Multi-Cell, Multi-Channel URLLC with Probabilistic Per-Packet Real-Time Guarantee

(Technical Report: ISU-DNC-TR-2020-01; October 26, 2020)

Zhibo Meng, Hongwei Zhang
Department of Electrical and Computer Engineering
Iowa State University
{zhibom,hongwei}@iastate.edu

***Abstract*—Ultra-reliable, low-latency communication (URLLC) represents a new focus in 5G-and-beyond networks, and it is expected to enable mission-critical sensing and control as well as AR/VR applications. URLLC requires controlling the communication quality of individual packets. Prior studies have considered probabilistic per-packet real-time guarantees for single-cell, single-channel networks with implicit deadline constraints, but they have not considered real-world complexities such as inter-cell interference and multiple communication channels. Towards ensuring URLLC in multi-cell, multi-channel wireless networks, we propose a real-time scheduling algorithm based on *local-deadline-partition (LDP)*. The LDP algorithm is suitable for distributed implementation, and it ensures probabilistic per-packet real-time guarantee for multi-cell, multi-channel networks with general deadline constraints. We also address the associated challenge of the schedulability test of URLLC traffic. In particular, we propose the concept of *feasible set* and identify a closed-form sufficient condition for the schedulability of URLLC traffic. We propose a distributed algorithm for the schedulability test, and the algorithm includes a procedure for finding the minimum sum work density of feasible sets which is of interest by itself. We also identify a necessary condition for the schedulability of URLLC traffic, and use numerical studies to understand a lower bound on the approximation ratio of the LDP algorithm. We experimentally study the properties of the LDP algorithm and observe that the URLLC traffic supportable by the LDP algorithm is significantly higher than that of a state-of-the-art algorithm.**

***Index Terms*—Wireless sensing and control networks, URLLC, per-packet real-time guarantee, probabilistic real-time guarantee**

I. Introduction

A key objective of 5G-and-beyond wireless networks is to support ultra-reliable, low-latency communication (URLLC) services. URLLC can be applied to mission-critical sensing and control in many domains. In real-time augmented vision, for instance, wireless networks can enable the fusion of real-time video streams from spatially distributed cameras to eliminate the line-of-sight constraint of natural human vision and thus enable seeing-through obstacles [1], [2]. In industrial automation, wireless-enabled mobile, pervasive, and reconfigurable instrumentation and the significant cost of planning, installing, and maintaining wired network cables have made wireless networks attractive for industrial monitoring and control; machine-type-communication for real-time sensing and control has also become a major focus of 5G wireless network research and development [3].

Unlike traditional, best-effort wireless networks designed for high-throughput applications, reliable and real-time delivery of individual packets is critical for URLLC applications [2]. In industrial sensing and control, for instance, tens or hundreds of nodes may periodically generate packets that need to be delivered to their destinations in a few milliseconds, and the probability of packet loss or deadline violation shall be no more than 10^{-6} or even 10^{-9} [4]. Several recent studies [5] [6] [7] [8] have considered long-term real-time communication guarantees (e.g., ensuring a long-term, asymptotic probability of real-time packet delivery), but they did not address the per-packet real-time communication guarantees required by URLLC applications. Chen et al. [2] have proposed an earliest-deadline-first (EDF) scheduling algorithm that ensures probabilistic per-packet real-time guarantee in single-cell, single-channel settings. However, the EDF-based algorithm tends to under-perform in multi-channel settings, and it does not address inter-cell interference in multi-cell settings.

In many envisioned URLLC applications such as those in industrial process control, factory automation, and precision agriculture farming, the network may well be deployed across a large area of space to provide URLLC services to a larger number of nodes. Thus it is important to deploy multiple base stations (BSes) to provide large spatial coverage. To improve overall network communication capacity, it is also important to leverage as many communication channels as possible instead of requiring the whole network to communicate over a single wireless channel. Therefore, it is critical to develop real-time scheduling algorithms that ensure per-packet real-time guarantee for multi-cell, multi-channel URLLC. Given the large scale and the dynamic, uncertain nature of such URLLC wireless networks, it is also 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 URLLC communication requests can be supported by the network and the associated scheduling algorithm. Thus there is the need to develop effective schedu-

This work is supported in part by the NSF awards CNS-1821962 and OAC-1827211.

lability test algorithms that can be deployed in practice.

Contributions. To address the aforementioned challenges and to provide probabilistic per-packet real-time communication guarantees for multi-cell, multi-channel URLLC, we propose a distributed real-time scheduling algorithm and an associated schedulability test method. Our main contributions are as follows:

- Building upon the idea of deadline partitioning in traditional real-time systems, 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 real-time scheduling algorithm that ensures probabilistic per-packet real-time guarantee in multi-cell, multi-channel URLLC networks with general deadline constraints.
- For the schedulability test of URLLC traffic, we propose the concept of *feasible set* and identify a closed-form sufficient condition for the schedulability of URLLC traffic. We then propose a distributed algorithm for the schedulability test, and the algorithm includes a procedure for finding the minimum sum work density of feasible sets which is of interest by itself.
- We also identify a necessary condition for the schedulability of URLLC traffic, and use numerical studies to understand a lower bound on the approximation ratio of the LDP algorithm and associated schedulability test.
- We experimentally study the properties of the LDP algorithm and observe that the URLLC traffic supportable by the LDP algorithm is significantly higher than that of the state-of-the-art algorithm G-schedule [9]. For instance, the LDP algorithm is able to support the URLLC requirement of a large network 36.8% of whose links cannot be supported by G-schedule.

The rest of the paper is organized as follows. We summarize related work in Section II, and we present the system model and problem definition in Section . We present, in Section IV, our LDP real-time scheduling algorithm and the associated schedulability test algorithm. We evaluate the properties of the LDP algorithm in Section V, we make concluding remarks in Section VI. For ease of reference, Table I summarizes the key notations used in the paper.

T_i	period of link i	$D_{i,j}$	absolute deadline of j-th packet along link i
P_i	reliability requirement of link i	$A_{i,j}$	arrival time of j-th packet along link i
p_i	the link reliability of link i	ρ_i	the work density of link i
$\sigma_{i,t}$	local deadline partition of link i at time t	$d'_{i,t}$	beginning time of $\sigma_{i,t}$
$d''_{i,t}$	absolute deadline of $\sigma_{i,t}$	$L_{i,t}$	the length of $\sigma_{i,t}$
X_i	the work demand of link i	$X'_{i,t}$	the number of times that link i has transmitted for the period at time t
$X''_{i,t}$	the remaining work demand of link i at time t	$X_{i,t}$	local traffic demand
$\rho_{i,t}$	local work density of link i at time t	M_i	the set of conflict links of link i
$K_{i,j}$	the j-th clique of link i that $K_{i,j} \subseteq M_i \cup i$	$\mathbb{K}_i$	the set of $K_{i,j}$
$U_{K_{i,j}}$	a union of cliques that $K_{i,j} \subseteq U_{K_{i,j}}$ and $U_{K_{i,j}} \subseteq \mathbb{K}_i$	$S_{i,K_{i,j}}$	the feasible set that $i \in S_{i,K_{i,j}}$ and $K_{i,j} \in S_{i,K_{i,j}}$
N	the number of channels	mis_S	a maximal independent set of the set S
MIS_S	the set of all maximal independent sets of the set S	$M_{i,2}$	two-hop interfering set of link i
C_i	the maximal conflict set of link i	c_i	a set of the maximal conflict set of link i
$\delta(i)$	the approximation ratio of link i	$\delta(i)'$	the topololgy approximation ratio of link i

TABLE I
NOTATION

II. Related Work

A. Real-time communication in wireless networks

Real-time communication has been extensively studied for industrial wireless networks, and well-known real-time systems scheduling algorithms such as earliest-deadline-first (EDF) and rate-monotonic (RM) have been applied in wireless settings. For instance, Chen et al. [2] and Wu et al. [10] 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. Xu et al. [11] have used EDF and RM scheduling algorithms and studied the corresponding admission control problems by considering different interference models in one-channel settings; without using multiple channel available in typical wireless networks, the solutions tend to suffer from limited throughput and do not consider predictable communication reliability.

There also exist real-time communication solutions that explicitly address the differences between traditional real-time systems and wireless real-time communication. Chipara et al.[12] have proposed fixed-priority scheduling algorithms for real-time flows and have studied the corresponding schedulability test problems. They have also considered interference relations between links. However, the definition of priority in [12] did not consider real-time communication parameters, and the scheduling algorithms cannot guarantee the satisfaction of deadline constraints. In addition, all of the scheduled algorithms are centralized, and they are not amenable to distributed implementation in dynamic network settings. Leng et al. [13] [14] have proposed a harmonic-chain-based method to minimize delay jitter and to satisfy real-time requirement. However, the method did not guarantee communication reliability; it is also difficult for each link to form a harmonic chain in real-world settings, since all the tasks have different period range with different time scale, thus the harmonic chain of all periods would go to infinity. Destounis et al.[15] have considered the probabilistic nature of wireless communication 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.[9] have proposed a centralized scheduling algorithm which is optimal for line networks. But it is difficult to implement the algorithm in distributed, multi-cell settings in practice. Additionally, they did not consider the deadline requirements of individual packets and did not ensure communication reliability. Saifullah et al. [16] [17] have proposed optimal Branch-and-Bound scheduling algorithms for one-hop transmission and considered fixed-priority scheduling for real-time flows in WirelessHart networks. However, the solutions did not consider channel spatial reuse, and the distributed scheduling algorithm for one-hop communications are a heuristic without considering admission control. Gunatilaka et al.[18] have proposed a conservative channel spatial reuse method in order to satisfy real-time and reliability requirements. But the method did not consider the probabilistic nature of wireless transmission and cannot ensure communication reliability.

Long-term real-time guarantees have also been considered in the literature [2]. For instance, Deng et al. [5] have proposed a randomized scheduling algorithm and some heuristics for single-cell settings. Hou et al. [6] have considered long-term real-time packet delivery ratio and have proposed solutions that assign priorities based on long-term delivery debts. Ji et al. [7] have proposed a hybrid algorithm to achieve rate-function delay optimality and throughput optimality in single-cell settings with time-varying channels. Kang et al. [8] have considered long-term timely-throughput and used local pooling factors to chracterize the schedulability test. Ganesan [19] has proposed a distance-d distributed admission control algorithm but did not consider the impact of different scheduling algorithms and per-packet deadline constraints. Focusing on long-term real-time communication requirements, the aforementioned studies have not considered the per-packet real-time requirements by URLLC applications.

Mean delay has been considered in distributed scheduling [20], [21], [22], [23], and age-of-information (AoI) has also been considered in recent studies [24], [25], [26]; however, these work have not considered ensuring predictable timeliness of individual packet transmissions in multi-cell, multi-channel network settings. Li et al. [27] have considered per-node maximum AoI in scheduling, but the study has considered single-cell settings without addressing the challenges of per-node AoI assurance in multi-cell, multi-channel networks.

There also exist channel allocation studies for multi-hop sensor networks. Wu et al. [28] partitioned the whole network into k subtrees and assign one unique channel to each subtree such that the data collection traffic from sensor nodes to the base station can be transmitted in parallel. However, they didn't consider inter-cell interference nor real-time constraints. Wang et al. [29] designed a disjoint paths search algorithm such that a destination node can collect data from k source nodes via k or more mutually node-disjoint paths each of which uses a different wireless channel but has the same packet delivery deadline. However, they didn't consider periodic traffic with heterogeneous deadline constraints. In addition, both studies focused on the multi-hop wireless sensor network architecture, instead of the cellular network architecture with D2D links which we consider in this paper.

B. Reliability guarantee

Ensuring predictable communication reliability is critical in wireless-networked sensing and control. To guarantee that packets are delivered at requested reliability levels, Zhang et al. [30] have proposed the PRKS scheduling algorithm for wireless sensing and control networks. PRKS is a TDMA-based distributed protocol to enable predictable link reliability based on the Physical-Ratio-K (PRK) interference model [31]. Through a control-theoretic approach, PRKS instantiates the PRK model parameters according to in-situ network and environmental conditions so that each link meets its reliability requirement. In particular, PRKS defines a conflict graph for a given wireless network: a node in the graph denotes a link with data transfer in the network, and a link exists between two nodes in the graph if the corresponding links in the network interfere with each other according to the PRK model. Under the condition that link reliability is ensured, PRKS schedules as many nodes as possible in the conflict graph. Based on the predictable link reliability as ensured by algorithms such as PRKS, this paper develops algorithms for ensuring predictable per-packet real-time communication in multi-cell, multi-channel settings.

III. System Model and Problem Definition

A. Network model

The network consists of m base stations (BSes) and n user equipment (UEs). Each UE links with either a BS or another UE. 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. As in LTE and 5G [32], 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, within a time slot, a transmitter can complete the transmission of one packet. The communication reliability along a link i is p_i, meaning that a packet transmission succeeds in probability p_i in the presence of interference from other concurrent transmissions in the network [33].

For multi-cell URLLC, interference needs to be controlled such that two mutually-interfering links shall not transmit in the same channel and at the same time [33]. The cross-link interference can be modeled as a *conflict graph* $G_c = (V_c, E_c)$, 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. 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 we will show in Section IV, the scheduling algorithm only requires a link to be aware of its local interfering links M_i, and the test of the schedulability of a link i also only requires information about links within two-hop distance from i in G_c. In practice, algorithms such as PRKS and its variants [30], [33], [34] can be used to enable each link i to identify its interfering links M_i in a purely distributed manner; the identified M_i ensures the communication reliability of p_i in the presence of interference, and it serves as an input to the real-time scheduling problem studied in this paper. As an example, Figure 1 shows a conflict graph with 8 nodes, where each node represents a link in the network G. Taking link 1 as an example, $M_1 = \{2, 3, 4, 5\}$. (We will also use Figure 1 to illustrate other concepts in the rest of the paper.)

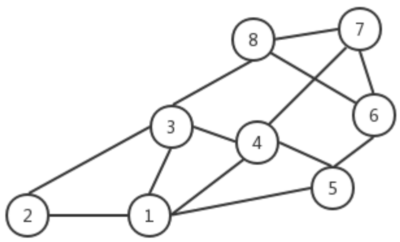


Fig. 1. Example conflict graph

B. Probabilistic real-time traffic model

The data traffic along link i is characterized by a 3-tuple (T_i, D_i, P_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. A packet arriving at time slot t should be successfully delivered no later than time slot $t + D_i$; otherwise, the packet is dropped. Since new packets with new information (e.g., sensing data or control signals) are generated every T_i time slots, we assume $D_i \le T_i$. Unlike Chen et al. [2], we don't assume $D_i = T_i$. Thus both implicit deadlines and constrained deadlines in classic real-time systems are considered. Our model also covers the cases of heterogeneous deadlines across different links.
- Application requirement P_i: due to inherent dynamics and uncertainties in wireless communication, real-time communication guarantees are probabilistic in nature. We adopt the following concept of probabilistic per-packet real-time guarantee first proposed by Chen el al. [2]:

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

For a packet that needs to be successfully delivered across a link i within deadline D_i and in probability no less than P_i, the requirement can be decomposed into two sub-requirements: 1) successfully delivering the packet in probability no less than P_i, and 2) the time taken to successfully deliver the packet is no more than D_i if it is successfully delivered [2]. 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 $X_i = \lceil \log_{1-p_i}(1 - P_i) \rceil$ [2]. Then, the second sub-requirement requires that these X_i transmission opportunities are used within deadline D_i. Note that the X_i reserved time slots do not have to be X_i consecutive time slots, and, for real-time packet delivery, they only have to be before the delivery deadline of the packet. In the above approach, X_i is derived from the channel statistics and the application requirement, thus transforming the probabilistic real-time delivery requirement into a problem of reserving a deterministic number of transmission opportunities for each link. Using X_i, we define the *work density* of link i as $\rho_i = \frac{X_i}{D_i}$.

The above derivation of X_i is based on the link model where the per-packet transmission success probability along link i is p_i, which is a valid model for cases where the per-packet transmission reliability is controlled at a certain level (e.g., p_i) using techniques such as channel-aware transmission power control [35]. For cases where other models such as Markovian models or empirical models of packet transmission successes may be more appropriate (e.g., when no mechanism is adopted to ensure a certain per-packet communication reliability in the presence of channel deep fading), the derivation of X_i needs to be revised accordingly, but the rest of our scheduling framework would still apply. As a first step in studying scheduling with probabilistic per-packet real-time guarantees in multi-cell, multi-channel settings, we focus on scenarios where the per-packet transmission reliability is controlled at p_i, and we relegate the detailed studies of alternative link models as future work.

C. Problem definition

Based on the aforementioned system model, an important question is as follows. Given a network $G = (V, E)$ where each link i has a link reliability p_i and real-time data traffic (T_i, D_i, P_i) $(D_i \le Ti)$, are the set of links schedulable to meet the probabilistic per-packet real-time communication requirements? If yes, develop an algorithm that schedules the data traffic to satisfy the real-time requirements; if not, indicate the infeasibility.

IV. Multi-cell, multi-channel scheduling for probabilistic per-packet real-time guarantee

A. Overview

For single-channel wireless networks with implicit deadlines, Chen et al. [2] 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 [36]. For high-performance multi-channel real-time scheduling, therefore, we turn to optimal multi-processor scheduling for inspiration. In particular, we develop our algorithm based on the idea of deadline partitioning (DP) [37] [36]. 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 [37] 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 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 real-time wireless network scheduling. To this end, we need to address two fundamental differences in multi-cell, multi-channel wireless networks and typical multi-processor systems: *Firstly*, not all the links interfere with one another in multi-cell 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 [30], and thus the probabilistic real-time scheduling problem studied here is also NP-hard. In addition, even though only close-by links interfere with one another [30] 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 schedulable, dynamic, multi-cell URLLC networks require distributed solutions.

Using the conflict graph to model inter-link interference (see Section III-A) and building upon the multi-channel distributed scheduling algorithm Unified Cellular Scheduling (UCS) [33], we observe that 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 URLLC traffic is schedulable as long as the link-local schedulability test is positive. In what follows, we first elaborate on our approach to extending the traditional DP method to local-deadline-partition (LDP) real-time scheduling for multi-cell, multi-channel URLLC, and then we study the associated schedulability test and approximation ratio.

B. Local-deadline-partition (LDP) real-time scheduling

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. Then, we define the *local deadline partition* of a link i as follows:

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* $d''_{i,t} = \min\{\min_{k\in M_i\cup\{i\},D_{k,j}>t} D_{k,j}, \min_{k\in M_i\cup\{i\},A_{k,j}>t} A_{k,j}\}$.

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, then $X''_{i,t} = X_i - X'_{i,t}$ is the remaining work demand of link i at time slot t. Accordingly, 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:*

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

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,*

$$\rho_{i,t} = \frac{X_{i,t}}{L_{i,t} - (t - d'_{i,t})} = \frac{X_{i,t}}{d''_{i,t} - t}. \tag{2}$$

Similar to the original DP-Fair scheduling algorithms [37], [36], the definitions of the local traffic demand and local work density are to ensure steady, proportionate progress towards completing the required workload (i.e., the number of transmissions required for the real-time communication guarantee) within deadlines. The local work density can also be used to prioritize packet transmissions along different links. Unlike traditional real-time systems where the deadline partition (DP) is based on global information (i.e, real-time parameters of all the tasks), however, the local-deadline-partition (LDP) spilts time based only on the information of one-hop links in the conflict graph. Based on these observations, we develop the LDP real-time scheduling algorithm by extending the multi-channel distributed scheduling algorithm Unified Cellular Scheduling (UCS) [33] to consider probabilistic per-packet real-time communication requirements by URLLC applications. In particular, at the beginning time slot t of each local deadline partition, the transmitter and receiver of a link i sets its local traffic demand $X_{i,t}$ according to the definition. Each link will execute the following procedure in a distributed manner in the algorithm:

1) The transmitter and receiver of each link $i \in E$ initializes its state as UNDECIDED for each channel $rb \in RB$, and

sets the state of every link l, $l \in M_i$, as UNDECIDED for each channel $rb \in RB$;

2) For each channel $rb \in RB$, both transmitter and receiver of link i computes a priority based on its local work density.

$$Prio.i.rb = \begin{cases} \rho_{i,t} & \rho_{i,t} < 1 \\ 1 & \rho_{i,t} \geq 1 \end{cases} \tag{3}$$

3) Both 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: A) For each channel rb in which the state of link i is UNDECIDED, if i's priority is higher than that of every other UNDECIDED and ACTIVE member of M_i, the state of i in channel rb is set as ACTIVE, and its remaining local work load $R_{i,t}$ is reduced by one; if any ACTIVE member of M_i has a higher priority than i, the state of i in channel rb is set as INACTIVE; if i's priority equals those of all the links in M_i, then the tie is broken based on link IDs, with the link of the largest ID becoming ACTIVE and the other links becoming INACTIVE; if $X_{i,t}$ becomes zero, i's state is set as INACTIVE for each channel in which its state is UNDECIDED; B) 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. C) The transmitter/receiver of link i update the state of link l ($\forall l \in M_i$), if it receives a state update about l.
4) 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 slot t.

The details of the above local-deadline-partition (LDP) scheduling algorithm for time slot t are shown in Algorithm 1. Similar to the UCS algorithm [33], Algorithm 1 can be readily 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 ACTIVE/INACTIVE based on lines 20 and 21 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 line 14, 15 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. ☐

Note that, for the simplicity of discussion, the presentation of Algorithm 1 assumes that the distributed coordination between links/nodes occur at the beginning of time slot t. In practice, it may take several rounds of coordination between links/nodes for the LDP algorithm to converge. If this coordination delay is too large, we can use the method of pipelined pre-computation [38] which pre-computes scheduling results by R slots ahead of time, where R is an upper bound on the number of rounds taken by the LDP algorithm to converge.

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;
$d''_{i,t}$: the local deadline of the current deadline partition;
Output: Perform the following actions:
1: state.i.rb = UNDECIDED, $\forall rb \in RB$
2: done = false;
3: **while** done == false **do**
4: done = true;
5: $\rho_{i,t} = X_{i,t}/(d''_{i,t} - t)$;
6: $Prio.i.rb = \min(\rho_{i,t}, 1)$, $\forall rb \in RB$;
7: **for** each $rb \in RB$ in increasing order of rb ID **do**
8: **if** $X_{i,t} > 0$ **and** state.i.rb == UNDECIDED **and** ((Prio.i.rb $>$ Prio.l.rb) **or** (Prio.i.rb $=$ Prio.l.rb **and** ID.i $>$ ID.l)) for each ACTIVE/UNDECIDED $l \in M_i$ **then**
9: state.i.rb = ACTIVE;
10: $X_{i,t}$ = $X_{i,t}$ - 1;
11: $\rho_{i,t} = X_{i,t}/(d''_{i,t} - t)$;
12: $Prio.i.rb = \min(\rho_{i,t}, 1)$, $\forall rb \in RB$;
13: **end if**
14: **if** ((Prio.i.rb $<$ Prio.l.rb) **or** (Prio.i.rb $=$ Prio.l.rb **and** ID.i $<$ ID.l)) for any ACTIVE $l \in M_i$ **then**
15: state.i.rb = INACTIVE;
16: **end if**
17: **if** $X_{i,t}$ == 0 **and** state.i.rb == UNDECIDED **then**
18: state.i.rb = INACTIVE;
19: **end if**
20: **if** state.i.rb == UNDECIDED **then**
21: done = false;
22: **end if**
23: **end for**
24: Share state.i.rb, $\forall rb \in RB$;
25: Update state.l.rb, $\forall l \in M_i \cup \{i\}$;
26: **end while**

More specifically, in time slot t, each node starts executing the scheduling algorithm for a future slot $(t + R)$, and only one round of coordination is executed in a single time slot. When it is time slot $(t+R)$, a node simply looks up its precomputed state and decides to become active or not.

C. Schedulability test

Given an arbitrary network G, it is not always possible to find a schedule to meet the probabilistic per-packet real-time communication 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, one approach to schedulability analysis of link

i is by jointly considering the URLLC traffic demand of link i and those of the links in M_i. Nonetheless, not every two links in M_i interfere with each other, and those links can be active in the same wireless channel and at the same time. So, an alternative approach is to jointly consider the URLLC traffic demand of the links in each clique $K_{i,j} \subseteq M_i \cup \{i\}$. For each 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). Therefore, we propose an approach that, for a given link i, jointly considers the URLLC 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 $M_i \cup \{i\}$. 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 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 $\mathbb{K}_i = \{clique \;\; K_{i,j'} : i \in K_{i,j'} \wedge K_{i,j'} \subseteq M_i \cup \{i\} \wedge K_{i,j'} \subseteq G_c\}$, 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 1 and the links in $M_1 \cup 1$, there are 3 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 URLLC 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 clique $K_{i,j} \subseteq M_i \cup \{i\}$, there exists at least one feasible set.*

Proof. We can let every link l in M_i inactive and link i active on all the channels, since every link j in $E \setminus \{M_i \cup \{i\}\}$ does not conflict with link i. Then, based on the definition of minimum scheduling rate, $M_i \cup i$ is a feasible set for any clique $K_{i,j}$, $K_{i,j} \subseteq M_i \cup \{i\}$. □

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 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 $S_{i,K_{i,j}} \subseteq M_i \cup i$,*

$$\sum_{l \in S_{i,K_{i,j}}} \rho_{l,t-1} \geq N + 1. \tag{4}$$

Proof. We prove this by contradiction. Suppose at time slot t, link i is not schedulable and, at time slot $t-1$, the sum of the local work density of at least one feasible set $S_{i,K_{i,j}}$ is less than $N+1$ and suppose $X_{i,t-1} > 0$. Since $d''_{i,t-1} - (t-1) = 1$ at time slot $t-1$ and the work demand is an integer, $1 \leq \rho_{i,t-1} = X_{i,t-1} \leq N$. This also implies that, for the feasible set $S_{i,K_{i,j}} \subseteq M_i \cup \{i\}$, there are at most $N - X_{i,t-1}$ links whose local work density equals 1, since $\rho_{i,t-1} + \sum_{l \in S_{i,K_{i,j}} \setminus i} \rho_{l,t-1} < N + 1$. For each channel $rb \in RB$ and the feasible set $S_{i,K_{i,j}}$, there will be at least one active link $l \in S_{i,K_{i,j}}$. In addition, Algorithm 1 will let the link $l' \in M_i \cup \{i\}$ with the highest priority (whose local work density is greater than or equal to 1) be active. Therefore, each link $l \in S_{i,K_{i,j}}$ with the highest priority which is equal to 1 can be scheduled. Then, link i will be active and be assigned with $X_{i,t-1}$ number of channels, and, by Definition 5 on Minimum Scheduling Rate, this holds no matter how the links other than those of $S_{i,K_{i,j}}$ are scheduled. Thus link i is schedulable at time t, which is a contradiction.

□

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

Lemma 3. *Given a link i, if, for every 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.*

Proof. According to Definition 2 on local deadline partitioning, each link $l \in M_i \cup \{i\}$ will choose the maximum value of the arrival time and deadline from the links in $M_l \cup l$ before time slot t as $d'_{l,t}$, and choose the minimum value of the arrival time and deadline from the links in $M_l \cup l$ after time slot t as $d''_{l,t}$, where $d'_{l,t}$and $d''_{l,t}$ are the starting time and local deadline for $\sigma_{l,t}$ respectively. This implies that, for every link $l \in M_i$ and every time slot t in the period $[A_{i,p}, D_{i,p})$ associated with the p-th packet at link i, $d'_{l,t} \geq A_{i,p}$ and $d''_{l,t} \leq D_{i,p}$. In addition, at $t_0 = A_{i,p}$, d'_{l,t_0} is the same for every link $l \in M_i \cup \{i\}$, and it is $A_{i,p}$; at $t_1 = D_{i,p} - 1$, d''_{l,t_1} is the same for every link $l \in M_i \cup \{i\}$, and it is $D_{i,p}$; the time slice $[A_{i,p}, D_{i,p})$ may include multiple deadline partitions for every link $l \in M_i \cup \{i\}$.

For the feasible set $S_{i,K_{i,j}}$, we have

$$\sum_{l \in S_{i,K_{i,j}}} \rho_l = \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l} \leq N. \tag{5}$$

Then we consider the total work demand (i.e., total number transmission opportunities required) for the links in $S_{i,K_{i,j}}$ during the interval $[A_{i,1}, D_{i,1})$. At time slot $t_1 = A_{i,1}$, every link $l \in S_{i,K_{i,j}}$ shares the same local arrival time $d'_{l,t_1} = A_{i,1}$, and, at time slot $t_2 = D_{i,1} - 1$, every $l \in S_{i,K_{i,j}}$ shares the same local deadline $d''_{l,t_2} = D_{i,1}$. Then, according to the proportionate allocation rule of the LDP scheduling algorithm (i.e., Algorithm 1), during the interval $[A_{i,1}, D_{i,1})$, we have

$$W_{l,A_{i,1}} = \sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1})} \rho_l \times L_{l,t}, \tag{6}$$

such that $W_{l,A_{i,1}}$ is the total work demand for link l in $[A_{i,1}, D_{i,1})$. If a link l has a constrained deadline (i.e., $D_i < T_i$), we have

$$\sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1})} L_{l,t} \leq D_{i,1} - A_{i,1}. \tag{7}$$

If link l has an implicit deadline (i.e., $D_i = T_i$), we have

$$\sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1})} L_{l,t} = D_{i,1} - A_{i,1}. \tag{8}$$

Therefore, for every link $l \in S_{i,K_{i,j}}$, we have

$$\begin{aligned} W_{l,A_{i,1}} &= \sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1})} \rho_l \times L_{l,t} \\ &\leq (D_{i,1} - A_{i,1}) \times \rho_l \end{aligned} \tag{9}$$

Then, we can get,

$$\begin{aligned} \sum_{l \in S_{i,K_{i,j}}} W_{l,A_{i,1}} &\leq (D_{i,1} - A_{i,1}) \times \sum_{l \in S_{i,K_{i,j}}} \rho_l \\ &\leq (D_{i,1} - A_{i,1}) \times N. \end{aligned} \tag{10}$$

Then we consider time slot $t_2 = D_{i,1} - 1$. Since at time t_2, the length of local deadline partition for all the links in $S_{i,K_{i,j}}$ is the same, the local work density can be shown as follows,

$$\sum_{l \in S_{i,K_{i,j}}} \rho_{l,t_2} = \sum_{l \in S_{i,K_{i,j}}} \frac{W_{l,A_{i,1}} - C_{l,t_2}}{D_{i,1} - t_2}, \tag{11}$$

where C_{l,t_2} is the number of transmission opportunities that have been assigned to link l in time slice $[A_{i,1}, D_{i,1} - 1)$. We know that,

$$\begin{aligned} &\sum_{l \in S_{i,K_{i,j}}} (W_{l,A_{i,1}} - C_{l,t_2}) \\ &\leq (D_{i,1} - A_{i,1}) \sum_{l \in S_{i,K_{i,j}}} \rho_l - \sum_{l \in S_{i,K_{i,j}}} C_{l,t_2} \\ &\leq (D_{i,1} - A_{i,1})N - \sum_{l \in S_{i,K_{i,p}}} C_{l,t_2}. \end{aligned} \tag{12}$$

Based on the definition of feasible sets, we also have

$$\sum_{l \in S_{i,K_{i,p}}} C_{l,t_2} \geq N. \tag{13}$$

Thus,

$$\begin{aligned} &(D_{i,1} - A_{i,1})N - \sum_{l \in S_{i,K_{i,p}}} C_{l,t_2} \\ &\leq (D_{i,1} - A_{i,1})N - (D_{i,1} - 1 - A_{i,1})N \\ &\leq N. \end{aligned} \tag{14}$$

Therefore, based on (11), (12), (14), we can get

$$\begin{aligned} \sum_{l \in S_{i,K_{i,j}}} \rho_{l,t_2} &= \sum_{l \in S_{i,K_{i,j}}} \frac{W_{l,A_{i,1}} - C_{l,t_2}}{D_{i,1} - t_2} \\ &\leq \sum_{l \in S_{i,K_{i,j}}} \frac{(D_{i,1} - A_{i,1})N - \sum_{l \in S_{i,K_{i,p}}} C_{l,t_2}}{D_{i,1} - t_2} \\ &\leq \sum_{l \in S_{i,K_{i,j}}} \frac{N}{D_{i,1} - t_2} = N \end{aligned} \tag{15}$$

Therefore, according to Lemma 2, link i does not miss its deadline for the first packet, that is, the transmissions of the first packet is completed by $D_{i,1}$. Then according to the proportionate allocation rule of the LDP algorithm, Equation 10 also holds for the second packet period of link i, $[A_{i,2}, D_{i,2})$, and, based on the same analysis, this lemma holds at time slot $D_{i,2} - 1$. By induction, the lemma also holds for any time slot $D_{i,p} - 1, p \geq 3$. □

Based on Lemma 3, we now derive the schedulability condition as follows:

Theorem 2 (Schedulability Condition)**.** *Given a link i and the conflict graph G_c, let $\mathbb{K}_i$ denote the set of 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*

$$\min_{S_{i,K_{i,j}} \in \mathbb{S}_{i,K_{i,j}}} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l} \leq N, \tag{16}$$

then the probabilistic real-time traffic of link i can be supported, that is, link i is schedulable.

Proof. Based on Lemma 3, we only need to show that, for every clique $K_{i,j} \in \mathbb{K}_i$, there exists a feasible set $S_{i,K_{i,j}}$ whose sum work density is no more than N. This would hold if, for every clique $K_{i,j} \in \mathbb{K}_i$ and the set $\mathbb{S}_{i,K_{i,j}}$ of feasible sets for link i and $K_{i,j}$, the feasible set with the minimum sum work density has a sum work density no more than N. Hence this theorem holds. □

From Theorem 2, we see that, to decide whether a link i is schedulable, we just need to identify the feasible set with the minimum sum work density and check whether its sum work density is no more than the number of channels N. To this end, we need an approach to identifying all the feasible sets of interest. By 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. Fortunately, 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 a unique property 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.*

Theorem 3. *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.*

Proof. According to Definition 6 on feasible sets, we first need to show that there exists at least one maximal independent set (MIS) of G_c whose intersection with S_i has only one element. To this end, note that any MIS mis_{G_c} that includes link i as an element will not include any link from M_i, thus mis_{G_c} will not include any link from $S_i \setminus \{i\}$. Therefore, for any MIS mis_{G_c} such that $i \in mis_{G_c}$, $mis_{G_c} \cap S_i$ includes one and only one element i.

Next, we need to show that there is no MIS of G_c whose intersection with S_i is empty. This trivially holds when $M'_i = \emptyset$. When $M'_i \neq \emptyset$, for each $mis \in MIS_{M'_i}$, there must exist a MIS of G_c, denoted by mis_{G_c}, that includes mis as a subset. In this case, $mis_{G_c} \cap S_i$ is not empty if and only if there is a link in S_i that does not interfere with any link in mis. Of course, a MIS of G_c may only include as a subset a non-maximal independent set of M'_i, denoted by mis'; in this case, there will exist a link in S_i that does not interfere with any link in mis', if, for for each $mis \in MIS_{M'_i}$ (which includes the mis that is a superset of mis'), there exists at least one link in S_i that does not interfere with any link in mis. Therefore, there is no MIS of G_c whose intersection with S_i is empty, if and only if, for 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. Hence Theorem 3 holds.

□

For the 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.

Based on the aforementioned property of feasible sets and Definitions 7, we develop Algorithm 2 for schedulability test. A key part of the schedulability test is to search the set of the feasible sets that include a specific clique $K_{i,j}$ as a subset and that may have the minimum sum work density. Given that a feasible set is the combination of multiple cliques, we start from $K_{i,j}$ and extend it by combining near-by cliques.

Algorithm 2 Schedulability Test

Input: G_c: conflict graph of the network;
N: the number of channels;
$\mathbb{K}_i$: the set of 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, T_l, D_l: traffic demand, period, and relative deadline of link $l \in M_i \cup \{i\}$;
Output: whether link i is schedulable;
for each clique $K_{i,j} \in \mathbb{K}_i$ **do**
  $S_{i,K_{i,j}} = K_{i,j}$;
  $exclude = \emptyset$;
  $U_{i,K_{i,j}} = \sum_{l \in M_i \cup \{i\}} \frac{X_l}{D_l}$;
  SearchFeasibleSets($S_{i,K_{i,j}}, G_c$);
end for
if $U_{i,K_{i,j}} \leq N, \forall K_{i,j} \in \mathbb{K}_i$ **then**
  link i is schedulable;
else
  link i is not schedulable;
end if

Algorithm 3 SearchFeasibleSets

Input: $S_{i,K_{i,j}}$: feasible set candidate;
G_c: conflict graph of the network;
Output: minimum sum work density of any feasible set including $K_{i,j}$ as a subset;
if $S_{i,K_{i,j}}$ is a feasible set according to Theorem 3 **then**
  $U_{i,K_{i,j}} = \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l}$;
else
  for each $K_{i,p} \in \mathbb{K}_i \setminus exclude$ **do**
    if $\sum_{l \in K_{i,p} \cup S_{i,K_{i,j}}} \frac{X_l}{D_l} < U_{i,K_{i,j}}$ **then**
      $S_{i,K_{i,j}} = S_{i,K_{i,j}} \cup K_{i,p}$;
      $exclude = exclude \cup K_{i,p}$;
      $SearchFeasibleSets(S_{i,K_{i,j}}, G_c)$;
      $exclude = exclude \setminus K_{i,p}$;
      $S_{i,K_{i,j}} = S_{i,K_{i,j}} \setminus K_{i,p}$;
    end if
  end for
end if

In Algorithm 2, after the initialization of the feasible set with $K_{i,j}$, we use Algorithm 3 to find the minimum sum work density, denoted by $U_{i,K_{i,j}}$, of any feasible set that includes clique $K_{i,j}$ as a subset. Since $M_i \cup \{i\}$ is the largest feasible set and has the largest sum work density, Algorithm 2 initializes $U_{i,K_{i,j}}$ as the sum work density of $M_i \cup \{i\}$. In Algorithm 3, if the current set $S_{i,K_{i,j}}$ is a feasible set, Algorithm 3 stops searching for additional feasible sets that include the current set as a subset since any such additional feasible set will have a larger sum work density; on the other hand, if the current set $S_{i,K_{i,j}}$ is not a feasible set (i.e., via the testing at Line 1), the algorithm searches for additional feasible sets by exploring the addition of new cliques that have not been added to the current set yet, until the algorithm has explored all the feasible sets that include $K_{i,j}$ as a subset and may potentially have the minimum sum work density. By Line 4 of Algorithm 3, a new clique $K_{i,p}$ is added to the current set only if the resulting sum work density is less than the minimum sum work density of all the feasible sets having been explored so far in the algorithm execution. This way, the algorithm can find the minimum sum work density of the feasible sets that include $K_{i,j}$ as a subset. After finding the minimum sum work density of any feasible set $S_{i,K_{i,j}}$ for every clique $K_{i,j} \in \mathbb{K}_i$, we can perform the schedulability test based on Theorem 2.

Note that, even though the schedulability test involves finding cliques which is a non-trivial problem in general, the complexity of the problem is significantly reduced because the test only needs to search for cliques in a small graph induced by $M_i \cup M_{i,2}$ and we only need to find the feasible set with the minimum sum of work density.

D. Optimality analysis

Given that the multi-cell, multi-channel real-time scheduling problem studied in this work is NP-hard, the LDP algorithm and the associated schedulability test are expected to be sub-optimal. As a first step towards understanding the optimality of the LDP algorithm and schedulability test, here we develop a necessary condition for URLLC schedulability and use it to derive a lower bound on the approximation ratio of LDP scheduling.

Theorem 4 (Necessary Condition for URLLC Schedulability). *Given a link i and the conflict graph G_c, let $\mathbb{K}_i$ denote the set of 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*

$$\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l} \leq N. \tag{17}$$

Proof. For any clique $K_{i,j} \in \mathbb{K}_i$, the maximum scheduling rate for each time slot is equal to the number of channels N. Therefore, the total utilization of the links of any clique shall be no more than N, where the utilization of a link l is defined as $\frac{X_l}{T_l}$. Thus the theorem holds. $\square$

Based on Theorems 2 and 4, we can explore the gap between the sufficient condition (16) and necessary condition (17). 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 (17) to that of the sufficient condition (16). That is,

$$\delta(i) = \frac{\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l}}{\max_{K_{i,j} \in \mathbb{K}_i} \min_{S_{i,K_{i,j}} \in \mathbb{S}_{i,K_{i,j}}} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l}}. \tag{18}$$

This lower bound depends on two factors: URLLC traffic and network topology. Typically, 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 also give 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

$$\delta(i)' = \frac{|k'_{max,i,j}|}{|s_{max,i}|} \tag{19}$$

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

V. Experimental Study

In what follows, we evaluate the properties of the LDP scheduling algorithm and the schedulability test algorithm in multi-cell URLLC networks.

A. Network settings

We consider two networks of different sizes, as shown in Figures 2 and 3. For Network 1, we uniform-randomly deploy 91 wireless nodes in a 1200×1200 square-meter region, generating a network of 83 links. There are nine cells which are organized in a 3×3 grid manner such that each cell covers a 400×400 square-meter region. There is a base station (BS) within each cell. For Network 2, we uniform-randomly deploy 151 wireless nodes in a 1200×1500 square-meter region, generating a network of 163 links. There are 12 cells which are organized in a 3×4 grid manner such that each cell covers a 400×375 square-meter region. The number of channels considered ranges from 3 to 10. 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.

Both networks have uplinks from UEs to BSes, downlinks from BSes to UEs, and UE-to-UE (i.e., D2D) communication links. The lengths of the uplinks and D2D communication links vary from 50 to 100 meters, while the lengths of the downlinks vary from 100 to 200 meters. Given a link i, an exclusion region is defined as a circular area centered around the receiver of i, and the radius of the exclusion region is r times the length of i where r is uniform-randomly selected from $[1.5, 2]$. Links i and j are regarded as interfering with each other and thus $(i, j) \in G_c$ if the transmitter of j (or i) lies in the exclusion region of i (or j) [31]. For Network 1, the maximum and average number of interfering links for a link are 18 and 11.62 respectively. Network 2 has higher node spatial distribution density and higher degree of cross-link interference, such that the maximum and average number of interfering links for a link is 32 and 20.78 respectively.

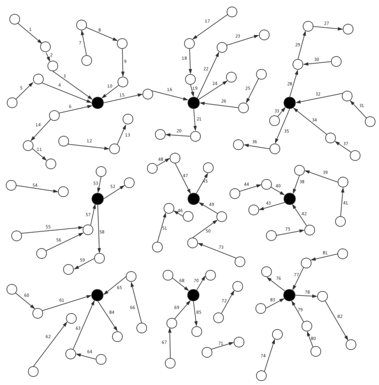


Fig. 2. Network 1

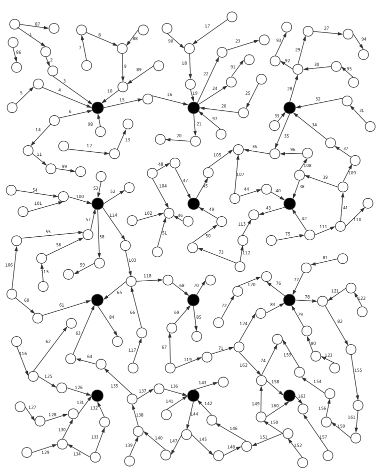
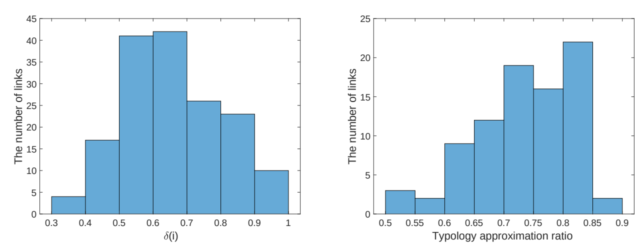


Fig. 3. Network 2

B. Experimental results

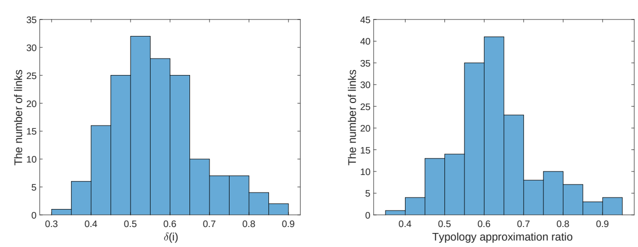
Numerical study of LDP. Here we evaluate the lower bound on the approximation ratio of the LDP scheduling algorithm for Networks 1 and 2. We randomly choose the real-time traffic parameters such that all the links can pass the schedulability test. The work demand X_i (i.e., required number of transmission opportunities per packet) along a link i is uniform-randomly chosen from $[2, \min(T_i - 1, 10)]$ such that the work density of each link i is less than 1. The period is greater than or euqal to the relative deadline. The relative deadline uniform-randomly ranges from 6 to 30, while the difference between a period and a relative deadline uniform-randomly ranges from 0 to 8.

Figure 4(a) and Figure 5(a) are drawn from Equation 18, and Figure 4(b) and Figure 5(b) are drawn from Equation 19. In particular, Figure 4(a) shows the histogram of the approximation ratio low bound $\delta(i)$ for all the links in Network 1. The mean value is 0.6628, and its 95 % confidence interval is $[0.6409, 0.6846]$. Figure 4(b) shows the histogram of the topology approximation ratio in Network 1. The mean value is 0.7294, and its 95 % confidence interval is $[0.7112, 0.7475]$. We see that network topology has significant impact on the approximation ratio, even though the URLLC traffic pattern also impacts the approximation ratio. Figure 5(a) shows the histogram of the approximation ratio lower bound $\delta(i)$ for all the links in Network 2. The mean approximation ratio is 0.5675, and its 95 % confidence interval is $[0.5506, 0.5843]$. Figure 5(b) shows the histogram of the topology approximation ratio in Network 2. The mean ratio is 0.6275, and its 95 % confidence interval is $[0.6104, 0.6447]$. We see that the approximation ratio lower bound in Network 2 is about 10% lower than that in Network 1. This is because the number of interfering links per link in Network 2 tends to be higher than that in Network 1. Accordingly, the size of cliques in the conflict graph of Network 2 is greater than that of Network 1, which makes the approximation ratio lower bound lower in Network 2.



(a) Approximation ratio lower bound (b) Topology approximation ratio

Fig. 4. Numerical results for Network 1



(a) Approximation ratio lower bound (b) Topology approximation ratio

Fig. 5. Numerical results for Network 2

Note that the approximation ratio lower bounds presented above are the lower bound on the performance of the LDP scheduling algorithm. How to tighten the lower bound to characterize more precisely the benefits of using the LDP algorithm will be an interesting topic for future studies. In what is next, we experimentally compare the perforance of the LDP algorithm with other state-of-the-art algorithms.

Comparative study. Here we comparatively study the performance of the LDP and G-schedule scheduling algorithm [9]. To this end, we implement both of the algorithms in Matlab and study their behavior in Network 2. To understand the benefits of using the LDP algorithm, we consider demanding real-time traffic whose work density is close to network capacity but can still be supported by the LDP algorithm. Figure 6 show the sum of work density in the feasible set when the number of channels is euqal to 5. Then we characterize the feasibility of supporting the real-time traffic using the G-schedule algorithms. The range of the relative deadline, as shown in figure 7(a), is from 6 to 30. The difference between a period and a relative deadline uniform-randomly ranges from 0 to 8. The work demand X_i is chosen from $[2, \min(T_i - 1, 10)]$ and the work density of each link is shown in figure 7(b).

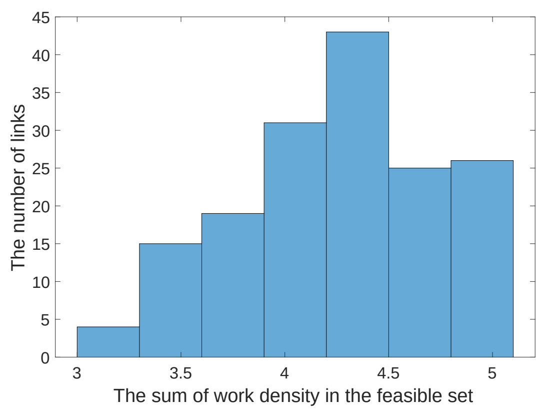
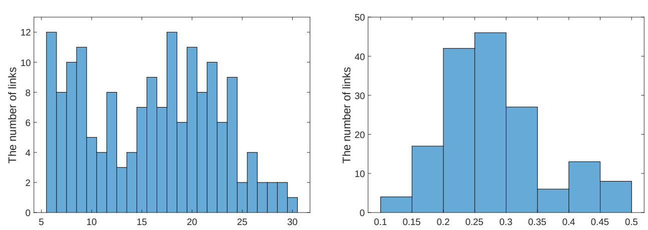


Fig. 6. The sum of work density in the feasible set



(a) The relative deadline

(b) The work density of each link

Fig. 7. Real-time traffic

We vary the number of channels from 3 to 10, but keep the conflict graph G_c unchanged. 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 the ratio of schedulable links increases with the number of channels. However, the ratio of schedulable links in the G-algorithm drops significantly as the number of channels decreases, for instance, being less than 65% when the number of channels is 3. In addition, the links with shorter

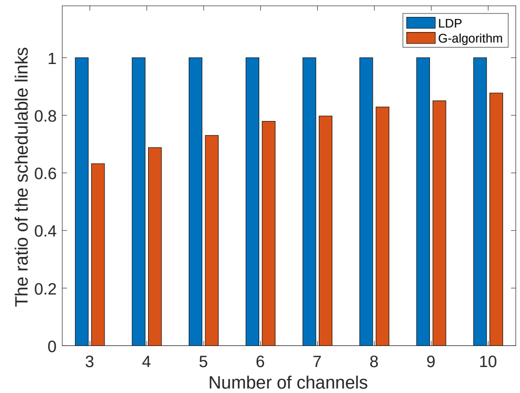


Fig. 8. Comparison with G-algorthm

deadlines are more likely to become unschedulable in the G-schedule. This is because the G-schedule algorithm greedily schedules links without considering heterogeneous deadline constraints, and the links with shorter deadlines are likely to be assigned with fewer transmission opportunities before their deadlines. On the other hand, the LDP algorithm dynamically updates packets' priorities based on in-situ work densities, and the links with higher work demand and closer to their absolute deadlines tend to get higher priorities. Accordingly, the LDP algorithm can support more demanding real-time traffic requirements than the G-schedule algorithm does.

VI. Concluding Remarks

We have proposed a distributed local-deadline-partition (LDP) scheduling algorithm to ensure probabilistic per-packet real-time guarantee for multi-cell, multi-channel URLLC. The LDP algorithm effectively leverages the two-hop information in the conflict graph and addresses the challenges of multi-channel and multi-cell real time scheduling in URLLC wireless networks. The concept of feasible set in this paper bridges traditional real time systems and real-time wireless communication. Leveraging the feasible set concept, we have identified a closed-form sufficient condition for schedulability test. Based on the properties of feasible sets in wireless networks, we have developed an algorithm for finding the minimum sum work density of feasible sets, upon which we have developed the shedulability test algorithm. Our experimental results have shown that the LDP algorithm can support significantly more URLLC traffic than state-of-the-art solutions.

This study represents a first step towards ensuring probabilistic per-packet real-time guarantee in multi-cell, multi-channel URLLC, and it serves as a foundation for exploring other interesting studies. For instance, to generate field-deployable URLLC systems, it will be worthwhile to implement and integrate the LDP scheduling algorithm with PRKS [30] in emerging open-source 5G platforms such as

OpenAirInterface [39]. Another interesting direction is to consider delay jitter control since URLLC applications such as AR/VR tend to require as small delay jitter as possible.

REFERENCES

[1] A. Gosain, M. Berman, M. Brinn, T. Mitchell, C. Li, Y. Wang, H. Jin, J. Hua, and H. Zhang, "Enabling campus edge computing using geni racks and mobile resources," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*.

[2] 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.

[3] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: current status and future perspectives toward 5g systems," *IEEE Communications Magazine*, vol. 53, no. 9.

[4] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolié, "Design of a low-latency, high-reliability wireless communication system for control applications," in *2014 IEEE International conference on communications (ICC)*.

[5] L. Deng, C. Wang, M. Chen, and S. Zhao, "Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6.

[6] I. Hou, V. S. Borkar, and P. R. Kumar, "A theory of qos for wireless," in *INFOCOM 2009*.

[7] B. Ji, G. R. Gupta, X. Lin, and N. B. Shroff, "Low-complexity scheduling policies for achieving throughput and asymptotic delay optimality in multichannel wireless networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6.

[8] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, "On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1.

[9] A. Tan, Q. Wang, N. Guan, Q. Deng, and X. S. Hu, "Inter-cell channel time-slot scheduling for multichannel multiradio cellular fieldbuses," in *RTSS 2015*.

[10] 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,*.

[11] X. Xu, X. Li, P. Wan, and S. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3.

[12] O. Chipara, C. Lu, and G. Roman, "Real-time query scheduling for wireless sensor networks," in *Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS 2007)*.

[13] Q. Leng, Y. Wei, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "Improving control performance by minimizing jitter in rt-wifi networks," in *RTSS 2014*.

[14] Q. Leng, W. Chen, P. Huang, Y. Wei, A. K. Mok, and S. Han, "Network management of multicluster rt-wifi networks," *ACM Trans. Sens. Networks*, vol. 15, no. 1.

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

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

[17] 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 *2015 IEEE Real-Time Systems Symposium, RTSS 2015*.

[18] D. Gunatilaka and C. Lu, "Conservative channel reuse in real-time industrial wireless sensor-actuator networks," in *ICDCS 2018*.

[19] A. Ganesan, "Performance Guarantees of Distributed Algorithms for QoS in Wireless Ad Hoc Networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1.

[20] 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.

[21] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *IEEE INFOCOM*, 2010.

[22] D. Lee, D. Yun, J. Shin, Y. Yi, and S.-Y. Yun, "Provable per-link delay-optimal CSMA for general wireless network topology," in *IEEE INFOCOM*, 2014.

[23] K. Kar, S. Sarkar, A. Ghavami, and X. Luo, "Delay guarantees for throughput-optimal wireless link scheduling," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2906–2911, 2012.

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

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

[26] Y. C. B. J. Y. H. C. Li, Y. Huang and W. Lou, "Kronos: A 5G Scheduler for AoI Minimization under Dynamic Channel Conditions," in *IEEE ICDCS*, 2019.

[27] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, "AoI Scheduling with Maximum Thresholds," in *IEEE INFOCOM*, 2020.

[28] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *IEEE INFOCOM 2008*. IEEE, 2008.

[29] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, "Flow-based real-time communication in multi-channel wireless sensor networks," in *European Conference on Wireless Sensor Networks*. Springer, 2009.

[30] 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.

[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] E. Dahlman, S. Parkvall, and J. Skold, *4G, LTE-Advanced Pro and The Road to 5G*. Academic Press, 2016.

[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] 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.

[35] L. Wang, H. Zhang, and P. Ren, "Distributed scheduling and power control for predictable iot communication reliability," in *IEEE ICC*, 2018.

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

[37] G. Levin, S. Funk, C. Sadowski, I. Pye, and S. Brandt, "DP-FAIR: A simple model for understanding optimal multiprocessor scheduling," in *2010 Euromicro Conference on Real-Time Systems*.

[38] X. Liu, Y. Chen, and H. Zhang, "A maximal concurrency and low latency distributed scheduling protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks (Hindawi)*, no. 603172, 2015.

[39] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.