Probabilistic Per-Packet Real-Time Guarantees for Wireless Networked Sensing and Control

Yu Chen, Hongwei Zhang, Nathan Fisher, Le Yi Wang, George Yin

***Abstract*—The mission-critical nature of wireless networked sensing and control (WSC) systems such as the control of industrial plants requires stringent real-time delivery of packets. Due to inherent dynamics and uncertainties in wireless communication, real-time communication guarantees are probabilistic in nature. In this paper, a probabilistic framework is therefore proposed for *per-packet* real-time delivery guarantee. The notion of real-time in this paper differs from existing work in the sense that it ensures, in an execution history of arbitrary length, every packet is successfully delivered before its deadline with a probability no less than a user-specified threshold (e.g., 99%). The framework has several novel building blocks: 1) "R3 (Requirement-Reliability-Resource) mapping" translates the upper-layer probabilistic real-time communication *requirement* and the lower-layer link *reliability* into the *resource* (i.e., optimal number of transmission opportunities) reserved for each packet; 2) "EDF (Earliest Deadline First)-based real-time scheduling" as well as the "admission test" and "traffic load optimization" maximize system utility while satisfying per-packet real-time communication requirements. The proposed admission test is proved to be both sufficient and necessary, and the simulation results show that the proposed framework ensures probabilistic per-packet real-time communication.**

***Index Terms*—Probabilistic per-packet guarantees, reliability, real-time, EDF, schedulability, optimization.**

I. Introduction

Embedded wireless networks are increasingly being explored for mission-critical sensing and control in many domains. In real-time augmented vision, for instance, wireless networks can enable the fusion of real-time video streams from spatially distributed cameras to eliminate the line-of-sight constraint of natural human vision and thus enable seeing-through obstacles [7]. In industrial automation, wireless-enabled mobile, pervasive, and reconfigurable instrumentation and the significant cost of planning, installing, and maintaining wired network cables have made wireless networks attractive for industrial monitoring and control; industrial wireless networking standards such as WirelessHART, ISA100.11a, and WIA-PA have also been deployed in practice [8]. Machine-type communication for real-time sensing and control has also become a major focus of the emerging 5G wireless network research and development [14].

This work was supported in part by the National Science Foundation awards CNS-1054634 and CNS-1647200.

H. Zhang is with the Department of Electrical and Computer Engineering at Iowa State University, hongwei@iastate.edu.

Y. Chen, N. Fisher, L. Y. Wang, and G. Yin are with the Department of Computer Science, Computer Science, Electrical and Computer Engineering, and Math at Wayne State University respectively, {yu_chen,fishern,lywang,gyin}@wayne.edu.

Unlike traditional, best-effort wireless networks, reliable and real-time delivery of messages is critical in wireless networked sensing and control (WSC) systems [21]. In industrial WSC systems, for instance, the reliability and timeliness of data delivery directly impact the safety and optimality of sensing and control tasks. Since WSC differs dramatically from the traditional throughput-centric wireless networks due to its mission-critical nature, new design objectives and perspectives are necessary for offering real-time Quality of Service (QoS) in WSC systems.

Per-packet real-time communication guarantee. Wireless communication inherently introduces non-zero packet error probability due to factors such as the uncertainty of lossy links and the contention among different nodes/links, and it introduces non-zero delay due to shared wireless medium and the time taken to transmit each packet. There exist a few recent studies that consider real-time communication guarantees in wireless networks [5], [6], [17], [18], [37], [38], [39], as shown in Table I. Focusing on the long-term ratio of packets that are delivered before their deadlines, the existing frameworks do not ensure short-term timeliness. For example, consider the scenario shown in Table II. A sensor node periodically sends data to a controller. The controller makes control decision in each period based on the information it collects and requires that at least 90% of the packets are received in time. Otherwise, control failure or even catastrophe may happen. In both cases, the average timely delivery ratios are 90%. In the second case, however, period 2 and period 4 fail to achieve the required delivery ratio due to channel fading, thus the system is subject to transient instability.

In many real-time WSC systems, stringent per-packet real-time communication guarantee is important since it is needed for predictable system performance. For example, in industrial real-time augmented reality applications [22][23][24], real-time delivery of each packet enables seamless, naturalistic 3D reconstruction of real-world scenes (e.g., industrial processes), and consecutive packet loss (which may happen if we only ensure long-term timely delivery ratio) may well lead to uncomfortable human experience. In networked feedback control, consecutive packet loss may well lead to system instability and cause safety concerns; in addition, a packet of sample data will be dropped if the packet has not been successfully delivered by the time a new sample data is collected, and, in this case, the probabilistic guarantee of real-time delivery of each packet enables the modeling of the packet drop process as a random sampling process, thus facilitating the use of random sampling theories to characterize the impact

TABLE I
COMPARISON OF RECENT STUDIES THAT CONSIDER REAL-TIME COMMUNICATION GUARANTEES IN WIRELESS NETWORKS

Paper	Model	Algorithm	Applications
[5]	N wireless clients and one access point, homogeneous delay requirement.	Largest Debt First.	VoIP, Video streaming.
[6]	N wireless clients and one access point, heterogeneous delay requirement.	Pseudo-debt-based Scheduling.	VoIP, Video streaming.
[17]	A WirelessHART network consisting of field devices, one gateway, and a network manager.	Branch-and-Bound (Optimal); Conflict-aware Least Laxity First (Heuristic).	WirelessHART.
[18]	A wireless network consisting of a set of field devices, a gateway, and multiple access points.	Earliest Deadline First (EDF).	Wireless Sensor-Actuator Networks (WSANs).
[37]	An IEEE 802.15.4 network operating in beacon-enabled mode and with a star topology.	A distance constrained real-time off-line message scheduling algorithm.	Industrial Wireless Sensor Networks.
[38]	A typical infrastructure WLAN.	QoS-enabled 802.11e scheme.	Industrial networks.
[39]	An IEEE 802.11 network.	A TDMA data link layer protocol based on IEEE 802.11 physical layer to provide deterministic timing on packet delivery.	Wireless Cyber-Physical Control Applications.

TABLE II
TWO CASES WITH THE SAME LONG-TERM AVERAGE TIMELY DELIVERY RATIO BUT DIFFERENT SHORT-TERM PERFORMANCE.

Case	Period 1	Period 2	Period 3	Period 4
1	90%	90%	90%	90%
2	99%	79%	99%	83%

of probabilistic wireless communication on networked control [27][28][29][30][31][32] and facilitating the joint design of wireless networking and networked control as well as the development of field-deployable WSC systems.

Another feature of most WSC systems is that traffic periods (i.e., inter-packet intervals) are allowed to vary within a certain range as long as such changes do not affect critical sensing and control functions [35], [36]. So there exist the challenge and opportunity of selecting the optimal traffic periods for optimal sensing and control performance while ensuring that the required real-time communication requirements are satisfied.

Contributions. For ensuring probabilistic real-time communication guarantee of each packet as required by real-time WSC systems, this work proposes a probabilistic real-time wireless communication framework that effectively integrates real-time scheduling theory with wireless communication. Major contributions are as follows:

- The per-packet probabilistic real-time QoS is formally modeled. To the best of our knowledge, this paper is the first to propose such notion of timeliness. Existing formulations and designs do not ensure the stringent per-packet assurance as required by WSC systems, since improving the average performance of the network in the long-term does not guarantee short-term QoS.
- R3 mapping is proposed to translate the upper-layer requirement and the lower-layer link reliability into the optimal number of transmission opportunities reserved for each packet.
- An optimal real-time scheduling algorithm and a sufficient and necessary admission test are proposed and formally proved .
- Algorithms are proposed to select packet generation periods to maximize system utility.

Organization of the rest of paper. Section II summarizes related work. In Section III, we present the system model and problem statement. In Section IV, a probabilistic per-packet real-time communication framework is proposed and an optimization problem is formulated and solved. Experimental results are presented in Section V. Section VI addresses the impact of the timescales of real-time communication, multi-channel settings, control signaling unreliability, and distributed implementations. Finally, Section VII concludes the paper.

II. RELATED WORK

A. Real-time scheduling

In traditional real-time systems such as computer processors, the most commonly used scheduling policies are priority-driven, including fixed-priority and dynamic-priority schemes. A fixed-priority algorithm assigns the fixed/same priority to all jobs (i.e., instances of a task) in each task. For example, the Rate Monotonic algorithm [11] assigns priority based on the rate (or period): the shorter the period of a task, the higher its priority. Another example is the Deadline Monotonic algorithm [12], which assigns a higher priority to a job with a shorter relative deadline. In contrast, dynamic-priority scheduling algorithms assign priorities dynamically. The Earliest Deadline First (EDF) algorithm [11] is a preemptive priority-based dynamic scheduling algorithm. The scheduler always schedules the active task with the earliest absolute deadline.

Time-Sensitive-Networking (TSN) [8] has recently been proposed to enable real-time communication in industrial networks. Focusing on extending the IEEE 802.1-standards-based Ethernet networks, TSN-oriented work examines issues in real-time wired communication instead of real-time wireless communication.

Real-time scheduling for wireless network has drawn increasing interest recently. By leveraging real-time scheduling theory for processors and incorporating unique wireless characteristics, a Conflict-aware Least Laxity First algorithm was proposed for WirelessHART [17]. Analysis for EDF in wireless sensor-actuator networks was investigated [18]. The key insight from these work is to draw an analogy between wireless channel and a processor such that the classic real-time theories can be leveraged. However, they did not reveal the inherent relationship between the application requirement, link reliability, and resource reservation. They also did not ensure short-term probabilistic real-time packet delivery. There also exist other research efforts that consider real-time communication guarantees in wireless networks [5][6]. Modeling the delivery ratio of packets that meet their deadlines as a long-term, asymptotic average over multiple (and infinitely many) transmission periods, the proposed frameworks in those studies do not assure short-term timeliness either. IEEE 802.15.4 or 802.11 based real-time protocols have also been investigated for mission-critical industrial systems. A real-time message-scheduling algorithm was proposed for IEEE 802.15.4-based industrial WSNs [37]. An analysis of the real-time performance that can be achieved in QoS-enabled 802.11 networks has been carried out [38]. RT-WiFi is a TDMA data link layer protocol based on 802.11 physical layer to provide deterministic timing guarantee on packet delivery in wireless control systems [39]. Again, these methods have not been designed for applications that need per-packet real-time guarantees. Probabilistic real-time scheduling methods have also been proposed for WirelessHART [1] and fault-tolerant EDF scheduling [2], but they have not considered the real-time wireless scheduling problems studied in this paper either.

B. Reliability-oriented wireless transmission scheduling

Providing predictable link reliability is critical in WSC systems. To guarantee that packets are delivered at requested reliability levels, Zhang et al. proposed PRKS for wireless networked control [4]. PRKS is a TDMA-based distributed protocol to enable predictable link reliability based on the Physical-Ratio-K (PRK) interference model [3]. Through a control-theoretic approach, PRKS instantiates the PRK model parameters according to in-situ network and environment conditions so that each link meets its reliability requirement. In particular, PRKS defines a conflict graph for a given wireless network: a node in the graph denotes a link with data transfer in the network, and a link exists between two nodes in the graph if the corresponding links in the network interfere with each other according to the PRK model. Under the condition that link reliability is ensured, PRKS schedules as many nodes as possible in the conflict graph. Based on the predictable link reliability as ensured by algorithms such as PRKS, this paper develops the framework and algorithm for ensuring predictable per-packet real-time communication.

III. System model and problem statement

This section presents the system models and real-time communication scheduling problem considered in this paper.

A. Communication model

Similar to [19][20], the system consists of n wireless sensors and one controller, i.e., access point (AP), as shown in Figure 1. All the sensors are within the communication range of the controller. Time is slotted and synchronized across the controller and sensors, and the wireless transmissions between the sensors and controller are scheduled in a TDMA manner.

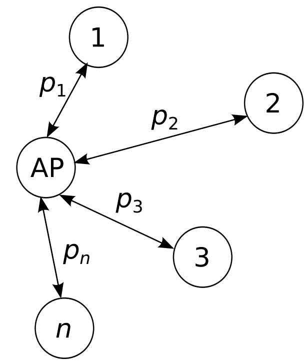

Fig. 1. System model: one AP and n sensors.

For simplicity of presentation, our discussion focuses on a centralized master-slave architecture where the scheduler lies in the AP. At the beginning of each time slot, the AP broadcasts a short control signaling message to indicate the sensor that shall transmit in the current time slot. For simplicity of presentation, we assume that the control signaling is reliable (e.g., through transmission power control). (We will discuss how to address potential unreliability in control signaling in Section VI-C, and we will discuss potential distributed approaches in Section VI-D.) The duration of a time slot is the time required for the controller to send the control packet plus the time for a sensor to transmit a data packet of certain fixed length. All the time slots are of the same length. To avoid transmission collision, there can be at most one active sensor transmitting at each time slot.

The link reliability between a sensor i and the controller is p_i, meaning that a packet transmission succeeds in probability p_i. Similar to [13], the event of successful transmissions along a link is assumed to be i.i.d. over time, thus whether a packet can be successfully delivered at a time slot is independent of the status of earlier transmissions. Note that the communication along a wireless link may well be i.i.d. in practice when mechanisms such as transmission power control is used to ensure a certain communication reliability at each time instant based on in-situ wireless channel conditions. In addition, as we will discuss in more detail in Section IV-B, our methodologies of probabilistic real-time scheduling also applies to other wireless link models such as Markovian models and links with deep fading.

B. Probabilistic QoS model

The traffic on link i is characterized by a 4-tuple $(T_i, T_{i,max}, D_i, P_i)$:

- Period T_i: sensor i generates one data packet every T_i time slots. Wireless networked control can usually tolerate a flexible traffic period, provided the period is chosen below an upper bound $T_{i,max}$, since a traffic period too large may not generate sufficient packets for the control application. In a very general framework, T_i is the design variable to represent the possible choices of the designer at the stage of the system design.
- Relative deadline D_i: for client i, each packet is associated with a relative deadline D_i in units of time slots.

A packet arriving at time slot t should be successfully delivered no later than time slot $t + D_i$; otherwise, the packet is dropped. Since the node always sends new sensing and control signal, $D_i \leq T_i$. In this paper, D_i and T_i are assumed to be equal, and this scenario is usually referred to as *implicit deadline* in classic real-time systems and finds its application in various areas [15]. Note that heterogeneous deadlines across different links are considered since more stringent deadlines are assigned to more critical sensor nodes in practice.

- Application requirement P_i: due to inherent dynamics and uncertainties in wireless communication, real-time communication guarantees are probabilistic in nature. Thus a probabilistic QoS metric is defined.

  Definition 1 Link i meets probabilistic per-packet real-time guarantee if $\forall j$, $Prob\{F_{ij} \leq D_i\} \geq P_i$, where F_{ij} is the delay (measured in the number of time slots) in successfully delivering the j-th packet of link i.

  Intuitively, the probabilistic per-packet real-time guarantee for link i ensures that in an *arbitrarily* long execution history, a *randomly* selected packet of link i will meet the deadline D_i at least in probability P_i.

Remarks. Considering TDMA scheduling in this work, the parameters T_i, $T_{i,max}$, and D_i are all integers for each link i. The proposed probabilistic QoS model differs fundamentally from the classic soft-real-time model where the QoS metric is the long-term timely delivery ratio (i.e., the long-term ratio of the number of packets successfully delivered in time to the total number of packets transmitted).

C. Problem statement

Based on the system model, an important question to ask is as follows. Given a set of n links with each link i having a link reliability p_i and periodic traffic $(T_i, T_{i,max}, D_i, P_i)$, where $D_i = T_i$ and T_i can vary in a range with an upper bound $T_{i,max}$, are the set of links schedulable to meet the probabilistic per-packet real-time communication requirements? If yes, find a set of optimal periods such that the system utility (to be defined precisedly in Section IV-E) is maximized while ensuring real-time schedulability; If no, indicate the infeasibility.

To answer this question, a probabilistic real-time wireless communication framework is proposed in the next section.

IV. Wireless Scheduling Framework for Probabilistic Per-Packet Real-Time Communication Guarantee

A. Overview

Compared with scenarios where every packet transmission is successful, probabilistic communication errors in wireless systems lead to increased delays in delivering packets to their destinations and thus introduce additional deadline violations and packet drops. This additional packet drop may well cause non-negligible degradation in the quality of data delivery. Thus, the decision regarding whether a set of links is schedulable must take into consideration the inherent nature of probabilistic wireless communication.

For example, consider a simple case where there are two links and each link generates one packet in a period of 3 time slots. If there are no channel errors (as assumed in classic real-time systems), the required delay guarantee can be provided to every packet of every link (Figure 2(a)). In the presence of

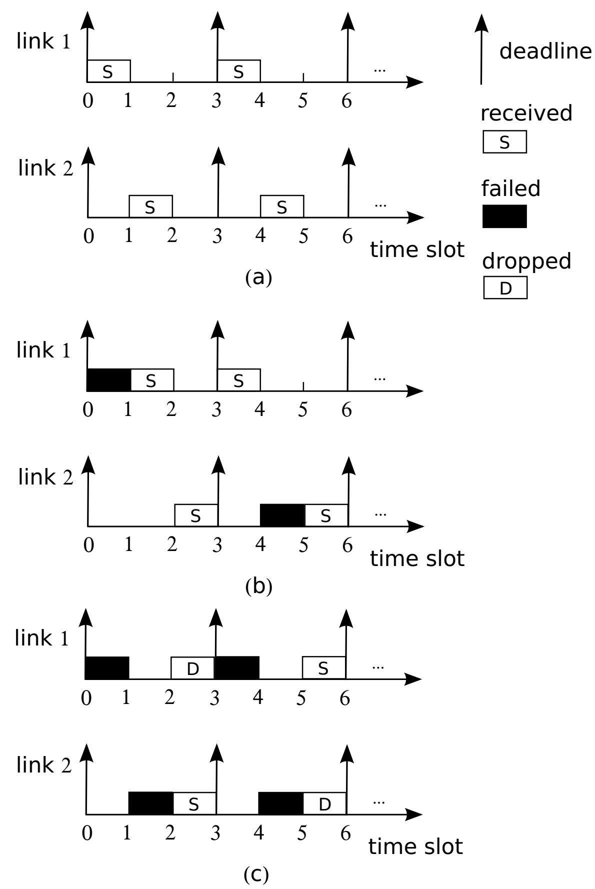

Fig. 2. An example showing the uncertainty of wireless channel.

slight channel errors (Figure 2(b)), the delay experienced by each packet is increased by at most one time slot, but both links can still deliver their packets on time. Figure 2(c) shows the case when the links are not schedulable. At the third time slot, both links need to send out the packet since this is the last slot in current period. However, only one can send, and the other has to drop its packet. Thus packet drop happens.

For a packet that needs to be successfully delivered across a link i within deadline D_i and in probability no less than P_i, the requirement can be decomposed into two sub-requirements: 1) successfully delivering the packet in probability no less than P_i, and 2) the time taken to successfully deliver the packet is no more than D_i if it is successfully delivered. Given a specific link reliability p_i, the first sub-requirement translates into the required minimum number of transmission opportunities that need to be provided to the transmission of the packet. Then, the second sub-requirement requires that these minimum number of transmission opportunities are used within deadline D_i. That is, the original problem of ensuring probabilistic per-packet real-time delivery is decomposed into two sub-problems of ensuring probabilistic delivery reliability and real-time packet transmission respectively.

With the above observation, an architecture for solving the probabilistic per-packet real-time scheduling problem of Section III-C is shown in Figure 3. Given a set of links with

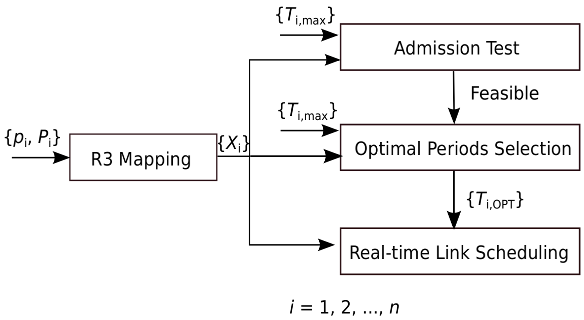

Fig. 3. Architecture of the proposed framework.

each link i having link reliability p_i, application requirement P_i, and traffic $(T_i, T_{i,max}, D_i)$, the "R3 Mapping" component uses p_i and P_i of link i to compute X_i, i.e., the number of time slots (i.e., transmission opportunities) needed to ensure the successful delivery of a packet along link i in probability P_i. Then the "Admission Test" component determines whether the links are schedulable or not based on X_i and the input $T_{i,max}$. If schedulable, a set of optimal periods $T_{i,OPT}$ are computed based on X_i, T_i, and $T_{i,max}$. Then the links are scheduled using an optimal real-time scheduling algorithm based on the optimal traffic periods, the transmission opportunities reserved, and delivery deadlines. In the following, we elaborate on each component of the framework.

B. Requirement-reliability-resource (R3) mapping

R3 mapping is invoked to compute the number of transmission opportunities needed to ensure successful delivery a packet along each link with the required probability guarantee.
Theorem 1 Given link reliability p_i and application requirement P_i, the minimum number of time slots (or transmission opportunities) needed to ensure successful delivery of a packet for link i in probability P_i, denoted as X_i, is as follows:

$$X_i = \lceil log_{1-p_i}(1 - P_i) \rceil. \tag{1}$$

Proof. Let

$$\begin{aligned} Q_i &= Prob\{\text{A packet delivered within } X_i \text{ transmissions}\} \\ &= 1 - Prob\{\text{Fail all } X_i \text{ transmissions}\} \\ &= 1 - \underbrace{(1 - p_i)(1 - p_i)...(1 - p_i)}_{X_i} \\ &= 1 - (1 - p_i)^{X_i}. \end{aligned} \tag{2}$$

To achieve the application requirement P_i,

$$Q_i = 1 - (1 - p_i)^{X_i} \geq P_i. \tag{3}$$

Thus, the minimum number of transmissions needed is $X_i = \lceil log_{1-p_i}(1 - P_i) \rceil$.

■

Note that the X_i reserved time slots do not have to be X_i consecutive time slots, and, for real-time packet delivery, the X_i time slots only have to be before the delivery deadline of the packet. By R3 mapping, the number of transmission opportunities needed to ensure probabilistic packet delivery success (i.e., X_i) is derived from the channel statistics and the application requirement, thus transforming the probabilistic real-time delivery requirement into a problem involving the reservation of a deterministic number of transmission opportunities for each link. This transformation facilitates the application/extension of traditional real-time CPU scheduling theory to the real-time wireless transmission scheduling as we will explain in Sections IV-C and IV-D.

The above derivation of X_i is based on the link model where the per-packet transmission success probability along link i is p_i, which is a valid model for cases where the per-packet transmission reliability is controlled at a certain level (e.g., p_i) using techniques such as channel-aware transmission power control. For cases where other models such as Markovian models or empirical models of packet transmission successes [33][34] may be more appropriate (e.g., when no mechanism is adopted to ensure a certain per-packet communication reliability in the presence of channel deep fading), the derivation of X_i needs to be revised accordingly, but the rest of our probabilistic real-time scheduling framework would still apply. As a first step in studying scheduling with probabilistic per-packet real-time guarantees, we focus on scenarios where the per-packet transmission reliability is controlled at p_i, and we relegate the detailed studies of alternative link models as future work.

C. Scheduling policy

Priority-based scheduling algorithms are usually used in real-time systems [11][12]. Consider the better resource utilization and agility that a dynamic-priority algorithm can achieve, in this paper, a dynamic-priority scheduling algorithm based on the Earliest Deadline First (EDF) concept is used. In the algorithm, we define the deadline of a link as the deadline of the packet that needs to be delivered across the link; if there is no packet that is yet to be delivered across the link at a time slot, we regard the link's deadline as positive infinity at the time slot. Then, at each time slot, the link with the earliest deadline is scheduled by the algorithm. This can be represented by Algorithm 1.

Algorithm 1 Scheduling algorithm

```
for each time slot
    target_link_index = 0;
    min_deadline = ∞;
    for i = 1 to n
        if link[i].deadline < min_deadline
            min_deadline = link[i].deadline;
            target_link_index = i;
    Schedule link[target_link_index];
```

In the following, the proposed scheduling algorithm is shown to be feasibility-optimal.
Theorem 2 (Feasibility Optimality of Algorithm 1) Algorithm 1 can produce a feasible schedule for a set of links L with arbitrary deadlines D_i on a single wireless channel if

and only if there exists any algorithm that can schedule the transmissions of links L to ensure the required probabilistic per-packet real-time guarantee.

Proof. The theorem is proved by transforming the wireless packet transmission scheduling problem into a real-time CPU scheduling problem [15] in the following manner: treat each link i as a task i; then each packet j along link i corresponds to a job j of task i, and the number of transmission opportunities X_i required to ensure the probabilistically successful delivery of a packet corresponds to the execution time C_i in task scheduling. Accordingly, the traffic demand $\{(T_i, T_{i,max}, D_i, P_i)\}_{i=1...n}$ can be transformed into a task system $\mathcal{J} = \{(T_i, D_i, X_i)\}_{i=1...n}$ of n periodic tasks, with each task i specified by its period T_i, relative deadline D_i, and execution time $X_i = \lceil log_{1-p_i}(1 - P_i)\rceil$. In a task i $(i = 1 \dots n)$, a job is repeated every T_i time slots, and every instance of the repeating job has a relative deadline of D_i and an execution time of X_i. Then, Algorithm 1 is the same as the EDF algorithm for preemptive CPU scheduling except for the following differences: 1) the task system $\mathcal{J}$ is such that the units of time is a time slot, and the period of a task as well as the arrival time, relative deadline, and execution time of each job of the task are all integers (instead of being real numbers as in a preemptive CPU scheduling problem setup); 2) given that the transmission of a packet in a time slot cannot be interrupted in general, the transmission along a link i can only preempt the transmission along another link i' at the boundary of two consecutive time slots, thus a task in the corresponding task system can be preempted only at the boundary of consecutive time slots (instead of being at any arbitrary point in time as in a preemptive CPU scheduling problem setup), and the task system is a limited preemptive system. Just as the EDF scheduling algorithm is optimal for preemptive single CPU scheduling, Algorithm 1 is optimal for the aforementioned task system $\mathcal{J} = \{(T_i, D_i, X_i)\}_{i=1...n}$, and the proof is based on induction as follows.

Let Δ denote an arbitrarily small positive number of time slots. Consider a schedule S for $\mathcal{J}$ in which all deadlines are met, and let $[t_0, t_0 + \Delta)$ denote the first time interval over which this schedule makes a scheduling decision different from the one made by Algorithm 1. Suppose that a job/packet $j_1 = (a_1, e_1, d_1)$ with arrival time a_1, execution time e_1, and deadline d_1 is scheduled in S over this interval, while another job $j_2 = (a_2, e_2, d_2)$ is scheduled in Algorithm 1. Since S meets all the deadlines, it is the case that S schedules j_2 to completion by the end of time slot d_2. In particular, this implies that S schedules j_2 for an amount at least Δ prior to d_2. But by the definition of Algorithm 1, $d_2 \le d_1$; hence, S schedules j_2 for an amount at least Δ prior to d_1 as well. Now the new schedule S' obtained from S by swapping the executions of j_1 and j_2 of length Δ time slots each would agree with Algorithm 1 over $[0, t_0+\Delta)$. The proof of optimality of Algorithm 1 now follows by induction on time, with S' playing the role of S in the above argument.

■

D. Admission policy

Given an arbitrary set of links, it is not always possible to find a schedule to meet the probabilistic deadlines. Therefore, an admission policy is needed to determine whether the set of links is feasible. In this paper, a reservation-based admission policy is used. Since link i generates one packet every T_i time slots, by R3 mapping, it is known that it needs to reserve X_i time slots to ensure the probabilistic per-packet real-time delivery. Thus the admission test is such that if the test is passed, each link is guaranteed for a certain number of reserved transmission oppotunities.

Definition 2 (Work density) The work density of link i, denoted by ρ_i, is defined as the ratio of the number of transmission opportunities needed to ensure the required probabilistic real-time communication guarantee along link i to the traffic period of link i. That is,

$$\rho_i = \frac{X_i}{T_i}. \tag{4}$$

Then, we have

Theorem 3 A set of n links is schedulable if and only if

$$\sum_{i=1}^{n} \rho_i \le 1. \tag{5}$$

Proof Sketch. As we have shown in the proof of Theorem 2 earlier, the real-time wireless transmission scheduling problem considered in this work can be transformed into a special preemptive CPU scheduling problem, and the EDF-based Algorithm 1 is optimal just as the EDF algorithm is optimal for real-time preemptive single CPU scheduling. For real-time preemptive single CPU scheduling, Liu [15] has shown, by Theorem 6.1 of [15], that a task system is schedulable if and only if the total work density of the task system is no more than 1. The proof of Theorem 6.1 on pages 124-126 of Reference [15] can be directly applied to prove Theorem 3 here, since the proof of Theorem 6.1 in Reference [15] applies to the special task system $\mathcal{J}$ presented in the proof of Theorem 2 earlier. For conciseness of presentation, we skip the detailed proof here.

■

Since the upper bound of period for each link is known, besides Theorem 3, a simple infeasibility test can be applied to avoid further searching by the following lemma.

Lemma 1 Given a set of n links, if the schedulability condition from Theorem 3 is not satisfied for $T_{i,max}$, $i = 1, .., n$, then it is not satisfied for any $T_i < T_{i,max}$.

Proof. If the link set fails the schedulability test in Theorem 3, then

$$\sum_{i=1}^{n} \frac{X_i}{T_{i,max}} > 1.$$

Using any $T_i < T_{i,max}$ will lead to

$$\textstyle\sum_{i=1}^{n} \frac{X_i}{T_i} > \sum_{i=1}^{n} \frac{X_i}{T_{i,max}} > 1.$$

This concludes the proof. ■

Then the admission test is summarized as Algorithm 2.

Algorithm 2 Admission test

$total\_density \leftarrow 0$;
for $i = 1$ to n
  Compute $X_i = \lceil log_{1-p_i}(1 - P_i) \rceil$;
  $total\_density \leftarrow total\_density + \frac{X_i}{T_{i,max}}$;
  if $total\_density > 1$
    return Infeasible;
return Feasible;

E. Selection of packet generation period

In this section, we address the issue of selecting the optimal packet generation periods while ensuring schedulability for a set of feasible links. Without loss of generality, assume the utility function is concave and monotonically increasing. Since T_i is an integer (i.e., number of time slots), the optimization problem to solve becomes a discrete optimization problem. More precisely, this problem is:

$$\underset{T_i, i=1\dots n}{\text{Maximize}} \quad \sum_{i=1}^{n} U_i\Big(\frac{1}{T_i}\Big), \tag{6}$$

subject to:

$$\sum_{i=1}^{n} \frac{X_i}{T_i} \le 1, \tag{7}$$

$$T_i \le T_{i,max}, \quad i = 1, ..., n. \tag{8}$$

The maximization is with respect to T_i's. The utility function U_i increases as the period T_i decreases, which is quite intuitive. The optimal solution for this problem can be obtained using Branch-and-Bound methods, especially when the problem size is small. Given that the time complexity of branch-and-bound methods is as high as that of exhaustive search in the worst case (i.e., $O({T_{i,max}}^n)$), however, here we also turn to pursue a more efficient approximation algorithm for problems of larger sizes. The basic idea is to transform the problem by replacing the discrete variable T_i with real number t_i, then solve the new problem using convex optimization technique. We first transform the problem as

$$\underset{t_i, i=1\dots n}{\text{Maximize}} \quad \sum_{i=1}^{n} U_i\Big(\frac{1}{t_i}\Big), \tag{9}$$

subject to:

$$\sum_{i=1}^{n} \frac{X_i}{t_i} \le 1, \tag{10}$$

$$t_i \le T_{i,max}, \quad i = 1, ..., n. \tag{11}$$

For ease of mathematical derivation, replace $\frac{1}{t_i}$ above with f_i, where f_i is the sensing frequency. Then the optimization problem can be rewritten as:

$$\underset{f_i, i=1\dots n}{\text{Maximize}} \quad \sum_{i=1}^{n} U_i(f_i), \tag{12}$$

subject to:

$$\sum_{i=1}^{n} X_i f_i \le 1, \tag{13}$$

$$f_{i,min} \le f_i, \quad i = 1, ..., n, \tag{14}$$

where $f_{i,min} = \frac{1}{T_{i,max}}$ is the lower bound of the sensing frequency. By transforming E.q. (10) into E.q. (13), we perform derivation over f_i rather than t_i when apply the KKT conditions.

Introducing Lagrange multipliers λ for the inequality constraints $\sum_{i=1}^{n} X_i f_i \le 1$, and λ_i for the inequality constraints $f_{i,min} \le f_i$, it follows that

$$I(f_1, ..., f_n) = \sum_{i=1}^{n} U_i(f_i) - \lambda(\sum_{i=1}^{n}(X_i f_i - 1)) - \sum_{i=1}^{n}(\lambda_i(f_{i,min} - f_i)).$$

Then the KKT conditions can be obtained as

$$\begin{cases} \nabla_f I(f_1, ..., f_n) = 0, \\ \sum_{i=1}^{n} X_i f_i - 1 \le 0, \\ f_{i,min} - f_i \le 0, \quad i = 1, ..., n, \\ \lambda(\sum_{i=1}^{n} X_i f_i - 1) = 0, \\ \lambda_i(f_{i,min} - f_i) = 0, \quad i = 1, ..., n, \\ \lambda \ge 0, \\ \lambda_i \ge 0, \quad i = 1, ..., n. \end{cases}$$

The first condition can be further expanded into n equations:

$$\nabla_f I(f_1, ..., f_n) = \begin{bmatrix} \frac{\partial I(f_1, ..., f_n)}{\partial f_1} \\ \frac{\partial I(f_1, ..., f_n)}{\partial f_2} \\ \vdots \\ \frac{\partial I(f_1, ..., f_n)}{\partial f_n} \end{bmatrix} = \begin{bmatrix} \frac{d}{df_1} U_1(f_1) - \lambda X_1 + \lambda_1 \\ \frac{d}{df_2} U_2(f_2) - \lambda X_2 + \lambda_2 \\ \vdots \\ \frac{d}{df_n} U_3(f_n) - \lambda X_n + \lambda_n \end{bmatrix} = 0.$$

For simplicity, rewrite the above equations as

$$\begin{cases} \frac{d}{df_i} U_i(f_i) - \lambda X_i + \lambda_i = 0, \quad i = 1, ..., n, & \text{(15a)} \\ \sum_{i=1}^{n} X_i f_i - 1 \le 0, & \text{(15b)} \\ f_{i,min} - f_i \le 0, \quad i = 1, ..., n, & \text{(15c)} \\ \lambda(\sum_{i=1}^{n} X_i f_i - 1) = 0, & \text{(15d)} \\ \lambda_i(f_{i,min} - f_i) = 0, \quad i = 1, ..., n, & \text{(15e)} \\ \lambda \ge 0, & \text{(15f)} \\ \lambda_i \ge 0, \quad i = 1, ..., n. & \text{(15g)} \end{cases}$$

Rewrite E.q. (15a) as $\lambda = \frac{\frac{d}{df_i} U_i(f_i) + \lambda_i}{X_i}$. Since the utility function $U_i(f_i)$ is monotonically increasing with f_i, $\frac{d}{df_i} U(f_i) > 0$. In addition, $X_i > 0$ and $\lambda_i \ge 0$, thus $\lambda > 0$. Then, E.q. (15d) implies that $\sum_{i=1}^{n} X_i f_i = 1$. In this case, the problem becomes:

$$\underset{f_i, i=1\dots n}{\text{Maximize}} \quad \sum_{i=1}^{n} U_i(f_i), \tag{16}$$

subject to:

$$\sum_{i=1}^{n} X_i f_i = 1, \tag{17}$$

$$f_{i,min} \leq f_i, \quad i = 1, ..., n. \tag{18}$$

Thus, the problem to be addressed contains only the equality and lower bound constraints. For many simple utility functions, closed form solutions can be obtained; in general, a set of optimal periods $t_{i,OPT}$ can be obtained in time $O(nlogn)$ by using the OPT-L algorithm [16]. Then we can approximate the optimal period $T_{i,OPT}$ by taking ceiling on $\frac{1}{f_{i,OPT}}$ for $1 \leq i \leq n$, that is, letting $T_{i,OPT} = \lceil \frac{1}{f_{i,OPT}} \rceil$.

Algorithm 3 summarizes our approach to selecting the packet generation periods.

Algorithm 3 Selection of packet generation period

```
if problem size is small (e.g., n ≤ 30)
    Use branch-and-bound method to solve (6)-(8).
else
    Use the OPT-L algorithm [16] to solve (16)-(18);
    T_{i,OPT} = ⌈1/f_{i,OPT}⌉, 1 ≤ i ≤ n.
```

V. Simulation Experiment

The proposed admission control and scheduling algorithm are denoted as prob-QoS. In this section, the performance for prob-QoS is evaluated via Matlab simulation. We first discuss the simulation methodology and then show the results.

A. Simulation methodology

- **Packet loss model**. The packet reception is modeled with Bernoulli trials, i.e., a transmission will succeed with probability p, where $0 \leq p \leq 1$.
- **Performance metrics**. Regarding the performance metrics, *QoS satisfaction ratio* and *idleness ratio* are used. First, QoS satisfaction ratio is the ratio that the required per-packet QoS is guaranteed. It is the primary goal of this work. Second, since the number of time slots needed X_i is estimated based on the channel statistics, it is possible that a slot assigned to a link at certain time is not used since it has already successfully delivered the packet. In this case, the remaining slots reserved for this link will be idle. The idleness ratio is the ratio between the idle slots and the total slots reserved. In the premise of QoS satisfaction, a smaller idleness ratio is favored since it implies higher channel utilization.
- **Application settings**. For reflecting the mission-critical scenarios, consider the cases when the application requirement P_i for each link is set to 90%, 95%, or 99% respectively. To understand the impact of link reliability p_i, experiments with varying p_i are also performed. The number of links n is 16, period T and deadline D are 100 time slots. This setting is to ensure there is feasible solution for the admission test.
- **Baseline algorithm**. To evaluate the per-packet real-time guarantee, the proposed scheme is compared with the state-of-the-art debt-based algorithm [5]. The debt of a client i at time t is defined as the number of time slots needed by i minus the actual number of time slots it has transmitted by t. At each time slot, the AP schedules the client with largest debt.

B. Simulation results

Figure 4 compares the QoS satisfaction ratios of prob-QoS and debt-based algorithm when the link reliability $p = 0.6$. As shown in the figure, prob-QoS can always guarantee per-packet QoS for all requirements, while the debt-based algorithm fails all cases. The reason is because prob-QoS reserves enough resource for each packet by R3 mapping and admission test, while the debt-based algorithm only focuses on long-term average performance and ignores short-term fluctuations

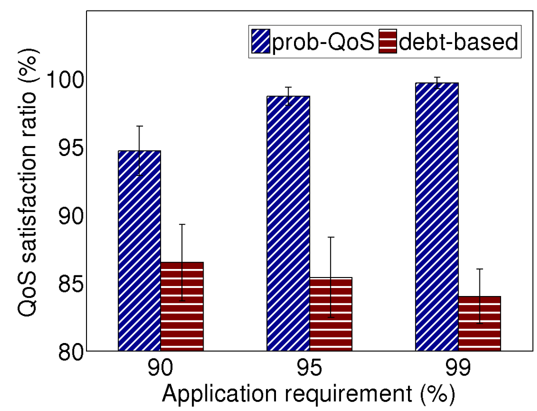

Fig. 4. QoS satisfaction ratio: prob-QoS vs debt-based algorithm with different application requirements

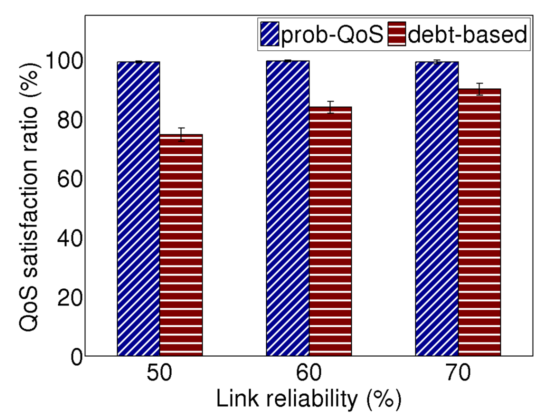

Fig. 5. QoS satisfaction ratio: prob-QoS vs debt-based algorithm with different link reliability

Figure 5 evaluates the QoS satisfaction ratio by fixing the application requirement to 99% and varying the link reliability. As shown in the figure, prob-QoS always ensures 99% QoS satisfaction ratio while the debt-based algorithm fails to.

For the idleness ratio, similar experiments are conducted. Figure 6 shows the case when the link reliability is 0.6 for both algorithms. As shown in the figure, prob-QoS has a larger idleness ratio than the debt-based algorithm. This is because the primary design objective of prob-QoS is to ensure per-packet real-time QoS. By Definition 1, such per-packet QoS must be satisfied for a *randomly* picked packet in an *arbitrarily* long execution history. Mission-critical applications require predictable behaviors under all possible circumstances, thus the conservativeness of reservation is the cost that needs to be taken, otherwise the required QoS cannot be met. Figure 7 shows similar results when the application requirement is set to 99% and link reliability varies. In Section VI-A, the inherent tradeoff between throughput and real-time guarantees will be discussed in detail.

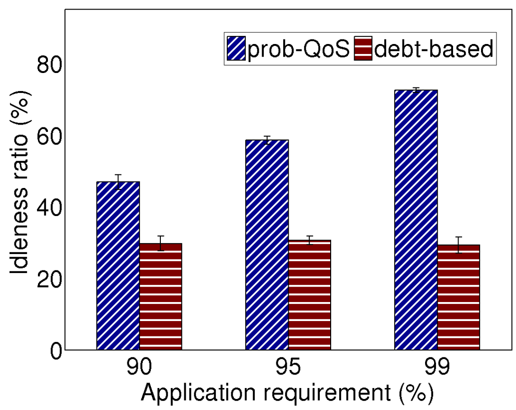

Fig. 6. Idleness ratio: prob-QoS vs debt-based algorithm with different application requirements

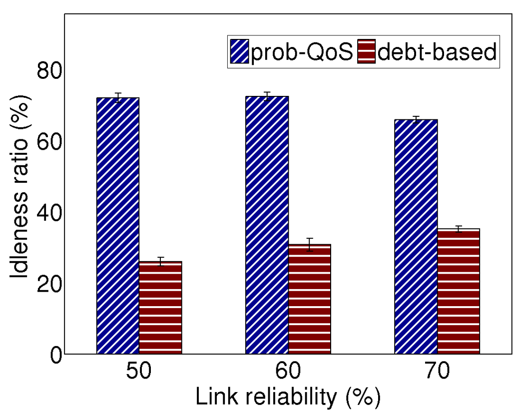

Fig. 7. Idleness ratio: prob-QoS vs debt-based algorithm with different link reliability

VI. Extensions

The previous sections have focused on centralized schemes to ensure per-packet real-time communication guarantees in single-channel settings. In this section, we discuss the impact of the timescale of real-time communication guarantees on system throughput, how to leverage multiple wireless channels, how to address the impact of control signaling unreliability, and how to implement the proposed algorithms in a distributed manner.

A. Impact of timescales of real-time communication guarantee

Targeting applications such as industrial control and augmented reality which are sensitive to the delay and loss of every packet (or every few consecutive packets), the studies in this paper have mainly focused on how to ensure probabilistic short-term, per-packet communication delay bound. Existing studies such as those by Hou et al. [5], [6], on the other hand, have focused on how to ensure a certain long-term, asymptotic average ratio of packets that are delivered before deadlines, instead of ensuring probabilistic real-time delivery of each packet. Therefore, a question worth asking is how the timescales (e.g., short-term vs. long-term) of real-time communication guarantees impact network behavior such as throughput and per-packet delivery reliability.

To answer the above question, here we examine the question of how to schedule the transmissions of packets so that, for each link i, out of every "window" of K_i consecutive packets, the probability $\mathbb{P}_i(m_i, p_i, K_i, r_i)$ for r_i fraction of the K_i packets to be delivered before deadlines is no less than q_i. In this case, K_i is the "timescale of real-time communication guarantee", and here we intend to understand the impact of K_i on network behavior such as throughput. To facilitate the analysis, we assume that, for each link i, each period has T_i time slots, and, as before, the per-transmission reliability along link i is p_i. Figure 8 shows the setup for the analysis.

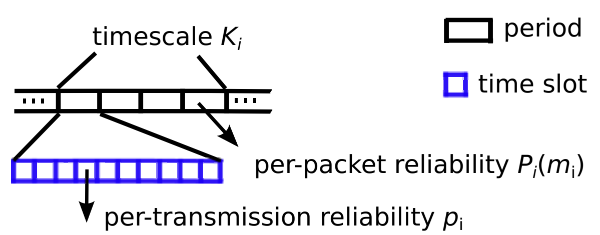

Fig. 8. Timescale analysis

(Note that, as special cases of the analytical setup, the problem considered in the previous sections of this paper is such that $K_i = 1$, $r_i = 1$, and $q_i = P_i$, and the problems considered by Hou et al. [5], [6] are such that $K_i = \infty$.)

Given that, with techniques such as power control, the per-transmission reliability can be controlled to be $i.i.d.$, the number of transmission opportunities reserved for each packet is assumed to be the same and denoted by m_i. That is, each packet is transmitted until success or up to m_i times in each period, and $1 \le m_i \le T_i$. Let $P_i(m_i) \triangleq Prob\{\text{Per-packet delivery success}\}$, then

$$\begin{aligned} P_i(m_i) &= Prob\{\text{A packet delivered within } m_i \text{ transmissions}\} \\ &= 1 - Prob\{\text{Fail all } m_i \text{ transmissions}\} \\ &= 1 - \underbrace{(1-p_i)(1-p_i)...(1-p_i)}_{m_i} \\ &= 1 - (1-p_i)^{m_i}. \end{aligned} \tag{19}$$

Let $\mathbb{P}_i(m_i, p_i, K_i, r_i)$ be the probability that the real-time delivery ratio along link i is at least r_i for every K_i consecutive packets, then

$$\begin{aligned}
&\mathbb{P}_i(m_i, p_i, K_i, r_i)\\
&= Prob\{\text{At least } \lceil K_i r_i \rceil \text{ packets out of every } K_i \text{ packets}\\
&\qquad \text{are delivered before deadlines}\}\\
&= \sum_{K_i' = \lceil K_i r_i \rceil}^{K_i} \text{Prob } \{K_i' \text{ packets out of every } K_i \text{ packets}\\
&\qquad\qquad \text{are delivered before deadlines}\}\\
&= \sum_{K_i' = \lceil K_i r_i \rceil}^{K_i} \binom{K_i}{K_i'} \cdot P_i(m_i)^{K_i'} \cdot (1 - P_i(m_i))^{K_i - K_i'}.
\end{aligned} \tag{20}$$

Given that $P_i(m_i)$ and $\mathbb{P}_i(m_i, p_i, K_i, r_i)$ are non-decreasing with m_i, the transmission scheduling scheme should be such that the minimum number of transmission opportunities are reserved for each packet to ensure that $\mathbb{P}_i(m_i, p_i, K_i, r_i) \geq q_i$, if possible. When p_i and T_i are relative small while r_i and q_i are large, it may be infeasible to ensure $\mathbb{P}_i(m_i, p_i, K_i, r_i) \geq q_i$ even with $m_i = T_i$, and the problem is infeasible in this case. For understanding the impact of the timescales of real-time communication guarantee, it is enough for the analysis in this section to focus on cases where the problem is feasible. With this assumption, the transmission scheduling scheme reserves $m_i^*(K_i, p_i, r_i, q_i)$ number of transmission opportunities for each packet, with m_i^* specified as follows:

$$m_i^*(K_i, p_i, r_i, q_i) = \underset{m_i}{\text{argmin}} \ \mathbb{P}_i(m_i, p_i, K_i, r_i) \geq q_i. \tag{21}$$

It is difficult to derive the close-form, analytical solution for computing $m_i^*(K_i, p_i, r_i, q_i)$, but, given that $\mathbb{P}_i(m_i, p_i, K_i, r_i)$ is non-decreasing with m_i, $m_i^*(K_i, p_i, r_i, q_i)$ can be found by numerically searching from 1, 2, ... until the first m_i such that $\mathbb{P}_i(m_i, p_i, K_i, r_i) \geq q_i$.

To understand the impact of the timescales of real-time communication guarantee (i.e., K_i), Figures 9 and 10 show the impact of K_i on m_i^* for different parameter configurations. The figures show that the overall trend is such that m_i^* decreases as K_i increases, even though there exist transient perturbations (i.e., m_i^* increases with K_i) embedded into the overall trend. The overall trend is clearer when the per-transmission reliability p_i is lower, and m_i^* becomes less sensitive to K_i as the per-transmission reliability p_i increases. Given that the supportable network throughput increases with decreasing m_i^*, the observations show that, when the per-transmission reliability is low (e.g., 0.1), the supportable network throughput increases in a non-negligible manner with increasing K_i. On the other hand, as the per-transmission reliability increases, the supportable network throughput becomes less sensitive to K_i. (Similar phenomena have been observed for other parameter configurations, but details are omitted here due to space constraint.)

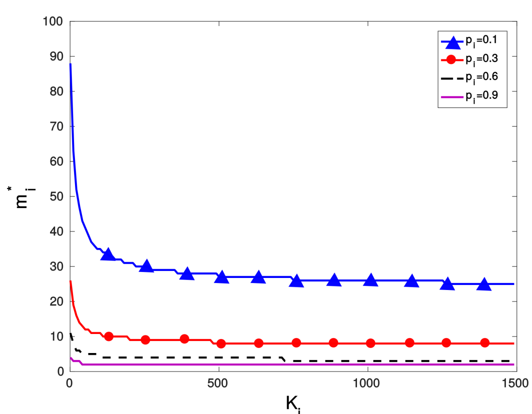

Fig. 9. An example showing the relation between K_i and m_i^*, where r_i = 0.9, q_i = 0.9999, and T_i = 100.

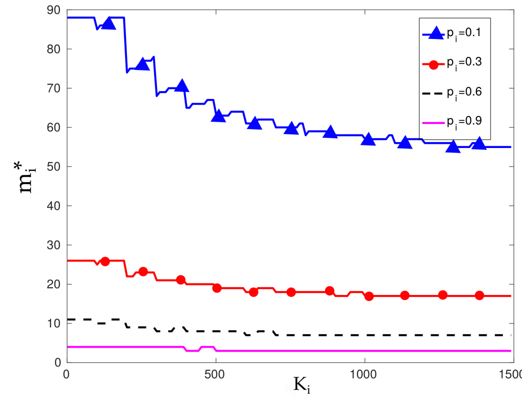

Fig. 10. An example showing the relation between K_i and m_i^*, where r_i = 0.99, q_i = 0.9999, and T_i = 100.

For a given m_i^* and p_i, there exists a corresponding per-packet success probability P_i^*. Note that, for a given K_i, r_i, and q_i, P_i^* is the same no matter what p_i is; that is, p_i impacts the choice of m_i^* to ensure the required P_i^*. Figures 11 and 12 show the relation between P_i^* and K_i for the same parameter configurations as those for Figures 9 and 10 respectively. The figures show that, as K_i increases, P_i^* tends to decrease as an overall trend (despite transient perturbations) and converges to r_i as K_i goes to infinity. When K_i is small, P_i^* tends to be higher than r_i to ensure $\mathbb{P}_i(m_i^*, p_i, K_i, r_i) \geq q_i$.

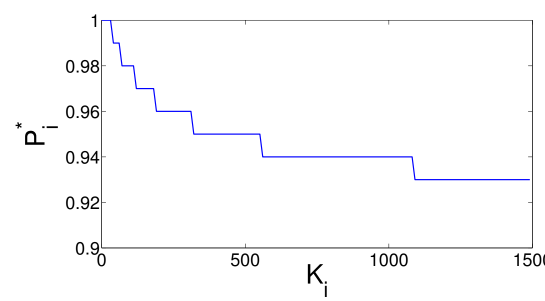

Fig. 11. An example showing the relation between K_i and P_i^*, where r_i = 0.9, q_i = 0.9999.

B. Extension to multichannel settings

In the previous sections, single channel settings are considered. When multiple channels are available for packet transmissions, the proposed scheme can be extended. The multichannel wireless scheduling problem is similar to the multiprocessor real-time scheduling, with each available channel treated as a processor.

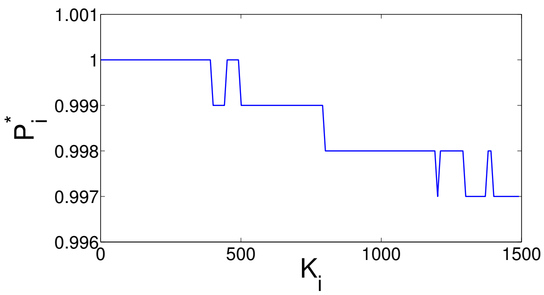

Fig. 12. An example showing the relation between K_i and P_i, where r_i = 0.99, q_i = 0.9999.

In multiprocessor real-time scheduling, different tasks are assigned to different processors. At any moment in time, there is only one active task on each processor while multiple tasks are running in parallel on different processors. Similarly, in multichannel wireless scheduling, transmission links are assigned to different channels, and within each channel, one link is scheduled to transmit at a time.

Then, by multi-core real-time scheduling theory [25], the P-Fair scheduling algorithm [25] is the optimal, capacity-achieving scheduling algorithm, and the real-time communication requirements of the n mutually-interfering links (see Section III-C) can be satisfied if the following condition holds:

$$\sum_{i=1}^{n} \rho_i \leq m, \tag{22}$$

where ρ_i is the work density of link i as defined by (4), and m is the number of available channels. For the usual case when $D_i = T_i$ for every link i (i.e., packet delivery deadline equals packet generation interval), Condition (22) is also a necessary condition.

C. Impact of control signaling unreliability

The presentations in this paper so far have assumed that the delivery of control signaling packets along the downlinks from the AP to sensors is perfectly reliable. When the downlink reliability is less than 1, the loss of a control signal will lead to the situation where no sensor transmits in the corresponding time slot, which can be treated as a packet loss. To deal with this in our framework, we just need to reflect the downlink unreliability in the "adjusted" uplink reliability from each sensor to AP. That is, if p_i^d is the downlink reliability from the AP to sensor i, the "adjusted" uplink reliability becomes $p_i^d * p_i$ for link i, and then the rest of the framework applies.

D. Distributed solution

Although a centralized network with a base station is considered in this paper, it is possible to extend the solution to a distributed scenario. Assume an *ad hoc* network with multiple sender-receiver pairs. Since the traffic pattern is deterministic for each link (i.e., the traffic period and deadline are known), such information can be shared among links through information exchange before transmissions. As a consequence, each link is aware of the packet arrival and deadline requirement of other interfering links, and then a distributed EDF scheduler can run at each link synchronously [26].

VII. Concluding remarks

In this paper, a probabilistic framework for per-packet real-time wireless communication guarantees is proposed. The notion of *real-time* in this paper differs fundamentally from existing work in the sense that it ensures in an *arbitrarily* long execution history, a *randomly* selected packet will meet its deadline in a user-specified probability. By R3 mapping, the upper-layer requirement and the lower-layer link reliability are translated into the number of transmission opportunities needed. By optimal real-time communication scheduling as well as admission test and traffic period optimization, the system utilization is maximized while the schedulability is maintained. The proposed admission test is proved to be both sufficient and necessary. The simulation results show that the proposed scheme can meet the probabilistic real-time communication requirement.

For future work, we plan to integrate the proposed probabilistic real-time framework with PRKS [4], where the link reliability is guaranteed by PRKS and the timeliness is provided by the proposed framework in this paper. Another interesting direction would be to consider aperiodic traffic coexisting with real-time traffic, in which case the channel utilization can potentially be improved by allowing the aperiodic traffic to utilize the time slots that are left idle by the periodic, real-time traffic. We have assumed that the per-packet transmission status (i.e., success or failure) is independent across different links in this work, and a future direction is to consider spatial link correlation and diversity in real-time scheduling which may increase system real-time capacity (i.e., amount of data deliverable within certain deadlines). Detailed study of alternative link models (e.g., when the communication reliability across a link is not i.i.d.) and distributed implementation of the proposed scheduling algorithm is also an interesting future direction. This study has focused on link-level scheduling, and an interesting future extension will be to consider end-to-end real-time guarantees which address the delay introduced by computational tasks and multi-hop communications.

References

[1] G. Alderisi, S. Girs, L. Lo Bello, E. Uhlemann, and M. Bjrkman. Probabilistic scheduling and Adaptive Relaying for WirelessHART networks. *IEEE ETFA*, 2015
[2] M. Short and J. Proenza. Towards efficient probabilistic scheduling guarantees for real-time systems subject to random errors and random bursts of errors. *IEEE ECRTS*, 2013
[3] H. Zhang, X. Che, X. Liu, and X. Ju. Adaptive instantiation of the protocol interference model in wireless networked sensing and control. *ACM Transactions on Sensor Networks*, 10(2), 2014.
[4] H. Zhang, X. Liu, C. Li, Y. Chen, X. Che, F. Lin, L. Y. Wang, and G. Yin. Scheduling with predictable link reliability for wireless networked control. In *IEEE IWQoS*, 2015.
[5] I. Hou, V. Borkar, and P. R. Kumar. A theory of QoS for Wireless. In *INFOCOM*, 2009.
[6] I. Hou, and P. R. Kumar. Scheduling heterogeneous real-time traffic over fading wireless channels. In *INFOCOM*, 2010.
[7] A. Gosain, M. Berman, M. Brinn, T. Mitchell, C. Li, Y. Wang, H. Jin, J. Hua, and H. Zhang. Enabling campus edge computing using GENI racks and mobile resources. In *ACM/IEEE SEC*, 2016.

[8] R. Zurawski (Editor). Industrial Communication Technology Handbook, CRC Press, 2015.

[9] S. Boyd, and L. Vandenberghe. Convex Optimization, Cambridge University Press New York, NY, USA, 2004.

[10] Y. Wu, G. Buttazzo, E. Bini, and A. Cervin. Parameter selection for real-time controllers in resource-constrained systems. *IEEE Transactions on Industrial Informatics*, 6(4), pp. 610-620, 2010.

[11] C. L. Liu, and J. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 20(1), 1973.

[12] J. Leung, and J. W. Whitehead. On the complexity of fixed priority scheduling of periodic real-time tasks. *Performance Evaluation*, 2(4), 1982.

[13] X. Zhang, and M. Haenggi. Delay-optimal power control policies. *IEEE Transactions on Wireless Communications*, 11(10), 2012.

[14] H. Shariatmadari et al. Machine-type communications: current status and future perspectives toward 5G systems. *IEEE Communications Magazine*, 53(9), 2015.

[15] J. Liu. Real-time systems, Prentice Hall PTR, 2000.

[16] H. Aydin, R. Melhem, and D. Mosse. Optimal reward-based scheduling of periodic real-time tasks. In *RTSS*, 1999.

[17] A. Saifullah, Y. Xu, C. Lu, and Y. Chen. Real-time scheduling for WirelessHART networks. In *RTSS*, 2010.

[18] A. Saifullah et al. Analysis of EDF scheduling for wireless sensor-actuator networks. In *IWQoS*, 2014.

[19] H. J. Korber, H. Wattar, and G. Scholl. Modular wireless real-time sensor/actuator network for factory automation applications. *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 111-119, May 2007.

[20] 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.

[21] J. Heo, J. Hong, and Y. Cho. EARQ: energy aware routing for real time and reliable communication in wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3-11, Feb. 2009.

[22] S. K. Ong, M. L. Yuan, and A. Y. C. NEE. Augmented reality applications in manufacturing: a survey. *International journal of production research*, vol. 46, no. 10, pp. 2707-2742, 2008.

[23] S. Henderson and S. Feiner. Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE transactions on visualization and computer graphics*, vol. 17, no. 10, pp. 1355-1368, 2011.

[24] F. Lamberti et al. Challenges, opportunities, and future trends of emerging techniques for augmented reality-based maintenance. *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 4, pp. 411-421, 2014.

[25] S. Baruah, and M. Bertogna, and G. Buttazzo. Multiprocessor Scheduling for Real-Time Systems. *Springer*, 2015.

[26] T. L. Crenshaw, A. Tirumala, and S. Hoke. A robust implicit access protocol for real-time wireless collaboration. In *ECRTS*, 2005.

[27] G. Yin and C. Tan and L.Y. Wang and C.Z. Xu. Recursive estimation algorithms for power controls of wireless communication networks. *Journal of Control Theory and Applications*, vol. 6, pp. 225-232, 2008.

[28] L. Xu and L.Y. Wang and G. Yin and H.W. Zhang. Communication information structures and contents for enhanced safety of highway vehicle platoons. *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 4206-4220, 2015.

[29] L.Y. Wang and A. Syed and G. Yin and A. Pandya and H. Zhang. Coordinated Vehicle Platoon Control: Weighted and Constrained Consensus and Communication Network Topologies. In *Proc. 51st Conference on Decision and Control*, 2012.

[30] L.Y. Wang and W. Feng and G. Yin. Joint state and event observers for linear switching systems under irregular sampling. *Automatica*, vol. 49, pp. 894-905, 2013.

[31] F. Wu and G. Yin and L.Y. Wang. Stability of a pure random delay system with two-time-scale Markovian switching. *J. Different. Eqs.*, vol. 253, pp. 878-905, 2012.

[32] G. Yin and Y. Sun and L.Y. Wang. Asymptotic properties of consensus-type algorithms for networked systems with regime-switching topologies. *Automatica*, vol. 47, pp. 1366-1378, 2011.

[33] W. Jiao, M. Sheng, K. S. Lui, and Y. Shi. End-to-end delay distribution analysis for stochastic admission control in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 13(3), 2014.

[34] Q. Liu, S. Zhou, and G. B. Giannakis. Cross-layer combing of adaptive modulation and coding with truncated ARQ over wireless links. *IEEE Transactions on Wireless Communications*, 3(5), 2004.

[35] D. Seto, J. P. Lehoczky, and L. Sha. Task period selection and schedulability in real-time systems. In *RTSS*, 1998.

[36] E. Bini and M. D. Natale. Optimal task rate selection in fixed priority system. In *RTSS*, 2005.

[37] S. Yoo et al. Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4 Star Topolgy. *IEEE Transactions on Industrial Electronics*, 57(11), 2010.

[38] 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*, 6(3), 2010.

[39] Y. H. Wei. et al. RT-WiFi: Real-Time High-Speed Communication Protocol for Wireless Cyber-Physical Control Applications. In *RTSS*, 2013.

Yu Chen received his B.S. and M.S. degrees in wireless communications from the University of Electronic Science and Technology of China in 2007 and 2011, respectively. He is currently a Ph.D. candidate in the Department of Computer Science at Wayne State University. His research focuses on wireless networked control and cyber-physical systems.

Hongwei Zhang (S01-M07-SM13) is an associate professor of electrical and computer engineering at Iowa State University, U.S.A. His research interests lie in the modeling, algorithmic, and systems issues in trustworthy wireless networking for real-time cyber-physical-human systems such as those in mixed-reality, agriculture, connected and automated vehicles, and industrial automation. He is a recipient of the NSF CAREER Award. (URL: http://www.ece.iastate.edu/~hongwei).

Nathan Fisher is an Associate Professor in the Department of Computer Science at Wayne State University. His research interests are in real-time and embedded computer systems, sustainable computing, resource allocation, game-theory, and approximation algorithms. His current funded research projects are on composability of real-time applications, multiprocessor real-time scheduling theory, thermal-aware real-time system design, and algorithmic mechanism design in competitive real-time systems. He received the NSF CAREER Award in 2010 and best paper awards from publication venues such as RTSS, ECRTS, and IEEE Transactions on Industrial Informatics.

Le Yi Wang (S'85-M'89-SM'01-F'12) received the Ph.D. degree in electrical engineering from McGill University, Montreal, Canada, in 1990. Since 1990, he has been with Wayne State University, where he is currently a Professor in the Department of Electrical and Computer Engineering. His research interests are in the areas of complexity and information, system identification, robust control, information processing and learning, as well as medical, automotive, communications, power systems, and computer applications of control methodologies. He is a Fellow of IEEE.

George Yin received the B.S. degree in Mathematics from the University of Delaware in 1983, M.S. degree in Electrical Engineering, and Ph.D. in Applied Mathematics from Brown University in 1987. He joined Wayne State University in 1987, and became Professor in 1996, and university Distinguished Professor in 2017. His research interests include stochastic systems and applications. He is a Fellow of IEEE, a Fellow of IFAC, and a Fellow of SIAM.