MIT/LIDS Technical Report #2719, July 2006

Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks

Gil Zussman, Andrew Brzezinski, and Eytan Modiano
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139
{gilz, brzezin, modiano}@mit.edu

***Abstract*— Efficient operation of wireless networks requires *distributed* scheduling and routing algorithms that take into account interference constraints. Recently, a few algorithms for networks with primary- or secondary-interference constraints have been developed. Due to their distributed operation, these algorithms can achieve *only a guaranteed fraction* of the maximum possible throughput. It was also recently shown that if a set of conditions (known as Local Pooling) is satisfied, simple distributed scheduling algorithms achieve 100% throughput. However, previous work regarding Local Pooling focused mostly on single-hop interference or single-hop traffic. In this paper, we study the *multihop* implications of Local Pooling. We show that in many cases, as the interference degree increases, the Local Pooling conditions are more likely to hold. Namely, multihop interference improves the performance of distributed algorithms. To prove this property, we identify several graph classes that satisfy Local Pooling and analyze their behavior under multihop interference. Regarding multihop traffic, we show that if the network satisfies only the single-hop Local Pooling conditions, distributed joint routing and scheduling algorithms are not guaranteed to achieve maximum throughput. Therefore, we present new conditions for Multihop Local Pooling, under which distributed algorithms achieve 100% throughout. Finally, we determine network topologies in which the conditions hold and discuss the algorithmic implications of the results.**

***Index Terms*— Stability, Distributed algorithms, Wireless networks, Local Pooling, Interference, Scheduling, Routing.**

I. INTRODUCTION

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. A *centralized* joint routing and scheduling policy that achieves the maximum attainable throughput region was presented in the seminal paper by Tassiulas and Ephremides [23]. However, the lack of central control in wireless networks calls for the design of *distributed* algorithms. Such algorithms can usually achieve only a fraction of the maximum throughput. Yet, it has been shown by Dimakis and Walrand [10] that there are network topologies in which distributed *scheduling* algorithms *achieve* 100% *throughput*. In this paper we study the effect of *multihop* interference on these topologies and characterize topologies in which simple distributed *joint routing and scheduling* algorithms achieve 100% throughput.

The policy of [23] applies to a multihop wireless network with a stochastic packet arrival process and is guaranteed to stabilize the network (i.e. provide 100% throughput) whenever the arrival rates are within the stability region. The results of [23] have been extended to various settings of wireless networks and input-queued switches (e.g. [1], [18], [20], and references therein). However, throughput optimal algorithms based on [23] require the repeated solution of a *global optimization problem*, taking into account the queue backlog information for every link in the network. For example, even under simple primary interference constraints[1] 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 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. Lin and Shroff [16] studied the impact of distributed imperfect scheduling on cross-layer rate control. Regarding primary interference constraints, they showed that using a *distributed maximal matching* algorithm along with a rate control algorithm may achieve 50% throughput. Similar results for different settings were also obtained in [7], [8], [15], [22]. It was also proved in [7], [15], [22], [25] that under secondary interference constraints[2] the stability region obtained by a distributed maximal scheduling algorithm may be significantly smaller than the stability region under a perfect (centralized) scheduler. In particular, Chaporkar et al. [7] showed that a distributed algorithm may achieve as low as $1/8$ of the possible throughput.

Recently, Dimakis and Walrand [10] 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[3] and study the performance a *greedy maximal weight scheduling* algorithm (termed Longest Queue First - LQF) that selects the set of served queues greedily according to

[1] Primary interference constraints imply that each pair of simultaneously active links must be separated by at least one hop (i.e. the set of active links at any point of time constitutes a matching) [7], [13], [16], [19], [26].

[2] 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 [7], [25].

[3] 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 [8].

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 can be easily obtained in a distributed manner). These conditions are referred to as *Local Pooling* (LoP) and are related to the properties of all maximal independent sets in the conflict graph. Using these conditions, in [10] it was shown that tree interference graphs satisfy LoP. In [4] a few other graphs satisfying LoP were identified and it was proved that under primary interference constraints, tree network graphs yield interference graphs that satisfy LoP.

Although some knowledge about LoP has been acquired, the results of [4] and [10] are constrained to networking environments that are inherently *single-hop*, where packets must depart the system upon transmission. This is an overly restrictive requirement in wireless networks. In addition, regarding network topologies that satisfy LoP, [10] provided mostly abstract conditions, while [4] focused on primary interference constraints. Although these constraints may hold for specific technologies, they are not realistic in most practical settings. Hence, in order to allow the development of algorithms that take advantage of LoP, we study the *fundamental implications of multihop interference and multihop traffic on LoP*.

We begin by presenting motivating examples which show that in many cases *multihop interference* assists the LoP conditions. For example, a 6-node ring network graph does not satisfy LoP under primary interference, whereas it satisfies LoP under secondary interference. Such examples demonstrate that the performance of distributed algorithms may be improved by increased interference, thereby motivating the systematic study of the effect of multihop interference on LoP. However, since the knowledge about graphs that satisfy LoP is limited, we first focus on identifying several new classes of LoP-Satisfying graphs. It is shown that within the class of perfect graphs, chordal graphs, chordal bipratite graphs, cographs, and a subgroup of co-comparability graphs all satisfy LoP. These observations increase the number of graphs that are known to satisfy LoP by a few orders of magnitude.

Subsequently, we use the acquired knowledge to study the effect of increased interference on LoP. We focus on a generalization of the primary (1-hop) and secondary (2-hop) interference models to a k-hop interference model, where k is termed the interference degree. We show that in many cases, as k increases, it is more likely that the LoP conditions hold, and thereby, it is more likely that simple distributed algorithms achieve 100% throughput. Moreover, for many network topologies, there is an interference threshold k, above which the corresponding interference graphs satisfy LoP. At first glance, it seems that since it is known that the worst case performance deteriorates as the interference degree increases [7], [15], [25], the results are counter-intuitive. However, the actual meaning of the results is that in many topologies as k increases, the resulting interference graph is such that maximal scheduling achieves the maximum throughput instead of the worst case throughput.

Networks with *multihop traffic*, where packets follow a fixed multihop path, have been studied by Wu and Srikant [24], who proposed the use of regulators along with a maximal matching scheduling algorithm. It was shown in [24] that under primary interference constraints, the throughput may be reduced to 50%. These results have been extended in [15], [25] where it was also pointed out that only a fraction of the throughput is attainable. Since the LoP results of [4], [10] have been constrained to single-hop traffic, it is desirable to identify topologies in which distributed algorithms can obtain 100% throughput in the multihop network setting.

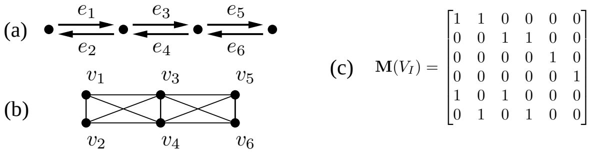
In this paper, we show that the single-hop LoP conditions introduced in [10] are *insufficient* to guarantee stability in the multihop routing environment. Therefore, we study the LoP properties of a distributed routing and scheduling framework which is based on the backpressure mechanism of [23]. In this framework the edge weights are obtained by the backpressure mechanism but unlike in [23], a *distributed* maximal scheduling algorithm is used to determine which edges should be activated. We derive new LoP conditions that are sufficient for guaranteeing that the framework achieves 100% throughput in the multihop routing environment. Then, we present a specific network topology that satisfies the multihop LoP conditions and show that the class of topologies satisfying these conditions is strictly included within the class of single-hop LoP-Satisfying graphs.

The main contributions of this paper are two-fold. First, we show that due to Local Pooling, as the interference degree increases, it is more likely that simple distributed algorithms achieve 100% throughput. The second contribution is the derivation of novel Local Pooling conditions for networks with multihop traffic. We note that an important byproduct is the identification of several graph classes that satisfy Local Pooling. To the best of our knowledge this is the first attempt to study the multihop implications of Local Pooling. The obtained results can serve as a basis for the development of Local Pooling based algorithms.

This paper is organized as follows. In Section II we present the network model and the single-hop LoP conditions. In Section III we demonstrate cases in which additional interference assists the LoP conditions. In Section IV we present several new classes of conflict graphs satisfying LoP. Then, in Section V we discuss the effect of multihop interference on the satisfaction of the LoP conditions. New LoP conditions for networks with multihop traffic are presented in Section VI. In Section VII we show that the multihop LoP conditions are distinct from the single-hop conditions and identify network topologies that satisfy them. We summarize the results and discuss future research directions in Section VIII.

II. Network Model and Local Pooling

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 $e_{12} = (v_1, v_2)$



$$\mathbf{M}(V_I) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Fig. 1. (a) Network graph G_N, (b) the corresponding interference graph G_I under primary interference, and (c) the matrix of maximal link activations.

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. Bold symbols are associated with vectors and matrices. For matrix $\mathbf{A}$, $\mathbf{A}_{\cdot,j}$ is the j-th column of $\mathbf{A}$, and $\mathbf{A}_{i,\cdot}$ is the i-th row. For vector $\mathbf{a}$ and index set E, $\mathbf{a}_E$ is the vector $(a_e : e \in E)$.

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 [8]. 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). Fig. 1 contains a network graph G_N and the corresponding interference graph G_I under primary interference constraints.

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 $\Pi(G_N)$. Continuing the example of Fig. 1, the matrix $\mathbf{M}(V_I)$ for interference graph G_I is contained in Fig. 1(c).

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 network link. There is no self-traffic. We will refer to packets destined to node $j \in V_N$ as *commodity j packets*. Let $A_{i,j}(t)$ denote the number of exogenous commodity j packets that arrived at node i by the end of time slot t. We assume that the arrivals have long term rates $\lambda_{i,j} = \lim_{t\to\infty} A_{i,j}(t)/t$, with overall system arrival rate vector $\boldsymbol{\lambda} = (\lambda_{i,j},\, i, j \in V_N)$.

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

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 link. We denote the corresponding *service activation matrix* by $\mathbf{S} = (S_{e,j}, e \in E_N, j \in V_N)$, where for edge $e \in E_N$, and commodity $j \in V_N$, $S_{e,j}$ can have value 0 or 1, depending on whether e is inactive or active for servicing 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 the set of admissible service activation matrices, $\mathcal{S}$:

$$\mathcal{S} = \left\{ \mathbf{S} \in \{0,1\}^{m \times n} : \textstyle\sum_{j \in V_N} \mathbf{S}_{\cdot,j} \in \Pi(G_N) \right\}.$$

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* $\mathbf{R}^j = (R^j_{i,e}, i \in V_N, e \in E_N)$, where: $R^j_{i,e} = 1$ if $\sigma(e) = i$; $R^j_{i,e} = -1$ if $\tau(e) = i$ and $i \neq j$; and $R^j_{i,e} = 0$ otherwise. Denote by $d_{i,j}(\mathbf{S})$ the service to queue $Q_{i,j}$ under activation matrix $\mathbf{S}$. Using the above routing matrix we can express $d_{i,j}(\mathbf{S}) = \mathbf{R}^j_{i,\cdot}\mathbf{S}_{\cdot,j}$.

A. 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 $\mathbf{S}^l \in \mathcal{S}$, $1 \le l \le L$ such that

$$\lambda_{i,j} \le \textstyle\sum_{l=1}^{L} \alpha_l d_{i,j}(\mathbf{S}^l), \text{ where } \alpha_l \ge 0\, \forall l \text{ and } \textstyle\sum_{l=1}^{L} \alpha_l \le 1.$$

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

A scheduling algorithm at each time slot 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.[4]

Definition 2 (Stable Algorithm): A scheduling algorithm is stable if for any arrival process with rate vector $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$,

$$\lim_{t\to\infty} Q_{i,j}(t)/t = 0 \quad w.p.1 \quad \forall i, j \in V_N.$$

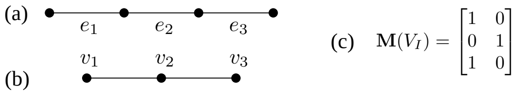
Tassiulas and Ephrimedes developed a stable scheduling algorithm that applies in this setting [23]. At time $t \ge 0$, their algorithm calculates for each link the maximum backpressure among all commodities for that link. Denote this by vector $\mathbf{Z}^*(t) = (Z^*_e(t), e \in E_N)$, where $Z^*_e(t) = \max_{j \in V_N} Z_{e,j}(t)$. The algorithm then selects a link activation

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

Routing is carried out over each edge e having $\pi^*_e(t) = 1$, by servicing any commodity $j \in \arg\max_j Z_{e,j}(t)$ across that edge (if any commodity j packets await service).

For general interference graph G_I, the algorithm of [23] must find the *maximum weight independent set* in G_I at each time slot to obtain an optimal solution to (1). Namely, it must solve an NP-Complete problem in every slot. Under primary interference, the graph 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 distributed algorithms can only obtain an approximate solution, resulting in a fractional throughput. Hence, even under simple traffic model and interference constraints, it is difficult to obtain 100% throughput in a distributed manner.

[4] This stability criterion is often termed *rate stability* [1], [7], [9].



$$\mathbf{M}(V_I) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Fig. 2. (a) Undirected network graph G_N, (b) the corresponding interference graph G_I under primary interfernce, and (c) the matrix of maximal link activations.

B. Simplifications for Single-Hop Traffic

When the network is subjected exclusively to single-hop traffic, a few notable simplifications occur in the model (see e.g. [7], [9], [15], [18], [19]). In this case, by definition, each network link e can only carry the traffic of a single commodity: $\tau(e)$. Thus, the differential backlog of link e equals the queue backlog of commodity $\tau(e)$. The throughput optimal algorithm of [23] specializes in this case to require that single-hop service be applied at each time t to the link activation

$$\boldsymbol{\pi}^*(t) \in \underset{\boldsymbol{\pi} \in \Pi(G_N)}{\arg\max}\, \boldsymbol{\pi}^T \mathbf{Q}(t). \tag{2}$$

Above we understand $\mathbf{Q}(t)$ as the vector $\mathbf{Q}(t) = (Q_e(t), e \in E_N)$, where $Q_e(t)$ is the queue backlog of packets awaiting single-hop service across link e.

Since routing plays no role in the single-hop scenario, it is convenient to treat the network graph G_N as undirected. This simplifies the interference graph (an example of an undirected graph and its primary interference graph appears in Figs. 2(a)-(b)). In this case, the weight at time t of each undirected edge $e = \{v_1, v_2\}$ equals the maximum weight of the queues that can be serviced across that link: $\max\{Q_{v_1,v_2}(t), Q_{v_2,v_1}(t)\}$. We will adopt this convention in our study of Local Pooling under multihop interference in sections III-V.

C. Local Pooling Conditions - Single Hop Traffic

We briefly reproduce important definitions and implications of Local Pooling (LoP) in networks with single-hop traffic, presented in [4], [10]. In Section VI we will derive the LoP conditions for the *multihop* case. Recall that $\mathbf{M}(V_I)$ is the collection of maximal independent vertex sets on G_I, organized as a matrix (an example appears in Fig. 2(c)). 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. We now define the following notions.

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.*

Continuing with the example of Fig. 2, we can see that SLoP is satisfied for the interference graph G_I using the vector $\boldsymbol{\alpha} = (1, 2, 1)$: $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = 2\mathbf{e}^T$. In a similar manner, it can be easily shown that all subgraphs of G_I satisfy SLoP, and therefore, G_I satisfies OLoP.

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 easily implemented in a distributed manner and has the following property.

Theorem 1 (Dimakis and Walrand, 2006 [10]): If interference graph G_I satisfies OLoP, a Maximal Weight Independent Set (MWIS) scheduling algorithm achieves 100% throughput.

III. Multihop Interference - Motivation

Here, we demonstrate the effect of multihop interference on LoP. For simplicity of presentation, we focus on single-hop traffic and use the model and LoP conditions of Sections II-B and II-C. Multihop traffic is discussed in Section VI.

Primary interference constraints (referred to also as 1-hop interference) are among the simplest possible constraints. Most technologies impose more complicated constraints. For example, in [2] and [25] it is indicated that in IEEE 802.11, each pair of simultaneously active links must be separated by at least two hops. Hence, the set of active links constitutes a distance-2 matching, also known as an *induced matching* [2], [5]. We refer to such an interference model as 2-hop interference and to the resulting constraints as secondary interference constraints.

In this paper, we study the generalization of the 1- and 2-hop interference models to a k-hop interference model, where the set of active links at each time slot is a distance-k matching [2], [6]. We refer to k as the *interference degree*. We denote the stability region under k-hop interference by $\boldsymbol{\Lambda}_k^*$. It is clear that $\boldsymbol{\Lambda}_k^*$ cannot increase with k (and often decreases with k), as interference between the links composing the network can only increase. Moreover, according to [7], [15], [25], as k increases, the worst case throughput fraction obtained by a distributed MWIS algorithm significantly decreases. For example, recall that in the 1-hop model, the stability region is reduced from $\boldsymbol{\Lambda}_1^*$ to $\boldsymbol{\Lambda}_1^*/2$, while in the 2-hop model it can be reduced from $\boldsymbol{\Lambda}_2^*$ to $\boldsymbol{\Lambda}_2^*/8$. Therefore, the intuition derived from these results is that more interference negatively affects the performance of simple distributed algorithms.

In this section and in Section V, we show that counter-intuitively, in many cases *more interference assists the operation of distributed algorithms*. Namely, as k increases, it is more likely that the OLoP conditions hold, and thereby, it is more likely that simple distributed algorithms will achieve $\boldsymbol{\Lambda}_k^*$.

We now demonstrate the intuition on which this observation is based. Consider a ring network graph with 6 nodes (referred to as the 6-ring), whose interference graph under primary interference is also a 6-ring. According to [10], the 6-ring interference graph *does not satisfy OLoP* and in general a *maximal weight matching* algorithm does not achieve 100% throughput. The best known result then provides that a maximal weight matching algorithm guarantees 50% throughput [16]. Under 2-hop interference, the interference graph has 6 more edges (see Fig. 3(a)). According to [4], this specific graph satisfies OLoP, and therefore, a MWIS algorithm achieves 100% throughput. Under 3-hop (or higher)

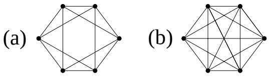


Fig. 3. (a) 2-hop and (b) 3-hop interference graphs of a 6-ring network graph

interference, the interference graph becomes a clique (see Fig. 3(b)) which satisfies OLoP [4].[5] Hence, although under 1-hop interference, a maximal weight algorithm guarantees 50% throughput, under k-hop interference ($k \geq 2$) 100% throughput is guaranteed.

Under k-hop interference, the interference graph becomes an OLoP-Satisfying clique when k equals the network diameter. It seems reasonable to expect that for a particular network graph G_N, as the interference degree increases there exists an *interference threshold*, below which OLoP fails, and above which OLoP is satisfied. We demonstrate this property by considering small graphs. In [4] it was shown that out of 1,252 simple interference graphs of up to 7 nodes 14 fail OLoP. The following observation is obtained by exhaustively considering the corresponding k-hop ($k \geq 2$) interference graphs.

Observation 1: All k-hop ($k \geq 2$) interference graphs corresponding to network graphs with up to 7 edges satisfy OLoP.

The observations of this section motivate us to study general multihop interference properties. We wish to understand how an increased interference degree affects OLoP in different classes of graphs. However, little is known about OLoP properties of graphs. Therefore, we next study this issue, which leads us to Section V, where we use the acquired understanding of OLoP to study the effect of increased interference.

IV. Interference Graphs Satisfying Local Pooling

The OLoP properties of graphs are only beginning to be understood. Small graphs were studied by exhaustive search [4]. Additionally, structural properties were used in [4], [10] to show that the following interference graphs satisfy OLoP: trees, forests, *clique trees*, where each pair of cliques shares at most a single vertex, and a *pair-of-cliques* connected by disjoint edges.

In order to better understand the effect of interference on LoP, we use structural properties to identify various graph classes that satisfy OLoP. We define a new class of graphs as the *OLoP-Satisfying* class. We identify known graph classes that are included within this class or intersect with it. It turns out that all the graph classes we identify using structural properties are subclasses of the class of perfect graphs. On the other hand, some of the graphs identified by the exhaustive search [4] are not perfect graphs. Hence, in the following discussion we differentiate between perfect and non-perfect graphs. Our investigation leads to the taxonomy of graph classes depicted in Fig. 4, showing the relationship of the OLoP-Satisfying class to the graph classes considered here.

Before proceeding, we present some basic graph theoretic definitions, required in the following sections. For brevity, we will refer to an induced subgraph over the a subset of nodes $V' \subseteq V_I$ as an *induced subgraph*. The complement $\overline{G} = (V, \overline{E})$ of a graph $G = (V, E)$ is defined by $\overline{E} = (u, v) : u, v \in V,\ u \neq v$ and $(u, v) \notin E$. A *chord* of a cycle (path) is an edge between two vertices of the cycle (path) that is not an edge of the cycle (path). A cycle (path) is *chordless*, if it contains no chords. We denote by C_n and P_n a chordless cycle and a chordless path (respectively) of length n. We will refer to a chordless cycle C_n and to an n-ring interchangeably. We denote by K_n a clique (complete graph) of n nodes.

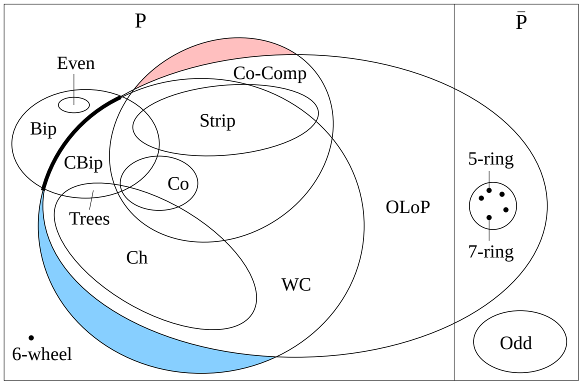


Fig. 4. The relations between the OLoP-Satisfying class and other graph classes: P - perfect, P̄ - non-perfect, WC - weakly chordal, Ch - chordal, CBip - chordal bipartite, Bip - bipartite, Co - cograph, Co-Comp - co-comparability, Strip - strip-of-cliques, Even - cycles C_n with n even and $n \geq 6$, Odd - graphs with induced C_n with n odd and $n \geq 9$.

A. Perfect Graphs

A graph is *perfect*, if for each induced subgraph the size of the largest clique equals the chromatic number. Several classical graph classes such as bipartite graphs, chordal graphs, comparability graphs, and their complements are perfect [3]. Here, we will identify a number of important classes of perfect graphs that are also subclasses of the OLoP-Satisfying class. We will show that all of the graphs identified in [4], [10] are *simple* special cases in these classes. Before describing the results we introduce some classes of perfect graphs [3].

Definition 5: A graph G is chordal *if each cycle in G of at least 4 nodes has at least one chord. A graph G is* weakly chordal *if G and its complement contain no induced chordless cycle C_n, $n \geq 5$. A bipartite graph B is* chordal bipartite *if each cycle in B of length at least 6 has a chord. A graph is a* cograph *if it does not contain the path graph P_4 (depicted in Fig. 2(a)) as an induced subgraph.*

Notice that the chordal bipartite class is the intersection of the weakly chordal and bipartite classes.

The following theorem summarizes five results concerning the OLoP properties of several large graph classes. The proof can be found in Appendix A.

Theorem 2:

1) *The following graph classes belong to the OLoP-Satisfying class: Chordal Graphs, Chordal Bipartite Graphs, and Cographs.*
2) *All even cycles C_n with $n \geq 6$ fail SLoP.*
3) *Bipartite Graphs that are not Chordal Bipartite Graphs do not belong to the OLoP-Satisfying class.*

Fig. 4 illustrates the inclusion of the chordal, chordal bipartite, and cograph classes within the OLoP-Satisfying class.

[5] This results from the fact that in a clique, a maximal weight algorithm obtains the maximum weight solution.

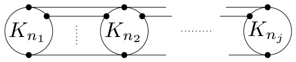


Fig. 5. The structure of a strip-of-cliques.

The class of chordal graphs has a few notable subclasses (i.e. classes of special graphs that are known to be chordal), including the strongly chordal, split, interval, threshold, and tree classes (for more information see [3]). Theorem 2 implies that all these subclasses satisfy OLoP. Therefore, the observation of [10] that trees satisfy OLoP immediately follows from Theorem 2. Similarly, since a clique tree is chordal, the observation of [4] that a clique tree satisfies OLoP is also a result of the theorem. Theorem 2 also implies that all subclasses of chordal bipartite graphs satisfy OLoP, including the convex and bipartite $\cap$ distance-heriditary classes.

The final contribution of Theorem 2 is its characterization of a *sharp* boundary separating the chordal bipartite graphs (OLoP-satisfying) from the bipartite graphs that are not chordal bipartite (not OLoP-satisfying). This boundary is depicted as a thick line in Fig. 4. This result follows directly from the failure of the OLoP conditions in even cycles C_n with $n \geq 6$. Hence, any graph class that includes the bipartite graphs as a subclass cannot be fully included within the OLoP-Satisfying class. This allows us to exclude many of the major subclasses of perfect graphs (e.g. preperfect, strongly perfect, quasi-parity, and bip* [3]) from the list of classes that can be fully included in the OLoP-Satisfying class.

Two major classes that have not been excluded as subclasses of the OLoP-Satisfying class are the weakly chordal class and the co-comparability class, which is defined below.

Definition 6: Co-comparability graphs *can be characterized as intersection graphs of a set of curves between two parallel lines in the plane, every curve has one endpoint on each of the lines*[6] [11].

In Fig. 4 we have shaded portions of the weakly chordal and co-comparability classes to indicate the uncertainty of their inclusion relations with OLoP-Satisfying. Determining the nature of these shaded regions (whether or not they exist) is left as an open problem.

We now present a subclass of the co-comparability class to which we refer as a *strip-of-cliques*. A graph is in this class, if it is composed from an ordered set of cliques $1, \ldots, j$, where two adjacent cliques i, $i+1$ are connected by any number of disjoint edges, and cliques that are not adjacent are not connected directly. Fig. 5 illustrates such a graph. Notice that the pair-of-cliques presented in [4] is a specific case of a strip-of-cliques. The following lemmas show that a strip-of-cliques graph satisfies OLoP and that any such graph is a co-comparability graph.

Lemma 1: Every strip-of-cliques graph satisfies OLoP.

Proof: Every connected induced subgraph of a strip-of-cliques is a strip-of-cliques. If the induced subgraph is disconnected, each component is a strip-of-cliques. According to [4, Prop. 1], if each component satisfies SLoP, the disjoint union satisfies SLoP. Thus, proving that a strip-of-cliques satisfies SLoP will yield that it satisfies OLoP.

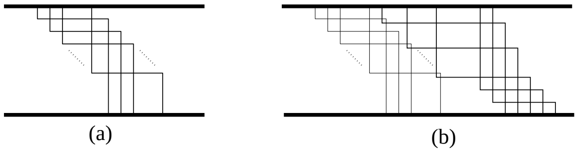


Fig. 6. Demonstrating that the strip-of-cliques is a co-comparability graph, with (a) a set of curves whose intersection graph is a clique, and (b) the introduction of a neighboring clique, where the curves corresponding to the original clique are thinner than the new ones.

If the strip-of-cliques is only a single clique, then according to [4, Lemma 4] it satisfies SLoP. If this is not the case, we shall refer to one of the two cliques that is connected to only a single clique as an *edge clique*. For example, in Fig. 5 K_{n_1} is an edge clique.

If the edge clique includes only a single node v, then it is connected by an edge to a node u in the neighboring clique. One of these nodes will be active in any independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices v and u, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. If the edge clique includes more than one node, one of the nodes in the edge clique will be active in any independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices of the edge clique, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Hence, a strip-of-cliques satisfies SLoP. Thus, we have that SLoP is satisfied in any subgraph of a strip-of-cliques, and therefore, OLoP is satisfied. ■

Lemma 2: Every strip-of-cliques graph is a co-comparability graph.

Proof: According to definition 6, if the strip-of-cliques is a co-comparability graph, then each vertex of the strip-of-cliques can be represented as a curve joining two parallel lines. An edge exists between two vertices in the strip-of-cliques if and only if the corresponding curves intersect at some point. We will describe a procedure for constructing the curves that represent an arbitrary strip-of-cliques.

Begin with the leftmost clique, having n_1 vertices. Cascade n_1 curves as shown in Fig. 6(a), making sure that each of the curves is exposed on the right, in a staircase fashion. Clearly, each of the curves intersects with all others, which implies a clique intersection graph, K_{n_1}.

We next demonstrate how to introduce the j-th clique in the strip-of-cliques, $j \geq 2$. Consider the curves that represent the $(j-1)$-th clique, in order, by descending the staircase on the right. If the vertex v_1 corresponding to one of these curves shares an edge with a vertex v_2 in the j-th clique, then a curve is drawn to represent v_2, by intersecting with the stair corresponding to v_1. This is depicted in Fig. 6(b), where the first, third, and last curves on the staircase intersect with curves corresponding to the adjacent clique. Any remaining vertices in the j-th clique that do not intersect vertices in the $(j-1)$-th clique are simply included as curves that do not intersect the staircase of the $(j-1)$-th clique. There are two such curves represented in Fig. 6(b). Note that the curves corresponding to the j-th clique are once again organized to form a staircase on the right.

[6] Since the graph is an intersection graph [17], every curve is represented by a node and nodes are connected by an edge, if the curves they represent intersect.

Clearly, the above procedure can be repeated iteratively until the entire strip-of-cliques is represented as an intersection graph of curves between two parallel lines. Consequently, the strip-of-cliques is a co-comparability graph. ■

Fig. 4 depicts the strip-of-cliques class as intersecting with the weakly chordal class and in particular with the subclass of chordal bipartite. One such graph that is chordal bipartite is C_4, which can be viewed as two cliques connected by two parallel links. Another such graph that is weakly chordal is composed of two K_3's connected by 2 disjoint links. On the other hand, a graph composed of two K_3's connected by 3 parallel links is actually a $\overline{C_6}$, and therefore, is not weakly chordal.

Finally, we note that the strip-of-cliques class can be generalized to a larger OLoP-Satisfying class by connecting cliques in a tree structure such that pairs of cliques are connected by a number of disjoint edges and no cycle-of-cliques is closed. Proving that such a structure satisfies OLoP can be done using similar arguments to the ones used in the proof of Lemma 1.

We finish this section by providing some context regarding the magnitude of the results. Consider the set of simple graphs having 7 nodes, of which there are 1,044 distinct graphs. Of these graphs, 393 are chordal, and 180 are cographs, with some overlap between these two classes. These numbers can be compared to the 37 forests and 11 trees that were known to satisfy OLoP. Similarly, when considering the set of simple 11 node graphs, the number of chordal graphs is 1,392,387, compared to 710 forests and 235 trees. We summarize that our understanding of the OLoP-Satisfying class has expanded significantly beyond the trees and forest graphs.

B. Non-Perfect Graphs

The *OLoP-Satisfying* class includes also graphs that are not perfect. We first use the numerical observations of [4] to identify non-perfect graphs that satisfy OLoP. The 5-ring, C_5, which is the only 5-node non-perfect graph, satisfies OLoP. Moreover, since all 6-node graphs except C_6 satisfy OLoP, all non-perfect 6-node graphs satisfy OLoP. Finally, all 7-node graphs satisfy OLoP besides a specific one illustrated in [4] and those that have an induced 6-ring, which leads us to the observation that *134 out of the 138 non-perfect 7-node graphs satisfy OLoP*. In Fig. 4 all these graphs appear in a single class (containing C_5 and C_7) within the OLoP-Satisfying class.

We now show that all non-perfect graphs that have an induced odd cycle with at least 9 nodes fail OLoP (see the Odd class in Fig. 4). This follows from the following theorem, whose proof can be found in Appendix B.

Theorem 3: All odd cycles C_n with $n \geq 9$ fail SLoP.

V. Local Pooling under Multihop Interference

In Section III we discussed the k-hop interference model and showed that in several cases, increasing the interference degree (k) results in an interference graph that satisfies OLoP. We now use the acquired knowledge regarding the OLoP-Satisfying class to study this phenomenon. We denote by G^k the k-th power of G: G^k has the same vertex set V as G, and $u, v \in V$ are adjacent in G^k, if the minimum path length between u and v in G is at most k. Given a 1-hop interference graph G_I^1, the corresponding k-hop interference graph is G_I^k.

We focus on graph classes that appear in Fig. 4. First, recall that according to Observation 1, all k-hop ($k \geq 2$) interference graphs with up to 7 nodes satisfy OLoP. Therefore, a number of 1-hop interference graphs outside the OLoP-Satisfying class yield k-hop interference graphs that are OLoP-Satisfying. These graphs are the 6-ring, the 6-wheel, and the four non-perfect 7-node graphs outside the OLoP-Satisfying class.

We now define the following subclass of chordal graphs.

Definition 7 (Strongly Chordal Graph [3]): A graph G is strongly chordal, *if G is chordal and each cycle in G of even length at least 6 has an odd chord (a chord (i, j) is an odd chord, if the distance in the cycle between i and j is odd).*

Since the strongly chordal graphs belong to the chordal class, Theorem 2 implies that strongly chordal graphs are OLoP-Satisfying. It is known that the strongly chordal class is *strongly closed under power*. Namely, if an interference graph G_I^k is strongly chordal, then G_I^{k+j} is strongly chordal for all $j \geq 1$ [3]. Therefore, even if the 1-hop interference graph is not strongly chordal, once an interference graph becomes strongly chordal (and thereby OLoP-Satisfying), increased interference degree will generate OLoP-Satisfying graphs. This agrees with the intuition in Section III, of an *interference threshold k* above which all interference graphs satisfy OLoP.

The strongly chordal class has a number of subclasses, the simplest ones being a tree and a clique tree. On the other hand, a *pair-of-cliques* which is a specific case of the *strip-of-cliques* (defined in Section IV-A) is not strongly chordal. However, in the proof of the following lemma we show that its corresponding 2-hop interference graph is chordal. By using similar methods it can also be shown to be strongly chordal.

Lemma 3: If the 1-hop interference graph G_I^1 is a tree, a clique tree, any strongly chordal graph, or a pair-of-cliques, G_I^k satisfies OLoP for every $k \geq 1$.

Proof: A tree is strongly chordal, since it has no cycles. The cycles in the tree of cliques have all possible the chords, and therefore, a tree of cliques is strongly chordal. Strongly chordal graphs are strongly closed under power. Therefore, given an interference graph G_I^1 which is strongly chordal (including a tree or a tree of cliques), the corresponding graphs $G_I^k \, \forall \, k \geq 1$ are strongly chordal and satisfy OLoP.

A G_I^1 which is a pair-of-cliques (i.e. K_1 and K_2 connected by a number of disjoint edges) is not strongly chordal, since it can have an induced chordless cycle C_4. We now show that the corresponding G_I^2 is chordal. G_I^2 is composed of 2 cliques (K_1 and K_2) that share a number of nodes. The nodes that are not shared are those that in G_I^1 are not connected directly to the other clique. Assume that G_I^2 is not chordal. In such a case, there has to be a cycle of at least 4 nodes that has no chords. Such a cycle must include at least one of the non-shared nodes from K_1 and one of the non-shared nodes from K_2. It must also include 2 of the shared nodes. Since the shared nodes are part of both cliques, they are connected by a chord. Therefore, the cycle is not chordless, which is a contradiction. Consequently, G_I^2 is chordal and satisfies OLoP. The corresponding G_I^3 is a clique, and therefore, according to [4], satisfies OLoP. $G_I^k \, \forall \, k > 3$ is still a clique. ■

The following lemma shows that other graphs, identified in Section IV-A, that in general are not strongly chordal, also satisfy OLoP for any interference degree.

Lemma 4: If the 1-hop interference graph G_I^1 is a cograph or a strip-of-cliques, G_I^k satisfies OLoP for every $k \geq 1$.

Proof: According to [3] every connected subgraph of a cograph has diameter of at most 2. Therefore, the corresponding $G_I^k \, \forall \, k \geq 2$ is a clique and according to [4] satisfies OLoP.

We now use similar terminology to the one used in the proof of Lemma 1. According to Lemma 1, if G_I^1 is a strip-of-cliques, it satisfies OLoP. The interference graph G_I^k corresponding to G_I^1 is composed of cliques that share some nodes with their neighboring cliques. In particular, the nodes of the edge clique of G_I^1 are included in a clique containing several other nodes. We refer to this clique as the k-edge-clique.

If the k-edge-clique includes all the nodes, then it satisfies OLoP. Otherwise, there are nodes of the G_I^1 edge clique that are not shared with neighboring cliques. In that case, one node of the k-edge-clique will be active in any independent set. Therefore, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the vertices of the k-edge-clique, where the entries are set to 1, yields $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = \mathbf{e}^T$. Hence, the interference graph G_I^k satisfies SLoP. Using a similar reasoning it can be shown that any subgraph of G_I^k satisfies SLoP, and therefore, G_I^k satisfies OLoP. ■

Thus far, we have studied the LoP properties under multihop interference for most graphs represented in Fig. 4. We next turn our attention to particular *network graph* structures. An example of an interference graph G_I^1 resulting from 1-hop interference is given in Fig. 2. A second example is the n-ring network graph C_n, whose 1-hop interference graph is also C_n. Recall from Section IV that C_n fails OLoP for $n = 6$ and $n \geq 8$. Our numerical tests show that the 2-hop interference graph of any C_n with $n \leq 8$ satisfies OLoP. Hence, we observe that rings are network graphs that benefit from additional interference degrees.

Clearly, any network graph whose corresponding interference graph is one of the structures indicated in Lemmas 3 and 4 satisfies OLoP for any $k \geq 1$. In particular, we can derive the following result.

Theorem 4: Distributed MWIS algorithms achieve 100% *throughput in a tree* network graph *under any interference degree k.*

Proof: The interference graph G_I^1 of a tree network graph is a tree of cliques. According to Lemma 3 for such an interference graph, the corresponding G_I^k satisfies OLoP for any $k \geq 1$. ■

The 2-hop interference model is important, since it represents the IEEE 802.11 transmission constraints [2], [25]. We obtain the following result regarding this model by using results regarding squares of line graphs[7] studied in [5], [6]

Theorem 5: Distributed MWIS algorithms achieve 100% throughput in a chordal network graph *under secondary interference constraints (2-hop interference model).*

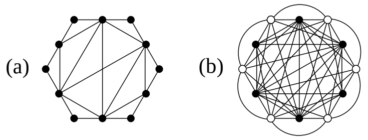


Fig. 7. (a) A chordal 1-hop interference graph and (b) the corresponding 2-hop interference graph that fails OLoP.

Proof: According to [5], given a chordal network graph G_N, the corresponding 2-hop interference graphs G_I^2 is chordal. According to Theorem 2, OLoP is satisfied in a chordal interference graph, and therefore, Distributed MWIS algorithms achieve 100% throughput. ■

Several important subclasses of chordal graphs have the potential to allow a MWIS algorithm to be throughput-optimal under 2-hop interference. One of the subclasses is the class of interval graphs [3], [6]. For that class the following stronger result holds.

Lemma 5: Distributed MWIS algorithms achieve maximum throughput in an interval network graph under a k-hop interference model ($k \geq 2$).

Proof: According to [6], given an interval network graph G_N, the corresponding 2-hop interference graphs G_I^2 is an interval graph. Interval graphs are strongly chordal [3], and therefore, the corresponding k-hop ($k \geq 2$) interference graphs G_I^k are is strongly chordal and OLoP-Satisfying. ■

Although for some chordal network graphs that are also strongly chordal distributed MWIS algorithms can be shown to achieve 100% throughput under a k-hop interference model ($k \geq 2$), this is not in general the case with the entire class of chordal graphs. In fact, if G^k is chordal, then G^{k+2} is chordal but it is not guaranteed that G^{k+1} is chordal [3]. Therefore, if the 2-hop interference graph G_I^2 is chordal, the corresponding k-hop interference graph G_I^k when k is even satisfies OLoP. When we study the transition from G_I^k to G_I^{k+1}, we find that there *are rare cases* where increasing the interference degree can result in a graph that fails OLoP. The following lemma summarizes this result.

Lemma 6: There are OLoP-Satisfying k-hop interference graphs for which OLoP is not satisfied in a corresponding j-hop ($j > k$) interference graph.

Proof: Assume that there are no such graphs. Consider the 1-hop interference graph G_I in Fig. 7(a). This is a chordal graph, and therefore, according to Theorem 2 it satisfies OLoP. The corresponding 2-hop interference graph G_I^2 appears in Fig. 7(b). The subgraph induced by the white nodes is a 6-ring, which fails SLoP. Therefore, OLoP fails in the 2-hop interference graph, which is a contradiction. ■

To conclude, this section has demonstrated that several (but not all) graph classes have the desirable property of having a threshold of interference degree above which the interference graphs are OLoP-Satisfying. In these instances, increasing interference degree positively affects the performance of simple distributed scheduling algorithms.

VI. LoP in Networks with Multihop Routing

In this section, we study the LoP properties in networks employing multihop routing, under *general* interference constraints. We focus on a framework based on a distributed

[7] In graph theoretic terminology, the interference graph resulting from 1-hop interference is called line graph [14].

MWIS scheduling algorithm using backpressure link weights. We obtain multihop local pooling conditions that are sufficient for guaranteeing 100% throughput under this framework.

A. Backpressure-based Scheduling and Routing

Recall from Section II-A that the optimal centralized scheduler (1) calculates *maximum* weight independent sets based on *backpressure* link weights. In our framework we consider the distributed *Maximal* Weight Independent Set (MWIS) algorithm used in the single-hop setting, but change the link weights to backpressure link weights. Thus, the MWIS algorithm operates on the interference graph with node weights derived from the backpressure link weights. This enables scheduling decisions for joint link activation and packet routing. As in the single-hop case, the framework is *independent of the global network topology and traffic statistics*.

Algorithm 1 Backpressure-based scheduling framework

```
1: for time index $t = 1, 2, \dots$ do
2:     For each directed edge $e \in E_N$ assign
           $Z_{e,j}(t) \leftarrow (Q_{\sigma(e),j}(t) - Q_{\tau(e),j}(t))$
3:     Assign $Z^*_e(t) = \max_j Z_{e,j}(t)$
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 $j^* \in$
       $\arg\max_j Z_{e,j}(t)$. Route $\min\{1, Q_{\sigma(e),j^*}(t)\}$ packets of
       commodity $j^*$ across $e$
6: end for
```

In step 4, the framework uses the MWIS algorithm to select a *maximal* weight link activation based upon maximum link backpressures, obtained in step 3. In step 5, the framework makes routing decisions to service commodities achieving maximum backpressure.

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 $E \subseteq E_N$. *Additionally*, given $E \subseteq E_N$, we must consider for each edge the set of possible combinations of commodities, subject to some restrictions. We formalize the possible edge/commodity combinations considered by introducing the Maximum Commodity Family (an example is given in Section VI-B).

Definition 8 (Maximum Commodity Family - $\mathcal{J}_E$): The Maximum Commodity Family for $E \subseteq E_N$, $E \neq \emptyset$, is given by $\mathcal{J}_E = \{(J^{\tilde{\mathbf{Q}}}_e, e \in E_N) : \tilde{\mathbf{Q}} \in \mathcal{Q}_E, \tilde{\mathbf{Q}} \neq 0\}$, where

$$\begin{aligned}\mathcal{Q}_E = \{&(\tilde{Q}_{i,j},\, i,j \in V_N, i \neq j) : \tilde{Q}_{i,j} \in \mathbb{R}_+ \forall i,j,\\ &E = \arg\max\nolimits_e \max\nolimits_j (\tilde{Q}_{\sigma(e),j} - \tilde{Q}_{\tau(e),j})\},\\ J^{\tilde{\mathbf{Q}}}_e = \{&j \in V_N : j \neq \sigma(e),\\ &\tilde{Q}_{\sigma(e),j} - \tilde{Q}_{\tau(e),j} \geq \tilde{Q}_{\sigma(e),j'} - \tilde{Q}_{\tau(e),j'} \ \forall j' \in V_N\}.\end{aligned}$$

The above definition relates closely to the fluid limit model for 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 7: 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 C. ■

Under the backpressure framework, 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 VI-B):

Definition 9 (Maximal Service Activation Set - $\mathcal{S}_{E,J}$): For $E \subseteq E_N$ and $J = (J_e, e \in E_N) \in \mathcal{J}_E$,

$$\begin{aligned}\mathcal{S}_{E,J} = \big\{&\mathbf{S} \in \mathcal{S} : \textstyle\sum_j \mathbf{S}_{E,j} \in \mathbf{M}(E),\\ &\qquad \mathbf{S}_{e,j} = 1 \textit{ implies } j \in J_e \textit{ when } e \in E_N\big\}\end{aligned}$$

In order to characterize the stability properties of the backpressure framework, 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. Recall that $d_{i,j}(\mathbf{S})$ is the service to queue $Q_{i,j}$ under activation matrix $\mathbf{S}$: $d_{i,j}(\mathbf{S}) = \mathbf{R}^j_{i,\cdot}\mathbf{S}_{\cdot,j}$.

Definition 10 (Backpressure Service Vector - $\mathbf{u}_{E,J}(\mathbf{S})$): 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 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 $e \in E, j \in J_e$:*

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

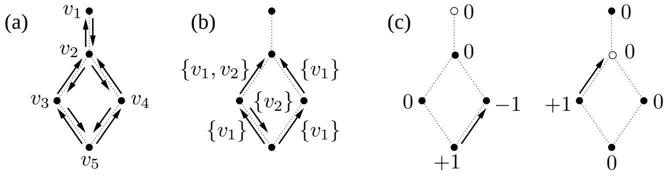


Fig. 8. (a) Network graph G_N, (b) the subset E of network graph edges, with corresponding commodity sets labeled at each edge, and (c) commodity graphs G_{v_1} (left) and G_{v_2} (right) for a particular maximal service activation.

B. Some Examples

In this section, we consider the network graph G_N of Fig. 8(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. In the next section we will show that certain conditions have to hold for each such combination. The subset E of network edges of interest is $E = \{e_{32}, e_{35}, e_{42}, e_{53}, e_{54}\}$, as depicted in Fig. 8(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. 8(c) the active link for servicing commodity v_1 packets in the graph on the left, and the active link for servicing 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.

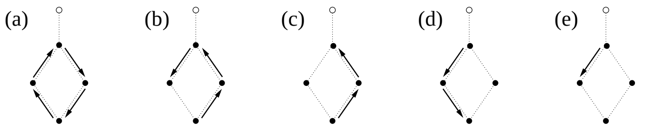


Fig. 9. Commodity graphs for commodity v_1, that are invalid based on the properties of Lemma 7.

We next provide examples to illustrate the properties of Lemma 7. Figs. 9(a)-9(d) show graphs that are inadmissible as the commodity v_1 graph, G_{v_1}, for the network graph depicted in Fig. 8(a): Fig. 9(a) fails Property 3 because G_{v_1} contains a directed cycle; Fig. 9(b) fails Property 4a since edge e_{53} provides a shorter path between vertices v_5, v_3; Fig. 9(c) fails Property 4b since edges e_{53}, e_{32} are not included in G_{v_1}; Fig. 9(d) fails Property 5a since the path $v_2 \to v_3 \to v_5$ belongs to G_{v_1}, while path $v_2 \to v_1$ belongs to G_N; and Fig. 9(e) fails Property 5b since edge e_{21} does not belong to G_{v_1}.

C. Stability of the Backpressure-Based Scheme

Here, we study the stability of the backpressure framework, and introduce new LoP conditions for stability. Recall that the quantity $d_{i,j}(\mathbf{S})$ is the amount of service at queue $Q_{i,j}$ resulting from service activation $\mathbf{S}$. Denote vector $\mathbf{d}(\mathbf{S}) = (d_{i,j}(\mathbf{S}),\ i, j \in V_N)$.

Definition 11 (Subgraph Multihop Local Pooling - 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

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

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq c, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J}. \tag{4}$$

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

Definition 12 (Overall Multihop Local Pooling - 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 the stability of the backpressure-based framework. The proof appears in Appendix D.

Theorem 6: If network graph G_N satisfies OMLoP, then the backpressure-based scheduling framework achieves 100% throughput.

Theorem 6 demonstrates the sufficiency of the OMLoP conditions for stability under the backpressure-based framework. In the next section, we consider natural questions that arise out of these conditions.

VII. Studying the OMLoP Conditions

We now show that the OMLoP conditions are distinct from the single-hop Local Pooling conditions studied in [4], [10], and demonstrate stability for a specific class of networks. We first show that any network graph G_N under which single-hop LoP fails should also fail the OMLoP conditions.

Lemma 8: If G_N fails OLoP, then it also fails OMLoP.

Proof: Suppose G_N fails single-hop OLoP. Then there exists a set of edges E of G_N for which the single-hop SLoP conditions fail. E can be considered without loss of generality as a set of directed edges, each of arbitrary directionality between its end nodes.

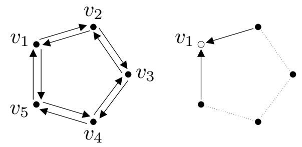
To demonstrate that SMLoP fails, consider the set of directed edges E, and commodity sets $J_e = \{\tau(e)\}$ for $e \in E_N$. It can be seen that $J = (J_e, e \in E_N) \in \mathcal{J}_E$. By definition, any active edge in a service activation $\mathbf{S} \in \mathcal{S}_{E,J}$ must be employed for single-hop service. This implies for each $\mathbf{S} \in \mathcal{S}_{E,J}$ that vector $\boldsymbol{\beta}$ can only lead to nonnegative contributions on the lefthand side of (3), as follows: each active edge has a value 1 associated with its origin vertex and a value 0 associated with its destination vertex, for the commodity being single-hopped across it. Since we require $\boldsymbol{\beta} \geq 0$, this implies that we can at best treat the second term on the left in (3) as zero for every $\mathbf{S} \in \mathcal{S}_{E,J}$.

Thus we must find nonzero $\boldsymbol{\alpha} \geq 0$, $c \geq 0$ such that $\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) = c\mathbf{e}^T$ for each $\mathbf{S} \in \mathcal{S}_{E,J}$. For any such $\mathbf{S}$, each active edge e services a packet to vertex $\tau(e)$, leading to a backpressure reduction across e of a single commodity $\tau(e)$ packet. Because each edge services a different commodity, all inactive edges in E see no change in the backpressure of their respective single-hop commodities. This implies $\mathbf{u}_{E,J}(\mathbf{S}) \in \mathbf{M}(E)$. Since all maximal activations over the edge set E are included in $\mathcal{S}_{E,J}$, the set of backpressure service vectors over $\mathcal{S}_{E,J}$ must then equal $\mathbf{M}(E)$. But $\mathbf{M}(E)$ fails the SLoP conditions: there does not exist nonzero $\boldsymbol{\alpha} \geq 0$, $c > 0$ such that $\boldsymbol{\alpha}^T \mathbf{M}(E) = ce^T$. Finally, $c = 0$ is invalid, because by its definition as the set of maximal link activations, each row of $\mathbf{M}(E)$ is nonzero, which means the inner product of any nonzero $\boldsymbol{\alpha} \geq 0$ with some column of $\mathbf{M}(E)$ exceeds $c = 0$. Thus G_N fails OMLoP. ■

In terms of Fig. 4, Lemma 8 implies that the class of graphs that are not OLoP-Satisfying can not be OMLoP-Satisfying. Namely, all network graphs having interference graphs with induced subgraphs that are bipartite and not weakly chordal, or induced C_n when $n = 6$ or $n \geq 8$ must fail OMLoP. The next theorem demonstrates that the OMLoP conditions are in fact *more restrictive* than their single-hop counterparts. Thus, the family of OMLoP-satisfying graphs is *strictly* smaller than that depicted in Fig. 4. It was indicated in Section IV-B that the 5-ring satisfies the single-hop OLoP conditions. Here we show that OMLoP fails for the 5-ring.

Theorem 7: The 5-ring (C_5) fails OMLoP.

Proof: Consider the network graph G_N depicted on the left below, and the subset of edges E depicted on the right. We denote by e_{ij} the directed edge from vertex v_i to v_j.



We consider the commodity collection $J = (J_e, e \in E_N)$, where for $e \in E_N$, $J_e = J_e^{\tilde{\mathbf{Q}}}$ and

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It can be seen that $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, which implies that J is a member of the maximum commodity family $\mathcal{J}_E$.

Each of the following edge/commodity activations is represented in the maximal service activation set $\mathcal{S}_{E,J}$:

$$\{(e_{21}, v_1), (e_{45}, v_1)\}, \{(e_{51}, v_1), (e_{32}, v_1)\}.$$

When we consider the backpressure service vectors associated with these activations, the second SMLoP conditions (4) require the existence of $\boldsymbol{\alpha}, c \geq 0$, $\boldsymbol{\alpha} \neq 0$, such that $\boldsymbol{\alpha}^T \mathbf{M}^1 \geq c$, where

$$\mathbf{M}^1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Since c is required to be nonnegative, this immediately implies that $c = 0$.

Each of the following edge/commodity activations is represented in the set $\mathcal{S}$:

$$\begin{aligned} &\{(e_{21}, v_1), (e_{54}, v_1)\}, \\ &\{(e_{51}, v_1), (e_{23}, v_1)\}, \\ &\{(e_{21}, v_1), (e_{34}, v_1)\}, \\ &\{(e_{51}, v_1), (e_{34}, v_1)\}, \\ &\{(e_{32}, v_1), (e_{45}, v_1)\}, \\ &\{(e_{32}, v_1)\}, \\ &\{(e_{45}, v_1)\}. \end{aligned}$$

When we consider the backpressure service vectors and queue backlog service associated with these activations, the first SMLoP conditions (3) require the existence of $\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$, $\boldsymbol{\alpha} \neq 0$, such that $\boldsymbol{\alpha}^T \mathbf{M}^2 + \boldsymbol{\beta}^T \mathbf{M}^3 \leq 0$, where

$$\mathbf{M}^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & 1 & -1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{M}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & -1 & 0 \\ 0 & -1 & 1 & 1 & 1 & 1 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & -1 & 0 & -1 \end{bmatrix}.$$

Simple algebraic manipulation (which we forgo) can be used to demonstrate that there exists no such $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Thus, the 5-ring, C_5, fails SMLoP under edge set E and commodity collection J, which implies that C_5 fails OMLoP. ■

We now verify that the OMLoP conditions hold for a class of graphs in which the backpressure-based framework is known to achieve 100% throughput. This class is the *forest of stars*, where every connected component of the network graph is a star graph, consisting of a central node v_0, connected to one or more vertices of degree 1. Under any k-interference model, the star's interference graph is a clique (appearing in Fig. 4 within the intersection region of the chordal and cograph classes). Therefore, only one edge can ever be active at once. Accordingly, a maximal weight edge activation is identical to a *maximum* weight edge activation, thereby achieving 100% throughput. The following lemma shows that OMLoP is satisfied in such graphs. The proof appears in Appendix E.

Lemma 9: The star network graph satisfies OMLoP.

Applying a result similar to [4, Prop. 1], we have the following corollary.

Corollary 1: Every forest of stars satisfies OMLoP.

VIII. Conclusions

The consideration of Local Pooling has the potential to enable efficient distributed operation of wireless networks. However, since previously LoP was studied mostly under the assumptions of single-hop traffic and primary interference, in this paper we focused on its multihop implications. We identified several graph subclasses of the OLoP-Satisfying class and increased the number of known graphs that satisfy LoP by a few orders of magnitude. Using these observations, we showed that increasing the interference degree usually has a positive effect on the performance of simple distributed algorithms. For example, it was proved that under *secondary* interference constraints, a maximal weight scheduling algorithm achieves 100% throughput in chordal network graphs. Moreover, we obtained the LoP conditions for networks with multihop traffic (OMLoP) and showed that they are distinct from the single-hop conditions. Finally, we showed that the class of graphs satisfying the OMLoP conditions is a strict subclass of the OLoP-Satisfying class.

We emphasize that our objective in this paper is to obtain a better *theoretical* understanding of LoP that will assist the development of future algorithms. Hence, although a theoretical contribution has been made, there remain many algorithmic open problems. For example, LoP-based algorithms can partition the network into LoP-satisfying subnetworks or add artificial interference constraints to generate a LoP-satisfying network. Our identification of several LoP-satisfying graph classes that can serve as building blocks for these networks, and the understanding of multihop traffic and interference effects are advances toward such algorithms. For instance, one can now develop algorithms that add artificial edges to the interference graph to yield a chordal graph.

Moreover, there are a number of theoretical issues that remain unresolved. For example, the analysis of multihop interference under multihop traffic requires further investigation. Additionally, Lemma 6 demonstrates that further study is necessary to determine the general evolution of the LoP property with varying interference degree. Finally, the complete characterization of the OLoP-Satisfying and the OMLoP-Satisfying graph classes is a subject for further research.

Acknowledgments

This work was supported by a Marie Curie International Fellowship within the 6th European Community Framework Programme, by NSF ITR grant CCR-0325401, by ONR grant number N000140610064, and by DARPA/ AFOSR through the University of Illinois grant no. F49620-02-1-0325.

Appendix A
Proof of Theorem 2

In order to prove the theorem, we prove the following lemmas. In the proofs we denote the set of neighbors of node v by $N(v)$.

Lemma A.1: Any chordal graph satisfies OLoP.

Proof: First, we prove that any chordal graph satisfies SLoP. It was shown in [21] that any chordal graph G has at least one vertex v for which $N(v)$ is a clique in G (such a vertex is called a simplicial vertex). Since $N(v)$ is a clique, any maximal independent set in a chordal interference graph G_I will include either the simplicial vertex v or exactly one of the nodes in $N(v)$. Consequently, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the simplicial vertex v and to the vertices in $N(v)$, where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Thus, any chordal graph satisfies SLoP.

According to [17], any induced subgraph (with respect to node removal) of a chordal graph has a simplicial vertex. Hence, by using the vector $\boldsymbol{\alpha}$ described above, we find that any induced subgraph of a chordal graph satisfies SLoP. Therefore, any chordal graph satisfies OLoP. ■

Lemma A.2: Any chordal bipartite graph satisfies OLoP.

Proof: Following [3], given a bipartite graph B, we define $(u, v) \in E$ as a bisimplicial edge, if $N(u) \cup N(v)$ induces a *complete* bipartite subgraph in B. It was shown in [12] that if a graph B is chordal bipartite, any induced subgraph (with respect to node removal) B' has a bisimplicial edge. Therefore, we have to show that a bipartite graph B' which has a bisimplicial edge satisfies SLoP. This will establish that a any chordal bipartite graph satisfies OLoP.

Denote by (u, v) the bisimplicial edge of B' and assume that there exists an independent set in B' that does not include nodes u and v. Such an independent set must include a neighbor of u and a neighbor of v. However, since $N(u) \cup N(v)$ induces a *complete* bipartite subgraph, an independent set cannot include nodes from both $N(u)$ and $N(v)$, and therefore, such an independent set cannot exist. This contradicts the assumption. Therefore, any independent set must include at least one node from u and v. Since u and v are adjacent, an independent set can include either u or v. Consequently, the vector $\boldsymbol{\alpha}$ having all zero entries except at the indices corresponding to the nodes of the bisimplicial edge (u, v), where the entries are set to 1, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Thus, any bipartite graph that has a bisimplicial edge satisfies SLoP. ■

Lemma A.3: Any cograph satisfies OLoP.

Proof: In every *induced subgraph* of a cograph, the intersection of any maximal clique and any maximal independent set contains precisely one vertex [3]. Hence, given an induced subgraph of a cograph, there is a maximal clique that has an active node in each independent set. Consequently, the vector $\boldsymbol{\alpha}$ having entries of one at the indices corresponding to nodes in this clique and having entries of zero otherwise, yields $\boldsymbol{\alpha}^T\mathbf{M}(V_I) = \mathbf{e}^T$. Therefore, SLoP holds for all the induced subgraphs of a cograph and OLoP holds for any cograph. ■

Lemmas A.1-A.3 establish the first part of the Theorem. The second part is established by the following lemma.

Lemma A.4: All even cycles C_n with $n \geq 6$ fail SLoP.

Proof: For the 6-ring interference graph, denoted by $C_6 = (V_6, E_6)$, it was shown in [10] that there is no $\boldsymbol{\alpha} \geq 0, c > 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}(V_6) = c\mathbf{e}^T$. Consider $n \geq 8$, with n even. We label the nodes of the n-ring, $C_n = (V_n, E_n)$, using $v_1, v_2, \ldots, v_n$. Then, the following are valid maximal independent sets

$$\{v_1, v_3, v_5, \ldots, v_{n-7}, v_{n-4}, v_{n-2}\} \tag{5}$$

$$\{v_1, v_3, v_5, \ldots, v_{n-7}, v_{n-4}, v_{n-1}\} \tag{6}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-4}, v_{n-2}, v_n\} \tag{7}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-4}, v_{n-1}\} \tag{8}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-3}, v_{n-1}\} \tag{9}$$

$$\{v_2, v_4, v_6, \ldots, v_{n-6}, v_{n-3}, v_n\} \tag{10}$$

From the requirement of $\boldsymbol{\alpha}^T\mathbf{M}(V_n) = c\mathbf{e}^T$ under the n-ring $C_n = (V_n, E_n)$, we draw the following conclusions. Equations (5) and (6) imply $\alpha_{n-2} = \alpha_{n-1}$. Combining this fact with (7) and (8) yields $\alpha_n = 0$. Finally, combining the fact that $\alpha_n = 0$ with (9) and (10) provides $\alpha_{n-1} = 0$. Thus, it is without loss of generality that we discard the two rows of $\mathbf{M}(V_n)$ corresponding to nodes v_{n-1}, v_n.

We now claim that the remaining rows of $\mathbf{M}(V)$ provide all the constraints corresponding to the $(n-2)$-ring. Consider any independent set of the n-ring containing node v_1 and node v_{n-1}. Note that this configuration mimics the $(n-2)$-ring by disallowing node v_{n-2} to be active simultaneously with v_1. Thus, all maximal independent sets of this type in the n-ring are maximal in the $(n-2)$ ring, and it can be easily seen that all $(n-2)$-ring maximal independent sets containing v_1 yield maximal independent sets in the n-ring when v_{n-1} is active. Further, consider any maximal independent set of the n-ring containing node v_2 and node v_n. Similar reasoning to above provides that all maximal independent sets in the $(n-2)$-ring containing v_2 are represented under this configuration. Finally, consider any maximal independent set of the n-ring containing nodes v_3, v_{n-2}, v_n. Again, it can be easily shown that all maximal independent sets in the $(n-2)$-ring containing v_3 and v_{n-2} are represented. This completes the characterization of all maximal independent sets of the $(n-2)$-ring, since each independent set in C_{n-2} contains either v_1 or v_2, or contains both v_3 and v_{n-2}. Thus, it must be true that the matrix of maximal independent sets of C_{n-2}, $\mathbf{M}(V_{n-2})$, is a submatrix of that of C_n, $\mathbf{M}(V_n)$.

Since $\alpha_{n-1} = \alpha_n = 0$, the existence of $\boldsymbol{\alpha} \geq 0$ and $c > 0$ such that $\boldsymbol{\alpha}^T\mathbf{M}(V_n) = c\mathbf{e}^T$ implies that

$$(\alpha_1, \ldots, \alpha_{n-2})\mathbf{M}(V_{n-2}) = c\mathbf{e}^T.$$

Applying this reasoning inductively, if the SLoP condition for any n-ring having $n \geq 8$ and n even is satisfied, then SLoP must be satisfied for the 6-ring. This is a contradiction and we conclude that every n-ring fails SLoP for $n \geq 8$ and n even. ■

The third part of the theorem holds because if a bipartite graph is not weakly chordal (i.e. not chordal bipartite), it includes an even cycle C_n with at least 6 nodes. This cycle is an induced subgraph that according to Theorem A.4 does not satisfy SLoP. Hence, OLoP fails in bipartite graphs that are not weakly chordal.

Appendix B
Proof of Theorem 3

In Lemma A.4 it was shown by contradiction that every n-ring fails SLoP for $n \geq 8$ and n even. The proof for $C_n = (V_n, E_n)$, $n \geq 9$ and n odd is based on a similar idea. First, the matrix of maximal independent sets for the 9-ring $C_9 = (V_n, E_n)$ is a characterized:

$$\mathbf{M}(V_9) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Using the same node labeling described above, we study the equation $\boldsymbol{\alpha}^T M(V_9) = c\mathbf{e}^T$. Columns 1 and 2 of $\mathbf{M}(V_9)$ imply $\alpha_7 = \alpha_8$. Columns 2 and 3 imply $\alpha_5 = \alpha_6$. Columns 3 and 4 imply $\alpha_3 = \alpha_4$. Columns 4 and 6 imply $\alpha_1 = \alpha_2$. Columns 6 and 7 imply $\alpha_8 = \alpha_9$. Columns 7 and 8 imply $\alpha_6 = \alpha_7$. Columns 8 and 9 imply $\alpha_4 = \alpha_5$. Columns 9 and 11 imply $\alpha_2 = \alpha_3$. Thus, all values α_i must be equal. But, note that columns 11 and 12 imply $\alpha_5 + \alpha_7 = \alpha_6$, which must give $\alpha_5 = 0$, and consequently $\alpha_i = 0$ for all i. We conclude that the 9-ring C_9 fails SLoP.

The remainder of the proof demonstrating that all rings C_n, for $n \geq 9$, with n odd fail SLoP follows identically to the even case considered in Lemma A.4, by reducing any such case to the 9-ring SLoP condition, which is not satisfied.

Appendix C
Proof of Lemma 7

Let $E \subseteq E_N$, with $E \neq \emptyset$. Consider any $J_E \in \mathcal{J}_E$, and suppose $J_E = (J_e^{\tilde{\mathbf{Q}}}, e \in E_N)$ for $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$. Item 1 follows because the set $J_e^{\tilde{\mathbf{Q}}}$ can never be empty. Item 2 follows by the definition of $J_e^{\tilde{\mathbf{Q}}}$. For Item 3, suppose that graph G_j contains a directed cycle, $v_1 \to v_2 \to \cdots \to v_L \to v_1$. Then since $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, it must be true that $\tilde{Q}_{v_i,j}$ 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 $\tilde{Q}_{\sigma(e),j} - \tilde{Q}_{\tau(e),j}$ exceeds the corresponding value across edges in the path joining v_1, v_2 in G_j. This violates that $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, 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 $\tilde{Q}_{\sigma(e),j} - \tilde{Q}_{\tau(e),j}$. 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 $\tilde{\mathbf{Q}} \in \mathcal{Q}_E$, which provides a contradiction. Item 5a follows by noting that $\tilde{Q}_{j,j} = 0$, 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 D
Proof of Theorem 6

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, $A_{i,j}(t), i,j \in \{1,\dots,n\}$, for $t \geq 0$. The queueing variables of the r-th system are given by $Q^r_{i,j}(t), A^r_{i,j}(t) = A_{i,j}(t), U^r_{i,j}(t)$ for all $i,j \in \{1,\dots,n\}$, and $F^r_{\mathbf{S}}(t)$ for all $\mathbf{S} \in \mathcal{S}$. At time $t = 0$, the r-th system is assumed to contain a total of 0 packets in queue. The following are the queue evolution properties of the r-th system:

$$\begin{aligned}
Q^r_{i,j}(t) &= A^r_{i,j}(t) - U^r_{i,j}(t), \quad t \geq 0 \\
U^r_{i,j}(t) &= \sum_{\mathbf{S} \in \mathcal{S}} d_{i,j}(\mathbf{S}) F^r_{\mathbf{S}}(t), \quad t \geq 0 \\
\sum_{\mathbf{S} \in \mathcal{S}} F^r_{\mathbf{S}}(t) &= t, \text{ and } F_{\mathbf{S}} \text{ is non-decreasing}, \quad t \geq 0 \\
A^r_{i,j}(0) = 0, U^r_{i,j}(0) &= 0, \forall i,j,\ F^r_{\mathbf{S}}(0) = 0, \forall \mathbf{S} \in \mathcal{S}
\end{aligned}$$

We extend the queueing variables to the reals using $Y(t) = Y(\lfloor t \rfloor)$ for $Y = Q^r_{i,j}, A^r_{i,j}, U^r_{i,j}, F^r_{\mathbf{S}}$. Now each of these processes is scaled according to $q^r_{i,j}(t) = Q^r_{i,j}(rt)/r$. We obtain the scaled processes $q^r_{i,j}, a^r_{i,j}, u^r_{i,j}, f^r_{\mathbf{S}}$. As in [**?**], we can infer the convergence with probability 1 of the scaled processes over some subsequence of system indices $\{r_k\}$ to a *fluid limit* $(q_{i,j}, a_{i,j}, u_{i,j}, f_{\mathbf{S}})$ having the following key properties:

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

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

Consider $z_{e,j}(t) = q_{\sigma(e),j}(t) - q_{\tau(e),j}(t)$, the fluid *differential backlog* of commodity j across the directed link e. Define the function $h : [0,\infty) \to [0,\infty)$ where $h(t) = \max_{e,j} z_{e,j}(t)$. Consider a regular time[8] $t \geq 0$, at which $h(t) > 0$. Assign

$$E = \{e \in E_N : \exists j \text{ such that } z_{e,j}(t) = h(t)\}, \tag{11}$$

and for $e \in E_N$, assign $J_e = \arg\max_j z_{e,j}(t)$. Note that using $\tilde{\mathbf{Q}} = (q_{i,j}(t), i,j \in V_N)$, 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,

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

Assuming an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{i,j}, i,j \in V_N)$, we have for $e \in E$ and $j \in J_e$,

$$\begin{aligned}
&\dot{z}_{e,j}(t) \\
&\quad = \lambda_{\sigma(e),j} - \lambda_{\tau(e),j} - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e),j}(\mathbf{S}) - d_{\tau(e),j}(\mathbf{S})) \\
&\quad = \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}}(d_{\sigma(e),j}(\mathbf{S}) - d_{\tau(e),j}(\mathbf{S})) \\
&\qquad - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t)(d_{\sigma(e),j}(\mathbf{S}) - d_{\tau(e),j}(\mathbf{S})) \\
&\quad = \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} u_{e,j}(\mathbf{S}) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{e,j}(\mathbf{S}) \qquad (12)
\end{aligned}$$

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.

Lemma D.1: Let $t \geq 0$ be a regular time at which $h(t) > 0$. Let $E \subseteq E_N$ satisfy (11) and $J_e = \arg\max_j z_{e,j}(t)$ for each $e \in E_N$. Suppose that the solution θ^* to the following optimization problem is $\theta^* \leq 0$:

$$\begin{aligned}
\text{Maximize} \quad & \theta && (13) \\
\text{Subject to} \quad & \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) \geq \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \nu_{\mathbf{S}} \mathbf{u}_{E,J}(\mathbf{S}) + \theta \mathbf{e} \\
& \mathbf{e}^T \boldsymbol{\mu} \leq 1 \\
& \sum_{\mathbf{S} \in \mathcal{S}} \mu_{\mathbf{S}} \mathbf{R}^j_{i,\cdot} \mathbf{S}_{\cdot,j} \geq 0 \quad i,j = 1,\dots,n && (14) \\
& \mathbf{e}^T \boldsymbol{\nu} = 1 && (15) \\
& \mu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S} \\
& \nu_{\mathbf{S}} \geq 0 \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} && (16)
\end{aligned}$$

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

Proof: Suppose $\theta^* \leq 0$. For an admissible arrival rate vector $\boldsymbol{\lambda} = (\lambda_{i,j}, i,j \in V_N)$, we have $\lambda_{i,j} = \sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} d_{i,j}(\mathbf{S}) \geq 0$, 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 (13). The solution $\theta^* \leq 0$ in the optimization clearly implies that there must exist $e \in E$ and $j \in J_e$ such that

$$\sum_{\mathbf{S} \in \mathcal{S}} \phi_{\mathbf{S}} u_{e,j}(\mathbf{S}) - \sum_{\mathbf{S} \in \mathcal{S}_{E,J}} \dot{f}_{\mathbf{S}}(t) u_{e,j}(\mathbf{S}) \leq 0. \tag{17}$$

[8] 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.

By (12), equation (17) implies that $\dot{z}_{e,j}(t) \leq 0$. Since t is a regular time, $\dot{z}_{e,j}(t) = \dot{h}(t)$, which provides $\dot{h}(t) \leq 0$, as desired. ■

It only remains to demonstrate that the multihop local pooling conditions (3)-(4) are sufficient for stability. The following lemma demonstrates this property by studying the dual optimization problem to that in (13).

Lemma D.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 (13) has solution $\theta^* \leq 0$.

Proof: Suppose that the optimization (13) has solution $\theta^* \leq 0$. This implies that there exists a dual solution and complementary slackness conditions hold. It is a simple exercise to demonstrate that the dual problem to (13) is:

$$\begin{aligned} \text{Minimize} \quad & c_1 + c_2 && (18) \\ \text{Subject to} \quad & \boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c_1, \quad \forall \mathbf{S} \in \mathcal{S} \\ & \boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq -c_2, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J} \\ & \mathbf{e}^T \boldsymbol{\alpha} = 1 \\ & \boldsymbol{\alpha}, \boldsymbol{\beta}, c_1 \geq 0 \end{aligned}$$

Since the solution to (13) 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 $(\boldsymbol{\alpha}/(\mathrm{e}^T \boldsymbol{\alpha}), \boldsymbol{\beta}, c, -c)$ is a feasible point in the dual optimization problem (18). This feasible point has cost 0. By duality, this implies that the primal problem must attain a solution $\theta^* \leq 0$, as desired. ■

Combining Lemmas D.1 and D.2, we conclude that if SMLoP is satisfied for any $E \subseteq E_N$, with 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 [9, Lemma 1], Lemma D.1 allows us to conclude that $h(t) = 0$ for almost every $t \geq 0$. This immediately implies that $q_{i,j}(t) = 0$ for almost every $t \geq 0$, which gives the rate stability of the backpressure based algorithm. Thus the OMLoP conditions are sufficient for stability, as desired.

Appendix E
Proof of Lemma 9

Consider a set of edges $E \subseteq E_N$ and the commodity collection $J = (J_e, e \in E_N) \in \mathcal{J}_E$. Consider the commodity graph $G_j = (V_N, E_j)$, where $E_j = \{e \in E : j \in J_e\}$. By the definition of $\mathcal{J}_E$, there *can not* exist two oppositely directed edges $(v, v'), (v', v)$ in E_j for all j. Graph $G_j = (V_N, E_j)$ is a star having $k \geq 0$ edges facing outward from v_0 and $l \geq 0$ edges facing inwards to v_0, with $k + l \geq 1$, as depicted in Fig. 10.

For the proof, we will use the value $c = 1$. Recall that only a single edge in the star can ever be active at one time. Thus, if we arrange in a matrix the backpressure service vectors corresponding to all $\mathbf{S} \in \mathcal{S}$, the columns of the matrix can be arranged to yield a block diagonal matrix $\mathbf{U}$, with each block corresponding to service activations involving different commodities. We will consider each commodity $j \in \cup_{e \in E} J_e$ in turn and determine the required assignment of the elements of $\boldsymbol{\alpha}$ for j.

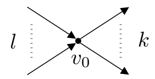


Fig. 10. Graph G^j of edges carrying commodity j in the commodity collection J_E.

Consider commodity $j \in \cup_{e \in E} J_e$:

Case 1. Suppose that $v_0 = j$. Then by the definition of $\mathcal{J}_E$, we must have $k = 0$. In this case, if edge $e \in E_j$ is selected for service of commodity j, link e sees a decrease in backpressure of 1 commodity j packet, and no other of the l links sees a change in backpressure, since the packet departs at v_0. If any other edge not in E_j is selected for service of commodity j packets to v_0, no change in backpressure occurs for any of the l links. Thus, the non-zero component of the block-diagonal matrix corresponding to commodity j is an identity matrix. In this case we assign $\boldsymbol{\alpha}_{e,j} = 1$ for all (e, j) where $e \in E_j$. We also assign $\boldsymbol{\beta}_{v,j} = 0$ for all v.

Case 2. Suppose that $v_0 \neq j$ and that none of the k outward-facing links terminates at node j. In this case, if an outward-facing edge $e \in E_j$ is selected for service of commodity j, e sees a service of 2 units, each of the other outward facing edges in E_j sees a service of 1 unit, and each of the l inward-facing edges sees a service of -1 units. Similarly, if an inward-facing edge $e \in E_j$ is selected for service of commodity j, e sees a service of 2 units, each of the other inward-facing edges in E_j sees a service of 1 unit, and each of the k outward-facing edges sees a service of -1 units. If any other edge e not in E_j is selected for service of commodity j, this leads to a service of 1 at all links facing v_0 in the same direction as e and a service of -1 at all links facing v_0 in the opposite direction to e. The non-zero component of the block-diagonal matrix corresponding to commodity j has the form,

$$\left[\begin{array}{cc|cc} \mathbf{I}_k + \mathbf{e}_{k,k} & -\mathbf{e}_{k,l} & \mathbf{e}_{k,1} & -\mathbf{e}_{k,1} \\ -\mathbf{e}_{l,k} & \mathbf{I}_l + \mathbf{e}_{l,l} & -\mathbf{e}_{l,1} & \mathbf{e}_{l,1} \end{array} \right], \qquad (19)$$

where $\mathbf{I}_p$ is the identity matrix of size p, and $\mathbf{e}_{p,q}$ is the $p \times q$ matrix of ones. The separator in (19) separates the activations in $\mathcal{S}_{E,J}$ (at left) from the remaining commodity j edge activations (at right). The rightmost two columns of (19) may or may not exist and there may be multiple copies of either column. Also these columns can dominate other inferior service vectors. In this case, we set $\boldsymbol{\alpha}_{e,j} = (2l+1)/(k+l+1)$ for each $e \in E_j$ facing outwards from v_0, and set $\boldsymbol{\alpha}_{e,j} = (2k+1)/(k+l+1)$ for each $e \in E_j$ facing inwards to v_0. It can be verified that for $k, l \geq 0$ with $k + l \geq 1$, the inner product of $\boldsymbol{\alpha}$ with the leftmost columns before the separator in (19) yields 1, while the remaining nonzero columns result in values less than 1. We assign $\boldsymbol{\beta}_{v,j} = 0$ for all v.

Case 3. Suppose one of the k outward-facing links terminates at node j. Through similar analysis as above, we obtain the non-zero component of the block-diagonal matrix corresponding to commodity j as,

$$\left[\begin{array}{ccc|cc} \mathbf{I}_{k'}+\mathbf{e}_{k',k'} & \mathbf{e}_{k',1} & -\mathbf{e}_{k',l} & \mathbf{e}_{k',1} & -\mathbf{e}_{k',1} \\ \mathbf{e}_{1,k'} & 1 & -\mathbf{e}_{1,l} & 1 & -1 \\ \mathbf{e}_{l,k'} & \mathbf{e}_{l,1} & \mathbf{I}_{l,l}+\mathbf{e}_{l,l} & -\mathbf{e}_{l,1} & \mathbf{e}_{l,1} \end{array}\right], \quad (20)$$

where $k' = k-1$. Note that (20) only differs from (19) in one column to the left of the separator, where the 2 is replaced by a 1. This corresponds to the edge whose destination is j. We assign $\boldsymbol{\alpha}_{e,j} = 2$ for each of the inward-facing links, and $\boldsymbol{\alpha}_{e,j} = (1+2l)/k$ for each of the outward-facing links. In this case, the inner product of $\boldsymbol{\alpha}$ with the first $k-1$ columns of (20) yields $1 + (1+2l)/k$, and the remaining columns to the left of the separator yield 1. Since we seek the value $c = 1$, the values $1 + (1+2l)/k$ are too high to satisfy (3). Consequently, we assign $\boldsymbol{\beta}_{v,j} = (1+2l)/k$ for all vertices v terminating the k outward-facing edges. Thus, activation of any one of these edges leads to a contribution of the $\boldsymbol{\beta}$ term in (3) of $-(1+2l)/k$, leading to satisfaction of (3) as desired.

For every commodity j not belonging to $\cup_{e\in E} J_e$ we assign $\boldsymbol{\alpha}_{e,j} = 0$ for all e, and $\boldsymbol{\beta}_{v,j} = 0$ for all v. The vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are then guaranteed to satisfy SMLoP, as desired. Since this holds for any $E \in E_N$, and any $J \in \mathcal{J}_E$, OMLoP is satisfied.

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. Inf. Theory*, Vol. 45, No. 3, pp. 1167–1174, Mar. 2005.

[2] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M.V. Marathe, and S. Thite,“The Distance-2 Matching Problem and Its Relationship to the Mac-Layer Capacity of Ad Hoc Wireless Networks”, *IEEE J. Sel. Areas Commun.*, Vol. 22, No. 6, pp. 1069–1079, Aug. 2004.

[3] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: A Survey*, SIAM, 1999.

[4] A. Brzezinski, G. Zussman, and E. Modiano, “Enabling Distributed Throughput Maximization in Wireless Mesh Networks - A Partitioning Approach” *To Appear in Proc. ACM MOBICOM’06*, Sep. 2006.

[5] K. Cameron, “Induced Matchings”, *Disc. Appl. Math.*, Vol. 24, pp. 97–102, 1989.

[6] K. Cameron, “Induced Matchings in Intersection Graphs”, *Disc. Math.*, Vol. 278, pp. 1–9, 2004.

[7] P. Chaporkar, K. Kar, and S. Sarkar, “Throughput Guarantees Through Maximal Scheduling in Wireless Networks”, *Proc. Allerton Conf. on Commun., Control, and Comp.*, Sep. 2005.

[8] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Optimal Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks”, *Proc. IEEE INFOCOM’06*, Apr. 2006.

[9] J. Dai and B. Prabhakar, “The Throughput of Data Switches with and without Speedup” *Proc. IEEE INFOCOM’00*, Mar. 2000.

[10] A. Dimakis and J. Walrand, “Sufficient Conditions for Stability of Longest Queue First Scheduling: Second Order properties using Fluid Limits”, *Adv. of Appl. Prob.*, Vol. 38, No. 2, pp. 505–521, June 2006.

[11] M. C. Golumbic, D. Rotem, and J. Urrutia, “Comparability Graphs and Intersection Graphs”, *Discrete Math.*, Vol. 43, pp. 37-46, 1983.

[12] M. C. Golumbic and C. F. Goss, “Perfect Elimination and Chordal Bipartite Graphs”, *J. Graph Theory*, Vol. 2, pp. 155–163, 1985.

[13] B. Hajek and G. Sasaki, “Link Scheduling in Polynomial Time”, *IEEE Trans. Inf. Theory*, Vol. 34, No. 5, pp. 910–917, Sep. 1988.

[14] F. Harary. *Graph Thoery*, Addison-Wesley, Reading, MA, 1969.

[15] X. Lin and S. Rasool, “Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks”, *Technical Report, Purdue*, 2006.

[16] X. Lin and N.B. Shroff, “The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Wireless Networks”, *IEEE/ACM Trans. Net.*, Vol. 14, No. 2, pp. 302-315, Apr. 2006.

[17] T. A. McKee and F. R. McMorris, *Intersection Graph Theory*, SIAM, 1999.

[18] 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.

[19] E. Modiano, D. Shah, and G. Zussman, “Maximizing Throughput in Wireless Networks via Gossiping”, *Proc. ACM SIGMETRICS’06*, June 2006.

[20] 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.

[21] D. J. Rose, “Tringulated Graphs and the Elimination Process”, *J. Math. Analys. Appl.*, Vol. 32, pp. 597–609, 1970.

[22] S. Sarkar, P. Chaporkar, and K. Kar, “Fairness and Throughput Guarantees with Maximal Scheduling in Multihop Wireless Networks” *Proc. WiOpt’06*, Apr. 2006.

[23] L. Tassiulas and A. Ephremides, “Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks”, *IEEE Trans. Autom. Control*, Vol. 37, No. 12, pp. 1936-1948, Dec. 1992.

[24] X. Wu and R. Srikant, “Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks With Node-Exclusive Spectrum Sharing”, *Proc. IEEE CDC-ECC’05*, Dec. 2005.

[25] X. Wu and R. Srikant, “Bounds on the Capacity Region of Multi-Hop Wireless Networks under Distributed Greedy Scheduling”, *Proc. IEEE INFOCOM’06*, Apr. 2006.

[26] Y. Yi and S. Shakkottai, “Hop-by-hop Congestion Control over a Wireless Multi-hop Network”, *Proc. IEEE INFOCOM’04*, Mar. 2004.