Local Pooling Conditions for Joint Routing and Scheduling

Andrew Brzezinski
Fidelity Investments
Boston, MA 02210
Andrew.Brzezinski@fmr.com

Gil Zussman
Dept. of Electrical Engineering
Columbia University
New York, NY 10027
gil@ee.columbia.edu

Eytan Modiano
Massachusetts Institute of Technology
Cambridge, MA 02139
modiano@mit.edu

***Abstract*— A major challenge in the design and operation of wireless networks is to jointly route packets and schedule transmissions to efficiently share the common spectrum among links in the same area. Due to the lack of central control in wireless networks, these algorithms have to be decentralized. It was recently shown that distributed (greedy) algorithms can usually guarantee only fractional throughput. It was also recently shown that if a set of conditions regarding the network topology (known as Local Pooling) is satisfied, simple distributed maximal weight (greedy) scheduling algorithms achieve 100% throughput. In this paper, we focus on networks in which packets have to undergo multihop routing and derive multihop Local Pooling conditions for that setting. In networks satisfying these conditions, a backpressure-based joint routing and scheduling algorithm employing *maximal* weight scheduling achieves 100% throughput.**

I. INTRODUCTION

Efficient operation of wireless networks requires *distributed* joint routing and scheduling algorithms that take into account the interference between simultaneous transmissions. A *centralized* joint routing and scheduling framework that achieves the maximum attainable throughput region was presented by Tassiulas and Ephremides [18]. Recently, several *distributed* algorithms that can achieve only a fraction of the maximum throughput have been presented. Dimakis and Walrand [8] presented a set of conditions (termed as Local Pooling) which are related to the topology of the network. They showed that if these conditions hold, greedy *scheduling* algorithms, which can be implemented in a distributed manner, *achieve* 100% *throughput*. In this paper, we generalize the Local Pooling conditions and derive conditions under which a *greedy joint routing and scheduling* algorithm achieves 100% throughput. The algorithm is directly based on the on the centralized framework of [18] but can be implemented in a distributed manner.

Joint routing and scheduling in a slotted multihop wireless network with a stochastic packet arrival process was considered in [18]. The routing and link activation policy presented there guarantees to stabilize the network (i.e. provide 100% throughput) whenever the arrival rates are within the stability region.[1] The results of [18] have been extended to various settings of wireless networks and input-queued switches (e.g. [1], [14], [17], and references therein). However, optimal algorithms based on [18] require repeatedly solving a *global optimization problem*, taking into account the queue backlog information for every link in the network. For example, even for the simple primary interference constraints[2] a maximum weight matching problem has to be solved in every slot. Obtaining a centralized solution to such a problem in a wireless network does not seem to be feasible, due to the communication overhead associated with continuously collecting the queue backlog information. Therefore, the design of *distributed algorithms* has attracted a lot of attention recently.

Assuming that the traffic is exclusively single-hop reduces the joint problem to a *scheduling* problem. Regarding primary interference constraints, it has been shown that in this setting *distributed maximal matching* algorithms achieve 50% throughput [5], [13]. It was also proved in [5], [12], [20] that under secondary interference constraints[3] the throughput obtained by a distributed maximal scheduling algorithm may be significantly smaller than the throughput under a centralized (optimal) scheduler. In particular, in [5] it was proved that a distributed algorithm may achieve as low as $1/8$ of the possible throughput.

Dimakis and Walrand [8] recently showed that although in *arbitrary topologies* the worst case performance of distributed maximal scheduling algorithms can be very low, there are some topologies in which they *can achieve* 100% *throughput*. In particular, they consider a graph of interfering queues[4] and study the performance of a *greedy maximal weight scheduling* algorithm (termed Longest Queue First - LQF) that selects the set of served queues greedily according to the queue lengths. They present sufficient conditions for such an algorithm to provide 100% throughput (notice that unlike a *maximum* weight solution a *maximal* weight solution

[1]We note that the algorithm presented in [3] deals with a similar setting by using similar methodologies.

[2]Primary interference constraints imply that *each station can converse with at most a single neighbor at a time* (i.e. the set of active links at any point of time constitutes a matching).

[3]Secondary interference constraints imply that each pair of simultaneously active links must be separated by at least two hops (links). These constraints are usually used to model IEEE 802.11 networks [5], [20].

[4]A graph of interfering queues can be constructed from the network graph according to the interference constraints and is usually referred to as an interference or conflict graph [10].

can be easily obtained in a distributed manner [9]). These conditions are referred to as *Local Pooling* (LoP) and are related to the properties of all maximal independent sets in the conflict graph. The LoP conditions were recently generalized in [11] that provided conditions under which a greedy maximal weight matching algorithm obtains some guaranteed fractional throughput. Moreover, in [4], [21] several graph classes that satisfy the LoP conditions have been identified and the effect of multihop interference has been studied. For example, it has been shown in [21] that under any interference degree, tree network graphs yield interference graphs that satisfy LoP (i.e. under any interference degree, distributed algorithms achieve 100% throughput in trees). Moreover, an application of the LoP conditions to channel allocation in Wireless Mesh Networks has been demonstrated in [4].

In general, networking environments in which the traffic is inherently *single-hop* and where packets must depart the system upon transmission across a link are rare. This is results from the fact that many connections are necessarily multihop connections due to geographical and physical constraints on user connectivity. Networks with *multihop traffic*, where packets follow a fixed multihop path, have been studied by Wu and Srikant [19], who proposed the use of regulators along with a maximal matching scheduling algorithm. It was shown in [19] that under primary interference constraints, the throughput may be reduced to 50%. These results have been extended in [12], [20] where it was also pointed out that only a fraction of the throughput is attainable.

Since the LoP results of [8] have been derived for networks with single-hop traffic, it is desirable to identify topologies in which distributed algorithms can obtain 100% throughput in the multihop traffic setting. Therefore, in this paper we study the LoP properties of a distributed routing and scheduling framework which is based on the backpressure mechanism of [18]. In this framework the edge weights are obtained by the backpressure mechanism but unlike in [18], a *distributed* maximal weight scheduling algorithm is used to determine which edges should be activated in every time slot. We derive new Multihop Local Pooling conditions that are sufficient for guaranteeing that this distributed framework achieves 100% throughput. Similarly to the conditions of [8], the new conditions are based on the interference graph and are not limited to networks with primary interference constraints.

This paper is organized as follows. In Section II we present the network model, stability definitions, and the single-hop LoP conditions. In Section III we present a distributed adaptation of the backpressure-based framework of [18]. The new LoP conditions for networks with multihop traffic are presented in Section IV. We summarize the results and discuss future research directions in Section V.

II. Model and Preliminaries

A. Network Model

Consider a wireless network $G_N = (V_N, E_N)$, where $V_N = \{1, \ldots, n\}$ is the set of nodes, and $E_N \subseteq \{(i,j) : i,j \in V_N, i \neq j\}$ is a set of directed links indicating pairs of nodes between which data flows can occur, with $m \triangleq |E_N|$. The directionality of data flows across links necessitates the treatment of the network graph G_N as a directed graph. Depending on the circumstances, we denote links as either (i,j) or as e_k. In G_N, if two nodes $v_1, v_2 \in V_N$ are within communication range, then the directed edges $e_{12} = (v_1, v_2)$ and $e_{21} = (v_2, v_1)$ both belong to E_N. For a directed edge e, let $\sigma(e)$ denote the source (initial) vertex, and $\tau(e)$ denote the terminal (destination) vertex. Throughout this paper, bold symbols are associated with vectors and matrices.

The interference between network links can be summarized in an *interference graph* (or *conflict graph*) $G_I = (V_I, E_I)$ based on the network graph G_N [10]. We assign $V_I \triangleq E_N$. Thus, each edge e_k in the network graph is represented by a vertex v_k of the interference graph, and an edge (v_i, v_j) in the interference graph indicates a conflict between network graph links e_i and e_j (i.e. transmissions on e_i and e_j cannot take place simultaneously).[5]

Let $\Pi(G_N)$ denote the set of available link activations in the network graph G_N: the vector $\boldsymbol{\pi} = (\pi_e, e \in E_N) \in \Pi(G_N)$ is a 0-1 column vector representing a possible link activation. The set $\Pi(G_N)$ corresponds to all possible independent sets in the interference graph $G_I = (V_I, E_I)$. Under primary interference, $\Pi(G_N)$ corresponds to the set of matchings in G_N. We denote by $\mathbf{M}(V_I)$ the matrix of *maximal* independent sets in G_I; that is, the set of maximal column vectors in $\Pi(G_N)$.

For simplicity, we assume that time is slotted and that packets are of equal size, each packet requiring one time slot of service across any link. There is no self-traffic. We will refer to packets destined to node $j \in V_N$ as *commodity j packets*. Let $A_{ij}(t)$ denote the number of exogenous commodity j packets that arrived at node i by the end of slot t. We assume that the arrivals have long term rates $\lambda_{ij} = \lim_{t\to\infty} A_{ij}(t)/t$, with overall system arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij},\, i,j \in V_N)$.

Every node is assumed to have a queue for each possible destination. For $i,j \in V_N$, let $Q_{ij}(t)$ be the number of packets enqueued at node i at time t, whose destination in the network is node j. Assume that $Q_{ij}(0) = 0$ for all i,j. The *differential backlog (backpressure)* of commodity j packets across edge $e \in E_N$ at time t is $Z_{ej}(t) = Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t)$, and the *maximum backpressure* is $Z^*_e(t) = \max_{j \in V_N} Z_{ej}(t)$.

Service is applied to the system at each time slot by activating a set of edges, and routing a packet of a single commodity across each active edge. We denote the corresponding $m \times n$ *service activation matrix* by $\mathbf{S} = (S_{ej}, e \in E_N, j \in V_N)$. Here, for edge $e \in E_N$ and commodity $j \in V_N$, S_{ej} can have value 0 or 1, depending on whether e is inactive or active for serving commodity j, respectively. Note that an admissible service activation matrix must have a valid underlying link activation belonging to $\Pi(G_N)$. This property characterizes

[5] Although it has been recently shown that in some cases the conflict graph does not fully capture the wireless interference characteristics [16], it still provides a reasonable abstraction. Extending the results to general SINR-based constraints (e.g. [17]) is a subject for further research.

the set of admissible service activation matrices, $\mathcal{S}$:

$$\mathcal{S} = \left\{ \mathbf{S} \in \{0,1\}^{m\times n} : \pi_e = \sum_{j\in V_N} S_{ej}, \boldsymbol{\pi} \in \Pi(G_N) \right\}.$$

The matrix $\mathbf{S} \in \mathcal{S}$ leads to packet transitions through the network. To model the queue evolution implied by invoking $\mathbf{S}$, we introduce for each commodity $j \in V_N$ the $n \times m$ *routing matrix* $\mathbf{R}^j = (R^j_{ie}, i \in V_N, e \in E_N)$, where:

$$R^j_{ie} = \begin{cases} 1, & \text{if } \sigma(e) = i \\ -1, & \text{if } \tau(e) = i \text{ and } i \neq j \\ 0, & \text{else} \end{cases}$$

Denote by $d_{ij}(\mathbf{S})$ the *net amount of service*, in number of packets per time slot, to queue Q_{ij} under activation matrix $\mathbf{S}$. Using the above routing matrix we can express $d_{ij}(\mathbf{S}) = \sum_l R^j_{il} S_{lj}$.

Denote by $D_{ij}(t)$ the total service applied to commodity j packets at node i up to the end of time-slot t. Finally, for each $\mathbf{S} \in \mathcal{S}$, denote by $F_{\mathbf{S}}(t)$ the number of time slots up to the end of time-slot t in which service activation matrix $\mathbf{S} \in \mathcal{S}$ was active. The following are the system dynamics for $t \geq 0$.

$$\begin{aligned} Q_{ij}(t) &= A_{ij}(t) - D_{ij}(t) \quad \forall (i,j) \\ D_{ij}(t) &= \sum_{\mathbf{S}\in\mathcal{S}} d_{ij}(\mathbf{S}) F_{\mathbf{S}}(t) \quad \forall (i,j) \\ \sum_{\mathbf{S}\in\mathcal{S}} F_{\mathbf{S}}(t) &= t, \text{ and } F_{\mathbf{S}} \text{ is non-decreasing} \end{aligned}$$

B. Stability Considerations

We can now define the stability region of the network.

Definition 1 (Admissible Rate Vector): A non-negative arrival rate vector $\boldsymbol{\lambda}$ is admissible if there exists a collection of service activation matrices $\mathbf{S}^l \in \mathcal{S}$, $1 \leq l \leq L$ such that

$$\lambda_{ij} \leq \sum_{l=1}^{L} \alpha_l d_{ij}(\mathbf{S}^l), \text{ where } \alpha_l \geq 0 \,\forall l, \sum_{l=1}^{L} \alpha_l \leq 1.$$

The set of all admissible rate vectors is called the stability region and is denoted by $\boldsymbol{\Lambda}^*$.

At each time slot, a joint scheduling and routing algorithm makes a link activation and routing decision that must satisfy the interference constraints. A stable algorithm, which we also refer to as a throughput optimal algorithm or an algorithm that achieves 100% throughput, is defined as follows.

Definition 2 (Stable Algorithm): An algorithm is stable if for any arrival process with rate vector $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$, $\lim_{t\to\infty} Q_{ij}(t)/t = 0$ with probability 1 $\forall i, j \in V_N$.

This stability criterion is termed *rate stability* [1]. Tassiulas and Ephremides developed a stable joint scheduling and routing algorithm that applies in this setting [18]. At time $t \geq 0$, their algorithm calculates for each edge $e \in E_N$ the maximum backpressure, which we express in vector form as $\mathbf{Z}^*(t) = (Z^*_e(t), e \in E_N)$. Their algorithm then selects a link activation vector

$$\boldsymbol{\pi}^*(t) \in \underset{\boldsymbol{\pi}\in\Pi(G_N)}{\arg\max}\, \boldsymbol{\pi}^T \mathbf{Z}^*(t). \tag{1}$$

Routing is carried out over each edge e having $\pi^*_e(t) = 1$, by serving a commodity $j \in \arg\max_j Z_{e,j}(t)$ across that edge (for more details regarding the algorithm of [18], see Section III).

For general interference graph G_I, the algorithm of [18] must find the *maximum weight independent set* in G_I at each time slot[6] to obtain an optimal solution to (1). Namely, it must solve an NP-hard problem in every time slot or time frame. Under primary interference, the optimization is simpler and the algorithm has to schedule the edges of a *maximum weight matching* in the network graph at each slot. This requires $O(n^3)$ computation time, using a centralized algorithm. In wireless networks, implementing a centralized algorithm is often not feasible and simple distributed algorithms usually obtain an approximate solution, resulting in a fractional throughput.

C. Local Pooling for Single Hop Traffic

We briefly reproduce important definitions and implications of Local Pooling (LoP) in networks with single-hop traffic, presented in [4], [8]. In Section IV we will introduce the LoP conditions for the *multihop* traffic case. Recall that $\mathbf{M}(V_I)$ is the collection of maximal independent vertex sets on G_I, organized as a matrix. We designate by $\mathbf{e}$ the vector having each entry equal to unity. We deliberately avoid specifying its size, because it will be obvious by the context of its use.

Definition 3 (Subgraph Local Pooling - SLoP): An interference graph G_I satisfies SLoP, if there exists nonzero $\boldsymbol{\alpha} \in \mathbb{R}_+^{|V_I|}$ and $c > 0$ such that $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = c\mathbf{e}^T$.

Definition 4 (Overall Local Pooling - OLoP): An interference graph G_I satisfies OLoP, if each induced subgraph over the nodes $V \subseteq V_I$ *satisfies SLoP.*

We can now describe the stability of the system when the service in each time slot is scheduled according to the Maximal Weight Independent Set (MWIS) algorithm. This algorithm is an iterative greedy algorithm that selects the node of G_I with the longest corresponding queue, and removes it and its neighbors from the interference graph. This process is repeated successively until no nodes remain. When multiple queues have the same length, a tie-breaking rule is applied. The set of selected nodes is a maximal independent set in the interference graph. Such a greedy algorithm can be implemented in a distributed manner and has the following property.

Theorem 1 (Dimakis and Walrand, 2006 [8]): If interference graph G_I satisfies OLoP, a Maximal Weight Independent Set (MWIS) scheduling algorithm achieves 100% *throughput.*

III. Backpressure-based Routing and Scheduling

In this section, we present a simple adaptation to the backpressure framework of [18] that allows a distributed implementation. Recall from Section II-B that the optimal centralized scheduler (1) calculates *maximum* weight independent sets based on backpressure link weights. Instead, the presented algorithm employs *maximal* weighted independent sets based on the backpressure link weights. Similarly to the single-hop

[6]In fact, it can be shown that throughput optimality is maintained when solutions are obtained at bounded time intervals that are longer than a time slot (e.g. [15]).

traffic setting [8], we use a *Maximal* Weight Independent Set (MWIS) algorithm but unlike in [8] we use the backpressure link weights (instead of the queue backlogs). The MWIS algorithm operates on the interference graph and since it is a greedy algorithm, it can be easily implemented in a distributed manner (e.g. the algorithm of [9] that can be applied to a network with primary interference constraints). As in the single-hop case and as in [18], the algorithm is *independent of the global network topology and traffic statistics*.

Algorithm 1 Backpressure Routing and (Maximal) Scheduling (BRMS)

1: **for** time index $t = 1, 2, \ldots$ **do**
2: For each directed edge $e \in E_N$ assign $Z_{ej}(t) \leftarrow (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t))$
3: Assign $Z_e^*(t) = \max_j Z_{ej}(t)$
4: Obtain a maximal link activation $\boldsymbol{\pi}^*(t) \in \Pi(G_N)$ using a decentralized MWIS algorithm, based on the edge weight vector $\mathbf{Z}^*(t) = (Z_e^*(t), e \in E_N)$
5: For each $e \in E_N$ such that $\pi_e^*(t) = 1$, choose $j^* \in \arg\max_j Z_{ej}(t)$. Route $\min\{1, Q_{\sigma(e)j^*}(t)\}$ packets of commodity j^* across e
6: **end for**

In step 4, the BRMS algorithm uses the MWIS algorithm to select a *maximal* weight link activation based upon maximum link backpressures, obtained in step 3 (notice that this is the main difference from [18]). In step 5, the BRMS algorithm makes routing decisions to service commodities achieving maximum backpressure.

IV. LoP in Networks with Multihop Routing

In this section, we study LoP properties in networks employing the algorithm described above. We derive new local pooling conditions that are sufficient for guaranteeing that the BRMS algorithm achieves 100% throughput in the multihop traffic environment.

A. Towards Multihop LoP Conditions

Recall that the OLoP conditions consider all possible vertex subsets of the interference graph, $V \subseteq V_I$. By the definition of the interference graph, the node set V corresponds to a subset of the network graph edges, $E \subseteq E_N$. Thus, the OLoP conditions effectively consider every subset of network graph edges $E \subseteq E_N$. In the multihop routing scenario, we must again consider each set of network graph edges $E \subseteq E_N$. Since routing across network graph edges is not unique in the multihop scenario, we must *additionally* consider various combinations of commodities associated with network graph edges. We formalize the possible edge/commodity combinations by introducing the Maximum Commodity Family (an example is given in Section IV-B).

Definition 5: [Maximum Commodity Family] *For $E \subseteq E_N$, $E \neq \emptyset$, the Maximum Commodity Family is given by* $\mathcal{J}_E = \{(J_e^{\mathbf{Q}}, e \in E_N) : \mathbf{Q} \in \mathcal{Q}_E, \mathbf{Q} \neq 0\}$, *where*

$$\begin{aligned}\mathcal{Q}_E = \{&(Q_{ij},\, i,j \in V_N, i \neq j) : Q_{ij} \in \mathbb{R}_+ \,\forall i,j,\\ &E = \arg\max_e \max_j (Q_{\sigma(e)j} - Q_{\tau(e)j})\},\\ J_e^{\mathbf{Q}} = \{&j \in V_N : j \neq \sigma(e),\\ &Q_{\sigma(e)j} - Q_{\tau(e)j} \geq Q_{\sigma(e)j'} - Q_{\tau(e)j'} \;\forall j' \in V_N\}.\end{aligned}$$

The Maximum Commodity Family $\mathcal{J}_E$ relates closely to a system of differential equations called a *fluid limit model* [6], derived from the queueing system. In order to better understand the Maximum Commodity Family, we next explore some of its properties. To this end, we introduce for each commodity $j \in V_N$ the directed *commodity graph* $G_j = (V_N, E_j)$, where $E_j = \{e \in E : j \in J_e\}$.

Lemma 1: For $E \subseteq E_N$, $E \neq \emptyset$, the commodity collection $J = (J_e, e \in E_N) \in \mathcal{J}_E$ satisfies:

1) $J_e \neq \emptyset$, $\forall e \in E_N$.
2) $J_e \subseteq V_N \setminus \{\sigma(e)\}$.
3) *For $j \in \cup_{e \in E} J_e$, G_j has no directed cycles.*
4) *If G_j has a directed path between vertices $v_1, v_2 \in V_N$ of length L, then*
   a) *the minimum length path between v_1 and v_2 in the network graph G_N is L, and*
   b) *the edges of all paths in G_N between v_1 and v_2 of length L are in G_j.*
5) *If G_j has a path of length L originating at vertex v, then*
   a) *G_N has no paths of length less than L originating at vertex v and terminating at vertex j, and*
   b) *the edges of all paths of length L in G_N, originating at vertex v and terminating at vertex j belong to G_j.*

Proof: See Appendix A. ∎

Under the BRMS algorithm, when the set of directed edges $E \subseteq E_N$ have backpressures exceeding those of the other edges in the graph, there must exist a commodity collection $(J_e, e \in E_N) \in \mathcal{J}_E$ for which J_e is the set of commodities maximizing differential backlog across $e \in E_N$. In this case, a MWIS algorithm must select a link activation $\boldsymbol{\pi}^*$ that is maximal among the edges in E: i.e. $\boldsymbol{\pi}_E^* \in \mathbf{M}(E)$. Additionally, the commodity j that is routed across edge $e \in E_N$ must belong to J_e. These properties characterize the Maximal Service Activation Set (an example is given in Section IV-B):

Definition 6 (Maximal Service Activation Set): For $E \subseteq E_N$ and $J = (J_e, e \in E_N) \in \mathcal{J}_E$, the Maximal Service Activation Set is given by

$$\mathcal{S}_{E,J} = \big\{\mathbf{S} \in \mathcal{S} : \textstyle\sum_j \mathbf{S}_{Ej} \in \mathbf{M}(E),\; S_{ej} = 1 \text{ implies } j \in J_e \text{ when } e \in E_N\big\}$$

Above, $\mathbf{S}_{Ej}$ is the vector $(S_{ej}, e \in E)$. The Maximal Service Activation Set $\mathcal{S}_{E,J}$ for a set of edges $E \in E_N$ consists of every service activation matrix whose underlying link activation is maximal over the edges in E. Recall that each edge $e \in E_N$ is a vertex in the interference graph G_I.

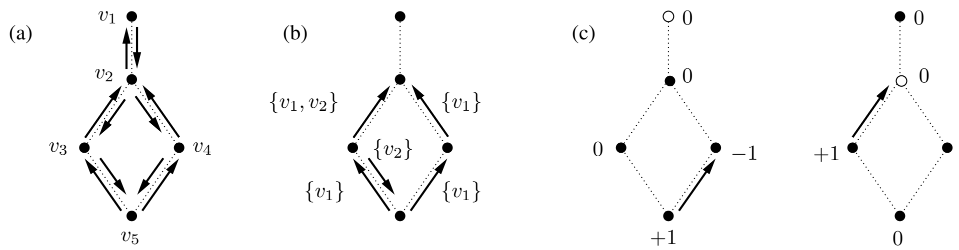

Fig. 1. (a) Network graph G_N, (b) the subset E of network graph edges, with corresponding commodity sets labeled at each edge, and (c) commodity graphs G_{v_1} (left) and G_{v_2} (right) for a particular maximal service activation.

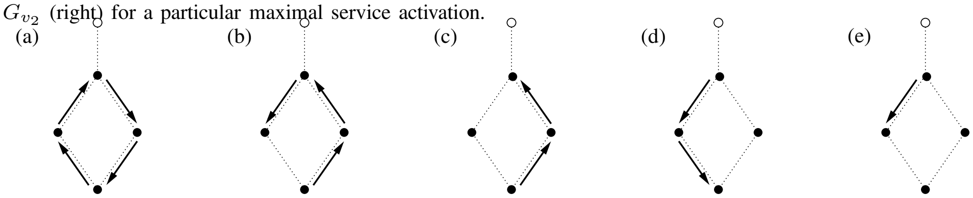

Fig. 2. Commodity graphs for commodity v_1, that are invalid based on the properties of Lemma 1.

In order to characterize the stability properties of the BRMS algorithm, we will track the dynamics of the link differential backlogs. Hence, we must understand how each service matrix $\mathbf{S} \in \mathcal{S}$ affects the distribution of commodity backpressures over the network links. We next introduce the Backpressure Service Vector.

Definition 7 (Backpressure Service Vector): For $E \subseteq E_N$, $J = (J_e, e \in E_N) \in \mathcal{J}_E$, *and service matrix* $\mathbf{S} \in \mathcal{S}$, *the Backpressure Service Vector* $\mathbf{u}_{E,J}(\mathbf{S})$ *contains the* decrease *in differential backlog of commodity* j *across link* e *under service matrix* $\mathbf{S}$ *for every edge/commodity pair* (e, j) *where* $e \in E, j \in J_e$*:*

$$\mathbf{u}_{E,J}(\mathbf{S}) = ((d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})), e \in E, j \in J_e).$$

The Backpressure Service Vector $\mathbf{u}_{E,J}(\mathbf{S})$ tracks the change in backpressure incurred by a set of edge/commodity pairs, when a particular service activation matrix $\mathbf{S}$ is employed for a single time slot.

B. Examples

In this section, we consider the network graph G_N of Fig. 1(a), with the convention that the directed edge from node v_i to v_j is labeled e_{ij}.

We begin by considering a specific feasible combination of edges and commodities. The subset E of network edges of interest is $E = \{e_{32}, e_{35}, e_{42}, e_{53}, e_{54}\}$, as depicted in Fig. 1(b). Each edge in E has associated with it a set of commodities: $J_{e_{32}} = \{v_1, v_2\}$, $J_{e_{35}} = \{v_2\}$, $J_{e_{42}} = \{v_1\}$, $J_{e_{53}} = \{v_1\}$, $J_{e_{54}} = \{v_1\}$. These commodity sets are elements of commodity collection $J = (J_e, e \in E_N)$. This collection is a member of the Maximum Commodity Family.

Assuming primary interference constraints, the Maximal Service Activation Set $\mathcal{S}_{E,J}$ is summarized by the following table of valid edge/commodity pairs. For example, activation (e_{32}, v_1) means that commodity v_1 is sent over link e_{32}. Additionally, each activation $\mathbf{S}$ is translated in the table below to backpressure service vectors $\mathbf{u}_{E,J}(\mathbf{S})$. The service vectors are ordered by (link, commodity) pairs as follows: $(e_{32}, v_1), (e_{42}, v_1), (e_{53}, v_1), (e_{54}, v_1), (e_{32}, v_2), (e_{35}, v_2)$.

Service activation $\mathbf{S}$	Backpressure service vector $\mathbf{u}_{E,J}(\mathbf{S})$
$\{(e_{32}, v_1), (e_{54}, v_1)\}$	$(2, 0, 0, 2, 0, 0)$
$\{(e_{42}, v_1), (e_{53}, v_1)\}$	$(0, 2, 2, 0, 0, 0)$
$\{(e_{32}, v_2), (e_{54}, v_1)\}$	$(0, -1, 1, 2, 1, 1)$
$\{(e_{35}, v_2), (e_{42}, v_1)\}$	$(1, 2, 0, -1, 1, 2)$

Consider the third service activation from the table, which activates edge e_{32} for service of commodity v_2, and edge e_{54} for service of commodity v_1. We have depicted in Fig. 1(c) the active link for serving commodity v_1 packets in the graph on the left, and the active edge for serving commodity v_2 packets in the graph on the right. At each node of the graph, we indicate the number of packets *departed* from that node under that service activation. The backpressure service for each edge/commodity combination (e, j), where $e \in E$ and $j \in J_e$, is then obtained by calculating on the graph corresponding to commodity j the difference between the quantity indicated at the source node of e and that indicated at the destination node of e. Edge e_{54} has a $+1$ at its source and a -1 at its destination in the graph for commodity v_1, which indicates a backpressure service of 2 commodity v_1 packets. Through similar computation, we find that edge e_{32} sees a backpressure service of 1 commodity v_2 packet. Note that although no other edge is active, some inactive edges do incur service under this service activation: edge e_{53} sees a backpressure service of 1 commodity v_1 packet, while edge e_{42} sees an *increase* of commodity v_1 backpressure of 1 packet (this implies -1 units of backpressure service). Finally, edge e_{35} sees a service of 1 commodity v_2 packet. No other edge/commodity pairs (e, j) where $e \in E$ and $j \in J_e$, see service. Thus, we have determined each entry in the backpressure service vector corresponding to this particular service activation.

We next provide examples to illustrate the properties of Lemma 1. Figs. 2(a)-2(e) show graphs that are inadmissible as the commodity v_1 graph, G_{v_1}, for the network graph depicted in Fig. 1(a) (the indices of the vertices in these examples are according to Fig. 1(a)). Fig. 2(a) fails Property 3 because G_{v_1} contains a directed cycle; Fig. 2(b) fails Property 4a since edge e_{53} provides a shorter path between vertices v_5, v_3; Fig. 2(c) fails Property 4b since edges e_{53}, e_{32} are not included in G_{v_1}; Fig. 2(d) fails Property 5a since the path $v_2 \rightarrow v_3 \rightarrow v_5$ belongs to G_{v_1}, while path $v_2 \rightarrow v_1$ belongs to G_N; and Fig. 2(e) fails Property 5b since edge e_{21} does not belong to G_{v_1}.

C. Stability of the Backpressure-based Algorithm

We now introduce the multihop LoP definitions, and prove a sufficient condition for stability of a network operated according to the BRMS algorithm, based on these conditions. Recall that the quantity $d_{ij}(\mathbf{S})$ is the amount of service at queue Q_{ij} resulting from applying service activation $\mathbf{S}$ for one time slot. Denote vector $\mathbf{d}(\mathbf{S}) = (d_{ij}(\mathbf{S}),\, i,j \in V_N)$.

Definition 8 (Subgraph Multihop LoP - SMLoP): The directed network graph $G = (V, E)$ with commodity collection $J \in \mathcal{J}_E$ satisfies SMLoP if there exist vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$ with $\boldsymbol{\alpha} \neq 0$, and a constant $c \geq 0$ such that

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c, \quad \forall \mathbf{S} \in \mathcal{S}, \tag{2}$$

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq c, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J}. \tag{3}$$

The SMLoP conditions associate with each link/commodity pair (e, j) a non-negative weight α_{ej}, where $e \in E, j \in J_e$. Further, for each node/commodity pair (v, j), the conditions associate a non-negative weight β_{vj}, where $v, j \in V_N$.

Definition 9 (Overall Multihop LoP - OMLoP): The network graph $G_N = (V_N, E_N)$ satisfies OMLoP if SMLoP is satisfied by each subgraph $G'_N = (V_N, E)$ with commodity collection $J \in \mathcal{J}_E$, where $E \subseteq E_N$.

We next state the main theorem regarding stability of the BRMS algorithm.

Theorem 2: If network graph G_N satisfies OMLoP, then the MWIS scheduling and routing algorithm achieves 100% throughput.

Proof: See Appendix B. ∎

V. Conclusions

We have derived new multihop Local Pooling conditions (OMLoP) that are sufficient for the stability of the backpressure-based joint routing and scheduling algorithm (BRMS) that makes *maximal* weight link activation decisions. Namely, in network graphs that satisfy these conditions, the BRMS algorithm archives 100% throughput. In [21] we have made some preliminary attempts to identify graphs that satisfy these conditions.

There are still several open problems in this area. For example, the complete characterization of the graph classes that satisfy the OMLoP conditions is a subject for further research. Moreover, deriving similar conditions for other joint routing and scheduling algorithms and studying the effect of generalizing the interference model from an interference graph model to a model based on SINR remain subjects for future research.

Acknowledgement

This work was supported by NSF ITR grant CCR-0325401, by NSF grant CNS-0626781, by ONR grant number N000140610064, and by a Marie Curie International Fellowship within the 6th European Community Framework Programme.

Appendix A
Proof of Lemma 1

Let $E \subseteq E_N$, with $E \neq \emptyset$. Consider any $J_E \in \mathcal{J}_E$, and suppose $J_E = (J_e^{\mathbf{Q}}, e \in E_N)$ for $\mathbf{Q} \in \mathcal{Q}_E$. Item 1 follows because the set $J_e^{\mathbf{Q}}$ can never be empty. Item 2 follows by the definition of $J_e^{\mathbf{Q}}$. For Item 3, suppose that graph G_j contains a directed cycle, $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_L \rightarrow v_1$. Then since $\mathbf{Q} \in \mathcal{Q}_E$, it must be true that $Q_{v_i j}$ strictly decreases across each edge in the cycle. This is clearly a contradiction. For Item 4a, suppose vertices v_1, v_2 are joined by a path of length L in G_j, and there exists a shorter path between v_1, v_2 in G_N. Then there must exist an edge e on this shorter path for which $Q_{\sigma(e)j} - Q_{\tau(e)j}$ exceeds the corresponding value across edges in the path joining v_1, v_2 in G_j. This violates that $\mathbf{Q} \in \mathcal{Q}_E$, which provides a contradiction. Item 4b follows similarly: suppose there exist two paths of length L in G_N, with every edge in the first path belonging to G_j. By definition, every edge e in the first path must have equal values $Q_{\sigma(e)j} - Q_{\tau(e)j}$. If this is not the case for the second path, then there must exist some edge e' whose corresponding value exceeds that of the edges in the first path. This violates that $\mathbf{Q} \in \mathcal{Q}_E$, which provides a contradiction. Item 5a follows by noting that $Q_{jj} = 0$, which implies that the differential backlog of commodity j along at least one edge on the shortest path from v to j exceeds that of the edges along the path of length L originating at v. This contradicts the set E. Item 5b follows similarly.

Appendix B
Proof of Theorem 2

The proof of stability makes use of the *fluid limit* technique. We consider a countably infinite sequence of queueing systems, indexed by r, subject to the same arrival process, $A_{ij}(t), i,j \in \{1,\ldots,n\}$, for $t \geq 0$. The queueing variables of the r-th system are given by $Q_{ij}^r(t), A_{ij}^r(t) = A_{ij}(t), U_{ij}^r(t)$ for all $i,j \in \{1,\ldots,n\}$, and $F_{\mathbf{S}}^r(t)$ for all $\mathbf{S} \in \mathcal{S}$. At time $t = 0$, the r-th system is assumed to contain zero packets in every queue. The following are the queue evolution properties of the r-th system:

$$Q_{ij}^r(t) = A_{ij}^r(t) - U_{ij}^r(t), \quad t \geq 0$$

$$U_{ij}^r(t) = \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}) F_{\mathbf{S}}^r(t), \quad t \geq 0$$

$$\sum_{\mathbf{S} \in \mathcal{S}} F_{\mathbf{S}}^r(t) = t, \text{ and } F_{\mathbf{S}} \text{ is non-decreasing}, \quad t \geq 0$$

$$A_{ij}^r(0) = 0, U_{ij}^r(0) = 0, \forall i,j,\ F_{\mathbf{S}}^r(0) = 0, \forall \mathbf{S} \in \mathcal{S}$$

We extend the queueing variables to the reals using $Y(t) = Y(\lfloor t \rfloor)$ for $Y = Q_{ij}^r, A_{ij}^r, U_{ij}^r, F_{\mathbf{S}}^r$. Now each of these processes is scaled according to $q_{ij}^r(t) = Q_{ij}^r(rt)/r$. We obtain the scaled processes $q_{ij}^r, a_{ij}^r, u_{ij}^r, f_{\mathbf{S}}^r$. As in [2], we can infer the convergence with probability 1 of the scaled processes over some subsequence of system indices $\{r_k\}$ to a *fluid limit* $(q_{ij}, a_{ij}, u_{ij}, f_{\mathbf{S}})$ having the following key properties:

$$\begin{aligned} q_{ij}(t) &= a_{ij}(t) - u_{ij}(t), \quad t \geq 0 \\ a_{ij}(t) &= \lambda_{ij} t, \quad t \geq 0 \\ u_{ij}(t) &= \sum_{\mathbf{S} \in \mathcal{S}} d_{ij}(\mathbf{S}) f_{\mathbf{S}}(t), \quad t \geq 0 \\ \sum_{\mathbf{S} \in \mathcal{S}} f_{\mathbf{S}}(t) &= t, \text{ and } f_{\mathbf{S}} \text{ is non-decreasing}, \quad t \geq 0 \\ a_{ij}(0) = 0, u_{ij}(0) &= 0, \forall i, j, \ f_{\mathbf{S}}(0) = 0, \forall \mathbf{S} \in \mathcal{S} \end{aligned}$$

The convergence of each process is uniform on compact sets for $t \geq 0$, and it easily follows that the limiting processes $q_{ij}, a_{ij}, u_{ij}, f_{\mathbf{S}}$ are Lipschitz-continuous in $[0, \infty)$.

Consider $z_{ej}(t) = q_{\sigma(e)j}(t) - q_{\tau(e)j}(t)$, the fluid *differential backlog* of commodity j across the directed link e. Define the function $h : [0, \infty) \to [0, \infty)$ where $h(t) = \max_{e,j} z_{ej}(t)$. Consider a regular time[7] $t \geq 0$, at which $h(t) > 0$. Assign

$$E = \{e \in E_N : \exists j \text{ such that } z_{ej}(t) = h(t)\}, \tag{4}$$

and for $e \in E_N$, assign $J_e = \arg\max_j z_{ej}(t)$. Note that using $\mathbf{Q} = (q_{ij}(t), i, j \in V_N)$ in conjunction with Definition 5, we have $J \triangleq (J_e, e \in E_N) \in \mathcal{J}_E$. Under the backpressure-based algorithm, it is simple to demonstrate that no link activation outside of $\mathcal{S}_{E,J}$ can have an increasing value $f_{\mathbf{S}}(t)$. Thus we have,

$$\sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) = 1$$

where $\dot{f}_{\mathbf{S}}(t)$ is the derivative of $f_{\mathbf{S}}(t)$. Assuming an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij}, i, j \in V_N)$, we have for $e \in E$ and $j \in J_e$,

$$\begin{aligned} &\dot{z}_{e,j}(t) = \\ &\lambda_{\sigma(e)j} - \lambda_{\tau(e)j} - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})) = \\ &\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}}(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})) - \\ &\sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})) = \\ &\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} u_{ej}(\mathbf{S}) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{ej}(\mathbf{S}) \end{aligned} \tag{5}$$

for some $\boldsymbol{\phi} = (\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ satisfying $\phi_{\mathbf{S}} \geq 0$, $\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} \leq 1$. The following lemma provides a condition under which the fluid differential backlogs are guaranteed to be *non-increasing* at any regular time. Recall our notation that $\mathbf{e}$ denotes the all-ones vector.

Lemma B.1: Let $t \geq 0$ be a regular time at which $h(t) > 0$. Let $E \subseteq E_N$ satisfy (4) and $J_e = \arg\max_j z_{ej}(t)$ for each $e \in E_N$. Suppose that the solution θ^* to the following optimization problem is $\theta^* \leq 0$:

$$\begin{aligned} \text{Maximize} \quad & \theta & (6) \\ \text{Subject to} \quad & \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) \geq \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \nu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) + \theta \mathbf{e} & \\ & \mathbf{e}^T \boldsymbol{\mu} \leq 1 & \\ & \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \sum_e R_{ie}^j S_{ej} \geq 0 \quad i, j = 1, \ldots, n & (7) \\ & \mathbf{e}^T \boldsymbol{\nu} = 1 & (8) \\ & \mu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S} & \\ & \nu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} & (9) \end{aligned}$$

Then $\dot{h}(t) \leq 0$.

Proof: Suppose $\theta^* \leq 0$. For an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij}, i, j \in V_N)$, we have $\lambda_{ij} = \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} d_{ij}(\mathbf{S}) \geq 0$, where $\phi_{\mathbf{S}} \geq 0 \forall \mathbf{S}$, and $\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} \leq 1$. Furthermore, $\sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) = 1$ and $\dot{f}_{\mathbf{S}}(t) \geq 0 \forall \mathbf{S}$. Thus, the vectors $(\phi_{\mathbf{S}}, \mathbf{S} \in \mathcal{S})$ and $(f_{\mathbf{S}}(t), \mathbf{S} \in \mathcal{S}_{E,J})$ are feasible as vectors $\boldsymbol{\mu}, \boldsymbol{\nu}$ respectively, in the linear program (6). The solution $\theta^* \leq 0$ in the optimization clearly implies that there must exist $e \in E$ and $j \in J_e$ such that

$$\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} u_{ej}(\mathbf{S}) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{ej}(\mathbf{S}) \leq 0. \tag{10}$$

By (5), equation (10) implies that $\dot{z}_{ej}(t) \leq 0$. Since t is a regular time, $\dot{z}_{ej}(t) = \dot{h}(t)$, which provides $\dot{h}(t) \leq 0$, as desired. ■

It only remains to demonstrate that the multihop local pooling conditions (2)-(3) are sufficient for stability. The following lemma demonstrates this property by studying the dual optimization problem to that in (6).

Lemma B.2: Consider graph $G = (V_N, E)$, where $E \subseteq E_N$. Then G satisfies SMLoP under commodity collection $J \in \mathcal{J}_E$ if and only if the corresponding optimization problem (6) has solution $\theta^* \leq 0$.

Proof: Suppose that the optimization (6) has solution $\theta^* \leq 0$. This implies that there exists a dual solution and complementary slackness conditions hold. It is simple to demonstrate that the dual problem to (6) is:

$$\begin{aligned} \text{Minimize} \quad & c_1 + c_2 & (11) \\ \text{Subject to} \quad & \boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c_1, \quad \forall \mathbf{S} \in \mathcal{S} & \\ & \boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq -c_2, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} & \\ & \mathbf{e}^T \boldsymbol{\alpha} = 1 & \\ & \boldsymbol{\alpha}, \boldsymbol{\beta}, c_1 \geq 0 & \end{aligned}$$

Since the solution to (6) is $\theta^* \leq 0$, the dual solution is attained at the point $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, c_1^*, c_2^*)$, where $c_1^* + c_2^* \leq 0$. Then the values $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*, \boldsymbol{\beta} = \boldsymbol{\beta}^*, c = c_1^*$ satisfy the SMLoP conditions, as desired.

Conversely, suppose that the SMLoP conditions are satisfied, with values $(\boldsymbol{\alpha}, \boldsymbol{\beta}, c) \geq 0$, where $\boldsymbol{\alpha} \neq 0$. Then, the point $(\boldsymbol{\alpha}/(\mathbf{e}^T \boldsymbol{\alpha}), \boldsymbol{\beta}, c, -c)$ is a feasible point in the dual optimization problem (11). This feasible point has cost 0. By

[7] A regular time is a point at which the system is differentiable. By the Lipschitz continuity of the fluid limit, almost every time in $[0, \infty)$ is regular.

duality, this implies that the primal problem must attain a solution $\theta^* \leq 0$, as desired. ■

Combining Lemmas B.1 and B.2, we conclude that if SMLoP is satisfied for any $E \subseteq E_N$, with any commodity collection $J \in \mathcal{J}_E$, then $\dot{h}(t) \leq 0$ for any regular time t at which $h(t) > 0$. Noting that $h(0) = 0$, and applying [7, Lemma 1], Lemma B.1 allows us to conclude that $h(t) = 0$ for almost every $t \geq 0$. This immediately implies that $q_{ij}(t) = 0$ for almost every $t \geq 0$, which gives the rate stability of the BRMS algorithm. Thus the OMLoP conditions are sufficient for stability, as desired.

References

[1] M. Ajmone Marsan, E. Leonardi, M. Mellia, and F. Neri, "On the stability of isolated and interconnected input-queueing switches under multiclass traffic," *IEEE Trans. Inform. Th.*, vol. 51, no. 3, pp. 1167–1174, Mar. 2005.

[2] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences*, vol. 18, pp. 191–217, 2004.

[3] B. Awerbuch and T. Leighton, "Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks," in *Proc. ACM STOC'94*, 1994.

[4] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach," in *Proc. ACM MOBICOM'06*, Sep. 2006.

[5] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," To appear in *IEEE Trans. Inform. Th.*, vol. 54, no. 2, Feb. 2008.

[6] J. G. Dai, "On the positive Harris recurrence for open multiclass queueing networks: a unified approach via fluid limit models," *Ann. Appl. Probab.*, vol. 5, no. 1, pp. 49–77, Feb. 1995.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM'00*, Mar. 2000.

[8] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, June 2006.

[9] J.-H. Hoepman, "Simple distributed weighted matchings," Oct. 2004, eprint cs.DC/0410047.

[10] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *ACM/Springer WINET*, vol. 11, no. 4, pp. 471–487, July 2005.

[11] C. Joo, X. Lin, and N. B. Shroff, "Performance limits of greedy maximal matching in multi-hop wireless networks," in *Proc. IEEE CDC'07*, Dec. 2007.

[12] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proc. IEEE CDC'06*, Dec. 2006.

[13] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[14] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

[15] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. ACM SIGMETRICS'06*, June 2006.

[16] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[17] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

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

[19] X. Wu and R. Srikant, "Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proc. IEEE CDC'05*, Dec. 2005.

[20] ——, "Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[21] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," To appear in *Proc. IEEE INFOCOM'08*, Apr. 2008.