome of performed transmissions, even approaches relying on feasibility analysis are unable to offer absolute guarantees on timeliness of data delivery. Nevertheless, as scheduling decisions are taken, for every single instance, based on both its deadline and the current system state (i.e., considering concurrency), tight control is provided on timing of transmission executions. Predictability of actions allows, whenever the statistical description of above random elements is available, to provide strict probabilistic guarantees on data instance delivery.

Scheduling deadline-constrained time-critical data transmissions and retransmissions closely resembles the problem of task scheduling in presence of faults, which has been extensively treated in the context of computer systems [26]. As an example, the algorithm described in [27] allows to determine if a task set is feasible when up to k faults, arbitrarily distributed, occur at runtime, and the corresponding recovery blocks are executed. However, having k errors affecting a single data transmission in a communication system is unlikely to happen, and checking whether a data set is k-fault tolerant is pessimistic. Moreover, algorithm in [27] is conceived for preemptive EDF and the authors show that the non-preemptive case is intractable. Other examples in this sense are works on message scheduling in the presence of transmission errors in CAN networks, e.g., [28], where, however, fixed priority strategies are considered. The idea to reuse the otherwise unused allocated share of processor time in order to improve average performance has been also proposed in the context of real-time OS, e.g., in the Shared Reclamation of Unused Bandwidth (SHRUB) algorithm [29].

III. SYSTEM MODEL

This section describes the scenario we consider and the time-critical communication model we employ in feasibility analysis for admission control. As the approach is of general validity, we purposely do not refer to any specific wireless technology, providing a high-level framework within which the characteristics of different solutions can easily be integrated.

We consider IDCS relying on a single-hop wireless network, consisting of an arbitrary number of nodes communicating over the same frequency range. We assume the network to be fully configured (i.e., any node knows the addresses of all nodes it has to communicate with) and stationary (i.e., nodes and communication links do not change at runtime). As in practical applications and with negligible loss of generality, time is discrete (the time unit being, e.g., the transmission time of a logical bit over the channel or the period of a reference clock). We suppose the implemented soft real-time control application to require the periodic exchange over the network of N time-critical data. Each time-critical data i originates at a source node src_i, is updated every T_i, and its updated values (*instances*) have to be delivered to a destination node dst_i within a predefined time interval D_i (*firm* relative deadline) from the update instant. To deliver time-critical data at the required rates and within the associated deadlines, the network relies on a MAC protocol based on centralized transmission scheduling, according to the EDF policy. A central node (*coordinator*), aware of time-critical data characteristics, explicitly schedules the transmission of each instance, borne in a *data* packet, by sending a *request* packet to the source node (see Fig. 1). Preemption is not allowed as the coordinator cannot stop ongoing transmissions from other nodes.

A. Time-critical communication model

Following a well-established approach, also adopted in [9], [10], the periodic transmission of any time-critical data i is modeled as a real-time periodic task τ_i defined as

$$\tau_i = (src_i, dst_i, \phi_i, T_i, C_i, D_i) \tag{1}$$

where ϕ_i is the phase, while T_i, C_i, and D_i are period, worst-case execution time, and relative deadline of the task (representing, respectively, the first update time, required delivery rate, maximum duration of a single transmission attempt, and relative deadline of time-critical data i). With minimal loss of generality, task parameters are assumed to be integer multiples of the time unit. We consider the constrained-deadline case where $D_i \leq T_i$, as in most control applications any update has to be delivered before a new update is available. The worst-case execution time C_i models the maximum duration of a single transmission attempt for any instance of time-critical data i. Thus, in the general case of node-to-node data delivery (see Fig. 1), it accounts for the transmission of both *request* and *data* packets, plus additional packets possibly introduced by the protocol stack (e.g., acknowledgments) and interframe times. Parameters C_i also account for processing delays (e.g., due to concurrently executing tasks) possibly introduced by involved nodes (e.g., coordinator, src_i, and dst_i). Typically, processing delays can be estimated based on device specifications or through preliminary measurement campaigns. If nodes rely on non real-time OS, including suitable safety margins may reveal necessary. See [16] for a thorough discussion on the issues, especially related to the definition of parameters C_i, encountered in the implementation of a similar framework on nodes running the Linux OS.

Periodic transmission of all time-critical data is modeled as a set of N real-time periodic tasks, defined as in (1), that is $\Gamma = \{\tau_i\}$, $i = 1, \ldots, N$.

B. Admission control in ideal channel conditions

Under the above hypotheses and assuming ideal channel conditions (i.e., transmissions on air never fail), checking whether the network satisfies timing requirements for the time-critical data set modeled by Γ means verifying feasibility of Γ under non-preemptive EDF.

To this aim, if $D_i = T_i$, Baker test [30] can be used, which considers task blocking on shared resources and, when preemption is not allowed, replaces the traditional Liu and Layland test [31]. If tasks in Γ are sorted by increasing relative deadlines (i.e., $j > i \Rightarrow D_j \geq D_i$, $\forall i, j = 1, \ldots, N$), a sufficient condition for Γ to be feasible under non-preemptive EDF is that the following inequalities hold

$$\sum_{i=1}^{k} \left(\frac{C_i}{T_i} \right) + \frac{B_k}{T_k} \leq 1, \quad k = 1, \ldots, N \quad \text{(Baker test)} \tag{2}$$

where B_k is the maximum blocking time experienced by the k-th task and corresponds to the maximum among durations of all other tasks in Γ [28]. Explicitly, inequalities (2) become

$$\sum_{i=1}^{k} \left(\frac{C_i}{T_i} \right) + \frac{\max_{i \neq k}\{C_i\}}{T_k} \leq 1, \quad k = 1, \ldots, N \tag{3}$$

If $D_i < T_i$, analysis in [32] applies. The duration of the first busy period in case of *asap* arrival pattern (i.e., $\phi_i = 0$, $\forall i = 1, \ldots, N$) is computed using the iterative formula

$$\begin{cases} L^{(0)} = \sum_{i=1}^{N} C_i \\ L^{(t+1)} = \sum_{i=1}^{N} \left\lceil \frac{L^{(t)}}{T_i} \right\rceil C_i \end{cases} \tag{4}$$

which, if $U(\Gamma) = \sum_{i=1}^{N} C_i/T_i \leq 1$, converges to the duration of the first busy period $\overline{L}$ in a finite number of steps. Then, a sufficient condition for Γ to be feasible is that, for any absolute deadline d in the first busy period $\overline{L}$, the following holds

$$\sum_{i:D_i \leq d} \left[\left(1 + \left\lfloor \frac{d - D_i}{T_i} \right\rfloor \right) C_i \right] + B_{k_d} \leq d \tag{5}$$

where B_{k_d} is the maximum blocking time experienced by the k_d-th task, that is the task with the largest relative deadline among those considered in the summation (i.e., among $\tau_i \in \Gamma : D_i \leq d$). Explicitly, inequality (5) becomes

$$\sum_{i:D_i \leq d} \left[\left(1 + \left\lfloor \frac{d - D_i}{T_i} \right\rfloor \right) C_i \right] + \max_{i \neq k_d}\{C_i\} \leq d \tag{6}$$

C. ARQ scheme and retransmission strategies

Even though combining centralized transmission scheduling and admission control guarantees timeliness, reliability still need to be addressed. Indeed, several phenomena affect the ether (e.g., path loss, signal attenuation, fading, noise, and extra-network interference), often resulting in high transmission error probabilities, and deadlines may be missed because of the failure of timely performed transmissions. Furthermore, as in our approach intra-network collisions are prevented by centralized transmission scheduling, no additional mechanisms are explicitly foreseen to defend ongoing transmissions on the channel (i.e., nodes are supposed not to perform any carrier sensing before transmitting). Therefore, there is an even higher probability of collisions due to transmissions coming from nearby networks operating on the same frequency range (extra-network interference).

To improve reliability towards the application functional requirements, we suppose the network to provide a basic truncated stop-and-wait ARQ protocol, where destination nodes acknowledge the correct reception of any time-critical data instance by transmitting an *ack* packet to the coordinator. When the *ack* packet is not received within a given timeout, the instance transmission is marked as failed and is, possibly, repeated. A refined description of tasks in Γ is then

$$\tau_i = (src_i, dst_i, \phi_i, T_i, \{C_{i,j}\}_{j\in\mathbb{N}^+}, D_i) \tag{7}$$

where $\{C_{i,j}\}_{j\in\mathbb{N}^+} = \{C_{i,1}, \ldots, C_{i,j}, \ldots\}$ is the sequence of worst-case durations $C_{i,j}$ of (possibly performed) j-th transmission attempt for instances of time-critical data i. Sequences $\{C_{i,j}\}_{j\in\mathbb{N}^+}$ model possibly different worst-case durations characterizing distinct transmission attempts. This way it is possible to describe, e.g., the adoption of strategies aimed at finding suitable trade-offs between reliability and timeliness. As an example, in Wi-Fi networks, predefined sequences of (decreasing) rates may be used for the different transmissions attempts of single instances, in a sort of RA performed on a per-instance basis. Analogously, the model is able to describe, e.g., specific DDC schemes employed for the transmission of each instance.

Since traffic is deadline-constrained, it makes no sense to retransmit instances an arbitrary number of times, as this would likely lead to exceeding deadlines and leave no time for other instances. Each time-critical data is thus assigned a configurable maximum number of planned retries and its instances are retransmitted until either they are successfully delivered or the limit is reached. We assume the characteristics of the radio environment to be known to the system designers and application functional requirements to be expressed as an array $R = [R_1 \ \ldots \ R_N]$, where $R_i \in \mathbb{N}_0^+, \ \forall i = 1, \ldots, N$, is the number of planned retries that must be ensured, within the deadline, to each instance of time-critical data i. Such guarantees provide the functional level of combined reliability and timeliness required by the soft real-time control application, which should never go unmet at runtime. However, any improvement (e.g., determined by additional retries), allowing for increased reliability within the limits imposed by timeliness, is highly desirable.

We consider two distinct retransmission strategies [16], denoted *consecutive* and *preemptable*. Consecutive retransmissions are uninterruptedly performed after the first attempt, while preemptable ones are assigned the same absolute deadline as the first attempt and are scheduled according to EDF (i.e., they can be interleaved with other, newly activated instance transmissions with tighter absolute deadlines). The main difference between the two strategies is that with consecutive retransmissions the scheduler is involved just in the first transmission attempt of any instance, while with preemptable retransmissions any retry undergoes a distinct scheduling decision. As a consequence, the consecutive strategy may be more effective if the scheduling overhead is significantly high, for example when the ARQ scheme is implemented at a lower layer in the coordinator protocol stack with respect to the scheduler. In this case, depending on the latency introduced by the interface between layers, retransmission management can be conveniently left to the lower layer or even moved to the source node (i.e., retries are not explicitly triggered by *request* packets, leading to lower $C_{i,j}$ when $j > 1$).

The actual implementation of time-critical data exchanges can be defined case by case, provided that the coordinator is aware of the instants in which instances are successfully delivered or their planned number of retries is reached, and can thus proceed with the schedule. Moreover, the consecutive strategy makes the adoption of RA techniques (e.g., ad-hoc ones operating on single instances as mentioned above) more effective, as longer sequences of packets are sent on the same communication links. Conversely, when the scheduling overhead is low, the preemptable strategy makes the system more responsive, since retransmissions of failed time-critical data instances can be postponed in case instances with higher priority (i.e., earlier absolute deadlines) become ready in the meanwhile, leading to lower blocking times and a more efficient bandwidth utilization (looser admission control).

D. Admission control with retransmissions

Given a time-critical data set modeled by Γ, with tasks defined as in (7), and functional requirements R, we say that set Γ is *R-admissible* under a given retransmission strategy if at least R_i retries can be performed at runtime per each instance of any time-critical data i within its deadline. Checking whether a task set Γ is R-admissible is equivalent to verifying feasibility under non-preemptive EDF of a new task set $\mathcal{R}^s(\Gamma, R)$, derived from Γ and R, and depending on the employed retransmission strategy $s \in \{c, p\}$, which models first transmission attempts and planned retries.

In the consecutive strategy, first attempt and subsequent retries are tied together and cannot be interleaved with other transmissions, so the periodic transmission of each time-critical data and its planned retries can be modeled as a single task with worst-case execution time computed as the time required to perform $1 + R_i$ transmission attempts. The corresponding task set $\mathcal{R}^c(\Gamma, R)$ can then be defined as

$$\mathcal{R}^c(\Gamma, R) = \{\tau'_1, \tau'_2, \ldots, \tau'_N\} \tag{8}$$

where $\tau'_i = (src_i, dst_i, \phi_i, T_i, C'_i = \sum_{j=1}^{1+R_i} C_{i,j}, D_i)$. If $D_i = T_i$, adapting (3) to the newly defined task set, we obtain that a sufficient condition for Γ to be R-admissible under consecutive retransmissions is that inequality

$$\sum_{i=1}^{k} \sum_{j=1}^{1+R_i} \left(\frac{C_{i,j}}{T_i} \right) + \frac{\max_{i \neq k}\{C'_i = \sum_{j=1}^{1+R_i} C_{i,j}\}}{T_k} \leq 1 \tag{9}$$

holds for any $k = 1, \ldots, N$. Similarly, if $D_i < T_i$, (4) and (5) can be adapted to task set $\mathcal{R}^c(\Gamma, R)$.

In case of preemptable retransmissions, first attempts and planned retries are modeled as a set of $\sum_{i=1}^{N}(1 + R_i)$ distinct real-time periodic tasks $\tau_{i,j} = (src_i, dst_i, \phi_i, T_i, C_{i,j}, D_i)$, $i = 1, \ldots, N, \ j = 1, \ldots, 1 + R_i$, where any $\tau_{i,j}$ represents the periodic sequence of exchanges determined by the j-th transmission attempts of instances of time-critical data i. Task

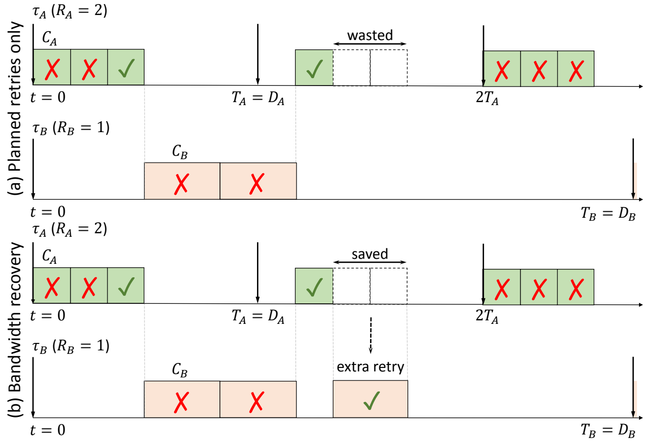

Fig. 2: Example of performance enhancement achieved by dynamic bandwidth management.

instances modeling retries are supposed to activate in the same instant as the one modeling first transmission attempt and to have the same relative deadline (i.e., first transmission attempt and planned retries have the same absolute deadline). Thus, the corresponding task set $\mathcal{R}^p(\Gamma, R)$ can be defined as

$$\mathcal{R}^p(\Gamma, R) = \{\underbrace{\tau_{1,1}, \ldots, \tau_{1,1+R_1}, \ldots, \tau_{N,1}, \ldots, \tau_{N,1+R_N}}_{\sum_{i=1}^{N}(1+R_i)}\} \tag{10}$$

If $D_i = T_i$, by applying test (3) to $\mathcal{R}^p(\Gamma, R)$, we obtain that a sufficient condition for Γ to be R-admissible under preemptable retransmissions is that inequality

$$\sum_{i=1}^{k}\sum_{j=1}^{1+R_i}\left(\frac{C_{i,j}}{T_i}\right) + \frac{\max_{(i,j)\neq(k,1+R_k)}\{C_{i,j}\}}{T_k} \leq 1 \tag{11}$$

holds for any $k = 1, \ldots, N$. In (11) only the final retry of instances of time-critical data k is considered. In fact, if condition (11) holds for the final retry $\tau_{k,1+R_k}$, by necessity it holds for all the former retries $\tau_{k,j}, \forall\ j < 1 + R_k$. Again, if $D_i < T_i$, (4) and (5) can be adapted to task set $\mathcal{R}^p(\Gamma, R)$.

A comparison of (9) and (11) shows that a major difference between admission control tests for the two retransmission strategies is the maximum blocking time that can be experienced by tasks. This makes admission control more restrictive with consecutive retransmissions. However, depending on the actual implementation, employing the consecutive strategy may lead to lower overhead (which means shorter worst-case duration of transmission attempts $C_{i,j}$).

IV. DYNAMIC BANDWIDTH MANAGEMENT

Given an R-admissible set Γ, the fraction of bandwidth which is statically allocated for real-time communication is $U_{max}(\Gamma, R) = \sum_{i=1}^{N}\sum_{j=1}^{1+R_i} C_{i,j}/T_i \leq 1$. However, admission control tests derived in Sect. III are pessimistic, as they consider the worst-case scenario where every instance undergoes all planned retries. If the number of retries is suitably selected, this is very unlikely to occur as, during operations, most time-critical data are successfully delivered using less retries than allotted. Thus, if no further action is undertaken, a substantial part of the statically allocated bandwidth U_{max} is likely wasted. In this section we consider mechanisms to increase, for each time-critical data i, the percentage of instances successfully delivered within the deadline (*delivery success percentage*, DSP_i) beyond the functional level determined by R_i, without jeopardizing statically provided guarantees and not exceeding $U_{max}(\Gamma, R)$.

In this direction, *a posteriori* knowledge on transmission outcomes can be exploited in order to dynamically reassign unused bandwidth to those instances that were not successfully delivered after all the planned retries and whose deadline has not yet expired. Fig. 2 shows a basic example which explains how dynamic bandwidth recovery works. A set $\Gamma = \{\tau_A, \tau_B\}$ of two time-critical data modeled by tasks $\tau_A = (T_A, \{C_{A,j}\}_{j\in\mathbb{N}^+}, D_A)$ and $\tau_B = (T_B, \{C_{B,j}\}_{j\in\mathbb{N}^+}, D_B)$ is considered, where $T_A = D_A = 6$, $C_{A,j} = 1$, $\forall j \in \mathbb{N}^+$, $T_B = D_B = 16$, and $C_{B,j} = 2$, $\forall j \in \mathbb{N}^+$, time units. We assume functional requirements to be expressed as $R = [R_A\ R_B]$, with $R_A = 2$ and $R_B = 1$. By applying test (11), Γ is R-admissible under the preemptable strategy. When no dynamic bandwidth reassignment is adopted, some of the planned retries are not performed and, as depicted in part (a) of Fig. 2, the corresponding preallocated bandwidth is wasted. Conversely, part (b) of Fig. 2 shows that, in the same conditions, unperformed retries of instances of time-critical data A can be reassigned to instances of time-critical data B to increase their DSP. Clearly, bandwidth reassignment has to be carefully managed to avoid guarantees on planned retries going unmet at runtime.

In this paper we consider two distinct dynamic bandwidth management strategies, namely *Saved Bandwidth First* (SBF) and *limited-Planned Transmissions First* (l-PTF). With SBF, any time a time-critical data instance is successfully delivered using less retries than planned, the saved bandwidth is immediately reassigned (if possible) to provide *extra* retries (i.e., retries exceeding the planned number) to other active time-critical data instances, selected according to EDF. Conversely, with l-PTF extra retries are performed only when the system is idle (i.e., after all active planned transmission attempts have been carried out). The first strategy tries to follow EDF rigorously, also when managing the reassigned bandwidth. However, in order not to impair the envisaged guarantees, the saved bandwidth cannot be inherited and exploited in any circumstance. By contrast, the second strategy fully leverages the unused bandwidth, but extra retries are carried out with lower priority than the planned ones. In theory, l-PTF may use the entire available network bandwidth (and not only recovered retries). However, to permit a fair comparison with SBF, we set the algorithm so as to not exceed $U_{max}(\Gamma, R)$. Besides, leaving some spare bandwidth for non real-time aperiodic exchanges (e.g., for exchanges related to resource reservation) is typically required.

In the following, we will suppose that durations of extra retries cannot exceed durations of planned attempts, i.e., $\forall i = 1, \ldots, N$, $C_{i,j} \leq \max_{k\leq 1+R_i}\{C_{i,k}\}\ \forall j > 1 + R_i$. This hypothesis is meant to simplify the definition of bandwidth management algorithms, and basically implies that extra retries can not affect feasibility analysis unexpectedly. If the hypoth-

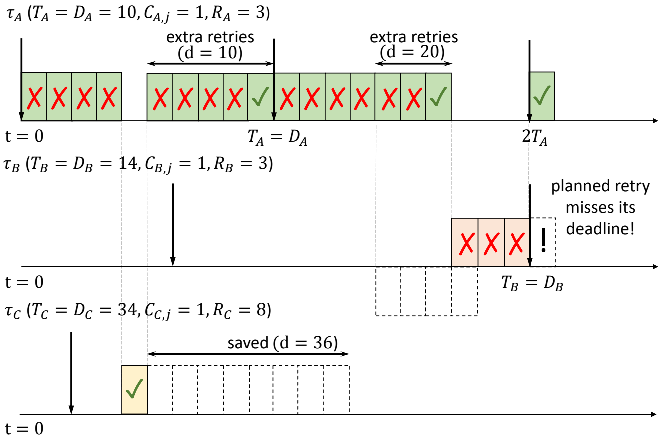

Fig. 3: Example of wrong dynamic bandwidth management.

esis is not satisfied in practice, simple modifications can be brought to the algorithms to account for the actual duration of extra retries. In addition, we will neglect the additional computation times introduced by the coordinator due to these strategies. This is typically acceptable, provided that optimized algorithms are implemented in firmware. On the contrary, $C_{i,j}$ values can be enlarged accordingly.

A. *Saved Bandwidth First (SBF)*

When SBF is adopted, each instance of any time-critical data i is statically allocated a *time budget* β_i which, upon instance activation, is set to $\bar{\beta}_i = \sum_{j=1}^{1+R_i} C_{i,j}$, i.e., the amount of time required to perform all planned retries (not necessarily in a consecutive way, e.g., if preemptable retransmissions are employed). After any transmission attempt j is performed (where $j = 1, \ldots, 1 + R_i$), β_i is decremented by a quantity $C_{i,j}$. Whenever an instance is successfully delivered within a number of attempts $r < 1 + R_i$, the pair $\langle \beta_i, d \rangle$, formed by the remaining time budget $\beta_i = \sum_{j=r+1}^{1+R_i} C_{i,j}$ and the absolute deadline d originally assigned to the instance, is stored in a common pool. Any time an instance is selected for transmission, according to the EDF policy, the *saved time* in the common pool whose associated absolute deadline is lower than the instance's absolute deadline (if any is available) is used first, without consuming preassigned endowments.

Constraints on saved bandwidth usage are necessary in order not to impair guarantees on planned retries. Fig. 3 shows an example of wrongly performed dynamic bandwidth management: unused retries of an instance from a low frequency task τ_C are inherited and employed by two different instances of a higher frequency task τ_A, without obeying the absolute deadline rule. As a consequence, an instance of task τ_B with intermediate frequency is preempted by a number of higher-priority retransmissions larger than considered in feasibility analysis, and hence it is no longer able to exploit all its planned retries before the deadline expires. To prevent this misbehavior, SBF forbids any instance to reuse portions of saved time whose associated absolute deadlines fall after the deadline of the instance itself (in reality, of the instance's successor in EDF order). Any block of unused saved time is definitively lost as soon as its deadline expires.

```
while (true) do
    $\boldsymbol{\tau_i}$=select_task_EDF();
    $\boldsymbol{\tau_k}$=succ_task($\boldsymbol{\tau_i}$)
    if $(\boldsymbol{\tau_i}.j < \boldsymbol{\tau_i}.R + 1)$ or $(\boldsymbol{\tau_i}.C_{j+1} \leq$ saved_time$(\boldsymbol{\tau_k}.d))$
then
        tx_success=transmit($\boldsymbol{\tau_i}$);
        if $(\boldsymbol{\tau_i}.C_{j+1} \leq$ saved_time$(\boldsymbol{\tau_k}.d))$ then
            sub_from_ST($\boldsymbol{\tau_i}.C_{j+1}$,$\boldsymbol{\tau_k}.d$);
        else
            $\boldsymbol{\tau_i}.\beta = \boldsymbol{\tau_i}.\beta - \tau_i.C_{j+1}$+saved_time($\boldsymbol{\tau_k}.d$);
            sub_from_ST(saved_time($\boldsymbol{\tau_k}.d$),$\boldsymbol{\tau_k}.d$);
        end if
        if (tx_success) then
            if $(\boldsymbol{\tau_i}.j < \boldsymbol{\tau_i}.R + 1)$ then
                add_to_ST($\boldsymbol{\tau_i}.\beta$, $\boldsymbol{\tau_i}.d$);
            end if
            $\boldsymbol{\tau_i}.j = 0$; $\boldsymbol{\tau_i}.\beta = \boldsymbol{\tau_i}.\bar{\beta}$; $\boldsymbol{\tau_i}.d = \boldsymbol{\tau_i}.d + \boldsymbol{\tau_i}.T$;
            $\boldsymbol{\tau_i}.s$=inactive;
        else
            $\boldsymbol{\tau_i}.j = \boldsymbol{\tau_i}.j + 1$;
            if $(\boldsymbol{\tau_i}.j = \boldsymbol{\tau_i}.R + 1)$ then
                add_to_ST($\boldsymbol{\tau_i}.\beta$, $\boldsymbol{\tau_i}.d$); $\boldsymbol{\tau_i}.\beta = 0$;
            end if
        end if
    end if

end while
```

Fig. 4: Saved Bandwidth First (SBF) service policy.

The SBF strategy for preemptable retransmissions is more formally described by the algorithm in Fig. 4. With slight abuse of notation, $\boldsymbol{\tau_i}$ is a structure containing all information (static and dynamic) relevant to data i, that is: period T, number of planned retries per instance R, sequence of transmission durations $\{C_j\}_{j \in \mathbb{N}^+}$, and initial (preassigned) time budget $\bar{\beta}$ associated to any instance, as well as, for the current instance of the task, absolute deadline d, number j of already performed transmission attempts, status s (either "active" if the instance is ready for transmission or "inactive" otherwise), and remaining (unused) part β of the preassigned time budget.

TQ is a queue which includes all current task instances (both active and inactive ones), while ST is the common pool of saved time intervals due to unperformed retransmissions, each one characterized by its duration and associated absolute deadline. Both TQ and ST store their elements in increasing absolute deadline order. Function $select\_task\_EDF()$ returns the identifier of the active instance with earliest absolute deadline in TQ, while $succ\_task(\boldsymbol{\tau_i})$ returns the identifier of $\boldsymbol{\tau_i}$ successor in TQ. Function $saved\_time(d)$ computes the total amount of saved time in ST whose associated deadline is earlier than d, while $add\_to\_ST(time, d)$ adds the time interval $time$ with associated deadline d to ST. Similarly, $sub\_from\_ST(time, d)$ subtracts an amount of time $time$ from saved time intervals in ST whose associated deadline is less than or equal to d. Finally, $transmit(\boldsymbol{\tau_i})$ performs one transmission attempt for the current instance of $\boldsymbol{\tau_i}$ while $tx\_success$ is used to store its outcome.

Pseudo-code in Fig. 4 only shows the scheduling algorithm for the current instance, while details concerning management of saved time expiration and periodic task activation are omitted. Algorithm refers to the preemptable retransmission strategy. When consecutive retransmissions are employed, first transmission attempts and planned retries are managed according to the consecutive strategy, while SBF bandwidth recovery works as in Fig. 4.

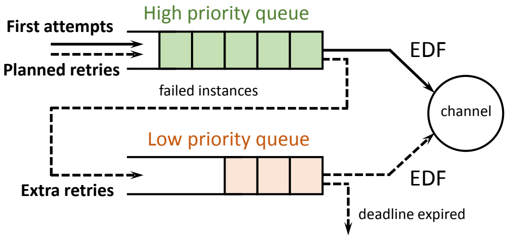

Fig. 5: l-PTF service policy

TABLE I: Task set Γ

task	src_i	dst_i	ϕ_i	$T_i = D_i$	$C_{i,j} = C_i$	$R_i = R$
τ_1, τ_2	s_1	s_3	0 μs	3000 μs	164 μs	2
τ_3	s_2	s_3	0 μs	5500 μs	164 μs	2
τ_4	s_3	s_4	0 μs	5500 μs	164 μs	2
τ_5	s_5	s_7	0 μs	7000 μs	164 μs	2
τ_6	s_6	s_7	0 μs	7000 μs	164 μs	2
τ_7, τ_8	s_8	s_7	0 μs	10000 μs	308 μs	2

B. Limited Planned Transmissions First (l-PTF)

The idea behind l-PTF is sketched in Fig. 5. First attempts and (possibly needed) planned retries are inserted in a high priority queue, and their transmission is scheduled according to the EDF policy under either the consecutive or preemptable retransmission strategy. Conversely, extra retries are activated any time they are needed (i.e., whenever an instance fails to be successfully delivered after all planned retries) and inserted in a lower priority queue, leaving their absolute deadline equal to the one associated to the instance they refer to. The total amount of saved time due to unperformed planned retries is continuously kept up to date (this is why the strategy is termed *limited*). For both consecutive or preemptable strategies, extra retries in the lower priority queue are singularly scheduled according to EDF. However, a single extra retry is performed only when both the following conditions hold: *i*) the higher priority queue is empty, and *ii*) the amount of saved time is enough to perform the retry.

Functional requirements are granted thanks to the double queue: basically, first attempts and planned retries are managed as real-time traffic, while extra retries are treated as best-effort and only performed when no real-time traffic is active. With l-PTF, execution of first attempts and planned retries is thus marginally affected by extra retries, the only difference being that, sometimes, planned transmission attempts may be blocked by (at most) one extra retry. However, since admission control tests (9) and (11), as well as those for the case $D_i < T_i$, assume (pessimistically) that any instance is blocked by the longest among the other instances in the set, and extra retries are supposed not to last longer, feasibility analysis is still valid.

V. SIMULATION RESULTS

To validate the proposed approach and compare the different retransmission and dynamic bandwidth recovery strategies, a simulation tool was developed, based on the OMNET++ discrete simulator engine [33], and used for extensive simulation campaigns. As our simulator focuses on application-layer time-critical data exchange, it does not rely on any protocol-specific modules. Conversely, it assumes that transmissions (including inter-frame times and processing delays) last exactly the specified worst-case durations and no other reliability-improving mechanisms (e.g., RA or DCC) are employed. These arrangements make the analysis below a good reference basis, and enable the simulator to be possibly extended considering different specific wireless technologies and channel conditions.

We considered the network configuration depicted in Fig. 1 and the time-critical data set modeled by Γ in Tab. I, whose characteristics resemble data exchanges in automation cells for food production and packaging. Requirements, in this kind of applications, are soft real-time, as occasional losses or delays in time-critical data delivery are tolerated at the expenses of (small) degradations in product quality, while no damage can be caused to either humans or equipment involved in plant operation. Typically, data exchange periods are in the order of milliseconds. Maximum transmission times ($C_{i,j} = C_i = 164$ μs or 308 μs, $\forall j \in \mathbb{N}^+$) are compatible with the duration of a three-way handshake implemented, e.g., on Wi-Fi, for the delivery of 30 and 350 byte time-critical data payloads, respectively. To provide results as general as possible, worst-case durations of transmission attempts for instances of the same time-critical data are assumed not to vary.

As our goal was to compare the different strategies in the same operating conditions, simulations rely on a simple channel model, where each transmission attempt of any instance of time-critical data i is modeled as an independent Bernoulli trial with failure probability e constant over time. Transmission error probability e is supposed to take into account all errors that may affect any of the packets involved in a single transmission attempt (e.g., *request*, *data*, and *ack*).

At first, we considered the case $D_i = T_i$. Assuming $R_i = 2$, identical for all time-critical data, and applying tests (9) and (11), Γ is found to be R-admissible under both the preemptable and consecutive retransmission strategies. In the following, with a slight abuse of notation, we use the same symbol R for both the vector of planned retries $R = [R_1 \; \dots \; R_8]$ and the single value chosen for all R_i in our simulations. In such conditions, the maximum bandwidth utilization for time-critical communication is $U_{max} = 0.8322$, leaving approximately 16% of the bandwidth available for best-effort traffic.

Simulations were carried out for every combination of retransmission and dynamic bandwidth management strategies, by varying the channel transmission error probability e in the range $[0, 1]$ with 0.01 step increments. For each value of e, the network behavior was observed for 300 s, corresponding to 454808 total scheduled instances, not counting retransmissions, of which 30000 for time-critical data τ_7 and τ_8 with the lowest frequency. In each simulation, the percentage of time-critical instances successfully delivered within the deadline (DSP_i) was evaluated for each time-critical data i.

As a baseline, we carried out two runs of simulations, for either of the considered retransmission strategies, without exploiting any dynamic bandwidth management. Since results with the two retransmission strategies are almost identical, we only report those obtained with preemptable retransmissions

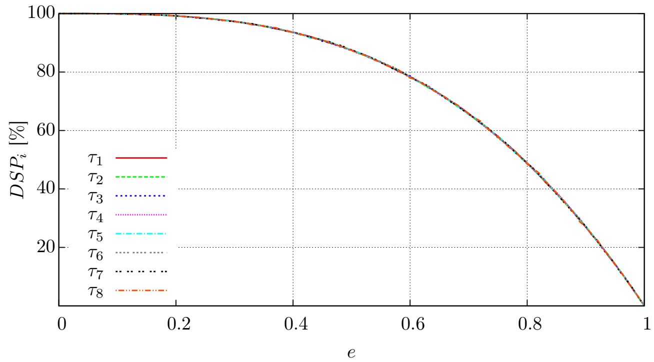

Fig. 6: Simulated delivery success percentage with planned retries only (*preemptable* retransmissions).

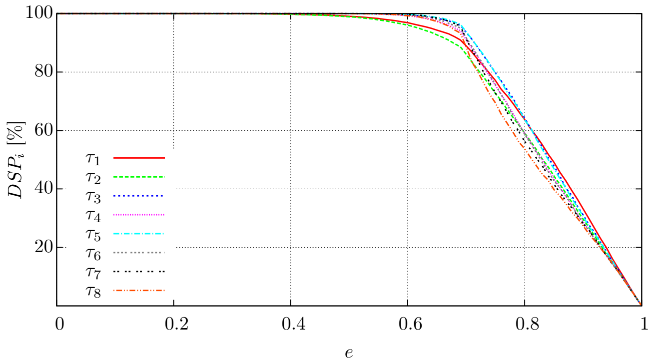

Fig. 8: Simulated delivery success percentage with l-PTF dynamic bandwidth reassignment (*preemptable* retransmissions).

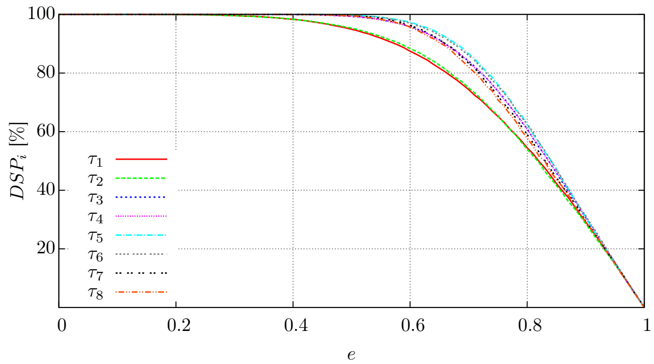

Fig. 7: Simulated delivery success percentage with SBF dynamic bandwidth reassignment (*preemptable* retransmissions).

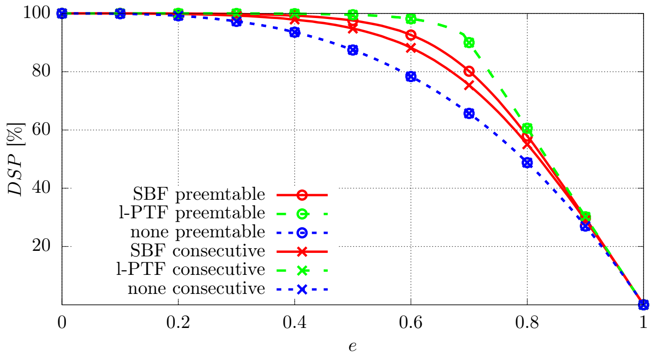

Fig. 9: Comparison among *cumulative DSP* with l-PTF, SBF or no recovery (*preemptable* and *consecutive* retransmissions).

(Fig. 6). As expected, since all time-critical data are assigned the same number of planned retries, curves related to different time-critical data overlap, and each DSP_i decreases as e grows higher according to the law $DSP_i = 1 - e^{1+R}$ (representing the probability that an instance is successfully delivered over the channel within R retries when the transmission error probability is e).

Figs. 7 and 8 show results obtained in the same conditions using SBF and l-PTF, respectively. Note that, in all simulations, no deadlines were missed by either first attempts or planned retries of any instances, confirming the validity of our analysis. These figures make it also clear that dynamic bandwidth reassignment significantly improves DSP_i. However, performance improvements in SBF are not fairly distributed among time-critical data. As can be noted from Fig. 7, time-critical data with the shortest period (τ_1 and τ_2) take less advantage from bandwidth recovery than those with larger T_i. This happens because, due to the limitation imposed by SBF on bandwidth reassignment, instances of τ_1 and τ_2 (having the shortest period/relative deadline) are less likely to find available saved times with earlier absolute deadline. Performance and fairness of l-PTF are even better than SBF. Improvements, in this case, are mainly due to the ability of l-PTF to reuse a larger portion of the saved bandwidth with respect to SBF, since constraints on reassignment are not as strict and do not depend on the inheriting time-critical data instance.

Fig. 9 compares results obtained with SBF and l-PTF to the baseline case (curves labeled "none") when preemptable and consecutive retransmissions are employed, respectively. For clarity reasons, plots concern the overall percentage of time-critical data successfully delivered within the deadline (*cumulative DSP*) instead of showing results on a per-data basis. The figure permits to better appreciate performance increases achieved by SBF and l-PTF with respect to the case in which only planned retries are performed. Note that results are slightly worse when SBF is coupled with consecutive retransmissions because, especially for tasks with shorter T_i, delays caused by larger blocking times reduce the likelihood of finding usable blocks of saved time.

Figs. 10 and 11 show the average number $\bar{r}_i$ of transmission attempts per time-critical data instance for both the preemptable and consecutive retransmission strategies. In each figure, the theoretical average number of per-instance transmission attempts required to successfully deliver all instances in the "ideal" case where infinite bandwidth is available and unlimited retries are allowed (computed as $\bar{r}_I = 1 + \dfrac{e}{1-e}$) is compared to the average number of transmission attempts per instance obtained for each time-critical data in the simulations when using SBF, l-PTF, or none of them. As expected, when only planned retries are performed, all time-critical data undergo the same average number of transmission attempts, which can be computed as $\bar{r}_P = 1 + \dfrac{e}{1-e}(1 - e^R) \leq \bar{r}_I$. Importantly, whatever the recovery strategy, provided that $U \leq U_{max}(\Gamma, R)$ the aggregate $\bar{r}$ evaluated over all time-critical data cannot exceed $\bar{r}_R = 1 + R$. With both SBF and l-PTF, the average number of attempts per instance $\bar{r}_i$ (which

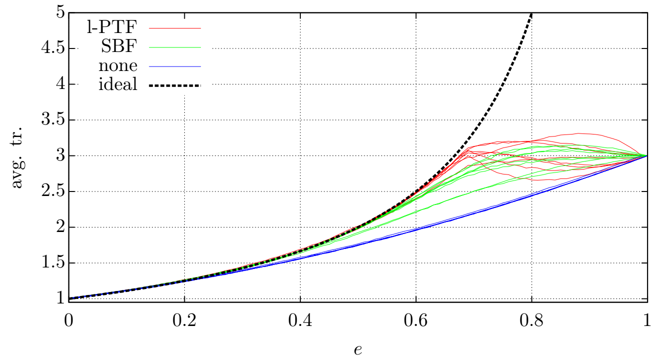

Fig. 10: Average number $\bar{r}$ of transmission attempts per instance (*preemptable* retransmissions).

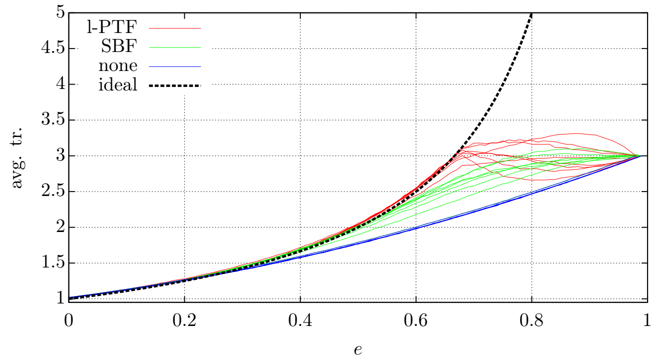

Fig. 11: Average number of $\bar{r}$ transmission attempts per instance (*consecutive* retransmissions).

should lie between $\bar{r}_P$ and $\min(\bar{r}_I, \bar{r}_R)$) increases towards the ideal case thanks to bandwidth recovery. In particular, until the transmission error probability e approaches the threshold $e_R = \frac{R}{1+R} \simeq 0.67$ (for which value $\bar{r}_I = \bar{r}_R = 3$), l-PTF manages to provide instances with almost the same amount of transmission opportunities as the ideal case. Since dynamic bandwidth management strategies exploit unused preallocated bandwidth, whenever conditions are such that there is no saved bandwidth, their advantages cannot grow further (this happens in very adverse channel conditions, see rightmost side of the plots where $e > e_R$).

Finally, we carried out additional simulations to evaluate performances when deadlines are shorter than periods. In particular, D_i values were selected equal to 95%, 85%, 75%, and 65% of T_i, for which the task set remains feasible under both considered retransmission strategies. As in the $D_i = T_i$ case, no first attempts or planned retries ever missed their deadlines during the simulation runs. The cumulative delivery success percentages DSP, collected for significant values of e ($e = \{0.2, 0.5, 0.7\}$) under different dynamic bandwidth management strategies in case of preemptable retransmissions, are shown in Tab. II. Results obtained with consecutive retransmissions are not dissimilar and are thus not reported.

As can be seen, performance improvements due to bandwidth reassignment are significant even when $D_i < T_i$. In general, l-PTF outperforms SBF, but differences between the two strategies tend to disappear when deadlines decrease and channel conditions are good (see, e.g., the case $D_i = 0.65 \cdot T_i$ for $e = 0.2$, which even shows a slight trend inversion). A fair performance comparison between SBF and l-PTF ultimately depends on several factors, including the considered time-critical data set, the underlying network architecture, as well as node computing resources, and cannot be generalized.

TABLE II: DSP with $D_i < T_i$ (*preemptable* retransmissions)

		Reallocation strategy	e		
			0.2	0.5	0.7
D_i	$0.95 \cdot T_i$	SBF	99.98	97.72	80.14
		l-PTF	99.99	99.43	89.80
		none	99.18	87.45	65.69
	$0.85 \cdot T_i$	SBF	99.98	97.73	80.11
		l-PTF	99.99	99.15	89.43
		none	99.18	87.45	65.69
	$0.75 \cdot T_i$	SBF	99.98	97.70	79.76
		l-PTF	99.98	98.76	87.37
		none	99.18	87.45	65.69
	$0.65 \cdot T_i$	SBF	99.98	97.76	79.57
		l-PTF	99.97	98.05	84.54
		none	99.18	87.46	65.71

VI. CONCLUSIONS

The intrinsic lack of determinism of wireless technologies makes them unsuitable to support time-critical data delivery in hard real-time IDCS. However, they can be profitably employed in soft real-time IDCS, provided that mechanisms are envisaged to guarantee that the impact of deadline misses is maintained below some application-dependent functional thresholds. The main contributions of this paper can be summarized as follows: 1) A framework has been defined which allows to combine timeliness and reliability on a per-instance basis, by a priori guaranteeing every instance of all periodic data streams to obtain, upon errors, a predefined number of retries within its deadline. Under realistic channel error hypotheses, this means that a certain probability of being successfully delivered on time is ensured to each data instance. The proposed framework grounds on real-time scheduling theory and relies on worst-case analysis, which in turn leads to subpar bandwidth utilization. 2) Admission control tests were derived, which provide above described guarantees under two practically-relevant retransmission strategies. 3) Two dynamic bandwidth reallocation strategies have been finally developed, which operate at runtime and prevent time reserved for planned retries, and actually left unemployed during system operation, to go wasted, significantly improving overall performance. Differently from other dynamic allocation strategies, which try to adapt to changing traffic/channel conditions by fairly sharing available communication resources among nodes, the presented strategies are completely aware of single deadline constraints and never impair provided per-instance guarantees.

It is worth pointing out that several techniques proposed in the literature and aimed at dealing with deadline-constrained traffic based on individual timing requirements can be included in our framework, as long as the retransmission strategy is properly selected (e.g., with RA it is preferable to rely on consecutive retransmissions) and parameters driving admission control and transmission scheduling ($\{C_{i,j}\}_{j \in \mathbb{N}^+}$) are defined accordingly. The two proposals that most closely resemble ours are [9] and [10]. The work in [9] mostly operates as the consecutive retransmission strategy, and therefore very similar performance is expected (when no dynamic bandwidth management is employed). Concerning [10], a quantitative comparison with our proposal is reported in [14]. Unlike these solutions, our framework has been conceived to ensure the most favorable admission control, yet assuring that a predefined number of retries are always available for each time-critical data instance within its deadline, independently of the actual conditions of the wireless channel. Moreover, suitable mechanisms that reassign the unused bandwidth at runtime increase overall network performance further, without impairing static guarantees. Simulation results show that, the percentage of time-critical data which are delivered on-time remains typically more than satisfactory, also when the channel transmission error probability increases significantly.

Clearly, centralized access schemes where a single coordinator polls every other node, like HCCA in Wi-Fi, are not as efficient as distributed protocols based on CSMA, like DCF. On the other hand, the latter are just unable to provide any guarantees on timings of data delivery. In IDCS, this behavior is usually not tolerated, since control algorithms must be ensured that communication quality never falls below a given threshold, so that the final goods could meet the intended production tolerances. For this reason, approaches like ours (which, when applied to Wi-Fi, can be seen as an enhanced version of HCCA, purposely tailored to cope with deadline-constrained traffic in soft real-time control systems) are not general purpose, but make sense to support, e.g., communication at the shop-floor in industrial plants.

As future research work we plan to introduce suitable heuristics aimed at improving behavior of dynamic bandwidth management strategies (e.g., with respect to fairness), as well as to develop a model to estimate performance enhancement they introduce at runtime in specific channel conditions.

REFERENCES

[1] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, May 2016.

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

[3] S. Vitturi, F. Tramarin, and L. Seno, "Industrial Wireless Networks: The Significance of Timeliness in Communication Systems," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 40–51, Jun 2013.

[4] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1265–1282, Dec. 2002.

[5] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 3rd ed. Springer Publishing Company, Incorporated, 2011.

[6] J. Baillieul and P. Antsaklis, "Control and Communication Challenges in Networked Real-Time Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, Jan 2007.

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

[8] M. Sojka, P. Píša, D. Faggioli, T. Cucinotta, F. Checconi, Z. Hanzálek, and G. Lipari, "Modular software architecture for flexible reservation mechanisms on heterogeneous resources," *Journal of Systems Architecture*, vol. 57, no. 4, pp. 366 – 382, 2011.

[9] M. Jonsson and K. Kunert, "Towards reliable wireless industrial communication with real-time guarantees," *IEEE Trans. Ind. Informat.*, vol. 5, no. 4, pp. 429–442, Nov 2009.

[10] E. Toscano and L. Lo Bello, "Bandwidth-Efficient Admission Control for EDF-Based Wireless Industrial Communication," in *IEEE Int. Symposium on Industrial Electronics (ISIE)*, June 2011, pp. 1186–1193.

[11] S. Girs, A. Willig, E. Uhlemann, and M. Björkman, "Scheduling for source relaying with packet aggregation in industrial wireless networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1855–1864, Oct 2016.

[12] G. Cena, S. Scanzio, and A. Valenzano, "Seamless link-level redundancy to improve reliability of industrial wi-fi networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 608–620, April 2016.

[13] S. Vitturi, L. Seno, F. Tramarin, and M. Bertocco, "On the Rate Adaptation Techniques of IEEE 802.11 Networks for Industrial Applications," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 198–208, Feb 2013.

[14] L. Seno, A. Valenzano, and C. Zunino, "A dynamic bandwidth reassignment technique for improving QoS in EDF-based industrial wireless networks," in *IEEE Int. Conference on Industrial Informatics (INDIN)*, July 2015, pp. 892–899.

[15] L. Seno, G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "Enhancing Communication Determinism in Wi-Fi Networks for Soft Real-Time Industrial Applications," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 866–876, April 2017.

[16] G. Cena, S. Scanzio, L. Seno, and A. Valenzano, "A soft real-time scheduling framework for wireless industrial sensor actuator networks," in *IEEE Int. Symposium on Industrial Embedded Systems (SIES)*, 2016.

[17] H. Bengtsson, E. Uhlemann, and P. A. Wiberg, "Protocol for wireless real-time systems," in *Euromicro Conference on Real-Time Systems*, 1999, pp. 168–174.

[18] E. Uhlemann, T. Aulin, L. K. Rasmussen, and P. A. Wiberg, "Concatenated hybrid ARQ - a flexible scheme for wireless real-time communication," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2002, pp. 35–44.

[19] F. Tramarin, S. Vitturi, and M. Luvisotto, "A Dynamic Rate Selection Algorithm for IEEE 802.11 Industrial Wireless LAN," *IEEE Trans. Ind. Informat.*, 2016, To appear.

[20] C. Schurgers, V. Raghunathan, and M. B. Srivastava, "Modulation scaling for real-time energy aware packet scheduling," in *Global Telecommunications Conference (GLOBECOM)*, vol. 6, 2001, pp. 3653–3657.

[21] M. Collotta, G. Pau, and G. Scatà, "Deadline-Aware Scheduling Perspectives in Industrial Wireless Networks: A Comparison between IEEE 802.15.4 and Bluetooth," *International Journal of Distributed Sensor Networks*, vol. 2013, no. 2, pp. 1–11, 2013.

[22] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, and Y. Chen, "Analysis of EDF scheduling for Wireless Sensor-Actuator Networks," in *IEEE Int. Symposium of Quality of Service (IWQoS)*, May 2014, pp. 31–40.

[23] G. Alderisi, S. Girs, L. L. Bello, E. Uhlemann, and M. Björkman, "Probabilistic scheduling and Adaptive Relaying for WirelessHART networks," in *IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, Sept 2015, pp. 1–4.

[24] G. Boggia, P. Camarda, L. Grieco, and S. Mascolo, "Feedback-based bandwidth allocation with call admission control for providing delay guarantees in IEEE 802.11e networks," *Computer Communications*, vol. 28, no. 3, pp. 325–337, 2005.

[25] G. Boggia, P. Camarda, L. A. Grieco, and S. Mascolo, "Feedback-Based Control for Providing Real-Time Services With the 802.11e MAC," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 323–333, April 2007.

[26] S. Punnekkat, "Schedulability analysis for fault tolerant real-time systems," Ph.D. dissertation, University of York, 1997.

[27] H. Aydin, "Exact fault-sensitive feasibility analysis of real-time tasks," *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1372–1386, Oct 2007.

[28] R. I. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller area network (can) schedulability analysis: refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, Apr. 2007.

[29] L. Palopoli, L. Abeni, T. Cucinotta, G. Lipari, and S. K. Baruah, "Weighted feedback reclaiming for multimedia applications," in *IEEE/ACM/IFIP Workshop on Embedded Systems for Real-Time Multimedia*, Oct 2008, pp. 121–126.

[30] T. P. Baker, "Stack-based scheduling for realtime processes," *Real-Time Systems*, vol. 3, no. 1, pp. 67–99, Apr. 1991.

[31] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[32] M. Spuri, "Analysis of Deadline Scheduled Real-Time Systems," INRIA, France, Tech. Rep., 1996.

[33] "OMNeT++, Discrete Event Simulator." [Online]. Available: http://omnetpp.org/

Lucia Seno (M'15) received the B.S. and M.S. degrees in automation Engineering and the Ph.D. degree in information Engineering from the University of Padova, Padova, Italy, in 2004, 2007, and 2011, respectively.

Since then, she has been with the Institute of Electronics, Information Engineering, and Telecommunications, National Research Council of Italy (CNR–IEIIT) in Padova, and, subsequently, in Torino, Italy, where she is currently a Researcher. Her research interests include industrial communication systems and technologies, wireless networks and wireless sensor networks for real-time and safety-critical control applications, formal verification of systems security and communication protocols, and model checking.

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 with the Department of Computer Engineering, Politecnico di Torino. Since 2005 he has been a Director of Research with the Institute of Electronics, Computer and Telecommunication Engineering of the National Research Council of Italy (CNR–IEIIT), Turin. His research interests include wired and wireless industrial communication systems, real-time protocols, and automotive networks. In these areas he has coauthored more than 130 technical papers and one international patent.

Dr. Cena served as a Program Co-Chairman for the 2006 and 2008 editions of the IEEE International Workshop on Factory Communication Systems, and as a Track Co-Chairman in six editions of the IEEE International Conference on Emerging Technologies and Factory Automation. Since 2009 he has been an Associate Editor of the IEEE Transactions on Industrial Informatics.

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 5th, 8th and 13th IEEE Workshops on Factory Communication Systems (WFCS 2004, WFCS 2010 and WFCS 2017). 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.

Claudio Zunino received the degree in computer engineering and the Ph.D. degree in software engineering from the Politecnico di Torino, Turin, Italy, in 2000 and 2005, respectively.

Since 2006, he has been a Researcher with the National Research Council of Italy (CNR). He is currently with the Institute of Electronics, Computer and Telecommunications Engineering (IEIIT). He has authored and coauthored several papers in international journals and conferences in the area of wireless communication and Industrial Ethernet protocols. He serves as reviewer for several international conferences and journals. He has also taken part in the program and organizing committees of many international conferences.