**Penn Libraries**
UNIVERSITY of PENNSYLVANIA

**University of Pennsylvania**
**ScholarlyCommons**

Departmental Papers (ESE)

Department of Electrical & Systems Engineering

November 2006

# Arbitrary Throughput Versus Complexity Tradeoffs in Wireless Networks using Graph Partitioning

Saikat Ray
*University of Pennsylvania*, saikat@seas.upenn.edu

Saswati Sarkar
*University of Pennsylvania*, swati@seas.upenn.edu

Follow this and additional works at: http://repository.upenn.edu/ese_papers

# Arbitrary Throughput Versus Complexity Tradeoffs in Wireless Networks using Graph Partitioning

**Abstract**

Several policies have recently been proposed for attaining the maximum throughput region, or a guaranteed fraction thereof, through dynamic link scheduling. Among these policies, the ones that attain the maximum throughput region require a computation time which is linear in the network size, and the ones that require constant or logarithmic computation time attain only certain fractions of the maximum throughput region. In contrast, in this paper we propose policies that can attain any desirable fraction of the maximum throughput region and require a computation time that is independent of the network size. First, using a combination of graph partitioning techniques and lyapunov arguments, we propose a simple policy for tree topologies under the primary interference model that requires each link to exchange only 1 bit information with its adjacent links and approximates the maximum throughput region using a computation time that depends only on the maximum degree of nodes and the approximation factor. We subsequently develop a framework for attaining arbitrary close approximations for the maximum throughput region in arbitrary networks and interference models and use this framework to obtain any desired tradeoff between throughput guarantees and computation times for a large class of networks and interference models. Specifically, given any $\varepsilon > 0$, the maximum throughput region can be approximated in these networks using a computation time that depends only on the maximum node degree and $\varepsilon$.

# Arbitrary Throughput Versus Complexity Tradeoffs in Wireless Networks using Graph Partitioning

Saikat Ray and Saswati Sarkar

Department of Electrical and Systems Engineering

University of Pennsylvania

### Abstract

Several policies have recently been proposed for attaining the maximum throughput region, or a guaranteed fraction thereof, through dynamic link scheduling. Among these policies, the ones that attain the maximum throughput region require a computation time which is linear in the network size, and the ones that require constant or logarithmic computation time attain only certain fractions of the maximum throughput region. In contrast, in this paper we propose policies that can attain any desirable fraction of the maximum throughput region and require a computation time that is independent of the network size. First, using a combination of graph partitioning techniques and lyapunov arguments, we propose a simple policy for tree topologies under the primary interference model that requires each link to exchange only 1 bit information with its adjacent links and approximates the maximum throughput region using a computation time that depends only on the maximum degree of nodes and the approximation factor. We subsequently develop a framework for attaining arbitrary close approximations for the maximum throughput region in arbitrary networks and interference models and use this framework to obtain any desired tradeoff between throughput guarantees and computation times for a large class of networks and interference models. Specifically, given any $\epsilon > 0$, the maximum throughput region can be approximated in these networks using a computation time that depends only on the maximum node degree and $\epsilon$.

## I. INTRODUCTION

Attaining the maximum throughput region, or a guaranteed fraction thereof, through dynamic link scheduling, is a key design goal in multihop wireless networks. The scheduling problem involves determination of which links transmit packets at any given time. Appropriate scheduling of links is key towards attaining throughput guarantees as the success of transmission in any given link depends on which other links transmit packets simultaneously. The transmission schedules can not be pre-computed, and needs to be determined at every transmission epoch, as the congestion levels in the nodes and the transmission conditions in the wireless medium vary with time, and the statistics of these temporal variations are oftentimes not known a priori. Thus, the time required to determine which links would transmit at any transmission epoch is a key performance metric for any dynamic scheduling policy. The contribution of this paper is to characterize tradeoffs between throughput guarantees and computation times for scheduling policies for different classes of wireless networks.

Owing to the lack of a central controller, at every transmission epoch each link needs to determine whether it would transmit based on its own state and the information it acquires about the states of other nodes. The throughput guarantees usually improve with increase in the information each link (or rather a node which is the source of the link) acquires about the states of other links. The time required for each link to decide whether to transmit at any given time depends on the time required (a) to exchange messages with other links to learn their states and (b) to perform the computations required to arrive at an appropriate decision based on the information acquired. We refer to the total time required in both parts as the schedule computation time, or rather the computation time. The time required in each part increases with increase in the amount of information a link acquires about the states of other links. Thus, an important question is how much information a link should acquire about the states of other links.

The scheduling policies that have been widely investigated can be classified in two broad classes based on the above qualifier: the policies that require each link to know some attribute that depends on the states of (a) all links in the network [14], [15] and (b) only the links that interfere with it [2], [9], [10],

[12], [16]. We refer to the two classes as INFORMATION($N$) and INFORMATION($1$) policies respectively, where $N$ is the number of links in the network. By this nomenclature, INFORMATION($k$) is the class of policies that require each link to learn the states of their $k$-hop interferers. A seminal result has established that policies in INFORMATION($N$) class can attain the maximum possible throughput region in arbitrary wireless networks using $O(N)$ computation time per scheduling decision [14]. Recently, it has been shown that a policy in INFORMATION($1$) class can attain a guaranteed fraction of the maximum throughput region using $O(\Delta_G \log N)$ computation time per scheduling decision where $\Delta_G$ is the maximum degree, or the maximum number of neighbors of any given node, in the network [2]. The contribution of this paper is to show that in certain important classes of wireless networks, for appropriate selection of $k$ between $1$ and $N$, policies can be designed in INFORMATION($k$) class so as to obtain arbitrary close approximations for the maximum throughput region, while requiring a computation time that depends only on $\Delta_G$ and the desired approximation factor and is otherwise independent of the size of the network.

We first consider the primary interference model which mandates that any set of links can be simultaneously scheduled provided they do not have any common node. Under this interference model, when the network topology is a tree, given any positive constant $\epsilon$, we obtain a distributed scheduling policy in INFORMATION($1$) class that (a) approximates the throughput region within a factor of $1 - \epsilon$ and (b) requires a computation time of $O(\Delta_G/\epsilon)$ (Section IV). We next present a general framework for designing INFORMATION($k$) policies for approximating the throughput region arbitrarily closely for arbitrary networks and interference models (Section V). We subsequently use this framework for obtaining arbitrary tradeoffs between throughput guarantees and computation times for large classes of networks, e.g., graphs with limited cyclicity and primary interference models (Section V-B), geometric graphs (Section V-C) and quasi-geometric graphs (Section V-D) under both primary and secondary interference models. For example, for the special case where nodes are embedded in a plane and two links interfere if and only if at least one end-node of one link is within a given distance $D$ of an end-node of the other link (i.e., geometric graphs and secondary interference model), given any positive constant $\epsilon$, we obtain a distributed scheduling policy in INFORMATION($O(\Delta_G^2/\epsilon^2)$) class that (a) approximates the throughput region within a factor of $1 - \epsilon$ and (b) requires a computation time of $(\Delta_G^2/\epsilon^2)^{O(1/\epsilon^2)}$. The throughput and computation time guarantees hold in all cases even when sessions traverse multiple links (Section VI).

Under the primary interference constraints in tree topologies existing policies attain (a) the maximum throughput region using a computation time of $\Theta(N)$ [14] (b) $2/3$ of the maximum throughput region using a computation time of $\Theta(\Delta_G (\log N)^2)$ [12] and (c) $1/2$ of the maximum throughput region using a computation time of $\Theta(\Delta_G)$ [9]. For geometric graphs and secondary interference model, existing policies attain (a) the maximum throughput region using a computation time of $\Theta(N)$ [14] (b) $1/8$ of the maximum throughput region using a computation time of $\Theta(\Delta_G \log N)$ [2] and (c) $1/\Delta_G$ of the maximum throughput region using a computation time of $\Theta(\Delta_G)$ [9]. Our policies therefore attain arbitrary desired tradeoffs between the best known guarantees for throughput and computation times. Specifically, for networks with bounded degree, our policies approximate the throughput region within any constant factor using a computation time which depends only on the approximation factor and does not depend on the network size, whereas existing algorithms that require constant computation time attain an approximation guarantee of at most $1/2$ and $1/8$ for the above cases respectively. For networks with degrees $O(\log N)$ (which happens in several topologies), our policies approximate the throughput region within any constant factor using poly-logarithmic computation time, whereas existing algorithms that use poly-logarithmic computation time attain an approximation factor of at most $2/3$ and $1/8$ for the above cases respectively.

We now briefly describe the design of the proposed policies, and provide the intuition behind the performance guarantees. The proposed policies partition the network in a collection of components - the size of the components depend only on $\Delta_G$ and $\epsilon$. The links that originate in a component but interfere with those in another component are "shut down" i.e., not scheduled. Thus, the links scheduled in each component will not interfere with those scheduled in other components irrespective of the scheduling policy in each component. Hence, the scheduling in different components can now be determined in parallel. Thus, the time required to compute the overall schedule now depends only on the size of each

component and is therefore determined only by $\Delta_G$ and $\epsilon$. We now describe how the links in each component are scheduled. The weight of each link is the number of packets waiting for transmission in the link, the weight of a set of links is the sum of the weights of the links in the set, and a set of links in which no two links interfere with each other is referred to as an independent set of links. In each component the set of links are scheduled such that they constitute the maximum weighted independent set of links in the component. When different partitioning schemes are used at different times and the size of the components in each partition is large enough, each link is shut down only a small fraction of time. Thus, the links selected as above, constitute an independent set whose weight is at least $(1-\epsilon)$ that of the weight of the maximum weighted independent set of links in the entire network. The throughput guarantee now follows from the existing result that a policy that schedules an independent set of links whose weight is at least $1-\epsilon$ that of the weight of the maximum weighted independent set of links attains $1-\epsilon$ fraction of the throughput region [10].

## II. RELATED LITERATURE

The problem of maximizing the throughput region in wireless networks, or attaining a guaranteed fraction thereof, has received significant attention. Tassiulas *et al.* have characterized the maximum throughput region and provided a policy that attains this throughput region in an arbitrary wireless network [15]. This policy schedules the maximum weighted independent set of links in each slot, and hence requires $\Omega(e^N)$ computation time unless $P = NP$. Later, Tassiulas [14] provided randomized scheduling schemes that attain the maximum achievable throughput region, which can be implemented in fully distributed manner using gossip based algorithms [4]. In each slot, this policy randomly selects an independent set of links, compares its weight with the weight of the set of links scheduled in the previous slot and schedules the set that has the larger weight. This policy requires $\Theta(N)$ computation time. All these policies are in the INFORMATION$(N)$ class.

Recently, provable throughput guarantees have been obtained with some policies in INFORMATION$(1)$ class. Dai *et. al.* [3], Lin *et al* [10] and Wu *et. al.* [16] proved that a simple greedy scheduling scheme, maximal matching, attains half the maximum throughput region for the primary interference model; the computation time for maximal matching is $\Theta(\log N)$. Chaporkar *et. al.* [2] proved that maximal matching can be generalized to attain guaranteed fraction of the maximum throughput region for arbitrary interference models, while retaining the logarithmic computation time. Sarkar *et. al.* [12] proved that for primary interference model and tree graphs, a queue length dependent maximal matching attains $2/3$ of the throughput region while using $\Theta\left(\Delta_G \log^2(N)\right)$ computation time. Lin *et. al.* [9] proved that a random access scheme, where links access the medium with a probability that depends on their and their interferers' queue lengths, attains $1/2$ and $1/\Delta_G$ the throughput region for arbitrary networks under primary interference model and secondary interference models respectively, while requiring a $O(\Delta_G)$ computation time.

Our contribution is to introduce the class of INFORMATION$(k)$ policies and prove that for appropriate choices of $k$, policies can be designed in the INFORMATION$(k)$ class so as to obtain arbitrary tradeoffs between the best throughput guarantees and the computation times obtained so far.

The design of our policies rely on the use of graph partitioning techniques. Hunt *et. al.* [7], Kuhn *et. al.* [8], Nieberg *et al* [11] and Sharma *et. al.* [13] have devised graph partitioning techniques for obtaining arbitrary close approximations of maximum weighted independent sets in polynomial growth bounded graphs. A graph is said to be polynomial growth bounded if the maximum number of pairwise independent nodes in any $r$-neighborhood of a node can be upper-bounded by a polynomial in $r$. Many of the graphs we consider, e.g., trees, are not polynomial growth bounded. Even in the polynomial growth bounded graphs we consider, i.e., geometric graphs, existing results [7], [11], [13] approximate maximum weighted independent sets within a factor of $1-\epsilon$ using policies in INFORMATION $(N)$ class which have computation times of $\Theta\left(N + \Delta_G^{f(\epsilon)}\right)$ where $f(\epsilon)$ is a function of $\epsilon$ that increases with decrease in $\epsilon$. Thus selecting the links using these approximation techniques require central control and $\Theta(N)$ time for

computing each schedule. We propose a policy in the INFORMATION($O(\Delta_G^2/\epsilon^2)$) class that computes each schedule in $O(\Delta_G^2/\epsilon^2)$ time using a simpler partitioning technique, and still attains desired approximation guarantees for the maximum throughput region. The partitioning technique used in [8] however requires $\Delta_G^{O(1/\epsilon^2)}$ time for computing each schedule which does not depend on $N$ as well, but this technique approximates a maximum weighted independent set arbitrarily closely only when the weights are all equal. Since different links have different queue lengths in a network, this partitioning technique does not provide throughput guarantees. Finally, Brzezinski *et. al.* have recently used graph partitioning techniques for providing throughput guarantees using $\Theta(N)$ scheduling schemes for networks with multiple channels [1]. Their goal is to divide the graph in subgraphs such that different subgraphs are assigned different channels, and a greedy maximal weight scheduling, which requires $\Theta(N)$ computation time, maximizes the throughput region in each subgraph. Driven by different goals, we use different partitioning schemes.

## III. SYSTEM MODEL

We consider scheduling at the MAC layer in a wireless network. We assume that time is slotted. The topology in a wireless network can be modeled as a graph $G = (V, E)$, where $V$ and $E$ respectively denote the sets of nodes and links. A link exists from a node $u$ to another node $v$ if and only if both $u$ and $v$ can receive each others' signals. Each session represents a triplet $(i, u, v)$ where $i$ is the identifier associated with the session and $u$ and $v$ are source and destinations of the session. At the MAC layer, each session traverses only one link, but multiple sessions may traverse a link. We consider a network with $N$ sessions.

We now introduce terminologies that we use throughout the paper. Some of these are well-known in graph theory; we mention these for completeness.

A node $i$ is a *neighbor* of a node $j$, if there exists a link from $i$ to $j$, i.e., $(i, j) \in E$. Two links (sessions) are *adjacent* to each other if they have common nodes. By definition, a link is adjacent to itself. The *degree of a node* $u$ is the number of links in $E$ originating from or ending at $u$. The *maximum degree* in $G$, $\Delta_G$, is the maximum degree of any node in $G$.

A link $i$ *interferes* with link $j$ if $j$ can not successfully transmit a packet when $i$ is transmitting. A subset of links is said to be *independent* if if no link in the subset interferes with another link in the subset. Let $\mathcal{X}$ be the collection of independent sets of links.

We now describe the packet arrival process. We assume that at most $\alpha_{\max} \geq 1$ packets arrive for any session in any slot. Let $\hat{A}_i(t)$ be the number of packets that session $i$ generates in slot $t$. We assume that a packet arriving in a slot arrives at the end of the slot, and may not be transmitted in the slot. The arrival process $\{\hat{A}_i(t)\}$ is independent and identically distributed for all $t$.

A *scheduling policy* is an algorithm that decides in each slot the subset of sessions that would transmit packets in the slot. Clearly, a subset of sessions can transmit packets in any slot if no two sessions in the subset traverse the same link and the links the sessions traverse constitute an independent set $X$, i.e., if $X \in \mathcal{X}$. Every packet has length 1 slot. Thus, if a session is scheduled in a slot, it transmits a packet in the slot.

Let $\hat{D}_i(t)$ be the number of packets that session $i$ transmits in slot $t$, $i = 1, \ldots, \ldots, N$. Now, $\hat{D}_i(t) \in \{0, 1\}$ and depends on the scheduling policy.

Let $\hat{Q}_i(t)$ be the queue length before the arrivals and the transmissions in slot $t$. Then $\hat{Q}_i(t+1) = \hat{Q}_i(t) + \hat{A}_i(t) - \hat{D}_i(t)$.

*Definition 1:* The network is said to be *stable* if there exists a finite real number $\Gamma$ such that with probability 1,

$$\limsup_{T\to\infty} \sum_{t=0}^{T-1} \hat{Q}_i(t)/T \leq \Gamma, \ i = 1, \ldots, N. \tag{1}$$

We consider a virtual-queue $Q_l$ associated with link $l$ that contains all packets waiting for transmission for all sessions that traverse $l$. Note that the virtual queue in a link $l$ may contain packets of sessions
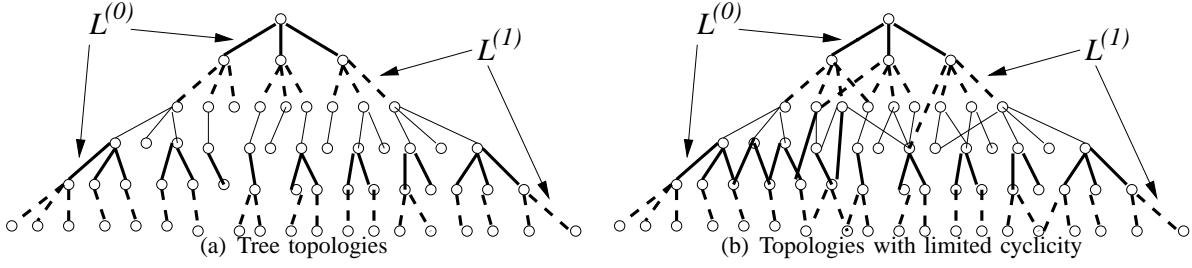
Fig. 1. The figures demonstrate the edge sets $L^{(0)}, L^{(1)}$ under the primary interference model for (a) a tree and (b) topology with limited cyclicity.
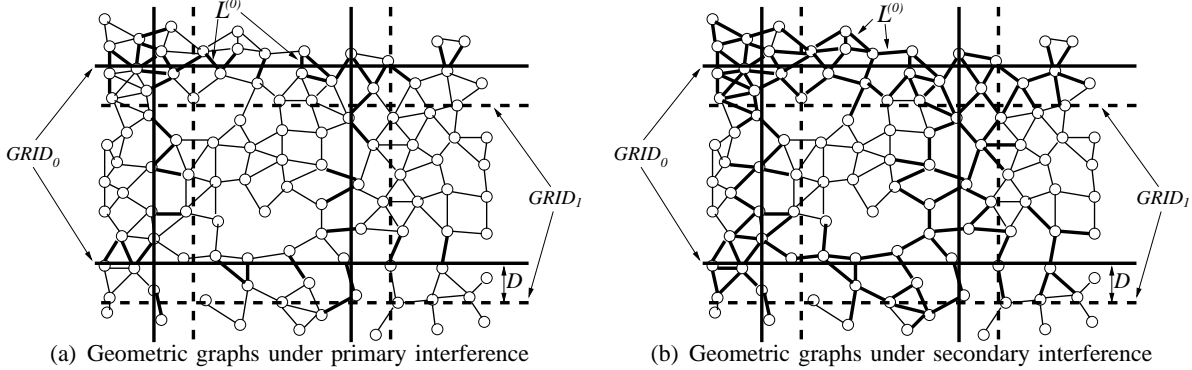


Fig. 2. The figures demonstrate the edge sets $L^{(0)}, L^{(1)}$ for a geometric graph under (a) primary and (b) secondary interference models.

traversing $l$ in both directions. Let $A_l(t)$ and $D_l(t)$ respectively denote the number of arrivals and departures in slot $t$ in virtual queue $Q_l$. Clearly, the arrival process $\{A_l(t)\}$ is independent and identically distributed for all $t$. Let $\mathbf{E}A_l(t) = \lambda_l$. The *arrival rate* of link $i$ is $\lambda_i$, $i = 1, \ldots, |E|$. The *arrival rate vector* $\vec{\lambda}$ is an $|E|-$dimensional vector whose components are the arrival rates.

Now, $Q_l(t+1) = Q_l(t) + A_l(t) - D_l(t)$. Also, (1) holds if and only if $\limsup_{T\to\infty} \sum_{t=0}^{T-1} Q_i(t)/T$ is finite.

The *throughput region* $\Lambda^\pi$ of a scheduling policy $\pi$ is the set of arrival rate vectors $\vec{\lambda}$ for which the network is stable under $\pi$. An arrival rate vector $\vec{\lambda}$ is said to be *feasible* if it is in the throughput region of some scheduling policy. The *maximum throughput region* $\Lambda$ is the set of feasible arrival rate vectors. A scheduling policy $\pi$ is said to approximate the maximum throughput region within a factor $1 - \epsilon$ if for each arrival rate vector $\vec{\lambda} \in \Lambda$, $(1 - \epsilon)\vec{\lambda} \in \Lambda^\pi$.

## IV. INFORMATION(1) POLICY FOR APPROXIMATING THE MAXIMUM THROUGHPUT REGION ARBITRARILY CLOSELY IN TREE TOPOLOGIES

We assume that $G$ is a tree and consider the primary interference model. Under this interference model, two links interfere if and only if they have a common end-point. A matching is a set of links such that no two links in the set are adjacent to each other. Thus, a valid schedule in a slot is a matching in the basic graph $G$. Thus, $\mathcal{X}$ is the set of all matchings in $G$. This interference model is encountered in networks like Bluetooth where each node has a single transceiver and a unique frequency in its neighborhood.

We now describe the scheduling policy which we refer to as TREE-PARTITION-MATCHING $(k)$, and abbreviate as TPM$(k)$. Here, $k$ is a parameter which determines the throughput region and the computation time of the policy.

We first introduce the following notations. The level of a node in a tree is its distance from the root of the tree. A link $l = (u, v)$ is the parent of a link $l' = (v, w)$ if the level of $v$ exceeds that of $w$, and then

$l'$ is a child of $l$. Links $(u, v_1), (u, v_2), \ldots$ are siblings of each other; different priorities are associated with different siblings such that between any two siblings one is older and the other is younger. Let $J_l = \{l' \in E : l' \text{ is a parent or older sibling of } l\}$. For $j = 0, \ldots, k - 1$, let $L^{(j)}$ be the set of links $(u, v)$ such that levels of $u$ and $v$ are $j$ and $j + 1$ modulo $k$ (Figure 1(a)).

A formal description of TPM($k$) follows.

---

TREE-PARTITION-MATCHING ($k$)

In slot $t$, every link selects an integer in the range $[0, \ldots k - 1]$; each integer is selected with probability $1/k$ and all links select the same integer. Let $i(t)$ be the integer selected in slot $t$. A link $l$ contends if and only if (a) its virtual-queue has packets to transmit, and (b) $l \in E \setminus L^{(i(t))}$.

A link schedules itself if and only if (a) it contends and (b) links in $J_l$ do not schedule themselves.

When a link is scheduled, the head of line packet in the corresponding virtual queue is served.

---

Note that TPM ($k$) belongs in the INFORMATION (1) class irrespective of the value of $k$, and is simple to implement since each link only needs to inform its adjacent links about whether its virtual queue is empty or non-empty. We now evaluate the computation time for TPM($k$). Note that in any slot the links that contend constitute a forest such that those in a tree of the forest do not interfere with those in a different tree of the forest. Thus, the scheduling in different components can be determined in parallel. The maximum length of a path in any tree in the forest is $k$. Each link that contends decides whether to schedule itself immediately after it knows the decisions of its parents and older siblings that contend. Thus, each link waits for the scheduling decision of at most $k\Delta_G$ links. Thus, the overall computation time is $O(k\Delta_G)$.

*Theorem 1:* If $\vec{\lambda} \in \mathrm{Int}(\Lambda)$, then $(1 - 1/k)\vec{\lambda} \in \Lambda^{\mathrm{TPM}(k)}$.

We first outline the intuition behind Theorem 1. First, intuitively a scheduling policy $\pi$ that schedules a link $l$ if and only if (a) it has a packet to transmit and (b) links in $J_l$ do not schedule themselves, maximizes the throughput region in a tree. This is because whenever a link $l$ has a packet to transmit, $\pi$ schedules either $l$ or a link in $J_l$; the optimum policy also schedules at most one link in $J_l \cup \{l\}$ in each slot. Clearly, the computation time for $\pi$ is $O(d\Delta_G)$ where $d$ is the depth of the tree, and $d$ is $O(|E|)$. Now, by preventing the contention of a subset $L^{(i(t))}$ of links in each slot $t$, TPM ($k$) partitions the graph in a forest where the depth of each tree is at most $k$, and uses the above scheduling policy in each tree of the forest. This reduces the computation time of TPM ($k$) to $O(k\Delta_G)$. The choice of $L^{(0)}, \ldots, L^{(k-1)}$, and different selections of $i(t) \in \{0, \ldots, k - 1\}$ in each slot $t$ ensures that a link contends with probability $1 - 1/k$ in each slot $t$; this in turn ensures that the maximum throughput region reduces only by a factor of $1 - 1/k$.

*Proof:* The result clearly holds if $k = 1$. Thus, we assume that $k > 1$. The arrival rate vector is $(1 - 1/k)\vec{\lambda}$ where $\vec{\lambda} \in \mathrm{Int}(\Lambda)$. Since $\vec{\lambda} \in \Lambda$ and $X$ constitutes of all matchings of the links, $\sum_{l' \in J_l \cup \{l\}} \lambda'_l < 1$ [6], [15]. Let $\delta = \min\left(\frac{1 - \max \sum_{l' \in J_l \cup \{l\}} \lambda_{l'}}{2|E| \max_{l'} \lambda_{l'}}, 1\right)$. Clearly, $\delta > 0$. Consider a link $l = (u, v)$ where level of $v$ exceeds $u$; then $\chi_l$ denotes the sum of the level of $u$ and the number of older siblings of $l$.

Observe that the queue lengths of the virtual queues constitute a Markov chain. We consider a lyapunov function

$$V(\vec{Q}) = \sum_l \delta^{\chi_l} Q_l^2 + 2 \sum_l \delta^{\chi_l} Q_l \sum_{l' \in J_l} Q_{l'}.$$

We prove that $\mathbf{E}\left(V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right) \mid \vec{Q}(t) = \vec{Q}\right) < -1$ for all sufficiently large $||\vec{Q}||$, where $||\vec{Q}|| = \sqrt{V(\vec{Q})}$. Then, from Foster's theorem (Theorem 2.2.3 in [5]) the Markov chain representing the queue length process $\vec{Q}_l(t)$ is positive recurrent. Also, $\mathbf{E}(Q_l(t)) < \infty$ for each $l$ under the steady state distribution for the above Markov chain. Thus, $\lim_{K \to \infty} \frac{\sum_{t=0}^{K-1} Q_l(t)}{K} < \infty$. The result follows.

$$V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right)$$

$$= \sum_l \delta^{\chi_l} \left(Q_l(t+1) - Q_l(t)\right)\left(Q_l(t+1) + Q_l(t)\right) + 2\sum_l \delta^{\chi_l} Q_l(t+1) \sum_{l' \in J_l} Q_{l'}(t+1)$$

$$- 2\sum_l \delta^{\chi_l} Q_l(t) \sum_{l' \in J_l} Q_{l'}(t)$$

$$\leq 2\sum_l \delta^{\chi_l} \left(A_l(t) - D_l(t)\right) Q_l(t) + 2\sum_l \delta^{\chi_l} \left(A_l(t) - D_l(t)\right)^2 + 2\sum_l \delta^{\chi_l} Q_l(t) \sum_{l' \in J_l} \left(A_{l'}(t) - D_{l'}(t)\right)$$

$$+ 2\sum_l \delta^{\chi_l} \left(A_l(t) - D_l(t)\right) \sum_{l' \in J_l} Q_{l'}(t) + 2\sum_l \delta^{\chi_l} \left(A_l(t) - D_l(t)\right) \sum_{l' \in J_l} \left(A_{l'}(t) - D_{l'}(t)\right)$$

$$\leq 2\sum_l \delta^{\chi_l} Q_l(t) \left( \sum_{l' \in J_l \cup \{l\}} \left(A_{l'}(t) - D_{l'}(t)\right) + \sum_{l': l \in J_{l'}} \delta^{\chi_{l'} - \chi_l} A_{l'}(t) \right) + 4N^2 \alpha_{\max}^2$$

$$\leq 2\sum_l \delta^{\chi_l} Q_l(t) \left( \sum_{l' \in J_l \cup \{l\}} \left(A_{l'}(t) - D_{l'}(t)\right) + \delta \sum_{l': l \in J_{l'}} A_{l'}(t) \right) + 4N^2 \alpha_{\max}^2. \tag{2}$$

The last inequality follows since $0 < \delta \leq 1$, $\chi_l < \chi_{l'}$ if $l \in J_{l'}$. From (2),

$$\mathbf{E}\left(V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right) | \vec{Q}(t) = \vec{Q}\right)$$

$$\leq (2/k) \sum_l \delta^{\chi_l} \sum_{m=0}^{k-1} \mathbf{E}\left(Q_l(t) \left( \sum_{l' \in J_l \cup \{l\}} \left(A_{l'}(t) - D_{l'}(t)\right) + \delta \sum_{l': l \in J_{l'}} A_{l'}(t) \right) | \vec{Q}(t) = \vec{Q}, i(t) = m \right)$$

$$+ 4N^2 \alpha_{\max}^2$$

$$\leq (2/k) \sum_l \delta^{\chi_l} Q_l \left( k(1 - 1/k) \sum_{l' \in J_l \cup \{l\}} \lambda_{l'} - (k-1) + k(1 - 1/k)\delta \sum_{l': l \in J_{l'}} \lambda_{l'} \right) + 4N^2 \alpha_{\max}^2$$

(since $l \in L^{(j)}$ for only one $j \in \{0, \ldots, k-1\}$

and $D_{l'}(t) = 1$ for some $l' \in J_l \cup \{l\}$ unless $Q_l(t) = 0$ or $l \in L^{(i(t))}$)

$$\leq 2(1 - 1/k) \sum_l \delta^{\chi_l} Q_l \left( \sum_{l' \in J_l \cup \{l\}} \lambda_{l'} - 1 + \delta \sum_{l': l \in J_{l'}} \lambda_{l'} \right) + 4N^2 \alpha_{\max}^2$$

$$\leq -2(1 - 1/k)|E| \max_l \lambda_{l'} \delta \sum_l \delta^{\chi_l} Q_l$$

$$< -1 \text{ for sufficiently large } ||\vec{Q}|| \text{ (since } \delta > 0 \text{ and } k > 1).$$

The result follows. ∎

Thus, TPM ($\lceil 1/\epsilon \rceil$) attains a throughput region that is at least $1 - \epsilon$ times that of the maximum throughput region. The computation time for TPM ($\lceil 1/\epsilon \rceil$) is $O(\Delta_G/\epsilon)$.

## V. INFORMATION $(k)$ POLICIES FOR APPROXIMATING THE MAXIMUM THROUGHPUT REGION ARBITRARILY CLOSELY FOR ARBITRARY NETWORKS AND INTERFERENCE MODELS

We first provide a general framework for approximating the maximum throughput region arbitrarily closely in arbitrary networks and interference models using policies in INFORMATION $(k)$ class (Section V-A). Subsequently, we elucidate the utility of the framework in several important classes of networks

and interference models (Section V-B, V-C, V-D). We consider both primary and secondary interference models. For the primary interference model, we generalize the throughput and computation time guarantees presented in the previous section to graphs with limited cyclicity (Section V-B) and geometric and quasi-geometric graphs (Section V-C.1). For the secondary interference model, we obtain similar results for geometric (Section V-C.2) and quasi-geometric graphs (Section V-D). In Section V-E, we discuss how these policies can be implemented.

## A. General Framework

We consider an arbitrary network and an interference model as described in Section III. We consider a policy $\pi(k)$ that consists of $k$ subsets of links $L^{(0)}, \ldots, L^{(k-1)}$ such that the links in a component of $G^{(j)} = (V, E \setminus L^{(j)})$ do not interfere with those in other components of $G^{(j)}$. In every slot $t$, every link selects an integer in the range $[0, \ldots k-1]$; each integer is selected with probability $1/k$ and all links select the same integer. In any slot $t$, the weight of a link is the number of packets waiting for transmission in the virtual queue associated with the link, and the links that constitute a maximum weighted independent set in the interference graph of any component of $G^{(i(t))}$ are scheduled. Without loss of generality, links with zero weight are not scheduled. When a link $l$ is scheduled, the virtual queue associated with $l$ transmits a packet.

Note that $\pi(k)$ is completely specified once $L^{(0)}, \ldots, L^{(k-1)}$ are specified. We now describe when $\pi(k)$ approximates the maximum throughput region within an approximation factor that depends only on $k$. We first introduce the following definition.

*Definition 2:* A collection of subsets $E_1, \ldots, E_q$ of $E$ is said to be *c-approximate* if for (a) any given $|E|$-dimensional vector of non-negative real numbers $\vec{W} = (W_1, \ldots, W_{|E|})$ and (b) any collection of subsets of $E$, $X_1, \ldots X_q$ such that $X_i \in \mathcal{X}$ and $X_i \subseteq E_i$

$$\sum_{i=1}^{q} \sum_{l \in X_i} W_l \le c \max_{X \in \mathcal{X}} \sum_{l \in X} W_l.$$

We now present the key technical lemma that allows us to obtain desired throughput guarantees.

*Lemma 1:* Let $L^{(0)}, \ldots, L^{(k-1)}$ be $c$-approximate. Then,

$$\mathbf{E} \left( \sum_i Q_i(t) D_i(t) | \vec{Q}(t) = \vec{Q} \right) \ge (1 - c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(t).$$

We first provide the intuition behind the result. Now, the weight of the links scheduled by $\pi(k)$ differs from the maximum weight of any schedule in the slot by at most the weight of the maximum weight independent set among links that do not contend in the slot. Now, if $L^{(0)}, \ldots, L^{(k-1)}$ are $c-$approximate, the expected weight of the maximum weight independent set in $L^{(j)}$ for $j = 0 \ldots k-1$ turns out to be at most $c/k$ times that of the weight of the maximum weight independent set in the slot. Thus, the expected weight of the scheduled links is at least $(1 - c/k)$ times that of the weight of the maximum weight of any schedule in the slot.

*Proof:* Let $i(t)$ be the integer selected by links in slot $t$, and

$$B(t) = \arg \max_{\substack{X \in \mathcal{X} \\ X \subseteq L^{(i(t))}}} \sum_{l \in X} Q_l(t).$$

Now, $\sum_i Q_i(t)D_i(t) \geq \left( \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(t) - \sum_{i \in B(t)} Q_i(t) \right)$. Now,

$$
\begin{aligned}
\mathbb{E}\left( \sum_{l \in B(t)} Q_l(t) | \vec{Q}(t) = \vec{Q} \right) &= \sum_{j=0}^{k-1} \mathbf{P}\left( i(t) = j | \vec{Q}(t) = \vec{Q} \right) \mathbb{E}\left( \sum_{l \in B(t)} Q_l(t) | \vec{Q}(t) = \vec{Q}, i(t) = j \right) \\
&= (1/k) \sum_{j=0}^{k-1} \mathbb{E}\left( \sum_{l \in B(t)} Q_l(t) | \vec{Q}(t) = \vec{Q}, i(t) = j \right) \\
&= (1/k) \sum_{j=0}^{k-1} \max_{\substack{X \in \mathcal{X} \\ X \subseteq L^{(j)}}} \sum_{l \in X} Q_l(t) \\
&\leq (c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(t) \text{ (since } L^{(0)}, \ldots, L^{(k-1)} \text{ are } c - \text{approximate)}.
\end{aligned}
$$

Thus, $\mathbb{E}\left( \sum_i Q_i(t)D_i(t) | \vec{Q}(t) = \vec{Q} \right) \geq (1 - c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(t)$. ∎

*Lemma 2:* Let $L^{(0)}, \ldots, L^{(k-1)}$ be $c$-approximate. Then, if $\vec{\lambda} \in \text{Int}(\Lambda)$ and $k > c$, $(1 - c/k)\vec{\lambda} \in \Lambda^\pi$. We first provide the intuition behind the above result. When $L^{(0)}, \ldots, L^{(k-1)}$ are $c$-approximate, from lemma 1 it follows that $\pi(k)$ schedules links such that the expected weight of the scheduled links in any slot is at least $(1 - c/k)$ times that of the maximum weight independent set of links in the slot. The throughput guarantee now follows using lyapunov arguments similar to those in [10], [15]. Refer to appendix A for the proof.

Once we prove that the collection $L^{(0)}, \ldots, L^{(k-1)}$ is $c$-approximate, Lemma 2 allows us to approximate the maximum throughput region within a factor of $1 - \epsilon$ for any $\epsilon > 0$ using $\pi(k)$ for $k = \lceil c/\epsilon \rceil$. In the next subsections we will prove that in large classes of networks the collection $L^{(0)}, \ldots, L^{(k-1)}$ can be selected so as to render it $c$-approximate for different constant factors $c$.

Note that different components in each $G^{(j)}$ can schedule the links in parallel as the links in different components do not interfere. Thus, $\pi(k)$ can be implemented provided in each slot and in each component either one, or all links, know the weights of all links in the component. In either case, $\pi(k)$ is in INFORMATION($\hat{k}$) class where $\hat{k}$ is the maximum diameter of the interference graph* of any component of $G^{(j)}$ for any $j \in \{0, \ldots, k-1\}$. The maximum diameter is upper bounded by the number of links in any component of $G^{(j)}$ for any $j \in \{0, \ldots, k-1\}$. The computation time for $\pi(k)$ will again be determined by the maximum size (number of links or number of nodes or both) of a component in $G^{(j)}$ for $j \in \{0, \ldots, k-1\}$. We will show that for a large class of networks, the size of each component and therefore the overall computation time depends only on $\Delta_G$ and $k$.

### B. Graphs with Limited Cyclicity

Using the above general framework, we generalize the tradeoffs between throughput and computation times to networks with limited cyclicity. Specifically, we assume that there exists a constant $H$ such that the maximum length of a cycle in $G$ is upper bounded by $H+1$. We still consider the primary interference model.

We now describe $L^{(0)}, \ldots, L^{(k-1)}$ for the scheduling policy which we refer to as $H$-LIMITED-CYCLICITY-PARTITION-MATCHING $(k)$, and abbreviate as $H$-LCPM$(k)$. Consider a spanning tree $T$ of $G$. For $H$-LCPM $(k)$, for $j = 0, \ldots, k-1$, $L^{(j)} \subseteq E$ is the set of links $(u, v)$ such that the levels of $u$ and $v$ in $T$ are (a) less than or equal to $jH$ modulo $kH$ and (b) greater than $jH$ modulo $kH$ respectively (Figure 1(b)). Intuitively, for $H$-LIMITED-CYCLICITY-PARTITION-MATCHING $(k)$, when $i(t) = j$, levels $jH$, $jH + kH$, $jH + 2kH, \ldots$ partition the graph, and $L^{(j)}$ consists of the links that cross these levels.

---

*The set of edges of a graph corresponds to the set of vertices in the interference graph. There exists an edge between two vertices $u$ and $v$ in the interference graph if at least one of the corresponding edges interferes with the other.

Clearly, the components of $G^{(j)}$ are such that the links in a component do not interfere with those in other components.

We now evaluate the computation time for $H$-LCPM ($k$). Let the set of edges in $T$ be $\hat{E}$. Note that the maximum length of a path in $T^{(j)} = (V, \hat{E} \setminus L^{(j)})$ is $kH$. Thus each component in $T^{(j)}$ has $O(\Delta_G^{kH})$ nodes. Each component of $G^{(j)}$ consists of several components of $T^{(j)}$. Let $u$ and $v$ be nodes that are in different components of $T^{(j)}$ but the same components of $G^{(j)}$. Then the common ancestor of $u$ and $v$ in $T$ is at a distance of at most $H$ from both $u$ and $v$ in $T$. Thus, at most $H\Delta_G^H$ components of $T^{(j)}$ can constitute the same component in $G^{(j)}$. Thus, each component in $G^{(j)}$ has $O(H\Delta_G^{(k+1)H})$ nodes. Now, each independent set $X$ of links in each component of $G^{(j)}$ is a matching in the corresponding component of $G^{(j)}$. The time needed to compute a maximum weighted matching in each such component is therefore $O(H^3 \Delta_G^{3(k+1)H})$. Thus, the overall computation time is $O(H^3 \Delta_G^{3(k+1)H})$. If $G$ is a bipartite graph, the overall computation time is $O(H^2 \Delta_G^{2(k+1)H})$.

The diameter of any component of $T^{(j)}$ is $O(kH)$. Since a component of $G^{(j)}$ consists of at most $H\Delta_G^H$ components of $T^{(j)}$, the diameter of any component of $G^{(j)}$ is $O(kH^2\Delta_G^H)$. Thus, $H$-LCPM ($k$) belongs in INFORMATION($kH^2\Delta_G^H$) class.

We now prove the following key result which will be used in obtaining throughput guarantees for $H$-LCPM ($k$).

*Lemma 3:* $L^{(0)}, \ldots, L^{(k-1)}$ is $4-$approximate.

*Proof:* Let $\vec{W}$ be an arbitrary $N$-dimensional vector of non-negative real numbers, $X^* = \arg\max_{X \in \mathcal{X}} \sum_{l \in X} W_l$, and $X_0, \ldots, X_{k-1}$ be arbitrary subsets of links such that $X_j \in \mathcal{X}$ (i.e., $X_i$ is a matching) and $X_j \subseteq L^{(j)}$, $j = 0, \ldots, k-1$. We need to prove that $\sum_{j=0}^{k-1} \sum_{l \in X_j} W_l \leq 4 \sum_{l \in X^*} W_l$. Note that for any link $l$,

$$W_l \leq \sum_{i \in X^* \cap S_l} W_i.$$

Let $\eta_l^{(j)} = |X_j \cap S_l|$. Thus,

$$\sum_{j=0}^{k-1} \sum_{l \in X_j} W_l \leq \sum_{l \in X^*} \left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) W_l. \tag{3}$$

Hence, we need to show that $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \leq 4$ for each $l \in X^*$.

Consider $l = (u, v) \in E$. Without loss of generality, let level of $u$ in $T$ be less than or equal to that of $v$ in $T$. There exists a unique $j_l$ such that level of $u$ in $T$ is in $((j_l - 1)H, j_l H] \mod kH$. Note that $l$ is not adjacent to any link in $L^{(q)}$ where $q < (j_l - 1) \mod k$ or $q > (j_l + 1) \mod k$. Since $X_j$s are matchings, at most 1 link in $X_j$ is adjacent to $l$ when $j \in \{(j_l - 1) \mod k, (j_l + 1) \mod k\}$, and at most 2 links in $X_{j_l}$ are adjacent to $l$. Thus, $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \leq 4$ for each $l \in X^*$. The result follows. ∎

*Theorem 2:* If $\vec{\lambda} \in \text{Int}(\Lambda)$, then $(1 - \epsilon)\vec{\lambda} \in \Lambda^{H\text{-LCPM}(\lceil 4/\epsilon \rceil)}$.

Using $k = \lceil 4/\epsilon \rceil$, $c = 4$, Theorem 2 follows from lemmas 3 and 2. Now, $H$-LCPM ($\lceil 4/\epsilon \rceil$) is in INFORMATION($H^2\Delta_G^H/\epsilon$) class and requires $O(H^3\Delta_G^{3H(1+\lceil 4/\epsilon \rceil)})$ computation time. Thus, $H$-LCPM will be useful for small values of $H$.

Finally, Theorem 2 provides the throughput guarantees of 1-LCPM ($\lceil 4/\epsilon \rceil$) for trees as well. But, 1-LCPM ($\lceil 4/\epsilon \rceil$) approximates the maximum throughput region for trees within a factor of $1 - \epsilon$ using a computation time of $\Delta_G^{O(1/\epsilon)}$, whereas TPM ($\lceil 1/\epsilon \rceil$) attains the same throughput guarantee using only $O(\Delta_G/\epsilon)$.

## C. Geometric Graphs

A graph is said to be geometric if nodes are embedded in the first quadrant of the 2-dimensional plane, and a link exists between nodes $u$ and $v$ if and only if the distance between them is less than a certain value say $D$. The distance $D$ is referred to as the transmission range. We first consider the primary interference model (Section V-C.1) and subsequently consider the secondary interference model (Section V-C.2).

*1) Geometric Graphs with primary interference model:* We consider a geometric graph $G$ with primary interference model. We now describe the $L^{(0)}, \ldots, L^{(k-1)}$ for the policy GEOMETRIC-GRAPH-PARTITION-MATCHING$(k)$ which we abbreviate as GGPM$(k)$. We will consider $k$ different grids each of which consists of a series of horizontal and vertical lines. Here, $L^{(j)}$ is the set of links that cross vertical or horizontal lines of grid $j$. We now describe how these grids are constructed. Each grid consists of horizontal and vertical lines parallel to the $x$ and $y$ axes respectively and the distance between any two closest horizontal (vertical) lines is $kD$. Each grid is specified by its first horizontal and vertical lines. The first horizontal and vertical lines of grid $j$ are given by $y = jD$ and $x = jD$ respectively for $j = 0, \ldots k - 1$. Figure 2(a) elucidates the grids and the choices of $L^{(0)}, \ldots, L^{(k)}$. Note that the links in a component of $G^{(j)}$ do not interfere with those in other components.

We first evaluate the computation time for GGPM$(k)$. The overall computation time equals the worst case computation time in a component. Let $\nu$ be the maximum number of nodes in any component of $G^{(j)} = (V, E \setminus L^{(j)})$ for any $j$. We next show that $\nu$ is $O(\Delta_G k^2)$. Thus, the computation time for GGPM$(k)$ is the time required to compute a maximum weighted matching in a component with $O(\Delta_G k^2)$ nodes, which is $O(\Delta_G^3 k^6)$.

*Lemma 4:* For any $j = 0, \ldots, k - 1$, a component in $G^{(j)} = (V, E \setminus L^{(j)})$ has $O(\Delta_G k^2)$ nodes.

*Proof:* Consider some $j \in 0, \ldots, k - 1$. A component in $G^{(j)}$ consists of nodes in a square enclosed by the closest horizontal and vertical lines of the $j$th grid. The side of such a square is at most $kD$ units. Such a square can be filled with $\kappa = \lceil \sqrt{2}k \rceil^2$ small squares with sides equal to $D/\sqrt{2}$. Clearly, $\kappa \le (\sqrt{2} + 1)^2 k^2$. Let $I$ be a maximal independent set of nodes in the component, i.e., there does not exist an edge between any two nodes in $I$ and every node in the component is either in $I$ or has an edge to some node in $I$. Since the distance between any two points in any small square is at most $D$, there cannot be more than one node from $I$ present in any small square. Therefore, $|I| \le \kappa$. Thus, the first part of the lemma follows. Clearly, $\nu \le |I|\Delta_G$. Thus, $\nu \le \kappa \Delta_G \le (\sqrt{2} + 1)^2 \Delta_G k^2$. ∎

Also, the maximum number of links in any component of $G^{(j)}$ is at most $\nu \Delta_G$ which is $O(\Delta_G^2 k^2)$. Thus, GGPM $(k)$ is in INFORMATION$(O(\Delta_G^2 k^2))$ class.

We now prove the following key result which will be used in obtaining throughput guarantees for GGPM$(k)$.

*Lemma 5:* $L^{(0)}, \ldots, L^{(k-1)}$ is $20-$approximate.

*Proof:* The proof is similar to that for lemma 3. We point out the differences. We need to prove that $\sum_{j=0}^{k-1} \sum_{l \in X_j} W_l \le 20 \sum_{l \in X^*} W_l$. Relation (3) holds in this case as well. Hence, we need to show that $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \le 20$ for each $l \in X^*$.

Now, note that the $k$ grids do not share any common line. Let SUPERGRID consist of all lines of all grids. Then SUPERGRID is a grid where the distance between any two closest horizontal (vertical) lines is $D$.

Clearly, $\eta_l^{(j)} = 1$ for any $l \in X_j \cap X^*$. If $l \in X^* \setminus X_j$, $\eta_l^{(j)}$ is the number of links in $X_j$ that interferes with $l$. Since these links are in $X_j$, they do not interfere with each other. Thus, $\eta_l^{(j)} \le 2$ since at most 2 links can be adjacent to $l$ but are not adjacent to each other. Thus, $\eta_l^{(j)} \le 2$ for any $l \in X^*$.

Next, for each $l \in X^*$ we upper-bound the number of $j$s in $\{0, \ldots, k - 1\}$ such that $\eta_l^{(j)} > 0$. Now, $\eta_l^{(j)} > 0$ if either $l \in L^{(j)}$ or $l \notin L^{(j)}$ but $l$ interferes with a link in $L^{(j)}$. Note that for any $l$, $l \in L^{(j)}$ for at most 4 $j$s in $\{0, \ldots, k-1\}$. The observation follows from the fact that $l \in L^{(j)}$ only if both end nodes of $l$ are within a distance of $D$ from a horizontal or vertical line of grid $j$; this can happen at most 2 times for vertical lines and 2 more times for horizontal lines of SUPERGRID. Next, for any $l$, $l \notin L^{(j)}$ but $l$ interferes with (i.e., is adjacent to) a link in $L^{(j)}$ for at most 6 $j$s in $\{0, \ldots, k - 1\}$. This happens only if one of the nodes of $l$ is within $D$ units of a horizontal or vertical line of grid $j$. This can happen at most 3 times for vertical grid lines and 3 more times for horizontal grid lines of SUPERGRID. Thus, for each $l \in X^*$, $\eta_l^{(j)} > 0$ for 10 $j$s in $\{0, \ldots, k - 1\}$. Hence, $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \le 2 \times 10 = 20$ for each $l \in X^*$. ∎

*Theorem 3:* If $\vec{\lambda} \in \text{Int}(\Lambda)$, then $(1 - \epsilon)\vec{\lambda} \in \Lambda^{\text{GPIS}(\lceil 20/\epsilon \rceil, \epsilon)}$.

Using $k = \lceil 20/\epsilon \rceil$, $c = 20$, Theorem 3 follows from lemmas 5 and 2. GGPM ($\lceil 20/\epsilon \rceil$) is in INFORMATION($O(\Delta_G^2/\epsilon^2)$) class and requires $O(\Delta_G^3/\epsilon^6)$ computation time. In the next subsection, we propose a technique that computes each schedule in $O(\Delta_G^2/\epsilon^2)$ time while attaining a throughput region of $(1 - \epsilon)$ times that of the maximum throughput region.

*2) Geometric Graphs with Secondary Interference Model:* We consider a geometric graph $G$ and the secondary interference model. In this interference model, a link $i$ interferes with link $j$ if one end point of $j$ is within distance $D$ from an end point of $i$. Note that if two links interfere under the primary interference model they also interfere under the secondary interference model but the converse is not true. This model is an abstraction of bidirectional wireless links where all transmissions use a single channel and overlapping packets always cause a collision. Note that an independent set of links is no longer a matching in $G$.

We now describe the $L^{(0)}, \ldots, L^{(k-1)}$ for policy GRAPH-PARTITION-INDEPENDENT-SET($k$) which we abbreviate as GPIS($k$). Just as in Section V-C.1, we consider $k$ different grids. Now, $L^{(j)}$ is the set of links that either cross or are adjacent to links that cross vertical or horizontal lines of grid $j$ (Figure 2(b)). Note that the links in a component of $G^{(j)}$ do not interfere with those in other components.

We first evaluate the computation time for GPIS($k$). Again, the overall computation time equals the worst case computation time in a component of $G^{(j)}$. The maximum size of any independent set of links in a component is $O(k^2)$ (in the proof of lemma 4 $|I|$ is $O(k^2)$ for any $I$). Also, each component of $G^{(j)}$ has $O(\Delta_G^2 k^2)$ links. Thus, in any component of $G^{(j)}$, the maximum weighted interference set can be computed in $(\Delta_G^2 k^2)^{O(k^2)}$. Thus, the computation time for GPIS($k$) is $(\Delta_G^2 k^2)^{O(k^2)}$. Again, like GGPM ($k$), GPIS ($k$) is in INFORMATION($O(\Delta_G^2 k^2)$) class.

We make the following observations about $L^{(0)}, \ldots, L^{(k-1)}$.

- Let $\psi_l = \{j : l \in L^{(j)}\}$. Then, $|\psi_l| \le 6$ for any $l \in E$ (*Observation* 1). This holds because $l \in L^{(j)}$ only if at least one of the nodes of $l$ is within a distance of $D$ from a horizontal or vertical line of grid $j$, which can happen at most 3 times for vertical lines and 3 more times for horizontal lines of SUPERGRID.

- For any $l$, $l \notin L^{(j)}$ but $l$ interferes with a link in $L^{(j)}$ for at most 8 $j$s in $\{0, \ldots, k-1\}$ (*Observation* 2). This happens only if one of the nodes of $l$ is within $2D$ units of a horizontal or vertical line of grid $j$ but none of the nodes of $l$ is within a distance of $D$ from any line of grid $j$. This can happen at most 4 times for vertical grid lines and 4 more times for horizontal grid lines of SUPERGRID.

We now prove the following key result which will be used in obtaining throughput guarantees for GPIS($k$).

*Lemma 6:* $L^{(0)}, \ldots, L^{(k-1)}$ is $112-$approximate.

*Proof:* The proof is similar to that for lemma 5. Like in lemma 5, we need to prove that $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \le 112$ for each $l \in X^*$. Now, $\eta_l^{(j)} \le 8$ for any $l \in X^*$ as the number of links that interfere with $l$ but do not interfere with each other is at most 8 [2]. Next, from observations 1 and 2, for each $l \in X^*$, $\eta_l^{(j)} > 0$ for 14 $j$s in $\{0, \ldots, k-1\}$ Hence, $\left( \sum_{j=0}^{k-1} \eta_l^{(j)} \right) \le 8 \times 14 = 112$ for each $l \in X^*$. ∎

*Theorem 4:* If $\vec{\lambda} \in \text{Int}(\Lambda)$, then $(1 - \epsilon)\vec{\lambda} \in \Lambda^{\text{GPIS}(\lceil 112/\epsilon \rceil))}$.

Using $k = \lceil 112/\epsilon \rceil$, $c = 112$, Theorem 4 follows from lemmas 6 and 2. GPIS ($\lceil 112/\epsilon \rceil$) is in INFORMATION($O(\Delta_G^2/\epsilon^2)$) class and requires $(\Delta_G/\epsilon)^{O(1/\epsilon^2)}$ computation time.

We now combine the graph partitioning technique with a policy design technique proposed by Tassiulas [14] so as to attain $1 - \epsilon$ times the maximum throughput region while computing each schedule in only $O(\Delta_G^2/\epsilon^2)$ time. We denote the policy as GRAPH-PARTITION-GRADUAL-IMPROVEMENT-INDEPENDENT-SET($k$) and abbreviate it as GPGIIS ($k$). Note that this policy does not belong in the general class of policies $\pi(k)$ described in Section V-A.

In GPGIIS ($k$) each link $l$ is associated with $k-6$ secondary virtual queues: $Q_{li}^{(S)}$, $i \in \{0, \ldots, k-1\} \setminus \hat{\psi}_l$ where $\hat{\psi}_l$ is the union of $\psi_l$ and $\max(0, 6 - |\psi_l|)$ arbitrary elements of $\{0, \ldots, k-1\} \setminus \psi_l$. Whenever a packet arrives in the virtual queue $Q_l$ it is routed to one of the secondary virtual queues with equal

probability. The policy divides the time axis in frames of $k$ slots. In the $j$th slot of each frame, for different links $l \in E$, the secondary virtual queues $Q_{lj}^{(S)}$ contend. Only the secondary virtual queues that contend can be scheduled for transmission and those that are scheduled for transmission transmit their head of line packets if they are non-empty.

We now describe which contending secondary virtual queues are scheduled for transmission in the $j$th slot of each frame. Note that $Q_{lj}^{(S)}$ does not exist if $l \in L^{(j)}$ as then $j \in \psi_l \subseteq \hat{\psi}_l$. Thus, in the $j$th slot of each frame, no secondary virtual queue associated with any link $l \in L^{(j)}$ contends and at most one secondary virtual queue associated with each link $l \in E \setminus L^{(j)}$ contends. A link is said to contend if one secondary virtual queue associated with it contends. Thus, for each $j$ the links that contend in the $j$th slot of each frame constitute components such that links in different components do not interfere, and the links in each component are a priori rank ordered in some manner. Links in the ordered list sequentially select themselves with a probability $p \in (0,1)$ and those that interfere with the selected links remove themselves from the list. The weight of each contending link is the number of packets waiting for transmission in the contending secondary virtual queue associated with it. The selected links are scheduled in each component if their total weight exceeds the total weight of the links scheduled in the same component in the $j$th slot of the previous frame; otherwise the links scheduled in the same component in the $j$th slot of the previous frame are scheduled again. The contending secondary virtual queues associated with the scheduled links are scheduled.

The computation time of GPGIIS $(k)$ is clearly $O(\gamma)$ where $\gamma$ is the maximum number of links in any component of $G^{(j)}$; hence this computation time is $O(\Delta_G^2 k^2)$. Also, GPGIIS $(k)$ is in INFORMATION$(O(\Delta_G^2 k^2))$ class.

*Theorem 5:* If $\vec{\lambda} \in \text{Int}(\Lambda)$, then $(1 - 8/k)\vec{\lambda} \in \Lambda^{\text{GPGIIS}(k))}$.

*Proof:*

Consider a fictitious system that consists of only the secondary virtual queues $Q_{lj}$ for all $l$. Let $\hat{\Lambda}^{(j)}$ be the maximum throughput region of this fictitious system. Then [15]

$$\text{Int}(\hat{\Lambda}^{(j)}) = \{\vec{\lambda} : \vec{\lambda} = \sum_{\substack{\vec{X} \in \mathcal{X} \\ X \subseteq E \setminus \{l : j \notin \hat{\psi}_l\}}} \beta_X \vec{I}^X, \text{ where } \sum_{\substack{\vec{X} \in \mathcal{X} \\ X \subseteq E \setminus \{l : j \notin \hat{\psi}_l\}}} \beta_X = 1, \beta_X \geq 0 \text{ for each } X \in \mathcal{X} \text{ and } \beta_\phi > 0\}.$$

Consider a scheduling policy $\pi$ that schedules secondary virtual queues that satisfy the following properties.

1) $Q_{lj}(t)$ constitutes an irreducible aperiodic markov chain.
2) In each slot $t$ there is a positive probability associated with scheduling the secondary virtual queues associated with links $l$ in $X^*(t)$ where

$$X^*(t) = \arg \max_{\substack{\vec{X} \in \mathcal{X} \\ X \subseteq E \setminus l : j \notin \hat{\psi}_l}} \sum_{l \in X} Q_{lj}(t).$$

3) If $X_0$ and $X_1$ are the sets of links associated with the secondary virtual queues scheduled in slots $t-1$ and $t$ then $\sum_{l \in X_1} Q_{lj}(t) \geq \sum_{l \in X_0} Q_{lj}(t)$.

Then $\pi$ stabilizes the fictitious system for any arrival rate vector $\vec{\lambda'} \in \text{Int}(\hat{\Lambda}^{(j)})$ [14], [4].

Let $(1 - 6/k)\vec{\lambda}$ be the arrival rate vector in the system where $\vec{\lambda} \in \text{Int}\vec{\lambda})$. Let $\vec{\lambda}^{(j)}$ consist of those components $l$ of $\vec{\lambda}$ for which $j \notin \hat{\psi}_l$. From (5), $\vec{\lambda}^{(j)} \in \text{Int}(\vec{\Lambda}^{(j)})$.

We now consider the secondary virtual queues $Q_{lj}$ for all $l$ at slots $j, k+j, 2k+j, \ldots$ in the actual system. Note that in the actual system these secondary virtual queues are scheduled only in these slots. We can therefore assume without loss of generality that packets arrive in these queues only in these slots as well while the number of arrivals in slot $mk+j$ is the number of arrivals in the actual system between slots $((m-1)k+j, mk+j]$ $([0,j])$ for a positive integer $m$ $(m = 0)$. Note that the expected number of arrivals in secondary virtual queue $Q_{lj}$ in slot $mk+j$ is now $k(1/(k-6))(1-6/k)\lambda_l = \lambda_l$. Thus, the arrival rate vector for these secondary virtual queues is $\vec{\lambda}^{(j)} \in \text{Int}(\vec{\lambda}^{(j)})$. Now, observe that GPGIIS$(k)$

satisfies properties (1) to (3) for these secondary virtual queues, since links that contend in different components of $G^{(j)}$ do not interfere. Thus, for each $j$, the system consisting of these virtual queues are stabilized. The result follows. ∎

Thus, for $k = \lceil 6/\epsilon \rceil$, a policy GPGIIS $(k)$ in INFORMATION($O(\Delta_G^2/\epsilon^2)$) class, attains a throughput region of $1 - \epsilon$ times that of the maximum throughput region using a computation time of $O(\Delta_G^2/\epsilon^2)$. Note that GGPM $(k)$ can be similarly modified to attain a throughput region of $1 - \epsilon$ times that of the maximum throughput region, using $k = \lceil 4/\epsilon \rceil$ and a computation time of $O(\Delta_G^2/\epsilon^2)$.

Finally, GPGIIS$(k)$ attains substantially better tradeoffs than GPIS$(k)$ between throughput and computation time guarantees. But, at the same time GPGIIS$(k)$ is likely to have substantially higher delay as compared to GPIS$(k)$. This is because since, unlike GPIS$(k)$, GPGIIS$(k)$ segregates the incoming traffic in each link in multiple queues and in each slot allows at most one queue in each link to contend, when the contending queue is empty it does not schedule the link even if the link's interferers are not scheduled and other queues in the same link are non-empty. More importantly, unlike GPIS$(k)$, GPGIIS$(k)$ does not schedule the queues whose expected weight is close to that of the maximum weight independent set of queues, and instead attains stability by gradually improving the weight of the scheduled queues. This behavior is known to significantly increase the delay, e.g., simulations have demonstrated that the policy proposed by Tassiulas *et al.* [15] that schedules the maximum weight independent set in each slot has substantially lower delay as compared to the randomized policy proposed again by Tassiulas [14] that attains stability through similar improvements as above. An interesting topic for future research is to investigate the tradeoffs between delay and computation times of scheduling policies.

### D. Quasi-Geometric Graphs

A graph is said to be quasi-geometric if nodes are embedded in the first quadrant of the 2-dimensional plane, and a link (a) exists between nodes $u$ and $v$ if the distance between them is less than $\iota D$ where $\iota < 1$ (b) may exist between nodes $u$ and $v$ if the distance between them is between $\iota D$ and $D$ and (c) does not exist between nodes $u$ and $v$ if the distance between them is greater than or equal to $D$. Under primary interference constraints, as before, two links interfere if and only if they are adjacent. Under secondary interference constraints, two links $l, l'$ interfere if and only if (a) they are adjacent and (b) there is an edge between at least one end node of $l$ and another end node of $l'$.

We first consider the secondary interference model. Now, links $L^{(0)}, \ldots, L^{(k-1)}$ are selected as in the previous subsection, and GPGIIS $(k)$ attains a throughput region which is $1 - 6/k$ of the maximum throughput region as before. However, each component of $G^{(j)}$ has $O(\Delta_G k^2/\iota^2)$ nodes, and $O(\Delta_G^2 k^2/\iota^2)$ links. Thus, the computation time for GPGIIS $(k)$ is $O(\Delta_G^2 k^2/\iota^2)$. Also, GPGIIS $(k)$ is in INFORMATION($O(\Delta_G^2 k^2/\iota^2)$) class. Thus, GPGIIS ($\lceil 6/\epsilon \rceil$) attains a throughput region which is $1 - \epsilon$ of the maximum throughput region, requires a computation time of $O\left(\Delta_G^2/(\iota^2\epsilon^2)\right)$ and is in INFORMATION($O\left(\Delta_G^2/(\iota^2\epsilon^2)\right)$) class. Similarly, under the primary interference model, a throughput region of $1 - \epsilon$ of the maximum throughput region can be attained using a policy in INFORMATION($O(\Delta_G^2/(\iota^2\epsilon^2))$) class which requires $O(\Delta_G^2/(\epsilon^2\iota^2))$ computation time.

### E. Distributed Implementation of the Scheduling Policies

We discuss two possible distributed implementations for $\pi(k)$. In one, in each slot one link in each component determines which links will be scheduled in the component and broadcasts the decisions in the entire component, and in another every link does this computation. For the first each contending link communicates its weight to the link that computes the decisions in its component, and for the second each contending link broadcasts its weight in its entire component. The problem with both implementations is that the size of the packets can not be bounded by any function of the network size since the queue lengths exceed any given number with positive probability. A better solution is to have each link broadcast the increase in its weight since the previous epoch in which the link was in the same component (that is the same random number was selected). Now, the expectation of the magnitude of this increase is $O(k)$

as the expected difference between the consecutive epochs at which the same random number is selected is $O(k)$. Furthermore, the time consumed by the broadcasts in each component is a linear function of the number of links in the component. Thus, the overall expected computation times for the policies are as given in subsections V-B to V-D.

Finally, a slight modification of $\pi(k)$ attains the same maximum throughput guarantees with guarantees on worst case, rather than expected, computation times. The modified policy, denoted as $\pi^{\mathrm{RR}}(k)$, differs from $\pi(k)$ in that it selects the integer $i(t)$ in a round-robin, rather than a random, manner. Now, $\pi^{\mathrm{RR}}(k)$ divides the time axis in frames of $k$ slots and $i(t) = j$ in the $j$th slot of each frame where $j \in \{0, \ldots, k-1\}$. The rest of the policy remains the same. Thus, the increase (and not merely its expectation) in the queue length since the previous epoch in which the link was in the same component is $O(k)$. Clearly, the worst case computation times for the schedules are now as given in subsections V-B to V-D. We next prove the throughput guarantees for $\pi^{\mathrm{RR}}(k)$.

Let $\vec{D}^{(j)}(J)$ denote the departure vector in the $j$th slot of the $J$th frame.

*Lemma 7:* Let $L^{(0)}, \ldots, L^{(k-1)}$ be $c$-approximate. Then, under $\pi^{\mathrm{RR}}(k)$,

$$\sum_i Q_i(Jk) \frac{\sum_{j=0}^k D_i^{(j)}(J)}{k} \geq (1 - c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk) - 3N\alpha_{\max}.$$

*Proof:* Let $Z$ be the set of sessions and $X^J = \arg\max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk)$. Let

$$B^{(j)}(J) = \arg \max_{\substack{X \in \mathcal{X} \\ X \subseteq L^{(j)}}} \sum_{l \in X} Q_l(Jk + j).$$

$$
\begin{aligned}
\sum_i Q_i(Jk) D_i^{(j)}(J) &\geq \sum_i Q_i(Jk+j) D_i^{(j)}(J) - N\alpha_{\max} \\
&\geq \max_{\substack{X \in \mathcal{X} \\ X \subseteq Z \setminus L^{(j)}}} Q_i(Jk+j) - N\alpha_{\max} \\
&\geq \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk+j) - \sum_{i \in B(Jk+j)} Q_i(Jk+j) - N\alpha_{\max} \\
&\geq \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk+j) - \sum_{i \in B(Jk+j)} Q_i(Jk) - 2N\alpha_{\max} \\
&\geq \sum_{i \in X^J} Q_i(Jk+j) - \sum_{i \in B(Jk+j)} Q_i(Jk) - 2N\alpha_{\max} \\
&\geq \sum_{i \in X^J} Q_i(Jk) - \sum_{i \in B(Jk+j)} Q_i(Jk) - 3N\alpha_{\max} \\
&= \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk) - \sum_{i \in B(Jk+j)} Q_i(Jk) - 3N\alpha_{\max}.
\end{aligned}
$$

Thus, $\displaystyle \sum_i Q_i(Jk) \frac{\sum_{j=0}^{k-1} D_i^{(j)}(J)}{k} \geq \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk) - (1/k) \sum_{j=0}^{k-1} \sum_{i \in B(Jk+j)} Q_i(Jk) - 3N\alpha_{\max}.$ (4)

From (4) and since $L^{(0)}, \ldots, L^{(k-1)}$ are $c$-approximate,

$$\sum_i Q_i(Jk) \frac{\sum_{j=0}^{k-1} D_i^{(j)}(J)}{k} \geq (1 - c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(Jk) - 3N\alpha_{\max}.$$

∎

*Lemma 8:* Let $L^{(0)}, \ldots, L^{(k-1)}$ be $c$-approximate. Then, if $\vec{\lambda} \in \text{Int}(\Lambda)$ and $k > c$, then $(1 - c/k)\vec{\lambda} \in \Lambda^{\pi^{\text{RR}}(k)}$.

*Proof:* Clearly, the queue length process $\vec{Q}(Jk)$ for $J = 0, 1, \ldots$ constitutes an irreducible, aperiodic Markov chain under $\pi$. Let the arrival rate vector be $(1 - c/k)\vec{\lambda}$ where $\vec{\lambda} \in \text{Int}(\Lambda)$. We will consider the lyapunov function $V(\vec{Q}) = \sum_i Q_i^2$. Using lemma 7 and arguments similar to that used for proving lemma 2, we can prove that under $\pi^{\text{RR}}(k)$

$$\mathbf{E}\left(V\left(\vec{Q}((J+1)k)\right) - V\left(\vec{Q}(Jk)\right) | \vec{Q}(Jk) = \vec{Q}\right)$$
$$\leq -2k(1 - c/k)\beta_\phi \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i + 6Nk\alpha_{\max} + Nk^2\alpha_{\max}^2$$
$$< -1 \text{ for all sufficiently large } ||\vec{Q}||.$$

Then, from Foster's theorem (Theorem 2.2.3 in [5]) the Markov chain is positive recurrent. Also, $\mathbf{E}\left(Q_i(Jk)\right) < \infty$ for each $i$ under the steady state distribution for the above Markov chain. Thus, $\lim_{K \to \infty} \frac{\sum_{J=0}^{K-1} Q_i(Jk)}{K} < \infty$. Then, since under $\pi$, the queue length of session $i$ in any slot in a frame exceeds the queue length of $i$ at the beginning of the frame by at most $\alpha_{\max}k$, $\limsup_{K \to \infty} \frac{\sum_{t=0}^{K-1} Q_i(t)}{t} \leq \lim_{K \to \infty} \frac{\sum_{J=0}^{K-1} Q_i(Jk)}{K} + \alpha_{\max}k < \infty$. The result follows. ∎

We now consider the throughput guarantees of $\pi^{\text{RR}}$ for different classes of networks considered in subsections V-B to V-D. The choice of $L^{(0)}, \ldots, L^{(k-1)}$ for different classes of networks remain the same as in subsections V-B to V-D. Using $k = \lceil 4/\epsilon \rceil$, $c = 3$ Theorem 2 follows from lemmas 3 and 8 for $H$-LCPM$^{\text{RR}}(k)$. Using $k = \lceil 20/\epsilon \rceil$, $c = 20$, Theorem 3 follows from lemmas 5 and 8 for GGPM$^{\text{RR}}(k)$. Using $k = \lceil 112/\epsilon \rceil$, $c = 112$, Theorem 4 follows from lemmas 6 and 8 for GPIS$^{\text{RR}}(k)$.

# VI. MULTI-HOP SESSIONS

We now allow sessions to traverse multiple hops. We first describe the modifications required in the system model and performance goals for accommodating this generalization. We subsequently generalize the framework presented in Section V for attaining arbitrary tradeoffs between throughput guarantees and computation times.

## A. Generalized System Model

We now assume that the network consists of $N$ end-to-end sessions, indexed as $1, \ldots, N$. Each end-to-end session can be viewed as a collection of several hop-by-hop connections, one for each link it traverses; each of these hop-by-hop connections is called a *session-link* of the session considered. Each session-link is of the form $(u, v)$, where $u$ and $v$ represent the transmitter and the receiver, respectively, of the session-link. We assume that there are $M$ session-links in the network (over all sessions), and these are indexed by $1, \ldots, M$. The interference relations are as in Section III.

Each session-link corresponds to a separate virtual queue and the number of virtual queues associated with each link equals the number of session-links traversing it; we assume that this number is at most $\mu$. The packet arrival process is the same as before, and only the first session-link of each session receives the exogenous arrivals. Now, the queue-length and departure vectors, $\vec{Q}(t), \vec{D}(t)$, are $M$-dimensional vectors respectively representing the queue lengths of the session-links and which session-links are served in slot $t$.

Let $R$ be a $M \times M$ dimensional matrix such that (a) $R_{ij} = 1$ if $i = j$ (b) $R_{ij} = -1$ if $i$ and $j$ are session-links of the same session and $i$ constitutes the hop after $j$ and (c) $R_{ij} = 0$ otherwise.

$$\vec{Q}(t+1) = \vec{Q}(t) - R\vec{D}(t) + \vec{A}(t).$$

The definition for stability is the same except that session-links are considered instead of sessions. The definitions for the throughput regions are the same as before.

## B. *Scheduling policies for approximating the maximum throughput region arbitrary closely*

We now generalize the policy $\pi(k)$ presented in Section V. The modified policy, denoted as $\pi^{\mathrm{MH}}(k)$, differs from $\pi(k)$, in only the assignment of link weights. For $\pi^{\mathrm{MH}}(k)$ in any slot $t$, the weight of a session-link (or a virtual-queue) $l = (u, v)$ of session $i$, $G_i(t)$, is (a) the difference between the queue lengths of session-links $l$ and $m$ where $m$ is the session-link of $i$ originating from $v$, if $v$ is not the destination for $i$ and (b) $Q_i(t)$ otherwise. The weight of a link is the maximum weight of a session-link traversing the link. Note that in the special case that all sessions traverse one link, only one virtual queue is associated with each link and for any virtual-queue $i = (u, v)$, $v$ is the destination of the session and hence its weight $G_i(t)$ equals $Q_i(t)$ as in Section V. Whenever a link is scheduled, the session-link that has the maximum weight among those that traverse the link is served. The policies $\pi^{\mathrm{MH}}(k)$ and $\pi(k)$ are otherwise the same.

*Lemma 9:* Let $L^{(0)}, \ldots, L^{(k-1)}$ be $c$-approximate. Then, if $\vec{\lambda} \in \mathrm{Int}(\Lambda)$ and $k > c$, then $(1 - c/k)\vec{\lambda} \in \Lambda^{\pi^{\mathrm{MH}}(k)}$.

We prove lemma 9 in appendix A.

We now consider the throughput guarantees of $\pi^{\mathrm{RR}}$ for different classes of networks considered in subsections V-B to V-D. The choice of $L^{(0)}, \ldots, L^{(k-1)}$ for different classes of networks remain the same as in subsections V-B to V-D. Using $k = \lceil 4/\epsilon \rceil$, $c = 3$ Theorem 2 follows from lemmas 3 and 8 for $H$-LCPM$^{\mathrm{MH}}(k)$. Using $k = \lceil 20/\epsilon \rceil$, $c = 20$, Theorem 3 follows from lemmas 5 and 8 for GGPM$^{\mathrm{MH}}(k)$. Using $k = \lceil 112/\epsilon \rceil$, $c = 112$, Theorem 4 follows from lemmas 6 and 8 for GPIS$^{\mathrm{MH}}(k)$.

Clearly, the computation times in each case increase only by an additive term of $\mu$; this increase is necessary to compute the weight of each link as the maximum of weights of $\mu$ virtual queues associated with it.

# VII. ACKNOWLEDGMENT

# APPENDIX

## A. *Proof for lemma 2*

*Proof:*

Let the arrival rate vector be $(1-c/k)\vec{\lambda}$ where $\vec{\lambda} \in \mathrm{Int}(\Lambda)$. Clearly, under $\pi$, $\vec{Q}(t)$ constitute an aperiodic irreducible Markov chain, and $\mathbf{E}\left(\sum_i Q_i(t) D_i(t) | \vec{Q}(t) = \vec{Q}\right) \geq (1 - c/k) \max_{X \in \mathcal{X}} \sum_{i \in X} Q_i(t)$. We will consider the lyapunov function $V(\vec{Q}) = \sum_i Q_i^2$, and prove that under $\pi$,

$\mathbf{E}\left(V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right) | \vec{Q}(t) = \vec{Q}\right) < -1$ for all sufficiently large $||\vec{Q}||$, where $||\vec{Q}|| = \sqrt{V(\vec{Q})}$. Then, from Foster's theorem (Theorem 2.2.3 in [5]) the Markov chain representing the queue length process is positive recurrent. Also, $\mathbf{E}(Q_i(t)) < \infty$ for each $i$ under the steady state distribution for the above Markov chain. Thus, $\lim_{K \to \infty} \frac{\sum_{t=0}^{K-1} Q_i(t)}{K} < \infty$. The result follows.

Let $\vec{I}^X$ denote the indicator vector for set $X \in \mathcal{X}$. Note that $\phi \in \mathcal{X}$. Then, $\mathrm{Int}(\Lambda)$ can be characterized as follows [15]:

$$\mathrm{Int}(\Lambda) = \{\vec{\lambda} : \vec{\lambda} = \sum_{\vec{X} \in \mathcal{X}} \beta_X \vec{I}^X, \text{ where } \sum_{\vec{X} \in \mathcal{X}} \beta_X = 1 \text{ and } \beta_X \geq 0 \text{ for each } X \in \mathcal{X} \text{ and } \beta_\phi > 0\}. \quad (5)$$

Now,

$$
\mathbf{E}\left(\left(\vec{A}(t)\right)^T \vec{Q}(t)|\vec{Q}(t)=\vec{Q}\right) = (1-c/k)\vec{\lambda}^T\vec{Q}
$$
$$
= (1-c/k)\sum_{X\in\mathcal{X}}\beta_X\left(\left(\vec{I}^X\right)^T\vec{Q}\right) \text{ where } \beta_\phi>0 \text{ (from (5))}
$$
$$
= (1-c/k)\sum_{X\in\mathcal{X}\setminus\{\phi\}}\beta_X\sum_{i\in X}Q_i
$$
$$
\leq (1-c/k)\left(\sum_{X\in\mathcal{X}\setminus\{\phi\}}\beta_X\right)\max_{X\in\mathcal{X}}\sum_{i\in X}Q_i
$$
$$
= (1-c/k)(1-\beta_\phi)\max_{X\in\mathcal{X}}\sum_{i\in X}Q_i. \tag{6}
$$

$$
\mathbf{E}\left(V\left(\vec{Q}(t+1)\right)-V\left(\vec{Q}(t)\right)|\vec{Q}(t)=\vec{Q}\right)
$$
$$
= \mathbf{E}\left(\left(\vec{Q}(t+1)-\vec{Q}(t)\right)^T\left(\vec{Q}(t+1)+\vec{Q}(t)\right)|\vec{Q}(t)=\vec{Q}\right)
$$
$$
= \mathbf{E}\left(\left(\vec{A}(t)-\vec{D}(t)\right)^T\left(2\vec{Q}(t)+\vec{A}(t)-\vec{D}(t)\right)|\vec{Q}(t)=\vec{Q}\right)
$$
$$
\leq 2\mathbf{E}\left(\left(\vec{A}(t)-\vec{D}(t)\right)^T\vec{Q}(t)|\vec{Q}(t)=\vec{Q}\right)+N\alpha_{\max}^2
$$
$$
\leq -2(1-c/k)\beta_\phi\max_{X\in\mathcal{X}}\sum_{i\in X}Q_i+N\alpha_{\max}^2 \text{ (from Lemma 1 and (6))}
$$
$$
< -1 \text{ for all sufficiently large } ||\vec{Q}|| \text{ (since } \beta_\phi>0, 0<c<k).
$$

∎

*Proof for lemma 9*

We first state and prove lemma 10 for $\pi^{\text{MH}}(k)$ which will be useful in proving lemma 9.

*Lemma 10:* Let $L^{(0)},\ldots,L^{(k-1)}$ be $c$-approximate. Then,

$$
\mathbf{E}\left(\sum_i G_i(t)D_i(t)|\vec{Q}(t)=\vec{Q}\right)\geq(1-c/k)\max_{X\in\mathcal{X}}\sum_{i\in X}G_i(t).
$$

*Proof:*

Let $i(t)$ be the number selected by links in slot $t$, and

$$
B(t)=\arg\max_{\substack{X\in\mathcal{X}\\X\subseteq L^{(i(t))}}}\sum_{l\in X}G_l(t).
$$

Again, $\sum_i G_i(t)D_i(t) \geq \left(\max_{X\in\mathcal{X}} \sum_{i\in X} G_i(t) - \sum_{i\in B(t)} G_i(t)\right)$. Now,

$$
\mathbb{E}\left(\sum_{l\in B(t)} G_l(t)/\vec{Q}(t)\right) = (1/k)\sum_{j=0}^{k-1} \max_{\substack{X\in\mathcal{X}\\X\subseteq L^{(j)}}} \sum_{l\in X} G_l(t)
$$

(using same arguments as in the proof for lemma 1)

$$
= (1/k)\sum_{j=0}^{k-1} \max_{\substack{X\in\mathcal{X}\\X\subseteq L^{(j)}}} \sum_{l\in X} \max\left(G_l(t),0\right)
$$

$$
\leq (c/k)\max_{X\in\mathcal{X}} \sum_{i\in X} G_i(t) \text{ (since } L^{(0)},\ldots,L^{(k-1)} \text{ are } c-\text{approximate).}
$$

The result follows. ∎

We now prove lemma 9. This proof follows from lemma 10 using techniques similar to those used by Tassiulas *et. al.* in [15].

*Proof:* Let the arrival rate vector be $(1-c/k)\vec{\lambda}$ where $\vec{\lambda} \in \text{Int}(\Lambda)$. Clearly, under $\pi^{\text{MH}}(k)$, $\vec{Q}(t)$ constitutes an aperiodic irreducible Markov chain, and $\mathbf{E}\left(\sum_i Q_i(t)G_i(t)|\vec{Q}(t)=\vec{Q}\right) \geq (1-c/k)\max_{X\in\mathcal{X}} \sum_{i\in X} G_i(t)$. We will consider the lyapunov function $V(\vec{Q}) = \sum_i Q_i^2$, and prove that under $\pi$, $\mathbf{E}\left(V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right)|\vec{Q}(t)=\vec{Q}\right) < -1$ for all sufficiently large $||\vec{Q}||$, where $||\vec{Q}|| = \sqrt{V(\vec{Q})}$. Then, from Foster's theorem (Theorem 2.2.3 in [5]) the Markov chain representing the queue length process is positive recurrent. Also, $\mathbf{E}(Q_i(t)) < \infty$ for each $i$ under the steady state distribution for the above Markov chain. Thus, $\lim_{K\to\infty} \frac{\sum_{t=0}^{K-1} Q_i(t)}{K} < \infty$. The result follows.

Let $\vec{I}^X$ denote the indicator vector for set $X \in \mathcal{X}$. Let $q(j)$ denote the session of session-link $j$. Let $\vec{f}$ be an $M$-dimensional vector such that $f_i = \lambda_{q(i)}$. Then, $\text{Int}(\Lambda)$ can be characterized as follows [15]:

$$
\text{Int}(\Lambda) = \{\vec{\lambda} : \vec{\lambda} = \sum_{\vec{X}\in\mathcal{X}} \beta_X R\vec{I}^X, \text{ where } \sum_{\vec{X}\in\mathcal{X}} \beta_X = 1 \text{ and } \beta_X \geq 0 \text{ for each } X \in \mathcal{X} \text{ and } \beta_\phi > 0\}. \quad (7)
$$

Now,

$$
\mathbf{E}\left(\left(\vec{A}(t)\right)^T \vec{Q}(t)/\vec{Q}(t)=\vec{Q}\right) = (1-c/k)\vec{\lambda}^T\vec{Q}
$$

$$
= (1-c/k)\sum_{X\in\mathcal{X}} \beta_X\left(\left(R\vec{I}^X\right)^T \vec{Q}\right) \text{ where } \beta_\phi > 0 \text{ (from (7))}
$$

$$
\leq (1-c/k)(1-\beta_\phi)\max_{X\in\mathcal{X}} \sum_{i\in X} G_i \quad (8)
$$

$$
\mathbf{E}\left(V\left(\vec{Q}(t+1)\right) - V\left(\vec{Q}(t)\right)|\vec{Q}(t)=\vec{Q}\right)
$$

$$
= \mathbf{E}\left(\left(\vec{A}(t)-R\vec{D}(t)\right)^T \left(2\vec{Q}(t)+\vec{A}(t)-R\vec{D}(t)\right)|\vec{Q}(t)=\vec{Q}\right)
$$

$$
\leq 2\mathbf{E}\left(\left(\vec{A}(t)-R\vec{D}(t)\right)^T \vec{Q}(t)|\vec{Q}(t)=\vec{Q}\right) + M\alpha_{\max}^2
$$

$$
\leq 2\mathbf{E}\left(\vec{A}^T(t)\vec{Q}(t)|\vec{Q}(t)=\vec{Q}\right) - 2\mathbf{E}\left(\sum_i G_i(t)D_i(t)|\vec{Q}(t)=\vec{Q}\right) + M\alpha_{\max}^2
$$

$$
\leq -2(1-c/k)\beta_\phi \max_{X\in\mathcal{X}} \sum_{i\in X} G_i + M\alpha_{\max}^2 \text{ (from Lemma 10 and (8))}
$$

$$
< -1 \text{ for all sufficiently large } ||\vec{Q}|| \text{ (since } \beta_\phi > 0, 0 < c < k).
$$

∎

## REFERENCES

[1] A. Brzezinski, G. Zussman, and E. Modiano. Distributed throughput maximization in wireless mesh networks - a partitioning approach. In *Proceedings of ACM MOBICOM*, Los Angeles, CA, September 2006.

[2] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in multihop wireless networks. In *Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, Allerton, Monticello, Illinois, September 28-30 2005.

[3] J. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proceedings of INFOCOM*, pages 556–564, Tel Aviv, Israel, Mar 2000.

[4] D. Shah E. Modiano and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proc. ACM SIGMETRICS / IFIP Performance'06*, June 2006.

[5] G. Fayolle, V. A. Malyshev, and M. V. Menshikov. *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge University Press, 1995.

[6] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910–917, Sep 1988.

[7] H. Hunt, M. Marathe, Radhakrishnan V, S. Ravi, D. Rosenkrantz, and R. Stearns. Nc-approximation schemes for NP- and PSPACE-HARD problems in geometric graphs. *Journal of Algorithms*, 26(2), February 1998.

[8] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Local approximation schemes for ad hoc and sensor networks. In *Proceedings of DIALM-POMC*, Cologne, Germany, September 2 2005.

[9] X. Lin and S. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. In *Proceedings of IEEE CDC-ECC'05*, San Diego, CA, Dec 2006.

[10] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proceedings of INFOCOM*, Miami, FL, Mar 2005.

[11] T. Nieberg, J. Hurink, and W. Kern. A robust ptas for maximum weight independent sets in unit disk graphs. In *Graph Theoretic Concepts in Computer Science*. Springer Berlin.

[12] S. Sarkar and K. Kar. Achieving 2/3 throughput approximation with sequential maximal scheduling under primary interference constraints. In *Proceedings of 44th Annual Allerton Conference on Communication, Control and Computing*, Allerton, Monticello, Illinois, September 27-29 2006.

[13] G. Sharma, R.R. Mazumdar, and N.B. Shroff. On the complexity of scheduling in wireless networks. In *Proceedings of ACM MOBICOM*, Los Angeles, CA, September 2006.

[14] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *Proceedings of INFOCOM*, pages 533–539, 1998.

[15] L. Tassiulas and A. Ephremidis. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, Dec 1992.

[16] X. Wu and R. Srikant. Regulated maximal matching: a distributed scheduling algorithm for multihop wireless networks with node-exclusive spectrum sharing. In *Proceedings of IEEE CDC-ECC'05*, Seville, Spain, Dec 2005.