Wireless Network Scheduling with Discrete Propagation Delays: Theorems and Algorithms

Shenghao Yang, Jun Ma and Yanxiao Liu

Abstract

The literature provides evidence that considering signal propagation delays can significantly enhance the scheduling rate region of wireless networks. This paper focuses on the link scheduling problem in networks where signal delays between nodes are multiples of a time interval. To model such networks, a directed hypergraph is employed, along with an integer matrix that specifies the delays. The link scheduling problem is closely connected to the independent sets of the periodic hypergraph induced by the network model. However, due to the infinite number of vertices, it is impractical to enumerate the independent sets of the periodic hypergraph using generic graph algorithms. To tackle this challenge, a graphical approach is proposed in this paper. The link scheduling rate region is characterized using a finite directed graph called a scheduling graph, which is derived from the network model. A collision-free schedule of the network corresponds to a path in the scheduling graph, and the rate region is determined by the convex hull of the rate vectors associated with the cycles in the scheduling graph. Although existing cycle enumeration algorithms can be employed to calculate the rate region, their computational complexity becomes prohibitively high as the size of the scheduling graph grows exponentially with the number of network links. To address this issue, the dominance property of a special scheduling graph called the step-T scheduling graph is investigated. This property allows the utilization of specific subgraphs of the step-T scheduling graph to characterize the scheduling rate region, achieving a reduction in both the number of cycles and their lengths. For common problems such as calculating the rate region and maximizing a weighted sum of the scheduling rates, algorithms leveraging the dominance property are developed. These algorithms can be more efficient than using generic graph algorithms directly on the scheduling graphs.

I. Introduction

Wireless communication media, such as radio, light, and sound, all have nonzero signal propagation delays between two communicating devices separated by a nonzero distance. In the existing theory of wireless communications, these signal propagation delays are typically regarded as a factor that can potentially generate interference [2]. However, studies on underwater acoustic communications and interference channels have observed that

This paper was presented in part at INFOCOM 2021 [1].

S. Yang is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Shenzhen, China. Email: shyang@cuhk.edu.cn

J. Ma is with QIANHAI FOF, Shenzhen, China, and with the School of Management, Xiamen University, Xiamen, China. Email: jma@qhfof.com

Y. Liu is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China. Email: yanxiaoliu@link.cuhk.edu.hk

This work was funded in part by the Shenzhen Science and Technology Innovation Committee (Grant JCYJ20180508162604311).

wireless communication networks can actually benefit from these signal propagation delays [3]–[9]. To gain a better understanding of the utilization of propagation delays, we present a theoretical framework for studying network scheduling when considering these signal propagation delays.

A. Background and Related Results

In most terrestrial radio-based wireless communication systems, such as the 5G cellular network, guard intervals are employed in network scheduling to mitigate the interference caused by signal propagation delays. The inclusion of guard intervals does not significantly degrade system performance since the duration for transmitting signals is much greater than the signal propagation delay between devices. For communication within a few kilometers, the propagation delay of radio waves typically falls within the range of tens of microseconds. In contrast, the signal frame length is typically a few milliseconds. This scheduling approach, characterized by a long signal frame length, is referred to as *framed scheduling* and constitutes a focal point of research in wireless network scheduling for terrestrial radio-based communications [10]–[13].

In underwater acoustic communications, the propagation delay of sound can be significantly longer, measured in seconds. For instance, the sound speed in underwater environments is approximately $1,500$ meters per second, resulting in a delay of around 2 seconds for sound to propagate over a distance of 3 kilometers. If framed scheduling is adopted in this scenario, the frame length should be on the order of tens or hundreds of seconds [14]. Researchers in underwater acoustic networks have been motivated to address delays in the network scheduling problem [3], [4], [6]–[9]. They have observed substantial performance advantages, such as improved energy consumption and throughput, by allowing smaller frame lengths that are comparable to the propagation delay between communication devices.

Researchers have demonstrated that in a network consisting of K pairs of closely located communication devices, framed scheduling allows only one device to transmit a signal at a given timeslot without collision. By carefully considering the delays, it is feasible to devise scheduling schemes where all K pairs can transmit simultaneously without generating collisions [6]. Moreover, recent work [15] has provided examples with relaxed delay constraints to illustrate the unbounded advantages of scheduling with propagation delays, particularly when the network size is large.

The advantage of utilizing propagation delays has also been discovered in the study of the time-domain interference alignment approach for multi-user interference channels [5]. Though the time-domain interference alignment and network scheduling are not equivalent problems,[1] it is possible to transform a network schedule into an achievable scheme for the interference channel [5]. Using this approach, it is shown in [5] that there exists a non-vanishing rate for each user when the number of users tends to infinity. Additionally, some papers discussing underwater acoustic networks also refer to scheduling with propagation delays as interference alignment [16], [17].

[1]For instance, in time-domain interference alignment, the timeslot size corresponds to the sampling time interval, and each timeslot contains a single sample value [5]. In contrast, in the network scheduling problem, each timeslot is typically considered as a radio frame containing a sequence of sample values.

TABLE I

COMPARISON OF THE SIGNAL PROPAGATION DELAY AND THE OFDM SYMBOL LENGTH IN BOTH UNDERWATER ACOUSTIC COMMUNICATION AND TERRITORY RADIO COMMUNICATION. THE OFDM SUBCARRIER SPACING FOR UNDERWATER ACOUSTIC COMMUNICATION REFERS TO [18]. THE 15 KHZ OFDM SUBCARRIER SPACING IS USED IN BOTH 4G LTE AND 5G NR, AND THE 15×2^6 KHZ OFDM SUBCARRIER SPACING IS USED IN 5G NR. THE OFDM SYMBOL LENGTH DOES NOT INCLUDE THE CYCLIC PREFIX.

	propagation speed (km/s)	transmission range (km)	propagation delay (s)	OFDM subcarrier spacing (kHz)	OFDM symbol length (s)
underwater acoustic	1.5	3	2	0.005	0.2
4G LTE	3×10^5	3	1×10^{-5}	15	6.67×10^{-5}
5G NR	3×10^5	3	1×10^{-5}	15×2^6	1.04×10^{-6}

The benefit of using propagation delays can be achieved in terrestrial radio communications when a sufficiently large bandwidth is utilized. We can consider orthogonal frequency-division multiplexing (OFDM) as an example, which is employed in many modern wireless communication systems. In OFDM, a frame is typically composed of multiple OFDM symbols. A comparison of OFDM numerology in different systems is presented in Table I. In underwater acoustic communications, the OFDM symbol length can often be much shorter than the typical signal propagation delay. However, for 4G LTE with a 15 kHz OFDM subcarrier spacing, the OFDM symbol length is more than 6 times the typical signal propagation delay. In 5G NR, larger OFDM subcarrier spacings up to 15×2^6 kHz are proposed for bandwidths up to 400 MHz in millimeter-wave frequencies. In this case, the OFDM symbol length can be approximately $1/10$ of the typical signal propagation delay. Furthermore, wireless communication in the frequency range of 100 GHz to 10 THz has been discussed for the next generation of cellular networks [19]. In such scenarios, the bandwidth can reach tens of gigahertz, and the OFDM symbol length can be several nanoseconds.

Although the study of wireless networks with propagation delays shows promise, it is still in its preliminary stages. Early works [3], [4] have utilized mixed integer linear programming models to capture collision constraints and derive heuristic algorithms. In cases where delays are integers, the scheduling problem with propagation delays is formulated as a weighted directed graph. In this graph, the vertices represent communication links, directed edges model collision relations between two links, and the weight of an edge denotes the corresponding delay [3], [5]. For complete weighted directed graphs (where any two links can generate collisions with each other), existing works [5], [6] have discovered that the network scheduling problem exhibits a periodic property. Dynamic programming approaches have been employed to maximize the (weighted) total scheduling rate. However, these approaches suffer from high computation costs due to the exponential state space and do not provide an explicit characterization of the scheduling rate region.

Without considering propagation delays, the network scheduling problem can be formulated using a graph or a hypergraph [10]–[13], [20]–[29]. In this formulation, a vertex represents a network link and an edge represent the collision relation among the links. When considering a *protocol or binary collision model* [10], [12], a graph can be used to capture the pair-wise collision relation among the network links. In a more practical *physical Signal-to-Interference-and-Noise Ratio (SINR) model* [30], a group of links can collectively generate a collision with another link, which is described using a hypergraph. Although the network links can be directed, the graphical representation of collision is usually undirected.

The graphical model of collisions plays a crucial role in network scheduling research. It allows for the explicit characterization of the scheduling rate region using the independent sets of the graph or hypergraph, enabling the analysis of the wireless network's performance based on graphical properties. However, when it comes to scheduling with delays, a graphical theory specifically addressing this aspect is currently lacking but highly desired. In this paper, our objective is to fill this gap by proposing a graphical approach that characterizes the scheduling rate region with delays.

B. Main Contributions

For a wireless network, if all the signal propagation delays are multiples of a fixed timeslot length, we say the delays are *discrete*. In this paper, we study wireless networks scheduling with discrete propagation delays, which serves as an initial step towards understanding networks with general real number delays. For a network with general delays, the rate region can be approximated by the rate region of a network with discrete delays [15]. A general approach is also presented in [15] to approximate a network with general delays using one that has discrete delays, with a controlled performance difference. Additionally, the problem of time-domain interference alignment can also be approximated as a problem with discrete signal propagation delays [5].

We propose a comprehensive network model that incorporates a matrix to represent delays and a *directed hypergraph* to describe collision relations. In the case of a binary collision model, the directed hypergraph simplifies to a directed graph. When all delays are zero, the model reduces to one without considering delays, allowing for the omission of edge direction without impacting the network scheduling problem. However, in the presence of general delays, edge direction becomes crucial, and the network scheduling problem is connected to independent sets within the *periodic (undirected) hypergraph* induced by our network model.

Despite the connection between the network scheduling problem and independent sets within the periodic hypergraph, finding a complete solution remains elusive due to the infinite nature of the periodic hypergraph. Notably, independent sets can have unbounded sizes, and there exists an infinite number of them. Consequently, existing approaches can only provide approximations of the scheduling rate region, and the computational cost is high, as demonstrated in prior research [5], [6]. In this paper, we address the challenges associated with the infinite number of independent sets in the periodic hypergraph, and our main contributions are twofold: i) exact and explicit characterizations of the scheduling rate region, and ii) efficient algorithms for calculating the rate region and maximizing a weighted sum of the link rates.

We show that the scheduling rate region of a network can be achieved using collision-free, periodic schedules. To provide an explicit characterization of this scheduling rate region, we adopt a graphical approach. For our network model, we establish a series of directed graphs called *scheduling graphs*. Each scheduling graph has two parameters: a blocklength T and a step size Q (where $1 \leq Q \leq T$). We show in general that a collision-free schedule is equivalent to a path within a scheduling graph with $T \geq 2D^*$, where D^* is a parameter derived from the delay matrix. Consequently, the scheduling rate region is the convex hull of the rate vectors associated with the cycles of the scheduling graph, and hence is a polytope.[2] In the case of binary collision, the characterization of the scheduling rate region can be achieved using a scheduling graph with $T \geq D^*$. It is worth noting that the scheduling graphs, regardless of whether they are induced by graph-based or hypergraph-based network models, share common properties, except for the variation in the bounds imposed on T. These common properties allow for a unified study of scheduling related problems based on the scheduling graphs.

We further study scheduling-related algorithms, specifically focusing on computing the rate region and maximizing a weighted sum of the link rates. Based on the characterization of the scheduling rate region, we explore various approaches to address these computational problems. As a straightforward approach, one can employ a backtracking algorithm, such as Johnson's algorithm [31], to enumerate cycles within a scheduling graph and consequently obtain the rate region. However, the computational cost of this approach becomes prohibitively high as the network size increases. This can be attributed to two main factors: the exponential growth of vertices and edges in a scheduling graph relative to the number of network links, and the exponential growth in the number of cycles as the size of the scheduling graph increases.

To simplify the characterization of the rate region, we investigate an additional property of the scheduling problem. Specifically, we introduce a dominance property for the step-T scheduling graph, where the step size $Q = T$. This property allows us to leverage subgraphs of the step-T scheduling graph to characterize the scheduling rate region, achieving a reduction in both the number of cycles and their respective lengths. To illustrate the benefits of the dominance property in the step-T scheduling graph, we provide an example involving a sequence of step-T scheduling graphs. In these scheduling graphs, the numbers of edges and vertices are exponential in the number of links, but the corresponding reduced scheduling graphs possess a constant number of cycles with constant lengths.

Based on the dominance property, we develop two algorithms for calculating the scheduling rate region. The first algorithm enables the calculation of the entire rate region by enumerating only the cycles present in a reduced scheduling graph. This approach proves to be more efficient than enumerating all cycles in the original scheduling graph. The second algorithm takes an incremental approach, specifically targeting a subset of the scheduling rate region characterized by cycles up to a certain length. Numerical evaluations demonstrate that this algorithm outperforms the direct enumeration of cycles up to a specific length, particularly in scenarios involving large network sizes.

To solve a maximization problem on the scheduling rate region, the straightforward approach involves two steps: computing the scheduling rate region or a subset thereof, and then maximizing the objective function within the feasible rate vectors obtained. However, this approach can become impractical for larger networks due to the substantial computational cost associated with calculating the rate region. To address this challenge, we propose an algorithm that leverages the insights from the dominance property. This algorithm maximizes a linear function without the need to compute the entire rate region, resulting in significantly lower computation costs compared to the straightforward approach.

[2]In this paper, a cycle in a directed graph has no repeated vertices (which is also called a simple circuit) and hence the total number of cycles of a finite graph is finite.

Last, our characterization of the independent sets in periodic hypergraphs holds potential for various applications in operational research problems [32], [33] as well as transportation systems [34]. The insights gained from our research can be leveraged to optimize decision-making processes in these domains.

C. Paper Organization

The remainder sections of the paper are organized as follows. In Sec. II, we present the network model and introduce the fundamental properties of the periodic graph induced by the network model. Additionally, we extend the isomorphism and connectivity properties of periodic graphs to periodic hypergraphs. Sec. III focuses on basic theoretical results. We provide a proof that the scheduling rate region can be attained through collision-free, periodic schedules, and establish the convexity of the scheduling rate region. Moreover, we explore how the isomorphism and connectivity properties of periodic hypergraphs can simplify the rate region problem. Moving on to Sec. III-C, we characterize the achievable rates using scheduling with guard intervals.

Our main results are presented from Sec. IV to Sec. VI. In Sec. IV, we introduce the concept of scheduling graphs and demonstrate that a collision-free schedule is equivalent to a directed path in a scheduling graph. We explore the use of cycles and paths within a scheduling graph to characterize the scheduling rate region effectively. Additionally, in Sec. IV-C, we enhance certain results specifically for the binary collision model. Moving on to Sec. V, we investigate the dominance property of step-T scheduling graphs. By analyzing this property, we derive refined characterizations of the rate region and develop algorithms to compute the scheduling rate region. In Sec. VI, we study how to maximize a linear function over the scheduling rate region.

Lastly, Sec. VII serves as the concluding remarks, where we discuss possible extensions of our results and future research directions. To facilitate understanding and reference, we have compiled a list of notations used throughout the paper in Table II.

II. Hypergraph Network Model and Periodic Hypergraph

We propose a general network model that consists of a matrix that specifies the delays and a *directed hypergraph* that describes the collision relations. Since the matrix contains only integer values, this model is also known as a *discrete network model*. We formulate the scheduling problem and demonstrate its relationship to the periodic hypergraph induced by the network model.

Let $\mathbb{Z}$ denote the set of integers and $\mathbb{Z}^+$ represent the set of nonnegative integers. Similarly, let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{R}^+$ correspond to the set of non-negative real numbers.

A. Discrete Network Model

We start with a node-based network model, but as we progress, we will discover that utilizing only network links is adequate for solving the network scheduling problem.

Suppose time is slotted and each timeslot is indexed by an integer $t \in \mathbb{Z}$. Consider a network of N nodes indexed by $1, 2, \ldots, N$. Each node has the capability to both transmit and receive a specific communication signal within a timeslot. The signal transmitted by node i at timeslot t propagates to node j at time $t + D(i, j)$, where $D(i, j) \in \mathbb{Z}^+$

TABLE II

SOME NOTATIONS USED IN THE PAPER, LISTED IN THE ALPHABETICAL ORDER.

Notation	Explanation	Section
$\mathrm{conv}\mathcal{A}$	the convex hull of a set $\mathcal{A}$	IV-B
$\mathrm{cl}(P)$	the cycle generated from a path $P = (A_0, \ldots, A_k)$ in $(\mathcal{M}_T, \mathcal{E}_T)$	V-B
$D_{\mathcal{L}}$	the link-wise delay matrix	II-B
$D^*_{\mathcal{N}}$, D^*	the character of network $\mathcal{N}$	III-A
$\mathrm{dom}\mathcal{A}$	the collection of all $B \in (\mathcal{R}^+)^{m\times n}$ that is dominated by some elements in $\mathcal{A}$	V-B
$\mathcal{E}^*$	$\mathcal{E}^* = M^*_{2T}$	V-A
$\mathcal{I}(l)$	the collision set of a link l	II-A
$\mathcal{I}$	the collision profile, $\mathcal{I} = (\mathcal{I}(l), l \in \mathcal{L})$	II-A
$\mathcal{L}$	the set of communication links	II-A
$\max_{\succcurlyeq} \mathcal{A}$	the set of maximal elements of the partially ordered set $(\mathcal{A}, \succcurlyeq)$	V
$(\mathcal{M}_T, \mathcal{E}_{T,Q})$	the scheduling graph with vertex set $\mathcal{M}_T$ and edge set $\mathcal{E}_{T,Q}$	IV-A
$(\mathcal{M}_T, \mathcal{E}_T)$	the step-T scheduling graph, $\mathcal{E}_T = \mathcal{E}_{T,T}$	IV-A
$\mathcal{M}^*_L$	$\{B : (B, B') \in \mathcal{E}^*$ for certain $B'\}$	V-A
$\mathcal{M}^*_R$	$\{B' : (B, B') \in \mathcal{E}^*$ for certain $B\}$	V-A
$\mathcal{N}$	the (link-wise) network model, $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$	II-B
$\mathcal{N}^\infty$	the periodic hypergraph induced by $\mathcal{N}$	II-B
$\mathcal{N}^{\text{line}}_{L,K}$	the uniform line network of L hops with the K-hop collision model	II-B
$\mathcal{N}^T$	the subgraph of $\mathcal{N}^\infty$ of T columns	III-C
Q	the step size of a scheduling graph	IV-A
R_P	the rate vector of a closed path in a scheduling graph	IV-B
$R^{\mathcal{N}}_S$, R_S	the rate vector of schedule S for network $\mathcal{N}$	III-A
$\mathcal{R}^{\mathcal{N}}$, $\mathcal{R}$	the (scheduling) rate region of network $\mathcal{N}$	III-A
$\mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$	the convex hull of the rate vectors of all the cycles in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$	IV-B
$\mathcal{R}_k$	the subset of $\mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$ generated by the cycles of $(\mathcal{M}_T, \mathcal{E}_T)$ up to length k	V-B
$\widetilde{\mathcal{R}}^{\mathcal{N}^T}$	the convex hull of the rate vectors of all the independent sets of $\mathcal{N}^T$	III-C
$\mathbb{R}$, $\mathbb{R}^+$	$\mathbb{R}$ is the set of real numbers, $\mathbb{R}^+$ is the set of non-negative real numbers	II
$S(l, t)$	the entry of a schedule S indexed by the link l and time t	II
$S[T, Q, k]$	the submatrix of a schedule S of T columns, starting from the kQ column	IV-A
T	the blocklength of a scheduling graph	IV-A
$(\mathcal{V}, \mathcal{F})$	the reduced scheduling graph	V-D
$\mathbb{Z}$, $\mathbb{Z}^+$	$\mathbb{Z}$ is the set of integers, $\mathbb{Z}^+$ is set of nonnegative integers	II
$\preccurlyeq$, $\succcurlyeq$	the partial order relation defined on $\mathbb{R}$, and can be applied on matrices of the same size component-wisely	III
$\wedge$	the minimum function of two real numbers, and can be applied on two matrices of the same size component-wisely	V
$\mathbf{1}$	a column vector with all entries 1	V-C

represents the signal propagation delay from node i to node j. The transmission of node i in the timeslot t does not affect the reception of node j in any other timeslots. The matrix $D = (D(i,j), 1 \leq i,j \leq N)$ is called the *delay matrix* of the network.

In our network model, a *(communication) link* is represented by an ordered pair (s,r), where $1 \leq s \neq r \leq N$, indicating the transmitting and receiving nodes, respectively. Links are directional, meaning that (i,j) and (j,i) are considered distinct links. Let $\mathcal{L}$ denote a finite set of all the links. For a given link l, we use s_l and r_l to denote the transmitting node and the receiving node of l, respectively. Each link can be in one of two states: active or inactive. A link l is considered *active* in a timeslot t if the transmitting node s_l sends a signal in timeslot t intended to be received by node r_l in timeslot $t + D(\mathrm{s}_l, \mathrm{r}_l)$. Conversely, a link l is deemed *inactive* in a timeslot t if no signal is transmitted by node s_l during that timeslot.

Example 1 (Uniform line networks)**.** A line network consisting of L hops comprises $L+1$ nodes and the link set defined as:

$$\mathcal{L} = \{l_i \triangleq (i, i+1) : i = 1, \ldots, L\}.$$

In this network, the delay matrix D is defined such that $D(i,j) = |i-j|$ for $1 \leq i,j \leq L+1$. Throughout this paper, we will utilize this network as an example for various definitions and results.

To incorporate the constraints of link activation, we assign to each link l a subset $\mathcal{I}(l)$ of $2^{\mathcal{L}}$, called the *collision set* of l. Each subset of links in the collision set $\mathcal{I}(l)$ has the potential to impact the reception of node r_l. In general, when link l is active in timeslot t, we declare that a *collision occurs* if there exists a subset $\theta \in \mathcal{I}(l)$ such that each link $l' \in \theta$ is also active in timeslot $t + D(\mathrm{s}_l, \mathrm{r}_l) - D(\mathrm{s}_{l'}, \mathrm{r}_l)$, i.e., the signal transmitted by $\mathrm{s}_{l'}$ propagates to r_l in the timeslot $t + D(\mathrm{s}_l, \mathrm{r}_l)$. In other words, when all the links in θ are active in specific timeslots such that their signals simultaneously propagate to node r_l at the same timeslot $t + D(\mathrm{s}_l, \mathrm{r}_l)$, the transmission of link l in timeslot t fails due to a collision.

The collision set defined above is flexible and inclusive of various collision scenarios. To specifically model the scenario where two links l and l' with the same transmitting node (i.e., $\mathrm{s}_l = \mathrm{s}_{l'}$) cannot be active simultaneously, we can define the collision sets with $l' \in \mathcal{I}(l)$ and $l \in \mathcal{I}(l')$. To model the constraint of half-duplex communication, where a node cannot transmit and receive signals simultaneously, the collision set $\mathcal{I}(l)$ should include all non-empty subsets of $\{l' : \mathrm{s}_{l'} = \mathrm{r}_l\}$.

Example 2. For the line network in Example 1 with $L = 4$, the collision sets of the links can be defined as follows:

$$\mathcal{I}(l_1) = \{\{l_2\}\}, \quad \mathcal{I}(l_2) = \{\{l_3\}, \{l_1, l_4\}\},$$
$$\mathcal{I}(l_3) = \{\{l_4\}\}, \quad \mathcal{I}(l_4) = \emptyset.$$

From these collision sets, we can observe that $\{l_{i+1}\} \in \mathcal{I}(l_i)$ for $i = 1, 2, 3$. This implies that nodes $2, 3, 4$ cannot transmit and receive signals simultaneously. Additionally, the set $\{l_1, l_4\}$ is in the collision set of link l_2. This means that if link l_1 is active in timeslot $t-1$ and link l_4 is active in timeslot t, a collision will occur if link l_2 is active in timeslot t.

In the network model described above, we define $\mathcal{I} = (\mathcal{I}(l), l \in \mathcal{L})$ as the *collision profile*. The collision relation among links can be represented as a *directed hypergraph*, denoted as $(\mathcal{L}, \mathcal{I})$, with the vertex set $\mathcal{L}$ and the directed edge set $\{(l, \theta) : l \in \mathcal{L}, \theta \in \mathcal{I}(l)\}$. It is worth noting that in a general directed hypergraph with the vertex set $\mathcal{L}$, an edge belongs to $2^{\mathcal{L}} \times 2^{\mathcal{L}}$ [35]. However, our directed hypergraph $(\mathcal{L}, \mathcal{I})$ is a special case where the tail of an edge must be a singleton. Therefore, we represent an edge in the hypergraph of our network model as $(l, \theta) \in \mathcal{L} \times 2^{\mathcal{L}}$. The relation of the hypergraph model and the physical model has been discussed in [5]. For the sake of completeness, we provide details in the Appendix on how to transform the results obtained to a wireless network in the physical model.

When all the collision sets $\mathcal{I}(l)$ consist only of singletons (i.e., for any $\theta \in \mathcal{I}(l)$, $|\theta| = 1$), the collision model and the network model are said to be *binary*. For a binary collision model, we can represent $\mathcal{I}(l)$ as a subset of $\mathcal{L}$ to simplify the notation, and $(\mathcal{L}, \mathcal{I})$ becomes a directed graph. For scheduling with propagation delays, the network model studied in [5], [6] is a binary model with $\mathcal{I}(l) = \mathcal{L} \setminus \{l\}$. While our primary focus is on the general hypergraph model, we will demonstrate that some of our results can be further improved for the binary collision model.

Example 3 (Line network with the K-hop collision model)**.** For the line network defined in Example 1, we consider a *binary collision model* called the K-hop model, where the reception of a node can only have collisions from nodes within K hops distance [13]. For each link l_i with $i = 1, \ldots, L$, the collision set $\mathcal{I}(l_i)$ of the K-hop model is defined as:

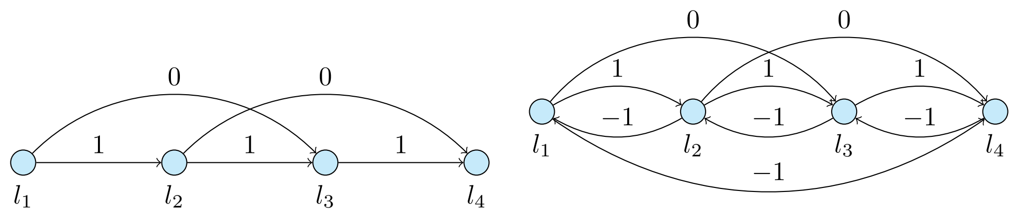
$$\mathcal{I}(l_i) = \{l_j : j \neq i, |j - i - 1| \leq K\}. \tag{1}$$

When $K \geq 1$, it can be observed that for $i = 1, \ldots, L-1$, $l_{i+1} \in \mathcal{I}(l_i)$. This implies that node $i+1$ is half-duplex, meaning it cannot transmit and receive signals simultaneously. In the case of $L = 4$ and $K = 1$, the collision sets are as follows:

$$\begin{aligned} &\mathcal{I}(l_1) = \{l_2, l_3\}, \quad \mathcal{I}(l_2) = \{l_3, l_4\}, \\ &\mathcal{I}(l_3) = \{l_4\}, \qquad \mathcal{I}(l_4) = \emptyset. \end{aligned} \tag{2}$$

Note that the collision relation among links is not necessarily symmetric. In the example where $L = 4$ and $K = 1$, link l_3 may generate collisions for l_1, but l_1 does not generate collisions for l_3. This can be understood by examining the corresponding network nodes: link l_1 represents the communication from node 1 to node 2, while l_3 represents the communication from node 3 to node 4. In the K-hop model with $K = 1$, the transmission of node 3 can affect the reception of node 2, but the transmission of node 1 cannot affect the reception of node 4.

When all delays are set to 0, the network model we defined corresponds to a model without considering delays. In the literature, undirected hypergraphs (or graphs) are commonly used to model collisions in such scenarios [10]–[12], [20], [22]–[29]. The reason for the collision model being undirectional is that the link scheduling does not depend on the direction in this special case. However, when considering general delays, it becomes necessary to use a *directed* hypergraph (or graph) to accurately model collisions. The necessity of using a directed hypergraph will be further elaborated after the link scheduling problem is formulated.



(a) $\mathcal{N}_{4,1}^{\text{line}}$ (b) $\mathcal{N}_{4,2}^{\text{line}}$

Fig. 1. The graphical representation of $\mathcal{N}_{4,1}^{\text{line}}$ and $\mathcal{N}_{4,2}^{\text{line}}$. In these graphs, as well as the following graphical representation of our discrete network models, the vertices in a graph represent links in the network. The number on an edge (l,l') is the value of $D_{\mathcal{L}}(l,l')$.

B. Link-wise Network Model and Link Schedule

To simplify the network model, we define the $|\mathcal{L}|\times|\mathcal{L}|$ *link-wise delay matrix* $D_{\mathcal{L}}$ with

$$D_{\mathcal{L}}(l,l') = D(\mathrm{s}_l,\mathrm{r}_l) - D(\mathrm{s}_{l'},\mathrm{r}_l).$$

The definition of the link-wise delay matrix does not depend on the collision profile. Collision can be determined using $D_{\mathcal{L}}$. Specifically, if a link l is active in a timeslot t, it has a collision if for a certain $\theta\in\mathcal{I}(l)$, every link $l'\in\theta$ is also active in the timeslot $t+D_{\mathcal{L}}(l,l')$. By utilizing the link-wise delay matrix $D_{\mathcal{L}}$, it is not necessary to directly refer to the network nodes when verifying collisions.

We define the (link-based) network model as $\mathcal{N}\triangleq(\mathcal{L},\mathcal{I},D_{\mathcal{L}})$. It is sufficient for us to use this link-based model in the following discussion. In the network model, $(\mathcal{L},\mathcal{I})$ represents a directed hypergraph of finite size. The entries of $D_{\mathcal{L}}$ are integers and can be negative. If $l'\in\cup_{\theta\in\mathcal{I}(l)}\theta$, then the value $D_{\mathcal{L}}(l,l')$ is required in collision checking and we consider the (l,l') entry of $D_{\mathcal{L}}$ as *relevant*. On the other hand, if the (l,l') entry of $D_{\mathcal{L}}$ is *not relevant*, meaning $l'\notin\cup_{\theta\in\mathcal{I}(l)}\theta$, then $D_{\mathcal{L}}(l,l')$ is not involved in collision checking. When the (l,l') entry of $D_{\mathcal{L}}$ is not relevant, we mark this entry as $*$ in a link-wise delay matrix. When the context is clear, we also call $D_{\mathcal{L}}$ the delay matrix.

Example 4. Following Example 3, the link-wise delay matrix $D_{\mathcal{L}}$ of the L-length, K-hop collision line network is given by:

$$D_{\mathcal{L}}(l_i,l_j) = D(i,i+1) - D(j,i+1) = 1-|j-i-1|. \tag{3}$$

The network is denoted as $\mathcal{N}_{L,K}^{\text{line}}=(\mathcal{L},\mathcal{I},D_{\mathcal{L}})$, where $\mathcal{L}=\{l_1,\dots,l_L\}$, $\mathcal{I}$ is defined in (1), and $D_{\mathcal{L}}$ is defined in (3). The graphical representation of $\mathcal{N}_{4,1}^{\text{line}}$ and $\mathcal{N}_{4,2}^{\text{line}}$ are shown in Fig. 1.

One fundamental question related to a discrete network $\mathcal{N}=(\mathcal{L},\mathcal{I},D_{\mathcal{L}})$ is the efficiency of link activation scheduling. A *(link) schedule* S is a matrix of binary digits indexed by pairs $(l,t)\in\mathcal{L}\times\mathbb{Z}$, where $S(l,t)=1$ indicates that l is active in timeslot t, and $S(l,t)=0$ indicates that link l is inactive in timeslot t.

Definition 1 (Collision-free schedule)**.** For a given schedule S and a pair $(l,t)\in\mathcal{L}\times\mathbb{Z}$, we say that $S(l,t)$ has a *collision* in the network $\mathcal{N}=(\mathcal{L},\mathcal{I},D_{\mathcal{L}})$ if for a certain $\theta\in\mathcal{I}(l)$, we have $S(l',t+D_{\mathcal{L}}(l,l'))=1$ for every $l'\in\theta$.

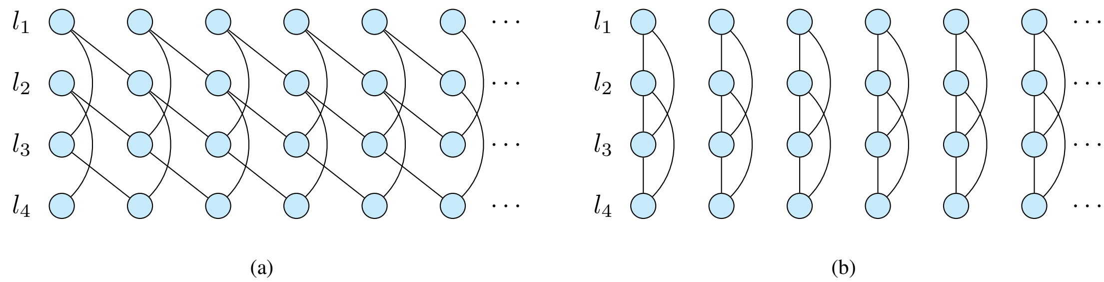


Fig. 2. Illustration of the periodic graphs. (a) is the periodic graph induced by $\mathcal{N}_{4,1}^{\text{line}}$. (b) is the periodic graph induced by a network that shares the same link set and collision profile as $\mathcal{N}_{4,1}^{\text{line}}$, but has $\mathbf{0}$ as the delay matrix.

On the other hand, if for all $\theta \in \mathcal{I}(l)$, $S(l', t + D_{\mathcal{L}}(l, l')) = 0$ for a certain $l' \in \theta$, we say $S(l,t)$ is *collision-free*. A schedule S is said to be *collision-free* if $S(l,t)$ is collision-free for all $(l,t) \in \mathcal{L} \times \mathbb{Z}$ with $S(l,t) = 1$.

Definition 2 (Periodic hypergraph)**.** Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$. The *periodic (undirected) hypergraph* induced by $\mathcal{N}$, denoted by $\mathcal{N}^\infty$, has the vertex set $\mathcal{L} \times \mathbb{Z}$. In $\mathcal{N}^\infty$, a subset $\{(l_i, t_i) : i = 1, \ldots, k\} \subset \mathcal{L} \times \mathbb{Z}$ is an edge if and only if there exists $j \in \{1, \ldots, k\}$ such that $\{l_i : i \in \{1, \ldots, k\}, i \neq j\} \in \mathcal{I}(l_j)$ and $t_i = t_j + D_{\mathcal{L}}(l_j, l_i)$ for all $i \neq j \in \{1, \ldots, k\}$. In other words, an edge is always of the form $\{(l,t), (l', t + D_{\mathcal{L}}(l, l')) : l' \in \theta\}$ for some $\theta \in \mathcal{I}(l)$.

When all the collision sets are binary, $\mathcal{N}^\infty$ becomes a graph with edges of the form $\{(l,t), (l', t + D(l, l'))\}$ for all $l' \in \mathcal{I}(l)$. See Fig. 2-(a) for an illustration of the periodic graph induced by $\mathcal{N}_{4,1}^{\text{line}}$. For a general hypergraph with the vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subset 2^{\mathcal{V}}$, a subset $\mathcal{A}$ of $\mathcal{V}$ is said to be *independent* if for any $\mathcal{U} \in \mathcal{E}$, $\mathcal{U} \nsubseteq \mathcal{A}$. The following theorem establishes the relation between a collision-free schedule of $\mathcal{N}$ and an independent set of $\mathcal{N}^\infty$.

Theorem 1. *A schedule S is collision-free for a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ if and only if the set $\{(l,t) \in \mathcal{L} \times \mathbb{Z} : S(l,t) = 1\}$ is an independent set in $\mathcal{N}^\infty$.*

Proof: For a schedule S, let $\mathcal{A} = \{(l,t) \in \mathcal{L} \times \mathbb{Z} : S(l,t) = 1\}$. For any edge e of $\mathcal{N}^\infty$, $e = \{(l,t), (l', t + D_{\mathcal{L}}(l, l')) : l' \in \theta\}$ for some $\theta \in \mathcal{I}(l)$. When S is collision-free, either $S(l,t) = 0$, or $S(l,t) = 1$ and there exists $l' \in \theta$ such that $S(l', t + D_{\mathcal{L}}(l, l')) = 0$. Therefore, e is not a subset of $\mathcal{A}$ and hence $\mathcal{A}$ is an independent set of $\mathcal{N}^\infty$.

Suppose S is not collision-free. Then there exists $(l,t) \in \mathcal{L} \times \mathbb{Z}$ with $S(l,t) = 1$, and a certain $\theta \in \mathcal{I}(l)$ such that $S(l', t + D_{\mathcal{L}}(l, l')) = 1$ for all $l' \in \theta$. We observe in this case that $e = \{(l,t), (l', t + D_{\mathcal{L}}(l, l')) : l' \in \theta\}$ is an edge of $\mathcal{N}^\infty$ and $e \subset \mathcal{A}$. Therefore, $\mathcal{A}$ is not an independent set in $\mathcal{N}^\infty$. ■

Theorem 1 gives an equivalence relation between a collision-free schedule S of $\mathcal{N}$ and an independent set I of $\mathcal{N}^\infty$. Specifically, the support of S forms an independent set in $\mathcal{N}^\infty$, and the indicator function of I, represented as a binary matrix, serves as a collision-free schedule for $\mathcal{N}$. Based on this equivalence relation, we can use one representation to refer to the other interchangeably.

Denote by $\mathbf{0}$ the matrix with all the entries 0. The network $(\mathcal{L}, \mathcal{I}, \mathbf{0})$ has a special periodic graph where for each $t \in \mathbb{Z}$, the set $\{(l,t), l \in \mathcal{L}\}$ forms a component that is isomorphic to $(\mathcal{L}, \mathcal{I})$, with the edge directions ignored. Fig. 2-(b) illustrates the periodic graph of the network generated by replacing the delay matrix in $\mathcal{N}_{4,1}^{\text{line}}$ as $\mathbf{0}$. The independent sets of the periodic graph of $(\mathcal{L}, \mathcal{I}, \mathbf{0})$ can be completely characterized by the independent sets of $(\mathcal{L}, \mathcal{I})$, where the edge directions are ignored. This is the reason why the scheduling problem without considering delays does not require the edge directions in $(\mathcal{L}, \mathcal{I})$.

The equivalence between a collision-free schedule of $\mathcal{N}$ and an independent set of $\mathcal{N}^\infty$ cannot provide an explicit and exactly solution to the scheduling problem in general when $D_{\mathcal{L}} \neq \mathbf{0}$. This is because the periodic graph has *infinitely* many vertices, which means that not only can an independent set have an unbounded size, but also the number of independent sets is infinite. In Sec. III-C, we will discuss how existing approaches to independent sets can only approximate the optimal scheduling solutions, and the corresponding computation cost is high. While some properties of periodic graphs, such as isomorphism and connectivity, have been studied in the literature [36], [37], the independent set problem has not been well understood. In Sec. III, we will formally define the scheduling rate region problem. In Sec. IV, we will further investigate the properties of the periodic hypergraph to enable an exact and explicit solution of the scheduling problem.

C. Useful Properties of Periodic Hypergraphs

Here, we will briefly introduce the isomorphism and connectivity properties of periodic graphs and discuss their extension to periodic hypergraphs. In Sec. III-B, we will leverage these properties to simplify the scheduling rate region problem.

1) Isomorphism: A *vertex assignment* for a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ is an integer-valued vector $\mathbf{b} = (b_l, l \in \mathcal{L})$. Each vertex assignment $\mathbf{b}$ induces a new link-wise delay matrix $D_{\mathcal{L}}^{\mathbf{b}} = (D_{\mathcal{L}}^{\mathbf{b}}(l,l'))$ where

$$D_{\mathcal{L}}^{\mathbf{b}}(l,l') = D_{\mathcal{L}}(l,l') + b_l - b_{l'}, \tag{4}$$

and hence a new network $\mathcal{N}_{\mathbf{b}} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}}^{\mathbf{b}})$. According to [36], if $(\mathcal{L}, \mathcal{I})$ is a graph, $\mathcal{N}^\infty$ and $\mathcal{N}_{\mathbf{b}}^\infty$ are isomorphic with respect to the bijection $f : \mathcal{L} \times \mathbb{Z} \to \mathcal{L} \times \mathbb{Z}$ with $f(l,t) = (l, t + b_l)$. In other words, $\mathcal{N}_{\mathbf{b}}^\infty$ is obtained by shifting all the vertices in the row l of $\mathcal{N}^\infty$ by b_l. The mapping is still an isomorphism when $(\mathcal{L}, \mathcal{I})$ is a hypergraph as the argument in [36] involves only the delay matrix.

2) Connectivity: In an undirected graph, two vertices are said to be connected if there exists a path between these two vertices. Exploring the connectivity of $\mathcal{N}^\infty$ can potentially simplify the scheduling problem by considering each component of $\mathcal{N}^\infty$ individually. We first discuss the connectivity when $(\mathcal{L}, \mathcal{I})$ is a graph, which has been studied in [36]. Let $g_{\mathcal{N}}$ be the greatest common divisor of $D_{\mathcal{L}}(l,l')$ for all $l \in \mathcal{L}$ and $l' \in \mathcal{I}(l)$. Under the condition that $D_{\mathcal{L}} \neq \mathbf{0}$, $g_{\mathcal{N}}$ is well-defined. Then $D_{\mathcal{L}}/g_{\mathcal{N}}$ is an integer matrix. According to [36], $\mathcal{N}^\infty$ has $g_{\mathcal{N}}$ components isomorphic to the periodic graph of $(\mathcal{L}, \mathcal{I}, D_{\mathcal{L}}/g_{\mathcal{N}})$. Fig. 3 illustrates a periodic graph with three components. In contrast to the case $D_{\mathcal{L}} = \mathbf{0}$, the components of $\mathcal{N}^\infty$ in general have an infinite size.

If $(\mathcal{L}, \mathcal{I})$ is a hypergraph, for $l \in \mathcal{L}$, define $\mathcal{I}'(l) = \bigcup_{\theta \in \mathcal{I}(l)} \theta$. Let $\mathcal{I}' = (\mathcal{I}'(l), l \in \mathcal{L})$. Then $\mathcal{N}' = (\mathcal{L}, \mathcal{I}', D_{\mathcal{L}})$ is a new network with a binary collision model. By [37], two vertices in $\mathcal{N}^\infty$ are connected if and only if the two

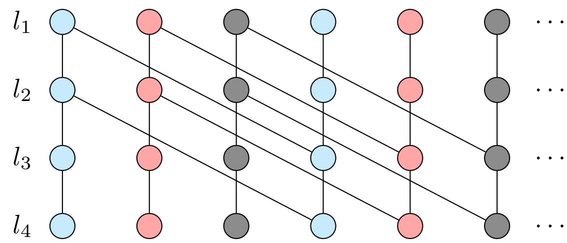


Fig. 3. An periodic graph with three components. Each component is illustrated by a different color (gray scale).

corresponding vertices in $(\mathcal{N}')^\infty$ are connected. Let $g_{\mathcal{N}'}$ be the greatest common divisor of $D_{\mathcal{L}}(l, l')$ for all $l \in \mathcal{L}$ and $l' \in \mathcal{I}'(l)$. We know that $\mathcal{N}^\infty$ has $g_{\mathcal{N}} \triangleq g_{\mathcal{N}'}$ components isomorphic to the periodic graph of $(\mathcal{L}, \mathcal{I}, D_{\mathcal{L}}/g_{\mathcal{N}})$.

III. Scheduling Rate Region

When delays are all 0, an independent set of $(\mathcal{L}, \mathcal{I})$ with the edge directions ignored represents an achievable scheduling rate vector, and the set of all maximal independent sets determines the scheduling rate region [11]–[13]. However, for networks with general delays, the concepts of achievable scheduling rate vectors and the scheduling rate region need to be extended to account for the characteristics of general periodic hypergraphs. In this section, we will formally define the schedule rate vector and the scheduling rate region for a general network with delays. We will discuss some fundamental properties of the scheduling rate region, and study a class of special schedules known as guarded schedules.

For two real matrices A and B of the same size, we write $A \preccurlyeq B$ if all the entries of A are not larger than the corresponding entries of B at the same position. We similarly define $A \succcurlyeq B$ to indicate that all entries of A are not smaller than the corresponding entries of B at the same position. For a matrix A and a scalar a, we write $A + a$ to denote the matrix obtained by adding a to each entry of A. We similarly define $A - a$ to be the matrix obtained by subtracting a from each entry of A.

A. *Scheduling Rate Vector*

For a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$, we denote for each schedule S and link l

$$R_S^{\mathcal{N}}(l) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \iota(S(l,t) = 1, S(l,t) \text{ is collision-free}), \tag{5}$$

where $\iota(A_1, A_2, \ldots)$ is the indicator function with a value 1 if the sequence of conditions A_i are all true, and 0 otherwise. To maintain consistency with network scheduling conventions, we only consider $S(l,t)$ with $t \geq 0$ when defining $R_S^{\mathcal{N}}(l)$. If the limit on the right-hand side of (5) exists, we say that $R_S^{\mathcal{N}}(l)$ exists. When $R_S^{\mathcal{N}}(l)$ exists, we call $R_S^{\mathcal{N}}(l)$ the *(scheduling) rate of link* l. If $R_S^{\mathcal{N}}(l)$ exists for all $l \in \mathcal{L}$, we call $R_S^{\mathcal{N}} = (R_S^{\mathcal{N}}(l), l \in \mathcal{L})$ the *rate vector* of S for $\mathcal{N}$. We may omit the superscript in $R_S^{\mathcal{N}}$ and $R_S^{\mathcal{N}}(l)$ when the network $\mathcal{N}$ is implied.

Definition 3 (Scheduling rate region)**.** For a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$, a rate vector $R = (R(l), l \in \mathcal{L})$ is said to be *achievable* if for any $\epsilon > 0$, there exists a schedule S such that $R_S \succcurlyeq R - \epsilon$. The set $\mathcal{R}^{\mathcal{N}}$ of all the achievable rate vectors is called the *scheduling rate region* of $\mathcal{N}$.

Define the *character* of the network $\mathcal{N}$ as

$$D^*_{\mathcal{N}} = \max_{l\in\mathcal{L}} \max_{\theta\in\mathcal{I}(l)} \max_{l'\in\theta} |D_{\mathcal{L}}(l,l')|. \tag{6}$$

In other words, $D^*_{\mathcal{N}}$ is the maximum relevant delay in $D_{\mathcal{L}}$. When $\mathcal{N}$ is known from the context, we also write $D^*_{\mathcal{N}}$ as D^*.

Example 5. For $\mathcal{N}^{\text{line}}_{L,K}$ defined in Example 4, by (1) and (3),

$$\begin{aligned} D^* &= \max_{1\le i\ne j\le L, |j-i-1|\le K} |1-|j-i-1|| \\ &= \max\{\min\{L,K\}-1, 1\}. \end{aligned}$$

So, when $K=1$, $D^*=1$, and when $L \ge K \ge 2$, $D^* = K-1$.

Definition 4 (Periodic schedule)**.** A schedule S is considered *periodic* if there exists a positive integer T_p such that $S(l,t) = S(l,t+T_p)$ for all $(l,t) \in \mathcal{L}\times\mathbb{Z}$. The positive integer T_p in this context is called a *period* of the schedule.

Similar to Definition 3, a rate vector R is considered *achievable by collision-free, periodic schedules* if for any $\epsilon > 0$, there exists a collision-free, periodic schedule S such that $R_S \not\succcurlyeq R-\epsilon$. Although the scheduling rate region $\mathcal{R}^{\mathcal{N}}$ is defined for general schedules, the following lemma states that collision-free, periodic schedules achieve the rate region $\mathcal{R}^{\mathcal{N}}$. Our result directly implies the special cases observed in the existing papers [5], [6] when the network has a binary collision with $\mathcal{I}(l) = \mathcal{L}\setminus\{l\}$. Note that we do not limit the period of the periodic schedules in Lemma 1. In the next section, we will further enhance the result by showing that we only need a finite set of periodic schedules to completely characterize $\mathcal{R}^{\mathcal{N}}$.

Lemma 1. *For a network $\mathcal{N}$, the rate region $\mathcal{R}^{\mathcal{N}}$ can be achieved using only collision-free, periodic schedules.*

Remark 1. Our proof is based on a constructive approach. First, for any given schedule, we can always find a collision-free schedule with the same rate vector by setting all the entries corresponding to collisions to 0. Second, for any collision-free schedule with a rate vector R, we can construct a periodic schedule using a segment of the schedule from time 0 to $T-1$, such that the rate vector of the periodic schedule converges to R as T tends to infinity.

Proof: Fix $R\in\mathcal{R}$ and $\epsilon > 0$. By Definition 3, there exists a schedule S such that

$$R_S(l) \ge R(l) - \epsilon/2, \text{ for all } l\in\mathcal{L}. \tag{7}$$

Define a schedule S' such that

$$S'(l,t) = \begin{cases} 1 & S(l,t)=1 \text{ and is collision-free} \\ 0 & \text{otherwise.} \end{cases}$$

We see that S' is collision-free and $R_{S'} = R_S$.

By the definition in (5), there exists a sufficiently large T_0 such that for all $T \geq T_0$ and all $l \in \mathcal{L}$,

$$\left| R_{S'}(l) - \frac{1}{T}\sum_{t=0}^{T-1} S'(l,t) \right| \leq \frac{\epsilon}{4}. \tag{8}$$

Fix any $T^* \geq \max\{T_0 + D^*, 4D^*/\epsilon\}$. Define a schedule S^* with period T^*:

$$S^*(l,t) = \begin{cases} S'(l,t) & t = 0,1,\ldots,T^*-1-D^*, \\ 0 & t = T^*-D^*,\ldots,T^*-1. \end{cases}$$

Now we argue that S^* is collision-free.

Fix (l,t) with $S^*(l,t) = 1$. According to the definition of S^*, there exists $t_0 \in \{0,1,\ldots,T^*-1-D^*\}$ such that $t = kT^* + t_0$. We show that $S^*(l,t)$ is collision-free by contradiction. Assume there exists $\theta \in \mathcal{I}(l)$ such that $S^*(l', t + D_{\mathcal{L}}(l,l')) = 1$ for every $l' \in \theta$, i.e., $S^*(l,t)$ has a collision. For $l' \in \theta$, let $t' = t + D_{\mathcal{L}}(l,l')$. As $l' \in \theta \in \mathcal{I}(l)$, we have $|D_{\mathcal{L}}(l,l')| \leq D^*$, and hence $kT^* - D^* \leq t' \leq kT^* + T^* - 1$. We discuss the possible range of t' in three cases:

1) When $kT^* - D^* \leq t' < kT^*$, by the definition of S^*, $S^*(l',t') = 0$.
2) When $(k+1)T^* - D^* \leq t' \leq (k+1)T^* - 1$, by the definition of S^*, $S^*(l',t') = 0$.
3) When $kT^* \leq t' \leq (k+1)T^* - 1 - D^*$, write $t' = kT^* + t_0'$. Due to the periodical property of S^*, $S^*(l',t') = S^*(l',t_0')$. As $t_0' \in \{0,1,\ldots,T^*-1-D^*\}$, by the definition of S^*, $S^*(l',t_0') = S'(l',t_0')$. Similarly, we have $S'(l,t_0) = S^*(l,t_0) = S^*(l,t) = 1$. As $S'(l,t_0)$ is collision-free, for certain $l' \in \theta$, $S'(l',t_0') = 0$, i.e., $S^*(l',t') = 0$.

Therefore, for all the three cases of t', we get a contradiction to the assumption that $S^*(l,t)$ has a collision.

As S^* is periodic and collision-free, we further have

$$\begin{aligned} R_{S^*}(l) &= \frac{1}{T^*}\sum_{t=0}^{T^*-1} S^*(l,t) \\ &= \left(1 - \frac{D^*}{T^*}\right)\frac{1}{T^*-D^*}\sum_{t=0}^{T^*-1-D^*} S'(l,t) \\ &\geq \left(1 - D^*/T^*\right)\left(R_{S'}(l) - \epsilon/4\right) \\ &\geq R_{S'}(l) - \epsilon/4 - D^*/T^* \\ &\geq R_{S'}(l) - \epsilon/2 \\ &\geq R(l) - \epsilon, \end{aligned}$$

where the first inequality follows from $T^* \geq T_0 + D^*$ and (8), the third inequality follows from $T^* \geq 4D^*/\epsilon$, and the last inequality is obtained by substituting (7). The proof of the theorem is complete. ■

The convexity is another fundamental property of the scheduling rate region $\mathcal{R}^{\mathcal{N}}$. In the next section, we will further show that $\mathcal{R}^{\mathcal{N}}$ is a polytope with a finite vertices.

Lemma 2. *The rate region $\mathcal{R}^{\mathcal{N}}$ of a network $\mathcal{N}$ is convex.*

Remark 2. Our proof is based on a constructive approach. For any two collision-free, periodic schedules, we can construct a new periodic schedule that has a rate vector close to a convex combination of the rate vectors of the two original schedules.

Proof: Fix R_1 and R_2 in $\mathcal{R}^{\mathcal{N}}$. Let $R = \alpha R_1 + (1-\alpha)R_2$ where $0 < \alpha < 1$. The lemma is proved by showing $R \in \mathcal{R}^{\mathcal{N}}$. Fix $\epsilon > 0$. By Lemma 1, there exists a collision-free schedule S_1 of period T_1 such that $R_{S_1} \succcurlyeq R_1 - \epsilon/2$, and a collision-free schedule S_2 of period T_2 such that $R_{S_2} \succcurlyeq R_2 - \epsilon/2$.

For a positive integer k_1, let $k_2 = \lceil \frac{1-\alpha}{\alpha} \frac{T_1}{T_2} k_1 \rceil$. Construct a schedule S of period $k_1T_1 + k_2T_2 + 2D^*$ such that $S(l,t) = S_1(l,t)$ for $t \in \{0, 1, \dots, k_1T_1 - 1\}$, $S(l,t) = S_2(l, t - k_1T_1 - D^*)$ for $t \in k_1T_1 + D^* + \{0, 1, \dots, k_2T_2 - 1\}$, and $S(l,t) = 0$ for other values of t in the first period. Similar to the proof of Lemma 1, we can argue that the schedule S is collision-free. The rate vector R_S satisfies

$$\begin{aligned} R_S &= \frac{k_1T_1R_{S_1} + k_2T_2R_{S_2}}{k_1T_1 + k_2T_2 + 2D^*} \\ &\succcurlyeq \frac{k_1T_1R_{S_1} + \frac{1-\alpha}{\alpha}T_1k_1R_{S_2}}{k_1T_1 + \frac{1-\alpha}{\alpha}T_1k_1 + T_2 + 2D^*} \\ &= \frac{\alpha R_{S_1} + (1-\alpha)R_{S_2}}{1 + \alpha(T_2 + 2D^*)/(T_1k_1)} \\ &\succcurlyeq \frac{R - \epsilon/2}{1 + \alpha(T_2 + 2D^*)/(T_1k_1)} \\ &= R - \frac{R\alpha(T_2 + 2D^*)/(T_1k_1) + \epsilon/2}{1 + \alpha(T_2 + 2D^*)/(T_1k_1)}. \end{aligned}$$

Therefore, when k_1 is sufficiently large, $R_S \succcurlyeq R - \epsilon$, and hence $R \in \mathcal{R}^{\mathcal{N}}$. ■

B. Simplification by Isomorphism and Connectivity

In Section II-C, we have discussed the concepts of isomorphism and connectivity of periodic hypergraphs. Now, we will demonstrate how these properties can be utilized to simplify the problem of scheduling rate region. Our discussion is self-contained, as we solely rely on the properties of schedules and rate regions introduced earlier.

Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ and a vector $\mathbf{b} = (b_l, l \in \mathcal{L})$. For a collision-free schedule S of $\mathcal{N}$, we define a schedule $S^{\mathbf{b}}$ as

$$S^{\mathbf{b}}(l,t) = S(l, t + b_l),$$

which has the same rate vector as S. Then by Definition 1, we can verify that $S^{\mathbf{b}}$ is collision-free for $\mathcal{N}_{\mathbf{b}} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}}^{\mathbf{b}})$, where $D_{\mathcal{L}}^{\mathbf{b}}$ is defined in (4). Due to symmetry, we can similarly argue that a collision-free schedule of $\mathcal{N}_{\mathbf{b}}$ induces a collision-free schedule of $\mathcal{N}$ of the same rate vector. The above discussion is summarized as follows:

Proposition 1. *For a network $\mathcal{N}$ and a vertex assignment* $\mathbf{b}$, $\mathcal{R}^{\mathcal{N}} = \mathcal{R}^{\mathcal{N}_{\mathbf{b}}}$.

Though $\mathcal{N}$ and $\mathcal{N}_{\mathbf{b}}$ are equivalent in terms of rate region, they may have different characters (see Example 6). Note that the character of a network may affect the complexity for the rate region calculation according to the characterization in Sec. III. Therefore, it is possible to use isomorphism to simplify the calculation of the rate region. In an extreme case, if $D_{\mathcal{L}}^{\mathbf{b}}$ becomes $\mathbf{0}$, the problem is resolved.

Example 6. Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_\mathcal{L})$ with the link set $\mathcal{L} = \{l_1, l_2, l_3, l_4\}$, the collision sets

$$\begin{aligned} \mathcal{I}(l_1) &= \{l_2, l_3, l_4\}, \quad \mathcal{I}(l_2) = \{l_1, l_3, l_4\}, \\ \mathcal{I}(l_3) &= \{l_2, l_4\}, \qquad \mathcal{I}(l_4) = \{l_3\}, \end{aligned}$$

and the link-wise propagation delay matrix

$$D_\mathcal{L} = \begin{bmatrix} * & 0 & -2 & -4 \\ 0 & * & 0 & -2 \\ * & 0 & * & 0 \\ * & * & 0 & * \end{bmatrix}.$$

The character $D^*_\mathcal{N} = 4$. For the vertex assignment $\mathbf{b} = (4, 3, 2, 1)$, the link-wise delay matrix becomes

$$D^\mathbf{b}_\mathcal{L} = \begin{bmatrix} * & 1 & 0 & -1 \\ -1 & * & 1 & 0 \\ * & -1 & * & 1 \\ * & * & -1 & * \end{bmatrix}.$$

The character of $\mathcal{N}_\mathbf{b} = (\mathcal{L}, \mathcal{I}, D^\mathbf{b}_\mathcal{L})$ is 1.

Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_\mathcal{L})$ with $D_\mathcal{L} \neq \mathbf{0}$. Let g be the greatest common divisor of $D_\mathcal{L}(l, l')$ for all $l \in \mathcal{L}$ and $l' \in \cup_{\theta \in \mathcal{I}(l)} \theta$. As we have discussed in Sec. II-C, $\mathcal{N}^\infty$ has g isomorphic components. We prove that the rate region of $\mathcal{N}$ is the same as the rate region of $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$. Our proof also gives the connection of the schedules for $\mathcal{N}$ and $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$.

Proposition 2. *Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_\mathcal{L})$ with $D_\mathcal{L} \neq \mathbf{0}$. Let g be the greatest common divisor of $D_\mathcal{L}(l, l')$ for all $l \in \mathcal{L}$ and $l' \in \cup_{\theta \in \mathcal{I}(l)} \theta$. Then, $\mathcal{R}^\mathcal{N} = \mathcal{R}^{(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)}$.*

Proof: As $D_\mathcal{L} \neq \mathbf{0}$, $g > 0$. For a collision-free schedule S of $\mathcal{N}$, we define schedules $S_1, \ldots, S_g$ as:

$$S_i(l, t) = S(l, tg + i).$$

Let's verify that S_i is collision-free for $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$. Suppose $S_i(l, t) = S(l, tg + i) = 1$. As S is collision-free for $\mathcal{N}$, we have for any $\theta \in \mathcal{I}(l)$, there exists $l' \in \theta$ such that $S(l, tg + i + D_\mathcal{L}(l, l')) = S_i(l, t + D_\mathcal{L}(l, l')/g) = 0$. Therefore, S_i is collision-free for $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$. Hence $R_S = \frac{1}{g}(R_{S_1} + \cdots + R_{S_g}) \in \mathcal{R}^{(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)}$, and $\mathcal{R}^\mathcal{N} \subset \mathcal{R}^{(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)}$.

To prove $\mathcal{R}^\mathcal{N} \supset \mathcal{R}^{(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)}$, consider a collision-free schedule S' of $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$. Define a schedule S'' for $\mathcal{N}$ such that,

$$S''(l, ig + j) = S'(l, i), i = 0, 1, \ldots, j = 0, 1, \ldots, g - 1.$$

To verify that S'' is collision-free for $\mathcal{N}$, consider (l, t) such that $S''(l, t) = 1$. Write $t = ig + j$, where i and j are integers such that $i \geq 0$ and $0 \leq j < g$. So $S'(l, i) = S''(l, t) = 1$. Since S' is collision-free for $(\mathcal{L}, \mathcal{I}, D_\mathcal{L}/g)$, for any $\theta \in \mathcal{I}(l)$, there exists $l' \in \theta$ such that $S''(l', ig + D_\mathcal{L}(l, l') + j) = S'(l', i + D_\mathcal{L}(l, l')/g) = 0$. Hence $S''(l, t)$

is collision-free for $\mathcal{N}$ as for any $\theta \in \mathcal{I}(l)$, there exists $l' \in \theta$ such that $S''(l', t + D_{\mathcal{L}}(l, l')) = 0$. The proof is completed as S'' and S' have the same rate vector. ∎

In the following example, we illustrate how to combine isomorphism and connectivity to simplify a network.

Example 7. Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ with the link set $\mathcal{L} = \{l_1, l_2, l_3, l_4\}$, the collision sets

$$\mathcal{I}(l_1) = \{l_2, l_3\}, \quad \mathcal{I}(l_2) = \{l_3, l_4\},$$
$$\mathcal{I}(l_3) = \{l_4\}, \qquad \mathcal{I}(l_4) = \emptyset,$$

and the link-wise propagation delay

$$D_{\mathcal{L}} = \begin{bmatrix} * & 1 & 5 & * \\ * & * & 1 & 5 \\ * & * & * & 1 \\ * & * & * & * \end{bmatrix}.$$

This network has $D^*_{\mathcal{N}} = 5$. Given a vertex assignment $\mathbf{b} = [0, 1, 2, 3]$, we get a new network $\mathcal{N}_{\mathbf{b}} = (\mathcal{L}, \mathcal{I}, D^{\mathbf{b}}_{\mathcal{L}})$, where

$$D^{\mathbf{b}}_{\mathcal{L}} = \begin{bmatrix} * & 0 & 3 & * \\ * & * & 0 & 3 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix}.$$

The periodic graph induced by $\mathcal{N}_{\mathbf{b}}$ is shown in Fig. 3. As the greatest common divisor of the relevant entries of $D^{\mathbf{b}}_{\mathcal{L}}$ is 3, we have $\mathcal{R}^{\mathcal{N}} = \mathcal{R}^{\mathcal{N}_{\mathbf{b}}} = \mathcal{R}^{\mathcal{N}'}$, where $\mathcal{N}' = (\mathcal{L}, \mathcal{I}, D^{\mathbf{b}}_{\mathcal{L}}/3)$. The character $D^*_{\mathcal{N}'}$ is 1.

C. *Scheduling with Guard Intervals*

Lastly in this section, we discuss the classical approach known as guarded scheduling, which involves using guard intervals to prevent collisions. While it is not necessary to be familiar with guarded scheduling in order to proceed with our approach of characterizing the scheduling rate region, this discussion can provide additional insights into the connection and distinction between scheduling with and without delays.

Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ with character D^*. We fix an integer $T_F \geq D^* + 1$, which is called the *frame length*. All timeslots t, $t \geq 0$, are grouped into *frames*, each consisting of T_F consecutive timeslots. For instance, frame k ($k = 0, 1, \ldots$) includes timeslots $kT_F + i$, $i = 0, 1, \ldots, T_F - 1$. In the context of this frame structure, a schedule S is considered a *guarded schedule* if the last D^* timeslots in each frame remain inactive. More precisely, for any frame k, the timeslots $kT_F + i$, $i = T_F - D^*, \ldots, T_F - 1$ are inactive. The last D^* timeslots in each frame are referred to as the *guard interval*. We illustrate a guarded schedule as follows, where $T_F = 5$ and $D^* = 2$:

$$\left[\begin{array}{c|ccccc|ccccc|ccccc|c} \cdots & x & x & x & 0 & 0 & x & x & x & 0 & 0 & x & x & x & 0 & 0 & \cdots \\ \cdots & x & x & x & 0 & 0 & x & x & x & 0 & 0 & x & x & x & 0 & 0 & \cdots \\ \cdots & x & x & x & 0 & 0 & x & x & x & 0 & 0 & x & x & x & 0 & 0 & \cdots \\ \cdots & x & x & x & 0 & 0 & x & x & x & 0 & 0 & x & x & x & 0 & 0 & \cdots \end{array}\right]$$

One notable property of a guarded schedule is that it eliminates inter-frame collisions by utilizing the guard interval. This characteristic allows us to analyze the schedule in each frame independently. To achieve high frame efficiency, the frame length T_F is typically chosen to be significantly larger than the character D^*.

1) Rate Region Approximation: In the context of framed scheduling, the schedule of each frame, excluding the guard interval, can be regarded as an independent set. Consider guarded scheduling with a frame length of $T + D^*$. Define $\mathcal{N}^T$ as the subgraph of $\mathcal{N}^\infty$ induced by the vertex set $\mathcal{L} \times \{0, 1, \dots, T-1\}$. An independent set of $\mathcal{N}^T$ can be represented by a binary $|\mathcal{L}| \times T$ matrix. For instance, the empty set is an independent set in $\mathcal{N}^T$ represented by the all zero $|\mathcal{L}| \times T$ matrix. For the guarded scheduling, each frame is collision-free if and only if the schedule of the first T time slots of the frame represents an independent set of $\mathcal{N}^T$.

The rate vector of an independent set of $\mathcal{N}^T$ is the vector obtained by summing the columns of the corresponding matrix presentation and normalizing the result by T. We define $\widetilde{\mathcal{R}}^{\mathcal{N}^T}$ as the convex hull of rate vectors associated with all independent sets of $\mathcal{N}^T$. Consequently, the achievable rate region using guarded scheduling with a frame length of $T + D^*$ is given by $\frac{T}{T+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^T}$, which is a subset of $\mathcal{R}^{\mathcal{N}}$.

Proposition 3. *For a discrete network $\mathcal{N}$, $\mathcal{R}^{\mathcal{N}}$ is equal to the closure of* $\cup_{T=1,2,\dots}\frac{T}{T+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^T}$.

Remark 3. This characterization of the rate region $\mathcal{R}^{\mathcal{N}}$ in this proposition involves the union of infinitely many sets, making it non-explicit. As the frame length T increases, the approximation of $\mathcal{R}^{\mathcal{N}}$ by $\cup_{t=1,2,\dots,T}\frac{t}{t+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^t}$ becomes more accurate. Nevertheless, calculating $\widetilde{\mathcal{R}}^{\mathcal{N}^T}$ using generic algorithms for enumerating maximal independent sets of $\mathcal{N}^T$ can become computationally expensive as T grows. This computational complexity arises because a graph with n vertices can have up to $3^{n/3}$ maximal independent sets [38]. Therefore, although the approximation becomes more accurate with larger T, the computational cost of obtaining the exact characterization of $\mathcal{R}^{\mathcal{N}}$ can be prohibitive due to the exponential growth in the number of maximal independent sets for larger graphs.

Proof: As $\cup_{T=1,2,\dots}\frac{T}{T+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^T} \subset \mathcal{R}^{\mathcal{N}}$, we only need to show $\mathcal{R}^{\mathcal{N}} \subset \cup_{T=1,2,\dots}\frac{T}{T+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^T}$. For any $R \in \mathcal{R}^{\mathcal{N}}$ and $\epsilon > 0$, by Lemma 1, there exists a collision-free, periodic schedule S such that

$$R_S(l) \geq R(l) - \epsilon \text{ for every } l \in \mathcal{L}.$$

Let $T_0 \geq D^*/\epsilon$ be a period of S. We have $R_S \in \widetilde{\mathcal{R}}^{\mathcal{N}^{T_0}}$ and hence

$$\frac{1}{1+\epsilon}(R - \epsilon) \in \frac{T_0}{T_0 + D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^{T_0}}.$$

As the above holds for any $\epsilon > 0$, R is in the closure of $\cup_{T=1,2,\dots}\frac{T}{T+D^*}\widetilde{\mathcal{R}}^{\mathcal{N}^T}$. ■

2) Framed Scheduling: Framed scheduling is a special type of guarded scheduling where each link is either active or inactive simultaneously for all the timeslots within a frame, except for the guard interval. Framed scheduling is motivated by the network scheduling schemes extensively used in the existing wireless networks. We illustrate a framed schedule as follows, where $T_F = 5$ and $D^* = 2$:

$$\left[\begin{array}{c|ccccc|ccccc|ccccc|c} \cdots & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \end{array}\right]$$

Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$. Recall that $(\mathcal{L}, \mathcal{I}, \mathbf{0})$ is a network with the all-0 delay matrix and $\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$ can be characterized by the independent sets of $(\mathcal{L}, \mathcal{I})$ while ignoring the direction of edges. In other words, $\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$ is the convex hull of the indicator vectors of all the independent sets of $(\mathcal{L}, \mathcal{I})$ with the edge directions ignored. We prove that $\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$ is the achievable rate region of framed scheduling for $\mathcal{N}$ when $T_F \geq 3D^* + 1$.

Lemma 3. *A framed schedule S with a frame length $T_F \geq 3D^* + 1$ is collision-free if and only if for any link l that is active in a frame, for all $\theta \in \mathcal{I}(l)$, there exists a certain $l' \in \theta$ that is inactive in the same frame.*

Proof: A framed schedule is also a guarded schedule, where a collision can only be generated by links within the same frame. Therefore, the sufficiency of the lemma holds (even without the condition that $T_F \geq 3D^* + 1$).

To prove the necessary condition, consider that link l is active in the first frame, and for a certain $\theta \in \mathcal{I}(l)$, all $l' \in \theta$ are active in the first frame. As $|D_{\mathcal{L}}(l, l')| \leq D^*$ for all $l' \in \theta$, we have $D^* + D_{\mathcal{L}}(l, l') \in \{0, 1, \ldots, 2D^*\}$ for all $l' \in \theta$. As $T_F \geq 3D^* + 1$, for all $l' \in \theta$, $S(l', D^* + D_{\mathcal{L}}(l, l')) = 1$, and hence $S(l, D^*)$ has a collision. ■

Based on the above lemma, the following statement is straightforward.

Proposition 4. *Consider a network $(\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$. For a framed schedule of frame length $T_F \geq 3D^* + 1$, if its rate vector exists, the rate vector is in $(1 - D^*/T_F)\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$. Moreover, any rate vector in $\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$ can be achieved by collision-free, framed schedules.*

Proof outline: First, the scheduling rate within each frame is in $(1 - D^*/T_F)\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$. Therefore, if the average rate of all the frames converge, it must be also in $(1 - D^*/T_F)\mathcal{R}^{(\mathcal{L},\mathcal{I},\mathbf{0})}$. Second, an independent set of $(\mathcal{L}, \mathcal{I})$ with edge direction ignored can be used to design a collision-free framed schedule, and hence the achievable part can be shown using a large T_F. ■

For a network with a binary collision profile (i.e., $\mathcal{I}(l)$ is a subset of $\mathcal{L}$ for all $l \in \mathcal{L}$), the above discussion for framed scheduling can be improved by relaxing the condition $T_F \geq 3D^* + 1$ to $T_F \geq 2D^* + 1$ in Lemma 3 and Proposition 4. The proof of the necessary condition of Lemma 3 can be modified as follows: Consider a certain $l' \in \mathcal{I}(l)$ is active in the first frame. As $|D_{\mathcal{L}}(l, l')| \leq D^*$, there exists $t_0 \in \{0, 1, \ldots, D^*\}$ such that $t_0 + D_{\mathcal{L}}(l, l') \in \{0, 1, \ldots, D^*\}$. As $T_F \geq 2D^* + 1$, $S(l', t_0 + D_{\mathcal{L}}(l, l')) = 1$, and hence $S(l, t_0)$ has a collision.

IV. Scheduling Graphs and Rate Region

The characterization of the scheduling rate region $\mathcal{R}^{\mathcal{N}}$ using guarded scheduling (Proposition 3) cannot be exactly computed in finite time. In this section, we provide an explicit characterization of $\mathcal{R}^{\mathcal{N}}$ (Theorem 4 and Theorem 5 below) that enables the computation of this region in finite time. Our approach leverages the periodic structure of $\mathcal{N}^{\infty}$.

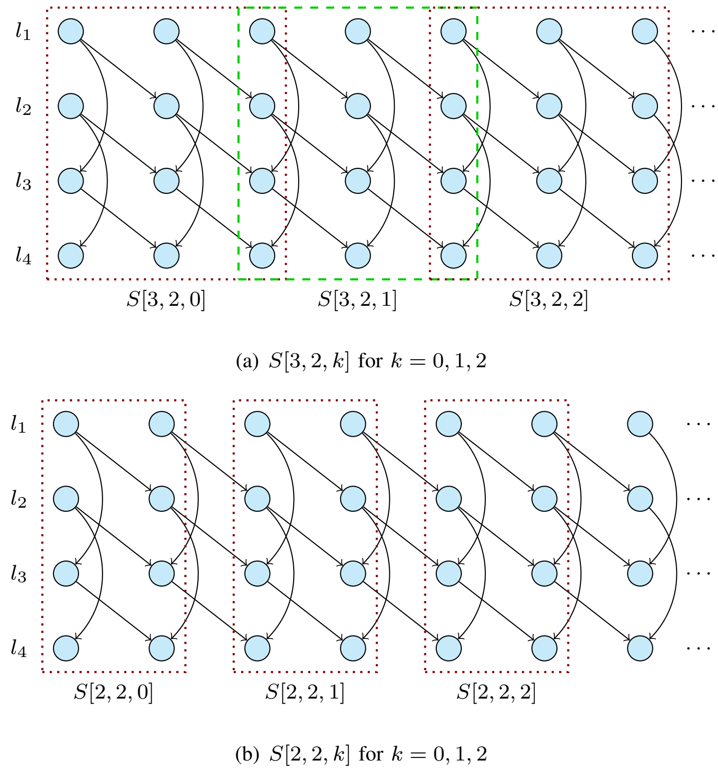


Fig. 4. Illustration of the associated part in the periodic graph of $S[T,Q,k]$.

We need some further concepts about directed graphs: In a directed graph $\mathcal{G}$, a *path* of length k is a sequence of vertices $v_0, v_1, \ldots, v_k$ where (v_i, v_{i+1}) $(i = 0, 1, \ldots, k-1)$ is a directed edge in $\mathcal{G}$. A path of length 0 is a vertex, while a path of length 1 is an edge. A path $(v_0, v_1, \ldots, v_k)$ is said to be *closed* if $v_k = v_0$. A path $(v_0, v_1, \ldots)$ of infinite length is said to be *periodic* if there exists a positive integer T such that $v_i = v_{i+T}$ for any $i \geq 0$, where each such value of T is called a period. For a periodic path with a period of T, the sub-path $(v_0, v_1, \ldots, v_T)$ is closed. A *cycle* in $\mathcal{G}$ is a closed path $(v_0, v_1, \ldots, v_k)$ where $v_i \neq v_j$ for any $0 \leq i \neq j \leq k-1$. In other words, the only repeated vertices in the cycle are the first and the last vertices. A cycle of length k is also called a k-cycle.

A. Scheduling Graphs and Collision-free Schedules

Recall that a schedule S is a matrix with columns indexed by $t \in \mathbb{Z}$. We begin by dividing S into submatrices, which are formed by consecutive columns, and then proceed to verify whether S is collision-free using these submatrices. For integers T, Q, and k satisfying $T \geq 1$ and $1 \leq Q \leq T$, we denote $S[T,Q,k]$ as the submatrix of S with columns $kQ, kQ+1, \ldots, kQ+T-1$. We refer to T as the *blocklength*, Q as the *step size*, and k as the *block index*. The definition is illustrated in Fig. 4. Note that $S[T,Q,0] = S[T,T,0]$ for all Q. When $Q < T$, there is overlap between $S[T,Q,k]$ and $S[T,Q,k+1]$. However, $S[T,T,k]$ and $S[T,T,k+1]$ are disjoint but adjacent.

For a positive integer T and a network $\mathcal{N}$, an $|\mathcal{L}| \times T$ binary matrices A is considered collision-free for $\mathcal{N}$ if $A = S'[T,T,0]$ for some collision-free schedule S', or equivalently, A represents an independent set of $\mathcal{N}^T$. Fix an integer Q with $1 \leq Q \leq T$. If a schedule S is collision-free, then $S[T,Q,k]$, $k = 0, 1, \ldots$ are all collision-free.

Conversely, we will show that for sufficiently large T, a schedule S is collision-free if $(S[T,Q,k], S[T,Q,k+1])$, $k = 0, 1, \dots$ satisfy a certain condition. To present this condition, we adopt a graphical approach that enables us to leverage results from graph theory conveniently. A pair of matrices (A_1, A_2), where $A_i \in \{0,1\}^{|\mathcal{L}|\times t_i}$, is also regarded as a matrix obtained by juxtaposing them.

Definition 5 (Scheduling graph)**.** For a network $\mathcal{N}$ and integers $1 \le Q \le T$, a *scheduling graph* is a directed graph, denoted by $(\mathcal{M}_T, \mathcal{E}_{T,Q})$, defined as follows: The vertex set $\mathcal{M}_T$ consists of all $|\mathcal{L}| \times T$ binary matrices that are collision-free for $\mathcal{N}$. The edge set $\mathcal{E}_{T,Q}$ includes all pairs of vertices (A, B) such that $A[T-Q, Q, 1] = B[T-Q, Q, 0]$ and $(A[Q, Q, 0], B)$ (considered as an $|\mathcal{L}| \times T$ matrix) is collision-free for $\mathcal{N}$.

The sets $\mathcal{M}_T$ and $\mathcal{E}_{T,Q}$ can be determined by the independent sets of $\mathcal{N}^T$ and $\mathcal{N}^{T+Q}$, respectively, as discussed in more detail in Sec. V-A. We also call $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ the *step-*Q scheduling graph. When $Q < T$, a necessary condition for $(A, B) \in \mathcal{E}_{T,Q}$ is that the last $T - Q$ columns of A and the first $T - Q$ columns of B are the same. Moreover, $(A, B) \in \mathcal{E}_{T,Q}$ if and only if $A = S[T, Q, 0]$ and $B = S[T, Q, 1]$ for a certain collision-free schedule S. We also write the step-T scheduling graph as $(\mathcal{M}_T, \mathcal{E}_T)$. A necessary and sufficient condition for $(A, B) \in \mathcal{E}_T$ is that (A, B) as an $|\mathcal{L}| \times 2T$ binary matrix represents an independent set of $\mathcal{N}^{2T}$.

Example 8 (Multihop line network)**.** We give $(\mathcal{M}_1, \mathcal{E}_1)$ of $\mathcal{N}^{\text{line}}_{4,1}$ as an example. Here $\mathcal{M}_1$ includes the 4×1 matrices v such that v can be a column of a certain collision-free schedule S of $\mathcal{N}^{\text{line}}_{4,1}$. We have $\mathcal{M}_1 = \{v_0, v_1, \dots, v_8\}$, where

$$\begin{bmatrix} v_0 & v_1 & \cdots & v_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

$\mathcal{E}_1$ includes all the pairs (v, v') such that $[v, v']$ is equal to two consecutive columns of a certain collision-free schedule, and can be denoted by the adjacency matrix:

$$\begin{array}{c} \\ v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{array}
\begin{array}{c} \begin{array}{ccccccccc} v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}. \qquad (9)$$

The following two theorems show that a collision-free schedule of a network $\mathcal{N}$ is equivalent to a directed path in a scheduling graph $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ with $T \ge 2D^*$. These results allow us to further investigate the scheduling problem using scheduling graphs.

Theorem 2. *Consider a network $\mathcal{N}$ and a schedule S. If S is collision-free for $\mathcal{N}$, then for any integers $1 \leq Q \leq T$, the sequence $(S[T,Q,k], k = 0, 1, \ldots)$ forms a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$.*

Proof: Suppose $\mathcal{S}$ is collision-free. We see that for $k = 0, 1, \ldots$, $S[T,Q,k]$ is collision-free for $\mathcal{N}$ and hence is in $\mathcal{M}_T$. Note that $S[T,Q,k]$ and $S[T,Q,k+1]$ are the first and the last T columns of $S[T+Q,Q,k]$, which is collision-free for $\mathcal{N}$. Hence, $(S[T,Q,k], S[T,Q,k+1]) \in \mathcal{E}_{T,Q}$. Therefore, $(S[T,Q,k], k = 0, 1, \ldots)$ is a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. ■

For $\mathcal{N}_{4,1}^{\text{line}}$, any schedule S that forms a path in $(\mathcal{M}_1, \mathcal{E}_1)$ as characterized in Example 8 is collision-free. However, the converse of Theorem 2 can only be proved in general when T is sufficiently large. The next example shows that for $T < 2D^*$, a schedule S that forms a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ may not be collision-free.

Example 9. Consider a network $\mathcal{N}_4 = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$, where $\mathcal{L} = \{l_1, l_2, l_3, l_4\}$. The collision sets of the links are

$$\begin{aligned} \mathcal{I}(l_1) &= \emptyset, & \mathcal{I}(l_2) &= \{\{l_1, l_3\}\}, \\ \mathcal{I}(l_3) &= \{\{l_2, l_4\}\}, & \mathcal{I}(l_4) &= \emptyset. \end{aligned}$$

The link-wise delay matrix $D_{\mathcal{L}}$ is

$$D_{\mathcal{L}} = \begin{bmatrix} * & * & * & * \\ -1 & * & 1 & * \\ * & -1 & * & 1 \\ * & * & * & * \end{bmatrix}. \tag{10}$$

For this network, the character

$$D^* = \max_{l \in \mathcal{L}} \max_{\theta \in \mathcal{I}(l)} \max_{l' \in \theta} |D_{\mathcal{L}}(l, l')| = 1. \tag{11}$$

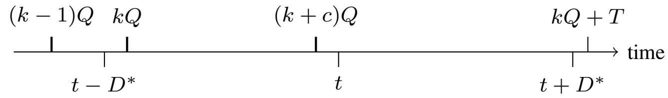
We illustrate that a schedule S that forms a path in $(\mathcal{M}_1, \mathcal{E}_1)$ may not be collision-free. First, we see that $(\mathcal{M}_1, \mathcal{E}_1)$ is a complete graph with the vertices set $\{0, 1\}^4$. Consider a schedule S with a submatrix S' formed by three consecutive columns:

$$S' = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

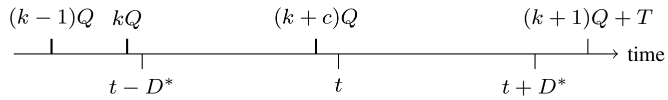
Because $S'(l_1, 0) = S'(l_3, 2) = 1$, $S'(l_2, 1)$ has a collision. As $S'(l_2, 1) = 1$, S is not collision-free.

The next theorem proves a converse of Theorem 2 for blocklength $T \geq 2D^*$. For binary collision, the converse of can be proved for $T \geq D^*$ (see Theorem 6).

Theorem 3. *Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ and a schedule S. If for certain integers T and Q such that $T \geq 2D^*$ and $T \geq Q \geq 1$, the sequence $(S[T,Q,k], k = 0, 1, \ldots)$ forms a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$, then S is collision-free.*



(a) case $0 \le t_0 < d$



(b) case $d \le t_0 < Q$

Fig. 5. Illustration of the proof of Theorem 3. A thick tick indicts the start position of a submatrix $S[T, Q, k]$, and a thin tick indicts the time.

Proof: Fix any $(l, t) \in \mathcal{L} \times \mathbb{Z}^+$ such that $S(l, t) = 1$, and fix any $\theta \in \mathcal{I}(l)$. To prove $S(l, t)$ is collision-free, we need to show that for a certain $l' \in \theta$ $S(l', t + D_{\mathcal{L}}(l, l')) = 0$. Find integers $c \ge 0$ and $0 \le d < Q$ such that $D^* = cQ + d$, and find integers k and $0 \le t_0 < Q$ such that $t = (k + c)Q + t_0$. For any $l' \in \theta$, we have

$$
\begin{aligned}
t' &\triangleq t + D_{\mathcal{L}}(l, l') \\
&\in [t - D^*, t + D^*] \\
&= [kQ + t_0 - d, kQ + t_0 - d + 2D^*].
\end{aligned}
$$

In the following, we discuss two cases of t_0: $0 \le t_0 < d$ and $d \le t_0 < Q$.

When $0 \le t_0 < d$, we have $-Q < t_0 - d < 0$. Hence for any $l' \in \theta$, as $T \ge 2D^*$, $(k-1)Q < t' < kQ + T$ (see Fig. 5 (a)). In other words, to verify the collision of $S(l, t)$ with respect to θ, we only need to consider $S[T, Q, k-1]$ and $S[T, Q, k]$. As $(S[T, Q, k-1], S[T, Q, k]) \in \mathcal{E}_{T,Q}$, we have $S[T, Q, k-1] = S'[T, Q, 0]$ and $S[T, Q, k] = S'[T, Q, 1]$ for a certain collision-free schedule S'. As $S(l, t) = S'(l, t_0 - d + Q + D^*) = 1$, we have $S(l', t') = S'(l', t_0 - d + Q + D^* + D_{\mathcal{L}}(l, l')) = 0$ for a certain $l' \in \theta$.

When $d \le t_0 < Q$, we have $0 \le t_0 - d < Q$. Hence for any $l' \in \theta$, as $T \ge 2D^*$, $kQ \le t' < (k+1)Q + T$ (see Fig. 5 (b)). In other words, to verify the collision of $S(l, t)$ with respect to θ, we only need to consider $S[T, Q, k]$ and $S[T, Q, k+1]$. As $(S[T, Q, k], S[T, Q, k+1]) \in \mathcal{E}_{T,Q}$, we have $S[T, Q, k] = S'[T, Q, 0]$ and $S[T, Q, k+1] = S'[T, Q, 1]$ for certain collision-free schedule S'. Therefore, as $S(l, t) = S'(l, t_0 - d + D^*) = 1$, we have $S(l', t') = S'(l', t_0 - d + D^* + D_{\mathcal{L}}(l, l')) = 0$ for a certain $l' \in \theta$.

For both cases, $S(l', t') = 0$ for a certain $l' \in \theta$. Therefore, $S(l, t)$ is collision-free. ■

B. Periodic Schedules and Scheduling Graphs

Theorem 2 and Theorem 3 together show that a collision-free schedule is equivalent to a directed path in a scheduling graph $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ with $T \ge 2D^*$. Hence we convert the independent set problem on a periodic hypergraph to a path problem on a scheduling graph. Although a scheduling graph has a finite size, the number of paths in it is infinite and the length of a path can be unbounded as well. We continue to study how to reduce the number and length of the paths based on periodic scheduling, which is rate region achieving as shown in Lemma 1.

We will establish a relationship between a periodic schedule and a closed path in a scheduling graph, thus providing a characterization of $\mathcal{R}^{\mathcal{N}}$ by cycles in the scheduling graph.

Definition 6. The rate vector R_P of a closed path $P = (A_0, A_1, \dots, A_k)$ in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ is defined as follows:

$$R_P = \frac{1}{kQ} \sum_{i=0}^{k-1} A_i \mathbf{1}_Q,$$

where $\mathbf{1}_Q$ is a length-T column vector with the first Q entries equal to 1 and the remaining $T-Q$ entries equal to 0.

Denote $\mathrm{cycle}(\mathcal{G})$ as the collection of all cycles in a directed graph $\mathcal{G}$. Define

$$\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})} = \mathrm{conv}\{R_C : C \in \mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_{T,Q})\},$$

where $\mathrm{conv}\mathcal{A}$ is the convex hull of a set $\mathcal{A}$. Since $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ is finite, $\mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_{T,Q})$ is finite and hence $\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$ is a closed convex polytope. We will show that when $T \geq 2D^*$, $\mathcal{R}^{\mathcal{N}} = \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$.

Lemma 4. *For a network $\mathcal{N}$ and a collision-free, periodic schedule S, the following statements hold:*

1) *For any period K of S, $(S[T,Q,i], i = 0, 1, \dots, K)$ forms a closed path in the scheduling graph $(\mathcal{M}_T, \mathcal{E}_{T,Q})$.*
2) *The rate vector R_S belongs to the set $\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$.*

Proof: By Theorem 2, $(S[T,Q,i], i = 0, 1, \dots)$ forms a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. As KQ is also a period of S, $S[T,Q,i] = S[T,Q,i+K]$. Therefore, the path $(S[T,Q,i], i = 1, 2, \dots)$ has a period K and hence $(S[T,Q,i], i = 0, 1, \dots, K)$ is a closed path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$.

A closed path can be decomposed into a sequence of (not necessarily distinct) cycles (see, e.g., [39]). Suppose $(S[T,Q,i], i = 0, 1, 2, \dots, K)$ has the decomposition of cycles $C_1, \dots, C_{K'}$ in $\mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_{T,Q})$, where C_i is of length k_i. Using this decomposition of the closed path, one obtains

$$R_S = \frac{1}{K} \sum_{i=1}^{K'} k_i R_{C_i} \in \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}.$$

■

Theorem 4. *For a network $\mathcal{N}$, we have $\mathcal{R}^{\mathcal{N}} \subset \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$ for integers $T \geq Q \geq 1$.*

Proof: Consider $R \in \mathcal{R}^{\mathcal{N}}$. By Lemma 1, for any $\epsilon > 0$, there exists a collision-free, periodic schedule S such that $R_S \not\succcurlyeq R - \epsilon$. By Lemma 4, $R_S \in \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$. As $\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$ is closed, we have $R \in \mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})}$. ■

For the general collision model, the converse of the above theorem ($\mathcal{R}^{(\mathcal{M}_T,\mathcal{E}_{T,Q})} \subset \mathcal{R}^{\mathcal{N}}$) can be proved for blocklength $T \geq 2D^*$. To show the converse, we first show that for a periodic schedule, it is sufficient to check a sufficiently long part of the schedule to verify whether it is collision-free.

Lemma 5. *For a network $\mathcal{N}$ and integers $T \geq 2D^*$ and $1 \leq Q \leq T$, suppose S is a periodic schedule with period kQ such that $(S[T,Q,i], i = 0, \dots, k)$ is a closed path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. Then S is collision-free for $\mathcal{N}$.*

Proof: Let $A_i = S[T,Q,i]$ for $i = 0, 1, \dots, k-1$. Fix any integer $i = ak + b$ where $a \geq 0$ and $0 \leq b \leq k-1$. First, $S[T,Q,i] = S[T,Q,b] = A_b \in \mathcal{M}_T$. Second, $(S[T,Q,i], S[T,Q,i+1]) = (S[T,Q,b], S[T,Q,b+1]) =$

$(A_b, A_{b+1}) \in \mathcal{E}_{T,Q}$. Therefore, the sequence $(S[T,Q,i], i = 1, 2, \ldots)$ is a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. As $T \geq 2D^*$, by Theorem 3, S is collision-free. ■

Theorem 5. *For a network $\mathcal{N}$ with a general collision profile and any integers T and Q such that $T \geq 2D^*$ and and $1 \leq Q \leq T$, it holds that $\mathcal{R}^{\mathcal{N}} \supset \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$.*

Proof: Fix $R \in \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$. We can write

$$R = \sum_{C \in \mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_{T,Q})} \alpha_C R_C,$$

where $\alpha_C \geq 0$ and $\sum_{C \in \mathrm{cycle}(\mathcal{M}_T, \mathcal{E}_T)} \alpha_C = 1$. For a cycle $C = (C_0, C_1, \ldots, C_k)$ in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$, we define a schedule S with period kQ such that $S[T,Q,i] = C_i$ for $i = 0, 1, \ldots, k-1$. By Lemma 5, S is collision-free and hence $R_C = R_S \in \mathcal{R}^{\mathcal{N}}$. As $\mathcal{R}^{\mathcal{N}}$ is convex (see Lemma 2), we have $R \in \mathcal{R}^{\mathcal{N}}$. ■

Theorem 4 and Theorem 5 together give an explicit characterization of $\mathcal{R}^{\mathcal{N}}$, i.e., when $T \geq 2D^*$, $\mathcal{R}^{\mathcal{N}} = \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$, where $\mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$ is explicitly determined by the cycles in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. Now we see that using a scheduling graph, only a finite number of cycles are required for determining the scheduling rate region. Moreover, from the proofs of Lemma 4 and Lemma 5, we also see that a periodic, collision-free schedule can be formed by cycles of $(\mathcal{M}_T, \mathcal{E}_{T,Q})$.

C. Enhanced Results for Binary Network Model

For the binary collision model, Theorem 5 can be proved for $T \geq D^*$ (see Theorem 7 below). For the binary collision model, the collision set $\mathcal{I}(l)$ has the property that for any $\theta \in \mathcal{I}(l)$, $|\theta| = 1$. In this case, we also write $\mathcal{I}(l)$ as a subset of $\mathcal{L}$, and the formula of character D^* given in (6) can be simplified as

$$D^*_{\mathcal{N}} = \max_{l \in \mathcal{L}} \max_{l' \in \mathcal{I}(l)} |D_{\mathcal{L}}(l, l')|.$$

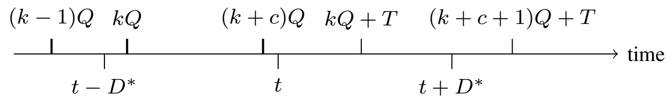
The following theorem improves Theorem 3 for the binary collision model with the lower bound on T improved from $2D^*$ to D^*.

Theorem 6. *Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$ with a binary collision profile and a schedule S. If for certain integers T and Q such that $T \geq D^*$ and $T \geq Q \geq 1$, the sequence $(S[T,Q,k], k = 0, 1, \ldots)$ forms a path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$, then S is collision-free.*

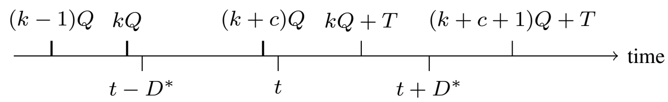
Proof: Fix any $(l,t) \in \mathcal{L} \times \mathbb{Z}^+$ such that $S(l,t) = 1$, and fix any $l' \in \mathcal{I}(l)$. To prove $S(l,t)$ is collision-free, we need to show that $S(l', t + D_{\mathcal{L}}(l,l')) = 0$. Find integers $c \geq 0$ and $0 \leq d < Q$ such that $D^* = cQ + d$, and find integers k and $0 \leq t_0 < Q$ such that $t = (k+c)Q + t_0$. For l', we have

$$\begin{aligned} t' &\triangleq t + D_{\mathcal{L}}(l,l') \\ &\in [t - D^*, t + D^*] \\ &= [kQ + t_0 - d, kQ + t_0 - d + 2D^*]. \end{aligned}$$

In the following, we discuss two cases of t_0: $0 \leq t_0 < d$ and $d \leq t_0 < Q$.



(a) case $0 \leq t_0 < d$



(b) case $d \leq t_0 < Q$

Fig. 6. Illustration of the proof of Theorem 6. A thick tick indicts the start position of a submatrix $S[T,Q,k]$, and a thin tick indicts the time.

When $0 \leq t_0 < d$, we have $-Q < t_0 - d < 0$, and hence $(k-1)Q < t' < kQ + 2D^*$. See Fig. 6 (a) for an illustration. Consider three subcases of t':

- $t' \in [(k-1)Q, kQ+T-1]$. Note that $t \in [kQ, kQ+T-1]$. As $(S[T,Q,k-1], S[T,Q,k]) \in \mathcal{E}_{T,Q}$, we have $S[T,Q,k-1] = S'[T,Q,0]$ and $S[T,Q,k] = S'[T,Q,1]$ for certain collision-free schedule S'. Therefore, as $S(l,t) = S'(l, t_0 + (1+c)Q) = 1$, $S(l',t') = S'(l', t_0 + (1+c)Q + D_{\mathcal{L}}(l,l')) = 0$.
- $t' \in [(k+c)Q, (k+c)Q+T-1]$. Note that $t \in [(k+c)Q, (k+c)Q+T-1]$ too. As $S[T,Q,k+c] \in \mathcal{M}_T$, we have $S[T,Q,k+c] = S'[T,0]$ for certain collision-free schedule S'. Therefore, as $S(l,t) = S'(l,t_0) = 1$, $S(l',t') = S'(l', t_0 + D_{\mathcal{L}}(l,l')) = 0$.
- $t' \in [(k+c+1)Q, (k+c+1)Q+T-1]$. Note that $t \in [(k+c)Q, (k+c)Q+T-1]$. As $(S[T,Q,k+c], S[T,Q,k+c+1]) \in \mathcal{E}_{T,Q}$, we have $S[T,Q,k+c] = S'[T,Q,0]$ and $S[T,Q,k+c+1] = S'[T,Q,1]$ for certain collision-free schedule S'. Therefore, as $S(l,t) = S'(l,t_0) = 1$, $S(l',t') = S'(l', t_0 + D_{\mathcal{L}}(l,l')) = 0$.

We see that for all the subcases of t', $S(l',t') = 0$.

When $d \leq t_0 < Q$, we have $0 \leq t_0 - d < Q$, and hence $kQ \leq t' < kQ+T+Q$. See Fig. 6 (b) for an illustration. Consider three subcases of t':

- $t' \in [kQ, kQ+T-1]$. Note that $t \in [kQ, kQ+T-1]$.
- $t' \in [(k+c)Q, (k+c)Q+T-1]$. Note that $t \in [(k+c)Q, (k+c)Q+T-1]$ too.
- $t' \in [(k+c+1)Q, (k+c+1)Q+T-1]$. Note that $t \in [(k+c)Q, (k+c)Q+T-1]$.

These subcases can be analyzed similarly as when $0 \leq t_0 < d$, and hence $S(l',t') = 0$.

For both cases of t_0, $S(l',t') = 0$ for any $l' \in \mathcal{I}(l)$. Therefore, $S(l,t)$ is collision-free. ∎

The following lemma improves Lemma 5 for the binary collision model. The proof is the same as that of Lemma 5 except that Theorem 6 is applied instead of Theorem 3.

Lemma 6. *For a network $\mathcal{N}$ with a binary collision profile and integers T and Q such that $T \geq D^*$ and $1 \leq Q \leq T$, suppose S is a periodic schedule with period kQ such that $(S[T,Q,i], i = 0,\ldots,k)$ is a closed path in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$. Then S is collision-free for $\mathcal{N}$.*

The following theorem improves Theorem 5 for the binary collision model. The proof is the same as that of

Theorem 5 except that Lemma 6 is applied instead of Lemma 5.

Theorem 7. *For a network $\mathcal{N}$ with a binary collision profile and any integers T and Q such that $T \geq D^*$ and $1 \leq Q \leq T$, it holds that $\mathcal{R}^{\mathcal{N}} \supset \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_{T,Q})}$.*

V. Algorithms for Calculating Scheduling Rate Region

In this section, our focus is on developing algorithms for calculating the scheduling rate region of a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$. Based on the discussion in Sec. IV, we understand that the rate region of $\mathcal{N}$ is determined by cycles of the scheduling graph $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ with a sufficiently large T. Therefore, a straightforward approach to compute the rate region is to apply the existing cycle enumerating algorithms on the scheduling graphs (e.g., Johnson's algorithm [31]). Though we may choose the minimum value of T to reduce the computation cost, the straightforward approach in general incurs a high computation cost. Note that $|\mathcal{M}_T| = O(2^{|\mathcal{L}|T})$ and the number of cycles in $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ can exceed $2^{|\mathcal{M}_T|}$ [31]. Consequently, cycle enumerating algorithms designed for generic graphs tend to exhibit a steep increase in computation cost as a function of $|\mathcal{L}|T$.

In the existing scheduling researches, to avoid the high the computational complexity of enumerating all the maximal independent sets, algorithms have been developed to achieve a subset of the scheduling rate region [24], [40]. Similarly, due to the even greater computational challenge of enumerating all cycles in a scheduling graph, it is also a reasonable approach to enumerate cycles up to a specific length k, which can provide an approximation of the scheduling rate region. Although the maximum cycle length theoretically is $O(|\mathcal{M}_T|)$, both analytical and numerical evidence suggests that a small value of k can yield a good approximation of the rate region.

Though an algorithm designed for generic graphs can be applied on a scheduling graph to enumerate cycles up to a specific length (e.g., [41]), its running time may not be optimal since it does not utilize the specific structure of the scheduling graph. In this section, we present an approach specifically designed for the step-T scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$. By leveraging a dominance property of $(\mathcal{M}_T, \mathcal{E}_T)$, we refine the characterization of the scheduling rate region using only subgraphs of $(\mathcal{M}_T, \mathcal{E}_T)$. We derive algorithms that calculate the subset of the scheduling rate region generated by the cycles of $(\mathcal{M}_T, \mathcal{E}_T)$ up to a specific length. Numerical evaluations demonstrate that our algorithms can achieve faster computation compared to using generic cycle enumeration algorithms directly on $(\mathcal{M}_T, \mathcal{E}_T)$, particularly when dealing with larger networks. Moreover, the techniques developed here can be adopted in the next section for maximizing a linear function of the rate vectors.

We denote $\max_{\succcurlyeq} \mathcal{A}$ as the set of maximal elements in the partially ordered set $(\mathcal{A}, \succcurlyeq)$. In other words, $\max_{\succcurlyeq} \mathcal{A}$ is the smallest subset $\mathcal{B}$ in $\mathcal{A}$ such that any element of $\mathcal{A}$ is dominated by some elements in $\mathcal{B}$. A sequence of matrices $A = (A_0, A_1, \ldots)$, where $A_i \in \{0,1\}^{|\mathcal{L}| \times t_i}$, is regarded as a matrix obtained by juxtaposing $A_0, A_1, \ldots$. Therefore, the relation $\succcurlyeq$ and $\preccurlyeq$ defined on matrices can be applied to pairs of sequence of matrices. For two real numbers a and b, we define $a \wedge b$ as the minimum of a and b. For two matrices $A = (a_{ij})$ and $B = (b_{ij})$ of the same size, we define $A \wedge B = (a_{ij} \wedge b_{ij})$. When A and B are binary matrices, $A \wedge B$ is the matrix resulting from the bitwise AND operation.

Algorithm 1 A algorithm for calculating $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ from $\mathcal{M}^*_{T+Q}$. The function SchedGraph recursively call ADDC to add vertices and edges to $(\mathcal{M}_T, \mathcal{E}_{T,Q})$.

function SCHEDGRAPH
    Input: $\mathcal{M}^*_{T+Q}$, T, Q
    Output: $(\mathcal{M}_T, \mathcal{E}_{T,Q})$
    $(\mathcal{M}_T, \mathcal{E}_{T,Q}) \leftarrow (\emptyset, \emptyset)$
    for each $C \in \mathcal{M}^*_{T+Q}$ **do**
        ADDC(C)
    procedure ADDC
        Input: C
        Output: update $(\mathcal{M}_T, \mathcal{E}_{T,Q})$
        if $C[T,Q,0] \in \mathcal{M}_T$, $C[T,Q,1] \in \mathcal{M}_T$ and $(C[T,Q,0], C[T,Q,1]) \in \mathcal{E}_{T,Q}$ **then**
            return
        $\mathcal{M}_T \leftarrow \mathcal{M}_T \cup \{C[T,Q,0], C[T,Q,1]\}$
        $\mathcal{E}_T \leftarrow \mathcal{E}_T \cup \{(C[T,Q,0], C[T,Q,1])\}$
        if C is the all-zero matrix **then**
            return
        for each $C' \precneqq C$ **do**
            ADDC(C')

A. Calculation of Scheduling Graphs

Before delving into our approach to the rate region, let's discuss the calculation of scheduling graphs. We first establish the equivalence between $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ and $\mathcal{M}_{T+Q}$. According to Definition 5, $\mathcal{M}_{T+Q}$ represents the subset of $|\mathcal{L}| \times (T+Q)$ binary matrices that correspond to the independent sets of $\mathcal{N}^{T+Q}$. On one hand, $\mathcal{E}_{T,Q}$ consists of pairs $(A,B) \in \mathcal{M}_T \times \mathcal{M}_T$ such that $(A[Q,Q,0], B) \in \mathcal{M}_{T+Q}$ and the last $T-Q$ columns of A and the first $T-Q$ columns of B are the same. On the other hand, for every $|\mathcal{L}| \times (T+Q)$ binary matrix $C \in \mathcal{M}_{T+Q}$, both $C[T,Q,0]$ and $C[T,Q,1]$ are elements of $\mathcal{M}_T$, resulting in $(C[T,Q,0], C[T,Q,1]) \in \mathcal{E}_{T,Q}$. Therefore, the calculation of $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ is equivalent to the calculation of $\mathcal{M}_{T+Q}$.

Denote $\mathcal{M}^*_t = \max_{\succcurlyeq} M_t$ as the collection of the $|\mathcal{L}| \times t$ binary matrices that represent the *maximal* independent sets of $\mathcal{N}^t$. When dealing with a binary collision model, the Bron–Kerbosch algorithm and its refinements [42]–[45] can be employed to enumerate $\mathcal{M}^*_t$. In the case of a general collision model where $\mathcal{N}^t$ forms a hypergraph, the corresponding problem of finding maximal independent sets has been discussed in [46]–[48]. The worst-case complexity of the Bron–Kerbosch algorithm is $O(3^{n/3})$, where n is the number of vertices in the network [43]. In our experience, the vertex pivoting technique [43], [44] can greatly improve the running time of the Bron–Kerbosch algorithm for $\mathcal{N}^t$.

For the straightforward approach to calculating the scheduling rate region, we require $\mathcal{M}_{T+Q}$ rather than just

$\mathcal{M}^*_{T+Q}$. Given $\mathcal{M}^*_{T+Q}$, we can generate $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ as follows: For each $C \in \mathcal{M}^*_{T+Q}$, and recursively for each $C' \preccurlyeq C$, we add $C'[T,Q,0]$ and $C'[T,Q,1]$ to $\mathcal{M}_T$, and we add $(C'[T,Q,0], C'[T,Q,1])$ to $\mathcal{E}_{T,Q}$. In Algorithm 1, we provide pseudocode for calculating $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ from $\mathcal{M}^*_{T+Q}$. Line 10 checks whether C has already been added to $(\mathcal{M}_T, \mathcal{E}_{T,Q})$ before.

In our approach to scheduling the rate region (to be elaborated in this section), we will use two subgraphs of the step-T scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$. Therefore, we do not require the computation of the entire scheduling graph based on $\mathcal{M}^*_{2T}$. The first subgraph of interest corresponds to $\mathcal{M}^*_{2T}$. If we consider $(A,B) \in \mathcal{E}_T$ as an $|\mathcal{L}| \times 2T$ binary matrix, then $(A,B) \in \mathcal{M}_{2T}$. Thus, we can express $\mathcal{E}_T = \mathcal{M}_{2T}$. Let $\mathcal{E}^* = \max_{\succcurlyeq} \mathcal{E}_T = \mathcal{M}^*_{2T}$, and let

$$\mathcal{M}^*_L = \{B : (B, B') \in \mathcal{E}^* \text{ for certain } B'\},$$

$$\mathcal{M}^*_R = \{B' : (B, B') \in \mathcal{E}^* \text{ for certain } B\}.$$

As $\mathcal{E}^* \subset \mathcal{M}^*_L \times \mathcal{M}^*_R$, $\mathcal{E}^*$ can be represented using an adjacency matrix with rows and columns indexed by elements in $\mathcal{M}^*_L$ and $\mathcal{M}^*_R$, respectively. We see that $(\mathcal{M}^*_L, \mathcal{M}^*_R, \mathcal{E}^*)$ is just another representation of $\mathcal{M}^*_{2T}$ and serves as a subgraph of $(\mathcal{M}_T, \mathcal{E}_T)$. The second subgraph of interest can be induced by $(\mathcal{M}^*_L, \mathcal{M}^*_R, \mathcal{E}^*)$, which will be discussed later in this section.

B. Dominance Property

According to the definition of collision, if a schedule is collision-free, then the schedule obtained by inactivating some entries is also collision-free. In other words, if $A \in \mathcal{M}_T$, then any $A' \preccurlyeq A$ is also in $\mathcal{M}_T$. The similar property applies to edges and paths in $(\mathcal{M}_T, \mathcal{E}_T)$. For two sequences $A = (A_0, A_1, \ldots)$ and $B = (B_0, B_1, \ldots)$ of the same length (which can be unbounded) with $A_i, B_i \in \{0,1\}^{|\mathcal{L}| \times T}$, we say A *dominates* B if $A \succcurlyeq B$.

Lemma 7 (Basic dominance property for $(\mathcal{M}_T, \mathcal{E}_T)$)**.** *For any $k \geq 0$, if $A = (A_0, A_1, \ldots, A_k)$ is a path in $(\mathcal{M}_T, \mathcal{E}_T)$, then any $B = (B_0, B_1, \ldots, B_k)$ with $B \preccurlyeq A$ is a path in $(\mathcal{M}_T, \mathcal{E}_T)$.*

Proof: For any edge $(A', A'') \in \mathcal{E}_T$ and any (B', B''), if $(B', B'') \preccurlyeq (A', A'')$, then (B', B'') is also an edge of $(\mathcal{M}_T, \mathcal{E}_T)$. The lemma can then be proved by checking $(B_i, B_{i+1}) \preccurlyeq (A_i, A_{i+1})$ for $i = 0, 1, \ldots, k-1$. ■

We now define some notations for presenting a main dominance property. Denote $\mathcal{P}_k$ as the set of length-k paths and $\mathcal{C}_k$ as the set of length-k cycles in $(\mathcal{M}_T, \mathcal{E}_T)$. Let

$$\mathcal{P}^*_k = \max_{\succcurlyeq} \mathcal{P}_k \quad \text{and} \quad \mathcal{C}^*_k = \max_{\succcurlyeq} \mathcal{C}_k.$$

In other words, the elements in $\mathcal{P}^*_k$ and $\mathcal{C}^*_k$ are the maximal paths and cycles, respectively, with respect to the partial order $\succcurlyeq$. Note that $\mathcal{C}_k \subset \mathcal{P}_k$, but $\mathcal{C}^*_k$ is not necessarily a subset of $\mathcal{P}^*_k$. We are going to show that maximal paths and cycles are sufficient for characterizing the scheduling rate region.

Let

$$\mathcal{R}_k = \operatorname{conv}\{R_C : C \in \cup_{i=1}^k \mathcal{C}_i\}, \tag{12}$$

where R_C is defined in Definition 6 with $Q = T$. It is worth noting that as k becomes sufficiently large, $\mathcal{R}_k$ becomes $\mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$, which is equal to $\mathcal{R}^{\mathcal{N}}$ when $T \geq 2D^*$ (or D^* for binary collision). Therefore, when $T \geq 2D^*$

(or D^* for binary collision) and k is sufficiently large, $\mathcal{R}_k$ is the scheduling rate region. The *lower shadow* of a set $\mathcal{A} \subset (\mathbb{R}^+)^{m\times n}$, denoted as $\mathrm{dom}\mathcal{A}$, refers to the collection of all $B \in (\mathbb{R}^+)^{m\times n}$ that are dominated by some elements in $\mathcal{A}$ [49], [50]. The following lemma states that the lower shadow of $\mathcal{R}_k$ is equal to the set $\mathcal{R}_k$ itself.

Lemma 8. *For each $k \geq 1$, $\mathrm{dom}\mathcal{R}_k = \mathcal{R}_k$.*

Proof: As $\mathcal{R}_k \subset \mathrm{dom}\mathcal{R}_k$, we show $\mathrm{dom}\mathcal{R}_k \subset \mathcal{R}_k$. For $R \in \mathrm{dom}\mathcal{R}_k$, there exists $R' \in \mathcal{R}_k$ such that $R' \succcurlyeq R$. Let $\mathcal{C}^* = \cup_{i=1}^k \mathcal{C}_i$. We can write $R' = \sum_{C'\in\mathcal{C}^*} \alpha_{C'} R_{C'}$, where $\alpha_{C'} \geq 0$ and $\sum_{C'\in\mathcal{C}^*} \alpha_{C'} = 1$. Assume $R(l) < R'(l)$ for a certain $l \in \mathcal{L}$. For each $C' \in \mathcal{C}^*$, construct a cycle C by setting the entries of the matrices in C' indexed by l to zero. Let $\alpha_l = \frac{R(l)}{R'(l)}$ and

$$R'' = \alpha_l \sum_{C'\in\mathcal{C}^*} \alpha_{C'} R_{C'} + (1-\alpha_l) \sum_{C'\in\mathcal{C}^*} \alpha_{C'} R_C.$$

We have $R'' \in \mathcal{R}_k$ as $C' \in \mathcal{C}^*$, $R''(l) = R(l)$ and $R''(l') = R'(l')$ for $l' \neq l$. By repeating the similar procedure for all the other links l' with $R'(l') > R(l')$, we can convert R'' to R and hence prove $R \in \mathcal{R}_k$. ∎

For a path $P = (A_0, A_1, \ldots, A_k)$ of length k, we define $\mathrm{cl}(P)$ as the closed path generated by

$$(A_0 \wedge A_k, A_1, \ldots, A_{k-1}, A_0 \wedge A_k).$$

Hence, the rate vector $R_{\mathrm{cl}(P)}$ is well-defined, following Definition 6 with $Q = T$. The following theorem shows that $\mathcal{R}_k$ can be determined by $\mathcal{P}_i^*$ and $\mathcal{C}_i^*$, $i = 1, \ldots, k$.

Theorem 8. *For a scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$ and any integer $k \geq 1$,*

$$\begin{aligned}\mathcal{R}_k &= \mathrm{dom\ conv}\{R_C : C \in \cup_{i=1}^k \mathcal{C}_i^*\} \\ &= \mathrm{dom\ conv}\{R_{\mathrm{cl}(P)} : P \in \cup_{i=1}^k \mathcal{P}_i^*\}.\end{aligned}$$

Proof: To simplify the notation, we use in the proof

$$\begin{aligned}\mathcal{A}^* &= \mathrm{dom\ conv}\{R_C : C \in \cup_{i=1}^k \mathcal{C}_i^*\}, \\ \mathcal{B}^* &= \mathrm{dom\ conv}\{R_{\mathrm{cl}(P)} : P \in \cup_{i=1}^k \mathcal{P}_i^*\}.\end{aligned}$$

By definition, $\mathcal{R}_k \subset \mathcal{A}^*$.

For $A = (A_0, \ldots, A_{k-1}, A_0) \in \mathcal{C}_k^*$, there exists $P = (B_0, \ldots, B_k) \in \mathcal{P}_k^*$ such that $B \succcurlyeq A$. As $B_0 \succcurlyeq A_0$ and $B_k \succcurlyeq A_0$, $B_0 \wedge B_k \succcurlyeq A_0$. So, $\mathrm{cl}(P) \succcurlyeq A$, and hence $R_A \preccurlyeq \mathrm{conv}\{R_{\mathrm{cl}(P)} : P \in P_k^*\}$. Therefore, $\mathcal{A}^* \subset \mathcal{B}^*$.

Last, for any $P = (B_0, \ldots, B_k) \in \mathcal{P}_k^*$, if $\mathrm{cl}(P)$ is a cycle, then $R_{\mathrm{cl}(P)} \in \mathcal{R}_k$. When $k = 1$, $\mathrm{cl}(P)$ must be a cycle. When $k > 1$, if $\mathrm{cl}(P)$ is not a cycle, it can be decomposed into multiple cycles, each of which is of length strictly less than k. Hence, $R_{\mathrm{cl}(P)} \in \mathcal{R}_{k-1} \subset \mathcal{R}_k$. The proof is completed by $\mathcal{B}^* \subset \mathrm{dom}(\mathcal{R}_k) = \mathcal{R}_k$ where the equality follows from Lemma 8. ∎

C. An Incremental Approach for Rate Region

We are motivated to study the calculation of $\mathcal{R}_k$ due to it's relation to the rate region. Based on Theorem 8, we will derive an approach to calculate $\mathcal{R}_k$ using $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$.

As $\mathcal{P}_1^* = \mathcal{E}^*$, according to Theorem 8, we have

$$\mathcal{R}_1 = \mathrm{dom\ conv}\{R_{\mathrm{cl}(P)} : P \in \mathcal{E}^*\}. \tag{13}$$

We can use an incremental approach to calculate $\mathcal{R}_k$ for $k \geq 2$. Denote $\mathbf{1}$ as a column vector with all entries equal to 1, where the length is known from the context. For $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$, let

$$\mathcal{W}_2(A, B) = \max_{\succcurlyeq}\{(B_1 \wedge A_2)\mathbf{1} : (A, B_1), (A_2, B) \in \mathcal{E}^*\}. \tag{14}$$

For $k \geq 3$, $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$, let

$$\mathcal{W}_k(A, B) = \max_{\succcurlyeq} \bigcup_{B' \in \mathcal{M}_R^*} \left(\mathcal{W}_{k-1}(A, B') + \{(B' \wedge A')\mathbf{1} : (A', B) \in \mathcal{E}^*\}\right), \tag{15}$$

where the addition of two sets $\mathcal{A} + \mathcal{B}$ is defined as $\{\mathbf{a} + \mathbf{b} : \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$. The next theorem justifies the use of $\mathcal{W}_k(A, B)$ to characterize the rate region $\mathcal{R}_k$. The proof of this theorem will be provided at the end of this subsection.

Theorem 9. *Consider the step-T scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$. For $k \geq 1$, we have $\mathcal{R}_k = \mathrm{dom\ conv}\ \mathcal{R}_1^* \cup \mathcal{R}_2^* \cup \cdots \cup \mathcal{R}_k^*$, where*

$$\mathcal{R}_1^* = \frac{1}{T} \max_{\succcurlyeq}\{(A \wedge B)\mathbf{1} : (A, B) \in \mathcal{E}^*\},$$

and for $i \geq 2$

$$\mathcal{R}_i^* = \frac{1}{iT} \max_{\succcurlyeq} \bigcup_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} \left(\mathcal{W}_i(A, B) + \{(A \wedge B)\mathbf{1}\}\right). \tag{16}$$

Before presenting the proof of Theorem 9, we discuss the algorithms for calculating $\mathcal{W}_k(A, B)$ and $\mathcal{R}_k^*$. According to Theorem 9, using $\mathcal{R}_1^*, \ldots, \mathcal{R}_k^*$, the vertex representation of the convex polytope $\mathcal{R}_k$ can be derived, which can then be converted to the half-space representation [51].

1) Algorithm for Rate Region Calculation: Algorithm 2 provides the pseudocode for calculating $\mathcal{W}_k(A, B)$ incrementally using (15), and Algorithm 3 provides the pseudocode for calculating $\mathcal{R}_k^*$ using the formula in Theorem 9. These algorithms assume that $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ has already been calculated (as mentioned in Sec.V-A). The explanations for these two algorithms are as follows. To simplify the notation, we write

$$\mathcal{W}_k \triangleq (\mathcal{W}_k(A, B), A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*).$$

In Algorithm 2, three functions are provided: W2AB, WAB and MAXADD. The W2AB function calculates $\mathcal{W}_2$ using (14), and the WAB function calculates $\mathcal{W}_k$ from $\mathcal{W}_{k-1}$ using (15). Line 7 and Line 16 call the function MAXADD to add a vector $\mathbf{u}$ to a set $\mathcal{S}$ that contains all the existing maximal vectors. If $\mathbf{u}$ is dominated by some vector in $\mathcal{S}$, $\mathcal{S}$ remains unchanged. If some vectors in $\mathcal{S}$ are dominated by $\mathbf{u}$, they are deleted from $\mathcal{S}$, followed by adding $\mathbf{u}$ to $\mathcal{S}$.

The computation cost of WAB depends on the size of $\mathcal{W}_k(A, B)$. Let

$$W_k = \max_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} |\mathcal{W}_k(A, B)|.$$

Algorithm 2 The pseudocode for calculating $\mathcal{W}_k$ includes two functions: W2AB, which calculates $\mathcal{W}_2$, and WAB, which calculates $\mathcal{W}_k$ from $\mathcal{W}_{k-1}$ for any $k > 2$. Additionally, there is a helper function called MAXADD, which is called by both W2AB and WAB to add an element to a set and output the maximal subset.

function W2AB
    Input: $\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*$
    Output: $\mathcal{W}_2$
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
        $\mathcal{W}_2(A, B) \leftarrow \emptyset$
        for each B_1 s.t. $(A, B_1) \in \mathcal{E}^*$ and A_2 s.t. $(A_2, B) \in \mathcal{E}^*$ **do**
            $\mathcal{W}_2(A, B) \leftarrow$ MAXADD($\mathcal{W}_2(A, B)$, $(B_1 \wedge A_2)\mathbf{1}$)
    return $\mathcal{W}_2$
function WAB
    Input: $\mathcal{W}_{k-1}, \mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*$
    Output: $\mathcal{W}_k$
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
        $\mathcal{W}_k(A, B) \leftarrow \emptyset$
        for each $B' \in \mathcal{M}_R^*$ and $A' \in \mathcal{M}_L^*$ s.t. $(A', B) \in \mathcal{E}^*$ **do**
            for each $\mathbf{r} \in \mathcal{W}_{k-1}(A, B')$ **do**
                $\mathcal{W}_k(A, B) \leftarrow$ MAXADD($\mathcal{W}_k(A, B)$, $\mathbf{r} + (B' \wedge A')\mathbf{1}$)
    return $\mathcal{W}_k$
function MAXADD
    Input: a set $\mathcal{S}$ of maximal vectors, a vector $\mathbf{u}$
    Output: $\max_{\succcurlyeq} \mathcal{S} \cup \{\mathbf{u}\}$
    for each $\mathbf{r} \in \mathcal{S}$ **do**
        if $\mathbf{u} \preccurlyeq \mathbf{r}$ **then**
            return
        if $\mathbf{r} \preccurlyeq \mathbf{u}$ **then**
            $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\mathbf{r}\}$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{u}\}$
    return $\mathcal{S}$

According to the definitions in (14) and (15), we have $W_2 \leq |\mathcal{M}_L^*||\mathcal{M}_R^*|$, and for $k > 2$, $W_k \leq |\mathcal{M}_L^*||\mathcal{M}_R^*|W_{k-1}$. The computation cost from W_{k-1} to W_k using WAB can be estimated as

$$O(|\mathcal{M}_L^*|^2|\mathcal{M}_R^*|^2 W_{k-1}(W_k + |\mathcal{L}|T))$$

integer and logical operations.

In Algorithm 3, two functions are provided: RateRegion and CALRR. The RateRegion function takes an integer $k_{\max}$ as input and calculates $\mathcal{R}_k^*$ for $k = 1, 2, \ldots, k_{\max}$ as output. The function calls W2AB and WAB to obtain $\mathcal{W}_k$ for $k = 2, \ldots, k_{\max}$ and then calculates $\mathcal{R}_k^*$ by calling CALRR on $\mathcal{W}_k$, using the formula provided in Theorem 9.

The value of $k_{\max}$ has an impact on both the computation cost and the subset of the rate region obtained. The computation cost of CALRR for $\mathcal{W}_k$ is $O(|\mathcal{M}_L^*|^2|\mathcal{M}_R^*|^2 W_k^2)$ integer and logic operations. Assuming $W_k \geq |\mathcal{L}|T$, the overall computation cost of RateRegion is $O(k_{\max}|\mathcal{M}_L^*|^2|\mathcal{M}_R^*|^2 W_{k_{\max}}^2)$. In the worst case, W_k grows exponentially with k, and hence using a larger value of $k_{\max}$ may significantly increase the computation cost. However, using a larger value of $k_{\max}$ allows for obtaining a larger subset of the rate region.

Furthermore, it is worth noting that as k increases, $\mathcal{R}_k$ converges to $\mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$. In other words, for sufficiently large values of k, evaluating $\mathcal{R}_k^*$ does not result in an increase in the polytope $\mathcal{R}_k$. Let k^* denote the smallest value of k such that $\mathcal{R}_k = \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$. An upper bound on k^* is the largest cycle length in $(\mathcal{M}_T, \mathcal{E}_T)$, which is not greater than $|\mathcal{M}_T|$. However, it is important to highlight that k^* can be much smaller than the largest cycle length in $(\mathcal{M}_T, \mathcal{E}_T)$. This will be demonstrated later, showing that the convergence can occur much earlier than expected based on the largest cycle length.

2) Proof of Theorem 9: The following lemma gives an incremental approach to enumerate a *superset* of $\mathcal{P}_k^*$ incrementally.

Lemma 9. *For any $k \geq 2$, $\mathcal{P}_k^*$ is a subset of*

$$\{(B_0, \ldots, B_{k-2}, B_{k-1} \wedge B'_{k-1}, B_k) : (B'_{k-1}, B_k) \in \mathcal{E}^*,$$
$$(B_0, \ldots, B_{k-1}) \in \mathcal{P}_{k-1}^*\}.$$

Proof: For any $(A_0, A_1, \ldots, A_k) \in \mathcal{P}_k^*$, there exist $(B_0, B_1, \ldots, B_{k-1}) \in \mathcal{P}_{k-1}^*$ and $(B'_{k-1}, B_k) \in \mathcal{E}^*$ such that

$$(B_0, B_1, \ldots, B_{k-1}) \succcurlyeq (A_0, A_1, \ldots, A_{k-1}),$$
$$(B'_{k-1}, B_k) \succcurlyeq (A_{k-1}, A_k).$$

We see $(B_0, \ldots, B_{k-2}, B_{k-1} \wedge B'_{k-1}, B_k)$ is a path of length k and dominates $(A_0, A_1, \ldots, A_k)$. As the latter is maximal, we further have $B_i = A_i$ for $i = 0, 1, \ldots, k-2, k$ and $B_{k-1} \wedge B'_{k-1} = A_{k-1}$. ■

Let $\mathcal{H}_1 = \mathcal{E}^*$. For $k \geq 2$, let

$$\mathcal{H}_k = \{(A_1, B_1 \wedge A_2, \ldots, B_{k-1} \wedge A_k, B_k) : (A_i, B_i) \in \mathcal{E}^*, i = 1, \ldots, k\}.$$

Lemma 10. *For $k \geq 1$, each element of $\mathcal{H}_k$ is a path in $(\mathcal{M}_T, \mathcal{E}_T)$, and $\mathcal{P}_k^* \subset \mathcal{H}_k$. Moreover, for $k \geq 1$,*

$$\mathcal{R}_k = \mathrm{dom}\ \mathrm{conv}\{R_{\mathrm{cl}(P)} : P \in \cup_{i=1}^k \mathcal{H}_i\}.$$

Algorithm 3 The pseudocode for calculating $\mathcal{R}_k^*$ for $k = 1, 2, \ldots, k_{\max}$ includes two functions: RateRegion and CALRR. The RateRegion function calculates $\mathcal{R}_k^*$ for $k = 1, 2, \ldots, k_{\max}$, while the CALRR function is called by RateRegion to calculate $\mathcal{R}_i^*$ using $\mathcal{W}_i$.

```
function RATEREGION
    Input: (M*_L, M*_R, E*), integer k_max
    Output: (R*_k, k = 2, ..., k_max)
    R*_1 ← ∅
    for each A ∈ M*_L and B ∈ M*_R s.t. (A, B) ∈ E* do
        R*_1 ← MAXADD(R*_1, (1/T)(B ∧ A)1)
    W_2 ← W2AB(M*_L, M*_R, E*)
    R*_2 ← (1/2T) CALRR(W_2, M*_L, M*_R)
    for k from 3 to k_max do
        W_k ← WAB(W_{k-1}, M*_L, M*_R, E*)
        R*_k ← (1/Tk) CALRR(W_k, M*_L, M*_R)
    return (R*_k, k = 1, 2, ..., k_max)
function CALRR
    Input: W_k, M*_L, M*_R
    Output: R̃_k
    R̃_k ← ∅
    for each A ∈ M*_L and B ∈ M*_R do
        for each r ∈ W_k(A, B)  do
            R̃_k ← MAXADD(R̃_k, r + (B ∧ A)1)
    return R̃_k
```

Proof: If $(A_0, A_1, \ldots, A_k) \in \mathcal{H}_k$, then $(A_{i-1}, A_i) \in \mathcal{E}_T$, $i = 1, \ldots, k$. Hence, each element of $\mathcal{H}_k$ is a path in $(\mathcal{M}_T, \mathcal{E}_T)$. The lemma holds directly when $k = 1$. Now we consider $k \geq 2$. By Lemma 9, $\mathcal{P}_2^* \subset \mathcal{H}_2$. As

$$\begin{aligned}\mathcal{H}_k = \{&(B_0, \ldots, B_{k-2}, B_{k-1} \wedge B'_{k-1}, B_k) :\\ &(B'_{k-1}, B_k) \in \mathcal{E}^*, (B_0, \ldots, B_{k-1}) \in \mathcal{H}_{k-1}\},\end{aligned}$$

by induction, $\mathcal{P}_k^* \subset \mathcal{H}_k$. By Theorem 8,

$$\begin{aligned}\mathcal{R}_k &= \text{dom conv}\{R_{\text{cl}(P)} : P \in \cup_{i=1}^k \mathcal{P}_i^*\}\\ &\subset \text{dom conv}\{R_{\text{cl}(P)} : P \in \cup_{i=1}^k \mathcal{H}_i\}.\end{aligned}$$

For any $P \in \cup_{i=1}^k \mathcal{H}_i$, $\text{cl}(P)$ is a cycle of length at most k, and hence $R_{\text{cl}(P)} \in \text{dom conv}\{R_C : C \in \cup_{i=1}^k \mathcal{C}_i^*\} = \mathcal{R}_k$. The proof is complete. ∎

Now we are ready to prove Theorem 9.

Proof Theorem 9: The case for $k=1$ is proved by (13). We first prove by induction that for $k \geq 2$, for any $(A_1, B_1), \ldots, (A_k, B_k) \in \mathcal{E}^*$,

$$\sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1} \in \text{dom } \mathcal{W}_k(A_1, B_k). \tag{17}$$

First, (17) holds when $k=2$ by the definition of $\mathcal{W}_2(A,B)$ in (14). For $k \geq 3$, suppose (17) holds for $k-1$. Then for any $(A_1, B_1), \ldots, (A_k, B_k) \in \mathcal{E}^*$,

$$\begin{aligned}
\sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1} &= \sum_{i=1}^{k-2} (B_i \wedge A_{i+1})\mathbf{1} + (B_{k-1} \wedge A_k)\mathbf{1} \\
&\in \text{dom } \mathcal{R}_{k-1}(A_1, B_{k-1}) + (B_{k-1} \wedge A_k)\mathbf{1} \\
&\in \text{dom } \mathcal{W}_k(A_1, B_k).
\end{aligned}$$

Let

$$\tilde{\mathcal{R}}_k = \{R_{\text{cl}(P)} : P \in \mathcal{H}_k\}.$$

By Lemma 10, $\mathcal{R}_k = \text{dom conv}(\cup_{i=1}^k \tilde{\mathcal{R}}_i)$ for $k \geq 1$. As $\mathcal{R}_1 = \tilde{\mathcal{R}}_1$, the theorem is proved if we can show that for $k \geq 2$,

$$\text{dom } \tilde{\mathcal{R}}_k = \text{dom } \mathcal{R}_k^* = \frac{1}{Tk} \text{dom} \bigcup_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} \left(\mathcal{R}_k(A,B) + (A \wedge B)\mathbf{1}\right).$$

For each $R \in \tilde{\mathcal{R}}_k$, there exists $(A_1, B_1), \ldots, (A_k, B_k) \in \mathcal{E}^*$ such that

$$TkR = (A_1 \wedge B_k)\mathbf{1} + \sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1}.$$

By (17), $\sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1} \in \text{dom}\mathcal{W}_k(A_1, B_k)$, and hence $R \in \frac{1}{Tk}\text{dom}(\mathcal{W}_k(A_1, B_k) + (A_1 \wedge B_k)\mathbf{1}) = \text{dom } \mathcal{R}_k^*$.

Fix $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$. For $R \in \mathcal{R}_k(A,B)$, there exist $A_2 \ldots, A_k \in \mathcal{M}_L^*$, $B_1, \ldots, B_{k-1} \in \mathcal{M}_R^*$ such that $(A, B_1), (A_2, B_2), \ldots, (A_{k-1}, B_{k-1}), (A_k, B) \in \mathcal{E}^*$ and $R = \sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1}$. Hence, $\frac{1}{Tk}(R + (A \wedge B)\mathbf{1}) = \frac{1}{Tk}\left(\sum_{i=1}^{k-1} (B_i \wedge A_{i+1})\mathbf{1} + (A \wedge B)\mathbf{1}\right) \in \tilde{\mathcal{R}}_k$. ∎

D. Reduced Scheduling Graph

Now, we provide an alternative representation of the dominance property. By utilizing $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$, we will derive a subgraph of $(\mathcal{M}_T, \mathcal{E}_T)$ and demonstrate that this subgraph can fulfill the same role as $(\mathcal{M}_T, \mathcal{E}_T)$ in characterizing the scheduling rate region. Let

$$\begin{aligned}
\mathcal{V} &= \{B \wedge B' : B \in \mathcal{M}_R^*, B' \in \mathcal{M}_L^*\}, \\
\mathcal{F} &= \{(B_1 \wedge A_2, B_2 \wedge A_3) : B_1 \in \mathcal{M}_R^*, (A_2, B_2) \in \mathcal{E}^*, A_3 \in \mathcal{M}_L^*\}.
\end{aligned}$$

The pair $(\mathcal{V}, \mathcal{F})$ forms a directed graph that serves as a subgraph of $(\mathcal{M}_T, \mathcal{E}_T)$.

In Table III, we evaluate the sizes of $\mathcal{M}_1, \mathcal{E}_1, \mathcal{V}, \mathcal{F}$ for the line network $\mathcal{N}_{L,2}^{\text{line}}$. The following example illustrates a network with $|\mathcal{M}_T|$ and $|\mathcal{E}_T|$ of exponential functions of the number of links, while $\mathcal{V}$ and $\mathcal{F}$ have a constant size.

TABLE III

EVALUATIONS OF SIZES OF $(\mathcal{M}_1, \mathcal{E}_1)$ AND $(\mathcal{V}, \mathcal{F})$ FOR THE LINE NETWORK WITH THE 2-HOP COLLISION MODEL $\mathcal{N}_{L,2}^{\text{LINE}}$.

L	4	5	6	7	8	9	10	11
$\lvert\mathcal{M}_1\rvert$	9	15	25	40	64	104	169	273
$\lvert\mathcal{E}_1\rvert$	49	121	304	676	1480	3481	8245	18769
$\lvert\mathcal{V}\rvert$	9	9	16	30	49	72	100	156
$\lvert\mathcal{F}\rvert$	49	49	120	324	800	1681	3074	6241

Example 10 (Single collision network)**.** Consider a network of L links with the link set $\mathcal{L} = \{l_1, \ldots, l_L\}$ and a binary collision model, where $\mathcal{I}(l_1) = \{l_2\}$ and $\mathcal{I}(l_i) = \emptyset$ for $i > 1$. The delay matrix $D_{\mathcal{L}}$ has $D_{\mathcal{L}}(l_1, l_2) = 1$. We denote this network as $\mathcal{N}_L^{\text{1-c}}$, which has the character $D^* = 1$. The scheduling graph $(\mathcal{M}_1, \mathcal{E}_1)$ of this network has $\mathcal{M}_1 = \{0,1\}^L$. For $A, B \in \mathcal{M}_1$, $(A, B) \in \mathcal{E}_1$ if either i) $A(l_1) = 0$ or ii) $A(l_1) = 1$ and $B(l_2) = 0$. Therefore, $|\mathcal{M}_1| = 2^L$ and $|\mathcal{E}_1| = 2^{2L-1} + 2^{2L-2}$, which increases exponentially with L. The reduced representation of $(\mathcal{M}_1, \mathcal{E}_1)$ has

$$\mathcal{E}^* = \{(\mathbf{1}, \mathbf{v}_2), (\mathbf{v}_1, \mathbf{1})\},$$

where $\mathbf{1}$ is the all-1 vector of L entries, $\mathbf{v}_1$ and $\mathbf{v}_2$ are obtained from $\mathbf{1}$ by setting the first and second entry to 0, respectively. Further $\mathcal{M}_L^* = \{\mathbf{v}_1, \mathbf{1}\}$ and $\mathcal{M}_R^* = \{\mathbf{v}_2, \mathbf{1}\}$. Let's calculate $(\mathcal{V}, \mathcal{F})$ for the single collision network $\mathcal{N}_L^{\text{1-c}}$. First, $\mathcal{V} = \{\mathbf{1}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_1 \wedge \mathbf{v}_2\}$. Then, $\mathcal{F} = \{(\mathbf{v}_1, \mathbf{v}_1), (\mathbf{v}_1, \mathbf{1}), (\mathbf{1}, \mathbf{v}_1 \wedge \mathbf{v}_2), (\mathbf{1}, \mathbf{v}_2), (\mathbf{v}_1 \wedge \mathbf{v}_2, \mathbf{v}_1), (\mathbf{v}_1 \wedge \mathbf{v}_2, \mathbf{1}), (\mathbf{v}_2, \mathbf{v}_1 \wedge \mathbf{v}_2), (\mathbf{v}_2, \mathbf{v}_2)\}$. We see that though the size of the scheduling graph $(\mathcal{M}_1, \mathcal{E}_1)$ of $\mathcal{N}_L^{\text{1-c}}$ is exponential in L, $(\mathcal{V}, \mathcal{F})$ has a constant size.

The graph $(\mathcal{V}, \mathcal{F})$, called the *reduced scheduling graph*, captures the essential connections and relationships from $(\mathcal{M}_T, \mathcal{E}_T)$. It is worth noting that a cycle in $(\mathcal{V}, \mathcal{F})$ is also a cycle in $(\mathcal{M}_T, \mathcal{E}_T)$. However, it is not necessarily true that a cycle in $(\mathcal{M}_T, \mathcal{E}_T)$ is dominated by a cycle in $(\mathcal{V}, \mathcal{F})$. The following theorem demonstrates the possibility of characterizing the scheduling rate region using cycles in $(\mathcal{V}, \mathcal{F})$.

Theorem 10. *For a scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$, for $k \geq 1$,*

$$\mathcal{R}_k = \text{dom conv}\{R_C : C \textit{ is a length-}i \textit{ cycle in } (\mathcal{V}, \mathcal{F}), i \leq k\}.$$

Therefore, $\mathcal{R}_k = \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$ when k is the largest cycle length in $(\mathcal{V}, \mathcal{F})$.

Proof: To simplify the notation, let

$$\mathcal{A}_k = \text{dom conv}\{R_C : C \text{ is a length-}i \text{ cycle in } (\mathcal{V}, \mathcal{F}), i \leq k\}.$$

Consider a length-k cycle $V = (V_0, V_1, \ldots, V_{k-1}, V_k = V_0)$ of $(\mathcal{V}, \mathcal{F})$. There exist $(A_i, B_i) \in \mathcal{E}^*$ such that $(A_i, B_i) \succcurlyeq (V_{i-1}, V_i)$ for $i = 1, \ldots, k$. As $P \triangleq (A_1, B_1 \wedge A_2, \ldots, B_{k-1} \wedge A_k, B_k) \succcurlyeq V$ and $P \in \mathcal{H}_k$, we have $R_V \preccurlyeq R_{\text{cl}(P)} \in \mathcal{R}_k$, where the inclusion follows from Lemma 10. Hence $\mathcal{A}_k \subset \mathcal{R}_k$.

We prove by induction that $\mathcal{R}_k \subset \mathcal{A}_k$. First, $\mathcal{R}_1 \subset \mathcal{A}_1$ as $\mathcal{E}^* \subset \mathcal{F}$. For $k \geq 2$, assume that $\mathcal{R}_{k-1} \subset \mathcal{A}_{k-1}$. For each $P \in \mathcal{H}_k$, $\mathrm{cl}(P)$ is a closed path in $(\mathcal{V}, \mathcal{F})$. If $\mathrm{cl}(P)$ is a cycle, then $R_{\mathrm{cl}(P)} \in \mathcal{A}_k$ by the definition of $\mathcal{H}_k$ and $\mathcal{F}$. If $\mathrm{cl}(P)$ is not a cycle, then it can be decomposed into multiple cycles of length strictly less than k. Then $R_{\mathrm{cl}(P)} \in \mathcal{R}_{k-1} \subset \mathcal{A}_{k-1} \subset \mathcal{A}_k$ by induction. Hence for both cases, $R_{\mathrm{cl}(P)} \in \mathcal{A}_k$. Last, by Lemma 10, $\mathcal{R}_k \subset \mathcal{A}_k$. ■

Theorem 10 shows that the largest cycle length in $(\mathcal{V}, \mathcal{F})$ is a sufficient value of k such that $\mathcal{R}_k = \mathcal{R}^{(\mathcal{M}_T, \mathcal{E}_T)}$, and this length is shorter than the largest cycle length in $(\mathcal{M}_T, \mathcal{E}_T)$. Consequently, $(\mathcal{V}, \mathcal{F})$ can be used for calculating the scheduling rate region with reduced computational complexity compared to using $(\mathcal{M}_T, \mathcal{E}_T)$. The complete rate region can be obtained by employing Johnson's algorithm [31] to enumerate cycles in $(\mathcal{V}, \mathcal{F})$. To calculate the subset of the rate region $\mathcal{R}_k$, one can enumerate cycles in $(\mathcal{V}, \mathcal{F})$ up to length k. In the following subsection, we compare these approaches for calculating $\mathcal{R}_k$ with Algorithm 3.

E. Numerical Evaluation

We compare the different approaches for calculating the scheduling rate region by numerical evaluations on the networks $\mathcal{N}_{L,2}^{\text{line}}$ for $L = 4$ to 11. Since each network has $D^* = 1$, the only scheduling graph is $(\mathcal{M}_1, \mathcal{E}_1)$. There are two main scenarios in the numerical evaluation:

1) Calculating the entire scheduling rate region by enumerating cycles:
    - SR-1: Enumerate cycles in in original scheduling graph $(\mathcal{M}_1, \mathcal{E}_1)$.
    - SR-2: Enumerate cycles in the reduced scheduling graph $(\mathcal{V}, \mathcal{F})$.
2) Calculating a subset of the rate region $\mathcal{R}_k$:
    - Rk-1: Enumerate cycles up to length k in the original scheduling graph $(\mathcal{M}_1, \mathcal{E}_1)$.
    - Rk-2: Enumerate cycles up to length k in the reduced scheduling graph $(\mathcal{V}, \mathcal{F})$.
    - Rk-3: Use Algorithm 3 to calculate $\mathcal{R}_k$.

All the approaches use $\mathcal{N}_{L,2}^{\text{line}}$ as the input. The operations of these approaches are summarized as follows:

- SR-1:
    1) Evaluate $\mathcal{M}_2^*$ using the Bron–Kerbosch algorithm with vertex pivoting [42]–[45].
    2) Generate $(\mathcal{M}_1, \mathcal{E}_1)$ using Algorithm 1.
    3) Enumerate cycles of $(\mathcal{M}_1, \mathcal{E}_1)$ using Johnson's algorithm [31].
- SR-2:
    1) Evaluate $\mathcal{M}_2^*$ using the Bron–Kerbosch algorithm with vertex pivoting.
    2) Generate $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ from $\mathcal{M}_2^*$.
    3) Generate $(\mathcal{V}, \mathcal{F})$ from $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$.
    4) Enumerate cycles of $(\mathcal{V}, \mathcal{F})$ using Johnson's algorithm.
- Rk-1:
    1) Evaluate $\mathcal{M}_2^*$ using the Bron–Kerbosch algorithm with vertex pivoting.
    2) Generate $(\mathcal{M}_1, \mathcal{E}_1)$ using Algorithm 1.

3) Enumerate cycles of $(\mathcal{M}_1, \mathcal{E}_1)$ up to length 4.

- Rk-2:
  1) Evaluate $\mathcal{M}_2^*$ using the Bron–Kerbosch algorithm with vertex pivoting.
  2) Generate $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ from $\mathcal{M}_2^*$.
  3) Generate $(\mathcal{V}, \mathcal{F})$ from $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$.
  4) Enumerate cycles of $(\mathcal{V}, \mathcal{F})$ up to length 4.
- Rk-3:
  1) Evaluate $\mathcal{M}_2^*$ using the Bron–Kerbosch algorithm with vertex pivoting.
  2) Generate $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ from $\mathcal{M}_2^*$.
  3) Execute RateRegion in Algorithm 3 with $k_{\max} = 4$.

In our evaluation, we implemented all these approaches using the Julia programming language. Johnson's algorithm and the algorithm for enumerating cycles up to a certain length were obtained from the Julia Graphs package [52]. To measure the execution time accurately, we utilized the Julia BenchmarkTools package [53], which runs a function multiple times to obtain a more stable estimate of the running time. Table IV presents a comparison of the execution times for these approaches. Based on our evaluation, we have made the following observations:

When calculating the entire rate region by enumerating all the cycles, SR-2 proves to be more efficient than SR-1. However, the computational costs of both approaches increase rapidly as the network length L increases. As a result, when L reaches 6 for SR-1 and 7 for SR-2, the memory requirements become substantial, causing the program to crash on our computer. As a result, we were unable to obtain the running time for larger networks using these two approaches. This limitation highlights the challenge of enumerating all the cycles in larger networks, as the computational and memory requirements become increasingly demanding.

All three approaches, Rk-1, Rk-2, and Rk-3, are capable of calculating $\mathcal{R}_4$ for networks with $L = 11$. Among these approaches, Rk-2 is more efficient than Rk-1 for all the considered networks. While Rk-2 outperforms Rk-3 for smaller networks, the advantage of Rk-3 becomes evident as the network size increases. It is worth noting that when compared to Johnson's algorithm for enumerating all cycles, enumerating cycles up to a certain length tends to be slower when the length is relatively large [41]. This is the case for $\mathcal{N}_{4,2}^{\text{line}}$, where SR-1 is faster than Rk-1.

In conclusion, we would like to remark that $\mathcal{R}_4$ serves as the rate region, although we did not provide a formal proof in this paper. If evaluated, it would become apparent that $\mathcal{R}_4 = \mathcal{R}_5 = \mathcal{R}_6 = \cdots$. However, relying solely on evaluating up to $\mathcal{R}_{|\mathcal{V}|}$ is not a feasible method to prove the rate region, as it would be impractical for larger networks. In another work, we have demonstrated that $\mathcal{R}_4$ aligns with an upper bound on the rate region of $\mathcal{N}_{L,2}^{\text{line}}$, obtained by leveraging a graphical property of the periodic graph associated with the network.

VI. Algorithms for Maximizing a Linear Function of Rate Vectors

In this section, our focus is on maximizing a linear function of rate vectors. This problem arises in network utility maximization, where the goal is to find a scheduling rate vector that maximizes a weighted sum (see, e.g., [21], [22]). For instance, one common objective is to maximize the sum rate of all the links in the network. While it is

TABLE IV

COMPARISON OF THREE METHODS FOR CALCULATING THE RATE REGION OF $\mathcal{N}_{L,2}^{\text{LINE}}$. ALL THE METHODS ARE IMPLEMENTED IN JULIA, AND EXECUTED ON A COMPUTER WITH A 2.3 GHZ QUAD-CORE CPU, 8 GB MEMORY AND JULIA 1.9.2.

Approach	$\mathcal{N}_{4,2}^{\text{line}}$	$\mathcal{N}_{5,2}^{\text{line}}$	$\mathcal{N}_{6,2}^{\text{line}}$	$\mathcal{N}_{7,2}^{\text{line}}$	$\mathcal{N}_{8,2}^{\text{line}}$	$\mathcal{N}_{9,2}^{\text{line}}$	$\mathcal{N}_{10,2}^{\text{line}}$	$\mathcal{N}_{11,2}^{\text{line}}$
SR-1	0.928 ms	5.538 s	-	-	-	-	-	-
SR-2	0.765 ms	0.765 ms	15.06 s	-	-	-	-	-
Rk-1	0.981 ms	4.321 ms	21.38 ms	131.5 ms	794.1 ms	3.518 s	13.16 s	49.952 s
Rk-2	0.685 ms	0.857 ms	2.850 ms	15.93 ms	127.6 ms	466.7 ms	2.939 s	8.462 s
Rk-3	1.242 ms	1.388 ms	5.976 ms	22.79 ms	101.6 ms	246.8 ms	1.002 s	3.204 s

technically possible to solve the maximization problem given the rate region, as we discussed earlier, evaluating the rate region can be challenging. To address this issue, we propose an algorithm that maximizes a linear function without explicitly calculating the rate region. This approach allows us to find an optimal scheduling rate vector without relying on the explicit determination of the rate region.

Previous works have considered this optimization problem for scheduling with delays [5], [6]. However, these works provide only approximate solutions to the optimization problem. They treat the step-1 scheduling graph $(\mathcal{M}_T, \mathcal{E}_{T,1})$ as a state transition graph and employ a dynamic programming approach similar to the Viterbi algorithm to optimize the state sequence. Their objective is to find an optimal path (not cycle) of a given length k in the step-1 scheduling graph. In other words, they find the optimal rate vector in $\widetilde{\mathcal{R}}^{\mathcal{N}^{T+k}}$, which is not necessarily equal to $\mathcal{R}^{\mathcal{N}}$ even when $k = |\mathcal{M}_T|$. However, as k tends to infinity, $\widetilde{\mathcal{R}}^{\mathcal{N}^{T+k}}$ converges to $\mathcal{R}^{\mathcal{N}}$ (as discussed in Sec. III-C).

In this section, we propose an approach to accurately compute the optimal value of a linear function on the rate vectors using the step-T scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$ and the dominance property in Sec. V-A. Our algorithm identifies an optimal cycle of a given length k in $(\mathcal{M}_T, \mathcal{E}_T)$ and has a computation cost that is linear in k.

A. *Problem Formulation and Simplification*

Consider a network $\mathcal{N} = (\mathcal{L}, \mathcal{I}, D_{\mathcal{L}})$, and a linear function $f : \mathbb{R}^{|\mathcal{L}|} \to \mathbb{R}$. In (12), we defined $\mathcal{R}_k$, which is the subset of the rate region generated by the cycles up to length k in the step-T scheduling graph $(\mathcal{M}_T, \mathcal{E}_T)$. In this context, we study how to calculate the optimal value $\max_{R\in\mathcal{R}_k} f(R)$ and identify a periodic schedule that achieves this optimal value.

As f is linear, we can express it as $f(R) = \sum_{l\in\mathcal{L}} c_l R(l)$, where c_l represents fixed linear combination coefficients. For our analysis, we will specifically focus on maximizing a linear function f in which all the coefficients c_l are positive. For linear functions with a mixture of positive and negative coefficients, the problem can be transformed into an equivalent problem with only positive coefficients, as discussed below.

Consider a linear function f with a coefficient $c_{l_0} \le 0$. If $R^* \in \mathcal{R}_k$ maximizes $f(R)$, then the rate vector obtained by setting $R^*(l_0)$ to 0 (which is also in $\mathcal{R}_k$ due to Lemma 8) is also optimal. This is because inactivating link l_0 generates no collision with other links. Therefore, we can reduce the problem by removing link l_0 as follows: First let $(\mathcal{L}', \mathcal{I}')$ be the directed graph or hypergraph obtained by removing l_0 from $(\mathcal{L}, \mathcal{I})$. In other words, $\mathcal{L}' = \mathcal{L} \setminus \{l_0\}$

and for $l \neq l_0$, $\mathcal{I}'(l)$ is obtained by excluding all $\theta \in \mathcal{I}(l)$ with $l_0 \in \theta$. Second, let $D_{\mathcal{L}'}$ be the submatrix of $D_{\mathcal{L}}$ obtained by removing the row and the column indexed by l_0. Last, define $f'(R) = \sum_{l \in \mathcal{L}'} c_l R(l)$. The new optimization problem is to maximize f' for the network $\mathcal{N}' \triangleq (\mathcal{L}', \mathcal{I}', D_{\mathcal{L}'})$. By repeating this procedure, we can continue removing links with negative coefficients until f' consists only of positive coefficients.

B. An Incremental Approach for Optimizing a Linear Function

Theorem 9 provides an approach for maximizing a linear function $f : \mathbb{R}^{|\mathcal{L}|} \to \mathbb{R}$ with positive coefficients. Specifically, since $f(R_1) \geq f(R_2)$ for any $R_1 \succcurlyeq R_2 \in \mathbb{R}^{|\mathcal{L}|}$, we know that $\max_{R \in \mathcal{R}_k} f(R) = \max_{i=1,\ldots,k} \max_{R \in \mathcal{R}_k^*} f(R)$. However, the complexity of this algorithm, however, can be exponential in k due to the size of $\mathcal{W}_k(A, B)$. If we are only interested in the optimal value of f, we can simplify the algorithm by replacing the set $\mathcal{W}_k(A, B)$ with a real value.

Define

$$U_1^* = \max_{R \in \mathcal{R}_1} f(R) = \max_{P \in \mathcal{E}^*} f(R_{\mathrm{cl}(P)}),$$

where the second equality follows from (13). For $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$, define

$$U_2(A, B) = \max\{f((B_1 \wedge A_2)\mathbf{1}) : (A, B_1), (A_2, B) \in \mathcal{E}^*\}, \tag{18}$$

and for $k \geq 3$, define

$$U_k(A, B) = \max\{U_{k-1}(A, B') + f((B' \wedge A')\mathbf{1}) : (A', B) \in \mathcal{E}^*, B' \in \mathcal{M}_R^*\}. \tag{19}$$

The next theorem gives an incremental algorithm for optimization a linear function over $\mathcal{R}_k$.

Theorem 11. *Let $f : \mathbb{R}^{|\mathcal{L}|} \to \mathbb{R}$ be a linear function with positive linear combination coefficients. For $k \geq 1$, $\max_{R \in \mathcal{R}_k} f(R) = \max\{U_1^*, U_2^*, \ldots, U_k^*\}$, where for $i \geq 2$,*

$$U_i^* = \frac{1}{iT} \max_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} U_i(A, B) + f((A \wedge B)\mathbf{1}). \tag{20}$$

Proof: As the linear combination coefficients in U are non-negative, by Theorem 9,

$$\max_{R \in \mathcal{R}_k} f(R) = \max\left\{\max_{R \in \mathcal{R}_1^*} f(R) = U_1^*, \max_{R \in \mathcal{R}_i^*} f(R), i = 2, \ldots, k\right\},$$

where for $i \geq 2$,

$$\begin{aligned}
\max_{R \in \mathcal{R}_i^*} f(R) &= \frac{1}{iT} \max_{R \in \bigcup_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} (\mathcal{R}_i(A,B) + (A \wedge B)\mathbf{1})} f(R) \\
&= \frac{1}{iT} \max_{A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*} \max_{R \in W_i(A,B)} f(R) + f((A \wedge B)\mathbf{1}).
\end{aligned}$$

We show that $\max_{R \in \mathcal{W}_i(A,B)} f(R) = U_i(A, B)$ for $i \geq 2$ by induction. First, by the definition in (14) and (18), $U_2(A, B) = \max_{R \in \mathcal{W}_2(A,B)} f(R)$. For $i > 2$, assume that $\max_{R \in \mathcal{R}_{i-1}(A,B)} f(R) = U_{i-1}(A, B)$. By the definition in (15),

$$\begin{aligned}
\max_{R \in \mathcal{W}_i(A,B)} f(R) &= \max_{B' \in \mathcal{M}_R^*} \max_{A' : (A', B) \in \mathcal{E}^*} \max_{R \in \mathcal{R}_{i-1}(A, B')} f(R) + f((B' \wedge A')\mathbf{1}) \\
&= \max_{B' \in \mathcal{M}_R^*} \max_{A' : (A', B) \in \mathcal{E}^*} U_{k-1}(A, B') + f((B' \wedge A')\mathbf{1}) \\
&= U_k(A, B).
\end{aligned}$$

The proof is completed by $\max_{R\in\mathcal{R}_i^*} f(R) = U_i^*$. ■

In the subsequent subsections, algorithms are presented for computing the optimal value of the linear function $f(R)$ as well as identifying a cycle that attains this optimal value.

C. Algorithm for Optimal Value

Algorithm 4 provides the pseudocode for calculating $\mathcal{U}_k$, and Algorithm 5 provides the pseudocode for determining the optimal value of $f(R)$ in $\mathcal{R}_k$. The structure of Algorithm 4 and Algorithm 5 remains similar to that of Algorithm 2 and Algorithm 3, respectively. The main difference lies in the computation of $U_k(A,B)$ instead of $\mathcal{W}_k(A,B)$.

Algorithm 4 and Algorithm 5 assume that $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ has already been calculated (see Sec.V-A). These two algorithms are described below. To simplify the notation, we denote

$$U_k = (U_k(A,B), A \in \mathcal{M}_L^*, B \in \mathcal{M}_R^*).$$

In Algorithm 4, two functions are provided: U2AB and UAB. The U2AB function calculates U_2 using (18), while the UAB function calculates U_k from U_{k-1} using (19). The computation cost of both U2AB and UAB is $O(|\mathcal{M}_L^*|^2|\mathcal{M}_R^*|^2|\mathcal{L}|T)$, accounting for the integer and logical operations involved in the calculation.

In Algorithm 5, two functions are provided: OptimalRate and UStar. The OptimalRate function takes an integer $k_{\max}$ as the input, and calculates U_k^* for $k = 1, 2, \ldots, k_{\max}$ as the output. The OptimalRate function calls U2AB and UAB to obtain U_k for $k = 2, \ldots, k_{\max}$ and then calculates U_k^* by calling UStar on U_k, applying the formula in Theorem 11. The computation cost of UStar for U_k^* is $O(|\mathcal{M}_L^*||\mathcal{M}_R^*||\mathcal{L}|T)$ floating point operations. The overall computation cost of OptimalRate is $O(k_{\max}|\mathcal{M}_L^*|^2|\mathcal{M}_R^*|^2|\mathcal{L}|T)$. To get the optimal value over the entire rate region, it is sufficient to use $k_{\max} = |\mathcal{M}_L^*||\mathcal{M}_R^*|$, so that the computational complexity is $O(|\mathcal{M}_L^*|^3|\mathcal{M}_R^*|^3|\mathcal{L}|T)$.

D. Algorithm for Optimizer

In addition to determining the optimal value U_k^*, it is also essential to identify the optimizer, which refers to the cycle that achieves the optimal value. Such a cycle can be utilized to construct an optimal periodic schedule.

Assume that U_k^* and the U_k are already calculated for $k = 1, \ldots, k_{\max}$, which can be done by Algorithm 4 and Algorithm 5. First, the optimizer of U_1^* can be find by enumerating all elements P in $\mathcal{E}^*$ until we find P such that $f(R_{\mathrm{cl}(P)}) = U_1^*$.

The case with $k \geq 2$ can be solved using backward searching. Enumerate $A_1 \in \mathcal{M}_L^*$ and $B_k \in \mathcal{M}_R^*$ until we find A_1 and B_k such that

$$U_k^* = U_k(A_1, B_k) + f((A_1 \wedge B_k)\mathbf{1}).$$

The existence of such A_1 and B_k is guaranteed by the definition of U_k^* (see (20)). Then for $i = k, k-1, \ldots, 3$, enumerate $A_i \in \mathcal{M}_L^*$ and $B_{i-1} \in \mathcal{M}_R^*$ until we find A_i and B_{i-1} such that $(A_i, B_i) \in \mathcal{E}^*$ and

$$U_i(A_1, B_i) = U_{i-1}(A_1, B_{i-1}) + f((B_{i-1} \wedge A_i)\mathbf{1}).$$

Algorithm 4 The pseudocode for calculating U_k consists of two functions: U2AB and UAB. The function U2AB is responsible for computing U_2, while the function UAB is used to calculate U_k from U_{k-1}, and this process is applicable for any value of $k \geq 2$.

function U2AB
    Input: $\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*$
    Output: U_2
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
        $U_2(A, B) \leftarrow 0$
        for each B_1 s.t. $(A, B_1) \in \mathcal{E}^*$ and A_2 s.t. $(A_2, B) \in \mathcal{E}^*$ **do**
            $R = f((B_1 \wedge A_2)\mathbf{1})$
            if $R > U_2(A, B)$ **then**
                $U_2(A, B) \leftarrow R$
    return U_2
function UAB
    Input: $U_{k-1}, \mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*$
    Output: U_k
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
        $U_k(A, B) \leftarrow 0$
        for each $B' \in \mathcal{M}_R^*$ and $A' \in \mathcal{M}_L^*$ s.t. $(A', B) \in \mathcal{E}^*$ **do**
            $R = U_{k-1}(A, B') + f((B' \wedge A')\mathbf{1})$
            if $R > U_k(A, B)$ **then**
                $U_k(A, B) \leftarrow R$
    return U_k

The existence of such A_i and B_{i-1} is guaranteed by the definition of $U_i(A, B)$ (see (19)). Last, enumerate $B_1 \in \mathcal{M}_R^*$ and $A_2 \in \mathcal{M}_L^*$ such that $(A_1, B_1), (A_2, B_2) \in \mathcal{E}^*$ and

$$U_2(A_1, B_2) = f((B_1 \wedge A_2)\mathbf{1}).$$

The existence of such A_2 and B_1 is guaranteed by the definition of $U_2(A, B)$ (see (18)).

According to the above construction, $(A_1, B_1), \ldots, (A_k, B_k) \in \mathcal{E}^*$ and

$$U_k^* = f((A_1 \wedge B_k)\mathbf{1}) + \sum_{i=3}^{k} f((B_{i-1} \wedge A_i)\mathbf{1}) + f((B_1 \wedge A_2)\mathbf{1}).$$

Therefore, $(B_1 \wedge A_2, B_2 \wedge A_3, \ldots, B_{k-1} \wedge A_k, B_k \wedge A_1, B_1 \wedge A_2)$ is a k-cycle in $(\mathcal{M}_T, \mathcal{E}_T)$ that achieves the optimal value U_k^*.

We give the pseudocode of this backward searching algorithm in Algorithm 6, where a function called Optimal-Cycle is provided. The whole procedure is as follows:

Algorithm 5 The pseudocode for calculating U_k^* for $k = 1, 2, \ldots, k_{\max}$ consists of two functions: OptimalRate and UStar. The OptimalRate function is responsible for determining the optimal rate $f(R)$, while the UStar function is called by OptimalRate to calculate U_i^* from U_i.

function OPTIMALRATE
    Input: f, $(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$ and integer $k_{\max}$
    Output: $\max_{R \in \mathcal{R}_k} f(R), k = 2, \ldots, k_{\max}$
    $U_1^* \leftarrow 0$
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ s.t. $(A, B) \in \mathcal{E}^*$ **do**
        $R \leftarrow \frac{1}{T}(B \wedge A)\mathbf{1}$
        if $R > U_1^*$ **then**
            $U_1^* \leftarrow R$
    $U_2 \leftarrow$ U2AB$(\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*)$
    $U_2^* \leftarrow$ UStar$(U_2, \mathcal{M}_L^*, \mathcal{M}_R^*, 2)$
    for k from 3 to $k_{\max}$ **do**
        $U_k \leftarrow$ UAB$(U_{k-1}, (\mathcal{M}_L^*, \mathcal{M}_R^*, \mathcal{E}^*))$
        $U_k^* \leftarrow$ UStar$(U_k, \mathcal{M}_L^*, \mathcal{M}_R^*, k)$
    return $U_k^*, k = 2, \ldots, k_{\max}$
function USTAR
    Input: U_k, $\mathcal{M}_L^*$, $\mathcal{M}_R^*$, k
    Output: $\max_{R \in \mathcal{R}_k} f(R)$
    $U_k^* \leftarrow 0$
    for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
        $R = \frac{1}{kT}(U_k(A, B) + f((A \wedge B)\mathbf{1}))$
        if $R > U_k^*$ **then**
            $U_k^* \leftarrow R$
    return U_k^*

1) Calculate U_k^* and the U_k for $k = 1, \ldots, k_{\max}$ by Algorithm 4 and Algorithm 5.
2) Determine k^* such that $U_{k^*}^* = \max_{k=1,\ldots,k_{\max}} U_k^*$.
3) Execute the OptimalCycle function to identify an optimizer for $U_{k^*}^*$.

The extra computation cost of OptimalCycle is $O(k^* |\mathcal{M}_L^*||\mathcal{M}_R^*||\mathcal{L}|T)$.

VII. CONCLUDING REMARKS

This work introduces a graphical framework for wireless network scheduling with discrete signal propagation delays. It extends the existing independent set-based scheduling framework, commonly used in traditional scheduling with guard intervals to prevent collisions. To gain a better understanding of the advantages and the feasibility of scheduling with delays in the real world, several further research directions can be explored:

Algorithm 6 The pseudocode for identifying a cycle that achieves U_k^* where $k \geq 2$. The values of U_i, $i = 2, \ldots, k$ are known. The output is a cycle C such that $U_k^* = f(R_C)$.

function OPTIMALCYCLE
  Input: U_k^*, k
  Output: $C = (C_1, \ldots, C_k, C_1)$
  for each $A \in \mathcal{M}_L^*$ and $B \in \mathcal{M}_R^*$ **do**
    if $U_k(A, B) + f(A \wedge B)\mathbf{1}) = U_k^*$ **then**
      $A_1 \leftarrow A$, $B_k \leftarrow B$
      $C_k \leftarrow B_k \wedge A_1$
      break
  for i from k down to 3 **do**
    for each A s.t. $(A, B_i) \in \mathcal{E}^*$ and $B \in \mathcal{M}_R^*$ **do**
      if $U_{i-1}(A_1, B) + f(B \wedge A)\mathbf{1}) = U_i(A_1, B_i)$ **then**
        $A_i \leftarrow A$, $B_{i-1} \leftarrow B$
        $C_{i-1} \leftarrow B_{i-1} \wedge A_i$
        break
  for each A and B s.t. $(A, B_2), (A_1, B) \in \mathcal{E}^*$ **do**
    if $f(A \wedge B)\mathbf{1}) = U_2(A_1, B_2)$ **then**
      $A_2 \leftarrow A$, $B_1 \leftarrow B$
      $C_1 \leftarrow B_1 \wedge A_2$
      break
  return $(C_1, \ldots, C_k, C_1)$

1) Outer bounds on the scheduling rate region: Due to the high computational cost involved in calculating the complete scheduling rate region, it may only be feasible to compute a subset of it in practical cases. Evaluating the quality of the computed subset can be facilitated by establishing an outer bound on the rate region.
2) Practical scheduling approaches: The algorithms proposed in this paper make ideal assumptions, such as assuming synchronization of all network nodes to a common clock and having complete and accurate delay and collision information. Further research is needed to relax these assumptions and develop practical scheduling approaches that can handle real-world network scenarios.
3) Network flow control: The presented framework opens up possibilities for systematically studying end-to-end communication flows in wireless networks with delays. Scheduling with delays should be jointly optimized with routing and congestion control mechanisms to ensure efficient and reliable network operation.
4) Real-world demonstrations: Conducting practical experiments and demonstrations can help assess the performance and practical advantages of such scheduling techniques in real-world wireless network environments. This includes evaluating the impact on throughput, latency, energy efficiency, and overall system performance.

Appendix
Physical Network Model

We introduce a physical model of wireless networks in [30], and discuss how to apply the results for our network model to the physical model. We denote this physical network model as $\mathcal{N}^{\text{phy}}$, which has the N nodes that share the same communication channel of bandwidth W. Denote by P_i the transmitting power of node i, and by h_{ij} the channel gain from the node i to the node j, where $1 \leq i \neq j \leq N$. So when the node i transmits, the node j can receive the signal power $h_{ij}P_i$. Denote by R_{ij}^{code} the coding rate from the node i to the node j, where $1 \leq i \neq j \leq N$. Here we assume h_{ij}, R_{ij}^{code} and P_i do not change over time.

Based on the physical model $\mathcal{N}^{\text{phy}}$, we can derive a network model $\mathcal{N}$ in Sec. II-A. The link set $\mathcal{L} = \{l_{ij} \triangleq (i,j) : R_{ij}^{\text{code}} > 0\}$. For any $l_{ij} \in \mathcal{L}$ and $\theta \subset \mathcal{L}$, we say θ is in the collision set $\mathcal{I}(l_{ij})$ if

$$\frac{1}{2}\log\left(1 + \text{SINR}\right) \leq R_{ij}^{\text{code}},$$

where the signal-to-interference-and-noise ratio $\text{SINR} = \frac{h_{ij}P_i}{\sum_{l \in \theta} h_{s_l j} P_{s_l} + N_0 W}$, and N_0 is the power spectral density of the white noise process. $\mathcal{N}^{\text{phy}}$ and $\mathcal{N}$ share the same delay matrix.

If a collision-free schedule S of $\mathcal{N}$ is applied to $\mathcal{N}^{\text{phy}}$, for each link $(i,j) \in \mathcal{L}$, rate R_{ij}^{code} can be achieved for any active timeslot. Hence, we obtain an achievable rate vector $(R_{ij}, 1 \leq i \neq j \leq N)$ for $\mathcal{N}^{\text{phy}}$, where

$$R_{ij} = \begin{cases} R_{ij}^{\text{code}} R_S^{\mathcal{N}}(l_{ij}), & l_{ij} \in \mathcal{L}, \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, the rate region of $\mathcal{N}$ induces an achievable rate region of $\mathcal{N}^{\text{phy}}$.

References

[1] J. Ma, Y. Liu, and S. Yang, "Rate region of scheduling a wireless network with discrete propagation delays," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[2] J. G. Proakis, "Intersymbol interference in digital communication systems," *Wiley Encyclopedia of Telecommunications*, 2003.

[3] C.-C. Hsu, K.-F. Lai, C.-F. Chou, and K.-J. Lin, "ST-MAC: Spatial-temporal MAC scheduling for underwater sensor networks," in *Proc. IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, Apr 2009, pp. 1827–1835.

[4] Y. Guan, C.-C. Shen, and J. Yackoski, "MAC scheduling for high throughput underwater acoustic networks," in *Proc. IEEE WCNC 2011*, Cancún, Mexico, Mar 2011, pp. 197–202.

[5] L. H. Grokop, D. N. C. Tse, and R. D. Yates, "Interference alignment for line-of-sight channels," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 5820–5839, 2011.

[6] M. Chitre, M. Motani, and S. Shahabudeen, "Throughput of networks with large propagation delays," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 645–658, Jul 2012.

[7] P. Anjangi and M. Chitre, "Unslotted transmission schedules for practical underwater acoustic multihop grid networks with large propagation delays," in *Proc. IEEE UComms 2016*, Lerici, Italy, Aug 2016, pp. 1–5.

[8] W. Bai, M. Motani, and H. Wang, "On the throughput of linear unicast underwater networks," in *Proc. IEEE GLOBECOM 2017*, Singapore, Dec 2017, pp. 1–6.

[9] J. Ma and S. Yang, "A hybrid physical-layer network coding approach for bidirectional underwater acoustic networks," in *Proc. MTS/IEEE OCEANS'19*, Marseille, France, Jun 2019, pp. 1–8.

[10] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 910–917, Sept 1988.

[11] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Trans. Commun*, vol. 38, no. 4, pp. 456–460, April 1990.

[12] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.

[13] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MobiCom'06*, Sep 2006, pp. 227–238.

[14] S. N. Le, Y. Zhu, Z. Peng, J.-H. Cui, and Z. Jiang, "PMAC: a real-world case study of underwater MAC," in *Proc. ACM WUWNet'13*, Kaohsiung, Taiwan, Nov 2013, pp. 1–8.

[15] Y. Fan, Y. Liu, and S. Yang, "Continuity of link scheduling rate region for wireless networks with propagation delays," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022.

[16] H. Zeng, Y. T. Hou, Y. Shi, W. Lou, S. Kompella, and S. F. Midkiff, "SHARK-IA: An interference alignment algorithm for multi-hop underwater acoustic networks with large propagation delays," in *Proceedings of the international conference on underwater networks & systems*, 2014, pp. 1–8.

[17] N. Zhao, N. Yao, and Z. Gao, "A message transmission scheduling algorithm based on time-domain interference alignment in UWANs," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1058–1070, 2021.

[18] L. Wan, H. Zhou, X. Xu, Y. Huang, S. Zhou, Z. Shi, and J.-H. Cui, "Adaptive modulation and coding for underwater acoustic ofdm," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 327–336, 2015.

[19] J. F. O'Hara, S. Ekin, W. Choi, and I. Song, "A perspective on terahertz next-generation wireless communications," *Technologies*, vol. 7, no. 2, p. 43, 2019.

[20] E. Arikan, "Some complexity results about packet radio networks (corresp.)," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 681–685, 1984.

[21] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 66–80.

[22] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Jul 2006.

[23] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE transactions on mobile computing*, vol. 6, no. 6, pp. 595–605, 2007.

[24] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," *IEEE Transactions on Information theory*, vol. 54, no. 2, pp. 572–594, 2008.

[25] R. J. McEliece and K. N. Sivarajan, "Performance limits for channelized cellular telephone systems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 21–34, 1994.

[26] S. Sarkar and K. N. Sivarajan, "Hypergraph models for cellular mobile communication systems," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 2, pp. 460–471, 1998.

[27] Q. Li and R. Negi, "Maximal scheduling in wireless ad hoc networks with hypergraph interference models," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 297–310, 2012.

[28] H. Zhang, L. Song, and Z. Han, "Radio resource allocation for device-to-device underlay communication using hypergraph theory," *IEEE Transactions on Wireless Communications*, vol. 15, no. 7, pp. 4852–4861, 2016.

[29] A. Ganesan, "On some distributed scheduling algorithms for wireless networks with hypergraph interference models," *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2952–2957, 2021.

[30] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[31] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM Journal on Computing*, vol. 4, no. 1, pp. 77–84, Mar 1975.

[32] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Applications in vlsi domain," *Proc. ACM/IEEE Design Automation Conf*, vol. 7, no. 1, pp. 526–529, 1997.

[33] N. Alon, U. Arad, and Y. Azar, "Independent sets in hypergraphs with applications to routing via fixed paths," in *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 1999, pp. 16–27.

[34] S. Harrod, "Modeling network transition constraints with hypergraphs," *Transportation Science*, vol. 45, no. 1, pp. 81–97, 2011.

[35] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen, "Directed hypergraphs and applications," *Discrete applied mathematics*, vol. 42, no. 2-3, pp. 177–201, 1993.

[36] J. B. Orlin, "Some problems on dynamic/periodic graphs," in *Progress in Combinatorial Optimization*. W. R. Pulleyblank, Ed. Orlando, FL: Academic Press, Jan 1984, pp. 273–293.

[37] I. L. Beckenbach, "Matchings and flows in hypergraphs," Ph.D. dissertation, Freie Universität Berlin, 2019.

[38] J. W. Moon and L. Moser, “On cliques in graphs,” *Israel journal of Mathematics*, vol. 3, no. 1, pp. 23–28, 1965.

[39] P. Gleiss, J. Leydold, and P. Stadler, “Circuit bases of strongly connected digraphs,” *Discussiones Mathematicae Graph Theory*, vol. 2, no. 23, pp. 241–260, Oct 2003.

[40] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on cross-layer congestion control in wireless networks,” *IEEE/ACM transactions on networking*, vol. 14, no. 2, pp. 302–315, 2006.

[41] H. Liu and J. Wang, “A new way to enumerate cycles in graph,” in *AICT-ICIW'06*. IEEE, 2006, pp. 57–57.

[42] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph,” *Commun. ACM*, vol. 16, no. 9, pp. 575–577, Sept 1973.

[43] E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” *Theoretical computer science*, vol. 363, no. 1, pp. 28–42, 2006.

[44] F. Cazals and C. Karande, “A note on the problem of reporting maximal cliques,” *Theoretical computer science*, vol. 407, no. 1-3, pp. 564–568, 2008.

[45] D. Eppstein, M. Löffler, and D. Strash, “Listing all maximal cliques in large sparse real-world graphs,” *Journal of Experimental Algorithmics (JEA)*, vol. 18, pp. 1–21, 2013.

[46] P. Beame and M. Luby, “Parallel search for maximal independence given minimal dependence,” in *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, 1990, pp. 212–218.

[47] E. Dahlhaus, M. Karpinski, and P. Kelsen, “An efficient parallel algorithm for computing a maximal independent set in a hypergraph of dimension 3,” *Information Processing Letters*, vol. 42, no. 6, pp. 309–313, 1992.

[48] I. O. Bercea, N. Goyal, D. G. Harris, and A. Srinivasan, “On computing maximal independent sets of hypergraphs in parallel,” in *Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures*, 2014, pp. 42–50.

[49] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 1998, vol. 184.

[50] A. Schrijver *et al.*, *Combinatorial optimization: polyhedra and efficiency*. Springer, 2003, vol. 24, no. 2.

[51] B. Grünbaum, *Convex polytopes*, ser. Graduate Texts in Mathematics, V. Kaibel, V. Klee, and G. M. Ziegler, Eds. Springer, 2003, vol. 221.

[52] “JuliaGraphs/Graphs.jl: an optimized graphs package for the Julia programming language,” 2023. [Online]. Available: https://github.com/JuliaGraphs/Graphs.jl

[53] “JuliaCI/benchmarktools.jl: a benchmarking framework for the Julia language,” 2023. [Online]. Available: https://github.com/JuliaCI/BenchmarkTools.jl