A Distributed and Provably-Efficient Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Multi-Radio Wireless Mesh Networks

Shahzada Rasool and Xiaojun Lin

Abstract

The capacity of wireless mesh networks can be substantially increased by equipping each mesh network node with multiple radio interfaces that can operate on multiple non-overlapping channels. However, new scheduling, channel-assignment, and routing algorithms are required to fully utilize the increased bandwidth in multi-channel multi-radio wireless mesh networks. In this paper, we develop a fully distributed algorithm that jointly solves the channel-assignment, scheduling and routing problem. Our algorithm is an online algorithm, i.e., it does not require prior information on the offered load to the network, and can adapt automatically to the changes in the network topology and offered load. We show that our algorithm is provably efficient. That is, even compared with the optimal centralized and offline algorithm, our proposed distributed algorithm can achieve a provable fraction of the maximum system capacity. Further, the achievable fraction that we can guarantee is larger than that of some other comparable algorithms in the literature.

1 Introduction

There is growing expectation that wireless mesh networks can be used to solve the decades-long *last-mile problem* of providing cost-effective broadband access, especially in areas where

DSL/Cable-Modem connections are not widely available. In this work, we are interested in developing high-capacity control protocols for *multi-channel multi-radio* wireless mesh networks. It has been shown recently that one can substantially increase the capacity of wireless mesh networks by equipping each mesh network node with multiple radio interfaces that operate on multiple non-overlapping channels. This is motivated by some current wireless LAN standards (in particular, IEEE 802.11) where the entire frequency band is divided into multiple channels, and each radio can only access one channel at a time. Hence, if each mesh network node has multiple radio interfaces, it can then utilize a larger amount of radio bandwidth, and hence achieves higher system capacity. Even if each node still has only one radio interface, by operating neighboring nodes at different channels, the amount of interference is reduced, which also leads to higher system capacity.

Multi-channel multi-radio networks pose a challenging set of resource allocation problems, including (1) *channel-assignment:* what are the set of channels that each node/link should operate on? (2) *scheduling:* when should each link be activated at each channel? and (3) *routing:* how to select paths that minimize interference and increase throughput? These three problems are inter-related with each other, and thus form a challenging cross-layer control problem across the MAC layer and the network layer.

In this work, we are interested in control protocols for multi-channel multi-radio wireless mesh networks that achieve high system capacity. Although such control protocols for channel-assignment, scheduling and routing can be obtained via the known *throughput-optimal** algorithms in [1, 2], these algorithms are centralized and often with exponential computational-complexity. Hence, they are not easy to be implemented in real systems. In this paper, we develop *distributed* and *provably-efficient* solutions to the above cross-layer control problem. By *provably-efficient*, we mean that for any offered load that a given multi-channel wireless mesh network can ever support (possibly by using the centralized and complex throughput-optimal

*Note that an algorithm in *throughput-optimal* if it can achieve the largest possible capacity region.

algorithms of [1, 2]), our algorithm can guarantee to support at least a constant fraction of this offered load on the same network. In other words, our algorithm can achieve a provable fraction of the maximum system capacity. Our algorithm is an *online* algorithm, i.e., it does not require prior knowledge of the offered load, and can automatically track the changes in the network topology and offered load. The *distributive* and *online* nature of our solution, combined with its *provable efficiency*, differentiates our work from existing control algorithms in the literature for multi-channel multi-radio wireless mesh networks that either do not guarantee provable performance bounds [3–10], or require centralized and offline solutions [1, 11, 12], or only provide order-optimal scaling laws [13].

Our solution is perhaps most comparable to the polynomial complexity (but centralized) algorithm in [11], which is also shown to guarantee a certain fraction of the maximum system capacity. Compared with the centralized solution of [11], our control algorithm is not only distributed and much simpler, but also guarantees a higher fraction of the maximum system capacity (see the comparison in Section 4.2 for details). One of the key differences between our approach and that of [11] is our assumption that, if the number of radio interfaces of a mesh network node is less than the number of channels, the mesh network node can switch radios from one channel to another dynamically. In contrast, the work in [11] requires radio-channel assignment to be fixed. Current state-of-art IEEE 802.11 hardware may take up to a few milliseconds to switch channels [14]. In addition, our solution requires some protocol overhead to collect local queue-length information, and to instruct the transmitter and receiver nodes which channel(s) they should tune to. In our theoretical analysis, we have assumed that this overhead is negligible, which could be reasonable if channel-switching occurs at the time-scale of hundreds of milliseconds. It is expected that the channel switching delay can be further reduced by more advanced physical layer communication technology, which will then make this overhead even smaller. More importantly, we believe that our results provide a strong motivation to pursue such improved channel-switching hardwares and protocols because, by allowing dynamic channel switching, one can obtain control protocols (like the one developed in this paper) that are both simpler and with higher performance. Finally, we note that the channel switching delay is not an issue if the number of radio interfaces at each node is equal to the number of channels.

Our work is related to the recent progress in developing distributed and provably-efficient scheduling algorithms for *single-channel* multi-hop wireless networks [15–20]. However, as we will show in Section 3.2, straight-forward extensions of these single-channel distributed scheduling algorithms to multi-channel networks may lead to very poor performance. The reason is that, in multi-channel networks, there may exist *channel-diversity*. That is, due to both frequency-selective multi-path fading and different amount of background interference, each link can have different rate at each channel. In Section 3.2, we will provide examples to show that straight-forward extensions of a well-studied single-channel distributed scheduling algorithm, i.e., the Maximal Scheduling algorithm, can perform arbitrarily poorly in multi-channel systems with channel diversity. In contrast, the algorithm that we develop in this paper can guarantee provable efficiency even with channel diversity. Our work is also related to the *opportunistic scheduling* algorithms in cellular networks that use OFDM (Orthogonal Frequency Division Multiplexing). Note that in OFDM, the transmitter sends information over a large number of sub-carriers (e.g., 52 sub-carriers in IEEE 802.11a). Hence, OFDM systems can also be viewed as a special type of multi-channel multi-radio wireless systems where the number of (sub-carrier) radio interfaces is equal to the number of channels/sub-carriers. Due to multipath fading, each sub-carrier can have different channel characteristics and capacity. *Opportunistic scheduling* algorithms that exploit such frequency-diversity (and also time-diversity) in wireless data networks have been extensively studied under the context of 3G cellular systems [21–23]. These algorithms *opportunistically* assign users to sub-carriers (or time-slots) with favorable channel conditions to improve overall system capacity. However, these opportunistic scheduling algorithms for cellular networks cannot be extended directly to wireless mesh networks because of the distributive nature of mesh networks. On the other hand, the algorithm that we develop in this paper can be viewed as a *distributed opportunistic scheduling* algorithm. Provided with appropriate firmware support, our algorithm can potentially be used to exploit the frequency-diversity in OFDM-based wireless mesh networks. (In contrast, current IEEE 802.11a standard does not support such subcarrier-level scheduling, and hence cannot exploit such frequency-diversity.)

The rest of the paper is organized as follows. We first outline the network model in Section 2. In Section 3, we illustrate the important effect of channel diversity, and show that straightforward extensions of a single-channel distributed scheduling algorithm can perform very poorly in multi-channel networks with channel diversity. We then present our new distributed and provably-efficient algorithm in Section 4. Then we conclude.

2 System Model

Consider a wireless network with N nodes and L links. Each link corresponds to a pair of transmitter node and receiver node. Let $b(l)$ and $e(l)$ denote the transmitter node and the receiver node, respectively, of link l. Let $E(i)$ denote the set of all links originating or terminating at node i. There are C frequency channels in the system. In order to take into account possible channel diversity, we use r_l^c to denote the rate at which link l can transfer data on channel c, provided that there are no interfering links transmitting on channel c at the same time. The interference relationship is defined as follows. For each link l, there is a set I_l of links that interfere with l. That is, if link l and another link in I_l are transmitting on the same channel at the same time, neither of the links can transfer any useful data. We assume that the interference relationship is symmetrical, i.e., $k \in I_l$ if and only if $l \in I_k$ for any two links k and l. For simplicity, we adopt the convention that $l \in I_l$. Note that the interference relationship is identical over all channels. Further, the channels are non-overlapping. Hence, it is perfectly fine that link l and another link $k \in I_l$ transmit at different channels at the same time. The above interference model is very general and can be used to model a large class of practical interference relationships, including IEEE 802.11 DCF [16, 18], Bluetooth, and FH-CDMA [24].

A key parameter that we will frequently used in this paper for characterizing the performance of control algorithms is the *interference degree* [11, 17–19]. We first define a *non-interfering subset* of I_l as a subset of links in I_l such that any two links in this subset do not interfere with each other. The interference degree $\mathcal{K}(l)$ of link l is the maximum cardinality of any non-interfering subset of I_l. In other words, the interference degree $\mathcal{K}(l)$ of link l characterizes the potential loss of system capacity if link l is scheduled, i.e., it is the maximum number of links that could have been turned on simultaneously (without interference) if link l is not turned on. The interference degree $\mathcal{K}$ of the whole network is the maximum possible interference degree over all links. Note that the interference degree of the system can often be determined directly from the physical interference model, and thus is independent of the exact network topology. For example, under the so-called node-exclusive interference model for Bluetooch and FH-CDMA networks, the interference degree is 2. For the so-called bi-directional equal-power model that approximates IEEE 802.11 DCF, the interference degree is 8 [18].

Let M_i be the number of radio interfaces available at node i. We assume that at any given time a radio can only tune to one channel. Therefore, for link l to successfully communicate on channel c, both the transmitting node $b(l)$ and the receiving node $e(l)$ must tune one radio to channel c. Like [4, 14], we assume that radios can switch channels dynamically.

There are S users in the system. Each user is associated with a source node and a destination node. The traffic from each user may be routed over multiple alternate paths. Let $J(s)$ denote the number of alternate paths for user s. Let $[H^l_{sj}]$ denote the routing matrix, where $H^l_{sj} = 1$ if path j of user s traverses link l, $H^l_{sj} = 0$, otherwise. For simplicity, we assume that user s injects packets into the system at a constant rate λ_s (while our results can also be easily generalized to the case when packets arrive according to some stationary and ergodic processes).

We assume that time is divided into slots of unit length, and the time that it takes to switch radios between channels is negligible compared to the length of each time slot. For ease of exposition, in the rest of the paper we will use the term "schedule" to refer to both the channel assignment and the link schedule at a time slot whenever there is no source of confusion. At time slot t, let $\mathcal{M}(t) = [\mathcal{M}^c(t)]$ denote the outcome of the scheduling policy, where $\mathcal{M}^c(t)$ is the set of non-interfering links that are chosen to transmit information at channel c at time t. Let $D_l(t)$ denote the number of packets that link l can serve at time slot t. Then $D_l(t) = \sum_{c:l \in \mathcal{M}^c(t)} r_l^c$. Let $P_{sj}(t)$ denote the fraction of traffic from user s that is routed to path j at time slot t. Further, let $q_l(t)$ denote the number of packets queued at link l at the beginning of time slot t. The evolution of $q_l(t)$ may be written as:

$$q_l(t+1) = [q_l(t) + \sum_{s=1}^{S} \sum_{j=1}^{J(s)} H_{sj}^l P_{sj}(t) \lambda_s - D_l(t)]^+, \tag{1}$$

where $[\cdot]^+$ denote the projection to $[0, +\infty)$. We say that the system is *stable* if the queue lengths at all links remain finite [2], i.e.

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}_{\{\sum_{l=1}^{L} q_l(t) > \eta\}} \to 0, \text{ almost surely as } \eta \to \infty. \tag{2}$$

Remark: Note that in (1) we have adopted the simplifying assumption that packets from each user s are applied to all links l along the path of user s simultaneously. This assumption simplifies the analysis of the paper, and allows us to focus on the channel assignment and scheduling components of the problem. In reality, packets have to traverse the links one at a time. There are a number of known methodologies that can extend our model to take into account this link-by-link packet dynamics [15–17,25]. One such approach is to assume that users can communicate the amount of transmitted traffic $P_{sj}(t)\lambda_s$ to all links along path j through an additional control channel, and let each link l updates a "virtual" queue according to (1). Then, with an appropriate packet scheduling policy at each link, one can show that the real queue (with link-by-link packet dynamics) is stable as long as the "virtual" queue defined by (1) is stable [15, 26]. Hence, in this paper we will neglect the link-by-link packet dynamics, and use (1) to describe the system dynamics.

Let $\vec{\lambda} = [\lambda_1, ...\lambda_S]$ denote the offered load to the network. The *capacity region* under a particular channel-assignment, scheduling and routing algorithm is the set of $\vec{\lambda}$ such that the system remains stable. Under possible routing constraints, we define the *optimal capacity region* Ω as the supremum of the capacity regions of all algorithms. An algorithm is *throughput-optimal* if it can achieve the optimal capacity region Ω. The *efficiency ratio* of a (possibly sub-optimal) algorithm is the largest number γ such that this policy can stablize the system under any load $\vec{\lambda} \in \gamma\Omega$. By definition, a throughput-optimal algorithm has an efficiency ratio of 1.

3 Generalizations of Single-Channel Scheduling Algorithms to Multi-Channel Networks

In this paper, we are interested in distributed control algorithms with provable efficiency ratios. A number of provably-efficient and low-complexity scheduling algorithms have been proposed for *single-channel* multi-hop wireless networks [15–20]. One would naturally hope that the generalization of these single-channel scheduling algorithms may lead to equally efficient and low-complexity scheduling algorithms for multi-channel networks. In this section, we will study the generalization of two such single-channel scheduling algorithms, i.e., Greedy Maximal Scheduling and Maximal Scheduling, to multi-channel networks. We will first show that the generalization of Greedy Maximal Scheduling, which is a low-complexity but centralized algorithm, can still guarantee efficiency-ratios almost as tight as in single-channel networks. While this result is encouraging, the Greedy Maximal Scheduling algorithm is not very easy to be implemented in a distributive fashion. It would be more desirable that we can develop distributed scheduling algorithms for multi-channel wireless mesh networks. However, we will show that straight-forward extensions of Maximal Scheduling, a low-complexity and distributed algorithm, can result in much lower efficiency ratios in multi-channel networks.

3.1 Greedy Maximal Scheduling

It is known that the maximum system capacity can be achieved by the following throughput-optimal scheduling algorithm [1,2,15]: at each time slot t, the schedule $\mathcal{M}(t)$ should be chosen to maximize the queue-weighted rate-sum $\sum\limits_{l,c:l\in\mathcal{M}^c} q_l(t)r_l^c$. The Greedy Maximal Scheduling algorithm can be viewed as an approximation to such throughput-optimal algorithms. Note that a schedule $\mathcal{M}$ is *maximal* if $\mathcal{M}$ is a non-interfering schedule, and no more links can be added to $\mathcal{M}^c(t)$ at any channel c without violating the interference constraint and radio interface constraint. The Greedy Maximal Scheduling algorithm computes a maximal schedule by always starting from the link with the largest queue-weighted rate $q_l(t)r_l^c$. Specifically, the Greedy Maximal Scheduling algorithm proceeds as follows. *(i)* Form a set $\mathcal{F}$ of all link-channel pairs (l,c). Define the weight of each link-channel pair (l,c) to be $q_l r_l^c$, where q_l is the current queue length at link l. Start from an empty schedule $\mathcal{M}(t)$. *(ii)* First search for the link-channel pair (l,c) with the largest weight $q_l r_l^c$. Add link l to $\mathcal{M}^c(t)$. Remove from $\mathcal{F}$ all link-channel pairs that cannot be scheduled due to (l,c) being scheduled. Specifically, remove from $\mathcal{F}$ all link-channel pairs (k,c) with $k\in I_l$. (Note that (l,c) is also removed since we define $l\in I_l$). Further, if by scheduling link l on channel c, the transmitting node $b(l)$ (or respectively, the receiving node $e(l)$) of link l already uses up all $M_{b(l)}$ (respectively, $M_{e(l)}$) radio interfaces, remove from $\mathcal{F}$ all link-channel pairs (k,c') with $k\in E(b(l))$ (or respectively, $k\in E(e(l))$). (In other words, all links incident to node $b(l)$ or $e(l)$ are removed from $\mathcal{F}$ since no radio interfaces are available.) *(iii)* Next, find the link-channel pair (l,c) with the largest weight $q_l r_l^c$ from the *remaining* pairs in $\mathcal{F}$, and similar to the previous step, remove from $\mathcal{F}$ all link-channel pairs that cannot be scheduled due to the new (l,c). Continue in this fashion until no link-channel pairs are left in $\mathcal{F}$. Note that the above Greedy Maximal Scheduling algorithm is a natural generalization of the Greedy Maximal Scheduling algorithm for single-channel networks [15, 27].

The following proposition shows that the above Greedy Maximal Scheduling algorithm can guarantee an efficiency ratio of $1/(\mathcal{K}+2)$ in multi-channel networks, where $\mathcal{K}$ is the interference degree defined in Section 2. Note that the Greedy Maximal Scheduling algorithm has been shown to achieve an efficiency ratio of $1/\mathcal{K}$ in single-channel networks [15]. Thus, we can conclude that its performance in multi-channel networks is similar. For simplicity, we only present the following result for the case where each user has one fixed path through the network. Let $H_s^l = 1$ if the path of user s uses link l, $H_s^l = 0$, otherwise.

Proposition 1 *Assume that each user only has one path through the network, and the routing matrix is $[H_s^l]$. The above Greedy Maximal Scheduling algorithm can achieve an efficiency ratio of $1/(\mathcal{K}+2)$.*

Proof: Let $\mathcal{M}_g$ denote the scheduling decision by the Greedy Maximal Scheduling algorithm, and let $\mathcal{M}^*$ denote the scheduling decision by the throughput-optimal policy that maximizes the queue-weighted rate-sum $\sum_{(l,c):l\in\mathcal{M}^c} q_l(t)r_l^c$. We first show that the queue-weighted rate-sum computed by Greedy Maximal Scheduling is at least $1/(\mathcal{K}+2)$ times the maximum queue-weighted rate-sum, i.e.,

$$\sum_{(l,c):l\in\mathcal{M}_g^c} q_l r_l^c \geq \frac{1}{\mathcal{K}+2} \sum_{(l,c):l\in\mathcal{M}^{*,c}} q_l r_l^c. \tag{3}$$

To see this, consider the first link-channel pair (l,c) picked by the Greedy Maximal Scheduling algorithm. Because (l,c) is scheduled, all link-channel pairs (k,c) with $k \in I_l$ cannot be scheduled any more. Among these interfering link-channel pairs, the throughput-optimal algorithm may at most pick $\mathcal{K}$ link-channel pairs into $\mathcal{M}^{*,c}$. Further, because (l,c) is scheduled, node $b(l)$ (or respectively, node $e(l)$) may uses up all available radio interfaces. If this is indeed the case, all links incident to $b(l)$ (or respectively, $e(l)$) cannot be scheduled any more. In this case, the throughput-optimal algorithm may at most pick 2 additional link-channel pairs from all links incident to $b(l)$ (and $e(l)$). Note that all these interfering link-channel pairs have smaller weights than (l,c). Hence, the sum of weights contributed by those link-channel pairs in $\mathcal{M}^*$ that are among the interferers described above is at most $(\mathcal{K}+2)$ times the weight of (l,c). Now remove all these interfering link-channel pairs from $\mathcal{F}$, $\mathcal{M}_g$ and $\mathcal{M}^*$. By applying the above argument inductively to each link-channel pair picked by Greedy Maximal Scheduling, we can thus prove (3).

Then, by applying Proposition 3 of [15] to the inequality (3), we can conclude that the efficiency ratio of Greedy Maximal Scheduling is at least $1/(\mathcal{K}+2)$. *Q.E.D.*

If $M_i = C$ for all node i, i.e., if the number of radio interfaces on each mesh network node is the same as the total number of channels in the system, the scheduling algorithm then does not need to consider radio interface constraints. Using a similar argument, we can show that the queue-weighted rate-sum computed by Greedy Maximal Scheduling is at most $1/\mathcal{K}$ times the maximum queue-weighted rate-sum. We then obtain the following tighter result.

Proposition 2 *Assume that each user only has one path through the network, and the routing matrix is* $[H_s^l]$. *Further, assume that* $M_i = C$ *for all* i. *The above Greedy Maximal Scheduling algorithm can achieve an efficiency ratio of* $1/\mathcal{K}$.

3.2 Maximal Scheduling

The Greedy Maximal Scheduling algorithm is essentially a centralized algorithm, and is not very easy to be implemented in a distributive fashion. We next turn to extensions of the Maximal Scheduling algorithm to multi-channel wireless mesh networks. Recall the definition of a maximal schedule in Section 3.1. Mathematically, a maximal schedule can be stated as follows.

Multi-Channel Maximal Scheduling: For any link-channel pair (l, c) such that link l is backloged (i.e., $q_l(t) \geq \sum_{c=1}^{C} r_l^c$), at least one of the following is true:

- Either link l is scheduled in channel c, i.e., $l \in \mathcal{M}^c(t)$, or,
- One of the interfering links to link l is scheduled in channel c, i.e. $k \in \mathcal{M}^c(t)$ for some $k \in I_l$, or

- Either the transmitter or the receiver of link l has used up all the radios, i.e.

$$\sum_{k\in E(b(l))}\sum_{d=1}^{C}\mathbf{1}_{\{k\in\mathcal{M}^d(t)\}} = M_{b(l)}, \text{ or}$$
$$\sum_{k\in E(e(l))}\sum_{d=1}^{C}\mathbf{1}_{\{k\in\mathcal{M}^d(t)\}} = M_{e(l)}.$$

Maximal schedules can be easily computed via a distributed algorithm [28]. For single-channel networks, such Maximal Scheduling algorithms have been shown to achieve an efficiency ratio of $1/\mathcal{K}$. However, for multi-channel networks with channel diversity, such algorithms can perform much worse, because the algorithm could pick the weakest (i.e., least capacity) links at each channel into the maximal schedule. A slightly improved version of Maximal Scheduling for multi-channel networks would be to enforce an additional constraint that the scheduling pattern on all channels must be the same. Assume that $M_i = C$ for all node i, i.e., each node has enough radio interfaces to access all channels simultaneously. Thus, if we aggregate all channels together, the capacity of link l is $\sum_{c=1}^{C} r_l^c$. We can then define the following Aggregated Maximal Scheduling as in single-channel networks.

Aggregated Maximal Scheduling:

- If link l is scheduled, it will transmit over all channels simultaneously.
- For any link l that is backloged (i.e., $q_l \geq \sum_{c=1}^{C} r_l^c$), either link l is scheduled, or some other link $k \in I_l$ is scheduled.

Note that this is in fact the way current IEEE 802.11a standard uses OFDM (readers can refer to the discussions in the Introduction regarding the relationship between OFDM and multi-channel networks). However, the performance of Aggregated Maximal Scheduling can still be very poor. In fact, the following example shows that its efficiency ratio can be as low as $1/\bar{I}$, where $\bar{I}$ is the maximum number of links that interfere with any link l. To see this, consider a node 0 communicating with nodes $1, ..., \bar{I}$. Label the link between node 0 and node i as link i. Assume that each link interferes with all other links if they operate on the same channel. Hence, only one link can be assigned to each channel. Assume that the total number of channels is also $\bar{I}$. Let the capacity of link i at channel i to be 1, while its capacity at all other channels to be ϵ. Thus, if we use Aggregated Maximal Scheduling, the aggregate capacity of link i is $1 + \epsilon(\bar{I} - 1)$. Since only one link can be activated at each time, the total capacity of the system is $1 + \epsilon(\bar{I} - 1)$. However, if we assign link i to operate on channel i, there will be no interference in the system, and the total capacity of the system is thus $\bar{I}$. Clearly, as ϵ approaches zero, the efficiency ratio of Aggregate Maximal Scheduling can be arbitrarily close to $1/\bar{I}$. In contrast, Greedy Maximal Scheduling can guarantee the efficiency ratio $1/(\mathcal{K}+2)$, where $\mathcal{K}$ is independent of network topology.

The above example clearly illustrates the weakness of Maximal Scheduling algorithms in multi-channel networks with channel diversity. In particular, the above extensions of Maximal Scheduling to multi-channel networks fail to take into account the channel diversity, and hence the performance of the scheduling decisions can be very poor.

4 A Distributed and Provably-Efficient Multi-Channel Multi-Radio Control Algorithm

Given the results in Section 3, a natural question is then: can we develop a distributed scheduling algorithm for multi-channel multi-radio wireless mesh networks that can guarantee the same efficiency ratio as the centralized Greedy Maximal Scheduling algorithm? In this section, we will develop such a distributed scheduling algorithm. Interestingly, our new scheduling algorithm still uses maximal schedules. Obviously, in order to avoid the inefficiency illustrated in Section 3.2, we must be able to properly take into account channel diversity. In this work, we introduce a novel *two-stage queueing* mechanism to address channel diversity, and to prevent links from using channels that are weak (i.e., with smaller capacity).

The basic idea of two-stage queueing is as follows. Packets arriving to each link l are served in two steps. The first step is a *logical* assignment of the packets to channels: packets that arrive at each link l are assigned to queues that correspond to each channel c. The second step is the *actual* scheduling of radio interfaces and links: radios are assigned to channels according to maximal schedules, and packets in the channel queues are served. The key of two-stage queueing is to ensure that, in the first step, packets are less likely to be assigned to link-channel pairs that are "weak." Thus, the "weak" links do not even participate in the maximal schedules in the second step. Clearly, the main difficulty is how to determine in a distributive and online fashion which link-channel pairs are "weak." As we will show next, our algorithm makes this decision intelligently by using the queue length information at both the per-link level and the per-channel level.

4.1 The Single-Path Case

For ease of exposition, in this subsection we first focus on the case where each user only has one fixed path through the network. Let $H_s^l = 1$ if the path of user s uses link l, $H_s^l = 0$, otherwise. Our proposed multi-channel multi-radio scheduling algorithm works as follows. Each link maintains $C+1$ queues. There is one queue q_l for each link l, which represents the backlog of packets at link l that have not been assigned to channel queues yet. At the same time, each link l maintains C channel-queues $\eta_l^c, c = 1, ..., C$. The per-channel queue η_l^c represents the backlog of packets assigned to channel c by link l that are still waiting to be served. At each time slot t, the following algorithm is executed:

Algorithm SP**:**

- Step 1: Define $x_l^c(t)$ to be the number of packets that link l can assign to channel c at time-slot t. For each link l, let

$$\begin{aligned}
x_l^c(t) = r_l^c, \text{ if } \tfrac{q_l}{\alpha_l} \geq \tfrac{1}{r_l^c} \Bigg[& \sum_{k \in I_l} \tfrac{\eta_k^c}{r_k^c} \\
& + \tfrac{1}{M_{b(l)}} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \tfrac{\eta_k^d}{r_k^d} + \tfrac{1}{M_{e(l)}} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \tfrac{\eta_k^d}{r_k^d} \Bigg] \\
x_l^c(t) = 0, \text{ otherwise}, &
\end{aligned} \tag{4}$$

where α_l is an arbitrary positive constant chosen for link l. Then, link l drains $\min\{q_l(t), \sum_{c=1}^{C} x_l^c(t)\}$ packets from q_l, and assign them to each channel queue η_l^c. Let $y_l^c(t)$ be the actual number of packets assigned to each channel queue η_l^c. Recall that in the same time slot, link l also receives $\sum\limits_{s=1}^{S} H_s^l \lambda_s$ new packets. The evolution of q_l is thus given by:

$$q_l(t+1) = q_l(t) + \sum_{s=1}^{S} H_s^l \lambda_s - \sum_{c=1}^{C} y_l^c(t), \tag{5}$$

where

$$\begin{aligned}
& \sum_{c=1}^{C} y_l^c(t) = \min\{q_l(t), \sum_{c=1}^{C} x_l^c(t)\} \\
& 0 \leq y_l^c(t) \leq x_l^c(t).
\end{aligned} \tag{6}$$

- Step 2: Based on the channel queues $\eta_l^c(t)$, Multi-Channel Maximal Scheduling (as in Section 3.2) is carried out to determine the channel-assignment and link schedules. Mathematically, this means that for any link-channel pair (l, c) that is backloged (i.e., $\eta_l^c(t) \geq r_l^c$), one of the statements under the Multi-Channel Maximal Scheduling algorithm in Section 3.2 must hold. Then at the end of step 2, each channel queue $\eta_l^c(t)$ is updated by

$$\eta_l^c(t+1) = \eta_l^c(t) + y_l^c(t) - r_l^c \mathbf{1}_{\{l \in \mathcal{M}^c(t)\}}.$$

Remark: The assignment in (4) is the key to ensure that links will only be scheduled on their "strong" channels. Note that in (4) each link only needs to know the channel-queue-length η_l^c at interfering links. This equation can be explained intuitively by interpreting the quantities q_l and η_l^c as "pricing" signals. The quantity q_l can be interpreted as the *congestion cost* at link l (due to the imbalance between the external arrivals and the system capacity). The quantity $\sum_{k\in I_l} \frac{\eta_k^c}{r_k^c}$ can be interpreted as the *contention cost* at link l (due to the interference on channel c from links in the interference set I_l). The quantities $\frac{1}{M_{b(l)}} \sum_{k\in E(b(l))} \sum_{d=1}^{C} \frac{\eta_k^d}{r_k^d}$ and $\frac{1}{M_{e(l)}} \sum_{k\in E(e(l))} \sum_{d=1}^{C} \frac{\eta_k^d}{r_k^d}$ can be interpreted as the *radio costs* at the transmitter node $b(l)$ and receiver node $e(l)$, respectively, of link l. Hence, each link l will assign traffic to channel c at the maximum rate r_l^c only if the contention cost of the channel plus the radio cost, weighted by the channel capacity r_l^c, is smaller than the congestion level at the link. Note that if r_l^c is small, the right hand side of (4) will increase, and hence it is less likely that link l will assign traffic to channel c. On the other hand, if the congestion level q_l is too high, it is still possible for link l to assign traffic to a channel c with low rate, provided that the contention cost and the radio cost is low.

The following main result shows that the strategy in Equation (4) strikes the right balance in identifying "weak" channels from "strong" channels. The result indicates that the efficiency ratio of Algorithm SP is identical to that of the centralized Greedy Maximal Scheduling algorithm of Section 3.1.

Proposition 3 *Assume that each user only has one path through the network, and the routing matrix is $[H_s^l]$. The efficiency ratio of our proposed algorithm SP is $1/(\mathcal{K}+2)$ where $\mathcal{K}$ is the interference degree defined in Section 2.*

Proof: We will show that for any $\vec{\lambda}$ such that some scheduling algorithm can stabilize the network at the offered load $(\mathcal{K}+2)\vec{\lambda}$, Algorithm SP will stablize the system at the offered load $\vec{\lambda}$. We use the following Lyapunov function to establish stability:

$$V(\vec{q}, \vec{\eta}) = V_q(\vec{q}) + V_\eta(\vec{\eta}), \tag{7}$$

where

$$
\begin{aligned}
V_q(\vec{q}) &= \sum_{l=1}^{L} \frac{(q_l)^2}{2\alpha_l}, \\
V_\eta(\vec{\eta}) &= \sum_{l=1}^{L} \sum_{c=1}^{C} \frac{\eta_l^c}{2r_l^c} \Bigg[\sum_{k \in I_l} \frac{\eta_k^c}{r_l^c} + \frac{1}{M_{b(l)}} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{\eta_k^d}{r_k^d} \\
&\quad + \frac{1}{M_{e(l)}} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{\eta_k^d}{r_k^d} \Bigg].
\end{aligned}
$$

Let $\Delta V(t) = V(\vec{q}(t+1), \vec{\eta}(t+1)) - V(\vec{q}(t), \vec{\eta}(t))$. First, note that

$$
\begin{aligned}
& \frac{(q_l(t+1))^2}{2} - \frac{(q_l(t))^2}{2} \\
= \;& q_l(t)(q_l(t+1) - q_l(t)) + \frac{(q_l(t+1) - q_l(t))^2}{2} \\
\leq \;& q_l(t) \Big[\sum_{s=1}^{S} H_s^l \lambda_s - \sum_{c=1}^{C} y_l^c(t) \Big] + C_1
\end{aligned}
$$

where $C_1 \triangleq \max_l (\sum_{s=1}^{S} \sum_{j=1}^{J(s)} H_s^l \lambda_s + \sum_{c=1}^{C} r_l^c)^2$ upper bounds $(q_l(t+1) - q_l(t))^2$ for any l. According to (6), $\sum_{c=1}^{C} y_l^c(t) = \sum_{c=1}^{C} x_l^c(t)$ whenever $q_l(t) \geq \sum_{c=1}^{C} r_l^c$. Hence, we can show that,

$$
\begin{aligned}
& V_q(\vec{q}(t+1)) - V_q(\vec{q}(t)) \\
\leq \;& \sum_{l=1}^{L} \frac{q_l(t)}{\alpha_l} \Big[\sum_{s=1}^{S} H_s^l \lambda_s - \sum_{c=1}^{C} x_l^c(t) \Big] + C_2
\end{aligned}
$$

for some constant C_2. Similarly, we can show that

$$
\begin{aligned}
& V_\eta(\vec{\eta}(t+1)) - V_\eta(\vec{\eta}(t)) \\
\leq \;& \sum_{l=1}^{L} \sum_{c=1}^{C} \frac{\eta_l^c(t)}{r_l^c} \Bigg[\sum_{k \in I_l} \frac{y_k^c(t)}{r_k^c} + \frac{1}{M_{b(l)}} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{y_k^d(t)}{r_k^d} \\
& + \frac{1}{M_{e(l)}} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{y_k^d(t)}{r_k^d} - \mu_l^c(t) \Bigg] + C_3 \\
\leq \;& \sum_{l=1}^{L} \sum_{c=1}^{C} \frac{\eta_l^c(t)}{r_l^c} \Bigg[\sum_{k \in I_l} \frac{x_k^c(t)}{r_k^c} + \frac{1}{M_{b(l)}} \sum_{k \in E(b(l))} \sum_{d=1}^{C} \frac{x_k^d(t)}{r_k^d} \\
& + \frac{1}{M_{e(l)}} \sum_{k \in E(e(l))} \sum_{d=1}^{C} \frac{x_k^d(t)}{r_k^d} - \mu_l^c(t) \Bigg] + C_3,
\end{aligned}
$$

where C_3 is a positive constant and

$$\begin{aligned}\mu_l^c(t) \;\; &= \;\; \sum_{k\in I_l} \mathbf{1}_{\{k\in\mathcal{M}^c(t)\}} + \frac{1}{M_{b(l)}} \sum_{k\in E(b(l))} \sum_{d=1}^{C} \mathbf{1}_{\{k\in\mathcal{M}^d(t)\}} \\ & \quad + \frac{1}{M_{e(l)}} \sum_{k\in E(e(l))} \sum_{d=1}^{C} \mathbf{1}_{\{k\in\mathcal{M}^d(t)\}}. \end{aligned} \tag{8}$$

Therefore, the Lyapunov drift $\Delta V(t)$ can be bounded by

$$\begin{aligned} & \mathbf{E}\big[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)\big] \\ \leq \; & \sum_{l=1}^{L} \frac{q_l(t)}{\alpha_l} \Big[\sum_{s=1}^{S} H_s^l \lambda_s - \sum_{c=1}^{C} x_l^c(t)\Big] \\ & + \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{\eta_l^c(t)}{r_l^c} \Bigg[\sum_{k\in I_l} \frac{x_k^c(t)}{r_k^c} + \frac{1}{M_{b(l)}} \sum_{k\in E(b(l))} \sum_{d=1}^{C} \frac{x_k^d(t)}{r_k^d} \\ & + \frac{1}{M_{e(l)}} \sum_{k\in E(e(l))} \sum_{d=1}^{C} \frac{x_k^d(t)}{r_k^d} - \mu_l^c(t)\Bigg] + C_4, \end{aligned}$$

for some positive constant C_4.

Since there exist some scheduling algorithm that can stabilize the system at the offered load vector $(\mathcal{K}+2)\vec{\lambda}$, there must exist some $\tilde{x}_l^c \in [0, r_l^c]$ for each (l,c) such that

$$(1+\epsilon)^2(\mathcal{K}+2)\sum_{s=1}^{S} H_s^l \lambda_s \leq \sum_{c=1}^{C} \tilde{x}_l^c, \text{ for all link } l, \tag{9}$$

$$\sum_{k\in I_l} \frac{\bar{x}_k^c}{r_k^c} \leq \mathcal{K}, \text{ for all link } l \text{ and channel } c, \tag{10}$$

$$\sum_{k\in E(i)} \sum_{c=1}^{C} \frac{\bar{x}_k^c}{r_k^c} \leq M_i, \text{ for all node } i, \tag{11}$$

where $\tilde{x}_l^c$ can be interpreted as the long-term average amount of service that link l received at channel c, and ϵ is a small positive number. Note that the inequality (9) is due to the rate-balance at link l. The inequality (10) is due to the interference constraint, i.e., there can be no more than $\mathcal{K}$ links activated simultaneously in any interference range I_l. The inequality (11) is due to the radio interface constraint, i.e., there can be no more than M_i link-channel pairs incident to node i that are activated simultaneously (see [11,12]). Let $\bar{x}_l^c = \frac{\tilde{x}_l^c}{(\mathcal{K}+2)(1+\epsilon)}$. We thus have,

$$(1+\epsilon)\sum_{s=1}^{S} H_s^l \lambda_s \le \sum_{c=1}^{C} \bar{x}_l^c, \text{ for all link } l \tag{12}$$

$$(1+\epsilon)\left[\sum_{k\in I_l}\frac{\bar{x}_k^c}{r_k^c} + \frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d} + \frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}\right] \le 1, \text{ for all } (l,c). \tag{13}$$

Therefore, we have

$$\begin{aligned}
& \mathbf{E}[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)] \\
\le\ & \sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}[\sum_{s=1}^{S} H_s^l\lambda_s - \sum_{c=1}^{C}\bar{x}_l^c] \\
& + \sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}[\sum_{c=1}^{C}\bar{x}_l^c - \sum_{c=1}^{C} x_l^c(t)] \\
& + \sum_{l=1}^{L}\sum_{c=1}^{C}\frac{\eta_l^c(t)}{r_l^c}\Bigg[\sum_{k\in I_l}\frac{\bar{x}_k^c}{r_k^c} + \frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d} \\
& \qquad + \frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d} - \mu_l^c(t)\Bigg] \\
& + \sum_{l=1}^{L}\sum_{c=1}^{C}\frac{\eta_l^c(t)}{r_l^c}\Bigg[\sum_{k\in I_l}\frac{x_k^c(t)-\bar{x}_k^c}{r_k^c} \\
& \qquad + \frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{x_k^d(t)-\bar{x}_k^d}{r_k^d} \\
& \qquad + \frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{x_k^d(t)-\bar{x}_k^d}{r_k^d}\Bigg] + C_4.
\end{aligned}$$

Let

$$m_l^c = [\sum_{k\in I_l}\frac{\bar{x}_k^c}{r_k^c} + \frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d} + \frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}].$$

By definition of Multi-Channel Maximal Scheduling (see Section 3.2), $\mu_l^c(t) \ge 1$ whenever $\eta_l^c(t) \ge$

r_l^c. Using (12)–(13), we thus have

$$
\begin{aligned}
&\mathbf{E}[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)] \\
\leq\ & -\epsilon \sum_{l=1}^{L} \frac{\sum_{s=1}^{S} H_s^l \lambda_s}{\alpha_l} q_l(t) - \epsilon \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{m_l^c(t)}{r_l^c} \eta_l^c(t) \\
& + \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{\bar{x}_l^c - x_l^c(t)}{r_l^c} \left[\frac{r_l^c q_l(t)}{\alpha_l} - \left(\sum_{k\in I_l} \frac{\eta_k^c(t)}{r_k^c} \right.\right. \\
& \left.\left. + \frac{1}{M_{b(l)}} \sum_{k\in E(b(l))} \sum_{d=1}^{C} \frac{\eta_k^d(t)}{r_k^d} + \frac{1}{M_{e(l)}} \sum_{k\in E(e(l))} \sum_{d=1}^{C} \frac{\eta_k^d(t)}{r_k^d} \right) \right] \\
& + C_5.
\end{aligned}
$$

Recall that $0 \leq \bar{x}_l^c \leq r_l^c$, then by Step 1 of Algorithm SP,

$$
\begin{aligned}
& \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{\bar{x}_l^c - x_l^c(t)}{r_l^c} \left[\frac{r_l^c q_l(t)}{\alpha_l} - \left(\sum_{k\in I_l} \frac{\eta_k^c(t)}{r_k^c} \right.\right. \\
& \quad \left.\left. + \frac{1}{M_{b(l)}} \sum_{k\in E(b(l))} \sum_{d=1}^{C} \frac{\eta_k^d(t)}{r_k^d} + \frac{1}{M_{e(l)}} \sum_{k\in E(e(l))} \sum_{d=1}^{C} \frac{\eta_k^d(t)}{r_k^d} \right) \right] \leq 0.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\mathbf{E}[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)] \\
\leq\ & -\epsilon \sum_{l=1}^{L} \frac{\sum_{s=1}^{S} H_s^l \lambda_s}{\alpha_l} q_l(t) - \epsilon \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{m_l^c}{r_l^c} \eta_l^c(t) + C_4.
\end{aligned}
$$

The stability of the system then follows [2]. *Q.E.D.*

If $M_i = C$ for all node i, i.e., when there are no radio interface constraints, a tighter efficiency ratio can be shown by slightly modifying Algorithm SP. In particular, we can remove the radio costs (i.e., the last two terms) in (4). We can then use the following Lyapunov function:

$$
V(\vec{q},\vec{\eta}) = \sum_{l=1}^{L} \frac{(q_l)^2}{2\alpha_l} + \sum_{l=1}^{L}\sum_{c=1}^{C} \frac{\eta_l^c}{2r_l^c} [\sum_{k\in I_l} \frac{\eta_k^c}{r_k^c}].
$$

Following the line of proof of Proposition 3, and noting that the last two terms of (13) can also be removed, We can then show the following tighter result.

Proposition 4 *Assume that each user only has one path through the network, and the routing matrix is* $[H_s^l]$*. Further, assume that* $M_i = C$ *for all node* i*. The efficiency ratio of the modified Algorithm* SP *is* $1/\mathcal{K}$*.*

In both cases, the distributed Algorithm SP achieves the same efficiency ratio as the centralized Greedy Maximal Scheduling algorithm in Section 3.1.

4.2 The Multi-Path Case

We next extend Algorithm SP to the case when each user can use multiple alternate paths. We start from the following model. Assume that each user is provided with $J(s)$ alternate paths through the network. For the moment, we assume that these paths are given, and we will study how each user should optimally route packets among these alternate paths. Then, in Section 4.3, we will discuss how these paths should be computed. Define P_{sj} to be the fraction of incoming packets from user s that are routed to path j. Let $\vec{P}_s = [P_{s1}, \ldots, P_{s,J(s)}]$. Obviously, $P_{sj} \geq 0$ and $\sum_{j=1}^{J(s)} P_{sj} = 1$ for all user s. Let $\vec{P} = [\vec{P}_1, \ldots, \vec{P}_s]$. We can then generalize Algorithm SP to the following joint channel-assignment, scheduling and routing algorithm.

Algorithm MP**:** At each time slot t:

- Step 1: Each user s computes the routing fractions $P_{sj}(t)$ as the solution to the following local optimization problem:

$$\begin{aligned} \max_{\vec{P}_s} \quad & -\frac{\beta_s}{2}\sum_{j=1}^{J(s)}(P_{sj})^2 - \sum_{j=1}^{J(s)} P_{sj} \sum_{l=1}^{L} H_{sj}^l q_l(t) \\ \text{subject to} \quad & P_{sj} \geq 0, \sum_{j=1}^{J(s)} P_{sj} = 1, \end{aligned} \tag{14}$$

  where $[H_{sj}^l]$ is the routing matrix, and β_s is a positive number chosen for each source s. Each user s then routes $P_{sj}(t)$ of the arrival traffic to path j.

- Step 2: This step is the same as Step 1 of Algorithm SP, except that the queue update

(5) becomes

$$q_l(t+1) = q_l(t) + \sum_{s=1}^{S} H_{sj}^l P_{sj}(t)\lambda_s - \sum_{c=1}^{C} y_l^c(t).$$

- Step 3: This step is the same as Step 2 in Algorithm SP.

Remark: The local optimization problem (14) is formulated in such a way that the routing fraction P_{sj} will be larger for a path j with a smaller congestion cost $\sum_{l=1}^{L} H_{sj}^l q_l(t)$. Note that each user only needs to know the sum of the queue length q_l along its alternate paths. The quadratic term in the objective function of (14) is essential to prevent potential routing oscillation. To see this, note that if $\beta_s = 0$, then when (14) is solved for any user s that has multiple alternate paths, only paths that have the smallest cost $\sum_{l=1}^{L} H_{sj}^l q_l(t)$ will have positive P_{sj}. This property can easily lead to oscillation of the routing fractions P_{sj} when the queue length q_l is being updated [29]. On the other hand, with the addition of a quadratic term, the objective function of (14) becomes strictly concave. The optimal routing fraction then becomes a continuous function of the queue length q_l. Thus, routing oscillation is eliminated. The parameter β_s can also control how sensitive the routing fraction is with respect to the queue length. By setting β_s to be a larger constant, the routing fraction will become less sensitive to the transient queue dynamics.

The following proposition establishes the efficiency ratio of Algorithm MP.

Proposition 5 *Assume that the set of alternate paths are given. The efficiency ratio of Algorithm MP is $1/(\mathcal{K}+2)$. Further, if $M_i = C$ for all node i, then the efficiency ratio can be improved to $1/\mathcal{K}$.*

Remark: Our algorithm is perhaps most comparable to the joint channel assignment, scheduling and routing algorithm in [11], which has been shown to achieve an efficiency ratio of $(\min_i M_i)/(\mathcal{K}C)$ (see Theorem 2 of [11]). The algorithm in [11] is an offline and centralized algorithm. In contrast, Algorithm MP that we developed in this paper is distributed and much simpler. Further, when Algorithm MP uses the same set of paths as computed by the Linear-Programming based algorithm in [11], it can guarantee an efficiency ratio of $1/(\mathcal{K}+2)$, which is higher than that of [11] if some mesh network node only has a small number of radio interfaces. We refer the readers to the discussions in the Introduction regarding the difference between our work and [11], and the potential implications.

Proof of Proposition 5 : We only provide the proof for the first part of the result (i.e., the efficiency ratio $1/(\mathcal{K}+2)$). The second part of the result can be shown by modifying the proof as for Proposition 4. Use the same Lyapunov function in (7). Then following the steps in the proof of Proposition 3, we can bound the Lyapunov drift as,

$$\begin{aligned}
& \mathbf{E}\big[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)\big] \\
\leq\ & \sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}\big[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}P_{sj}\lambda_s-\sum_{c=1}^{C}x_l^c(t)\big] \\
& +\sum_{l=1}^{L}\sum_{c=1}^{C}\frac{\eta_l^c(t)}{r_l^c}\left[\sum_{k\in I_l}\frac{x_k^c(t)}{r_k^c}+\frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{x_k^d(t)}{r_k^d}+\frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{x_k^d(t)}{r_k^d}-\mu_l^c(t)\right] \\
& +C_4.
\end{aligned}$$

Since $(\mathcal{K}+2)\vec{\lambda}$ can be supported by some scheduling algorithm, similar to (12) and (13), there must exist some $[\bar{x}_l^c]$ and $[\bar{P}_{sj}]$, such that

$$(1+\epsilon)\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s\leq\sum_{c=1}^{C}\bar{x}_l^c,\quad \bar{x}_l^c\in[0,r_l^c], \tag{15}$$

$$(1+\epsilon)\left[\sum_{k\in I_l}\frac{\bar{x}_k^c}{r_k^c}+\frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}+\frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}\right]\leq 1.$$

We thus have,

$$
\begin{aligned}
& \mathbf{E}[\Delta V(t)|\vec{q}(t),\vec{\eta}(t)] \\
\leq\ & \sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}P_{sj}\lambda_s-\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s] && (16)\\
& \quad +\sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s-\sum_{c=1}^{C}\bar{x}_l^c] && (17)\\
& +\sum_{l=1}^{L}\frac{q_l(t)}{\alpha_l}[\sum_{c=1}^{C}\bar{x}_l^c-\sum_{c=1}^{C}x_l^c(t)] \\
& \quad +\sum_{l=1}^{L}\sum_{c=1}^{C}\frac{\eta_l^c(t)}{r_l^c}\Big[\sum_{k\in I_l}\frac{\bar{x}_k^c}{r_k^c}+\frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}+\frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{\bar{x}_k^d}{r_k^d}-\mu_l^c(t)\Big] \\
& \quad +\sum_{l=1}^{L}\sum_{c=1}^{C}\frac{\eta_l^c(t)}{r_l^c}\Big[\sum_{k\in I_l}\frac{x_k^c(t)-\bar{x}_k^c}{r_k^c}+\frac{1}{M_{b(l)}}\sum_{k\in E(b(l))}\sum_{d=1}^{C}\frac{x_k^d(t)-\bar{x}_k^d}{r_k^d} \\
& \quad +\frac{1}{M_{e(l)}}\sum_{k\in E(e(l))}\sum_{d=1}^{C}\frac{x_k^d(t)-\bar{x}_k^d}{r_k^d}\Big]+C_4.
\end{aligned}
$$

Note that except the first two terms (16) and (17), the remaining terms are the same as in the proof of Proposition 3. To show a negative Lyapunov drift, it only remains to ensure that the first two terms ((16) and (17) have negative drifts. Note that

$$
\begin{aligned}
& \sum_{l=1}^{L}q_l(t)[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}P_{sj}\lambda_s-\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s] \\
=\ & \sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{\beta_s(P_{sj})^2}{2}+\sum_{l=1}^{L}q_l(t)[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}P_{sj}\lambda_s] \\
& -\sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{\beta_s(\bar{P}_{sj})^2}{2}-\sum_{l=1}^{L}q_l(t)[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s] \\
& -\sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{\beta_s(P_{sj})^2}{2}+\sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{(\beta_s\bar{P}_{sj})^2}{2}.
\end{aligned}
$$

According to Step 1 of Algorithm MP,

$$
\begin{aligned}
& \sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{\beta_s(P_{sj})^2}{2}+\sum_{l=1}^{L}q_l(t)[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}P_{sj}\lambda_s] \\
\leq\ & \sum_{s=1}^{S}\sum_{j=1}^{J(s)}\frac{\beta_s(\bar{P}_{sj})^2}{2}+\sum_{l=1}^{L}q_l(t)[\sum_{s=1}^{S}\sum_{j=1}^{J(s)}H_{sj}^{l}\bar{P}_{sj}\lambda_s].
\end{aligned}
$$

Hence, the term (16) can be bounded by

$$\begin{aligned}
&\sum_{l=1}^{L} \frac{q_l(t)}{\alpha_l} [\sum_{s=1}^{S} \sum_{j=1}^{J(s)} H_{sj}^l P_{sj} \lambda_s - \sum_{s=1}^{S} \sum_{j=1}^{J(s)} H_{sj}^l \bar{P}_{sj} \lambda_s] \\
\leq \ & -\sum_{s=1}^{S} \sum_{j=1}^{J(s)} \frac{\beta_s (P_{sj})^2}{2\alpha_l} + \sum_{s=1}^{S} \sum_{j=1}^{J(s)} \frac{(\beta_s \bar{P}_{sj})^2}{2\alpha_l} \\
\leq \ & \sum_{s=1}^{S} \frac{\beta_s J(s)}{\alpha_l}.
\end{aligned}$$

On the other hand, using (15), the second term (17) is less than

$$-\sum_{l=1}^{L} \frac{\epsilon \sum_{s=1}^{S} \sum_{j=1}^{J(s)} H_{sj}^l \bar{P}_{sj} \lambda_s}{\alpha_l} q_l(t).$$

The rest of the proof then follows along the same line as the proof of Proposition 3. *Q.E.D.*

4.3 How to Generate Alternate Paths

The set of alternate paths, denoted by the matrix $[H_{sj}^l]$, could potentially be the enumeration of all possible paths between each source-destination pair. In practice, however, a much smaller set of alternate paths suffices. We next describe options to compute and maintain this set of alternate paths.

Option 1: Use paths that appear to be "heuristically good." For example, given a source-destination pair, we can use various known routing schemes in the literature (e.g., in [8, 14]) to generate a number of potential paths.

Option 2: Use historical data. This can be viewed as a traffic engineering step. We first take measurements of typical traffic demands at different times of the day. For each demand pattern, we can compute the optimal paths *offline*, e.g., using the linear programming approach in [11]. The union of the alternate paths under all demand patterns can then be used as the set of candidate paths. The role of Algorithm MP is to shift the traffic load among these candidate paths automatically as the network condition changes.

Option 3: An even better approach is to discover new paths online. The queue-length q^l provides us with the signal to discover potentially better alternate paths. Given a set of alternate paths, we can easily verify the following property for Step 1 of Algorithm MP: For each user s, only paths j that satisfy the following condition will see positive routing fractions P_{sj}:

$$\beta_s P_{sj} + \sum_{l=1}^{L} H_{sj}^l q_l = \max_{k=1,..,J(s)} \beta_s P_{sk} + \sum_{l=1}^{L} H_{sk}^l q_l \triangleq q_{s,\max}, \tag{18}$$

Therefore, adding paths with congestion costs $\sum_{l=1}^{L} H_{sj}^l q_l$ larger than $q_{s,\max}$ will not yield any gain.

We can use the above property to iteratively generate the candidate paths online. Starting from any initial set of candidate paths, we execute Algorithm MP for joint routing, channel-assignment and scheduling. Then, from time to time, we can run a minimal cost routing algorithm using the queue-length as the cost metric for each link. If the minimal cost is smaller than $q_{s,\max}$ defined in (18) by a certain threshold, we add this new path into the set of alternate paths, and continue. Otherwise, we can conclude that no further alternate paths need to be added.

5 Conclusion

In this paper, we develop a distributed algorithm that jointly solves the channel-assignment, scheduling and routing problem for multi-channel multi-radio networks. Our algorithm is very simple, does not require prior information on the offered load to the network, and can adapt automatically to the changes in the network topology and offered load. We show that our algorithm is provably efficient. That is, even compared with the optimal centralized and offline algorithm, our proposed algorithm can achieve a provable fraction of the maximum system capacity.

References

[1] 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. 37, no. 12, pp. 1936–1948, December 1992.

[2] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic Power Allocation and Routing for Time Varying Wireless Networks,” in *Proceedings of IEEE INFOCOM*, San Francisco, April 2003.

[3] J. So and N. H. Vaidya, “Routing and Channel Assignment in Multi-Channel Multi-Hop Wireless Networks with Single Network Interface,” in *2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, August 2005.

[4] P. Kyasanur and N. H. Vaidya, “Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks,” University of Illinois, Urbana-Champaign, Tech. Rep., October 2004.

[5] J. So and N. H. Vaidya, “Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver,” in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2004.

[6] A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network,” in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[7] A. Raniwala, K. Gopalan, and T. Chiueh, “Centralized Algorithms for Multi-channel Wireless Mesh Networks,” in *ACM Mobile Computing and Communications Review*, April 2004.

[8] R. Draves, J. Padhye, and B. Zill, “Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks,” in *ACM MobiCom*, Philadelphia, Pennsylvania, September 2004.

[9] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, “A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks,” in *Proceedings of IEEE Broadnets 2004*, San Jose, CA, October 2004.

[10] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering Wireless Systems with Multiple Radios," in *ACM Sigcomm Computer Communications Review*, October 2004.

[11] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," in *ACM MobiCom*, Cologne, Germany, August 2005.

[12] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," in *ACM MobiCom*, Cologne, Germany, August 2005.

[13] P. Kyasanur and N. H. Vaidya, "Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces," in *ACM Mobicom*, Cologne, Germany, August 2005.

[14] ——, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, January 2006.

[15] X. Lin and N. B. Shroff, " The Impact of Imperfect Scheduling on Cross-Layer Rate Control in Multihop Wireless Networks," in *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.

[16] X. Wu and R. Srikant, "Bounds on the Capacity Region of Multihop Wireless Networks Under Distributed Greedy Scheduling," in *Proceedings of IEEE INFOCOM*, Barcelona, Spain, April 2006.

[17] X. Wu, R. Srikant, and J. R. Perkins, "Queue-Length Stability of Maximal Greedy Schedules in Wireless Network," in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[18] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput Guarantees in Maximal Scheduling in Wireless Networks," in *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2005.

[19] ——, "Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks," in *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[20] X. Lin and S. Rasool, "Constant-Time Distributed Scheduling Policies for Ad Hoc Wireless Networks," *to appear in IEEE CDC 2006, also available at http://min.ecn.purdue.edu/∼linx/papers.html*, 2006.

[21] Y. Liu and E. Knightly, "Opportunistic Fair Scheduling over Multiple Wireless Channels," in *Proceedings of IEEE INFOCOM*, San Francisco, CA, April 2003.

[22] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Communications magazine*, pp. 70–77, July 2000.

[23] X. Liu, E. K. P. Chong, and N. B. Shroff, "A Framework for Opportunistic Scheduling in Wireless Networks," *Computer Networks*, vol. 41, no. 4, pp. 451–474, March 2003.

[24] B. Hajek and G. Sasaki, "Link Scheduling in Polynomial Time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, September 1988.

[25] X. Wu and R. Srikant, "Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks With Node-Exclusive Spectrum Sharing," in *Proceedings of IEEE CDC*, Seville, Spain, December 2005.

[26] S. H. Lu and P. R. Kumar, "Distributed Scheduling Based on Due Dates and Buffer Priorities," *IEEE Transactions on Automatic Control*, pp. 1406–1416, December 1991.

[27] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "Maximum Weighted Matching with Interference Constraints," in *IEEE International Workshop on Foundations and Algorithms For Wireless Networking*, Pisa, Italy, March 2006.

[28] M. Hanckowiak, M. Karonski, and A. Panconesi, "On the Distributed Complexity of Computing Maximal Matching," *SIAM Journal of Discrete Mathematics*, vol. 15, no. 1, pp. 41–57, 2001.

[29] X. Lin and N. B. Shroff, "Utility Maximization for Communication Networks with Multi-path Routing," *IEEE Transactions on Automatic Control*, 2007, to appear.