

Scheduling Stochastic Traffic With End-to-End Deadlines in Multi-hop Wireless Networks

Christos Tsanikidis, Javad Ghaderi
Columbia University, New York, NY, USA
{c.tsanikidis, jghaderi}@columbia.edu

Abstract—Scheduling deadline-constrained packets in multi-hop networks has received increased attention recently. However, there is very limited work on this problem for wireless networks where links are subject to interference. The existing algorithms either provide approximation ratio guarantees which diminish in quality as parameters of the network scale, or hold in an asymptotic regime when the time horizon, network bandwidth, and packet arrival rates are scaled to infinity, which limits their practicality. While attaining a constant approximation ratio has been shown to be impossible in the worst-case traffic setting, it is unclear if the same holds under the stochastic traffic, in a non-asymptotic setting. **In this work, we show that, in the stochastic traffic setting, constant approximation ratio or near-optimal algorithms can be achieved.** Specifically, we propose algorithms that attain $\Omega((1 - \epsilon)/\beta)$ or $\Omega(1 - \epsilon)$ fraction of the optimal value, when the number of channels is $C = \Omega(\frac{\log(L/\epsilon)}{\epsilon^2})$ or $C = \Omega(\frac{\chi^* \log(L/\epsilon)}{\epsilon^3})$ respectively, where L is the maximum route length of packets, χ^* is the fractional chromatic number of the network's interference graph, and β is its interference degree. This marks the first near-optimal results under nontrivial traffic and bandwidth assumptions in a non-asymptotic regime.

I. INTRODUCTION

Delivering packets on time has become increasingly important in real-time applications such as video streaming and conferencing, as well as in emerging applications such as cyber-physical systems, vehicular networks, and Internet of Things. These networks often feature complex multi-hop connectivity and packets need to traverse several links in order to reach their destinations. Packets that fail to meet specific *deadlines*, defined as constraints on the total time from generation of a packet at its source until delivery to its destination, are typically discarded by the application. However, traditional networking algorithms, e.g. [1], [2], have been designed for maximizing throughput and do not provide guaranteed on-time packet delivery over such networks.

Despite the importance of the problem due to its broad applicability, there is very limited work on techniques with strong theoretical guarantees in multi-hop wireless networks. This can be attributed to the inherent complexity of the problem, which includes making online decisions, the exponential growth of scheduling decisions for packet routes and link activations, and the strict deadline constraints on packets. Since the problem is in general NP-hard, efforts have been focused on designing algorithms that provide good performance in terms of “*approximation ratio*”, which is the fraction of the optimal objective value (the weighted sum of timely delivered packets) that the algorithm obtains. Previous

works on scheduling packets with deadlines in multi-hop networks have either focused on worst-case traffic scenarios, yielding pessimistic approximation ratios [3], or stochastic traffic scenarios with guarantees under certain conditions, such as relaxed capacity constraints or asymptotic regimes [4], or considering wired networks [5], [6].

This paper addresses the open question of whether it is feasible to design algorithms with strong performance guarantees (e.g. constant approximation ratio that does not depend on parameters of the network), for scheduling packets with strict deadlines in multi-hop wireless networks. We present near-optimal and constant-approximation algorithms that require a minimum bandwidth (number of channels) that is logarithmic in the maximum length of a packet's route L in the network (with $L \leq d_{max}$, where d_{max} is the maximum deadline of any packet). Our proposed algorithms are of varying computational complexity and guarantees, but rely on careful randomization in their scheduling and routing decisions.

Unlike prior techniques on the problem [3], [5], [6], our obtained approximation ratios are independent of packet weights and the maximum length of any route, L . Additionally, unlike the past work in the stochastic setting [4], our algorithms are applicable to finite time horizons, finite arrival rates, and finite network bandwidth, without scaling these quantities to infinity, and hence, are considerably more practical.

A. Related Work

There has been progress on deadline-constrained scheduling in wireless networks but it has mainly focused on single-hop traffic, e.g., [7], [8], [9], [10], [11]. These works typically evaluate performance based on the attained fraction of the so-called real-time capacity region. This can be related to the approximation-ratio metric [11], [12], which is the considered metric in this paper. Specifically, for stochastic traffic scenarios, it is possible to design $(\frac{\epsilon-1}{\epsilon})$ -approximation algorithms in complete interference graphs and for $C = 1$ [11]. In more general interference graphs, an extension to the well-known Max Weight Scheduling [1] achieves an approximation ratio of $\frac{1}{2}$, and the best-known polynomial-time algorithm yields a $(\frac{1}{\beta+1})$ -approximation [13], [11], where β is the *interference degree* of the network, which is the maximum number of non-interfering links in any link's neighborhood.

The work on multi-hop traffic has mainly focused on *wired* networks (no interference) [14], [15], [16], [17], [18], [6], [5]. The works that provide theoretical guarantees on the problem

in the presence of wireless interference is very limited [4], [3]. In [3], the authors consider the worst-case traffic setting and provide algorithms that achieve $\Omega(1/(\beta \log(\rho \Delta L)))$ approximation ratio, at best, when the number of channels C meets $C = \Omega(\beta \log(\rho \Delta L))$, with L the maximum route length, ρ the maximum-to-minimum packet weight ratio, and Δ the maximum degree of the interference graph. As L or ρ increases, the guarantee deteriorates, and a constant approximation ratio is not provided. In the case of stochastic traffic, [4] proposes an algorithm that is analyzed for the case that the capacity constraints (number of used channels) are relaxed to hold only on average as opposed to strictly at each time. It then argues that as the time horizon $T \rightarrow \infty$, and as arrival rates and number of channels are scaled concurrently to infinity, the loss in the performance due to the relaxation goes to zero. These results are therefore asymptotic and not applicable to practical networks, which are characterized by finite packet arrival rates and finite available bandwidth, and moreover, required to be evaluated in finite time.

In this paper, we provide the first algorithms that yield strong (constant, near-optimal) approximation guarantees in a non-asymptotic regime (finite time horizon, finite number of channels, finite arrival rates). We achieve these results through techniques which are distinct from the prior work.

Finally, we point out that there are works that consider other objectives as opposed to timely delivery within strict deadlines, e.g., providing delay bounds on the packets [19], [20], or optimizing the *age* of the packets [21], [22], [23].

B. Contributions.

The main contributions of this paper can be summarized as follows.

Near-optimal algorithms for wireless networks with general interference graph. We introduce an algorithm that is near-optimal, attaining a $(1 - \epsilon)$ -approximation for the problem of scheduling packets with hard deadlines in multi-hop wireless networks. This is guaranteed in the case of a number of channels that satisfy $C = \Omega(\frac{\chi^* \log(L/\epsilon)}{\epsilon^3})$, and for i.i.d. Bernoulli or Binomial packet arrival processes, where χ^* is the fractional chromatic number of the network's interference graph. Further, our results can be extended to more general arrival processes. The algorithm relies on probabilistic admission and routing of packets in the network and probabilistic scheduling of independent sets over the channels. To the best of our knowledge, this is the first result that provides near-optimal performance in general interference graphs, for finite time horizon, finite number of channels, and finite packet arrival rates.

Efficient greedy maximal scheduling with probabilistic forwarding. The near-optimal algorithm requires randomization over all independent sets, which in general might not be efficient. We propose an efficient algorithm that provides $\Omega((1 - \epsilon)/\beta)$ -approximation when $C = \Omega(\frac{\log(L/\epsilon)}{\epsilon^2})$. The algorithm relies on greedily selecting maximal independent sets over the channels and probabilistic forwarding of packets in the network. Further, this algorithm admits a distributed

implementation. Finally, we show that no polynomial-time algorithm can attain an approximation ratio better than $\Omega(1/\beta^{\epsilon_0})$ for some $\epsilon_0 > 0$. Hence, a polynomial dependence in β in the approximation ratio is unavoidable for any efficient algorithm.

Efficient near-optimal algorithms in perfect graphs.

We show that in the important class of perfect interference graphs, such as bipartite graphs, our methods yield near-optimal performance with polynomial-time complexity. The solution relies on a clique-based formulation which is shown to be polynomially solvable through an ellipsoid algorithm.

C. Notations

We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. Further we denote $[n]_0 := [n] \cup \{0\}$. We use $\mathbb{N} := \{1, 2, 3, \dots\}$, and use \mathbb{R} for the set of real numbers and \mathbb{R}^+ for the set of positive real numbers. We define $a \wedge b := \min\{a, b\}$, and $(a)^+ := \max(a, 0)$.

II. MODEL AND DEFINITIONS

Network Model. We consider a wireless network, comprising a set of nodes \mathcal{V} and a set of communication links \mathcal{L} between these nodes, which form a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$. Time is divided into discrete slots, i.e., $t = 1, 2, 3, \dots$. We assume a packet requires one time slot to be transmitted over any link $\ell \in \mathcal{L}$ using a channel. The network is equipped with C orthogonal channels, as in [4], [3]. Transmissions on different channels do not interfere with one another, however, transmissions on the same channel may interfere with each other. To represent interference between links, we use the conflict (or interference) graph model (e.g. [10], [13], [24]), denoted by a graph $\mathcal{G}_I = (\mathcal{L}, \mathcal{E}_I)$. An edge $(\ell, \ell') \in \mathcal{E}_I$ indicates that links ℓ, ℓ' cannot be scheduled simultaneously within the same time slot and channel.

For convenience, we define an extension of the set of links by including self-loops, i.e., $\bar{\mathcal{L}} = \mathcal{L} \cup \{(u, u) : u \in \mathcal{V}\}$. Scheduling a packet at a given time slot over a self-loop has the interpretation that the packet will remain at the same node for that time slot. We define $\text{Out}(v)$ as the set of outgoing links of a node v or a self-loop, i.e., $\text{Out}(v) = \{(v, u) \in \bar{\mathcal{L}}\}$, and similarly, $\text{Inc}(v)$ as the set of incoming links, i.e., $\text{Inc}(v) = \{(u, v) \in \bar{\mathcal{L}}\}$. The self-loop link (v, v) belongs to both $\text{Out}(v)$ and $\text{Inc}(v)$.

Multi-hop Traffic Model. Packets of different types arrive during a time horizon of length T . We use the set $[J] = \{1, \dots, J\}$ to denote the set of packet types. A packet type $j \in [J]$ is characterized by its source $s_j \in \mathcal{V}$, destination $z_j \in \mathcal{V}$, relative deadline $d_j \in \mathbb{N} \cup \{0\}$, and weight $w_j \in \mathbb{R}^+$. If the packet arrives at the beginning of time slot t , it must reach its destination before the end of time slot $d_j + t$ to yield reward w_j ; otherwise, it is discarded. We denote the number of type $j \in [J]$ packet arrivals at time $t \in [T]$ as $a_j^t \geq 0$, with an arrival rate $\lambda_j^t = \mathbb{E}[a_j^t]$. We use $d_{\max} := \max_{j \in [J]} d_j$ and $\lambda_{\min} := \min_{j \in [J]} \lambda_j$ to denote the maximum deadline and minimum arrival rate of any packet type, respectively. Packets can therefore be in the network until time $\bar{T} := T + d_{\max}$.

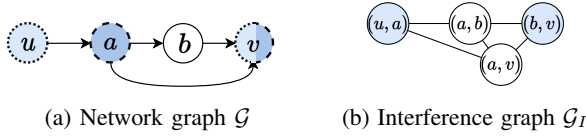


Fig. 1: In (a), a network \mathcal{G} with nodes $\{u, a, b, v\}$ and 4 links is shown, with a single channel. Consider 2 packet types $[J] = \{1, 2\}$ with sources $s_1 = u, s_2 = a$, destinations $z_1 = z_2 = v$, and deadlines $d_1 = 2$ and $d_2 = 1$. Two valid route-schedules for packet type 1 and 2 are: $k = [(u, u) \ (u, a) \ (a, v)]$ and $k' = [(a, v) \ (v, v)]$. In (b), a corresponding one-hop interference graph is shown. A valid network-schedule consisting of links $\{(u, a), (b, v)\}$ is highlighted.

Route and Network Schedules. A type- j packet arriving at time t and scheduled for transmission over the network must be routed through a sequence of links that are activated at specified time slots over specific channels. This sequence carries the packet from its source s_j to its destination z_j before the end of time slot $t + d_j$. We first define the notion of relative route-schedule below, which describes the sequence of links and time slots the packet follows.

Definition 1 (Relative Route-Schedule). A (relative) route-schedule k for a type- j packet, is a walk on \mathcal{G} , $[k_0 \ k_1 \ \dots \ k_{d_j}]$, where $k_\tau \in \bar{\mathcal{L}}$ denotes the link over which the packet is scheduled at the τ -th time slot following its arrival, with $k_0 \in \text{Out}(s_j)$ and $k_{d_j} \in \text{Inc}(z_j)$. We denote the set of all valid route-schedules for packet-type $j \in [J]$ as \mathcal{K}_j .

For notational convenience, we define $k_\tau = \emptyset$ for $\tau > d_j$, where $k \in \mathcal{K}_j$. In addition to the route-schedule, a packet must also be assigned to a specific channel at each time slot it is scheduled over a link (which is not a self-loop). The scheduling algorithm must determine which links are activated in each channel at each time slot. We formalize this through the notion of Network-Schedule.

Definition 2 (Network-Schedule). A network schedule at a time slot t is a collection of C maximal independent sets on the interference graph \mathcal{G}_I . Let \mathcal{I} denote the set of all maximal independent sets of \mathcal{G}_I . On channel c at time t , we use binary variable Z_I^{ct} to indicate whether an independent set $I \in \mathcal{I}$ is selected or not. The independent set specifies the links on each channel that can transmit without causing interference.¹

See Figure 1 for an illustration of route-schedules and network-schedules, for a one-hop interference graph².

Optimization Problem. Our objective is to maximize the weighted sum of the packets that are successfully delivered from their sources to their destinations within their deadlines.

¹An independent set in a graph is a subset of vertices in which no two vertices are adjacent. In the context of the interference graph \mathcal{G}_I , scheduling links from an independent set on each channel ensures that no two interfering links are scheduled simultaneously within the same time slot and channel.

²A one-hop interference, is an interference graph in which, all links that are adjacent in \mathcal{G} , interfere with each other.

Given a random instance of the packets arrival sequence, this optimization can be formulated as an integer program over time horizon T , defined in (1a)-(1f) below. We refer to this optimization as $\text{RI}(T)$, which stands for random integer problem over time horizon of length T . In $\text{RI}(T)$, the optimization is over scheduling decisions $\mathbf{y} = \{y_{jk}^{nt}\}$ and channel activations $\mathbf{Z} = \{Z_I^{ct}\}$, where each y_{jk}^{nt} is a binary variable indicating whether the n -th arriving packet of type j at time t is scheduled using a relative route-schedule $k \in \mathcal{K}_j$ or not, and $\{Z_I^{ct}\}$ are the network-schedule binary variables (Definition 2).

$$\max_{\mathbf{y}, \mathbf{Z}} \sum_{t=1}^T \sum_{j=1}^J w_j \sum_{n=1}^{a_j^t} \sum_{k \in \mathcal{K}_j} y_{jk}^{nt} \quad (:= \text{RI}(T)) \quad (1a)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}_j} y_{jk}^{nt} \leq 1, \quad \forall t \in [T], j \in [J], n \in [a_j^t], \quad (1b)$$

$$\sum_{\tau=(t-d_{\max})^+}^{t \wedge T} \sum_{j=1}^J \sum_{n=1}^{a_j^\tau} \sum_{k: k_t - \tau = \ell} y_{jk}^{n\tau} \leq \sum_{c=1}^C \sum_{I: \ell \in I} Z_I^{ct}, \quad \forall \ell \in \mathcal{L}, t \in [\bar{T}], \quad (1c)$$

$$\sum_{I \in \mathcal{I}} Z_I^{ct} \leq 1, \quad \forall c \in [C], t \in [\bar{T}], \quad (1d)$$

$$Z_I^{ct} \in \{0, 1\}, \quad \forall c \in [C], t \in [\bar{T}], I \in \mathcal{I}, \quad (1e)$$

$$y_{jk}^{nt} \in \{0, 1\}, \quad \forall j \in [J], t \in [T], k \in \mathcal{K}_j. \quad (1f)$$

Constraints (1b) and (1f) indicate that each arriving packet should be assigned to one route-schedule (or not scheduled at all). Constraints (1d) and (1e) indicate that at each time and channel, one independent set is selected for the network-schedule. Constraint (1c) ensures that all the packets scheduled for transmission on link ℓ at time t (left-hand side of (1c)) can be transmitted using the allocated number of channels to that link in the network-schedule (right-hand side of (1c)).

Performance Metric. Our goal is to develop online algorithms that guarantee a strong performance compared to the optimal offline value of $\text{RI}(T)$, on average. The performance is measured based on the achieved approximation ratio γ formally defined below.

Definition 3. Assuming the optimal objective value of $\text{RI}(T)$ is $W_{\text{RI}(T)}^*$, an algorithm ALG provides a γ -approximation to $\text{RI}(T)$ if the objective value achieved using ALG, W_{ALG} , satisfies:

$$\mathbb{E}[W_{\text{ALG}}] \geq \gamma \mathbb{E}[W_{\text{RI}(T)}^*],$$

where the expectation in $\mathbb{E}[W_{\text{RI}(T)}^*]$ is with respect to the randomness in the arrival sequence $\{a_j^t\}$, and $\mathbb{E}[W_{\text{ALG}}]$ is with respect to the randomness in the arrival sequence, and, if applicable, the random decisions of ALG.

III. ALGORITHMS AND MAIN RESULTS

A. Near-Optimal Algorithm for General Interference Graphs

In this section, we introduce MINOS (Multi-hop Interference-aware Near-Optimal Scheduling), a near-

optimal algorithm (w.r.t. Definition 3) for RI(T). MINOS is an algorithm that decides what links to activate on each channel (i.e., selecting a network-schedule defined in Definition 2), and how to route each packet, through randomization. Specifically, **as a preprocessing step, MINOS probabilistically selects an independent set for each channel $c \in [C]$, to obtain a network-schedule which remains fixed for the entire execution of the algorithm.** Subsequently, each packet is forwarded probabilistically based on its age τ (time slots elapsed since the packet's arrival), its type j , and its current node v within the network. The distribution over independent sets, as well as the forwarding probabilities are both selected based on the Linear Program (LP) below (2a-2f) which is solved only once, at the beginning of MINOS.

$$\max_{\mathbf{f}, \bar{\mathbf{Z}}} \sum_{j=1}^J w_j \lambda_j \sum_{\ell \in \text{Inc}(z_j)} f_{j\ell}^{d_j} \quad (:= F_{S0}) \quad (2a)$$

$$\text{s.t.} \quad \sum_{\ell \in \text{Out}(s_j)} f_{j\ell}^0 = 1, \forall j, \quad (2b)$$

$$\sum_{\ell \in \text{Inc}(v)} f_{j\ell}^{\tau-1} = \sum_{\ell \in \text{Out}(v)} f_{j\ell}^{\tau}, \quad \forall \tau \in [d_j], \forall v, \forall j, \quad (2c)$$

$$f_{j\ell}^{\tau} \geq 0, \forall j, \forall \ell \in \bar{\mathcal{L}}, \forall \tau, \quad f_{j\ell}^0 = 0, \quad \forall j, \forall \ell \notin \text{Out}(s_j), \quad (2d)$$

$$\sum_{j=1}^J \sum_{\tau=0}^{d_j} \lambda_j f_{j\ell}^{\tau} \leq (1-\epsilon) C \sum_{\mathbf{I}: \ell \in \mathbf{I}} \bar{Z}_{\mathbf{I}}, \quad \forall \ell \in \bar{\mathcal{L}}, \quad (2e)$$

$$\sum_{\mathbf{I} \in \mathcal{I}} \bar{Z}_{\mathbf{I}} \leq 1, \quad \bar{Z}_{\mathbf{I}} \geq 0, \forall \mathbf{I} \in \mathcal{I}. \quad (2f)$$

The solution variables to LP F_{S0} will guide all scheduling decisions for every $t \in \bar{T}$. Each forwarding variable $f_{j\ell}^{\tau}$ controls the probability that a packet of type j and age τ , that is in the buffer of the transmitter node of link ℓ , is forwarded over ℓ . Note that ℓ might be a self-loop, in which case, the packet remains in the same node for that time slot. Constraints (2b) and (2c) ensure that all packets are continuously forwarded until they expire (their age reaches their deadline). Constraint (2c) ensures the conservation of packets flow over time at each node. Note that packets might arrive at the destination z_j earlier than d_j , in which case, they can continuously be forwarded over the self-loop (z_j, z_j) until expiration (Constraint (2c)). Packets of type j that upon expiry have arrived at their destination z_j yield reward w_j . As a result, the objective of the optimization (2a) is to maximize the expected weighted number of packets that are forwarded to the destination, with age $\tau = d_j$.

Recall that all packets forwarded over links, need to be assigned to channels. To decide what links are activated on each channel c , we use variable $\bar{Z}_{\mathbf{I}}$, which controls the probability of activating the links in independent set \mathbf{I} , i.i.d. for each channel. Therefore $\bar{Z}_{\mathbf{I}}$ is a distribution over independent sets (Constraints (2f)). Constraints (2d) ensure non-negativity of forwarding variables and that all arriving packets can only be forwarded out of their sources. Finally, Constraint (2e)

Algorithm 1: MINOS: Multi-hop Interference-aware Near-Optimal Scheduling

```

1.1 Input: Packet types  $\{(s_i, z_i, d_i, w_i, p_i)\}$ . Distribution
       $\delta = \{\delta_{\mathbf{I}} : \sum_{\mathbf{I} \in \mathcal{I}} \delta_{\mathbf{I}} = 1\}$ .
1.2 Find optimal solution  $\mathbf{f}^* = \{f_{j\ell}^{*}\}$  and  $\bar{\mathbf{Z}}^* = \{\bar{Z}_{\mathbf{I}}^*\}$ 
      (Optimization (2a)-(2f)).
1.3 Initialize link service counters,  $\Phi_{\ell} \leftarrow 0, \forall \ell \in \bar{\mathcal{L}}$ .
1.4 foreach channel  $c \in [C]$  do
1.5    $Y^c \leftarrow \text{select } \mathbf{I} \text{ w.p. } (1 - \frac{\epsilon}{2}) \bar{Z}_{\mathbf{I}}^* + \frac{\epsilon}{2} \delta_{\mathbf{I}}$ .
1.6    $\Phi_{\ell} \leftarrow \Phi_{\ell} + \mathbb{1}(\ell \in Y^c), \forall \ell \in \bar{\mathcal{L}}$ 
1.7 end
1.8 foreach time  $t = 1, 2, \dots, T$  do
1.9    $\Psi_{\ell}^t \leftarrow \emptyset, \forall \ell \in \bar{\mathcal{L}}$ 
1.10  foreach unexpired packet of type  $j$  with age  $\tau$  at
      a node  $v \in \mathcal{V}$  do
1.11    Select link  $\ell \in \text{Out}(v)$  w.p.  $\frac{f_{j\ell}^{\tau}}{\sum_{\ell \in \text{Out}(v)} f_{j\ell}^{\tau}}$ .
1.12    Add the packet to set  $\Psi_{\ell}^t$  for selected link  $\ell$ .
1.13  end
1.14  Transmit up to  $\Phi_{\ell}$  packets arbitrarily from set  $\Psi_{\ell}^t$ 
      and drop any remaining packets.
1.15 end

```

ensures that the average number of packets scheduled over a link (left-hand-side of (2e)), from different packet types and ages, is not more than the average number of channels that have been allocated to that link (right-hand-side). This ensures that on average there is enough channels allocated to that link to successfully schedule all packets. Note that since (2e) only holds on average, it does not guarantee that all the packets are actually transmitted at each time. We, therefore, included a scaling factor $(1-\epsilon)$, which, as we will see in the analysis, allows us to significantly reduce the probability of scheduling more packets than the number of allocated channels.

MINOS, as described in Algorithm 1, starts by solving the LP (F_{S0}), to find \mathbf{f}^* and $\bar{\mathbf{Z}}^*$. Then, using distribution $\bar{\mathbf{Z}}^*$, and an additional input distribution over independent sets $\{\delta_{\mathbf{I}}\}$, a network-schedule is selected in Lines 1.4-1.7. The input-distribution $\{\delta_{\mathbf{I}}\}$ is selected such that each link receives a minimum non-zero transmission rate (this is primarily for technical reasons, adding slack that limits packet drops). After selecting the network-schedule (Lines 1.4-1.7), at each slot t , the algorithm iterates through all unexpired packets of type j at any node v (Line 1.10). For each packet, the algorithm selects an outgoing link based on the optimal forwarding variables, normalized to form a distribution (Line 1.11). For each link ℓ , all packets selected for that link (stored in Ψ_{ℓ}^t , Line 1.12) will be transmitted if the total number of channels assigned to the link is sufficient. Any packets exceeding the available channels for the link will be dropped arbitrarily (Line 1.14).

1) *Performance guarantee for MINOS:* We focus on arrivals that are i.i.d. Bernoulli or Binomial random variables, for each packet type, for each time slot. However, our results can be extended to more general distributions with a slight

modification, as will be discussed in Section IV-A.

Theorem 1. *MINOS achieves $(1-3\epsilon)$ -approximation to $\text{RI}(\mathcal{T})$ for $C \geq \frac{32}{\epsilon^3} \log(L/\epsilon)$, using an input distribution $\delta = \{\delta_I\}$ with $\sum_{I:\ell \in I} \delta_I \geq \zeta, \forall \ell$, and when $T \geq \frac{2d_{\max}^2}{\epsilon}$.*

We now discuss concrete choices for $\{\delta_I\}$. First, we may maximize ζ by solving the *fractional coloring problem* defined in Definition 4 below.

Definition 4 (Fractional coloring problem). *Consider the optimization below, over distribution δ over independent sets, that maximizes the probability of selecting a link:*

$$\max_{\delta, \zeta} \zeta \text{ s.t. } \sum_{I:\ell \in I} \delta_I \geq \zeta, \forall \ell, \quad \sum_{I \in \mathcal{I}} \delta_I = 1, \quad \delta_I \geq 0, \forall I. \quad (3)$$

$\chi^* := 1/\zeta^*$ is called the *fractional chromatic number*.

Identifying δ based on the optimal fractional coloring as in Definition 4, yields $\zeta^* = 1/\chi^*$ in Theorem 1. Consequently:

Corollary 1.1. *MINOS achieves a $(1-3\epsilon)$ -approximation for $C \geq \frac{32\chi^*}{\epsilon^3} \log(L/\epsilon)$ and $T \geq \frac{2d_{\max}^2}{\epsilon}$, where χ^* is the fractional chromatic number, and using $\{\delta_I\}$ obtained from (3).*

Note that solving (3) is no harder than F_{S0} , and both problems are tractable when the number of maximal independent sets of the interference graph \mathcal{G}_I is not very large. Alternatively, one may use Theorem 1 with a suboptimal ζ :

Corollary 1.2. *It is easy to obtain $\delta = \{\delta_I\}$ with $\zeta \geq \frac{1}{\Delta+1}$, where Δ is the maximum degree of any node in the interference graph \mathcal{G}_I , through a simple greedy coloring of \mathcal{G}_I with $\Delta + 1$ colors. Therefore, we obtain Theorem 1 with $C \geq \frac{6(\Delta+1)}{\epsilon^3} \log(L/\epsilon)$.*

Remark 1. Theorem 1 is the first algorithm, to our knowledge, that solves the deadline-constrained scheduling problem near-optimally in wireless networks, under stochastic traffic for finite bandwidth, arrival rates and horizon. Notably, the required number of channels (bandwidth) is independent of the arrival rates and is effectively constant given \mathcal{G}_I . This is in contrast to prior work that requires $C \rightarrow \infty, T \rightarrow \infty, \lambda_j \rightarrow \infty$ [4], or provides pessimistic approximation ratios (assuming worst-case input sequences) dependent on the traffic parameters [3].

B. Greedy Maximal Scheduling with Probabilistic Forwarding

In this section, we introduce GMS-PF (Algorithm 2), an algorithm which reduces the required number of channels compared MINOS, and further, it does not require randomizing over all maximal independent sets. GMS-PF leverages similar forwarding variables $\{f_{j\ell}^\tau\}$ as in MINOS, however, it does not maintain variables for randomizing over independent sets. Instead, packets are assigned greedily to independent sets at

each time slot. The modified LP, referred to as F_S , is:

$$\max_{\mathbf{f}} \sum_{j=1}^J w_j \lambda_j \sum_{\ell \in \text{Inc}(z_j)} f_{j\ell}^{d_j} \quad (:= F_S) \quad (4a)$$

$$\text{s.t. } \sum_{\ell' \in \mathcal{N}_\ell} \sum_{j=1}^J \sum_{\tau=0}^{d_j} \lambda_j f_{j\ell'}^\tau \leq \frac{C}{1+\epsilon}, \quad \forall \ell \in \mathcal{L}, \quad (4b)$$

$$\text{Constraints (2b), (2c), (2d)}. \quad (4c)$$

F_S is similar to F_{S0} , but as implied earlier, lacks static variables for selecting independent sets. Further, the capacity constraint (2e) is modified to (4b), which requires the average total number of packets scheduled over links in the neighborhood $\mathcal{N}_\ell := \{\ell' : (\ell, \ell') \in \mathcal{E}_I\}$ of any link ℓ to be less than the total number of channels (divided by a $(1+\epsilon)$ factor). Algorithm 2 describes GMS-PF.

Algorithm 2: Greedy Maximal Scheduling with Probabilistic Forwarding (GMS-PF)

```

2.1 Input: Packet types  $\{(s_i, z_i, d_i, w_i, p_i)\}$ .
2.2 Find optimal solution  $\mathbf{f}^* = \{f_{j\ell}^{t*}\}$  to  $F_S$  ((4a)-(4c)).
2.3 foreach time  $t = 1, 2, \dots, T$  do
2.4    $\Psi_\ell^t \leftarrow \emptyset, \forall \ell \in \mathcal{L}$     $\blacktriangleright$  Packets assigned to
      each link
2.5   for each unexpired packet of type  $j$  with age  $\tau$  at
      a node  $v$  do
2.6     Select link  $\ell \in \text{Out}(v)$  w.p.  $\frac{f_{j\ell}^{t*}}{\sum_{\ell' \in \text{Out}(v)} f_{j\ell'}^{t*}}$ 
2.7     Add packet to the buffer (set)  $\Psi_\ell^t$  of link  $\ell$ 
2.8   end
2.9   for each channel  $c \in [C]$  do
2.10     $I \leftarrow$  Choose any maximal independent set
      over links with a non-empty buffer  $\Psi_\ell^t$ 
2.11    Schedule packets from the buffers of links
       $\ell \in I$  and remove them from the buffers
2.12  end
2.13  Drop remaining packets in the buffers of links
       $\Psi_\ell^t$ .
2.14 end

```

Compared to MINOS, GMS-PF finds the values of forwarding variables for routing through F_S as opposed to F_{S0} . Further, in the network-schedule (Lines 2.9-2.12), for each channel, a maximal independent set is selected greedily, over the links with packets assigned for scheduling, and an arbitrary packet from the link of the selected independent set is transmitted. The maximal independent set can optionally be selected greedily, by prioritizing links with more pending packets.

Theorem 2 states the performance guarantee of Algorithm 2.

Theorem 2. *Given $\epsilon \in (0, 1/3)$, GMS-PF provides $\frac{1-3\epsilon}{\beta}$ -approximation to $\text{RI}(\mathcal{T})$ when $C \geq 2\left(\frac{1+\epsilon}{\epsilon}\right)^2 \log(L/\epsilon)$ and $T \geq \frac{2d_{\max}^2}{\epsilon}$.*

Efficient Distributed Implementation. Here, we discuss an efficient implementation for selecting maximal independent sets in a distributed manner in Lines 2.9-2.12.

Subroutine for Timer-Based Independent Set Selection (Lines 2.9-2.12): Divide a time slot into a control phase of duration $\omega \ll 1$ and a data-transmission phase of duration 1. At the beginning of the control phase, each link runs C parallel timers, drawn i.i.d. from an exponential distribution with rate $\nu \geq \frac{1}{\omega} \log \frac{CL^2}{\epsilon}$. Each link's timer corresponds to an independent channel. Once a timer of a link runs down to zero, it broadcasts a claim for the corresponding channel, unless a neighbor has claimed the channel earlier. When a link has claimed enough channels to transmit all pending packets, it cancels the remaining timers.

Corollary 2.1. *GMS-PF with Timer-Based Channel Selection in Lines (2.9)-(2.12) provides a $\frac{1-4\epsilon}{\beta}$ -approximation to RI(T) when $C \geq 2(\frac{1+\epsilon}{\epsilon})^2 \log(L/\epsilon)$, $T \geq \frac{2d_{max}^2}{\epsilon}$, and neglecting the control phase overhead.*

C. Efficient Near-Optimal Algorithm for Perfect Graphs

Although, as we will see in Section IV-C, it is generally not possible to achieve significantly better approximations than that of *GMS-PF* with polynomial algorithms, many difficult graph theoretic problems related to our problem, such as the fractional coloring problem, become easier in certain well-behaved families of graphs such as the class of *perfect graphs*³, which among others, contains all bipartite graphs and line graphs. It is of interest to investigate if our techniques can be adapted in this case, to obtain polynomial algorithms with near-optimal approximation as opposed to a fraction of the optimal (as in *GMS-PF*). We show that this is indeed the case, through a variant of our methods of theoretical interest, that leverages the ellipsoid algorithm to solve RI(T) near optimally. First, define F_{sq} , a variant of F_{s0} , that instead of a constraint on the average number of scheduled packets on each link, it considers a constraint on the scheduled packets on each maximal clique $Q \in \text{CLIQUESES}(\mathcal{G}_I)$ of the interference graph \mathcal{G}_I (Constraint (5b)). Conceptually, limiting the average number of packets in every clique to C , replicates to a degree the constraint that packets can be assigned to C independent sets. Typically this leads to a relaxation, but there is a tight connection in perfect graphs [25].

$$\max_{\mathbf{f}} \sum_{j=1}^J w_j \lambda_j \sum_{\ell \in \text{Inc}(z_j)} f_{j\ell}^{d_j} \quad (:= F_{sq}) \quad (5a)$$

$$\text{s.t.} \quad \sum_{\ell \in Q} \sum_{j=1}^J \sum_{\tau=0}^{d_j} \lambda_j f_{j\ell}^{\tau} \leq (1-\epsilon)C, \quad \forall Q \in \text{CLIQUESES}(\mathcal{G}_I), \quad (5b)$$

$$\text{Constraints (2b), (2c), (2d).} \quad (5c)$$

Note that the above LP could have exponentially many clique constraints. However, we can still solve it in polynomial time in perfect graphs, as stated in the following lemma.

³A graph is perfect if in all its induced subgraphs, the chromatic number of the subgraph equals the maximum size of a clique.

Lemma 3. *F_{sq} admits a polynomial time separation oracle, in the class of perfect graphs, and hence is polynomially solvable through the ellipsoid algorithm [26].*

The proof is omitted due to the page constraint.

Algorithm: Near-Optimal Multi-hop Scheduling over Perfect Graphs (NOPG). Consider a variant of MINOS, which, solves F_{sq} (as opposed to F_{s0}) through the ellipsoid algorithm in polynomial time. As in MINOS, we select statically independent sets at the beginning of the execution of the algorithm. We do so as follows. Inspired by [27, Section 2], we can solve a fractional coloring problem over a modification of \mathcal{G}_I , where each link ℓ is replaced by a clique, with a number of nodes proportional to the average number of packets on it: $\sum_{j=1}^J \sum_{\tau=0}^{d_j} \lambda_j f_{j\ell}^{\tau}$. Solving the fractional coloring problem (which is polynomial for perfect graphs [28]) on this graph yields a distribution $\{\bar{Z}_I\}$ that satisfies (2e). We then proceed forwarding packets hop-by-hop as in MINOS (Lines 1.8-1.15).

Theorem 4. *NOPG yields $(1 - O(\epsilon))$ -approximation, when $C = \Omega(\frac{\chi^* \log(L/\epsilon)}{\epsilon^3})$, $T \geq \frac{2d_{max}^2}{\epsilon}$.*

Our clique-based formulation can potentially be used to obtain methods that yield improved approximation ratios in “near perfect” graphs as quantified in [27].

D. Fractional-Coloring-Based Approximation Algorithms

The performance of *GMS-PF* depends on β , a parameter often significantly smaller than Δ [3]. However, in certain interference graphs, such as a star graph, a guarantee that diminishes with β is not ideal, as in this case $\beta = \Delta = |\mathcal{L}| - 1$. In interference graphs with similar behavior, an approximation diminishing with χ might be preferred. For instance, the fractional coloring in a star graph is in contrast only 2. Further, there are classes of graphs where the fractional coloring can be found efficiently or/and bounded by a constant, e.g., in planar interference graphs, $\chi \leq 4$ due to the well-known four-color theorem [29]. A simple modification of our methods allows us to obtain an improved guarantee for such cases, as described below. In particular, given a fractional coloring of \mathcal{G}_I with fractional chromatic number χ , we define a variant of F_{s0} which replaces the capacity constraints in F_{s0} with

$$\sum_{j=1}^J \sum_{\tau=0}^{d_j} \lambda_j f_{j\ell}^{\tau} \leq \frac{C}{\chi(1+\epsilon)}, \quad \forall \ell. \quad (6a)$$

Then selecting independent sets based on the fractional coloring distribution, the expected number of independent sets that include each link is $\frac{C}{\chi}$. Then by using the corresponding forwarding variables, we proceed as in MINOS.

Corollary 4.1. *The fractional-coloring variant yields a $\frac{(1-3\epsilon)}{\chi}$ -approximation when $C = \Omega(\frac{\chi}{\epsilon^2} \log \frac{L}{\epsilon})$ and $T \geq \frac{2d_{max}^2}{\epsilon}$.*

IV. DISCUSSIONS

A. Extension to general distributions

Our main results, e.g., Theorem 1 and Theorem 2, were presented for i.i.d. Bernoulli or Binomial arrivals at each time

slot. However, these assumptions can be relaxed, to allow general stationary arrival distributions. Our generalization has no impact on the attained approximation ratios in Section III, but it adds a multiplicative factor to the required number of channels to achieve the desired performance, dependent on the *dependency-degree* of the distribution of arrivals, a parameter that roughly expresses the maximum number of arrivals of a packet type which are dependent on each other. We omit the details due to the page constraint, but simulation results under different distributions are provided in Section VI-B. Similarly, our techniques can be extended to nonstationary distributions.

B. Estimating the arrival rates

So far, we assumed the knowledge of the arrival rates for simplicity. Our algorithms can be directly modified to leverage estimated or predicted arrival rates. Under stationary arrival rates, the horizon T can be divided into a number of $\lceil \log_2(1/\epsilon) \rceil$ phases. In each phase, the algorithm estimates the arrival rates using all prior arrivals and uses the estimated rates to solve F_{SQ} (or F_S , F_{SQ}). The solution is used for the scheduling decisions in the next phase, according to, e.g., Algorithm 1. Similar theoretical results are obtained, as in e.g. Theorem 1, with an increased required lower bound on T . For nonstationary traffic, predicting the arrival rates becomes important. Given a routine that predicts the arrival rates, our earlier results can be extended. Details are omitted due to page constraints.

C. Upper bound on the approximation ratio

In Section III-A, we presented a near-optimal algorithm which can be computationally demanding, and in Section III-B, we provided efficient constant approximation algorithms for any network topology. In Theorem 5, we present a negative result, showing that in general there is no “efficient” near-optimal algorithm for solving $RI(T)$ ((1a)-(1f)), indicating that our results cannot be improved significantly.

Theorem 5. *No polynomial-time algorithm can yield better than $\Omega(\frac{1}{|\mathcal{C}|^{\epsilon_0}})$ -approximation, for some $\epsilon_0 > 0$, unless $P = NP$, even in the case of Bernoulli packet arrival processes.*

Proof. The proof leverages [30], and is omitted. \square

V. PROOF TECHNIQUES

In this section, we provide a general overview of our proof techniques, which share many similarities across our results. We leverage distinct techniques compared to [3], [4], [5], [6].

$RI(T)$ ((1a)-(1f)) presents five important challenges: (I) it has potentially many variables, due to a variable for each independent set and channel, (II) it is an integer program, (III) the number of constraints and variables grows with T , (IV) it may have exponentially many variables due to a variable for each route-schedule, and (V) it is an online program, with the packet arrival sequence revealed over time. We address these challenges in 5 steps. We mainly focus on the proof outline for Theorem 1 due to space constraints.

A. Proof Outline of Theorem 1

Step 1. Here, we accept the cost of the independent set variables as we seek a near-optimal algorithm (I). A relaxation is later used for, e.g., GMS-PF (Section V-B), to eliminate issue (I) when it makes the execution computationally prohibitive.

Step 2. To address issue (II), we construct the *expected-instance* $EI(T)$ of $RI(T)$, which is a linear program whose optimal value serves as an upper bound to that of $RI(T)$. Consequently, if we design an algorithm based on $EI(T)$, approximating its optimal objective value, we can compare with that of $RI(T)$. The variables of $EI(T)$ can be interpreted as the probabilistic (relaxed) versions of those in $RI(T)$, i.e., x_{jk}^t can be viewed as the probability of setting $y_{jk}^{nt} = 1$.

$$\max_{\mathbf{x}, \mathbf{Y}} \sum_{t=1}^T \sum_{j=1}^J w_j \lambda_j^t \sum_{k \in \mathcal{K}_j} x_{jk}^t \quad (:= EI(T)) \quad (7a)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}_j} x_{jk}^t \leq 1, \quad \forall t \in [T], \forall j \in [J], \quad (7b)$$

$$\sum_{\tau=(t-d_{max})^+}^{t \wedge T} \sum_{j=1}^J \lambda_j^\tau \sum_{k: k_{t-\tau}=\ell} x_{jk}^\tau \leq C \sum_{I: \ell \in I} \bar{Z}_I^t, \quad \forall \ell, t, \quad (7c)$$

$$\sum_{I \in \mathcal{I}} \bar{Z}_I^t \leq 1, \quad \forall t, \quad \bar{Z}_I^t \geq 0 \quad \forall t, I, \quad (7d)$$

$$x_{jk}^t \geq 0, \quad \forall j \in [J], \forall t \in [T], \forall k \in \mathcal{K}_j. \quad (7e)$$

Then, $EI(T)$ can be viewed as maximizing the expected reward if every packet type j has $\mathbb{E}[a_j^t] = \lambda_j^t$ arrivals at each time slot t . Then, we have the following lemma.

Lemma 6. *Let $W_{EI(T)}$ denote the optimal value of the expected instance $EI(T)$. Then it follows:*

$$W_{EI(T)} \geq \mathbb{E}[W_{RI(T)}].$$

Step 3. Here, we show that in the case of fixed arrival rates ($\lambda_j^t \equiv \lambda_j$), a near-optimal stationary solution (time-independent) to the expected instance exists, which allows us to obtain a simplified stationary problem that does not scale with T , hence resolving the issue (III).

Lemma 7. *When $\lambda_j^t \equiv \lambda_j$, there is a stationary solution $x_{ik}^t = x_{ik}^*$ for $EI(T)$ with optimal value $\bar{W}_{EI(T)}$, that satisfies*

$$\bar{W}_{EI(T)} \geq W_{EI(T)} \left(1 - 2 \frac{d_{max}^2}{T} \right). \quad (8)$$

Proof. We provide a proof sketch due to space constraints. Inequality (8) can be shown in three steps. In the first step, we derive from $EI(T)$, a new program $\bar{EI}(T)$, which is symmetric with respect to time shifts (“cyclical”). In the second step, we leverage the symmetry of $\bar{EI}(T)$ to argue that it admits a stationary optimal solution. In third step, we argue that the optimal value for $\bar{EI}(T)$ is close to that of $EI(T)$.

First, for $\bar{EI}(T)$ we modify capacity constraints (7c) to:

$$\sum_{\tau=(t-d_{max})}^t \sum_{j=1}^J \lambda_j \sum_{k: k_{t-\tau}=\ell} x_{jk}^{\tau \bmod (T+1)} \leq C \sum_{I: \ell \in I} \bar{Z}_I^t. \quad (9)$$

This modification only impacts constraints (7c) for $t < d_{max}$ or $t > T$ in $\text{EI}(T)$, now replaced with stronger constraints (9) for $t \in [d_{max}]$ in $\overline{\text{EI}}(T)$. *Second*, we show that $\overline{\text{EI}}(T)$ has a stationary optimal solution. To see this, consider any optimal solution to $\overline{\text{EI}}(T)$ (possibly time-dependent). As the program is time-shift invariant, time shifts of all variables by any amount, should also be optimal. There are at most T distinct shifts (which are all optimal). Since $\overline{\text{EI}}(T)$ is an LP, the average of a set of optimal solutions yields an optimal feasible solution. Further, that solution is a time averaging of the original variables, and hence stationary. *Third*, we argue that the optimal value to $\overline{\text{EI}}(T)$ cannot be significantly worse than that of $\text{EI}(T)$. Indeed, any feasible solution of $\text{EI}(T)$ can be converted to a feasible solution for $\overline{\text{EI}}(T)$ program by setting all variables for times $t < d_{max}$ or $t > T$ to 0. The maximum loss in the objective value, however, due to setting the variables to 0, can be shown to be, per slot, at most $W_{\text{EI}(T)} d_{max}/T$, and over $2d_{max}$ time slots, $W_{\text{EI}(T)} 2 \frac{d_{max}^2}{T}$. \square

Step 4. We resolve (IV) by considering a forwarding-variable based formulation, equivalent to the simplified expected instance with variables $\{x_{jk}\}$ for each route-schedule. The new program is F_{s0} ((2a)-(2f)) (with additionally the number of channels, scaled by $(1 - \epsilon)$).

Step 5. We tie our analysis together, and explain how we obtain a near-optimal online algorithm for $\text{RI}(T)$, using our earlier steps, and thereby addressing (V). To that end, and based on our earlier steps, we argue that can attain the near-optimal value of F_{s0} guided by its solution, as seen in MINOS . This leads to an algorithm that attains a fraction of $\text{EI}(T)$ ((7a)-(7e)) and therefore due to Step 2 and Step 1, a fraction of $\text{RI}(T)$ as well. First, we crucially leverage a careful application of Bernstein-style concentration bounds [31], to argue that packets will not be dropped with high probability when scheduled based on MINOS .

Lemma 8. *The probability of a scheduled packet being dropped due to insufficient channels under MINOS is at most ϵ , if $C \geq \frac{32}{\epsilon^3 \zeta} \log(L/\epsilon)$.*

Proof. We provide a proof sketch due to space constraints. First, we introduce a concentration inequality

Lemma 9 ([31, Chapter 2.2]). *Consider independent random variables $\{X_i \leq B\}$, for $i \in [n]$. If $\mathbf{X} := \sum_{i=1}^n X_i$ and $S := \sum_{i=1}^n \mathbb{E}[X_i^2]$, it follows:*

$$\Pr[\mathbf{X} \geq \mathbb{E}[\mathbf{X}] + \lambda] \leq e^{-\frac{\lambda^2}{2S + 2B\lambda/3}}. \quad (10)$$

We leverage Lemma 9 to bound the probability that any link is assigned more packets than channels, which upper bounds the probability of any packet scheduled on that link to be dropped. We define the following sets of random variables, which count the number of channels assigned to link ℓ , Φ_ℓ ,

number of packets scheduled on the link (equal to $|\Psi_\ell^t|$, at the end of Line 1.13), and their difference:

$$\begin{aligned} \Phi_\ell^c &:= \mathbb{1}(\ell \text{ is selected on channel } c \text{ at } t), \\ \Psi_{j\ell}^\tau &:= \mathbb{1}(\text{type-}j \text{ packet arrives at } \tau, \text{ scheduled on } \ell \text{ at } t), \\ \mathbf{X}_\ell^t &:= \sum_{j=1}^J \sum_{\tau=t-d_{max}}^t \Psi_{j\ell}^\tau - \sum_{c \in \mathcal{C}} \Phi_\ell^c := |\Psi_\ell^t| - \Phi_\ell^t. \end{aligned}$$

We use Lemma 9 based on \mathbf{X}_ℓ^t and S defined below:

$$\begin{aligned} S &:= \sum_{j=1}^J \sum_{\tau=t-d_{max}}^t \mathbb{E}[\Psi_{j\ell}^\tau] + \sum_{c \in \mathcal{C}} \mathbb{E}[(-\Phi_\ell^c)^2] \\ &= \mathbb{E}[\Psi_\ell^t] + \mathbb{E}[\Phi_\ell^t] \leq (2 - \epsilon) \mathbb{E}[\Phi_\ell^t] := S_u. \end{aligned} \quad (11)$$

The probability of violating the channel constraint is:

$$\Pr(|\Psi_\ell^t| > \Phi_\ell) \stackrel{(a)}{\leq} \Pr(|\Psi_\ell^t| - \Phi_\ell > \mathbb{E}[|\Psi_\ell^t| - \Phi_\ell] + \frac{\epsilon}{2} \mathbb{E}[\Phi_\ell]).$$

For $\lambda = \frac{\epsilon}{2} \mathbb{E}[\Phi_\ell]$, and using the bound on S (11):

$$-\frac{\lambda^2}{2S + 2\lambda/3} \leq -\frac{\epsilon^2/4 \mathbb{E}[\Phi_\ell^t]^2}{4\mathbb{E}[\Phi_\ell^t]} \leq -\frac{\epsilon^2 \mathbb{E}[\Phi_\ell^t]}{16} \stackrel{(a)}{\leq} -\frac{\epsilon^3 \zeta C}{32}.$$

Inequality (a) is due to Line 1.5, and the assumption on δ . Now if $C \geq \frac{32}{\epsilon^3 \zeta} \log(L/\epsilon)$, the probability of a packet drop due to insufficient channels is at most $\Pr(|\Psi_\ell^t| > \Phi_\ell) \leq \epsilon/L$ (Lemma 9). A packet can be dropped if it fails at any link along its path. By taking a union bound over the traversed links, we obtain ϵ as a probability bound on the packet drop. \square

We conclude by finishing the proof of Theorem 1 below.

Proof of Theorem 1. Combining our results, we can show:

$$\mathbb{E}[W_{\text{ALG}}] = \mathbb{E}\left[\sum_{t=1}^T \sum_{j=1}^J w_j \sum_{n=1}^{a_j^t} \sum_{k \in \mathcal{K}_j} y_{jk}^{nt}\right] \quad (12)$$

$$\stackrel{(a)}{\geq} \sum_{t=1}^T \sum_{j=1}^J w_j \lambda_j^t \sum_{\ell \in \mathcal{O}(s_j)} (1 - \epsilon) f_{j\ell}^{0*} = (1 - \epsilon) W_{F_{s0}} \quad (13)$$

$$\stackrel{(b)}{\geq} (1 - \epsilon)^2 \overline{W}_{\text{EI}(T)} \stackrel{(c)}{\geq} (1 - \epsilon)^2 (1 - 2d_{max}^2/T) \mathbb{E}[W_{\text{RI}(T)}] \stackrel{(d)}{\geq} (1 - 3\epsilon) \mathbb{E}[W_{\text{RI}(T)}], \quad (14)$$

where (a) follows by Lemma 8, (b) by Step 4, (c) by Lemma 7, and (d) by $T \geq \frac{2d_{max}^2}{\epsilon}$. \square

B. Proof Outline of Theorem 2 for Algorithm 2

The analysis remains largely similar to that of Theorem 1 (Section V-A). We emphasize the main differences here. In Step 1, we relax $\text{RI}(T)$ to obtain $\widetilde{\text{RI}}(T)$, which is identical to $\text{RI}(T)$, but with the constraints for each link ℓ , (1c), summed over the link's neighborhood \mathcal{N}_ℓ , to obtain a new set of constraints (15):

$$\sum_{\tau=(t-d_{max})^+}^{t \wedge T} \sum_{\ell' \in \mathcal{N}_\ell} \sum_{j=1}^J \sum_{n=1}^{a_j^\tau} \sum_{k: k_t - \tau = \ell} y_{jk}^{n\tau} \leq \beta C, \quad \forall \ell, t. \quad (15)$$

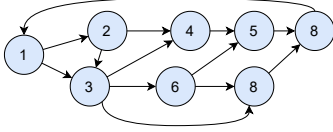


Fig. 2: Network topology used in the simulations for Figure 3a.

β appears, as in each link's neighborhood, it is the maximum number of links that can be activated. Then: $W_{\text{RI}(T)}^* \geq W_{\text{RI}(T)}^*$. In Step 4, the LP in F_S (4a)-(4c), can be related to (15), with the capacity scaled down by $\frac{1}{\beta(1+\epsilon)}$, which is reflected in the obtained approximation ratio.

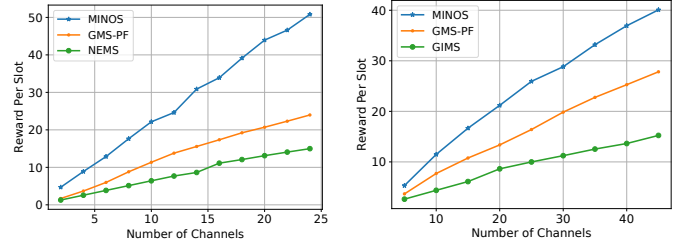
VI. SIMULATION RESULTS

A. Evaluation under one-hop interference and general interference graphs

We compared the performance of MINOS, GMS-PF, with the state-of-the-art worst-case algorithms from [3]. Specifically, the authors in [3] introduced NEMS, for one-hop interference networks, and GIMS, for general interference networks. Due to space constraints, we present the results for two network topologies. For the traffic distribution, we used 20 packet types, with randomly selected source-destination pairs, randomly selected arrival rates in $(0, 100)$ with Binomial distribution on the arrivals, and weights selected randomly between $(0, 1)$. The deadlines of the packets were set to 10. Different choices of these simulation parameters resulted in similar results. For the selected distribution, we measured the per slot reward of timely-delivered packets, for different number of channels.

One-hop interference. We compare MINOS, GMS-PF with NEMS [3] over the 8-node network in Figure 2 under one-hop interference. Note that NEMS outperforms prior algorithms significantly [3], and hence, we focus our comparison with this algorithm. The results are presented in Figure 3a. We observe that MINOS has the best performance, and GMS-PF follows with roughly half the reward of MINOS. Our algorithms outperform the previous state of the art, NEMS. We remark that in this network, there were a total of 27 maximal independent sets considered for randomization by MINOS. The results in the figure are averaged over 10 runs of the algorithms.

General interference. For the general interference model, we used a random geometric graph, with 20 nodes and 30 edges, with links formed between nodes randomly located in proximity. For determining interference between links, each link was assigned a "location" as the midpoint of the locations of its end-nodes. Then, links with distance less than a certain threshold, were assumed to interfere with each other. The resulting interference graph had 185 edges, and 539 maximal independent sets. The results are shown in Figure 3b. Here we compare our algorithms with GIMS [3]. MINOS has the best performance. Recall that MINOS has a variable for each independent set, and therefore, we require 539 variables for the maximal independent set probabilities. The more efficient GMS-PF outperforms GIMS as well, by a considerable margin.



(a) One-Hop Interference

(b) General Interference

Fig. 3: Results for two types of interferences, compared to the best performing worst-case algorithms from [3].

Analogous results are obtained in other graphs as well, but omitted due to space constraints.

B. Evaluation of approximation ratio bounds

A valuable consequence of our results is the availability of an upper bound on the optimal reward of the offline optimal for $\text{RI}(T)$, which only depends on the arrival rates of the traffic distribution, and is based on the optimal value of F_{S0} (see Lemma 6, Lemma 7 and Step 4 of Section V-A). This allows us to provide empirical lower bounds on the approximation ratios for different algorithms and different traffic distributions.

First, we focus on MINOS, which has the best performance in our evaluations, and study the obtained approximation ratio bounds for three different arrival distributions: Binomial, Poisson, and a scaled Bernoulli distribution, with 0 or a_{\max} arrivals, for some a_{\max} . This time, we varied the number of channels and scaled the number of packet types concurrently for Figure 2. In each case, the maximum number of arrivals per time slot for each packet type was chosen between $[10, 20]$, and the arrival rates were kept identical across the three distributions. The performance in each case is averaged over 20 runs. The results are shown in Figure 4a. The Binomial distribution, has the lowest dependency degree (Section IV-A) and exhibits the best behavior. The Poisson distribution has similar performance. A scaled Bernoulli, as expected, exhibits the worst performance (highest dependency degree), but for larger capacity, it becomes near-optimal as well. Finally, in Figure 4b, we evaluated the approximation ratio obtained by different algorithms under one-hop interference model. Note that in this case $\beta = 1/2$. The empirical approximation ratios we obtained are aligned with Theorem 1 and Theorem 2.

VII. CONCLUSIONS

In this paper, we revisited the important problem of scheduling packets subject to strict deadlines and interferences in multi-hop wireless networks. Despite the difficulty of the problem, we obtain significant theoretical improvements under stochastic traffic. In particular, our work provides the first near-optimal and constant approximation algorithms given a finite number of channels and a finite time horizon. Moreover, we showed that in general our results cannot be improved significantly. Finally, our algorithms and techniques are versatile and generalize to a wide range of stochastic traffics.

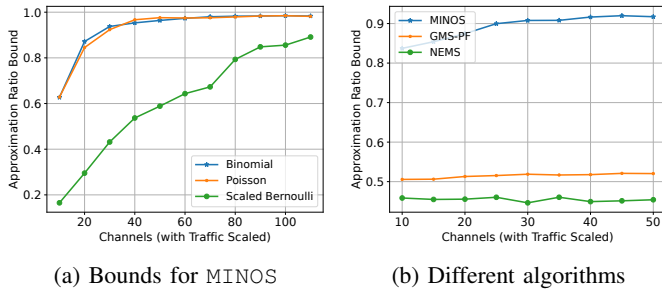


Fig. 4: Empirical approximation ratio bounds over different arrival distributions, and algorithms

REFERENCES

- [1] L. Tassioulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*. IEEE, 1990, pp. 2130–2132.
- [2] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE Journal on Selected areas in Communications*, vol. 24, no. 8, pp. 1452–1463, 2006.
- [3] C. Tsanikidis and J. Ghaderi, "Online scheduling and routing with end-to-end deadline constraints in multihop wireless networks," in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2022, pp. 11–20.
- [4] R. Singh and P. Kumar, "Adaptive csma for decentralized scheduling of multi-hop networks with end-to-end deadline constraints," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1224–1237, 2021.
- [5] Y. Gu, B. Liu, and X. Shen, "Asymptotically optimal online scheduling with arbitrary hard deadlines in multi-hop communication networks," *IEEE/ACM Transactions on Networking*, 2021.
- [6] H. Deng, T. Zhao, and I.-H. Hou, "Online routing and scheduling with capacity redundancy for timely delivery guarantees in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1258–1271, 2019.
- [7] I.-H. Hou, V. Borkar, and P. Kumar, *A theory of QoS for wireless*. IEEE, 2009.
- [8] I.-H. Hou, "Scheduling heterogeneous real-time traffic over fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1631–1644, 2013.
- [9] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, "On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, 2013, pp. 99–108.
- [10] C. Tsanikidis and J. Ghaderi, "On the power of randomization for scheduling real-time traffic in wireless networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 59–68.
- [11] —, "Randomized scheduling of real-time traffic in wireless networks over fading channels," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.
- [12] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 2018.
- [13] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, "On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 72–84, 2014.
- [14] R. Li and A. Eryilmaz, "Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1649–1662, 2012.
- [15] X. Liu, W. Wang, and L. Ying, "Spatial-temporal routing for supporting end-to-end hard deadlines in multi-hop networks," *Performance Evaluation*, vol. 135, p. 102007, 2019.
- [16] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay: A novel routing metric for multi-radio wireless mesh networks," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 46–54.
- [17] Q. Wang, P. Fan, D. O. Wu, and K. B. Letaief, "End-to-end delay constrained routing and scheduling for wireless sensor networks," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.
- [18] Z. Mao, C. E. Koksal, and N. B. Shroff, "Optimal online scheduling with arbitrary hard deadlines in multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 177–189, 2014.
- [19] M. Andrews and L. Zhang, "Packet routing with arbitrary end-to-end delay requirements," in *Proceedings of the thirty-first annual ACM symposium on Theory of Computing*, 1999, pp. 557–565.
- [20] M. Andrews, A. Fernández, M. Harchol-Balter, T. Leighton, and L. Zhang, "General dynamic routing with per-packet delay guarantees of $O(\text{distance} + 1/\text{session rate})$," *SIAM Journal on Computing*, vol. 30, no. 5, pp. 1594–1623, 2000.
- [21] V. Tripathi, R. Talak, and E. Modiano, "Information freshness in multihop wireless networks," *IEEE/ACM Transactions on Networking*, 2022.
- [22] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 486–493.
- [23] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1844–1852.
- [24] R. Singh, P. Kumar, and E. Modiano, "Throughput optimal decentralized scheduling of multi-hop networks with end-to-end deadline constraints: II wireless networks with interference," *arXiv preprint arXiv:1709.01672*, 2017.
- [25] P. Erdős, Z. Füredi, A. Hajnal, P. Komjáth, V. Rödl, and A. Seress, "Coloring graphs with locally few colors," *Discrete mathematics*, vol. 59, no. 1-2, pp. 21–34, 1986.
- [26] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*. Springer Science & Business Media, 2012, vol. 2.
- [27] P. Chalmersook and D. Vaz, "A note on fractional coloring and the integrality gap of lp for maximum weight independent set," *Electronic Notes in Discrete Mathematics*, vol. 55, pp. 113–116, 2016.
- [28] K. Jansen and L. Porkolab, "Preemptive scheduling with dedicated processors: applications of fractional graph coloring," *Journal of Scheduling*, vol. 7, pp. 35–48, 2004.
- [29] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, "The four-colour theorem," *journal of combinatorial theory, Series B*, vol. 70, no. 1, pp. 2–44, 1997.
- [30] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *Journal of the ACM (JACM)*, vol. 41, no. 5, pp. 960–981, 1994.
- [31] F. Chung, R. Fan, F. R. Chung, F. C. Graham, L. Lu, K. F. Chung *et al.*, *Complex graphs and networks*. American Mathematical Soc., 2006, no. 107.