Scheduling with Probabilistic Per-Packet Real-Time Guarantee for 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 also address the associated challenge of schedulability test. In particular, we propose the concept of *feasible set*, identify a closed-form sufficient condition for the schedulability of PPRC traffic, and then propose an efficient distributed algorithm for the schedulability test. 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*— URLLC, probabilistic per-packet real-time communications (PPRC) guarantee, large-scale cellular networks.**

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.

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.

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], [13]. 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 [14], [15], [16]. Simultaneously, there are studies focusing on providing long-term real-time guarantees, such as mean delay and age-of-information (AoI) for wireless systems [17], [18], [19], [20]. 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. Thus it is necessary to deploy multiple base stations (BSes) to provide large spatial coverage leading to well-known interference effects. 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 [21], [22], [23]. 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. Given the large scale and the dynamic, uncertain nature of such networks, it is important for the scheduling algorithm to be amenable to distributed implementation without requiring centralized coordination or centralized knowledge of the whole network. Given that every network has a limited communication capacity, it is also important to be able to decide whether a set of PPRC requests can be supported by the network and the associated scheduling algorithm. Thus, there is the need to develop effective schedulability test algorithms that can be deployed in practice. We show that corresponding scheduling algorithms can simultaneously provide real-time guarantees while effectively mitigating interference 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.
- For the schedulability test of PPRC traffic, we propose the concept of *feasible set*, which bridges the real-time computing systems theory and URLLC. We then identify a closed-form sufficient condition for the schedulability of PPRC traffic, and we propose an efficient distributed algorithm for the schedulability test.
- We also identify a necessary condition for the schedulability of PPRC traffic, and use numerical studies to understand a lower bound on the approximation ratio of the LDP algorithm and associated schedulability test.
- 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 [15] 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 summarize related work in Section II, present the system model and problem definition in Section III, present the LDP real-time scheduling algorithm and the associated schedulability test algorithmin Section IV, evaluate the properties of the LDP algorithm in Section V, and make concluding remarks in Section VI. *Due to the limitation of space, we relegate to the technical report [24] the notation summary as well as the proofs of the theorems and lemmas of the paper.*

II. Related Work

By controlling the allocation of available radio resources (e.g., time, frequency, and power) to meet the specific requirements of URLLC applications, resource allocation is an essential aspect of algorithm development for URLLC. Firstly, URLLC resource allocation has been studied as an *optimization* problem. For instance, Li et al. [4] aims to optimize bandwidth allocation and power control for both uplink and downlink transmissions, with the objective of minimizing the average total power while ensuring URLLC transmission requirements. However, this approach assumes the availability of global information and homogeneous delay constraints. Wang et al. [5] focuses on investigating the computational latency and total energy consumption in cell-free massive MIMO systems. These factors are considered as the total cost, and the study examines minimizing this cost by controlling bandwidth and task allocation. However, the approach does not guarantee reliability nor latency in the optimization. Secondly, *machine learning* methods have also been applied in URLLC resource allocation algorithms. For instance, resource allocation with unsupervised learning in URLLC has been studied in [6], where delay and reliability constraints are converted into bandwidth constraints, and unsupervised deep learning is used to find optimal bandwidth allocation. Yu et al. [7] presents a multi-agent reinforcement learning algorithm structure to optimize the overall completion time of uplink and downlink processes, as well as minimizing energy consumption in downlink transmissions. In addition, Min et al. [8] introduces objective-specific meta-scheduling policies using deep reinforcement learning techniques. The proposed approach considers user utilities, such as average data rate and latency requirements, as the basis for optimizing scheduling decisions. However, these ML methods do not guarantee the latency and reliability requirements of URLLC. Thirdly, *3GPP* includes technology elements for supporting URLLC. Hamidi et al. [29] demonstrated how 3GPP Release 16 extends the real-time capabilities of Release 15 for enhancing latency and reliability; these technology elements alone do not guarantee meeting URLLC requirements, and careful resource allocation methods such as what we present in this paper are needed. The need for URLLC real-time scheduling has also been recognized in recent 3GPP Release 18 standard [30].

Real-time communications have also been extensively studied for industrial wireless networks. Initial works applied real-time systems scheduling algorithms such as earliest-deadline-first (EDF) and rate-monotonic (RM) to wireless systems. Chen et al. [3] and Wu et al. [25] have proposed EDF-based scheduling algorithms which ensure per-packet real-time and reliability guarantee. However, both work have avoided wireless channel spatial reuse to satisfy reliability requirements, and the scheduling algorithms therein are not applicable to multi-cell settings. Real-time scheduling in WirelessHART industrial wireless networks has been studied [9], [10], [11], [12], [26], [27]. They apply EDF, deadline-monotonic, or fixed-priority scheduling algorithms which under-perform in multi-channel settings [10], [11], [26], [27]. In addition, they do not consider channel spatial reuse [12] as frequently applied

TABLE I: Related work in real-time scheduling

	Multi-cell wireless network	Predictable per-packet real-time guarantee	Heterogeneous real-time requirement	Distributed algorithm	multi-channel optimal scheduling
EDF or RM-based scheduling [3], [9], [25], [10], [11], [12], [26], [27]		✓	✓	✓	
Centralized scheduling [15], [14], [16]	✓	✓			
Mean delay distributed scheduling [28], age-of-information (AoI) studies [17]	✓			✓	
Resource allocation for URLLC [4], [5], [6], [7], [8], [29]	✓				✓
LDP scheduling	✓	✓	✓	✓	✓

in multi-cell deployments.

In a second group of works, centralized real-time communication solutions are proposed that explicitly address the differences between traditional real-time systems and wireless real-time communications. Destounis et al. [14] have considered the probabilistic nature of wireless communications and tried to maximize the utility subject to real-time and reliability constraints in communication. However, the study did not consider heterogeneous real-time requirements across links, and the proposed approach is only suitable for single-cell settings. Tan et al. [15] and Peng et al. [16] have proposed a centralized scheduling algorithm which is optimal for line networks. But the above centralized solutions are difficult to implement in distributed, multi-cell settings in practice, and they did not consider the deadline requirements of individual packets and did not ensure communication reliability.

A third group of works have considered the provisioning of long-term real-time guarantees for wireless systems. For instance, mean delay has been considered in distributed scheduling [28], and age-of-information (AoI) has also been considered in recent studies [17]. However, these work have not considered ensuring predictable timeliness of individual packet transmissions in multi-cell, multi-channel network settings.

Table I summarizes the key differences between the LDP scheduling algorithm and existing work.

III. 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, [1] 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 address the challenge, Zhang et al. [31] 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 [31] 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 $T_{S,R}$. With local, distributed coordination among close-by links alone, PRK-based scheduling achieves *close-to-optimal communication capacity* while ensuring the required mean communication reliability [32]. The PRK model and PRK-based scheduling have been observed to be *generically applicable* to diverse wireless platforms (e.g., IEEE 802.15.4, LTE, and 5G radios) [31], [32], both immobile networks [32] and mobile networks [21], and both ad-hoc networks [32] and cellular networks with D2D links [33], and in real-world scenarios of complex wireless behavior such as multi-path fading and cumulative interference from all concurrent transmitters in networks.

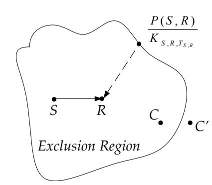

Fig. 1: Physical-Ratio-K (PRK) interference model

[1] In 5G URLLC, time slots of different lengths are used, and the length of a time slot may be the multiple of that of the shortest time slot. The PPRC methodology of this paper is readily extensible to the case of varying-length time slots by treating the length of the shortest time slot as the smallest time unit. Detailed study of this is worthwhile but beyond the scope of this work (which represents a first step towards understanding fundamental PPRC scheduling issues).

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 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]. Note that, when $S_i > p_i$, $X_i > 1$ and retransmissions are required to ensure the required packet delivery probability S_i. Then, the second sub-requirement requires that these X_i transmission opportunities are used within deadline D_i. [2] 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 $\rho_i = \frac{X_i}{D_i}$.

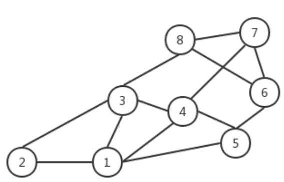

Fig. 2: Example conflict graph G_c (Note: this example will be used to illustrate other concepts in the rest of the paper too.)

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) $(D_i \leq T_i)$, is the set of links schedulable to meet the PPRC requirement? If yes, develop an algorithm that schedules the data traffic to satisfy the PPRC requirements; if not, indicate the infeasibility.

IV. 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 [34]. Therefore, we turn to optimal multi-processor scheduling for inspiration. In particular, we develop our algorithm based on the idea of deadline partitioning (DP) [34][35]. 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 [35] 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

[2] The X_i reserved time slots do not have to be consecutive, and, for real-time packet delivery, they only have to be before the delivery deadline of the packet.

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. Yet the problem of identifying the maximum set of links that can share the same channel is NP-hard itself [31]. In addition, even though only close-by links interfere with one another [32] and have to directly coordinate in accessing wireless channels, links far-away from one another are still indirectly coupled due to the chaining effect in connected networks. *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 IV-B, and IV-C, 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) [33], the network can be decoupled, and each link only needs to coordinate with the other links in the two-hop neighborhood of the conflict graph in applying DP-based real-time scheduling. Similarly, schedulability can be tested locally at individual links, and the network-wide PPRC traffic is schedulable as long as the link-local schedulability test is positive. However, the PPRC scheduling problem is NP-hard as formally shown in the online technical report [24]. Therefore, we first develop an approximate solution by extending the traditional DP method to local-deadline-partition (LDP) real-time scheduling, and then we study the associated schedulability test and approximation ratio.

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 [34][35] 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 $d'_{i,t} = \max\{\max_{k\in M_i\cup\{i\}, D_{k,j}\le t} D_{k,j}, \max_{k\in M_i\cup\{i\}, A_{k,j}\le t} A_{k,j}\}$, 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, as shown in the proof of Lemma 3, 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:*

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

where $D_{i,P_{i,t}} \le d'_{i,t}$ 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) [33] to consider PPRC requirements.

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:

```
State.i.rb.t = UNDECIDED, ∀rb ∈ RB;
Prio.i.t = X_{i,t}/(d''_{i,t} − t);
Share Prio.i.t with the links in M_i;
done = false;
while done == false do
    done = true;
    for each rb ∈ RB and in the increasing order of the ID of rb do
        if having received updates on State.l.rb.t or Prio.l.t from a link l ∈ M_i then
            Update the local copy of State.l.rb.t or Prio.l.t at link i;
        end if
        if X_{i,t} == 0 and State.i.rb.t == UNDECIDED then
            State.i.rb.t = INACTIVE;
            break;
        end if
        if ∃l ∈ M_i : State.l.rb.t == ACTIVE then
            State.i.rb.t = INACTIVE;
            break;
        end if
        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 ∈ M_i then
            State.i.rb.t = ACTIVE;
            X_{i,t} = X_{i,t} − 1;
        end if
        if State.i.rb.t == UNDECIDED then
            done = false;
        end if
    end for
    Share State.i.rb.t, ∀rb ∈ RB, with the transmitters and receivers of the links in M_i;
end while
```

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 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 Section IV-E. 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 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)

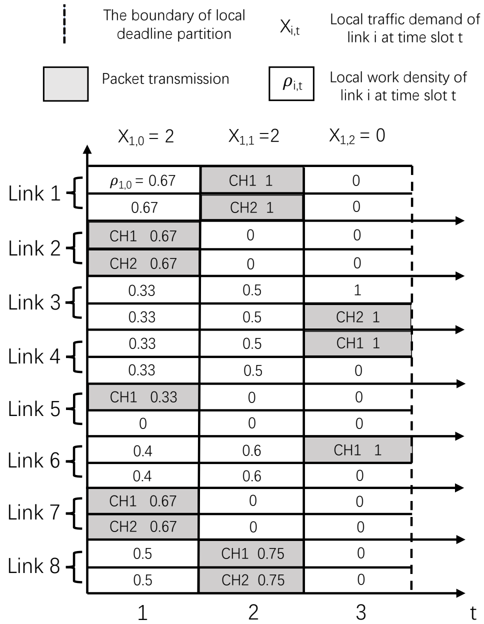

Fig. 3: Example of LDP scheduling

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 [24]. Algorithm 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.*

C. PPRC schedulability test

Given an arbitrary network G, it is not always possible to find a schedule to meet the PPRC requirement. Therefore, an important task is to determine the schedulability of a set of real-time communication links. To this end, we consider the schedulability of each individual link, and a set of links is schedulable if every link of the set is schedulable. Given that a link i interferes with every link in M_i, i shares the N wireless channels with the links in M_i. Therefore, we try to map the schedulability test of traditional real-time systems into the set of links in $M_i \cup \{i\}$. However, unlike tasks in real-time systems where two or more tasks cannot access the same CPU core at the same time, certain links in M_i may not interfere with one another, and those non-interfering links can be active in the same wireless channel and at the same time. For the conflict graph shown in Figure 2, for instance, link 1 can transmit concurrently with link 6, 7, or 8 in the same channel. Therefore, to utilize the findings from traditional real-time systems, we shall focus on the PPRC traffic demand of the links in every maximal clique $K_{i,j} \subseteq M_i \cup \{i\}$ such that all the links in the clique are interfering with one another.

For each maximal clique $K_{i,j} \subseteq M_i \cup \{i\}$, $i \in K_{i,j}$, and there could be at most one active link in any channel at any moment in time. Due to transmissions along the links other than $M_i \cup \{i\}$, however, it is possible that, for a given wireless channel and time slot, none of the links in a clique $K_{i,j}$ can be active (i.e., when their interfering links are active in the given channel and time slot). For instance, for the conflict graph shown in Figure 2, if links 2, 5, and 8 are active, then none of the links in the clique $\{1, 3, 4\}$ can be active in the same channel.

Therefore, we propose the concept of *feasible set* that, for a given link i, jointly considers the PPRC traffic demand of each set of links that is the union of a set of cliques in $M_i \cup \{i\}$ and that, for any given wireless channel and time slot, can have at least one active link in all cases but can have only one active link in the worst case of the transmissions along the links other than $M_i \cup \{i\}$. That is, in a feasible set, there will be at least N number of packets transmitted for each time slot. (As we will show shortly in Theorem 2, the concept of feasible set is a foundation for the PPRC schedulability test.)

More precisely, we define the concepts of *minimum scheduling rate* and *feasible set* to capture the core intuition of this approach.

Definition 5 (Minimum Scheduling Rate)**.** *Given a conflict graph G_c, a set of links $S \subseteq G_c$, and the set of all maximal independent set of G_c, denoted by MIS_{G_c}, the* minimum scheduling rate of *S is $N \times min_{mis \in MIS_{G_c}} |mis \cap S|$, where $|mis \cap S|$ is the number of links in the set $mis \cap S$.*

Definition 6 (Feasible Set)**.** *Given a link i and a maximal clique $K_{i,j}$ in the conflict graph G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$. Let $\mathbb{K}_i = \{$maximal clique $K_{i,j'} : i \in K_{i,j'} \wedge K_{i,j'} \subseteq M_i \cup \{i\} \wedge K_{i,j'} \subseteq G_c\}$, and $U_{K_{i,j}} \subseteq \mathbb{K}_i$ such that $K_{i,j} \in U_{K_{i,j}}$.* A feasible set*, denoted by $S_{i,K_{i,j}}$, is defined as the set of links in a $U_{K_{i,j}}$ whose minimum scheduling rate is N (i.e., the number of communication channels in the network).*

As an example, for the conflict graph shown in Figure 2 and the links in $M_1 \cup \{1\}$, there are 3 maximal cliques, that is, $K_{1,1} = \{1, 2, 3\}$, $K_{1,2} = \{1, 3, 4\}$, and $K_{1,3} = \{1, 4, 5\}$. For $K_{1,2}$, the set of feasible sets for $\{1\}$ and $K_{1,2}$, denoted by $\mathbb{S}_{1,K_{1,2}}$, is $\{\{1, 2, 3, 4\}, \{1, 3, 4, 5\}, \{1, 2, 3, 4, 5\}\}$. Note that $\{1, 3, 4\}$ is not a feasible set because its minimum scheduling rate is zero, which in turn is due to the fact that $\{2, 5, 8\}$ is a maximal independent set for the example conflict graph and it does not include any of the links from $\{1, 3, 4\}$. On the other hand, for link 1 and $K_{1,1}$, the clique $K_{1,1}$ itself is also a feasible set since its minimum scheduling rate is N.

The objective of defining the feasible set concept is to understand the schedulability of PPRC traffic and to enable schedulability test. Therefore, we need to know whether there exists a feasible set for all the links.

Lemma 1. *Given a link $i \in E$ and any maximal clique $K_{i,j} \subseteq$*

$M_i \cup \{i\}$, *there exists at least one feasible set.*

Then, to understand the conditions for schedulability, we first study the conditions under which schedulability is violated. In general, if the work density of link i's interfering links is heavy, then link i is more likely to be unschedulable. Specifically, the violation condition is as follows:

Lemma 2. *Given a link i and any maximal clique $K_{i,j}$ such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$, if link i misses its absolute deadline at a time slot t, then for each feasible set $S_{i,K_{i,j}} \subseteq M_i \cup i$, $\sum_{l \in S_{i,K_{i,j}}} \rho_{l,t-1} \geq N+1$.*

Next, we derive a sufficient condition that ensures the schedulability of a link i all the time.

Lemma 3. *Given a link i, if, for every maximal clique $K_{i,j}$ where $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$, there exists a feasible set $S_{i,K_{i,j}}$ such that $i \in S_{i,K_{i,j}}$, $K_{i,j} \subseteq S_{i,K_{i,j}}$, $S_{i,K_{i,j}} \subseteq M_i \cup \{i\}$, and the sum of the work density of all the links in $S_{i,K_{i,j}}$ is no more than N, then the X_i number of transmissions of each packet at link i will be completed before the associated deadline.*

Based on Lemma 3, we now derive the schedulability condition. To decide whether a link i is schedulable, we just need to identify the associated feasible set(s) with the minimum sum work density and check whether the minimum sum work density is no more than the number of channels N. More precisely, the schedulability condition is as follows:

Theorem 2 (Schedulability Condition)**.** *Given a link i and the conflict graph G_c, let $\mathbb{K}_i$ denote the set of maximal cliques $K_{i,j}$ in G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$, and let $\mathbb{S}_{i,K_{i,j}}$ denote the set of feasible sets for a clique $K_{i,j} \in \mathbb{K}_i$. If $\forall K_{i,j} \in \mathbb{K}_i$, we have,*

$$\min_{S_{i,K_{i,j}} \in \mathbb{S}_{i,K_{i,j}}} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l} \leq N, \tag{2}$$

then the PPRC traffic of link i can be supported, that is, link i is schedulable.

From Theorem 2, the PPRC schedulability test requires a method of identifying the associated feasible set(s) with the minimum sum work density. To this end, we need an approach to identifying all the feasible sets of interest. By the definition of feasible sets (i.e., Definitions 5 and 6), whether a set $S_{i,K_{i,j}} \subseteq M_i \cup \{i\}$ is a feasible set depends on the maximal independent sets (MIS) of the conflict graph G_c. Yet searching for all the MISes of a graph is NP-hard, and, for large graphs, it tends to be computationally undesirable and may even be infeasible in practice. To address the challenge, we observe that, *instead of checking all the MISes of G_c, we only need to check the MISes of the subgraph of G_c induced by the links within two-hop distance from link i*, since only these links directly impact whether certain links in $M_i \cup \{i\}$ can be active at certain wireless channels and time slots. More precisely, we define the Two-Hop Interference Set of a link i and identify two unique properties of feasible sets as follows.

Definition 7 (Two-hop Interference Set)**.** *Given a conflict graph G_c and a node $i \in G_c$, the* two-hop interference set *of link i, denoted by $M_{i,2}$, is the set of links whose distances from i in G_c are two hops.*

Then, we only need to consider $M_{i,2}$ to determine whether a set is a feasible set. For instance, for link 1 and set $S_{1,K_{1,2}} = \{1,3,4,5\}$ in the example conflict graph and network represented by Figure 2, $M_{1,2} = \{6,7,8\}$, and $M_1' = \{2,6,7,8\}$. $MIS_{M_1'} = \{\{2,6\},\{2,7\},\{2,8\}\}$. It is easy to verify that, for any of the set $\{2,6\}$, $\{2,7\}$, or $\{2,8\}$, there exists a link in $S_{1,K_{1,2}}$ that does not interfere with any links of the chosen set. Therefore, $S_{1,K_{1,2}}$ is a feasible set. More precisely, we give the following theorem to determine a feasible set.

Theorem 3 (Checking Feasible Set)**.** *Given a link i, a set of links S_i that is the union of a set of cliques each of which includes i as an element and is a subset of $M_i \cup \{i\}$, define $M_i' = (\{i\} \cup M_i \cup M_{i,2}) \setminus S_i$, and, when $M_i' \neq \emptyset$, denote all the maximal independent sets of M_i' as $MIS_{M_i'}$. When $M_i' = \emptyset$, S_i is a feasible set; when $M_i' \neq \emptyset$, S_i is a feasible set if and only if, for each $mis \in MIS_{M_i'}$, there exists at least one link in S_i that does not interfere with any link in mis.*

To leverage Theorems 2 and 3 in developing the PPRC schedulability test algorithm for link i, we need a mechanism of identifying the feasible set(s) of minimum sum work density for every maximal clique $K_{i,j}$ in G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$. To this end, we observe the following property of feasible sets.

Theorem 4 (Feasible Set Generation)**.** *Consider a link i and a set of links $S_{i,K_{i,j}}$ such that $i \in S_{i,K_{i,j}}$ and $S_{i,K_{i,j}} \subseteq M_i \cup \{i\}$. For any maximal clique $K_{i,j'}$ in G_c such that $i \in K_{i,j'}$ and $K_{i,j'} \subseteq M_i \cup \{i\}$, if $S_{i,K_{i,j}}$ is a feasible set, then the link set $S_{i,K_{i,j}} \cup K_{i,j'}$ is also a feasible set; if $S_{i,K_{i,j}}$ is not a feasible set, then the set $S_{i,K_{i,j}} \setminus K_{i,j'}$ is not a feasible set either.*

Let $\mathbb{K}_i$ denote the set of maximal cliques $K_{i,j}$ in G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$. Then, per Definition 6, for every $K_{i,j} \in \mathbb{K}_i$, every corresponding feasible set $S_{i,K_{i,j}}$ is the links of a subset of $\mathbb{K}_i$ that includes $K_{i,j}$. According to Theorem 4, the feasible set $S^*_{i,K_{i,j}}$ that has the minimum sum work density will be the union of $K_{i,j}$ and a minimal number of elements in $\mathbb{K}_i \setminus \{K_{i,j}\}$ that makes a feasible set. Therefore, for every $K_{i,j} \in \mathbb{K}_i$, to identify the feasible set $S^*_{i,K_{i,j}}$ having the minimum sum work density, we only need to search the subsets of $\mathbb{K}_i \setminus \{K_{i,j}\}$ of increasing cardinality and stop once every subset of a certain cardinality plus $K_{i,j}$ is a feasible set. Accordingly, we develop Algorithm 2 for PPRC schedulability test.

In Algorithm 2, we firstly verify if each maximal clique $K_{i,j}$ is a feasible set or not. If it is, then the feasible set with minimum sum work density has been found. Otherwise, let $\mathbb{K}_i'$ denote the set of cliques that cannot be a feasible set individually. This means that for the set of cliques in $\mathbb{K}_i'$, we need to find a combination of cliques that can form a feasible set. For each $K_{i,j} \in \mathbb{K}_i'$, let $A_{c,p}$ denote a union of c number of maximal cliques $K_{i,m} \in \mathbb{K}_i \setminus \{K_{i,j}\}$, and let $\mathbb{A}_c$ denote the set of all possible $A_{c,p}$. For each $A_{c,p} \in \mathbb{A}_c$, if the set of links in $A_{c,p} \cup K_{i,j}$ is a feasible set according to Theorem 3, then the sum work density of the links in $A_{c,p} \cup K_{i,j}$ will be compared with $U_{i,K_{i,j}}$, and the smaller one will be the new $U_{i,K_{i,j}}$. Given a specific set cardinality c, if the set of links in $A_{c,p} \cup K_{i,j}$ is a feasible set for each $A_{c,p} \in \mathbb{A}_c$, then the algorithm does not need to check the subsets of greater cardinality and will terminate immediately.

Algorithm 2 Schedulability Test at Link i

Input: N: the number of channels;
$\mathbb{K}_i$: the set of maximal cliques $K_{i,j}$ in G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq \mathbb{K}_i$;
M_i: set of interfering links of a link $i \in E$;
$M_{i,2}$: set of two-hop interference links of a link $i \in E$;
X_l, D_l: traffic demand and relative deadline of link $l \in M_i \cup \{i\}$;
Output: whether link i is schedulable;

```
1:  U_{i,K_{i,j}} = ∞, ∀K_{i,j} ∈ 𝕂_i;
2:  𝕂'_i = 𝕂_i;
3:  for each clique K_{i,j} ∈ 𝕂_i do
4:    if the set of links in K_{i.j} is a feasible set according to
      Theorem 3, M_i, and M_{i,2} then
5:      U_{i,K_{i,j}} = Σ_{l∈K_{i.j}} X_l/D_l;
6:      𝕂'_i = 𝕂'_i \ {K_{i,j}};
7:    end if
8:  end for
9:  for each clique K_{i,j} ∈ 𝕂'_i do
10:   done = 0;
11:   for c = 1, ..., |𝕂_i \ {K_{i,j}}| do
12:     done = 1;
13:     for each A_{c,p} (i.e., a union of c cliques from 𝕂_i \
        {K_{i,j}}) do
14:       if the set of links in A_{c,p} ∪ K_{i.j} is a feasible set
          according to Theorem 3, M_i, and M_{i,2} then
15:         U_{i,K_{i,j}} = min(U_{i,K_{i,j}}, Σ_{l∈{A_{c,p}∪K_{i.j}}} X_l/D_l);
16:       else
17:         done = 0;
18:       end if
19:     end for
20:     if done == 1 then
21:       break;
22:     end if
23:   end for
24: end for
25: if U_{i,K_{i,j}} ≤ N, ∀K_{i,j} ∈ 𝕂_i then
26:   link i is schedulable;
27: else
28:   link i is not schedulable;
29: end if
```

The "for" loop in line 9 needs at most $|\mathbb{K}_i|$ iterations, and there will be at most $|\mathbb{K}_i| - 1$ iterations for the "for" loop in line 11. For each c, there will be $\binom{|\mathbb{K}_i|-1}{c}$ times to check Theorem 3, and $\binom{|\mathbb{K}_i|-1}{c} \leq \binom{|\mathbb{K}_i|-1}{\lfloor \frac{|\mathbb{K}_i|-1}{2} \rfloor}$ for $\forall c = 1, \ldots, |\mathbb{K}_i|-1$. Therefore, the total computational complexity of Algorithm 2 is $\mathcal{O}(|\mathbb{K}_i|(|\mathbb{K}_i|-1)\binom{|\mathbb{K}_i|-1}{\lfloor \frac{|\mathbb{K}_i|-1}{2} \rfloor})$.

D. Optimality analysis

Given that the PPRC scheduling problem (see Section III-D) is NP-hard, the LDP algorithm and the associated schedulability test are approximations of the optimal solutions. Here we develop a necessary condition for PPRC schedulability and use it to derive a lower bound on the approximation ratio of LDP scheduling[3].

Theorem 5 (Necessary Condition for PPRC Schedulability)**.** *Given a link i and the conflict graph G_c, let $\mathbb{K}_i$ denote the set of maximal cliques $K_{i,j}$ in G_c such that $i \in K_{i,j}$ and $K_{i,j} \subseteq M_i \cup \{i\}$. Then, if link i is schedulable, we have*

$$\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l} \leq N. \tag{3}$$

Based on Theorems 2 and 5, we can explore the gap between the sufficient condition (2) and necessary condition (3). In particular, a *lower bound on the approximation ratio*, denoted by $\delta(i)$, is the ratio of the left-hand side of the necessary condition (3) to that of the sufficient condition (2). That is,

$$\delta(i) = \frac{\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l}}{\max_{K_{i,j} \in \mathbb{K}_i} \min_{S_{i,K_{i,j}} \in \mathbb{S}_{i,K_{i,j}}} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l}}. \tag{4}$$

The lower bound depends on two factors: PPRC traffic and network topology, with the former impacting the work densities at individual links and the latter impacting the interference relations among links. Given a specific PPRC traffic, the sum of work density for a set of links increase with the number of links in the set. Hence, to explore the impact of network topology, we define the *topology approximation ratio* as follows. For each clique $K_{i,j}$, $K_{i,j} \in \mathbb{K}_i$, let $S_{min,i,K_{i,j}} = \arg\min_{S_{i,K_{i,j}} \in \mathbb{S}_{i,K_{i,j}}} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l}$. Then, the topology approximation ratio can be defined as

$$\delta(i)' = \frac{|K'_{max,i,j}|}{|S_{max,i}|} \tag{5}$$

where $K'_{max,i,j}$ is the clique in $\mathbb{K}_i$ that has the maximum number of links, and $S_{max,i}$ is, for all $K_{i,j} \in \mathbb{K}_i$, the feasible set $S_{min,i,K_{i,j}}$ that has the maximum number of links.

We can obtain a closed-form solution to the approximation ratio lower-bound for the following network settings: network G is large, the link reliability p_i is the same for all the links, and the exclusive regions of all links include the same number of interfering links. That is,

Theorem 6 (Approximation ratio lower-bound)**.** *For network G, the approximation ratio of algorithm LDP scheduling is greater than $\frac{1}{6}$.*

E. Remarks: implementation strategies

While this study mainly focuses on the fundamental algorithmic aspects of the PPRC scheduling problem in URLLC applications, here we briefly present a sketch of an implementation strategy of our approach regarding a cellular network

[3]The approximation ratio is defined as, for each link i, the PPRC traffic load regarded as schedulable by the LDP algorithm and associated schedulability test (2) divided by the PPRC traffic load schedulable by any optimal scheduling algorithm.

architecture. This strategy builds on the implementation of Unified Cellular Scheduling (UCS) using the open-source cellular software platform OpenAirInterface [33] and USRP software defined radios. Both the LDP scheduling algorithm and the PPRC schedulability test algorithm require coordination between base stations (BSes) and user equipment (UEs), as well as among the BSes. The BS-UE coordination can be achieved by using the physical downlink control channel (PDCCH) and physical uplink control channel (PUCCH) to carry relevant control information (e.g., input needed for Algorithm 1). Inter-BS coordination can be achieved through the Xn interface, and the Xn interface is usually implemented using high-throughput, low-latency fiber networks.

In terms of quantitative estimates, when it comes to the coordination overhead of the LDP algorithm, a link i only needs to share the priority (1 byte) and state information (2 bits) with other links $l \in M_i$ at each time slot, via their associated BSes. The other necessary inputs can be stored in the BS when the system boots up and only need to be modified if the network changes. Based on the information gathered through coordination with its UEs and neighboring BSes, each BS executes the LDP scheduling algorithm on behalf of all the cellular and D2D links in its cell. Like existing packet transmission scheduling algorithms in cellular networks, the LDP scheduling algorithm is executed at the beginning of each time slot. Given that the input to the LDP algorithm only involves low-overhead, local coordination between a BS and its UEs and neighboring BSes, the control overhead tends to be low and is not a barrier to the field-deployment of the LDP algorithm.

Compared to the LDP algorithm, the PPRC schedulability test algorithm is executed at much lower frequencies and at a timescale of URLLC session dynamics (e.g., emergence of a new URLLC session) and conflict graph dynamics. Thus, the associated control overhead is even lighter than that of LDP.

V. Numerical Study

TABLE II: 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 and the PPRC schedulability test 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 [36] to determine the interference effect among links. (Detailed typology information of Networks 1, 2, and 3 can be found in [24].)

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[4] 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. [5] 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 $[0, D_i/6]$. The overall network settings are shown in Table II.

B. Numerical results

a) Approximation ratio of LDP: Here we evaluate the lower bound on the approximation ratio of the LDP scheduling algorithm for Networks 1, 2 and 3. To this end, we consider scenarios of demanding PPRC traffic that is close to the

[4]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.

[5]Instead of using absolute time values such as 1ms, here we use time-slot as the unit of time specification. Depending on the numerology used in a cellular network, the duration of a time-slot can be configured as 1ms, 0.5ms, 0.1ms etc.

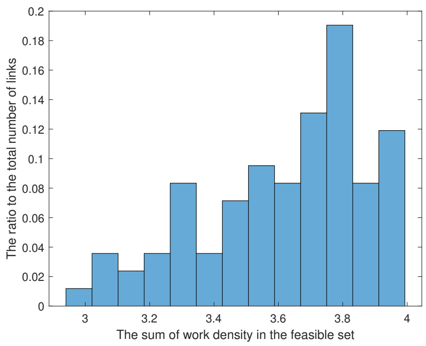

Fig. 4: Minimum sum work density of feasible sets

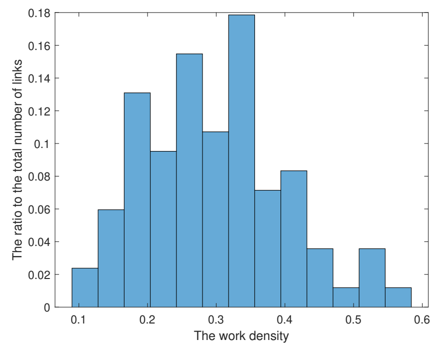

Fig. 5: The work density of each link

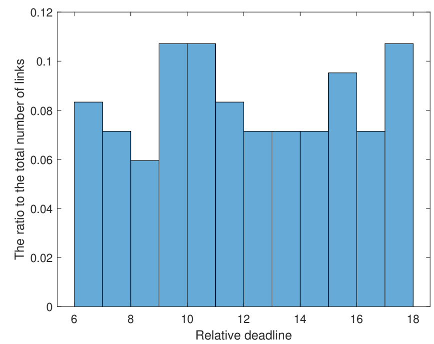

Fig. 6: Relative deadline

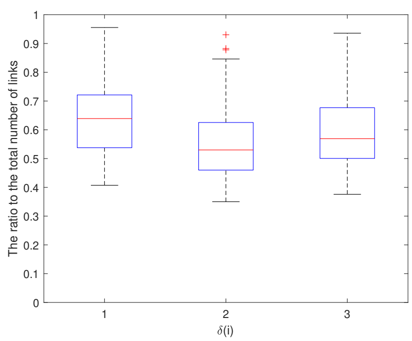

Fig. 7: Approximation ratio lower bound

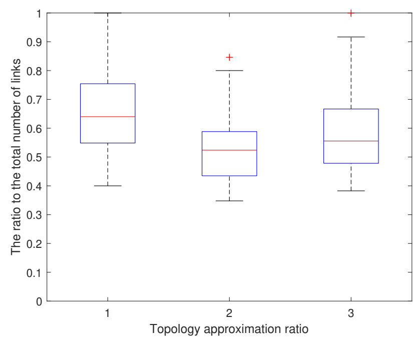

Fig. 8: Topology approximation ratio

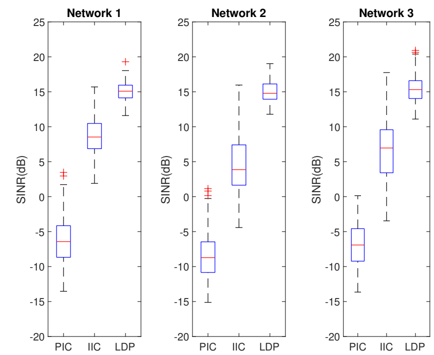

Fig. 9: Interference effect on receiver-side SINR

network capacity but can still be supported by the LDP algorithm. We take Network 1 as an example, Figure 4 shows the histogram of the minimum sum work density of the feasible sets when the number of channels is 4, Figure 5 shows the histogram of the links' work densities, and Figure 6 shows the histogram of the relative deadlines. Then, we show the numerical results for the three networks. Figures 7 is drawn from Equation 4, and they show the approximation ratio lower bound $\delta(i)$ for all the links in Networks 1, 2 and 3 respectively. Figures 8 is drawn from Equation 5, and they show the topology approximation ratio in Networks 1, 2 and 3 respectively. For Network 1, the mean approximation ratio lower bound is 0.639, and its 25%-75% percentiles is $[0.5376, 0.7212]$; the mean topology approximation ratio is 0.643, and its 25%-75% percentiles is $[0.5486, 0.7543]$. We see that network topology has significant impact on the approximation ratio, even though the PPRC traffic pattern also impacts the approximation ratio. For Network 2, the mean approximation ratio lower bound is 0.5309, and its 25%-75% percentiles is $[0.4615, 0.6254]$; the mean topology approximation ratio is 0.5238, and its 25%-75% percentiles is $[0.4348, 0.5882]$. For Network 3, the mean approximation ratio lower bound is 0.5691, and its 25%-75% percentiles is $[0.5006, 0.6767]$; the mean topology approximation ratio is 0.5556, and its 25%-75% percentiles is $[0.47826, 0.6667]$. We see that the approximation ratio lower bound in Network 2 and 3 is about 10% lower than that in Network 1. This is because the number of interfering links per link in Networks 2 and 3 tends to be higher than that in Network 1. Accordingly, the size of cliques in the conflict graph of Networks 2 and 3 is greater than that of Network 1, which makes the approximation ratio lower bound relatively lower in Networks 2 and 3. The mean approximation ratio lower bound is more than 0.53 for the three networks, and it is up to 0.9553, 0.933 and 0.941 in Networks 1, 2 and 3 respectively.

b) Impact of interference coordination: To understand the importance of considering interference control in multi-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 9. 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

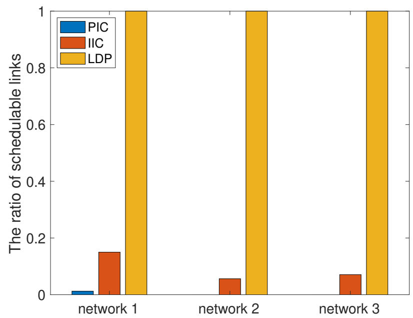

Fig. 10: Interference effect on schedulability ratio

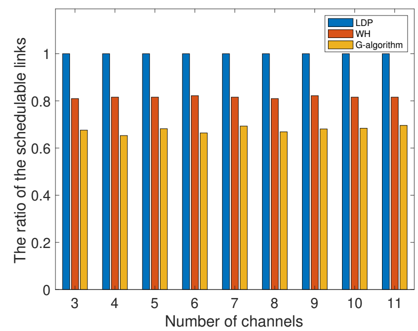

Fig. 11: Comparison with G-Schedule and WH

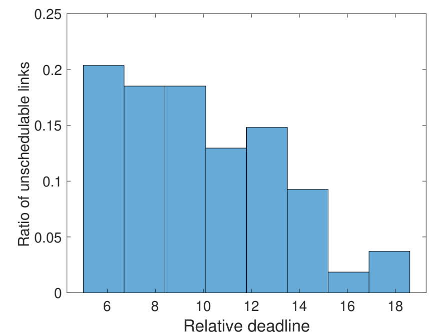

Fig. 12: Infeasible links for G-Schedule

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 10. 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.

c) 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 [15] 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 [15]. 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 11 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 12 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.

VI. 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. The concept of feasible set in this paper bridges real-time computing systems and URLLC. Leveraging the feasible set concept, we have identified a closed-form sufficient condition for PPRC schedulability test; we have also developed an algorithm for finding the minimum sum work density of feasible sets, upon which we have developed the schedulability test algorithm. 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, given that the LDP scheduling algorithm and associated schedulability test are amenable to real-world implementation in cellular networks, a next-step is to implement and integrate the LDP scheduling algorithm with PRKS [32] in emerging open-source cellular platforms such as OpenAirInterface. Another interesting direction is to consider delay jitter control since URLLC applications such as XR tend to require as small delay jitter as possible.

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] Z. Meng, H. Zhang, and J. Gross, "Scheduling with probabilistic per-packet real-time guarantee for URLLC," in *IEEE International Conference on Industrial Cyber-Physical Systems*, 2024.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

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

[25] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, and Y. Chen, "Analysis of EDF scheduling for wireless sensor-actuator networks," in *IWQoS*, 2014.

[26] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for wirelesshart networks," in *IEEE RTSS*, 2010.

[27] A. Saifullah, D. Gunatilaka, P. B. Tiwari, M. Sha, C. Lu, B. Li, C. Wu, and Y. Chen, "Schedulability analysis under graph routing in wirelesshart networks," in *IEEE RTSS*, 2015.

[28] V. Venkataranmanan, X. Lin, L. Ying, and S. Shakkottai, "On scheduling for minimizing end-to-end buffer usage over multihop wireless networks," in *IEEE INFOCOM*, 2010.

[29] F. Hamidi-Sepehr, M. Sajadieh, S. Panteleev, T. Islam, I. Karls, D. Chatterjee, and J. Ansari, "5g urllc: Evolution of high-performance wireless networking for industrial automation," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 132–140, 2021.

[30] P. Hande, P. Tinnakornsrisuphap, J. Damnjanovic, H. Xu, M. Mondet, H. Y. Lee, and I. Sakhnini, "Extended Reality over 5G – Standards Evolution," *IEEE Journal on Selected Areas in Communications*, vol. to appear, 2023.

[31] 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.

[32] 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.

[33] 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.

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

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

[36] 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.