

# Spectrum Allocation for Distributed Throughput Maximization under Secondary Interference Constraints in Wireless Mesh Networks

Tong Shu<sup>1,2</sup>, Min Liu<sup>1</sup>, Zhongcheng Li<sup>1</sup>

<sup>1</sup>Institute of Computing Technology, Chinese Academy of Sciences

<sup>2</sup>Graduate University of Chinese Academy of Sciences  
Beijing 100190, P.R. China

**Abstract**—Secondary interference constraints are important, because of representing the transmission constraints of the widespread and promising IEEE 802.11 wireless technology. Under secondary interference constraints, distributed link scheduling algorithms for multihop wireless networks can only achieve a fraction of the maximum possible throughput in general, but distributed Greedy Maximal Scheduling (GMS) algorithms can achieve optimal throughput in some network graph structures. It is possibly helpful for the improvement of distributed throughput to partition a network into subnetworks such that the subnetwork assigned to each frequency channel achieves distributed throughput maximization. In this paper, we investigate the structure characteristics of the subnetwork in which GMS achieves optimal throughput under secondary-interference constraints, and define a type of network subgraph structures meeting the requirement — special chordal subgraphs. Based on this, we propose a channel assignment algorithm, including a network partitioning algorithm and a topology balancing algorithm. By simulation, we evaluate the achievable throughput and fairness in a distributed manner using our algorithm, in comparison with the existing Max K-cut based channel assignment algorithm.

**Keywords**—channel assignment; greedy maximal scheduling; network partitioning

## I. INTRODUCTION

We consider how to improve the distributed throughput in multi-channel wireless mesh networks. The backbone of a wireless mesh network is typically represented by a network graph  $G(V, E)$ , where  $V$  denotes the set of  $n$  nodes in the network and  $E$  denotes a set of  $m$  bi-directional wireless links among them. Suppose that each node has multiple radios. Generally, the interference between wireless links can be summarized in an interference graph  $G_I = (V_I, E_I)$  based on the network graph  $G$ . Here,  $V_I = E$  and an edge  $(v_i, v_j)$  in the interference graph indicates the conflict between corresponding links  $e_i$  and  $e_j$  in the network graph. In particular, the  $k$ -hop interference models imply that each pair of simultaneously active links must be separated by at least  $k$  hops. In graph theoretic terminology, the interference graph resulting from the 1-hop interference model is called a line graph and the interference graph resulting from the 2-hop interference model is called the square of a line graph [1].

Assume that time is slotted with normalized slots  $t \in \{0, 1, 2, \dots\}$ , and arrivals of packets in all time slots are mutually independent and identically distributed random processes. Let

the column vector  $\lambda = (\lambda_e, e \in E)$  denote the arrival rate vector, where  $\lambda_e$  is the arrival rate of packets arriving at link  $e$ . Let  $\Pi(G)$  denote the set of all feasible link activations in the network graph  $G$ .  $\pi = (\pi_e, e \in E) \in \Pi(G)$  is a 0-1 column vector representing a possible link activation. An arrival rate vector  $\lambda$  is called admissible, if there exists a collection of link activations,  $\pi_l, 1 \leq l \leq L$ , such that  $\lambda = \sum_{l=1}^L \alpha_l \pi_l, \alpha_l \geq 0$  and  $\sum_{l=1}^L \alpha_l < 1$ . The set of all admissible rate vectors is called the stability region.

Let  $S(t) = (S_e(t), e \in E)$  denote the scheduling decision vector, where  $S_e(t)$  is a 0-1 variable indicating whether link  $e$  is active at time slot  $t$ . For any time slot  $t$ ,  $S(t) \in \Pi(G)$  must hold. Let  $Q(t) = (Q_e(t), e \in E)$  denote the queue-size vector, where  $Q_e(t)$  is the number of packets waiting to be transmitted on link  $e$  at the beginning of time slot  $t$ . A scheduling algorithm is stable (i.e. achieves optimal throughput), if for any admissible  $\lambda$ , the Markov chain  $(Q(t), t \geq 0)$  is positive recurrent.

There exists a stable scheduling algorithm [2] according to  $S(t) = \arg \max_{\pi \in \Pi(G)} Q'(t)\pi$ , where  $Q'(t)$  is the transpose of vector  $Q(t)$ . Given an interference graph  $G_I$ , the algorithm has to find the maximum weight independent set in each time slot, so the algorithm is called a maximum weight independent set (MWIS) scheduling algorithm. Under the  $k$ -hop interference model ( $k \geq 2$ ), the problem that the centralized MWIS algorithm has to solve in every time slot is an NP-Complete problem so that it is difficult to be put into practice.

In general, a distributed greedy maximal scheduling (GMS) algorithm is used to replace the MWIS algorithm to obtain a suboptimal schedule. GMS repeatedly selects link  $l$  satisfying  $l = \arg \max_{e \in E} Q_e(t)$  in each time slot  $t$ , until selected links compose a maximal independent set in  $G_I$ . Unfortunately, a distributed GMS algorithm can only achieve a fraction of the maximum achievable throughput in the worst case [3], [4]. However, when  $G_I$  meet the overall local pooling (OLOP) conditions, GMS can obtain the maximum weight independent set in  $G_I$  and achieves optimal throughput [5]. An interference graph  $G_I$  satisfies **Overall Local Pooling** (OLOP) if each induced subgraph over the vertices  $V \subseteq V_I$  satisfies the subgraph local pooling conditions. An interference graph  $G_I$  satisfies **Subgraph Local Pooling** (SLOP), if there exist  $\alpha \in R_+^{|\mathcal{I}|}$  and  $c > 0$  such that  $\alpha' M(V_I) = ce'$ , where  $M(V_I)$  is a 0-1 matrix that includes all the maximal independent sets in  $G_I$  [5].

Under  $k$ -hop interference models, the interference graphs of some special classes of network graph structures just meet the OLOP conditions [6]. Partitioning the network into subnetworks with such structures and assigning them to nonoverlapping channels are likely to improve distributed throughput in a

wireless mesh network. Under primary interference constraints (i.e. the 1-hop interference model), GMS achieves optimal throughput in any tree network graph and any complete bipartite graph of size up to  $2 \times n$ . Based on the tree network components, existing channel assignment algorithms partition a network graph into several forests, and enables the subnetwork in each channel to achieve distributed throughput maximization under primary interference constraints [6], [7].

Under secondary interference constraints (i.e. the 2-hop interference model), the worst-case efficiency ratio of GMS in geometric network graphs is not more than 1/3, although the efficiency ratio of GMS is at least 1/2 under primary interference constraints [8]. However, GMS achieves optimal throughput in a network of the chordal graph structure [9] and in a network with 8 nodes or less [10] under the 2-hop interference model. Hence, network partitioning is possibly more helpful to improve the distributed throughput under secondary interference constraints. However, none of the existing channel assignment algorithms partition a network into subnetworks to enable distributed throughput maximization under secondary-interference constraints. Network partitioning under the 2-hop interference model faces new problems and is a great challenge. Firstly, we cannot simply partition a network graph into forests under the 2-hop interference model, because there may be interference between different trees in the same subnetwork. Secondly, under the 2-hop interference model unlike the 1-hop interference model, GMS may not achieve optimal throughput in a subnetwork with some structures, although it achieves optimal throughput in a whole network with the same structures.

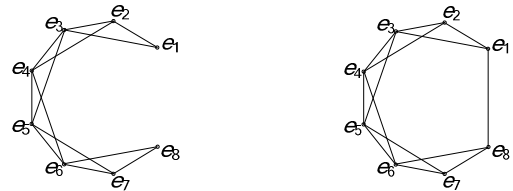
The main **contributions** of this paper are as follows. (1) We investigate the characteristics of subnetwork structures in which GMS achieves optimal throughput under secondary-interference constraints, and define a type of such subnetwork structures — special chordal subgraphs. (2) Based on the type of subnetwork structures, we present a network partitioning approach for a minimum number of subnetworks and topology balancing algorithm for approximately equivalent contention on each channel.

The rest of this paper is organized as follows. Section II provides a sufficient condition that a network subgraph ensures the stability of GMS under the 2-hop interference model. In Section III, we present a channel assignment algorithm, including a network partitioning approach and a topology balancing algorithm. In Section IV, we evaluate the performance of our channel assignment algorithm. Some conclusions are provided in Section V.

## II. NETWORK SUBGRAPH STRUCTURES FOR A STABLE GMS ALGORITHM UNDER SECONDARY INTERFERENCE CONSTRAINTS

Under the 2-hop interference model, the condition for network graph structures that ensure GMS to achieve optimal throughput in a whole network is different from the condition for network graph structures that ensure GMS to achieve optimal throughput in a subnetwork. We take the example of  $L_8$ , which is a chain composed of 8 links. If the network graph of a whole network is just  $L_8$ , its interference graph shown as Fig. 1(a) satisfies OLoP according to Theorem 5 of [9], so GMS achieves optimal throughput in  $L_8$ . However, if the network graph of a whole network is a ring network with 9 links, denoted by  $C_9$ , the interference graph of its network subgraph  $L_8$  shown as Fig. 1(b) does not satisfy OLoP. If the traffic loads of all links are the same, the optimal schedule from MWIS is

$\{\{e_1, e_4, e_7\}, \{e_2, e_5, e_8\}, \{e_3, e_6\}\}$ , and a schedule from GMS is likely to be  $\{\{e_1, e_5\}, \{e_2, e_6\}, \{e_3, e_7\}, \{e_4, e_8\}\}$ . This means that GMS only achieves no more than 3/4 of the maximum possible throughput in the network subgraph  $L_8$  in the case. The ultimate reason resulting in the phenomenon is that the interference graphs only based on network subgraphs maybe lose some interference relationships. For ease of exposition, the network graphs of a whole network and a subnetwork are called a network supergraph and a network subgraph, respectively. For a network supergraph  $G$ , its interference graph is the square of the line graph of  $G$ . For a network subgraph  $H = (U, D) \in G$ , its real interference graph is the subgraph induced by a vertex set  $D \subset V_I$ , and includes the square of the line graph of  $H$ . For a network subgraph, its real interference graph and the square of its line graph are generally different. Therefore, GMS may not achieve optimal throughput in  $H$  when the square of the line graph of  $H$  satisfies OLoP.



(a) The interference graph of  $L_8$       (b) The interference graph of  $C_9 / e_9$   
Figure 1. Partitioning the network graph of the 9-node ring,  $C_9$

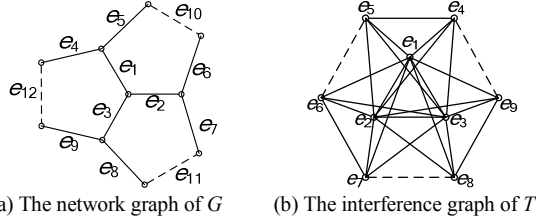
In order to partition a network into subnetworks of distributed throughput maximization, we have to identify some characteristics of network subgraphs in which GMS achieves optimal throughput. In this section, we give a sufficient condition that a network subgraph ensures the stability of GMS under the 2-hop interference model.

**Theorem 1:** If a network subgraph  $H (\subset G)$  satisfies the following two conditions that

- (1)  $H$  is a chordal graph, where a graph is chordal if every cycle of length 4 or more has a chord,
- (2) for any two edges in  $H$ , if their distance is 1 in the supergraph  $G$ , their distance is also 1 in  $H$ ,

a GMS algorithm achieves optimal throughput in  $H$  under the 2-hop interference model.

*Proof:* Let  $H = (U, D)$ . From the condition (2), for any two edges in  $H$ , if their distance is 1 in the supergraph  $G$ , they are in the same connected component of  $H$ . Therefore, the distance between any two edges in the different connected components of  $H$  is more than 1 in  $G$ . That is, there is no interference between any two connected components under the 2-hop interference model. This means that the interference graph of  $H$  is the disjoint union of the interference graphs of all its connected components. There exists interference in two cases. In the first case, two edges are adjacent or their distance is 1 in both  $H$  and  $G$ . In the second case, the distance between two edges is more than 1 in  $H$ , but is 1 in  $G$ . Correspondingly, all edges in the interference graph  $H_I = (U_I, D_I)$  of  $H$  fall into two sets  $I_1(H), I_2(H)$ . From the condition (2),  $I_2(H) = \emptyset$  and  $I_1(H) = D_I$ . Hence,  $H_I$  is the square of the line graph of  $H$ , and is also the subgraph induced by the vertex set  $U_I (\subset V_I)$  from the interference graph  $G_I = (V_I, E_I)$  of  $G$ . From the condition (1),  $H$  is a chordal graph. The square of the line graph of every chordal graph is again chordal [11]. From Theorem 2 of [9], a chordal graph satisfies OLoP. Thus, the interference graph  $H_I$  of  $H$  satisfies OLoP. According to Theorem 1 of [5], GMS achieves optimal throughput in  $H$ .  $\square$



(a) The network graph of  $G$  (b) The interference graph of  $T$   
Figure 2. An instance of a tree network subgraph

The condition (2) in Theorem 1 is not too strong. The class of trees is part of the class of chordal graphs and has some good properties in general. Even in a tree network subgraph  $T$  ( $\subset G$ ), if there are two edges in  $T$  such that their distance is 1 in  $G$  and 2 in  $T$ , GMS may not achieve optimal throughput in  $T$ . We take the tree network subgraph in Fig. 2(a) for example. The tree  $T = \{V, \{e_1, e_2, \dots, e_9\}\}$  is a network subgraph of  $G = \{V, \{e_1, e_2, \dots, e_{12}\}\}$ . The distances between  $e_5$  and  $e_6$ , between  $e_7$  and  $e_8$ , between  $e_4$  and  $e_9$  are all 1 in  $G$  and 2 in  $T$ . The interference graph  $T_I$  of  $T$  shown as Fig. 2(b) does not satisfy OLoP, because the subgraph induced by a vertex subset  $\{e_4, e_5, e_6, e_7, e_8, e_9\}$  from  $T_I$  is a 6-node ring,  $C_6$ , which does not satisfy SLoP. If the traffic loads of all links in  $T$  are the same, the optimal schedule from MWIS is  $\{\{e_4, e_6, e_8\}, \{e_5, e_7, e_9\}, \{e_1\}, \{e_2\}, \{e_3\}\}$ , and a schedule from GMS is likely to be suboptimal, such as  $\{\{e_4, e_7\}, \{e_5, e_8\}, \{e_6, e_9\}, \{e_1\}, \{e_2\}, \{e_3\}\}$ . In the case, GMS only achieves 5/6 of throughput obtained by MWIS.

### III. NETWORK PARTITIONING BASED ON SPECIAL CHORDAL SUBGRAPHS

First of all, we define a **special chordal subgraph** as a subgraph satisfying the conditions in Theorem 1. If it is connected, we name it a special connected chordal subgraph. If a network is partitioned into several special chordal network subgraphs and the edges in different network subgraphs are assigned to nonoverlapping frequency channels, GMS achieves optimal throughput in each subnetwork under the 2-hop interference model. Therefore, in this section, our objective is to design a channel assignment algorithm to provide a high capability based on the network partitioning.

#### A. Partitioning Approach for the Least Subgraphs

In order to meet the limit of the number of channels, the network should be partitioned into as few subnetworks as possible. We design a heuristic algorithm (denoted by Algorithm 1) for a suboptimal solution of the problem. Its major advantage is its low complexity. The idea of Algorithm 1 is constructing a subnetwork with as many edges as possible and then moving on to the next subnetwork. Every time that a subnetwork has been constructed, delete all its edges from the network graph. Consider that given a graph, it is NP-hard to solve its maximum chordal subgraph [12]. Hence, for each subnetwork construction, we select a maximal special chordal subgraph from the residual network graph. In order to look for such a subnetwork, we initially select a maximal special connected chordal subgraph and add it to the current subnetwork. Once a maximal special connected chordal subgraph is found, all the edges incident to its neighbor vertices are immediately forbidden to join the current subnetwork. Like this, we repeatedly select a maximal special connected chordal subgraph from unselected edges in the residual network graph, until no more edges can be added to the current subnetwork.

Here, a key problem is given a supergraph  $G$  and a subgraph  $G'$  of  $G$ , to find a special connected chordal subgraph

of  $G$  which is maximal in  $G'$ . Thus, we presents an algorithm (denoted by Algorithm 1-1) to solve the problem. The basic idea behind Algorithm 1-1 is as follows. Let  $G$  be a supergraph. Initially, pick a vertex of the subgraph  $G'$  of  $G$  and add it to an empty vertex set  $U$ . Let  $D$  be an empty edge set. Then, repeatedly select a new vertex from neighbor vertices of  $U$  in  $G'$ , and add it to  $U$ , until  $U$  has no neighbor vertices in  $G'$ . Every time that a new vertex  $v$  is added, examine all the edges between  $v$  and  $U \setminus v$  in  $G'$ , and add the most edges from them to  $D$  such that the subgraph  $H(U, D)$  satisfies the two conditions of Theorem 1. Ultimately, the output subgraph  $H(U, D)$  is a special connected chordal subgraph of  $G$  which is maximal in  $G'$ .

---

#### Algorithm 1-1 maximal\_special\_connected\_chordal\_subgraph( $G, G', k$ )

---

Input: a supergraph  $G(V, E)$ , a subgraph  $G(V', E')$  of  $G$ , and the  $k$  th channel

Output: a special connected chordal subgraph  $H = (U, D)$  of  $G$  which is maximal in a subgraph  $G'$

Global variables: channel[ $m$ ], an array of  $m$  elements recording the channel of every edge; label[ $n$ ], an array of  $n$  elements indicating whether every node is explored or unexplored.

---

01. Select an unexplored non-isolated vertex  $s$  from  $G'$ . // It is better that  $s$  has a larger degree in two nodes of the edge whose adjacent edges are the most in  $G'$ .
  02.  $U = \{s\}; D = \emptyset; H = (U, D); \text{label}[s] = \text{EXPLORED}$
  03. for all  $w \in N_G(s)$  //  $N_G(s)$  is a set of neighbor vertices of  $s$  in  $G'$
  04.   Label[ $w$ ] = NEIGHBOR\_EXPLORED
  05. Let tree  $T$  be an isolated node //  $T$  is a clique tree of  $H$
  06. while  $U \subset V'$
  07.   Select a neighbor vertex  $u$  of  $U$  from  $V' \setminus U$  // It is better that the edges between  $u$  and vertices of  $U$  is the most in  $G'$ .
  08.    $N = U \cap N_G(u); U = U \cup \{u\}; H = (U, D)$
  09.   for all  $w \in N$
  10.     if  $\{e' \mid e' \in D \text{ and the distance between } e \text{ and } e' \text{ is 1 in } G\} \subset \{\text{edges that are 1-hop away from } (u, w) \text{ in } (U, D \cup \{un \mid n \in N\})\}$
  11.        $N = N \setminus \{w\}$
  12.   do
  13.      $U' = U; D' = D; H' = (U', D')$
  14.      $T' = T$  //  $T'$  is a clique tree of  $H'$
  15.      $N' = N; A = \emptyset$
  16.     while  $N' \neq \emptyset$
  17.       select  $v$  from  $N'$ ;  $N' = N' \setminus \{v\}$  // It is better that  $v$  has the maximum degree in  $H'$ .
  18.        $X = \text{minimal\_separator\_union}(H', T', u, v)$
  19.        $R = \{ux \mid x \in X\}$  // That is  $R(H', u, v)$
  20.       if  $X \subset N'$
  21.          $A = A \cup \{v\} \cup X; N' = N' \setminus X; D' = D' \cup \{uv\} \cup R$
  22.         clique\_tree\_update( $H', T', u, v, X$ )
  23.        $N = A$
  24.       for all  $w \in N$
  25.         if  $\{e' \mid e' \in D \text{ and the distance between } e \text{ and } e' \text{ is 1 in } G\} \subset \{\text{edges which are 1-hop away from } (u, w) \text{ in the subgraph } (U, D \cup \{un \mid n \in N\})\}$
  26.          $N = N \setminus \{w\}$
  27.       while  $N \subset A$
  28.          $U = U'; D = D'; H = (U, D); T = T'$
  29.         for all  $e \in \{un \mid n \in N\}$
  30.         channel[ $e$ ] =  $k$
  31.         if  $N \neq \emptyset$
  32.         Label[ $u$ ] = EXPLORED
  33.         for all  $w \in N_G(u)$
  34.         if Label[ $w$ ]  $\neq$  EXPLORED
  35.         Label[ $w$ ] = NEIGHBOR\_EXPLORED
  36. return  $H$
- 

Figure 3. The pseudo code for computing a special connected chordal subgraph of a supergraph  $G$  which is maximal in a subgraph  $G'$  of  $G$

In detail, Algorithm 1-1 is described in Fig. 3. For ease of description, we introduce two concepts here. (1) A graph  $G$  is chordal if and only if there exists a tree  $T$ , whose vertex set is the set of maximal cliques of  $G$ , with the following property: for every vertex  $v$  in  $G$ , the set of maximal cliques containing  $v$  induces a connected subtree of  $T$  [13]. The tree  $T$  is called a **clique tree**; nodes and edges in  $T$  are called tree nodes and tree edges, respectively. (2) In a graph  $G = (V, E)$ , for any vertex set  $S \subseteq V$ ,  $S$  is a  **$u, v$ -separator** if vertices  $u$  and  $v$  are in different connected components of  $G(V \setminus S)$ , and a **minimal  $u, v$ -separator** if no subset of  $S$  is a  $u, v$ -separator [14].

In Algorithm 1-1, the following two operations are used. (1) Compute the union  $X$  of all minimal  $u, v$ -separators in the chordal subgraph  $H'$  (Line 18 in Algorithm 1-1). (2) If  $R(H', u, v) \cup \{uv\}$  is to be added to the chordal subgraph  $H'$ , where  $R(H', u, v) = \{ux \mid x \text{ belongs to a minimal } u, v\text{-separator of } H'\}$ , update its clique tree  $T'$  to reflect this modification of  $H'$  (Line 22 in Algorithm 1-1). An implementation of each of the two operations is described in Section 5 of [15]. Each of the two operations can be done in  $O(n)$  time for each examined edge  $uv$  of  $G$ .

In the following, we illuminate the validity of Algorithm 1-1.

Lemma 1 (Corollary 3.9 of [15]): Let  $G = (V, E)$  be a chordal graph and let  $u$  and  $v$  be any pair of non-adjacent vertices in  $G$ . Then,  $M = (V, E \cup \{uv\} \cup R(G, u, v))$  is the unique minimal triangulation of  $G' = (V, E \cup \{uv\})$  obtained from  $G'$  by adding edges incident to  $u$  only, where triangulation is the process that a chordal graph is obtained from a non-chordal graph by adding edges until the graph becomes chordal.

**Theorem 2:** Given a network graph  $G$  and a subgraph  $G'$  of  $G$ , Algorithm 1-1 can obtain a special connected chordal subgraph of  $G$  which is maximal in  $G'$ .

*Proof:* (Induction) Let  $\{u_1, u_2, \dots, u_n\}$  be the sequence of vertices of  $G'$  successively added to  $U$  in an execution of Algorithm 1-1, and let  $U_i = \{u_1, u_2, \dots, u_i\}$  for any  $i$  from 1 to  $n$ . Let  $H$  be the output graph of Algorithm 1-1.  $G'(U_i)$  and  $H(U_i)$  denote the subgraphs induced by a vertex set  $U_i$  from  $G'$  and  $H$ , respectively.  $D(U_i)$  is the set of all the edges in  $H(U_i)$ .  $H_i$  is a set of all special connected chordal subgraphs of  $G$  which is maximal in  $G'(U_i)$ .

Initially, Algorithm 1-1 can obtain  $H(U_1) = (\{s\}, \emptyset) \in H_1$  and thus the base case  $i = 1$  holds. Assume that  $H(U_{i-1}) \in H_{i-1}$  ( $1 < i < n$ ), and we will show that  $H(U_i) \in H_i$  in the following.

Let  $E_i = \{u_i v \mid v \in U_{i-1} \text{ and } u_i v \in G'\}$ . A special connected chordal subgraph of  $G$  which is maximal in  $(U_i, D(U_{i-1}) \cup E_i)$  belongs to  $H_i$ . Assume that  $D' \subset D(U_{i-1}) \cup E_i$ . From Lemma 1, if  $(U_i, D')$  be a chordal graph,  $(U_i, D' \cup \{u_i v\} \cup R(G, u_i, v))$  is the unique minimal triangulation of  $(U_i, D' \cup \{u_i v\})$  obtained by adding edges in  $E_i$ . Therefore, Line 16-22 in Algorithm 1-1 can obtain the unique maximal connected chordal subgraph of  $(U_i, D(U_{i-1}) \cup L)$  which includes  $H(U_{i-1})$ , by adding minimal edges in  $L$  at every step for maintaining the chordality, where  $L \subseteq E_i$ . We define a series of subgraphs  $T_0 \supseteq S_1 \supseteq T_1 \supseteq S_2 \supseteq T_2 \supseteq S_3 \supseteq \dots \supseteq H(U_{i-1})$ . Here,  $T_0 = (U_i, D(U_{i-1}) \cup E_i)$ ;  $S_j$  is the unique maximal subgraph of  $T_{j-1}$  which satisfies the second condition in Theorem 1.  $T_j$  is the unique maximal connected chordal subgraph of  $S_j$  which includes  $H(U_{i-1})$ . Since  $E_i$  is a finite set,  $k$  exists such that  $S_k = T_{k-1}$ . Here,  $S_k$  is just  $H(U_i)$  that Algorithm 1-1 obtains in the  $i$ th loop (Line 6-35). Hence,  $H(U_i) = S_k = T_{k-1} \in H_i$ .  $\square$

**Theorem 3:** The time complexity of Algorithm 1-1 is  $O(nm^2)$ .

*Proof:* In Algorithm 1-1, some edges of  $G$  may be examined repeatedly. The edge examination is implemented at most  $\sum_{i=1}^n |E_i|^2/2$  times. Since  $\sum_{i=1}^n |E_i| = m$ ,  $\sum_{i=1}^n |E_i|^2/2 \leq m^2/2$ . The operations (1) and (2) both require only  $O(n)$  time for each edge examination. Thus, Algorithm 1-1 can be done in  $O(nm^2)$  time.  $\square$

### B. Topology Balancing for Expanding the Capacity

An undesirable feature of Algorithm 1 is that each successive channel has a maximal number of wireless links assigned to it, given the assignment to the previous channels. This causes that links are not dispensed uniformly over all channels. We wish to balance the subnetworks on all channels to expand the network capacity, thereby improving the achievable throughput.

The necessary condition of the existence of a feasible schedule is that in each maximal clique of the interference graph, the sum of air-time fractions of all links is not more than a constant  $\varepsilon$  [16], as the clique capacity constraint in (1).

$$\sum_{l \in Q_i} f_l / C_l \leq \varepsilon, \forall i \quad (1)$$

where  $0 < \varepsilon \leq 1$  is the clique capacity;  $Q_i$  is the  $i$ th maximal clique of the interference graph;  $f_l$  is the flow rate over link  $l$ ;  $C_l$  is the capacity of link  $l$ .

Selecting  $\varepsilon = 1$  is a sufficient condition for the feasibility of flow rates if and only if the interference graph is a perfect graph [17]. A graph is perfect, if for every induced subgraph, its chromatic number is equal to its clique number [1], where the clique number of a graph is the number of vertices in its maximum clique, and the chromatic number of a graph is the minimum number of colors required to color it such that any two adjacent vertices have different colors. Because a chordal graph is perfect [18] and the interference graph of a special chordal subgraph is a chordal graph, the constraint in (1) with  $\varepsilon = 1$  is a sufficient condition for the existence of a feasible schedule for given flow rates in a special chordal network subgraph. Since GMS achieves optimal throughput in a special chordal network subgraph, we have the following theorem.

**Theorem 4:** In a special chordal network subgraph  $H(U, D)$ , a GMS algorithm achieves flow rates  $\{f_l, l \in D\}$ , if and only if

$$\sum_{l \in Q_i} f_l / C_l \leq 1, \forall i \quad (2)$$

where  $Q_i$  is the  $i$ th maximal clique of the interference graph of  $H$ ;  $f_l$  is the flow rate over link  $l$ ;  $C_l$  is the capacity of link  $l$ .

Consider the case that flow rates of all the links are the same and their capacities are also the same for intuitive understanding. If all maximal cliques over all subnetworks have the same number of links, the achievable throughput of a whole network is maximum according to Theorem 4. For a network partition  $P = \{P_1, P_2, \dots, P_k\}$ , let its interference graph  $P_l$  be the union of interference graphs of all subnetworks  $P_i$ ,  $1 \leq i \leq k$ . In order to balance the subnetworks on all channels, we wish to minimize the maximum number of the vertices over all the maximal cliques in  $P_l$  and also hope to reduce the vertex number of any maximal clique  $Q_i$  so long as the reduction does not lead to an increase of the vertex number of another maximal clique  $Q_j$  whose previous vertex number is not less than the previous vertex number of  $Q_i$  minus 1.

Based on the above idea, we present an algorithm (denoted by Algorithm 2) to balance special chordal network subgraphs on all channels. For ease of explanation, we define the **interference clique number** of an edge  $e$  in a network graph  $G$

as the maximum number of the vertices over all the maximal cliques containing vertex  $e$  in the interference graph of  $G$ , denoted by  $q_G^I(e)$ . In addition, the interference degree of an edge  $e$  in network graph  $G$  is the number of edges interfering with  $e$  in  $G$ , denoted by  $d_G^I(e)$ . Algorithm 2 greedily selects an eliminable link  $l$  from a maximal clique with as many vertices as possible in the interference graphs over all channels, and seeks the best channel for link  $l$  in the channels which it can be reallocated to, such that its previous and new subnetworks after the reallocation satisfy the conditions of Theorem 1, and the interference clique numbers of link  $l$  and all the links sharing a maximal clique with it on the new channel are all less than the interference clique number of link  $l$  on the previous channel. Algorithm 2 repeats the above operation until no link can be reallocated to a better channel. Here, for a link, the best channel denotes the channel on which the link has a less interference clique number, or the same interference clique number but a less interference degree, in comparison with other channels which it can be reallocated to.

In the reallocation process of a link, the subnetworks on its previous and new channels need to maintain satisfying the conditions of Theorem 1. Hence, Algorithm 2 has to judge whether a link can be deleted from its current subnetwork and whether the link can be added to another subnetwork. It is an important problem how to judge these. In order to maintain that two conditions of Theorem 1 hold, we present two sufficient and necessary conditions for respectively judging whether to be able to delete and insert an arbitrary edge.

**Theorem 5:** In a graph  $G$ , a special chordal subgraph  $H$  of  $G$  contains an edge  $(u, v)$ .  $H - (u, v)$  is a special chordal subgraph of  $G$  if and only if  $H$  has exactly one maximal clique containing  $(u, v)$ , and the degree of  $u$  or  $v$  in  $H$  equals the number of the vertices of the maximal clique minus 1.

*Proof:* ( $\Rightarrow$ )  $H - (u, v)$  is a special chordal subgraph of  $G$ . Since  $H$  and  $H - (u, v)$  are both chordal,  $H$  has exactly one maximal clique  $Q$  containing  $(u, v)$ , according to Theorem 4 of [19]. Assume that the degrees of  $u$  and  $v$  in  $H$  are both more than the number of the vertices of  $Q$  minus 1. This means that  $H$  contain edges  $(u, v_1)$  and  $(v, v_2)$  such that  $(u, v_1) \notin Q$  and  $(v, v_2) \notin Q$ . Thus,  $(u, v_2) \notin H$ . Otherwise, another maximal clique  $Q'$  containing  $(u, v)$  exists in  $H$ . Similarly,  $(v, v_1) \notin H$ . Since  $H$  is chordal,  $(v_1, v_2) \notin H$ . Otherwise, exists the chordless cycle  $u-v_2-v_1-u$  of length 4. Therefore, the distance between  $(u, v_1)$  and  $(v, v_2)$  is 1 in both  $H$  and  $G$ , but the distance between  $(u, v_1)$  and  $(v, v_2)$  is more than 1 in  $H - (u, v)$ . This is a contradiction to the fact that  $H - (u, v)$  is a special chordal subgraph of  $G$ . Thus, the degree of  $u$  or  $v$  in  $H$  equals the number of the vertices of the maximal clique minus 1.

( $\Leftarrow$ ) Because  $H$  is a chordal graph and has exactly one maximal clique  $Q$  containing  $(u, v)$ ,  $H - (u, v)$  is a chordal graph, according to Theorem 4 of [19]. The degree of  $u$  or  $v$  in  $H$  equals the number of the vertices of  $Q$  minus 1. Without loss of generalization, assume that the degree of  $u$  in  $H$  equals the number of the vertices of  $Q$  minus 1. That is, for each vertex outside  $Q$ ,  $u$  is not adjacent to it in  $H$ . Then, for any two edges  $(u, v_1), (v, v_2) \in H - (u, v)$  ( $v_1 \neq v_2$ ), there is  $(v, v_1) \in H - (u, v)$ . Thus, for any two edges in  $H - (u, v)$ , if their distance is 1 in  $G$ , their distance is still 1 in  $H - (u, v)$ .  $\square$

Before giving the condition of adding an edge to a special chordal subgraph, we need to introduce the concept of the weight of an edge in a clique tree of a chordal graph. Let  $K_G$  be the set of maximal cliques of  $G$ . The **weighted clique intersection graph**  $W_G$  of  $G$  is the weighted graph on  $K_G$ ,

where nodes  $x, y$  are adjacent exactly when  $K_x \cap K_y \neq \emptyset$  for their corresponding cliques  $K_x, K_y$  in  $G$ , and edge  $(x, y)$  is weighted by  $w(x, y) = |K_x \cap K_y|$ . For any connected chordal graph  $G$ , a tree  $T$  is the clique tree of  $G$  if and only if  $T$  is a maximum weight spanning tree of the weighted clique intersection graph  $W_G$  of  $G$  [20]. Based on Theorem 5 and 6 of [19] as well as the second condition of Theorem 1, Theorem 6 can be proved.

**Theorem 6:** In a graph  $G$  with an edge  $(u, v)$ , there is a special chordal subgraph  $H$  of  $G$  without  $(u, v)$ . Let  $T$  be a clique tree of  $H$  and let  $x$  and  $y$  be the closest nodes in  $T$  such that  $u \in K_x, v \in K_y$ , where  $K_x$  and  $K_y$  are maximal cliques of  $H$  corresponding to tree nodes  $x$  and  $y$ , respectively.  $H + (u, v)$  is a special chordal subgraph of  $G$  if and only if the minimum weight edge  $e$  on the path between  $x$  and  $y$  in  $T$  satisfies  $w(e) = w(x, y)$ , and for any edge in  $H$ , if the distance between it and  $(u, v)$  is 1 in  $G$ , their distance is also 1 in  $H + (u, v)$ .

Theorem 6 uses a clique tree in the condition for judging whether to be able to add an edge, so we need to update the clique tree corresponding to the modified special chordal subgraph every time deleting or adding an edge. The reader can refer to [19] about maintaining a clique tree representation of a chordal graph when deleting or adding an edge.

#### IV. PERFORMANCE EVALUATION

In this section, we evaluate the distributed throughput of our special chordal subgraph-based partitioning algorithm in comparison with the existing Max K-cut algorithm [21]. We generate a wireless mesh network of 20 nodes as the evaluation scenario shown in Fig. 4. The nodes are randomly dispersed in a square area of  $300 \times 300$  m<sup>2</sup>. We assume the maximum transmission distance of 90 m and the maximum interference distance of 135 m. We adopt the parameters of link capacity from 0.3 bps/Hz to 2.7 bps/Hz like IEEE 802.11a. There are 2 channels with the channel width of 20 MHz. The traffic demand is the product of the load factor and the maximum traffic demand, where the maximum traffic demand of each link (shown in Table I) is randomly generated with a minimum of 6 Mbps and a maximum of 30 Mbps. Arrivals of packets follow the poisson distribution.

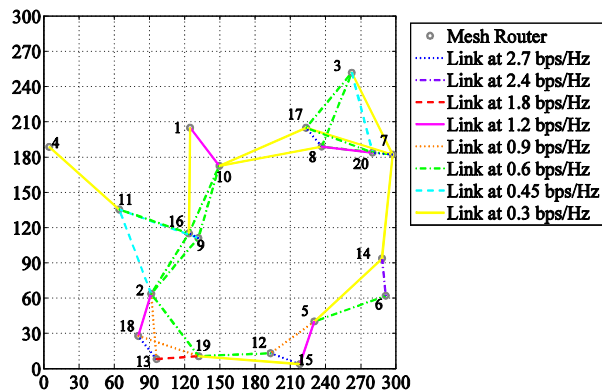


Figure 4. A wireless mesh network with a random topology

Figure 5 shows the two channel assignments according to our channel assignment algorithm and the Max K-cut algorithm, respectively. For each channel assignment, we use distributed GMS on each channel. Figure 6 plots the aggregate throughput and fairness (Jain's fairness index) of both channel assignments. From Fig. 6, we can see that the special chordal subgraph-based partitioning algorithm provides averagely 5.6% and at

most 8.3% higher aggregate throughput and averagely 4.1% and at most 9.0% higher fairness than the Max K-cut algorithm.

TABLE I. LINK LOAD

Links	Max Load (Mbps)	Links	Max Load (Mbps)	Links	Max Load (Mbps)
(1, 10)	28.341	(5, 6)	6.827	(9, 11)	20.107
(1, 16)	19.319	(5, 12)	10.269	(9, 16)	25.565
(2, 9)	10.648	(5, 14)	11.641	(10, 16)	7.1359
(2, 11)	12.426	(5, 15)	28.523	(10, 17)	14.089
(2, 13)	9.7485	(6, 14)	23.551	(11, 16)	19.669
(2, 16)	13.593	(7, 8)	17.792	(12, 15)	10.87
(2, 18)	27.496	(7, 14)	14.59	(12, 19)	14.208
(2, 19)	11.95	(7, 17)	27.401	(13, 18)	17.894
(3, 7)	20.863	(7, 20)	19.788	(13, 19)	7.3997
(3, 8)	25.715	(8, 10)	10.336	(15, 19)	16.791
(3, 17)	12.654	(8, 17)	10.996	(17, 20)	9.5894
(3, 20)	26.714	(8, 20)	20.277	(18, 19)	27.177
(4, 11)	29.238	(9, 10)	9.6757		

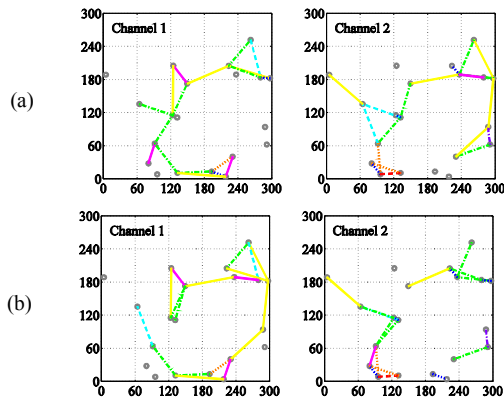


Figure 5. Channel assignments by (a) Special chordal subgraph based partitioning and (b) Max K-cut

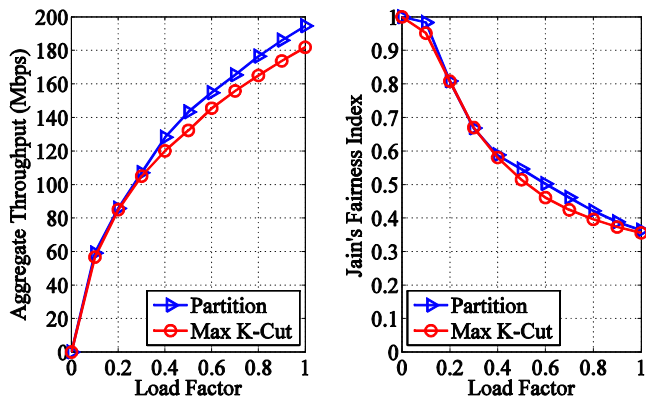


Figure 6. The performance of channel assignments

## V. CONCLUSIONS

In this paper, we explore how to partition a network into the least subnetworks to assign them to nonoverlapping channels, such that a distributed GMS algorithm is guaranteed to achieve optimal throughput in each subnetwork under secondary interference constraints. Under the  $k$ -hop interference model ( $k \geq 2$ ), the interference graph of a network subgraph depends on the structure of its network supergraph, so that the structure of a whole network in which GMS achieves optimal throughput cannot be regarded as the structure of such a subnetwork. Therefore, we provide a sufficient condition on network subgraph structures that ensure GMS to achieve optimal throughput under secondary interference constraints. Then, we define a type of network subgraphs satisfying the sufficient

condition — special chordal subgraphs. Based on the special chordal subgraphs, we propose a network partitioning algorithm, and then present a topology balancing algorithm to further expand the capacity. Evaluation results validate that our channel assignment algorithm can improve the distributed achievable throughput in a wireless mesh network.

## REFERENCES

- [1] A. Brandstadt, V. B. Le and J. P. Spinrad, "Graph Classes: A Survey", SIAM Monographs on Discrete Mathematics and Applications, 1999
- [2] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks", IEEE Transactions on Automatic Control, vol. 31, no. 12, pp. 1936-1948, Dec. 1992
- [3] B. Han, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy and A. Srinivasan, "Distributed Strategies for Channel Allocation and Scheduling in Software-Defined Radio Networks", IEEE INFOCOM, pp. 1521-1529, Apr. 2009
- [4] X. Lin and S. Rasool, "A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks", IEEE INFOCOM, pp. 1118-1126, Apr. 2007
- [5] A. Dimakis and J. Walrand, "Sufficient Conditions for Stability of Longest Queue First Scheduling: Second Order Properties Using Fluid Limits", Advances in Applied Probability, vol. 38, no. 2, pp. 505-521, Jun 2006
- [6] A. Brzezinski, G. Zussman and E. Modiano, "Distributed Throughput Maximization in Wireless Mesh Networks via Pre-Partitioning", IEEE/ACM Transactions on Networking, vol. 16, no. 6, Dec. 2008
- [7] K. Xing, X. Cheng, L. Ma and Q. Liang, "Superimposed Code Based Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks", ACM MobiCom, pp. 15-26, Sep. 2007
- [8] C. Joo, X. Lin and N. B. Shroff, "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multihop Wireless Networks", IEEE/ACM Transactions on Networking, vol. 17, no. 4, Aug. 2009
- [9] G. Zussman, A. Brzezinski and E. Modiano, "Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks", IEEE INFOCOM, pp. 1813-1821, Apr. 2008
- [10] M. Leconte, J. Ni and R. Srikant, "Improved Bounds on the Throughput Efficiency of Greedy Maximal Scheduling in Wireless Networks", ACM MobiHoc, pp. 165-174, May 2009
- [11] K. Cameron, "Induced Matchings in Intersection Graphs", Discrete Mathematics, vol. 278, no. 1-3, pp. 1-9, Mar. 2004
- [12] A. Natanzon, R. Shamir and R. Sharan, "Complexity Classification of Some Edge Modification Problems", Discrete Applied Mathematics, vol. 113, no. 1, pp. 109-128, Sep. 2001
- [13] F. Gavril, "The Intersection Graphs of Subtrees in Trees are Exactly the Chordal Graphs", Journal of Combinatorial Theory, Ser. B, vol. 16, no. 1, pp. 47-56, Feb. 1974
- [14] C. W. Ho and R. C. T. Lee, "Counting Clique Trees and Computing Perfect Elimination Schemes in Parallel", Information Processing Letters, vol. 31, no. 2, pp. 61-68, Apr. 1989
- [15] A. Berry, P. Heggernes and Y. Villanger, "A Vertex Incremental Approach for Maintaining Chordality", Discrete Mathematics, vol. 306, no. 3, pp. 318-336, Feb. 2006
- [16] Z. Fang and B. Bensaou, "Fair Bandwidth Sharing Algorithms based on Game Theory Frameworks for Wireless Ad-hoc Networks", IEEE INFOCOM, vol. 2, pp. 1284-1295, 2004
- [17] A. H. M. Rad, and V. W. S. Wong, "Joint Channel Allocation, Interface Assignment and MAC Design for Multi-Channel Wireless Mesh Networks", IEEE INFOCOM, pp. 1469-1477, May 2007
- [18] N. Alon, "Generating Pseudo-Random Permutations and Maximum Flow Algorithms", Information Processing Letters, vol. 35, no. 4, pp. 201-204, Aug. 1990
- [19] L. Ibarra, "Fully Dynamic Algorithms for Chordal Graphs and Split Graphs", ACM Transactions on Algorithms, vol. 4, no. 4, article 40, pp. 1-20, Aug. 2008
- [20] P. A. Bernstein and N. Goodman, "Power of Natural Semijoins", SIAM Journal on Computing, vol. 10, no. 4, pp. 751-771, 1981
- [21] A. P. Subramanian, H. Gupta, S. R. Das and J. Cao, "Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks", IEEE Transactions on Mobile Computing, vol. 7, no. 12, pp. 1459-1473, Dec. 2008