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

Zhibo Meng, Hongwei Zhang

Abstract

For mission-critical sensing and control applications such as those to be enabled by 5G Ultra-Reliable, Low-Latency Communications (URLLC), it is critical 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 with implicit deadline constraints, but they have not considered real-world complexities such as inter-cell interference and multiple communication channels. Towards ensuring PPRC 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 PPRC traffic. In particular, we propose the concept of feasible set and identify a closed-form sufficient condition for the schedulability of PPRC 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 PPRC 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 PPRC 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, PPRC, probabilistic per-packet real-time guarantee.

This work is supported in part by the NSF awards 2112606, 2130889, and 1827211 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
I. INTRODUCTION

Wireless networks are increasingly being explored for mission-critical sensing and control applications. 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. Ultra-Reliable, Low-Latency Communications (URLLC) for mission-critical sensing and control has also become a major focus of 5G-and-beyond 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-type sensing and control applications [2]. For example, in Extended Reality (XR) applications [4], 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]. In networked feedback control, consecutive packet loss may well lead to system instability and cause safety concerns; in addition, a packet of sample data will be dropped if the packet has not been successfully delivered by the time a new sample data is collected, and, in this case, the probabilistic guarantee of real-time delivery of each packet enables the modeling of the packet drop process as a random sampling process, thus facilitating the use of random sampling theories to characterize the impact of probabilistic wireless communication on networked control and facilitating the joint design of wireless networking and control [2].

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 Probabilistic Per-packet Real-time Communications (PPRC) guarantee in multi-cell, 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.

Contributions. To address the aforementioned challenges and to provide Probabilistic Per-packet Real-time Communications (PPRC) guarantees in multi-cell, multi-channel wireless networks, 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 PPRC guarantee in multi-cell, multi-channel networks with general deadline constraints.
- For the schedulability test of PPRC traffic, we propose the concept of feasible set and identify a closed-form sufficient condition for the schedulability of PPRC 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 PPRC 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 PPRC 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 PPRC requirement of a large network 33.1% 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 III. 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.

| Table I: Notation |
|-------------------|
| $T_i$ | 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 $i$ |
| $\rho_i$ | 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 $K_{i,j} \subseteq M_i \cup i$ |
| $\mathcal{K}_i$ | the set of $K_{i,j}$ |
| $U_{K_{i,j}}$ | a union of cliques that $K_{i,j} \subseteq U_{K_{i,j}}$ and $U_{K_{i,j}} \subseteq \mathcal{K}_i$ |
| $S_{i,K_{i,j}}$ | the feasible set that $i \in S_{i,K_{i,j}}$ and $K_{i,j} \subseteq S_{i,K_{i,j}}$ |
| $M_{i,2}$ | two-hop interfering set of link $i$ |
| $N$ | 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 $S$ |
| $C_i$ | 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 topology approximation ratio of link $i$ |

II. RELATED WORK

A. Real-time communications in wireless networks

Real-time communications have 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. Real-time scheduling in WirelessHART industrial wireless networks has been extensively studied [11], [12], [13], [14], [15]. These studies consider multi-hop wireless sensor-actuator networks, and they do not consider multi-cell cellular networks. They use EDF, deadline-monotonic, or fixed-priority scheduling algorithms which under-perform in multi-channel settings [11], [12], [14], [15], or they do not consider channel spatial reuse [13]. Xu et al. [16] 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. Gunatilaka et al. [17] 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 communications and cannot ensure 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. [18] 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 [18] did not consider real-time communication parameters, and the scheduling algorithms cannot guarantee the satisfaction of deadline constraints. In addition, these scheduling algorithms are centralized, and they are not amenable to distributed implementation in dynamic network settings. Leng et al. [19] [20] have proposed a harmonic-chain-based method to minimize delay jitter and 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 different tasks have different periods at different time scales and the harmonic chain of all the periods would potentially go to infinity. Destounis et al. [21] 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. [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.

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 characterize the schedulability test. Ganesan [22] 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 [23], [24], [25], [26], and age-of-information (AoI) has also been considered in recent studies [27], [28], [29], [30]; however, these work have not considered ensuring predictable timeliness of individual packet transmissions in multi-cell, multi-channel network settings. Li et al. [30] 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. [31] 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. [32] 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. [33] 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 [34]. 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 studies how to ensure 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). 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 [35], 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. We assume that, for each link \( i \), mechanisms such as transmission power control is used to ensure a certain communication reliability \( p_i \) at each time instant based on in-situ network conditions.
For multi-cell PPRC, interference needs to be controlled such that two mutually-interfering links shall not transmit in the same channel and at the same time [37]. 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 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, $M_1 = \{ 2, 3, 4, 5 \}$. In practice, algorithms such as PRKS [33], [37], [38] 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 from the whole network (including the interference from the links beyond the two-hop neighbors of $i$ in $G_c$), and it serves as an input to the real-time scheduling problem studied in this paper.

B. PPRC traffic model

The PPRC data traffic along link $i$ is characterized 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. 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 $D_i \leq T_i$. Unlike Chen et al. [2], we don’t assume $D_i = T_i$. Thus both implicit

\[1\] The 5G URLLC framework uses time slots of different lengths, 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 5G URLLC by treating the length of the shortest time slot as the smallest time unit. Similarly, the PPRC methodology can be extended to cases where link reliability is Markovian instead of i.i.d. Detailed study of these topics is worthwhile future effort but beyond the scope of this work which represents a first step towards understanding fundamental multi-cell PPRC scheduling issues.
deadlines and constrained deadlines in classic real-time systems are considered. Our model also covers the cases of heterogeneous deadlines across different links.

- **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 et al. [2]:

**Definition 1.** Link $i$ ensures PPRC guarantee if $\forall j$, $\text{Prob}\{ F_{ij} \leq D_i \} \geq S_i$, 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 [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 $X_i = \lceil \log_{1-p_i}(1-S_i) \rceil$ [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. 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 $\rho_i = \frac{X_i}{D_i}$.

**C. 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. 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 [39]. 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) [39][40]. 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 [40] 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. Compared with other optimal multi-processor scheduling algorithms such as P-Fair [41], DP-fair allows more freedom in scheduling jobs within a deadline partition. This is because, at each time slot, only the jobs whose local work density is 1 have to be executed and, for the jobs whose work density is less than 1, there is freedom in choosing which one to execute in terms of satisfying the deadline constraints. Such freedom can enable optimizations such as minimizing delay jitter which is an interesting topic for future studies.

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 between 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 [34]. In addition, even though only close-by links interfere
with one another [33] 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 networks require distributed solutions.

To address the aforementioned differences, 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) [37], 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 below.

**Theorem 1.** The PPRC scheduling problem is NP-hard.

**Proof.** See Appendix A. Therefore, based on the aforementioned observations, 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 [39][40] to each local conflict set. That is, 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), however, 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. 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 \([d'_{i,t}, d''_{i,t}]\), where

\[
d'_{i,t} = \max\{\max_{j \in M_i \cup \{i\}, A_{k,j} \leq t} A_{k,j}, \max_{j \in M_i \cup \{i\}, D_{k,j} \leq t} D_{k,j}\},
\]

and

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

Note that, different from DP-fair which only uses deadlines for demarcation, LDP uses both release times and deadlines for demarcation. This is to ensure that the transmissions along all the links in \( M_i \cup \{i\} \) can make proportionate progress in every deadline partition and in the presence of other links in the multi-cell network. For instance, consider the network conflict graph as shown in Figure 1, link 1, and the interval \([D_{1,j-1}, A_{1,j} - 1]\) between the absolute deadline of the \((j-1)\)-th packet and the \( j \)-th packet along link 1. If \( A_{1,j} \) is not a demarcation, then, due to the interference from links two-hop away from link 1, more urgent transmissions along links 2, 3, 4, and 5 may not be scheduled until after \( A_{1,j} \). Then, at the beginning of time slot \( A_{1,j} + 1 \), the overall sum work density among links \{1, 2, 3, 4, 5\} can be greater than \( N \), which is undesirable and negatively impacts the schedulability of PRRC traffic as we will see in Section IV-C. Note that the above issue does not exist in traditional real-time systems since all the jobs share the same demarcation and the executed jobs are the most urgent in the system.

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 \( X''_{i,t} = X_{i,t} - X'_{i,t} \) 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:

$$X_{i,t} = \begin{cases} 
X_{i,d}^t & D_{i,P_{t}}, t = d_{i,t}^t, \\
X_{i,d}^t - \left( X_{i,t}^t - X_{i,d}^t \right) & D_{i,P_{t}} > d_{i,t}^t, t > d_{i,t}^t, \\
0 & D_{i,P_{t}} \leq d_{i,t}^t,
\end{cases}$$  

(1)

where $D_{i,P_{t}} \leq d_{i,t}^t$ 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,

$$\rho_{i,t} = \frac{X_{i,t}}{L_{i,t} - (t - d_{i,t}^t)} = \frac{X_{i,t}}{d_{i,t}^t - t}.$$  

(2)

Based on these definitions, we develop the LDP real-time scheduling algorithm by extending the multi-channel distributed scheduling algorithm Unified Cellular Scheduling (UCS) [37] to consider PPRC requirements.

In particular, at a time slot $t$ in each local deadline partition, the transmitters and receivers of links set their local work densities which are then used to define the links’ relative priorities, with links having larger local work densities being given higher priority in channel access. Each link $i$ compares priority with its interfering links (i.e., $M_i$) in scheduling, and it executes 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;
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 $i \in E$; 
$T_i, D_i$: period and relative deadline of link $l \in M_i \cup \{i\}$; 
$X_{i,t}$: local traffic demand at link $l \in M_i \cup \{i\}$; 
$state.l.rb.t$: state of links $l \in M_i$ for $\forall rb \in RB$; 
$Prio.l.t$: priority of links $l \in M_i$;

**Output:** Perform the following actions:

1. $state.i.rb.t = UNDECIDED$, $\forall rb \in RB$ 
2. $Prio.i.t = X_{i,t}/(d_{i,t} - t)$; 
3. Share $Prio.l.t$ with the links in $M_i$; 
4. $done = false$; 
5. **while** $done == false$ do 
6. \hspace{1em} $done = true$; 
7. \hspace{2em} **for** each $rb \in RB$ in increasing order of $rb$ ID do 
8. \hspace{3em} if having received updates on $state.l.rb.t$ or $Prio.l.t$ from a link $l \in M_i$ then 
9. \hspace{4em} Update the local copy of $state.l.rb.t$ or $Prio.l.t$ at link $i$; 
10. \hspace{3em} end if 
11. \hspace{2em} if $X_{i,t} == 0$ and $state.i.rb.t == UNDECIDED$ then 
12. \hspace{3em} $state.i.rb.t = INACTIVE$; 
13. \hspace{3em} break; 
14. \hspace{2em} end if 
15. \hspace{2em} if $\exists l \in M_i : state.l.rb.t == ACTIVE$ then 
16. \hspace{3em} $state.i.rb.t = INACTIVE$; 
17. \hspace{3em} break; 
18. \hspace{2em} end if 
19. \hspace{2em} 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 each UNDECIDED $l \in M_i$ then 
20. \hspace{3em} $state.i.rb.t = ACTIVE$; 
21. \hspace{3em} $X_{i,t} = X_{i,t} - 1$; 
22. \hspace{2em} end if 
23. \hspace{2em} if $state.i.rb.t == UNDECIDED$ then 
24. \hspace{3em} $done = false$; 
25. \hspace{2em} end if 
26. \hspace{2em} end for 
27. Share $state.i.rb.t$, $\forall rb \in RB$, with the nodes in $M_i$; 
28. **end while**

- 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 link \( l \in M_i \), if it receives 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 slot \( t \).

The detail of the above local-deadline-partition (LDP) scheduling algorithm for time slot \( t \) is shown in Algorithm 1.

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 1. 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 since the behavior of the LDP algorithm is the same for every channel.

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 0 to 3 is shown in Figure 2.

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 \([0, 3)\). The local traffic demand of link 1 at time slot 0 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 time slot 0 to the absolute deadline of 6, that is, being \( 4 \times \frac{3}{6-0} = 2 \), and the priority (i.e., local work density) of link 1 is the local traffic demand divided by the length from the current time slot to the end of the current local deadline partition, that is, \( \frac{2}{3} = 0.67 \). At the

\(^2\)Approaches similar to that of the UCS algorithm implementation [37] can be used to implement the LDP algorithm in the existing cellular architecture, but detailed study of this is beyond the scope of this work.
beginning of time slot 0, to decide whether link 1 shall transmit at time slot 0, 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 0 for CH1 according to line 16 in algorithm 1 for CH1. Following similar analysis, links 5 and 7 can be shown to have the highest priority among their interfering links and can be ACTIVE concurrently with link 2 at time 0 for CH1. Then, for CH2, link 2 also has the same priority with link 1 and a larger ID than link 1. Therefore, link 2 uses CH2 to transmit a packet and sets its state in channel CH2 as ACTIVE. After receiving the state updates from link 2, links 1 and 3 become INACTIVE at time slot 0 for CH1 and CH2 according to line 16 in algorithm 1. At the beginning of time slot 1, the local traffic demand of link 1 remains 2 since no transmission happened for link 1 in time slot 0. The priority of link 1 is 1 since there are only two time slots left for link 1 to complete its remaining transmissions. Link 1 has the highest priority among its interfering links in time slot 1, thus link 1 uses both CH1 and CH2 in time slot 1 to complete its transmissions. At the beginning of time slot 2, the priority of link 1 becomes 0 since there is no remaining work demand.

Algorithm 1 can be shown to converge for each time slot $t$, and we have

**Theorem 2.** 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 23 and 24 of Algorithm 1. For each INACTIVE link $i$ with non-zero local traffic demand in any channel, there always exists at least one ACTIVE link $l$, $l \in M_i$, based on lines 15, 16 in Algorithm 1. Therefore, changing any INACTIVE link to an ACTIVE link would cause two interfering links active at the same time slot in the same channel, which is not allowed. Hence, the set of all ACTIVE link for any channel is a maximal independent set.

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 [42].
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. 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. 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 the traditional real-time system into the set of links in $M_i \cup \{i\}$. Nonetheless, different from the tasks in real-time system in which every two tasks cannot run concurrently, 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. For the conflict graph shown in Figure 1, for instance, link 1 can transmit concurrently with links 6, 7, and 8 in the same channel. So, an alternative approach is to jointly consider the PPRC traffic demand of the links in each clique $K_{i,j} \subseteq M_i \cup \{i\}$.

For each clique $K_{i,j} \subseteq M_i \cup \{i\}$, $i \in K_{i,j}$, 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 1, 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 3, 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 $\text{MIS}_{G_c}$, the minimum scheduling rate of $S$ is $N \times \min_{\text{mis} \in \text{MIS}_{G_c}} |\text{mis} \cap S|$, where $|\text{mis} \cap S|$ is the number of links in the set $\text{mis} \cap S$.

**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 $\mathcal{K}_i = \{\text{clique } K_{i,j'} : i \in K_{i,j'} \land K_{i,j'} \subseteq M_i \cup \{i\} \land K_{i,j'} \subseteq G_c\}$, and $U_{K_{i,j}} \subseteq \mathcal{K}_i$ 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 $S_{1,K_{1,2}}$, 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 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 $S_{i,K_{i,j}} \subseteq M_i \cup i$, 
\[
\sum_{l \in S_{i,K_{i,j}}} \rho_{l,t-1} \geq N + 1.
\] (3)

Proof. See Appendix B.

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 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. See Appendix C.

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 3 (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 $S_{i,K_{i,j}}$ 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 
\[
\min_{S_{i,K_{i,j}} \in \mathbb{K}_i} \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l} \leq N,
\] (4)
then the PPRC traffic of link $i$ can be supported, that is, link $i$ is schedulable.

Proof. See Appendix E.

From Theorem 3, we need to identify 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 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 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 can use $M_{i,2}$ only to determine the feasible set. For instance, 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 1, $M_{1,2} = \{6, 7, 8\}$, and $M'_1 = \{2, 6, 7, 8\}$. $M_{i} = \{\{2, 6\}, \{2, 7\}, \{2, 8\}\}$. 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 4.** 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 $\text{MIS}_{M'_i}$. 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 $\text{mis} \in \text{MIS}_{M'_i}$, there exists at least one link in $S_i$ that does not interfere with any link in mis.

**Proof.** See Appendix E

Checking a feasible set based on theorem 4 may also be complex since we need to check all the maximal independent set of two-hop interference set. Fortunately, we observer a unique property of feasible sets such that it can help reduce the complexity when checking the feasible set. That is, any combination of two feasible sets is still a feasible set. More, precisely, we identify the property as follows:

**Theorem 5.** Given a link $i$ and a set of links $S_{i,p}$ such that $i \in S_{i,p}$ and $S_{i,p} \subseteq M_i \cup i$. If $S_{i,p}$ is a feasible set, then for any link $l \in M_i \cup i \setminus S_{i,p}$, the set $S_{i,p} \cup \{l\}$ is still a feasible set; if $S_{i,p}$
is not a feasible set, then for any link \( l' \in S_{i,p} \), the set \( S_{i,p} \setminus \{l'\} \) is not a feasible set.

**Proof.** See Appendix F. \[ \square \]

Based on the aforementioned properties 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. Based on Theorem 5, we know that, out of a feasible set, multiple feasible sets with fewer number of links may be derived. Thus, to identify the feasible set with the minimum sum work density, our goal is to find the feasible sets that cannot be further reduced to feasible sets of fewer links. Therefore, the approach used by Algorithm 2 is to start the search by considering relatively large feasible sets and then, for each of these feasible sets, gradually reduce it until no links can be removed.

In Algorithm 2, we start from \( K_{i,j} \). After the initialization of the sum work density with \( K_{i,j} \), we use \( fix \) to denote the set of cliques in \( \mathcal{K}_i \) that have been determined to be a part of the feasible set that we try to identify. If the set of links in \( fix \) is a feasible set, then we move to the next clique \( K_{i,j+1} \) since any additional feasible set with \( K_{i,j} \) will increase the sum work density; on the other hand, if the set of links in \( fix \) is a not feasible set, we need to combine \( fix \) with other cliques in \( \mathcal{K}_i \setminus K_{i,j} \). For each clique \( K_{i,x} \in \mathcal{K}_i \setminus K_{i,j} \), we combine \( fix \) with \( K_{i,x} \), and then we set up a waiting list \( \mathcal{K}_{i,x} = \mathcal{K}_i \setminus fix \) in the order \([K_{i,x+1}, K_{i,x+2}, ..., K_{i,|\mathcal{K}_i|}, K_{i,1}, ..., K_{i,x-1}]\). The waiting list defines a sequence of cliques in \( \mathcal{K}_i \setminus fix \), and the “for” loop in Line 3 of Algorithm 3 will go through all the cliques in \( \mathcal{K}_i \setminus fix \) by following the sequence of the waiting list.

Then we use Algorithm 3 to find the feasible set that may have the minimum sum work density, denoted by \( U_{i,K_{i,j}} \). In Algorithm 3, we define two local parameters \( Choice \) and \( LocalWaitingList \). For each clique \( K_{i,p} \) in the waiting list, if the set of links in \( fix \cup LocalWaitingList \cup K_{i,p} \) is a feasible set, then \( K_{i,p} \) belongs to \( Choice \); otherwise, \( K_{i,p} \) belongs to \( LocalWaitingList \). \( Choice \) and \( LocalWaitingList \) are defined such that, when the “for” loop in Line 5 completes, the links of each clique in \( Choice \) can make the links in \( fix \cup LocalWaitingList \) form a feasible set, and the links in any combination of the cliques in \( fix \cup LocalWaitingList \) cannot be a feasible set according to Theorem 5. Therefore, for the links of each clique in \( Choice \), their union with the links in all the cliques in \( fix \cup LocalWaitingList \) is a feasible set. However, for each clique \( K_{i,p} \in Choice \), the feasible set composed of the links of all the cliques in
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 $i \in E$;
$X_l, D_l$: traffic demand and relative deadline of link $l \in M_i \cup \{i\}$;

Output: whether link $i$ is schedulable;

1: for each clique $K_{i,j} \in \mathbb{K}_i$ do
2: $U_{i,K_{i,j}} = \sum_{l \in M_i \cup \{i\}} X_l D_l$;
3: $fix = K_{i,j}$;
4: if the set of links in $fix$ is a feasible set according to Theorem 3 then
5: $U_{i,K_{i,j}} = \sum_{l \in fix} X_l D_l$;
6: else
7: for each clique $K_{i,x} \in \mathbb{K}_i \setminus K_{i,j}$ do
8: $fix = K_{i,j} \cup K_{i,x}$;
9: $\mathbb{K}_{i,x} = [K_{i,x+1}, K_{i,x+2}, \ldots, K_{i,|\mathbb{K}_i|}, K_{i,1}, \ldots, K_{i,x-1}]$;
10: Reduce($fix, \mathbb{K}_{i,x}$);
11: end for
12: end if
13: end for
14: if $U_{i,K_{i,j}} \leq N, \forall K_{i,j} \in \mathbb{K}_i$ then
15: link $i$ is schedulable;
16: else
17: link $i$ is not schedulable;
18: end if

$fix \cup LocalWaitingList \cup K_{i,p}$ may be further reduced while still being a feasible set. Therefore, if $LocalWaitingList \neq \emptyset$, Algorithm 3 tries to further reduce the current feasible set; on the other hand, if $LocalWaitingList = \emptyset$, the union of the links of each clique in $Choice$ with the links in $fix$ is a feasible set, or the links in $fix$ themselves alone form a feasible set. Then, we select one feasible set with the minimum sum work density as $U_{i,K_{i,j}}$.

Let $|WaitingList|$ denote the number of cliques in $WaitingList$ in Algorithm 3. The “for” loop in Line 3 of Algorithm 3 needs $|WaitingList|$ iterations, and $|LocalWaitingList|$ is at most $|WaitingList| - 1$. Therefore, given $|WaitingList|$, the recursion of Algorithm 3 happens at most $|WaitingList|$ times. The computation complexity to check Theorem 4 in Algorithm 3 is $O\left(\left(|WaitingList|+1\right) \times |WaitingList|\right)$. Let $|\mathbb{K}_i|$ denote the number of cliques in $M_i \cup \{i\}$. The “for” loop in Line 1 of Algorithm 2 needs $|\mathbb{K}_i|$ iterations, the “for” loop in Line 7 of Algorithm 2 needs at most $|\mathbb{K}_i| - 1$ iterations. Therefore, The computation complexity of Algorithm 2 is
Algorithm 3 Reduce

**Input:** fix: the fixed set of cliques of a feasible set candidate;
WaitingList: the waiting list of cliques of a feasible set candidate;

**Output:** \( U_{i,K_i,j} \): sum work density of a feasible set including link \( i \) and links in \( K_{i,j} \);

1: Choice = [];  
2: LocalWaitingList = [];  
3: for each \( K_{i,p} \in \text{WaitingList} \) do  
4: if the set of links in \( \text{fix} \cup \text{LocalWaitingList} \cup K_{i,p} \) is a feasible set according to Theorem 4 then  
5: Choice = Choice \cup K_{i,p};  
6: else  
7: LocalWaitingList = LocalWaitingList \cup K_{i,p};  
8: end if  
9: end for  
10: if \(|\text{LocalWaitingList}| \neq 0\) then  
11: \( \text{choice}_{\text{min}} = \arg \min_{C \in \text{Choice}} \sum_{l \in \text{fix} \cup C} \frac{X_l}{D_l}; \)  
12: Reduce(\( \text{fix} \cup \text{choice}_{\text{min}}, \text{LocalWaitingList} \));  
13: else  
14: if the set of links in \( \text{fix} \) is a feasible set according to Theorem 4 then  
15: \( U_{i,K_i,j} = \min(U_{i,K_i,j}, \sum_{l \in \text{fix}} \frac{X_l}{D_l}); \)  
16: else  
17: \( \text{choice}_{\text{min}} = \arg \min_{C \in \text{Choice}} \sum_{l \in \text{fix} \cup C} \frac{X_l}{D_l}; \)  
18: \( U_{i,K_i,j} = \min(U_{i,K_i,j}, \sum_{l \in \text{fix} \cup \text{choice}_{\text{min}}} \frac{X_l}{D_l}); \)  
19: end if  
20: end if

\( O(|\mathcal{K}_i| \times (|\mathcal{K}_i| - 1)) \). Since \( |\text{WaitingList}| \) equals to \(|\mathcal{K}_i| - 2\), the total computation complexity to check Theorem 4 is \( O(|\mathcal{K}_i| \times (|\mathcal{K}_i| - 1) \times \frac{|\mathcal{K}_i| - 1}{2} \times |\mathcal{K}_i| - 2) \) which is \( O(|\mathcal{K}_i|^4) \). 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 \( M_i \cup M_{i,2} \).

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 approximations of the optimal solutions. As a first step towards understanding the optimality of the LDP algorithm and schedulability test, here we develop a necessary condition for PPRC schedulability and use it to derive a lower bound on the approximation ratio of LDP scheduling.
Theorem 6 (Necessary Condition for PPRC 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

\[
\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l} \leq N. \tag{5}
\]

Proof. See Appendix G.

Based on Theorems 3 and 6, we can explore the gap between the sufficient condition (4) and necessary condition (5). 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 (5) to that of the sufficient condition (4). That is,

\[
\delta(i) = \frac{\max_{K_{i,j} \in \mathbb{K}_i} \sum_{l \in K_{i,j}} \frac{X_l}{T_l}}{\max_{K_{i,j} \in \mathbb{K}_i} \min_{S_i,K_{i,j} \in \mathbb{S}_i,K_{i,j}} \sum_{l \in S_i,K_{i,j}} \frac{X_l}{D_l}}. \tag{6}
\]

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 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

\[
\delta(i)' = \frac{|k'_{\max,i,j}|}{|s_{\max,i}|} \tag{7}
\]

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 and the link reliability \( p_i \) is the same for all the links such that the exclusive regions of all links include the same number of interfering links. That is,

Theorem 7 (Approximation ratio lower-bound). For network \( G \), the approximation ratio of
algorithm LDP scheduling is greater than $\frac{1}{6}$.

Proof. See Appendix H.

As we will show in Section V, the approximation ratios for typical wireless networks tend to be more than 0.5 and up to over 0.99, demonstrating the close-to-optimal performance of the LDP algorithm and associated schedulability test.

V. EXPERIMENTAL STUDY

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

A. Network and PPRC traffic settings

We consider two networks of different sizes, as shown in Figures 3 and 4. The networks are configured to represent typical cellular/infrastructure networks commonly considered in the literature [43], [18], [44], [45]. In particular, 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 example, in LTE-Advanced and 5G, the density of microcell base stations (BSes) can be up to about 10 BSes per square kilometers [43], and the transmission range of the access points in 802.11ac is about 70 meters. Therefore, for Network 1, we uniform-randomly deploy 91 wireless nodes in a $1200 \times 1200$ square-meter region, generating

![Fig. 3: Network 1](image1.png)

![Fig. 4: Network 2](image2.png)
a network of 83 links. There are nine cells which are organized in a $3 \times 3$ grid manner such that each cell covers a $400 \times 400$ square-meter region. There is a base station (BS) within each cell. For Network 2, we uniform-randomly deploy 151 wireless nodes in a $1200 \times 1500$ square-meter region, generating a network of 163 links. There are 12 cells which are organized in a $3 \times 4$ grid manner such that each cell covers a $400 \times 375$ square-meter region.

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 $[1.5, 2]$ which represent typical values in mission-critical sensing and control networks [34]. 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 $j$) [34]. 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. The number of channels considered ranges from 3 to 10.

URLLC requires that the probability of packet loss or deadline violation shall be no more than $10^{-6}$ or even $10^{-9}$. Thus ensuring URLLC needs at least 9 transmission opportunities if the per-packet communication reliability is about 90% (e.g., as in [33]). Therefore, we assume that the relative deadline $D_i$ uniform-randomly ranges from 10 to 40 time slots. The period is assumed to be greater than or equal to the relative deadline, and we experiment with different periods that differ from the relative deadline by a value uniformly distributed in $[0, D_i/6]$. To 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 $[D_i/6, 5D_i/6]$.

B. Experimental results

Approximation ratio of LDP. Here we evaluate the lower bound on the approximation ratio of the LDP scheduling algorithm for Networks 1 and 2. To this end, we consider scenarios

---

3 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.
of demanding PPRC traffic that is close to the network capacity but can still be supported by the LDP algorithm. As an example, Figure 5 shows the histogram of the minimum sum work density of the feasible sets when the number of channels is 4, Figure 6 shows the histogram of the links’ work densities, and Figure 7 shows the histogram of the relative deadlines. Figures 8 and 10 are drawn from Equation 6 and they show the histograms of the approximation ratio lower bound $\delta(i)$ for all the links in Networks 1 and 2 respectively. Figures 9 and 11 are drawn from Equation 7 and they show the histograms of the topology approximation ratio in Networks 1 and 2 respectively. For Network 1, the mean approximation ratio lower bound is 0.6805, and its 95% confidence interval is [0.6509, 0.7101]; the mean topology approximation ratio is 0.6768, and its 95% confidence interval is [0.6451, 0.7085]. 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 0.6080, and its 95% confidence interval is [0.5871, 0.6290]; the mean topology approximation ratio is 0.5283, and its 95% confidence interval is [0.5104, 0.5457]. 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. The mean approximation ratio lower bound is more than 0.52 for both Networks 1 and 2, and it is up to 0.9924 and 0.9679 in Networks 1 and 2 respectively.

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
follows, we experimentally compare the performance of the LDP algorithm with another state-of-the-art algorithm G-schedule [9].

Comparative study.

Out of the existing real-time scheduling algorithms, the G-schedule algorithm [9] addresses a problem that is closest to the PPRC scheduling problem. For real-time multi-channel scheduling in multi-cell cellular networks, G-schedule greedily schedules non-interfering links based on their IDs. G-schedule considers the inter-cell interference, and it has been shown to be optimal for the special line networks where all the nodes are located along a straight line [9]. To comparatively study the performance of LDP and G-schedule, we implement both algorithms in Matlab and study their behavior in Network 2. (Similar phenomena have been observed for Network 1.) 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.
We consider scenarios of demanding PPRC traffic that is close to the network capacity but can still be supported by the LDP algorithm. Then we characterize the feasibility of supporting the real-time traffic using the G-schedule algorithm. In particular, Figure 12 shows the ratio of schedulable links in the network. We see that, while LDP is able to schedule demanding PPRC traffic, the average ratio of schedulable links in G-Schedule is only 0.6985. 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 13 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 based on their IDs 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

We have proposed a distributed local-deadline-partition (LDP) scheduling algorithm to ensure Probabilistic Per-packet Real-time Communications (PPRC) guarantee in multi-cell, multi-channel wireless networks. The LDP algorithm effectively leverages the two-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 traditional real time systems and real-time wireless communications. 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 experimental results have shown that the LDP algorithm can support significantly more PPRC traffic than the state-of-the-art solution G-schedule.

This study represents a first step towards enabling PPRC in multi-cell, multi-channel wireless networks, and it serves as a foundation for exploring other interesting studies. For instance, to generate field-deployable PPRC systems, it will be worthwhile to implement and integrate the LDP scheduling algorithm with PRKS [33] in emerging open-source 5G platforms such
as OpenAirInterface [46]. Another interesting direction is to consider delay jitter control since PPRC applications such as XR 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). IEEE, 2016, p. ”.
[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. Ratnasekara, S. Iraji, A. Layya, 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, p. ”, 2015.
[4] F. Lamberti, F. Manuri, A. Sanna, G. Paravati, P. Pezzolla, and P. Montuschi, “Challenges, Opportunities, and Future Trends of Emerging Techniques for Augmented Reality-Based Maintenance,” IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 4, 2014.
[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, p. ”, 2017.
[6] I. Hou, V. S. Borkar, and P. R. Kumar, “A theory of qos for wireless,” in INFOCOM 2009. IEEE, 2009, p. ”.
[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, p. ”, 2014.
[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, p. ”, 2016.
[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 IEEE RTSS, 2015.
[10] C. Wu, M. Sha, D. Gunati laka, A. Saifullah, C. Lu, and Y. Chen, “Analysis of EDF scheduling for wireless sensor-actuator networks,” in IWQoS, 2014.
[11] V. P. Modekurthy, D. Ismail, M. Rahman, and A. Saifullah, “A Utilization-Based Approach for Schedulability Analysis in Wireless Control Systems,” in IEEE ICHI, 2018.
[12] V. P. Modekurthy, A. Saifullah, and S. Madria, “DistributedHART: A distributed real-time scheduling system for WirelessHART Networks,” in IEEE RTAS, 2019.
[13] 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.
[14] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, “Real-time scheduling for wireless hort networks,” in IEEE RTSS, 2010.
[15] A. Saifullah, D. Gunati laka, P. B. Tiwari, M. Sha, C. Lu, B. Li, C. Wu, and Y. Chen, “Schedulability analysis under graph routing in wireless hort networks,” in IEEE RTSS, 2015.
[16] 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, p. ”, 2012.
[17] D. Gunati laka and C. Lu, “Conservative channel reuse in real-time industrial wireless sensor-actuator networks,” in IEEE ICDCS, 2018.
[18] 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). IEEE Computer Society, 2007, p. .
[19] 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. IEEE Computer Society, 2014, p. .
[20] 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, p. ., 2019.
[21] A. Destounis and G. S. Paschos, “Complexity of URLLC scheduling and efficient approximation schemes,” CoRR, vol. abs/1904.11278, 2019.
[22] A. Ganesan, “Performance Guarantees of Distributed Algorithms for QoS in Wireless Ad Hoc Networks,” IEEE/ACM Transactions on Networking, vol. 28, no. 1, p. ., 2020.
[23] V. Venkataramanman, X. Lin, L. Ying, and S. Shakkottai, “On scheduling for minimizing end-to-end buffer usage over multihop wireless networks,” in IEEE INFOCOM, 2010.
[24] 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.
[25] 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.
[26] 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.
[27] Y. Kuo, “Minimum Age TDMA Scheduling,” in IEEE INFOCOM, 2019.
[28] I. Kadota and E. Modiano, “Minimizing the Age of Information in Wireless Networks with Stochastic Arrivals,” in ACM Mobihoc, 2019.
[29] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y. Hou, and W. Lou, “Kronos: A 5G Scheduler for AoI Minimization under Dynamic Channel Conditions,” in IEEE ICDCS, 2019.
[30] C. Li, S. Li, Y. Chen, Y. T. Hou, and W. Lou, “AoI Scheduling with Maximum Thresholds,” in IEEE INFOCOM, 2020.
[31] 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.
[32] 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.
[33] 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.
[34] 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.
[35] E. Dahlman, S. Parkvall, and J. Skold, 4G, LTE-Advanced Pro and The Road to 5G. Academic Press, 2016.
[36] L. Wang, H. Zhang, and P. Ren, “Distributed scheduling and power control for predictable iot communication reliability,” in IEEE ICC, 2018.
[37] 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.
[38] 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.
A. PROOF OF THEOREM

We prove the theorem by showing that the NP-hard problem of graph k-coloring can be reduced to a case of the PPRC scheduling problem in polynomial time.

The input of k-coloring problem is as following: given a graph \( G = (V, E) \) where \( V \) and \( E \) are the vertex set and edge set respectively. The problem is to decide whether we can color the vertices of the graph with \( k \) colors such that the color of the endpoints of every edge is different.

We reduce the graph k-coloring problem to the PPRC scheduling problem as follows. We assume the the first packet arrives at each link at the same time. For each link \( i \), we also suppose that the link reliability is 1. Therefore, the required number of transmission \( X_i \) equals to 1. The period \( P_i \) and the relative deadline \( D_i \) are both equal to \( k \). The conflict graph is the same with the k-coloring problem graph \( G = (V, E) \) and the number of channels is 1. This reduction can be done in polynomial time, since this reduction takes \( O(1) \) time for each construction.

If \( G \) is k-colorable, then each vertex will be assigned a color and the endpoints of each edge will be colored differently. Then, we can construct a schedule for the links in \( G \) according to the coloring: if vertex \( V_i \) is colored by \( s \in \{1, 2, 3, ..., k\} \), then link \( i \) can transmit its packet at the time slot \( s \). Since the color of vertex \( V_i \) is different with its connected vertexes, link \( i \) will also transmit at different time slot compared with its one-hop neighboring links in the conflict.
graph. In addition, since there are totally $k$ colors for every vertex, each link in the network will complete its packet delivery in one of the $k$ time slots, that is, before the packet deadline. Therefore, there will be no deadline miss in the network.

Conversely, if there exists a valid schedule for the PPRC scheduling problem, we can color vertex $V_i$ in $G$ with the $s$th ($s \in \{1, 2, 3, \ldots, k\}$) color if link $i$ transmits its packet at $s$th time slot. Then, since no two neighboring links are assigned with the same time slot, then no two connected vertices have the same color. This completes the proof.

**B. Proof of Lemma 2**

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.

**C. Proof of Lemma 3**

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 $d'_l$ and $d''_l$ 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 $i$, $d'_l \geq A_{i,p}$ and $d''_l \leq D_{i,p}$. 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

$$\sum_{l \in S_{i,K_{i,j}}} \rho_l = \sum_{l \in S_{i,K_{i,j}}} \frac{X_l}{D_l} \leq N. \quad (8)$$

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

$$W_{l,A_{i,1}} = \sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1}]} \rho_l \times L_{l,t}, \quad (9)$$

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., $D_l < T_l$), we have

$$\sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1}]} L_{l,t} \leq D_{i,1} - A_{i,1}. \quad (10)$$

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

$$\sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1}]} L_{l,t} = D_{i,1} - A_{i,1}. \quad (11)$$

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

$$W_{l,A_{i,1}} = \sum_{\forall \sigma_{l,t} \subseteq [A_{i,1}, D_{i,1}]} \rho_l \times L_{l,t} \leq (D_{i,1} - A_{i,1}) \times \rho_l \quad (12)$$

Then, we can get,

$$\sum_{l \in S_{i,K_{i,j}}} W_{l,A_{i,1}} \leq (D_{i,1} - A_{i,1}) \times \sum_{l \in S_{i,K_{i,j}}} \rho_l \leq (D_{i,1} - A_{i,1}) \times N. \quad (13)$$

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,

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

(14)

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,

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

(15)

Based on the definition of feasible sets, we also have

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

(16)

Thus,

$$
(D_{i,1} - A_{i,1}) N - \sum_{l \in S_{i,K_{i,p}}} C_{l,t_2}
\leq (D_{i,1} - A_{i,1}) N - (D_{i,1} - 1 - A_{i,1}) N
\leq N.
$$

(17)

Therefore, based on (14), (15), (17), we can get

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

(18)

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 13 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 $D_{i,p} - 1, p \geq 3$.

D. Proof of Theorem 3

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 $S_{i,K_{i,j}}$ 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.

E. Proof of Theorem 4

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 $mis'$. Hence, there is no MIS of $G_c$ whose intersection with $S_i$ is empty, if and only if, 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 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 4 holds.

F. Proof of Theorem 5

If $S_{i,p}$ is a feasible set, let $M_i' = (\{i\} \cup M_i \cup M_{i,2}) \setminus S_{i,p}$. Then we consider $S_{i,p}' = S_{i,p} \cup \{l\}$ and $M_i'' = (\{i\} \cup M_i \cup M_{i,2}) \setminus S_{i,p}'$. Given the sets of maximal independent sets $MIS_{M_i'}$, there will be two sets of maximal independent sets. Let $MIS_{M_i',1}$ denote the sets of maximal independent sets such that $l \in mis_1, \forall mis_1 \in MIS_{M_i',1}$. Let $MIS_{M_i',2}$ denote the sets of maximal independent sets such that $l \notin mis_2, \forall mis_2 \in MIS_{M_i',2}$. Together, we have $MIS_{M_i'} = MIS_{M_i',1} \cup MIS_{M_i',2}$. Since
\( M_i'' = M_i' \setminus \{l\} \) and the conflict graph is stable, \( MIS_{M_i''} = MIS_{M_i',2} \). Therefore, \( MIS_{M_i'} \subset MIS_{M_i''} \). Then based on Theorem 4 we can know for each \( mis \in MIS_{M_i'} \), there exists at least one link in \( S_{i,p} \) that does not interfere with any link in \( mis \). Since \( MIS_{M_i''} \subset MIS_{M_i'} \), for each \( mis' \in MIS_{M_i''} \), there exists at least one link in \( S_{i,p}' \) that does not interfere with any link in \( mis' \). Then, \( S_{i,p}' \) is a feasible set.

If \( S_{i,p} \) is not a feasible set, let \( M_i'' = (\{i\} \cup M_i \cup M_i') \setminus S_{i,p} \). Then we consider \( S_{i,p}' = S_{i,p} \setminus \{l'\} \) and \( M_i'' = (\{i\} \cup M_i \cup M_i') \setminus S_{i,p}' \). Since \( S_{i,p} \) is not a feasible set, there exists at least one \( mis' \in MIS_{M_i'} \) such that any link in \( S_{i,p} \) interfere with at least one link in \( mis' \). That implies link \( l' \in S_{i,p} \) interfere with at least one link in \( mis' \). Therefore, the set of links \( mis' \) is a maximal independent set for the set \( M_i'' \). Since each link in \( S_{i,p} \) interfere with at least one link in \( mis' \), each link in \( S_{i,p}' \subset S_{i,p} \) also interfere with at least one link in \( mis' \in MIS_{M_i''} \). Therefore, \( S_{i,p}' \) is not a feasible set. Hence Theorem 5 holds.

G. **Proof of Theorem 6**

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.

H. **Proof of Theorem 7**

The numerator of Equation 6 means the maximum sum work density for all the cliques \( K_{i,j} \in \mathbb{K}_i \). To determine the lower bound of approximation ratio, let link \( i \) and every other link \( l \in M_i \) form a clique, which gives the minimum value of the numerator of Equation 6 (since any additional links in the clique will increase the value of the sum work density). The denominator of Equation 6 means the maximum sum work density of all the feasible sets chosen by Theorem 4 for schedulability test. To determine the lower bound of the approximation ratio, let all the links in \( M_i \cup \{i\} \) form the chosen feasible set and then we can get the maximum value of the denominator (since removing any link from \( M_i \cup \{i\} \) will decrease the value of the denominator). In what is next, we construct a network setting where the aforementioned properties hold.

![Star graph](image-url)
We let every link $l \in M_i$ and link $i$ be on the boundary of other links’ exclusive regions. Then, link $i$ and every other link $l \in M_i$ form a clique of two links and the angle between every two adjacent links is exactly 60 degrees. If there were any additional interfering link $q$, it will interfere with link $i$ and any two adjacent links $l \in M_i$ and form a clique of more than 2 links, which increases the value of the numerator of Equation 6. Therefore, the star conflict graph containing 6 links as shown in Figure 14 can determine the approximation ratio lower bound. We let the sum work density of link $i$ and any link $l \in M_i$ is $\rho_0$, then we can get

$$
\delta(i) = \frac{\sum_{l \in K_{i,j}} \frac{X_l}{T_l}}{\sum_{l \in M_i \cup i} \frac{X_l}{T_l}} = \frac{\rho_0}{6\frac{X_i}{T_i} + \frac{X_i}{T_i}} = \frac{\rho_0}{6\rho_0 - 5\frac{X_i}{T_i}} > \frac{1}{6}.
$$

(19)