Resource Allocation in Multi-Radio Multi-Channel Multi-Hop Wireless Networks

Technical Report (July 2007)

Simone Merlin †, Nitin Vaidya ♯, Michele Zorzi †
† Padova University, DEI, Padova, Italy; ♯ University of Illinois at Urbana-Champaign
simone.merlin@dei.unipd.it, nhv@uiuc.edu, michele.zorzi@dei.unipd.it

***Abstract*—A joint congestion control, channel allocation and scheduling algorithm for multi-channel multi-interface multi-hop wireless networks is discussed. The goal of maximizing a utility function of the injected traffic, while guaranteeing queues stability, is defined as an optimization problem where the input traffic intensity, channel loads, interface to channel binding and transmission schedules are jointly optimized by a dynamic algorithm. Due to the inherent NP-Hardness of the scheduling problem, a simple centralized heuristic is used to define a lower bound for the performance of the whole optimization algorithm. The behavior of the algorithm for different numbers of channels, interfaces and traffic flows is shown through simulations.**

I. Introduction

New challenges in wireless network design refer to a more efficient bandwidth utilization and the use of new networking paradigms. The former goal is related to the growing bandwidth demand and the scarcity of available spectrum. The latter refers to the need for flexible and easy deployment, self configuration and adaptation to the working condition. Multi-hop wireless networks have been identified as a valuable networking paradigm able to fulfil the previous requirements. Examples of multi-hop wireless networks include ad hoc networks and mesh networks. Practical interest in multi-hop wireless networks is confirmed by the recent development of standards which explicitly encompass the mesh paradigm, where the backhaul network is organized in an ad hoc topology. The IEEE 802.16 standard [3] is one example. In the context of 802.11 networks a special working group is dedicated to the mesh extension, which is referred to as 802.11s [4]. Other standardization efforts are focusing on the introduction of mesh-like support in their network architecture, such as 802.15.3/4, where the network architecture implicitly supports a mesh-like structure, and 802.15.5, which is working to define a mesh structure for personal area networks. It is clear that a deep understanding and the ability to optimize the performance of multi-hop wireless networks will benefit significantly in these contexts.

With the motivation of improving performance of multi-hop wireless networks, in the last few years great attention has been devoted to networks where each node is provided with multiple radio interfaces and can operate on multiple channels [1]. This new degree of freedom has been proved to potentially allow for increased capacity with respect to single-channel single-interface networks [2]. This approach is particularly interesting if applied to 802.11 networks, since multiple channels are already available and devices provided with multiple wireless networking cards are being designed and already exist in some testbeds.

A lot of effort has also been spent in the last few years to understand the challenges related to resource allocation in such networks, where the increased number of variables to be jointly optimized has represented a big issue. The problem has been approached from different perspectives, ranging from heuristic and protocol oriented solutions [3], [4], [5], [6], whose performance is far from being exactly defined, to the determination of theoretical bounds [7], [8], [9], whose practical implementation is not straightforward. It is thus worth investigating an approach aiming at the design of practical algorithms based on a solid theoretical background, which can be analytically proved to guarantee some performance bounds [9].

In this paper, we consider the problem of joint congestion control, channel allocation and scheduling for multi-hop wireless networks in a general communication and interference scenario. The problem is formulated as a joint optimization, which is then solved by a dynamic algorithm and is potentially able to achieve the optimum solution under certain assumptions. A specific simplified scenario is also evaluated, where the scheduling is actually an inherently NP-Hard problem, and thus a heuristic is proposed. The channel loading and scheduling approach is somewhat similar to the one proposed in [9] but this paper focuses on a throughput optimal approach [10], is inherently multi-hop, and congestion control is also integrated in the framework. We build on the past work on network utility optimization (discussed in Section III), by using the notion of *virtual links* to facilitate analysis of multi-channel networks.

The paper is organized as follows. The complete system model and the goal of the proposed analysis are presented in Section II. Related work is reviewed in Section III. The optimization problem is formulated in Section IV and the proposed solution is presented in Section V together with stability issues, addressed in Section VI. The scheduler is defined in Section VII and simulation results for the whole algorithm are in Section VIII. Conclusions end the paper.

II. System model

Our system model is derived from similar models used in past work [10], [9], with suitable modifications to capture the availability of multiple channels, as described below.

Consider a multi-hop wireless network. Each node in $\{n : n = 1, \ldots, N\}$ is provided with I_n half duplex wireless interfaces. At any given time, each interface can tune to any

Research reported here is supported in part by U.S. National Science Foundation award CNS 06-27074, U.S. Army Research Office grant W911NF-05-1-0246 and "Ing. Aldo Gini" Foundation, Padova, Italy. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

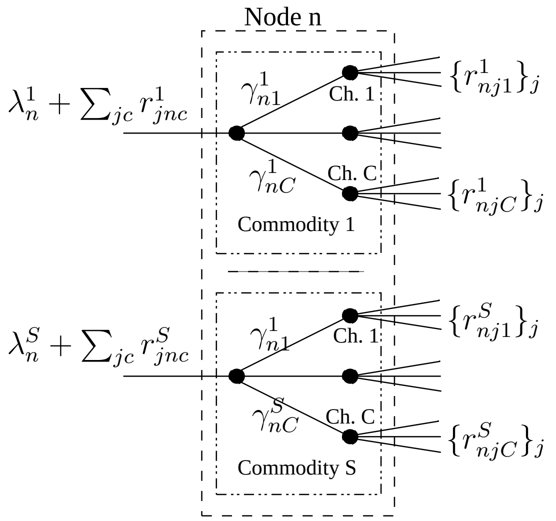

Fig. 1. Node model

one of C channels $\{c : c = 1, \ldots, C\}$. The channel used by an interface may change over time. For the algorithm definition, a general interference model is initially assumed (which can also encompass non-orthogonal channels). In Section VII, a simplified interference model based on orthogonal channels, and communication and interference graphs, is used in order to define a greedy heuristic.

Traffic flows are, in general, carried over multi-hop routes. Each end-to-end unicast connection will be referred to as a *commodity* in the following. Let $\{s : s = 1, \ldots, S\}$ be the commodities set. The input rate for commodity s at node n is λ_n^s. Let $\overline{\lambda}$ be the vector of all input rates. Each input rate can assume values $\lambda_n^s \in \Lambda_n^s$.

As a result of the proposed algorithm, each node n will be provided with an input queue $U^s_{n,in}$ for each commodity s, and $C \times S$ output queues, $U^s_{n,c,out}$ one for each channel-commodity pair. All the incoming traffic for commodity s is loaded on queue $U^s_{n,in}$. Output queues for commodity s are loaded using packets stored in queue $U^s_{n,in}$, according to the policy described in the next sections. Inside each node, and for each commodity, a connection is defined between the input queue $U^s_{n,in}$ and each of the output queues $U^s_{n,c,out}$ on different channels, for the same commodity. Such connections will be referred to as *virtual links* in the following. Let $\gamma^s_{n,c}$ be the rate at which data is transferred from the input queue $U^s_{n,in}$ to the output queue $U^s_{n,c,out}$, i.e. the rate of the associated virtual link. Let $\overline{\gamma}$ denote the vector for all $\gamma^s_{n,c}$ and Ψ its feasible set, which represents the rate region for the virtual links. The set Ψ will be defined in Section VI, based on a stability argument and in order not to modify the capacity region of the actual network.

Let $r^s_{a,b,c}$ be the transmission rate associated with the flow between nodes a and b on channel c, carrying traffic for the commodity s and $\overline{r}$ be the corresponding vector for all a,b and c.

The physical layer capacity for the link between nodes a and b on channel c is denoted as $w_{a,b,c}$. Let us denote by $\overline{w}$ the vector consisting of $w_{a,b,c}$ for all nodes a, b and channel c. The feasible rate region, i.e. the set of all feasible $\overline{w}$ vectors, is denoted as $\mathcal{W}$, which depends on the interference model, and, in general, is also constrained by the limited number of wireless interfaces at each node.

The utility function for commodity s associated with each source node n is denoted $G^s_n(\lambda^s_n)$. To allow the use of convex optimization techniques, all the utility functions are assumed to be strictly concave, and the rate vectors $\overline{w}$ will actually be considered as belonging to the convex hull of the set $\mathcal{W}$, $\overline{w} \in$ Co($\mathcal{W}$). Similar assumptions have been made in past work as well [10], [11].

The goal of the proposed algorithm is to jointly define

- congestion control
- routing
- channel loading
- interface binding and scheduling

with provable properties in terms of stability (achieved when the following property is satisfied: $\lim_{t\to\infty} E[\sum_{n,c,s}(U^s_{n,c,out} + U^s_{n,in})] < +\infty$) and network utility maximization.

In the following, a general formulation is presented in terms of an optimization problem on a network flow. A Lagrangian relaxation allows to define a distributed utility maximization, channel loading and scheduling which turns out to be based on the concept of "back pressure" scheduling [10]. Our approach makes use of "virtual links" for loading the queues on each channel. A stability issue in the definition of virtual link rates is discussed below, and a Lyapunov argument is used to justify the solution. A heuristic way to solve the scheduling optimization is also discussed for the case of a simplified transmission and interference model. A lower bound for the performance of the joint algorithm is also identified later in the paper.

III. Related work

The concept of "layering as optimization decomposition" has been investigated in the last few years as a powerful way to analytically define cross layer optimization problems and at the same time design feasible algorithms for their solution [11]. In particular, joint algorithms for congestion control and transmission scheduling have been proposed [12] which are able to jointly optimize source rate and link scheduling [13], [10], [14] including also the power control operation [15], [16]. The mathematical tools widely used in this new approach are essentially optimization problem decomposition by Lagrange relaxation, sub gradient algorithms and Lyapunov stability [17], [18]. The work presented in this paper is based on the decomposition of an optimization problem defined over the multi-channel network model.

The solution is related to the general scheduling algorithm presented in [10], [15]. In that theoretical formulation, the stability of the system is defined as the property of having bounded queue lengths, and the capacity region for the network is defined as the set of input rates for which there exists an algorithm which is able to keep the network stable. Given a set of input rates which lies inside the capacity region of the system, then the proposed algorithm is able to guarantee stability. The core of the scheduler is based on the maximization of a metric which depends on the rate allocated to each link, multiplied by the difference between the queue length at the link receiver side minus the queue length at the transmitter side (thus the name "back pressure"). In [10], a congestion controller is added on top of the scheduling algorithm which is proved to converge to a solution close to the optimum.

The impact of an imperfect scheduler in the joint scheduling and congestion control can in general lead to poor performance [19]. In case an imperfect scheduler is used, the joint algorithm presented in [10] is proved to be able to guarantee the stability within a capacity region scaled by a factor which depends on the imperfect scheduler. This opens the way to the implementation of reduced complexity schedulers.

The most challenging part of the previous approach refers to the scheduling operation, which in general requires a centralized and potentially very complex maximization. To move toward a practical solution, the following simplifications in the interference model are commonly considered: each link l only interferes with a set $\eta(l)$ of neighboring links, and each link has capacity c_l when scheduled.

In this case the scheduling, for a single channel scenario, becomes a weighted maximum independent set problem. The problem is in general NP-hard [20]. Clearly, a greedy centralized algorithm which selects at each step the link with the highest metric and discards all the interfering links can achieve a capacity region reduced by a factor of $1/K$ where K is the interference degree [19]. In [21], it is pointed out that such a greedy approach is optimal in graphs with particular structure. A novel approach for the same problem is also proposed in [22], where a gossiping algorithm is used.

Algorithms based on a maximal independent set scheduler (non weighted) are known for single hop networks and are presented in [23], [24], but this approach can not be extended to the multihop case. In this case a different scheduler has to be used, which exploits additional information on the traffic intensity or number of hops [25].

The closest work for multi-channel multi-radio wireless networks is the one in [9]. The authors propose a channel loading mechanism which, combined with a multi-channel maximal scheduler, is able to keep the network stable inside a subset of the capacity region. The network model is such that each node is provided with an input queue for each commodity and an output queue for each commodity-channel pair. A known traffic rate is applied at each input queue for each node and, based on a metric accounting for the queue lengths of all interfering nodes, a channel loading policy is defined. A multi-channel maximal scheduler is then applied to schedule the backlogged links. This approach is extended to the multihop case only for the case where information on the source rate is available and a congestion control is not considered.

An optimization approach is also used in [8], where an LP network flow problem is defined to model routing and channel loading. The solution is used to obtain an upper bound for the performance. A greedy scheduler based on the outcome of the previous LP solution is then applied for solving the actual resource allocation. A similar analysis is also found in [7].

In this paper, a network structure similar to the one in [9] is assumed, and the optimization approach is based on the decomposition presented in [11] and the argumentation in [10]. We propose an algorithm that works in a multihop scenario, and whose simple channel loading mechanism is based only on local information. The complexity is moved to the scheduling operation, which in general can be very complex. The proposed algorithm makes use of one input queue at each node for each commodity and one output queue for each channel-commodity pair at each node. The queue lengths are used, at each time step, to make dynamic decisions about congestion control, channel loading and transmission scheduling. In particular, "virtual links" are introduced in order to model the channel loading operation. The algorithm is analytically formulated and then tested by simulations in a simplified communication and interference scenario. The impact of numbers of channels, interfaces and commodities in the network performance is investigated.

IV. Formulation as an optimization problem

The goal of the proposed algorithm is to solve the following optimization problem (see Table I for a summary of the symbol definitions):

$$\max_{\overline{\lambda},\overline{r},\overline{w},\overline{\gamma}} \sum_{n,s} G_n^s(\lambda_n^s) \quad (1)$$

s.t.:

$$\sum_{i,c} r_{i,n,c}^s + \lambda_n^s \leq \sum_c \gamma_{n,c}^s \;\forall n,s \quad (2)$$

$$\gamma_{n,c}^s \leq \sum_j r_{n,j,c}^s \;\forall n,c,s \quad (3)$$

$$\sum_s r_{i,n,c}^s \leq w_{i,n,c} \;\forall i,n,c \quad (4)$$

$$\overline{\gamma} \in Co(\Psi) \quad (5)$$

$$\overline{w} \in Co(\mathcal{W}) \quad (6)$$

$$\lambda_n^s \in \Lambda_n^s \;\forall n,s \quad (7)$$

In the previous model:

- (1) is the objective function
- (2) is the flow conservation constraint at the input of each node
- (3) is the flow conservation constraint at the output of each node
- (4) is the constraint that the aggregate flow on a link must be less than the physical rate
- (5) is the constraint on the flow in the virtual links for the channel loading: this will be specified to model different requirements. Note the convex hull operator.
- (6) is the feasible rate regions for the actual links.
- (7) is the feasible set for the input rates.

Symbols:	
$G_n^s(\lambda_n^s)$	Utility function
$\overline{\lambda} = [\lambda_n^s]$	Injected input rate
$\overline{r} = [r_{a,b,c}^s]$	Flows associated to channel-link-commodity connections
$\overline{\gamma} = [\gamma_{n,c}^s]$	Flows that load output channel-commodity queues
$\overline{w} = [w_{a,b,c}]$	Physical rates associated to physical channel-link
Ψ	Feasible "virtual rate region" for channel loading
$\mathcal{W}$	Feasible rate region for actual physical links
Λ_n^s	Feasible input rates

TABLE I
Symbols

Based on the assumption on the utility functions and on the convexity of the domain, (1)-(7) is a convex optimization problem.

V. NETWORK FLOW OPTIMIZATION

The solution to the optimization problem is obtained via its dual problem, relaxing all the constraints (2) and (3).

Let $\mathbf{U_{in}} = [U^s_{n,in}]$ and $\mathbf{U_{out}} = [U^s_{n,c,out}]$ be the vectors for all the Lagrange multipliers associated to constraints (2) and (3) respectively. Let $\mathbf{U} = [\mathbf{U_{in}}, \mathbf{U_{out}}]$ be the vector for all the Lagrange multipliers.

Relaxing the constraints (2) and (3), the Lagrange dual function for the problem is:

$$L(\mathbf{U}) = \max_{\overline{\lambda},\overline{r},\overline{w},\overline{\gamma}} \left\{ \sum_{n,s} G^s_n(\lambda^s_n) + \right.$$
$$+ \sum_{n,s} U^s_{n,in} \left(- \sum_{j,c} r^s_{j,n,c} - \lambda^s_n + \sum_c \gamma^s_{n,c} \right) +$$
$$\left. + \sum_{s,c,n} U^s_{n,c,out} \left(-\gamma^s_{n,c} + \sum_j r^s_{n,j,c} \right) \right\},$$

where the optimization variables $\overline{\lambda}, \overline{r}, \overline{w}, \overline{\gamma}$ are still subject to constraints (4)-(7) (here, and in the following, constraints are omitted to simplify notation).

The previous expression can be rewritten as:

$$L(\mathbf{U}) = \max_{\overline{\lambda}} \left\{ \sum_{n,s} G^s_n(\lambda^s_n) - \lambda^s_n U^s_{n,in} \right\} + \tag{8}$$
$$+ \max_{\overline{r},\overline{w}} \left\{ \sum_{i,j,s,c} \left(U^s_{i,c,out} - U^s_{j,in}\right) r^s_{i,j,c} \right\} + \tag{9}$$
$$+ \max_{\overline{\gamma}} \left\{ \sum_{s,n,c} \left(U^s_{n,in} - U^s_{n,c,out}\right) \gamma^s_{n,c} \right\}. \tag{10}$$

Note how each maximization represents a different "layer" in the optimization task:

- (8) congestion control;
- (9) flow allocation, routing and physical rate allocation;
- (10) channel management (stability, channel loading, ...)

Let $\tilde{\lambda}(\mathbf{U})$, $\tilde{r}(\mathbf{U})$, $\tilde{w}(\mathbf{U})$, $\tilde{\gamma}(\mathbf{U})$ be the vectors of optimum values for a given set of Lagrange multipliers, that clearly depend on $\mathbf{U}$. Each of them can be computed locally, based only on local information, except for $\tilde{r}$ and $\tilde{w}$ in (9) which require the knowledge of the feasible rate region $\mathcal{W}$. In particular, in order to optimize (9), for each link between nodes i and j on channel c define $s^* = \arg\max_s \left\{ \left(U^s_{j,c,out} - U^s_{i,in}\right) \right\}$. The flow allocation is given by setting $r^{s^*}_{i,j,c} = w_{i,j,c}$ and $r^s_{i,j,c} = 0$ for $s \neq s^*$. Once chosen the flow to be potentially loaded on a physical link, the following maximization has to be performed: $\tilde{w} = \arg\max_{\overline{w}} \left\{ \sum_{i,j,c} \left[U^{s^*}_{i,c,out} - U^{s^*}_{j,in}\right]^+ w_{i,j,c} \right\}$. This is the back pressure algorithm [10].

In Section VII, an assumption about a specific feasible rate region will be discussed, and the design of a greedy algorithm to compute $\tilde{w}$ will be presented, together with a lower bound performance index.

Suppose for the moment that the only constraint imposed to the virtual link rates is:

$$\sum_{c,s} \gamma^s_{nc} < \Gamma_n$$

were each Γ_n is a constant, which is set according the stability and capacity preservation discussed in Section VI.

Under this assumption, the maximization in (10) requires that for each node n and commodity s, $c^* = \arg\max_c \left\{ \left(U^s_{n,in} - U^s_{n,c,out}\right) \right\}$ is chosen. If $\left(U^s_{n,in} - U^s_{n,c^*,out}\right) > 0$ then set $\gamma^s_{n,c^*} = \Gamma_n$ and all $\gamma^s_{n,c} = 0$ for $c \neq c^*$, else set all $\gamma^s_{n,c} = 0\ \forall c$ (so that the summation is 0; otherwise the summation would be negative). This is essentially the back pressure based algorithm for the virtual links $\overline{\gamma}$.

The Lagrange function is convex thus the multipliers can be computed using a sub gradient algorithm. It is known that a sub gradient for a given vector of Lagrange multipliers is the vector consisting of all multiplicative terms in the Lagrange function. Note that such multiplicative terms are the results of maximizations (8)–(10). With this choice, the Lagrange multipliers are computed using a sequential algorithm which, at each step, updates them based on the value of the local sub gradient. Let t be the iteration index, which can be associated with a time-slot in the system evolution. Thus the updating rules for each multiplier at time $t+1$ are:

$$U^s_{n,in}(t+1) = \Big[U^s_{n,in}(t) + \alpha_1 (\tilde{\lambda}^s_n(\mathbf{U}(t)) + \\ + \sum_{j,c} \tilde{r}^s_{j,n,c}(\mathbf{U}(t)) - \sum_c \tilde{\gamma}^s_{n,c}(\mathbf{U}(t))) \Big]^+ \tag{11}$$

$$U^s_{n,c,out}(t+1) = \Big[U^s_{n,c,out}(t) + \alpha_2 (\tilde{\gamma}^s_{n,c}(\mathbf{U}(t)) + \\ - \sum_j \tilde{r}^s_{n,j,c}(\mathbf{U}(t))) \Big]^+ . \tag{12}$$

Note that the Lagrange multipliers actually behave like queues in the case $\alpha_1 = \alpha_2 = 1$.

In order to get a solution which converges to a point close to the optimum, α_1 and α_2 should be set to be small constants. In the case $\alpha_1 = \alpha_2 = 1$ the solution will be oscillating around the optimum point.

Note that at each time t a new sub gradient has to be computed, thus the optimizations (8)–(10) has to be repeated at each time slot. Let $\tilde{\lambda}(t)$, $\tilde{r}(t)$, $\tilde{w}(t)$, $\tilde{\gamma}(t)$ denote the vectors solutions of the optimization optimization variables where the time index has been explicitly shown, neglecting the $\mathbf{U}$ to simplify notation.

Based on the previous argumentation, the proposed algorithm for joint congestion control, channel allocation and scheduling is presented in Algorithm 1.

VI. CHANNEL LOADING: STABILITY BY LYAPUNOV DRIFT

In the previous section the feasible rate set for the virtual links used for the channel loading has not been specified. Here it is proved that a sufficient condition for stability requires the aggregated rate of the virtual links, used for the channel loading, to be bounded.

The stability of the system is derived using a Lyapunov argumentation. Consider the input rate for all commodities as fixed (no congestion control) and assume it falls within the capacity region of the network.

Algorithm 1 Joint optimization

At each time step t, perform the following operations.

1) Congestion control. For each commodity s and node n:
   $\tilde{\lambda}_n^s(t) = \sup_{\{\lambda_n^s \in \Lambda_n^s\}} \left\{ G_n^s(\lambda_n^s) - \lambda_n^s U_{n,in}^s(t) \right\}$
2) Channel allocation. For each commodity s and node n:
   $c^* = \arg\max_c \left\{ \left(U_{n,in}^s(t) - U_{n,c,out}^s(t) \right) \right\}$
   if $\left(U_{n,in}^s(t) - U_{n,c^*,out}^s(t) \right) > 0$ **then**
   set $\tilde{\gamma}_{n,c^*}^s(t) = \Gamma_n$ and all $\tilde{\gamma}_{n,c}^s(t) = 0$ for $c \neq c^*$
   else
   set all $\tilde{\gamma}_{n,c}^s(t) = 0 \ \forall c$
   end if
3) Scheduling and routing. For each link between nodes i and j on channel c:
   $s^* = \arg\max_s \left\{ \left(U_{i,c,out}^s(t) - U_{j,in}^s(t) \right) \right\}$
   if $\left(U_{i,c,out}^{s^*}(t) - U_{j,in}^{s^*}(t) \right) > 0$ **then**
   set $\tilde{r}_{i,j,c}^{s^*}(t) = w_{i,j,c}(t)$ and all $\tilde{r}_{i,j,c}^s(t) = 0$ for $s \neq s^*$
   else
   set all $\tilde{r}_{i,j,c}^s(t) = 0 \ \forall s$
   end if
   $\tilde{w} = \arg\max_{\overline{w}} \left\{ \sum_{i,j,c} \left[U_{i,c,out}^{s^*} - U_{j,in}^{s^*} \right]^+ w_{i,j,c} \right\}$
4) Queues update:
   the queues are automatically updated according to the rules in (11) and (12) with $\alpha_1 = \alpha_2 = 1$.

The considered Lyapunov function is $L = \sum_{n,s} (U_{n,in}^s)^2 + \sum_{n,s,c} (U_{n,c,out}^s)^2$ and the proof is derived from the one in [10].

Consider the queue updating rules (11), (12) with $\alpha_1 = \alpha_2 = 1$. The drift associated to the Lyapunov function L is denoted with $\Delta(\mathbf{U}(t)) = E\left[L(\mathbf{U}(t+1)) - L(\mathbf{U}(t)) | \mathbf{U}(t)\right]$ and it can be easily bounded as

$$\Delta(\mathbf{U}(t)) \leq B + 2 \sum_{ns} U_{n,in}^s(t) E\left[\lambda_n^s(t)\right] + \quad (13)$$

$$-2E\left[\sum_{i,j,c,s} r_{i,j,c}^s [U_{i,c,out}^s(t) - U_{j,in}^s(t)] \,|\, \mathbf{U}(t) \right] + \quad (14)$$

$$-2E\left[\sum_{c,n,s} \gamma_{n,c}^s [U_{n,in}^s(t) - U_{n,c,out}^s(t)] \,|\, \mathbf{U}(t) \right] + \quad (15)$$

$$\sum_{n,s} \left(\sum_c \gamma_{n,c}^s \right)^2 + \sum_{n,c,s} (\gamma_{n,c}^s)^2 \quad (16)$$

where B is a constant term depending on the r terms.

According to Corollary 3.9 in [10], if the input rate λ_n^s (which is loaded only on the input queue) is such that $\lambda_n^s + \epsilon \ \forall n, s$ (for a small ϵ) lies inside the capacity region, then there exists a randomized scheduling $\hat{r}$, $\hat{w}$ and $\hat{\gamma}$, such that

$$E\left[\sum_c \hat{\gamma}_{n,c}^s - \sum_{i,c} \hat{r}_{i,n,c}^s \right] = \epsilon + \lambda_n^s \ \forall n, s, c$$

$$E\left[\sum_j \hat{r}_{n,j,c}^s - \hat{\gamma}_{n,c}^s \right] = 0 \ \forall n, s$$

and thus choosing a schedule $\tilde{r}, \tilde{w}, \tilde{\gamma}$ according to the maximization in (9) and (10), then (14) and (15) can be bounded leading to

$$\Delta(\mathbf{U}(t)) \leq B' - 2\epsilon \sum_{n,s} U_{n,in}^s +$$

$$+ \sum_{n,s} \left(\sum_c \gamma_{n,c}^s \right)^2 + \sum_{n,c,s} (\gamma_{n,c}^s)^2. \quad (17)$$

Note that if $\sum_{n,s}(\sum_c \gamma_{n,c}^s)^2 + \sum_{n,c,s}(\gamma_{n,c}^s)^2$ is bounded, the drift becomes negative as the queue lengths increase above a given threshold. Thus we define the feasible rate for the virtual link as $\Psi = \{\gamma_{n,c}^s : \sum_c \gamma_{n,c}^s < \Gamma_n\}$, with Γ_n suitable positive constants. The proposed network model artificially adds the virtual links to the original network structure, thus we have to make sure the resulting network is able to provide the same capacity region as the original one. To guarantee such a property, a value for each Γ_n can be chosen as the smallest value, greater than the maximum possible output rate for a node (which is bounded).

VII. Scheduling

The scheduler defined in general terms in the previous section requires a centralized optimization. Here a specific communication and interference model is considered which reduces the problem to a maximum weighted independent set with some constraints imposed by the reduced number of interfaces. Note that the maximum weighted independent set problem (defined for the single channel case) is known to be NP-Hard [20].

Assume a network with N nodes, C channels and I interfaces per node.

A secondary interference model is considered where each radio link (i, j, c) interferes with a set of surrounding links in the same channel $\eta(i, j, c)$. Inside this set only one transmission is possible at a given time. Each link has a known physical rate $\hat{c}_{i,j,c}$ when scheduled (without interference).

It is known [9], [19] that a greedy sequential and centralized algorithm that at each step selects the link with the highest metric and drops all the interfering links, can reach at least a fraction $\beta = 1/K$ of the maximum value in (9), where K is the maximum number of links that cannot be scheduled because of a given link has already been scheduled. Note that the constraint on the number of interfaces can cause a link activation to prevent the use of other links in different channels. Thus K depends on the topology, number of channels and interfaces.

Thus, this greedy scheduler allows for the solution of the whole optimization problem to converge to the optimum referred to a capacity region scaled by a factor β [10].

The previous lower bound is very conservative and the actual performance of the greedy procedure is expected to be much better than the stated bound. Some reasons are listed below:

i) the number of contending nodes is actually only the number of backlogged nodes with positive back pressure, thus, as long as the network is not heavily loaded, this number is much smaller than K;

ii) the loss in optimality β represents itself a lower bound. The bound assumes that each time a link is scheduled, all the dropped links have a weight which is close to the one of the scheduled link.

iii) the maximal scheduling is close to the maximum scheduling in most practical topologies [21].

In the following, the proposed cross layer algorithm has been tested using such a greedy centralized scheduler for a given topology.

VIII. Simulation Results

In this section the whole algorithm is tested using the greedy centralized scheduler previously described. The presented results are referred to a simplified topology and interference models and give some insight on the network performance as a function of the number of channels C, number of interfaces per node I, and number of commodities S.

A. Grid topology

The algorithm has been simulated in a single network snapshot composed by $N = 16$ nodes, placed in a regular mesh with a distance of 0.2 units between adjacent nodes.

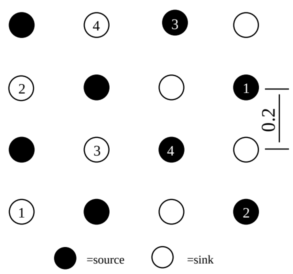

Fig. 2. Topology and commodities. Each number represents a different commodity.

Each node can potentially communicate with all neighbors within a distance of 0.3 and, when transmitting, it causes interference to neighbors within a distance of 0.3. All the communication links have the same capacity, which has been set equal to $1/C$, in order to allow for a comparison among results with different number of channels. Commodities source and destination have been associated with distinct nodes according to the scheme shown in Figure 2. The utility function is the same for all the nodes and is defined as $G_n^s(x) = log(x)$, which implements a fairness based congestion control. The system has been tested with $C = \{1, 2, 4, 8\}$, $I = \{1, ..., C\}$, $S = \{1, 2, 4\}$.

As can be seen from Figure 3, in all cases the aggregated utility increases as the number of interfaces increases. Anyway, the additional utility gained adding a new card decreases as the number of cards increases. For instance, in case of $C = 8$, only 4 interfaces are enough for achieving the maximum utility. This is in accordance with the asymptotic analysis presented in [2]. Please note that the utility is negative because a logarithmic function is used and, in the simulated scenario, each flow turns out to be smaller than one. Moreover the utility decreases as the number of concurrent flows increases. This is due to the specific scenario where the rate experienced by a single flow decreases as the number of flows increases.

Even if the throughput maximization is not the main goal of the simulated algorithm, in Figure 4 the aggregated transmission rate of all commodities is shown. Similarly to the utility behavior, the aggregated rate increases as the number of interfaces increases and the maximum value is reached using a number of interfaces smaller than the number of channels. As the number of commodities increases the aggregated rate

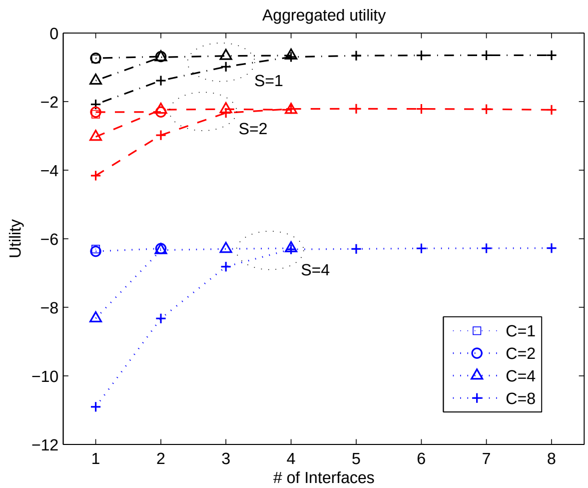

Fig. 3. Total utility for different numbers of channels, interfaces and commodities.

increases, showing that the spatial reuse of the medium is exploited.

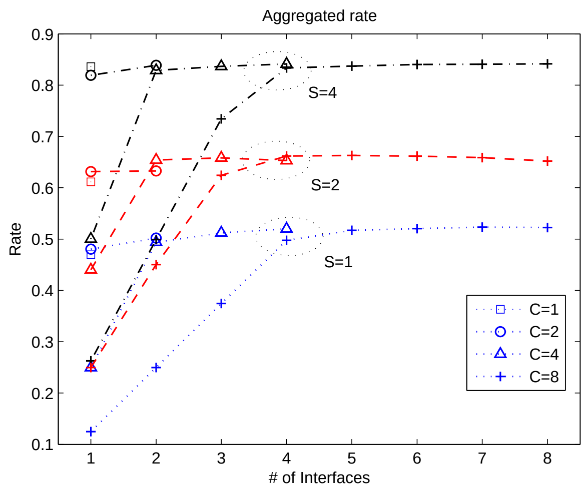

Fig. 4. Aggregated transmission rate for different numbers of channels, interfaces and commodities.

In Figure 5 the average queue length in the stationary regime is shown as a function of the number of interfaces and channels. As the number of interfaces increases the average length decreases. This has an impact on the end-to-end delay, which results to be smaller if a higher number of interfaces is used.

As the number of channels increases, the queue length decreases as well. Note in particular that in all cases $I = 4$, $C = \{4, 8\}$ the maximum throughput is reached (see Figure 4). On the other hand a higher number of channels allows for a reduced queue length and thus a reduced delay.

The proposed algorithm has been proved to asymptotically converge to the solution of the joint resource allocation problem, but the proposed analysis gives no insight on the time required for the algorithm to converge. Figure 6 shows a typical trend for the time evolution of aggregated queue lengths, aggregated transmitted rate by the sources and aggregated

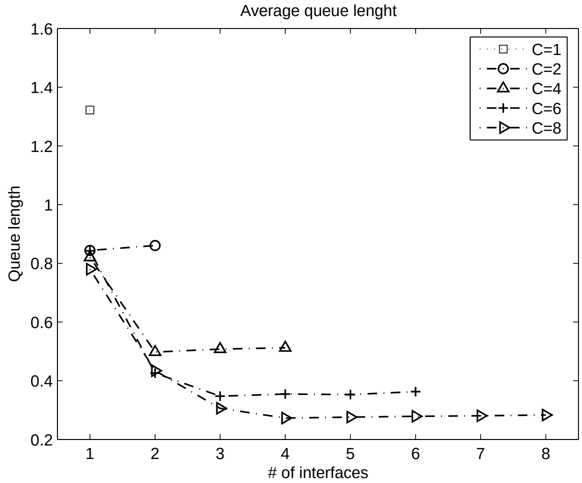

Fig. 5. Average queue length for different numbers of channels and interfaces. $S = 4$.

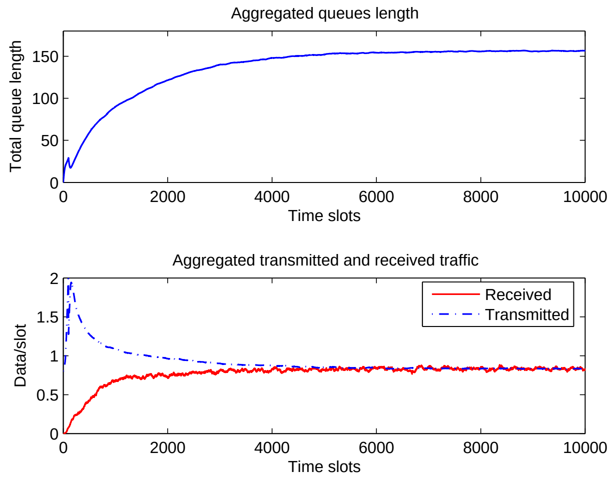

Fig. 6. System time evolution. The curves shown are averaged over a moving window of 100 samples. $C = 8$, $S = 4$, $I = 4$.

received rate at the sinks. As can be seen, the convergence is reached after a relatively high number of iterations. A more exhaustive investigation of the convergence time is presented in Figure 7 where the time needed to reach a stationary condition is plotted for different number of interfaces, channels, and commodities. As can be seen, the convergence time decreases as the number of interfaces increases, and increases as the number of channel or commodities increases.

Our interpretation for this behavior is that, as the number of queues in the system increases, more time is required for all the queues to be served and thus reach a stable configuration. This transient phase could be interpreted as a route discovery mechanism. Increasing the number of interfaces leads to a higher number of concurrent transmissions, which speeds up the convergence process.

In some cases the time required for convergence is very long. This can limit the practical implementation of such an algorithm in an actual network. A reason for the slow convergence is related to the routing mechanism, which imposes no constraints on the feasible paths for the traffic. The traffic thus can travel in all directions until a stable configuration is reached.

It would be interesting to define a policy for setting a reduced number of feasible paths for each commodity.

Convergence delay also depends on the particular congestion controller. A detailed investigation is out of the scope of this paper and represents an open research issue as pointed out in [11].

B. *Random topology*

The algorithm has also been tested using random topologies where nodes are uniformly placed in a unit square area. Presented results are averaged over 10 random topologies. Only connected topologies are considered. As in the previous case, each node can potentially communicate with all neighbors within a distance of 0.3 and, when transmitting, it causes interference to neighbors within a distance of 0.3.

In Figure 8 and Figure 9 the utility and the rate for different number of interfaces, channels and commodities are shown. Results are similar to the ones in Figure 3 and Figure 4,

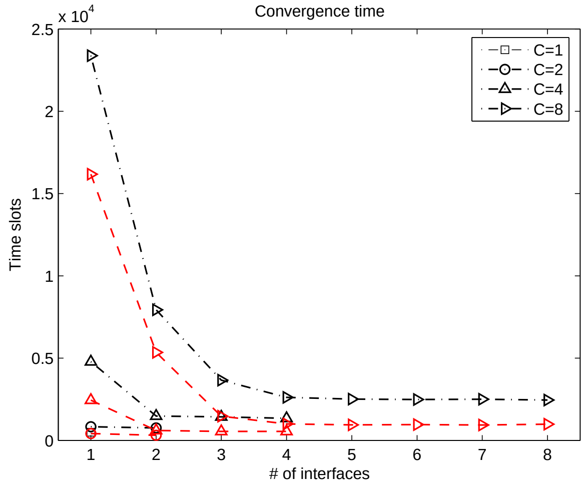

Fig. 7. Number of slots required for convergence. It is measured as the number of iterations needed to reach an aggregated queue length within 10% of the stationary value. Dashed: $S = 2$; Dash-dotted: $S = 4$.

confirming that the marginal utility and rate gained adding an interface is a decreasing function of the number of interfaces.

In Figure 10 (Figure 11) it is shown the ratio between the experienced utility (rate) for a given set of parameters and the maximum utility (rate) achieved with the highest number of interfaces. Results are averaged over $S = \{1, 2, 4\}$ and 10 random topologies for each value of S. Two different node densities are considered, which are obtained setting the average number of nodes within the communication and interference range to 3 and 7.

The behavior is similar to the one already described for the grid topology. It can be noted that a higher node density allows for a reduced number of interfaces needed to reach the same utility and rate values. Once again this is in accordance with the analysis in [2].

In Figure 12 the rate scaling factor with respect to the single channel case is plotted, as a function of the ratio between the number of channels and the number of interfaces. The behavior is similar to the one described in [2].

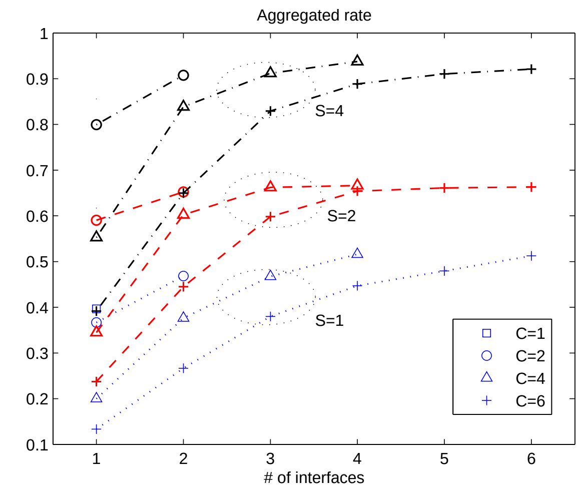

Fig. 8. Aggregated transmission rate for different numbers of channels, interfaces and commodities. Results are averaged over random topologies with 7 nodes in each communication range, on average.

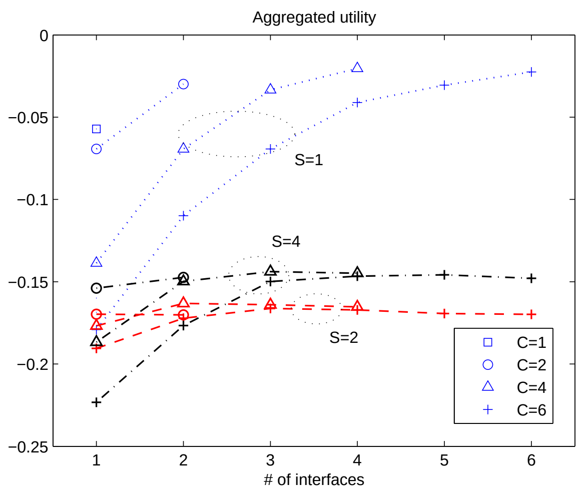

Fig. 9. Aggregated utility for different numbers of channels, interfaces and commodities. Results are averaged over random topologies with 7 nodes in each communication range, on average.

C. Comparison with results in [8]

The scenario considered in [8] has been reproduced and a comparison between the performance of our algorithm and the one presented in [8] has been made. The algorithm in [8] formulates the resource allocation as a network flow problem and solves a linear program problem in order to define an upper bound on the achievable performance. Then a greedy algorithm is applied for the scheduling operations. All the sources have the same traffic requirements (no congestion control) and the objective is to find the maximum input rate scaling factor for which a solution exists. Note that our algorithm aims at the utility maximization rather than to the maximization of the input rate scaling factor. Nonetheless, assuming a logarithmic utility, fairness among different flows is enforced, thus pushing our input rate scenario towards the one defined in [8].

Note that the optimization in [8] uses a centralized LP solution, while our algorithm can be run in a fully distributed way, as long as a distributed scheduling mechanism is available.

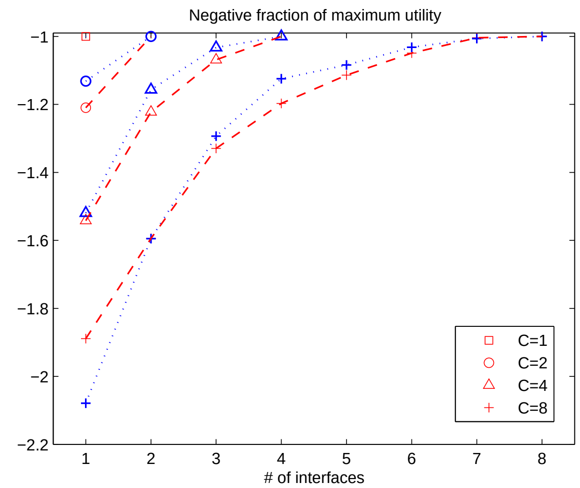

Fig. 10. Utility normalized with respect to the maximum utility attained with the maximum number of interfaces. The negative ratio is plotted in order to provide an easier comparison with Figure 11. Results are averaged over three different numbers of commodities $S = \{1, 2, 4\}$ and 10 random topologies. Node density: (D= nodes within communication range) dotted: $D = 7$, dashed: $D = 3$.

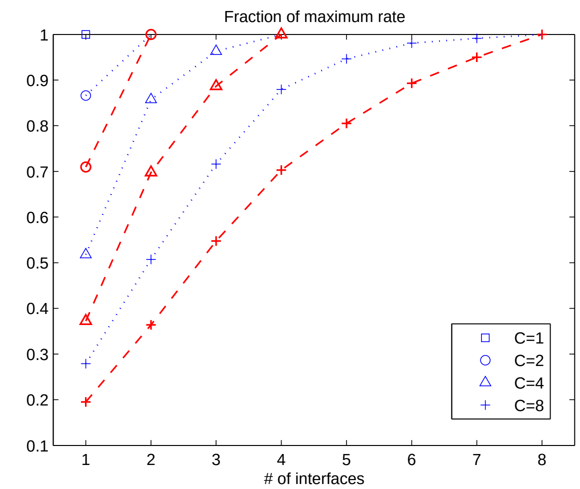

Fig. 11. Fraction of the maximum rate. Results are averaged over three different number of commodities $S = \{1, 2, 4\}$ and 10 random topologies. Node density: (D= nodes within communication range) dotted: $D = 7$, dashed: $D = 3$.

Thus our solution is closer to a practical implementation.

A grid 5×6 topology is considered; each node has at most 4 neighbors. 4 sinks for the traffic are considered (S=4) and results are averaged using $\{5, 10, 15, 20, 25\}$ traffic sources. Each sink is placed on a different quadrant. Sources are connected to the closest sink. Results are shown in terms of the aggregate rate scaling factor with respect to the single channel case. In this formulation each channel has a fixed capacity, so that the total bandwidth is increasing with the number of channels.

As [8] does not provide all the details of the considered scenario, in trying to reproduce it in our framework we had to make some assumptions. Although this makes a detailed quantitative comparison difficult, it still allows to verify that the two approaches exhibit consistent behaviors. Figure 13 shows

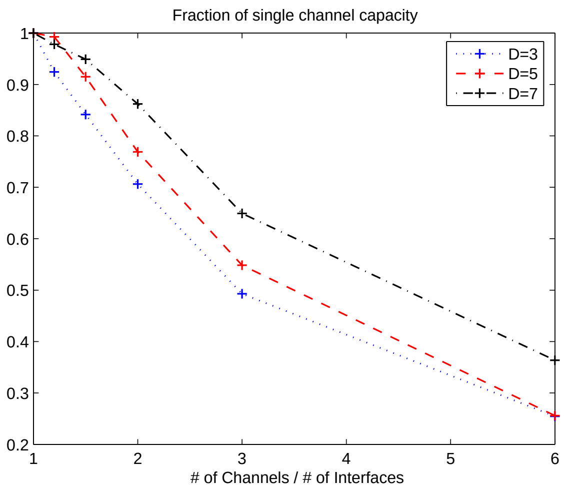

Fig. 12. Rate scaling factor with respect to the single channel case, as a function of the ratio between the number of channels and the number of interfaces. D is the average number of nodes in the communication and interfering range. Results are referred to $C = 6$, $I = \{1, 2, 3, 4, 5, 6\}$ and averaged over $S = \{1, 2, 4\}$ and 10 random topologies.

the rate gain in the two cases as a function of the number of channels and interfaces. It is clear from these results that the two approaches, though based on different techniques, have a qualitatively similar behavior. On the other hand, while the scheme in [8] is completely centralized and is more useful as a benchmark than as a practical solution, the features of our scheme make it easier to implement and therefore practically relevant.

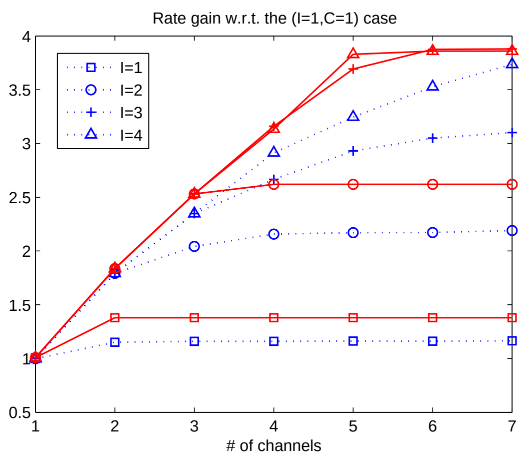

Fig. 13. Rate gain factor with respect to the single channel case. Dotted: our algorithm. Line: copied from Figures 7 and 8 in [8]

IX. Conclusions

A joint congestion control, channel allocation and scheduling algorithm for multi-channel multi-interface multi-hop wireless networks has been presented. The problem of maximizing a utility function of the source rate has been defined as an optimization problem and then solved by a dynamic algorithm.

The algorithm decomposes the whole optimization in different functional sub-optimizations and uses the queues length as a way to allow a joint solution of different optimization tasks.

A queue at the input of each node for each commodity and a queue at the output of each node for each channel-commodity pair have been used; a mechanism for loading the output queues on different channels has been defined introducing the notion of virtual links.

The algorithm has been presented for a general communication and interference scenario. In order to test the behavior of the full algorithm, an instance of the problem, based on a simplified communication and interference model, has been simulated using a greedy centralized scheduler.

The network performance has been evaluated as a function of the number of channels, interfaces and traffic flows. The results are consistent with previous theoretical findings, and confirm the goodness of the approach. On the other hand, the specific features of our algorithm make it more suitable for practical implementation in a distributed setting.

References

[1] P Kyasanur, J. So, C. Chereddi, and N. H. Vaidya. Multi-channel mesh networks: challenges and protocols. *IEEE Wireless Communication*, 2005. Invited paper.

[2] N.H. Vaidya and P. Kyasanur. Multi-channel wireless networks: capacity and protocols. Technical report, UIUC, 2005.

[3] N. Jain and S. Das. A multichannel CSMA MAC protocol with receiver based channel selction for multihop wireless networks. In *IC3N*, 2000.

[4] P. Kyasanur and N. Vaidya. Routing and link layer protocols form multi-channel multi-interface ad-hoc wireless networks. Technical report, UIUC, 2005.

[5] R Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04*, pages 114–128, 2004.

[6] A. Raniwala and T-C Chiueh. Architecture and algorithms for an IEEE 802.11 based multi-channel wireless mesh networks. In *INFOCOM*, 2005.

[7] M. Alicherry, R. Bathia, and L. Li. Joint channel asignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom*, 2005.

[8] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-raadio multi-channel wireless networks. In *MobiCom*, 2005.

[9] S. Rasool and X. Lin. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks. In *INFOCOM*, 2007.

[10] L. Tassiulas, L. Georgiadis, and M. J. Neely. *Resource Allocation and Cross-Layer Control in Wireless Networks*. Now Publishers Inc., 2006.

[11] M. Chiang, S.H. Low, A.R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[12] A.L. Stolyar. A tutorial on cross-layer optimization in wireless networks. *JSAC*, 24, 2006.

[13] X. Lin and N.B. Shroff. Joint rate control and scheduling in multihop wireless networks. In *CDC*, 2004.

[14] L. Chen, S. Low, M. Chiang, and J. Doyle. Cross layer congestion control, routing and scheduling design in ad-hoc wireless networks. In *INFOCOM*, 2006.

[15] M. Neely, E. Modiano, and C. Rohrs. Dynamic power allocation and routing for time-varying wireless networks. *JSAC*, 23, Jan 2005.

[16] M. Chiang. Balancing transport and physical layer in wireless multihop networks: jointly optimal congestion control and power control. *JSAC*, 23, Jan 2005.

[17] R. Srikant. *The mathematics of internet congestion control*. Birkhauser, 2003.

[18] P. D. Palomar and M. Chiang. A tutorial on decomposition methods for network utility maximization. In *JSAC*, volume 24, 2006.

[19] X. Lin and B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *INFOCOM*, 2005.

[20] G. Sharma and N. Shroff. Maximum weighted matching with interference constraints. In *PERCOMW06*, 2006.

[21] E. Modiano, G. Zussman, and A. Brzezinski. Enabling distributed throughput maximization in wireless mesh networks via local pooling. In *MIT Techreport*, 2006.

[22] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks with gossiping. In *SIGMETRIC*, 2006.

[23] X. Wu, R. Srikant, and J. R. Perkins. Queue length stability of maximal greedy schedules in wireless networks. In *ITA Workshop*, 2006.

[24] P. Chaporkar, K. Kar, and S. Sarkar. Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks. In *ITA*, 2006.

[25] R. Srikant and X. Wu. Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node exclusive spectrum sharing. In *CDC*, 2005.