Scheduling with Probabilistic Per-Packet Real-Time Guarantee for Industrial URLLC

Zhibo Meng, Hongwei Zhang, James Gross

***Abstract*—For ultra-reliable, low-latency communications (URLLC) applications in industrial wireless networks such as mission-critical industrial control, it is important to ensure the communication quality of individual packets. Prior studies have considered Probabilistic Per-packet Real-time Communications (PPRC) guarantees for single-cell, single-channel networks but they have not considered real-world complexities such as inter-cell interference in large-scale networks with multiple communication channels and heterogeneous real-time requirements. To fill the gap, we propose a real-time scheduling algorithm based on *local-deadline-partition (LDP)*, and the LDP algorithm ensures PPRC guarantee for large-scale, multi-channel networks with heterogeneous real-time constraints. We numerically study the properties of the LDP algorithm and observe that it significantly improves the network capacity of URLLC, for instance, by a factor of 5-20 as compared with a typical method. Furthermore, the PPRC traffic supportable by the LDP algorithm is significantly higher than that of state-of-the-art comparison schemes. This demonstrates the potential of fine-grained scheduling algorithms for URLLC wireless systems regarding interference scenarios.**

***Index Terms*— Industrial wireless networks, URLLC, probabilistic per-packet real-time communications (PPRC) guarantee.**

I. Introduction

Industrial ultra-reliable, low-latency communications (URLLC) as enabled by 5G-and-beyond technologies is expected to play a pivotal role in enhancing the performance, flexibility, and robustness of industrial cyber-physical systems (CPS). Typically, industrial CPS focuses on meeting stringent deadlines in tasks such as sensing, robotic control, process control, and power-grid management [1]. Unlike traditional, best-effort wireless networks designed for high-throughput applications, reliable and real-time delivery of individual packets is critical for sensing and control in these URLLC industrial applications. For example, in Extended Reality (XR) applications [2], real-time delivery of each packet enables seamless, naturalistic 3D reconstruction of real-world scenes (e.g., industrial processes), and consecutive packet loss (or long-delay in packet delivery) may well lead to uncomfortable human experience [2], [3]. In networked industrial control, consecutive packet loss may well lead to system instability and negatively impact system safety.

Per-packet real-time communication guarantee in multi-cell industrial wireless networks. Wireless communications inherently introduce non-zero delay due to shared wireless medium and the time taken to transmit each packet [3]. In the literature, different approaches consider real-time communication guarantees in wireless networks. Primarily, resource allocation, which involves controlling available radio resources (time, frequency, and power), is a pivotal consideration. This aspect is explored as both an optimization problem [4], [5] and a machine learning problem [6], [7], [8]. Additionally, real-time communications in wireless systems have been extensively investigated using scheduling algorithms such as earliest-deadline-first (EDF) and rate-monotonic (RM) [3], [9], [10], [11], [12]. Another line of research proposes centralized real-time communication solutions that explicitly address the distinctions between traditional real-time systems and wireless real-time communications [13], [14], [15]. Simultaneously, there are studies focusing on providing long-term real-time guarantees, such as mean delay and age-of-information (AoI) for wireless systems [16], [17], [18], [19]. Despite these valuable contributions, certain limitations persist, including the inability to guarantee both reliability and latency, inapplicability to multi-cell settings, and practical challenges in implementation in distributed, multi-cell scenarios. Addressing these limitations is still crucial for advancing the effectiveness and applicability of real-time communications solutions in industrial wireless networks.

Addressing interference is also important for large-scale industrial wireless networks. In many envisioned industrial URLLC applications such as those for industrial process control, factory automation, and power grids, the network is expected to be deployed across a large area to provide industrial URLLC services to a larger number of nodes. While interference coordination is the traditional standard solution for such deployments, the effectiveness of interference coordination in industrial URLLC scenarios is little explored to date [20], [21], [22]. Crucially, existing interference coordination solutions for industrial URLLC cannot provide per-packet real-time guarantees.

Contributions. In this work, we tackle the challenge of per-packet real-time guarantees under inter-cell interference constraints in industrial wireless networks from a novel perspective. Central to our work are real-time scheduling algorithms with provable characteristics that ensure Probabilistic Per-packet Real-time Communications (PPRC) guarantee in large-scale, multi-cell, and multi-channel settings. We show that corresponding scheduling algorithms can simultaneously provide real-time guarantees while effectively mitigating in-

This work is supported in part by the NSF awards 2212573, 2229654, 2232461, 2112606, 2130889, and NIFA award 2021-67021-33775.

Zhibo Meng and Hongwei Zhang are with Iowa State University, U.S.A. E-mail:{zhibom,hongwei}@iastate.edu.

James Gross is with KTH Royal Institute of Technology, Sweden. Email: jamesgr@kth.se.

terference in multi-cell deployments. Our main contributions are as follows:

- Expanding upon the concept of deadline partitioning (DP) within real-time computing systems, our approach involves implementing DP in a distributed manner. In this context, we introduce the concept of a local deadline partition, which captures local traffic demand and local work density. Notably, these metrics are derived exclusively from the information available on one-hop links within the conflict graph.
- With the local deadline partition defined, we develop a distributed scheduling algorithm based on local-deadline-partition (LDP). To the best of our knowledge, the LDP scheduling algorithm is the first distributed URLLC scheduling algorithm that ensures PPRC guarantee in large-scale, multi-cell, and multi-channel networks with heterogeneous real-time requirements.
- We numerically study the properties of the LDP algorithm and observe that the intra-cell and inter-cell interference coordination via the LDP scheduling algorithm can significantly increase the capacity of URLLC, for instance, by a factor of 5-20 as compared with the typical industry practice today that only considers intra-cell interference. We also observe that the PPRC traffic supportable by the LDP algorithm is significantly higher than that of the state-of-the-art algorithms G-schedule [14] and WirelessHART-based algorithm (WH) [9]. For instance, the LDP algorithm can support the PPRC requirement of a large network 32.25% and 18.41% of whose links cannot be supported by G-schedule and WH respectively.

The rest of the paper is organized as follows. We present the system model and problem definition in Section II, present the LDP real-time scheduling algorithm in Section III, evaluate the properties of the LDP algorithm in Section IV, and make concluding remarks in Section V.

II. System Model and Problem Definition

A. Network model

The network consists of m base stations (BSes) and n user equipment (UEs). The links between BSes and UEs are called cellular links, and the links between UEs are called device-to-device (D2D) links. The corresponding wireless network can be modeled as a *network graph* $G = (V, E)$, where V is the set of nodes (i.e., the union of the BSes and UEs) and E is the set of wireless links. The edge set E consists of pairs of nodes which are within the communication range of each other. The network has access to N non-overlapping frequency channels, denoted by RB. Time is slotted and synchronized across the transmitters and receivers. Wireless transmissions are scheduled along frequency and time, with each transmission taking place in a specific frequency channel and time slot. All the time slots are of the same length, and a transmitter can complete the transmission of one packet within a time slot.

B. Interference model

For PPRC guarantee in industrial URLLC applications, it is important to control interference among concurrent transmissions so that a certain link reliability p_i is guaranteed for each link i. To this end, Zhang et al. [23] have identified the *Physical-Ratio-K (PRK) interference model that defines pair-wise interference relations between close-by nodes only while ensuring communication reliability (i.e., receiver-side SINR) in the presence of background noise and real-world wireless complexities such as multi-path fading and cumulative interference from all concurrent transmitters in the network.* In the PRK model [23] as shown in Figure 1, a node C' is regarded as not interfering and thus can transmit concurrently with the transmission from another node S to its receiver R in the same frequency band if and only if $P(C', R) < \frac{P(S,R)}{K_{S,R,T_{S,R}}}$, where $P(C', R)$ and $P(S, R)$ is the average strength of signals reaching R from C' and S respectively, $K_{S,R,T_{S,R}}$ is the minimum rational number chosen such that the probability for R to successfully receive packets from S is no less than a minimum link reliability $T_{S,R}$.

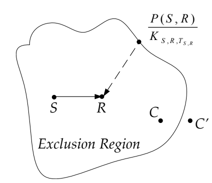

Fig. 1: Physical-Ratio-K (PRK) interference model

To ensure predictable communication reliability, the PRK interference model is used in this paper to provide the conflict set information for each link. In particular, a *conflict graph* G_c = (V_c, E_c) is defined for the network G, where each node in V_c represents a unique communication link in the network G, and $(i, j) \in E_c$ if links i and j interfere with each other, that is, if the transmitter of link i (link j) is in the exclusion region of link j (link i). Given a link i, we let M_i denote the set of links interfering with i, that is, $M_i = \{j : (i, j) \in E_c\}$. As an example, Figure 2 shows a conflict graph with 8 nodes, where each node represents a link in the network G. Taking link 1 as an example, $M_1 = \{2, 3, 4, 5\}$. Based on the conflict graph, if one link is active, then none of its interfering links can be active at the same time and frequency. In this way, the mean link reliability of the active links is ensured in the presence of background noise, path loss, fading, and cumulative interference from all concurrent transmitters in the network (including the interference from the links beyond the two-hop neighbors of i in G_c). Based on predictable link reliability enabled by the PRK model, this paper studies how to ensure predictable per-packet real-time communications in multi-cell, multi-channel settings. Therefore, we assume that the link packet delivery reliability for each link i is ensured

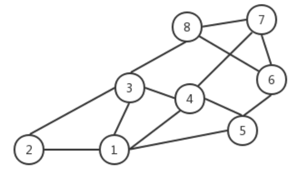

Fig. 2: Example conflict graph G_c (Note: this example will be used to illustrate other concepts in the rest of the paper too.)

and denoted by p_i.

C. PPRC traffic model

To support industrial URLLC applications with heterogeneous real-time requirements, we characterize the PPRC data traffic along each link i by a 3-tuple (T_i, D_i, S_i):

- Period T_i: the transmitter of link i generates one data packet every T_i time slots.
- Relative deadline D_i: each packet along link i is associated with a relative deadline D_i in units of time slots, and $D_i \leq T_i$.
- PPRC requirement S_i: due to inherent dynamics and uncertainties in wireless communication, real-time communication guarantees are probabilistic in nature. We adopt the following concept of PPRC guarantee first proposed by Chen el al. [3]:

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

For a packet that needs to be successfully delivered across a link i within deadline D_i and in probability no less than S_i, the requirement can be decomposed into two sub-requirements: 1) successfully delivering the packet in probability no less than S_i, and 2) the time taken to successfully deliver the packet is no more than D_i if it is successfully delivered [3]. Given a specific link reliability p_i, the first sub-requirement translates into the required minimum number of *transmission opportunities*, denoted as X_i, that need to be provided to the transmission of the packet, and $X_i = \lceil \log_{1-p_i}(1 - S_i) \rceil$ [3]. Then, the second sub-requirement requires that these X_i transmission opportunities are used within deadline D_i. Accordingly, the probabilistic real-time delivery requirement for a packet along link i is transformed into a problem of reserving a deterministic number of transmission opportunities, i.e., X_i, before the associated relative deadline D_i, and X_i is similar to the job execution time in classical real-time scheduling theory. Using X_i, we define the *work density* of link i as $\rho_i = \frac{X_i}{D_i}$.

D. PPRC scheduling problem

Based on the aforementioned system model, the PPRC scheduling problem is as follows: Given a network $G = (V, E)$ where each link i has a link reliability p_i and PPRC data traffic (T_i, D_i, S_i) $(D_i \leq T_i)$, develop an algorithm that schedules the data traffic to satisfy the PPRC requirements.

III. LDP SCHEDULING WITH PROBABILISTIC PER-PACKET REAL-TIME GUARANTEE

A. Overview

For single-channel wireless networks with implicit deadlines (i.e., packet delivery deadlines being equal to inter-packet-generation intervals), Chen et al. [3] have shown that an earliest-deadline-first (EDF) scheduling algorithm is optimal for ensuring probabilistic per-packet real-time guarantee. However, just as how EDF scheduling is not optimal in multi-processor systems, EDF-based scheduling is not expected to perform well in multi-channel networks since it cannot support proportionate progress as in fluid models [24]. Therefore, we turn to optimal multi-processor scheduling for inspiration. In particular, we develop our algorithm based on the idea of deadline partitioning (DP) [24][25]. In traditional real-time systems, DP is the technique of *partitioning time into slices*, demarcated by the deadlines of all the jobs in the system. Within each slice, all the jobs are allocated a *workload* for the time slice, and these workloads share the same deadline. Then, the DP-fair [25] scheduling algorithm allocates a workload to a job in proportion to the *work density* of the job (i.e., the work to be completed divided by the allowable time to complete the work). Therefore, DP-fair ensures proportionate progress in all the jobs and is optimal for computational job scheduling in multi-processor systems.

Given that the availability of multiple channels in industrial wireless networks is similar to the availability of multiple processors in multi-processor computer systems, we explore in this study the application of the DP methodology to PPRC scheduling for industrial URLLC applications. To this end, we need to address two fundamental differences between multi-cell industrial wireless networks and typical multi-processor systems: *Firstly*, not all the links interfere with one another in multi-cell industrial wireless networks, thus each communication channel can be used by more than one link at the same time. *Secondly*, unlike multi-processor systems where centralized solutions are feasible, dynamic, multi-cell PPRC scheduling requires distributed solutions.

To address the aforementioned differences and as we will present in detail in Sections III-B, we observe that, using the conflict graph to model inter-link interference and building upon the multi-channel distributed scheduling algorithm Unified Cellular Scheduling (UCS) [26], the network can be decoupled, and each link only needs to coordinate with the other links in the one-hop neighborhood of the conflict graph in applying DP-based real-time scheduling. However, the PPRC scheduling problem is NP-hard as formally shown in the online technical report [27]. Therefore, we first develop an approximate solution by extending the traditional DP method to local-deadline-partition (LDP) real-time scheduling.

B. Local-deadline-partition (LDP) PPRC scheduling

Each link i and its interfering links in M_i shall not transmit in the same channel at the same time. Thus the set of links in $M_i \cup \{i\}$ can be treated as a conflict set competing for the same set of resources, just as how a set of jobs compete for the same computing resource in a multi-processor system. Therefore, we can extend the concepts of deadline partition, workload, and work density in DP-Fair scheduling [24][25] to each conflict set. In particular, we can define the concepts of local deadline partition, local traffic demand, and local work density to ensure steady, proportionate progress towards completing the required workload (i.e., the number of transmissions required for the PPRC guarantee) within deadlines, and use the local work density to prioritize packet transmissions along different links of a conflict set.

Unlike traditional real-time systems where the deadline partition (DP) is based on global information (i.e, real-time parameters of all the tasks), the local-deadline-partition (LDP) splits time based only on the information of one-hop links in the conflict graph. In particular, for a link $i \in E$ and $j = 1, 2, \ldots$, let $A_{i,j}$ and $D_{i,j}$ denote the arrival time and absolute deadline of the j-th packet along link i, respectively; if the j-th packet arrives at the beginning of time slot t, $A_{i,j} = t-1$, and $D_{i,j} = A_{i,j}+D_i = t+D_i-1$. Then, we sort the arrival times and absolute deadlines of the packets along the links in $M_i \cup \{i\}$ in a non-decreasing order, and regard each non-zero interval between any two consecutive instants of packet arrival/deadline as a *local deadline partition*. More specifically,

Definition 2 (Local Deadline Partition)**.** *At a time slot t, the* local deadline partition (LDP) *at a link $i \in E$, denoted by $\sigma_{i,t}$, is defined as the time slice $\left[d'_{i,t}, d''_{i,t}\right)$, where* $d'_{i,t} = \max\{\max_{k \in M_i \cup \{i\}, D_{k,j} \leq t} D_{k,j}, \max_{k \in M_i \cup \{i\}, A_{k,j} \leq t} A_{k,j}\}$, *and* $d''_{i,t} = \min\{\min_{k \in M_i \cup \{i\}, D_{k,j} > t} D_{k,j}, \min_{k \in M_i \cup \{i\}, A_{k,j} > t} A_{k,j}\}$.

Note that, different from DP-fair which only uses deadlines for deadline-partition demarcation, LDP uses both arrival times and deadlines in the demarcation. This is because, unlike in traditional real-time computing systems where all the competing jobs are conflicting with one another, not all the links interfere with one another in multi-cell industrial wireless systems. Consequently, using both arrival times and deadlines in deadline-partition demarcation ensures finer-grained proportionate progress in packet transmissions than using deadlines alone in the demarcation, and this finer-grained control of proportionate progress in link packet transmissions is required for ensuring per-packet real-time guarantee in LDP.

We denote the length of $\sigma_{i,t}$ by $L_{i,t}$, which equals $d^{''}_{i,t}$ - $d^{'}_{i,t}$. Let $P_{i,t} = \lceil \frac{t-A_{i,1}}{T_i} \rceil$, then link i is in its $P_{i,t}$-th period at a time slot t for all $t > A_{i,1}$. Let $X'_{i,t}$ denote the number of times that the $P_{i,t}$-th packet at link i has been transmitted along link i till time slot t, then $X''_{i,t} = X_i - X'_{i,t}$ is the remaining work demand of link i at time slot t. At the beginning of each deadline partition, we allocate a local traffic demand to link i, and it equals the link's remaining work demand multiplied by the ratio of the length of the current deadline partition (i.e., $L_{i,t}$) to the length of the interval between the current time slot and the absolute deadline (i.e., $D_{i,P_{i,t}} - d'_{i,t}$). Inside the deadline partition $\sigma_{i,t}$, the local traffic demand decreases as packets are transmitted in $\sigma_{i,t}$. Precisely, we define the local traffic demand and local work density of a local deadline partition $\sigma_{i,t}$ as follows:

Definition 3 (Local Traffic Demand)**.** *For link $i \in E$ and time slot t, the* local traffic demand *of link i in $\sigma_{i,t}$, denoted by $X_{i,t}$, is as follows:*

$$X_{i,t} = \begin{cases} X''_{i,d'_{i,t}} \frac{L_{i,t}}{D_{i,P_{i,t}} - d'_{i,t}} & D_{i,P_{i,t}} > d'_{i,t}, t = d'_{i,t} \\ X_{i,d'_{i,t}} - (X'_{i,t} - X'_{i,d'_{i,t}}) & D_{i,P_{i,t}} > d'_{i,t}, t > d'_{i,t} \\ 0 & D_{i,P_{i,t}} \leq d'_{i,t} \end{cases} \tag{1}$$

where $D_{i,P_{i,t}} \leq d'_{i,t}$ indicates the case of link i having completed its current packet transmissions and thus having a zero local traffic demand at time t.

Definition 4 (Local Work Density)**.** *For link i, the* local work density *of $\sigma_{i,t}$, denoted by $\rho_{i,t}$, is defined as the ratio of the local traffic demand $X_{i,t}$ to the time duration till the local deadline of completing the transmission of these local traffic. That is, $\rho_{i,t} = \frac{X_{i,t}}{L_{i,t}-(t-d'_{i,t})} = \frac{X_{i,t}}{d''_{i,t}-t}$.*

Based on these definitions, we develop the LDP real-time scheduling algorithm by extending the multi-channel distributed scheduling algorithm Unified Cellular Scheduling (UCS) [26] to consider PPRC requirements.

In particular, at a time slot t in each local deadline partition, the transmitters and receivers of links set their local work densities which are then used to define the links' relative priorities, with links having larger local work densities being given higher priorities in channel access. The transmitter and receiver of a link i compare the priority of link i with its interfering links (i.e., M_i) in scheduling, and they execute the following algorithm in a distributed manner:

1) The transmitter and receiver of each link $i \in E$ initializes its state as UNDECIDED for each channel $rb \in RB$ and calculate its local work density in time t. Note that we use the local work density as the priority. Then, the priority will be shared with interfering links through a control channel.
2) The transmitter and receiver of link i iterates over the following steps until the state of link i in each channel is either ACTIVE or INACTIVE:
    - For a channel rb in which the state of link i is UNDECIDED, if the local traffic demand $X_{i,t}$ is zero or if there exists an interfering ACTIVE link, the state of link i is set as INACTIVE;
    - If link i is UNDECIDED and if it has higher priority or the same priority but larger ID than every other UNDECIDED link in M_i, the state of i in channel rb is set as ACTIVE, and its local traffic demand $X_{i,t}$ is reduced by one;
    - Both the transmitter and receiver of link i share the state of link i with every other node that has at least one associated link interfering with i;
    - The transmitter and receiver of link i update the state and priority of a link $l \in M_i$, if the transmitter and/or receiver receive a state update about l.

If the state of a link i is ACTIVE for channel rb at time slot t, link i can transmit a data packet at channel rb and time t. The detail of the local-deadline-partition (LDP) scheduling algorithm for time slot t is shown in Algorithm 1. For conciseness of presentation, the above discussion regards all the links as

Algorithm 1 Local-Deadline-Partition (LDP) Real-Time Scheduling at Link i and Time Slot t

Input: $A_{i,1}$: the arrival time of the first packet along link i;
  M_i: set of interfering links of a link $i \in E$;
  T_l, D_l: period and relative deadline of link $l \in M_i \cup \{i\}$:
  $X_{i,t}$: local traffic demand at link i;
  $State.l.rb.t$: the transmission state of links $l \in M_i \cup \{i\}$ for $\forall rb \in RB$ at time t;
  $Prio.l.t$: priority of links $l \in M_i \cup \{i\}$;

Output: Perform the following actions at the transmitter and receiver of link i:

1: $State.i.rb.t = UNDECIDED$, $\forall rb \in RB$;
2: $Prio.i.t = X_{i,t}/(d''_{i,t} - t)$;
3: Share $Prio.i.t$ with the links in M_i;
4: done = false;
5: **while** done == false **do**
6:   done = true;
7:   **for** each $rb \in RB$ and in the increasing order of the ID of rb **do**
8:     **if** having received updates on $State.l.rb.t$ or $Prio.l.t$ from a link $l \in M_i$ **then**
9:       Update the local copy of $State.l.rb.t$ or $Prio.l.t$ at link i;
10:     **end if**
11:     **if** $X_{i,t}$ **==** 0 **and** $State.i.rb.t == UNDECIDED$ **then**
12:       $State.i.rb.t = INACTIVE$;
13:       break;
14:     **end if**
15:     **if** $\exists l \in M_i : State.l.rb.t == ACTIVE$ **then**
16:       $State.i.rb.t = INACTIVE$;
17:       break;
18:     **end if**
19:     **if** $State.i.rb.t == UNDECIDED$ **and** $((Prio.i.rb.t > Prio.l.rb.t)$ **or** $(Prio.i.rb.t = Prio.l.rb.t$ **and** ID.i $>$ ID.l)) holds for every UNDECIDED $l \in M_i$ **then**
20:       $State.i.rb.t = ACTIVE$;
21:       $X_{i,t} = X_{i,t} - 1$;
22:     **end if**
23:     **if** $State.i.rb.t == UNDECIDED$ **then**
24:       done = false;
25:     **end if**
26:   **end for**
27:   Share $State.i.rb.t, \forall rb \in RB$, with the transmitters and receivers of the links in M_i;
28: **end while**

playing similar roles in executing LDP. For cellular networks, the base stations (BSes) usually take on more roles and assist the user equipment (UEs) in algorithm execution, and we will briefly discuss this in our technical report [27]. To illustrate the key concepts of the LDP scheduling algorithm, let's look at how the algorithm is executed for the network whose conflict graph is Figure 2. For conciseness of exposition, here we assume the number of channels $N = 2$; the key intuition from

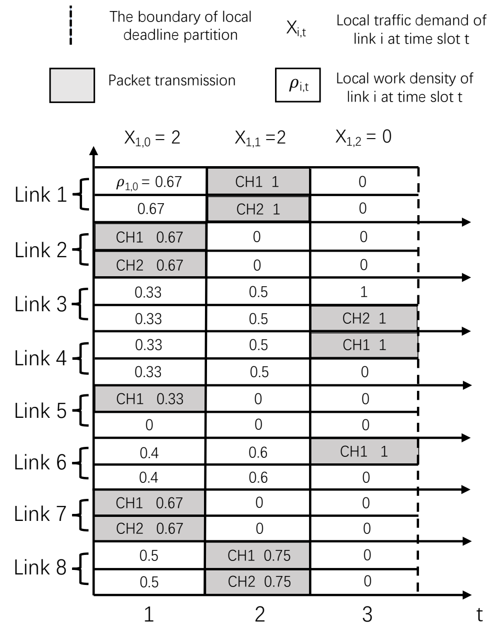

Fig. 3: Example of LDP scheduling

the example is applicable to general multi-channel settings. Suppose the real-time traffic of a link i is characterized as $\phi_i = (T_i, D_i, X_i)$, and the network traffic is such that $\phi_1 = (6, 6, 4)$, $\phi_2 = (4, 3, 2)$, $\phi_3 = (6, 6, 2)$, $\phi_4 = (12, 12, 4)$, $\phi_5 = (12, 12, 4)$, $\phi_6 = (6, 5, 2)$, $\phi_7 = (6, 6, 4)$, $\phi_8 = (4, 4, 2)$. Then, the scheduling results from time slot 1 to 3 is shown in Figure 3. Let's first focus on link 1. By ordering the arrival times and absolute deadlines of the packets along the links in $M_1 \cup \{1\}$ in an increasing order, the first local deadline partition for link 1 is $[1, 3)$. The local traffic demand of link 1 at the beginning of time slot 1 equals the (remaining) traffic demand (i.e., 4) multiplied by the ratio of the length of the deadline partition (i.e., 3) to the duration from the beginning of time slot 1 to the end of the absolute deadline of 6, that is, being $4 \times \frac{3}{6-0} = 2$, and the priority (i.e., local work density) of link 1 is the local traffic demand divided by the length from the current time slot to the end of the current local deadline partition, that is, $\frac{2}{3} = 0.67$. At the beginning of time slot 1, to decide whether link 1 shall transmit at time slot 1, it compares its priority with those of the links in M_1. Even though link 1 has higher priority than links 3, 4, and 5, it has the same priority as link 2 and has a smaller ID than link 2. Therefore, according to line 19 in Algorithm 1, link 2 uses CH1 to transmit a packet and sets its state in channel CH1 as ACTIVE. Since the ACTIVE link 2 conflicts with links 1 and 3, links 1 and 3 become INACTIVE at time slot 1 for CH1 according to line 16 in algorithm 1 for CH1. Similar analysis can be applied to other time slots and links shown in Figure 3 [27]. lgorithm 1 can be shown to converge for each time slot t, and we have

Theorem 1. *For each frequency channel and time slot, the set of ACTIVE links is a maximal set of links that are mutually non-interfering and have data packets yet to be delivered.*

Proof. When the iteration terminates, a link is either AC-

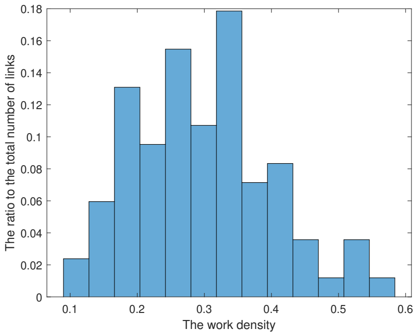

Fig. 4: The work density of each link

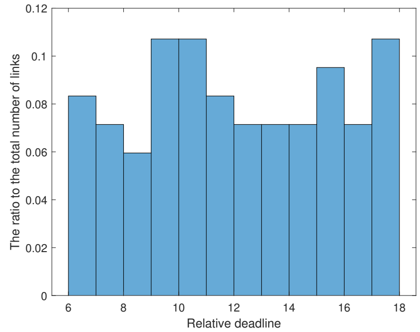

Fig. 5: Relative deadline

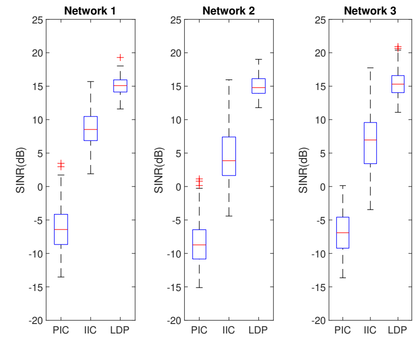

Fig. 6: Interference effect on receiver-side SINR

TIVE/INACTIVE based on lines 23 and 24 of Algorithm 1. For each INACTIVE link i with non-zero local traffic demand in any channel, there always exists at least one ACTIVE link l, $l \in M_i$, based on lines 15, 16 in Algorithm 1. Therefore, changing any INACTIVE link to an ACTIVE link would cause two interfering links active at the same time slot in the same channel, which is not allowed. Hence, the set of all ACTIVE link for any channel is a maximal independent set. □

IV. Numerical Study

TABLE I: NETWORK SETTINGS

Network size	120*120 m^2, 240*240 m^2.
Number of links	83, 151, 320.
Channel model	Wireless Industrial Indoor path loss model with path loss coefficient 3.
Bandwidth	20MHz.
Number of channels	3-11.
Packet size	1,000bytes.
Modulation	16QAM.
SINR threshold	15dB.
PPRC traffic	([6,21], [6,18], $[10^{-3}, 10^{-9}]$).

Here we numerically evaluate the properties of the LDP scheduling algorithm in diverse multi-cell industrial wireless networks.

A. Network and PPRC traffic settings

We consider two networks of different sizes. The network size, number of channels, link/node spatial distribution density, and number of conflicting links per link are chosen to represent different real-time network settings.

For Network 1, we uniform-randomly deploy 91 wireless nodes in a 120×120 square-meter region, generating a network of 83 links. There are nine cells which are organized in a 3×3 grid manner. There is a base station (BS) within each cell. For Network 2, we uniform-randomly deploy 151 wireless nodes in a 120×120 square-meter region, generating a network of 163 links. There are nine cells which are organized in a 3×3 grid manner. For Network 3, we uniform-randomly deploy 320 wireless nodes in a 240×240 square-meter region, generating a network of 324 links. There are 36 cells which are organized in a 6×6 grid manner. In addition, we apply the Wireless Industrial Indoor path loss model [28] to determine the interference effect among links. (Detailed typology information of Networks 1, 2, and 3 can be found in [27].)

Regarding the number of channels, with a numerology similar to 5G Numerology 4, the subcarrier spacing is 240KHz, and, assuming that each resource-block (RB) consists of 12 subcarriers, each RB occupies 2.8MHz spectrum. Assuming a communication bandwidth of 20MHz, it gives 7 RBs (i.e., N =7). To represent industrial URLLC scenarios having diverse timing requirements and thus diverse transmission-time-intervals (TTI) and numerologies, the number of channels considered here ranges from 3 to 11. Assuming that the packet size is 1,000 bytes[1] and 16QAM modulation is applied, the bit error rate could achieve 10^{-6} when the SINR threshold is 15dB, and the link reliability can achieve 99%. When the per-packet communication reliability is 99%, we need at least 5 transmission opportunities if industrial URLLC applications require the probability of packet loss or deadline violation to be no more than 10^{-3} or even 10^{-9}. To experiment with different work densities and to include scenarios of both light and heavy PPRC traffic, the traffic demand X_i (i.e., required number of transmission opportunities per packet) along a link i is uniform-randomly chosen from $[2, 5]$. Most industrial URLLC use cases such as discrete automation can accept 10-20ms one-way delay, thus we assume that the relative deadline D_i uniform-randomly ranges from 6 to 18 time slots. The period is assumed to be greater than or equal to the relative deadline, and we experiment with different periods that differ from the relative deadline by a value uniformly distributed in $[0, D_i/6]$. To this end, we consider scenarios of demanding PPRC traffic that is close to the network capacity but can still be supported by the LDP algorithm. Figure 4 shows the histogram of the links' work densities when the number of channels is 4, and Figure 5 shows the histogram of the relative deadlines. The overall network settings are shown in Table I.

B. Numerical results

a) Impact of interference coordination: To understand the importance of considering interference control in multi-

[1]The packet size for industrial URLLC control message may have short packet size, while industrial URLLC media data require large packet size. If the packet size for control messsage is 100 bytes, then the link reliability can achieve 99.9% according to the network settings, and the required transmission opportunities is at least 4, which is considered in our traffic demand.

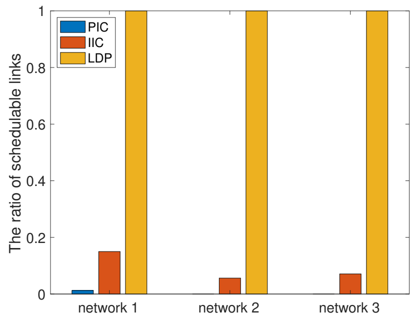

Fig. 7: Interference effect on schedulability ratio

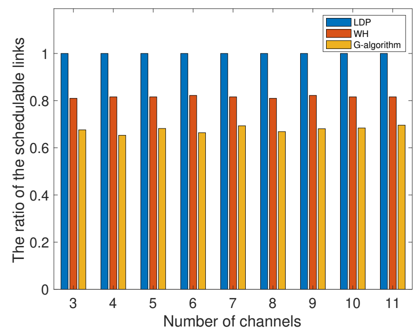

Fig. 8: Comparison with G-Schedule and WH

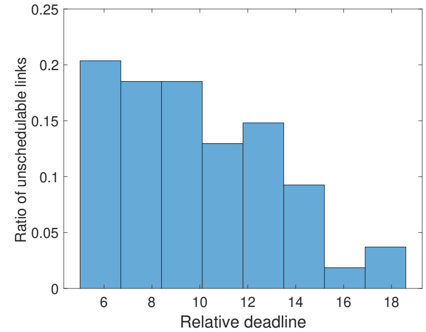

Fig. 9: Infeasible links for G-Schedule

cell industrial wireless networks, we consider the impact of three different interference coordination methods. The first method only considers primary interference control (PIC). That is, only those links sharing a common transmitter or receiver are regarded as conflicting with one another. The second method only considers primary interference control and intra-cell interference control (IIC). That is, the links in the same cell cannot transmit at the same time slot and through the same frequency channel. The third method considers the PRK-based intra-cell and inter-cell interference control which is utilized by LDP, and we set the SINR threshold as 15dB. Each interference coordination method has its associated conflict graph for Networks 1, 2 and 3 respectively, and we use the LDP scheduling algorithm with the different interference coordination methods to understand their impact. For each network, we generate the traffic demand that is close to the respective network capacity but can still be supported by the LDP algorithm, and then measure the SINR value as shown in Figure 6. For network 1, the mean SINR of PIC is -6.4281dB, and its 25%-75% percentiles is [-8.67, -4.16]; the mean SINR of IIC is 8.53dB, and its 25%-75% percentiles is [6.87, 10.47]; the mean SINR of LDP is 15.09dB, and its 25%-75% percentiles is [14.13, 15.95]. For network 2, the mean SINR of PIC is -8.72dB, and its 25%-75% percentiles is [-10.84, -6.46]; the mean SINR of IIC is 3.86dB, and its 25%-75% percentiles is [1.64, 7.41]; the mean SINR of LDP is 14.79dB, and its 25%-75% percentiles is [13.94, 16.12]. For network 3, the mean SINR of PIC is -6.91dB, and its 25%-75% percentiles is [-9.22, -4.85]; the mean SINR of IIC is 6.96dB, and its 25%-75% percentiles is [3.42, 9.57]; the mean SINR of LDP is 15.31dB, and its 25%-75% percentiles is [14.03, 16.59]. Therefore, considering the intra-cell interference and inter-cell interference in LDP ensures the required receiver-side SINR, and it significantly increases the receiver-side SINR as compared with PIC and IIC, e.g., by a margin of over 20dB.

We also consider the impact of the receiver-side SINR on real-time schedulability, which is shown as the ratio of schedulable links in Figure 7. The ratio of schedulable links greatly increases with the decreasing interference. In particular, the ratio of schedulable links of network 1 increase from 0.0125 of PIC, 0.15 of IIC to 1 of LDP; the ratio of schedulable links of network 2 increase from 0 of PIC, 0.056 of IIC to 1 of LDP; the ratio of schedulable links of network 3 increase from 0 of PIC, 0.07 of IIC to 1 of LDP. We see that using the LDP scheduling algorithm to address intra-cell and inter-cell interference can significantly increase the real-time capacity (i.e., ratio of scheduable links) by a huge margin, e.g., a factor of about 5-20 as compared with IIC. Having demonstrated the impact of considering interference control in URLLC, we next examine the benefit of LDP as compared with other real-time wireless scheduling algorithms that consider interference control.

b) Comparative study: Out of the existing real-time wireless scheduling algorithms that consider intra-cell and inter-cell interference, the WirelessHART-based algorithm (WH) [9] and G-schedule algorithm [14] address problems that are closest to the PPRC scheduling problem. For real-time multi-channel scheduling in multi-cell cellular networks, the WirelessHART-based algorithm considers the scheduling methods EDF and DM (where the link with the shortest deadline acquires the highest priority), and it gives the worst-case delay analysis and a closed-form schedulability test. G-schedule greedily schedules non-interfering links based on their IDs, and it has been shown to be optimal for the special line networks where all the nodes are located along a straight line [14]. To study the performance of LDP and the two other algorithms, we calculate the schedulability test of WirelessHART-based algorithm for each node in Network 2 and implement the LDP and G-algorithm in Matlab and study their behavior in Network 2. (Similar phenomena have been observed for other networks.) We execute each algorithm for 200,000 time slots and observe the ratio of the number of schedulable links (i.e., the links whose probabilistic per-packet real-time requirement is met) to the total number of links.

Figure 8 shows the ratio of schedulable links in the network. We see that, while LDP is able to schedule demanding PPRC traffic (i.e., the ratio of the schedulable links is 100%), the average ratio of schedulable links in WH algorithm and G-Schedule are 0.82 and 0.68. The cause for the difference between WH algorithm and LDP is that the former only considers the worst case for each node and underestimates the feasibility, while LDP can improve such worst case analysis and calculate the schedulability test based on the topology information. The ratio of schedulable links in WH does not increase with the number of channels since the WH-based schedulability test considers the sum work density of a set of links and not the number of channels. To understand the cause for the difference between G-Schedule and LDP, we divide the links into different groups according to their relative deadlines, and calculate the ratio of the number of unschedulable links in G-schedule to the total number of links in the corresponding group. Figure 9 shows the relationship between unschedulable links and their relative deadline. We see that the links with shorter deadlines are more likely to become unschedulable in G-schedule. This is because G-schedule greedily schedules links without considering heterogeneous deadline constraints, and the links with shorter deadlines tend to be assigned with fewer transmission opportunities with respect to their deadlines. On the other hand, LDP dynamically updates packets' priorities based on in-situ work densities, and the links with higher work densities and closer to their absolute deadlines tend to get higher priorities. Accordingly, LDP can support more demanding real-time traffic than what G-schedule can.

V. Concluding Remarks

For supporting heterogeneous industrial URLLC applications in large-scale networks, we have proposed a distributed local-deadline-partition (LDP) scheduling algorithm to ensure Probabilistic Per-packet Real-time Communications (PPRC) guarantee in large-scale, multi-cell, and multi-channel network settings. The LDP algorithm effectively leverages the one-hop information in the conflict graph and addresses the challenges of multi-cell, multi-channel PPRC scheduling. Our numerical results have shown that the LDP algorithm can significantly improve the network capacity of URLLC (e.g., by a factor of 5-20) and can support significantly more PPRC traffic than the state-of-the-art solutions.

Focusing on the fundamental PPRC scheduling problem for industrial URLLC applications, this study represents a first step towards enabling URLLC in large-scale industrial networks with multiple channels and heterogeneous real-time requirements, and it serves as a foundation for exploring other interesting studies. For instance, a next-step is to develop the associated schedulability test, and implement the LDP scheduling algorithm with PRKS [29] in emerging open-source cellular platforms such as OpenAirInterface.

References

[1] M. Luvisotto, Z. Pang, and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2016.

[2] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, 2020.

[3] Y. Chen, H. Zhang, N. Fisher, L. Y. Wang, and G. Yin, "Probabilistic per-packet real-time guarantees for wireless networked sensing and control," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2133–2145, 2018.

[4] K. Li, P. Zhu, Y. Wang, F.-C. Zheng, and X. You, "Joint uplink and downlink resource allocation towards energy-efficient transmission for urllc," *IEEE Journal on Selected Areas in Communications*, 2023.

[5] K. Wang, D. Niyato, W. Chen, and A. Nallanathan, "Task-oriented delay-aware multi-tier computing in cell-free massive mimo systems," *IEEE Journal on Selected Areas in Communications*, 2023.

[6] T. Q. Duong, *Ultra-reliable and Low-Latency Communications (URLLC) theory and practice: Advances in 5G and beyond*. John Wiley & Sons, Ltd., 2023.

[7] W. Yu, T. J. Chua, and J. Zhao, "Asynchronous hybrid reinforcement learning for latency and reliability optimization in the metaverse over wireless communications," *IEEE Journal on Selected Areas in Communications*, 2023.

[8] K. Min, Y. Kim, and H.-S. Lee, "Meta-scheduling framework with cooperative learning towards beyond 5g," *IEEE Journal on Selected Areas in Communications*, 2023.

[9] V. P. Modekurthy, A. Saifullah, and S. Madria, "DistributedHART: A distributed real-time scheduling system for WirelessHART networks," in *IEEE RTAS*, 2019.

[10] V. P. Modekurthy, D. Ismail, M. Rahman, and A. Saifullah, "A Utilization-Based Approach for Schedulability Analysis in Wireless Control Systems," in *IEEE ICII*, 2018.

[11] V. P. Modekurthy, A. Saifullah, and S. Madria, "DistributedHART: A distributed real-time scheduling system for WirelessHART Networks," in *IEEE RTAS*, 2019.

[12] T. Gong, T. Zhang, X. S. Hu, Q. Deng, M. Lemmon, and S. Han, "Reliable dynamic packet scheduling over lossy real-time wireless networks," in *ECRTS*, 2019.

[13] A. Destounis and G. S. Paschos, "Complexity of URLLC scheduling and efficient approximation schemes," *CoRR*, vol. abs/1904.11278, 2019.

[14] A. Tan, Q. Wang, N. Guan, Q. Deng, and X. S. Hu, "Inter-cell Channel Time-Slot Scheduling for Multichannel Multiradio Cellular Fieldbuses," in *IEEE RTSS*, 2015.

[15] Y. Peng, A. Jolfaei, and K. Yu, "A novel real-time deterministic scheduling mechanism in industrial cyber-physical systems for energy internet," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5670–5680, 2021.

[16] X. Zhang, J. Wang, and H. V. Poor, "Aoi-driven statistical delay and error-rate bounded qos provisioning for urllc over wireless networks in the finite blocklength regime," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 3115–3120.

[17] Y. Kuo, "Minimum Age TDMA Scheduling," in *IEEE INFOCOM*, 2019.

[18] G. Zhang, C. Shen, Q. Shi, B. Ai, and Z. Zhong, "Aoi minimization for wsn data collection with periodic updating scheme," *IEEE Transactions on Wireless Communications*, 2022.

[19] I. Kadota and E. Modiano, "Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals," in *ACM Mobihoc*, 2019.

[20] C. Li, H. Zhang, T. Zhang, J. Rao, L. Y. Wang, and G. Yin, "Cyber-Physical Scheduling for Predictable Reliability of Inter-Vehicle Communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, 2020.

[21] N. H. Mahmood, O. A. López, H. Alves, and M. Latva-Aho, "A Predictive Interference Management Algorithm for URLLC in Beyond 5G Networks," *IEEE Communication Letters*, vol. 25, no. 3, pp. 995–999, 2021.

[22] H. Malik, M. M. Alam, Y. L. Moullec, and Q. Ni, "Interference-Aware Radio Resource Allocation for 5G Ultra-Reliable Low-Latency Communication," in *IEEE Globecom Workshops*, 2018.

[23] 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 (TOSN)*, vol. 10, no. 2, 2014.

[24] H. Cho, B. Ravindran, and E. D. Jensen, "An optimal real-time scheduling algorithm for multiprocessors," in *(RTSS'06)*. IEEE, 2006, pp. 101–110.

[25] G. Levin, S. Funk, C. Sadowski, I. Pye, and S. Brandt, "DP-FAIR: A simple model for understanding optimal multiprocessor scheduling," in *ECRTS*, 2010.

[26] Y. Xie, H. Zhang, and P. Ren, "Unified Scheduling for Predictable Communication Reliability in Cellular Networks with D2D Links," *Computer Communications (Elsevier)*, vol. 167, 2021.

[27] Z. Meng, H. Zhang, and J. Gross, "Scheduling with probabilistic per-packet real-time guarantee for URLLC," in *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2101.01768v7

[28] M. Cheffena, "Propagation channel characteristics of industrial wireless sensor networks [wireless corner]," *IEEE Antennas and Propagation Magazine*, vol. 58, no. 1, pp. 66–73, 2016.

[29] 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," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6135–6150, 2017.