Optimal Scheduling and Routing in Wireless Networks: *A New Approach*

Naghmeh S. Moayedian
Isfahan University of Technology
Isfahan, Iran
Email: moayedian@ec.iut.ac.ir

S. Jamaloddin Golestani
Isfahan University of Technology
Isfahan, Iran
Email: golestani@ieee.org

***Abstract*—A joint routing and scheduling algorithm for multi-hop wireless networks, based on a unified convex optimization framework, is proposed. Our approach is novel in that it integrates optimal scheduling with a modified version of distributed minimum delay routing. Accordingly, the algorithm performs packet routing based on a complete multi-hop view of the network and its traffic conditions. This stands in sharp contrast to joint routing and scheduling algorithms, such as the Tassiulas algorithm which is throughput-optimal, that rely on per-session queue differential between adjacent nodes for channel scheduling and packet routing. Simulation results illustrate that the proposed algorithm performs much better than Tassiulas, in terms of packet delay and jitter, packet loss and misordering, and energy consumption. Moreover, in terms of capacity region, simulation results do not reflect any noticeable difference between the two algorithms.**

I. Introduction

In this paper, we propose and study a joint routing and scheduling algorithm for multi-hop wireless networks, based on a single convex optimization problem. The novel aspect of our approach is that, in our formulation of the problem, we are able to integrate optimal scheduling with a modified version of the well-known minimum delay routing algorithm [2], [7]. As we shall see, this approach to joint routing and scheduling provides considerable performance advantage over the alternative approach discussed in [13], [11], [5], [10].

In recent years, there have been a large number of research papers dealing with the design of routing and scheduling algorithms, sometimes in conjunction with flow control and power control. In some of these works, routing and scheduling are formulated as optimization problems, each solved on a different time scale [6], [1], [9]. Other works deal with finding approximate solutions to graph theoretic problems, such as the maximum weighted independent set (MWIS) problem, that frequently arise in wireless network scheduling [3], [12], [8], [4]. To our knowledge, joint formulation of routing and scheduling as a unified problem has been previously addressed in [13], [11], [5], [10]. Of these, the theoretical approach and the proposed algorithm in [11] are similar to [13], while [5] and [10] derive a similar routing and scheduling algorithms, albeit using a different theoretical approach.

The scheduling and routing algorithm originated by Tassiulas in [13] (and further studied in [11], [5], [10]) is very attractive due to being conceptually simple yet powerful. The capacity region provided by this algorithm is proved to be optimal in the sense that no other algorithm can provide service to a set of input demands, if this algorithm fails to do so.

Tassiulas algorithm is essentially a scheduling algorithm that provides routing as a byproduct. Unfortunately, the routing aspect of the algorithm is not desirable since it does not use any multi-hop view of the network map and traffic conditions. The only information guiding packet routing through the network is the per-session queue differential between adjacent nodes. Until these queues are built up to appropriate levels, packets will wander in the network in every possible direction. The result, as we show in Section VI, is poor performance due to unnecessary increase in packet delay and delay jitter, packet misordering, packet loss, and energy consumption. These problems will become more pronounced as the network diameter in hop count is increased.

In this paper, we circumvent the above problems by taking a totally different approach to the joint formulation of scheduling and routing problems. Our convex optimization formulation essentially integrates optimal scheduling with a modified version of the minimum delay routing algorithm [7], [2]. We discuss the optimality conditions and present an algorithm for the solution, which is distributed as far as its routing aspect is concerned. The scheduling aspect of the algorithm boils down to solving a maximum weighted independent set problem on the conflict graph of the network, as is the case in other approaches to optimal scheduling [13], [5], [8]. Simulation results on a wireless network with 15 nodes and 39 links show considerable superiority of our algorithm, compared to Tassiulas, in terms of packet delay and jitter, packet misordering, packet loss, and energy consumption. Moreover, these simulation results do not reflect any reduction of the capacity region in our algorithm, compared to the throughput-optimal Tassiulas algorithm.

The rest of the paper is organized as follows. In sections II and III, the system model and the problem formulation as convex optimization are respectively presented. In Section IV, we see how to reformulate the problem, in order to facilitate the distributed implementation of routing. The proposed minimum cost joint routing and scheduling (MCRS) algorithm is presented in Section V, while Section VI deals with the simulation results and performance comparison with the Tassiulas algorithm. We close the paper with some concluding remarks in Section VII.

II. System Model

Consider a wireless network with a set $\mathcal{N}$ of nodes and a set $\mathcal{L}$ of transmission links. Each link represents direct communication from a given node i to another node k, and is labeled by its corresponding ordered node pair (i, k). Note that link (i, k) is distinct from link (k, i). In this paper, we assume that there is a single transmission channel, shared by all nodes of the network. The extension of results to the case of multiple transmission channels is possible. We consider an interference model for the network, based on a conflict graph. In this graph, each link of the network is represented by a vertex. The presence of an arc between two vertices of the conflict graph indicates that the corresponding network links cannot be activated at the same time, due to interference. We denote the transmission rate of a node i, when one of its outgoing links (i, k) is activated, by μ_i. The implied assumption here is that the transmission rate μ_i is independent of the receiving node k, although our approach can be extended to more general scenarios. In our analysis here, we assume a static network topology so that the set of links $\mathcal{L}$ and node transmission rates μ_i do not change with time. In practical terms, this means that the topology changes on a time scale much slower than the time scales of concern in the algorithms to be discussed.

Denote by r_i^j the average rate of data entering the network at node i, and destined for the destination j. Let f_{ik} stand for the average rate of data sent over link $(i, k) \in \mathcal{L}$, and f_{ik}^j, $j \neq i$, denote the average rate of data sent over link (i, k) that is destined for j. Clearly, $f_{ik} = \sum_{j \neq i} f_{ik}^j$. We use the notification f to collectivity refer to all parameters f_{ik}^j, $(i, k) \in \mathcal{L}, j \neq i$. Similarly, we denote by r the set of parameters r_i^j, $j \neq i$.

For the purpose of channel scheduling, we divide time into equal slots and activate a subset of links during each slot; this subset of links must correspond to an independent set of the conflict graph, in order to avoid interference. Let there be M independent sets in the conflict graph of the network, denoted by $m = 1, ..., M$. Each independent set m may be specified by a vector $\vec{\eta}^m = [\eta_{ik}^m]$, with binary entries η_{ik}^m, $(i, k) \in \mathcal{L}$, defined as $\eta_{ik}^m = 1$, if (i, k) belongs to the independent set m, and $\eta_{ik}^m = 0$, otherwise. It turns out that each vector $\vec{\eta}^m$, specifies a feasible set of links to be activated during any given time slot. Therefore, we refer to each $\vec{\eta}^m$, as a *schedule*. We take a probabilistic view of the scheduling in the network, and denote the probability of using schedule $\vec{\eta}^m$ during an arbitrary time slot, by ψ_m. Any stationary scheduling policy used in the network, will result into some set of scheduling probabilities ψ_m, and, conversely, any set of scheduling probabilities ψ_m can be used to specify a scheduling policy. Obviously $\sum_{m=1}^{M} \psi_m = 1$. We use the notation Ψ to refer to the set of probabilities ψ_m.

It is clear from the above definitions that, during a given time slot, if links are activated according to the schedule $\vec{\eta}^m$, the transmission rate provided to link (i, k) would be $\mu_i\ \eta_{ik}^m$. Now, denote by C_{ik} the transmission rate provided to link (i, k), *on the average*. It follows that:

$$\begin{aligned} C_{ik} &= \sum_{m=1}^{M} \psi_m \text{ [transmission rate provided to } (i,k) \\ &\qquad\qquad \mid \text{schedule } m \text{ is chosen]} \qquad (1) \\ &= \sum_{m=1}^{M} \psi_m\ \mu_i\ \eta_{ik}^m \quad , \quad \forall (i,k) \in \mathcal{L}. \end{aligned}$$

We refer to C_{ik}, simply as the capacity of link (i, k). Notice that C_{ik} may be compared to the capacity of a wired link, with the difference that here, the transmission rate C_{ik} is not provided on a steady basis; the actual transmission rate is fluctuating and its average C_{ik}, depends on the scheduling probabilities ψ_m. Finally, let $\vec{C}$ be the vector of link capacities, i.e., $\vec{C} = [C_{ik}]$. We rewrite (1) in vector form as:

$$\vec{C} = \sum_{m=1}^{M} \psi_m\ \vec{\xi}^m, \tag{2}$$

where vectors $\vec{\xi}^m = [\xi_{ik}^m]$ are defined as

$$\xi_{ik}^m = \mu_i\ \eta_{ik}^m \quad , \quad \forall (i,k) \in \mathcal{L}, m = 1, ..., M.$$

Eq. (2) shows that, under any scheduling policy P, the capacity vector $\vec{C}$ can be expressed as a convex combination of M vectors $\vec{\xi}^m$, which are independent of P; the coefficients in this convex combination being equal to the scheduling probabilities ψ_m. We can readily obtain a physical interpretation for the vectors $\vec{\xi}^m$, by considering a scheduling policy P^n in which schedule $\vec{\eta}^n$ is used during all the slots. Under P^n, we have $\psi_m = 1$, for $m = n$, and $\psi_m = 0$, otherwise; this gives rise to the capacity vector $\vec{C} = \vec{\xi}^n$. We conclude that each vectors $\vec{\xi}^m$ in (2) is equal to the capacity vector $\vec{C}$ that results from repeated use of the schedule $\vec{\eta}^m$ during all the slots. Finally, we denoted by $\mathfrak{C}$, the set of capacity vectors $\vec{C}$ that can be obtained under some feasible scheduling policy. The important conclusion that we draw from (2) is that $\mathfrak{C}$ is a convex set.

III. Problem Formulation as Convex Optimization

Let us assign to each link (i, k) of the network a cost h_{ik} as a function of the link capacity C_{ik} and the link flow f_{ik}, with the following properties:

- P1: $h_{ik}(C_{ik}, f_{ik})$ is defined over the region $0 \leq f_{ik} < C_{ik}$, and is continuously differentiable in this region.
- P2: $h_{ik}(C_{ik}, f_{ik})$ is strictly convex with respect to the joint variables (C_{ik}, f_{ik}).
- P3: $h_{ik}(C_{ik}, f_{ik})$ is increasing in f_{ik}, and decreasing in C_{ik}.
- P4: $h_{ik}(C_{ik}, f_{ik}) \to \infty$ as $f_{ik} \to C_{ik}^-$, and for $f_{ik} \geq C_{ik}$, $h_{ik}(C_{ik}, f_{ik}) = \infty$.

Now, consider the optimization problem

$$
\begin{aligned}
&\min_{\vec{C},f} H(\vec{C},f) = \sum_{(i,k)\in\mathcal{L}} h_{ik}(C_{ik}, f_{ik}),\\
&\text{s.t. } f_{ik} = \sum_{j\neq i} f_{ik}^j,\\
&\quad f_{ik}^j \geq 0 \quad , \quad \forall (i,k)\in\mathcal{L}, j\neq i,\\
&\quad r_i^j + \sum_{k:(k,i)\in\mathcal{L}} f_{ki}^j = \sum_{k:(i,k)\in\mathcal{L}} f_{ik}^j, \forall i\in\mathcal{N}, j\neq i,\\
&\quad \vec{C}\in\mathfrak{C}.
\end{aligned} \tag{3}
$$

We make the following observations about this problem. First, it is a convex optimization problem defined over a convex region of variables $\vec{C}$ and f. Second, given property P4, any feasible point $(\vec{C}, f)$ of this optimization problem satisfies $f_{ik} < C_{ik}$. Third, any optimal point $(\vec{C}^*, f^*)$, and indeed any feasible point $(\vec{C}, f)$, corresponds to a set of scheduling probabilities Ψ (see Eq. (2)) and a routing policy for sending the input traffic r to the corresponding destinations. Finally, if the cost functions $h_{ik}(C_{ik}, f_{ik})$ are a close indicator of the queue size or the congestion level at the corresponding links, then any optimal point $(\vec{C}^*, f^*)$ of the problem specifies a joint routing and scheduling policy that minimizes the overall delay or congestion in the network.

The optimization problem (3) can be viewed as a generalization of the minimum delay routing problem [2]. The most natural choice for the cost functions $h_{ik}(C_{ik}, f_{ik})$ would be to set them equal to the average queue size of the corresponding links, $D_{ik}(C_{ik}, f_{ik})$. Unfortunately, D_{ik} is not a convex function of the joint variables (C_{ik}, f_{ik}), although it is convex with respect to C_{ik} or f_{ik}, alone. To circumvent this problem, and also to alleviate the need for estimating derivatives of D_{ik} (needed in an iterative algorithm for solving (3)), we may use a cost function of the form

$$
h_{ik}(C_{ik}, f_{ik}) = C_{ik}\int_0^{\rho_{ik}} D_{ik}(\rho_{ik})d\rho_{ik}, \tag{4}
$$

where $\rho_{ik} = f_{ik}/C_{ik}$, and $D_{ik}(\rho_{ik})$ is the average queue size on link (i,k). This cost function can be shown to possess all the required properties P1-P4. While the theoretical results of this section and Section IV are valid for any choice of cost functions that conform to P1-P4, the simulation results in Section VI are based on the choice in (4).

IV. Formulation for Distributed Implementation

In [7], Gallager develops a distributed algorithm for solving the minimum delay routing problem. The key to his approach is a new set of routing variables that facilitate distributed implementation. Since the structure of our problem, as far as routing variables are concerned, is identical to the minimum delay routing problem, we adopt his approach here. Let t_i^j be the total expected traffic (or node flow) at node i destined for node j. Thus t_i^j includes both r_i^j and the traffic from other nodes that is routed through i for destination j. Finally, define the routing parameter $\varphi_{ik}^j = f_{ik}^j/t_i^j$, and denote by Φ the set of fractions φ_{ik}^j. Clearly, Φ satisfies $\sum_{k:(i,k)\in\mathcal{L}}\varphi_{ik}^j = 1$ and $\varphi_{ik}^j \geq 0$, for $j\neq i$. Moreover we have:

$$
t_i^j = r_i^j + \sum_{k:(k,i)\in\mathcal{L}} t_k^j\ \varphi_{ki}^j \quad , \tag{5}
$$

$$
f_{ik}^j = t_i^j\varphi_{ik}^j. \tag{6}
$$

We also take $\varphi_{ik}^j = 0$, for $i = j$, and assume that for each $i, j, i\neq j$, there is a routing path from i to j, which means there is a sequence of nodes $i, k, l, ..., m, j$, such that $\varphi_{ik}^j > 0, \varphi_{kl}^j > 0, ..., \varphi_{mj}^j > 0$. We take the following theorem from [7]:

Theorem 1: Let a network have input set r and routing parameter set Φ. Then the set of equations (5) has a unique solution for t. Each component t_i^j is nonnegative and continuously differentiable as a function of r and Φ.

Changing the routing variable set from f to Φ, and also utilizing Eq. (2), we are able to restate the optimization problem (3) in terms of the new parameter sets Φ and Ψ, as follows:

$$
\begin{aligned}
&\min_{\Psi,\Phi} H(\Psi,\Phi) = \sum_{(i,k)\in\mathcal{L}} h_{ik}\big(\mu_i\sum_m \psi_m\ \eta_{ik}^m\ , \sum_j t_i^j\ \varphi_{ik}^j\big),\\
&\quad \text{s.t.} \quad t_i^j = r_i^j + \sum_{k:(k,i)\in\mathcal{L}} t_k^j\ \varphi_{ki}^j \qquad \forall i,j\in\mathcal{N}, j\neq i,\\
&\qquad \sum_m \psi_m = 1, \quad \psi_m\geq 0, \quad \forall m,\\
&\qquad \sum_{k:(i,k)\in\mathcal{L}} \varphi_{ik}^j = 1, \quad \varphi_{ik}^j\geq 0, \quad \forall (i,k)\in\mathcal{L}, j\neq i.
\end{aligned} \tag{7}
$$

Optimality Conditions: The objective function in (7) is not convex in terms of the variable set Φ, due to the nonlinear relationship between Φ and f. We have established the following optimality conditions for (7), which are stated here without proof due to lack of space:

Theorem 2: The following Conditions are sufficient for optimality of any feasible point (Φ^*, Ψ^*) of (7):

$$
\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}} + \frac{\partial H(\Psi,\Phi)}{\partial r_k^j} \geq \frac{\partial H(\Psi,\Phi)}{\partial r_i^j} \qquad \forall (i,k)\in\mathcal{L}, i\neq j, \tag{8}
$$

$$
\sum_{(i,k)\in\mathcal{L}} \eta_{ik}^m\ \mu_i\frac{\partial h_{ik}(C_{ik}^*, f_{ik}^*)}{\partial C_{ik}} \geq \gamma_m, \quad m = 1, ..., M, \tag{9}
$$

with equality in (8) and (9), if $\varphi_m^{j*} > 0$ and $\psi_m^* > 0$, respectively. In (8) and (9), C_{ik}^* and f_{ik}^* are the link capacities and flows corresponding to Ψ^* and Φ^*, γ_m, $m = 1, ..., M$, are arbitrary lagrange variables, and $\partial H/\partial r_i^j$ satisfy

$$
\frac{\partial H(\Psi,\Phi)}{\partial r_i^j} = \sum_{k:(i,k)\in\mathcal{L}} \varphi_{ik}^j\left(\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}} + \frac{\partial H(\Psi,\Phi)}{\partial r_k^j}\right). \tag{10}
$$

Theorem 3: For any optimal point $(\vec{C}^*, f^*)$ of problem (3), there exists some feasible Ψ^* and Φ^* such that (8) and (9) hold true.

V. A Minimum Cost Joint Routing and Scheduling (MCRS) Algorithm

In this section, we present a distributed algorithm for optimal routing and scheduling according to problem (7). We shall refer to this algorithm as the minimum cost joint routing and scheduling (MCRS) algorithm. The routing aspect of this algorithm is similar to [7], with some important differences indicated below.

Routing: Each time a packet for destination j arrives at a transition node i, it is added to the queue of the outgoing link (i, k^*), determined as:

$$k^* = \arg \min_{k:\ (i,k)\in\mathcal{L},\ k\notin B_i^j} \partial H(\Psi,\Phi)/\partial\varphi_{ik}^j, \tag{11}$$

where $\frac{\partial H(\Psi,\Phi)}{\partial\varphi_{ik}^j} = t_i^j \left(\frac{\partial h_{ik}(C_{ik},f_{ik})}{\partial f_{ik}} + \frac{\partial H(\Psi,\Phi)}{\partial r_k^j}\right)$, and $B_i^j = \left\{k : (i,k) \in \mathcal{L}, \frac{\partial H(\Psi,\Phi)}{\partial r_i^j} < \frac{\partial H(\Psi,\Phi)}{\partial r_k^j} + \frac{\partial h_{ik}(C_{ik},f_{ik})}{\partial f_{ik}}\right\}$.

The routing parameters Φ are updated continually by the corresponding nodes, based on the packet routing decisions made so far, and using some averaging algorithm such as the exponentially weighted rule. Notice that the algorithm proposed in [7] differs from our approach, since it first updates Φ and then uses Φ to *randomly* choose the outgoing link for each packet.

The role of the blocking sets B_i^j in the algorithm is to prevent the occurrence of routing loops, as discussed in [7]. The marginal costs $\partial H/\partial r_k^j$ are regularly updated by each node k, using (10), and the results are sent to the neighboring nodes [7].

Scheduling: For each time slot, the optimal schedule $\vec{\eta}^{m*}$ is chosen as:

$$m^* = \arg\max_m \sum_{(i,k)\in\mathcal{L}:\ \eta_{ik}^m=1} \mu_i\ w_{ik}, \tag{12}$$

where the link weights w_{ik} are defined as

$$w_{ik} = -\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial C_{ik}}.$$

It turns out that the scheduling parameters Ψ need not be determined explicitly in order to run this algorithm. Obviously, if necessary, Ψ can be estimated from the past scheduling decisions. The optimization in (12) is the well-known maximum weighted independent set (MWIS) problem over the conflict graph of the network.

We have analyzed the above MCRS algorithm and shown that it has desirable convergence properties. Details are beyond the scope of this paper and the available space.

Choice of Cost Functions $h_{ik}(C_{ik}, f_{ik})$: While the methodology, optimality conditions,the algorithm proposed in this paper and its convergence are valid for any choice of the cost functions $h_{ik}(C_{ik}, f_{ik})$ that comply with P1-P4, we have used the cost functions (4) for the simulations discussed in Section VI. With this choise, the marginal cost functions needed to compute (11) and (12) in each step of the algorithm, can be

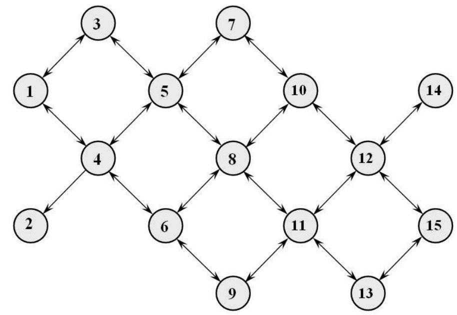


Fig. 1. The wireless network used for simulation with 15 nodes and 39 one-way links.

expressed as:

$$\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}} = D_{ik}(\rho_{ik}), \tag{13}$$

$$\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial C_{ik}} = \int_0^{\rho_{ik}} D_{ik}(\rho_{ik})d\rho_{ik} - \rho_{ik}D_{ik}(\rho_{ik}). \tag{14}$$

It can be shown that, on the RHS of (14), the integral term is strongly dominated by the second term, particularly when ρ_{ik} is close to 1. This allows us to evaluate the integral term by applying rough numerical approximations, without worrying about the overall accuracy of the algorithm.

Tassiulas Algorithm (for comparison): The Tassiulas algorithm [13], when applied to our network model, may be stated as follows. For each time slot t, first the best session j_{ik} to be served on each link (i, k), is determined as:

$$j_{ik} = \arg\max_j\ [Q_i^j - Q_k^j], \tag{15}$$

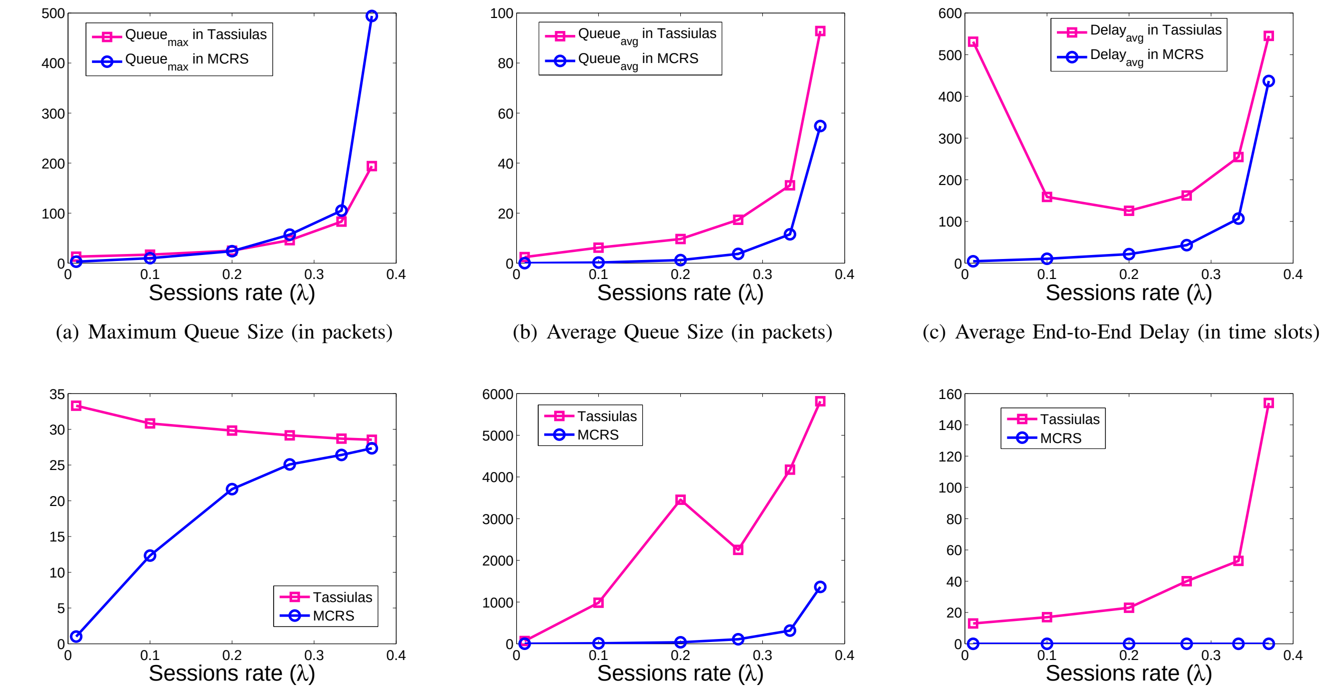
where Q_i^j is the total amount of data stored at i and destined for j, at the beginning of t. Then, the optimal schedule $\vec{\eta}^{m*}$ is determined from the MWIS problem in (12), with the link weights w_{ik} defined as:

$$w_{ik} = Q_i^{j_{ik}} - Q_k^{j_{ik}}. \tag{16}$$

VI. Simulation Results

In this section, we use simulation to study the behavior of the proposed MCRS algorithm as outlined in (11) and (12), and to compare its performance with the throughput-optimal Tassiulas algorithm stated in (15) and (16). We consider a network topology, with 15 wireless nodes and 39 one-way links, as shown in Fig. 1. Packets have equal length and the node transmission rates are assumed to be $\mu_i = 1$ packet/slot for all i, except for the central nodes $i = 4$, $i = 8$, $i = 12$, for which we let $\mu_8 = 4$ packets/slot, and $\mu_4 = \mu_{12} = 2$ packets/slot. To reveal a potential problem in the Tassiulas algorithm, node 2 is assumed to have no outgoing link. This could occur, for example, when a node runs out of sufficient power for transmission.

We consider a one-hop interference model, i.e., we assume that two links (i, k) and (m, n) cannot be activated at the same



(a) Maximum Queue Size (in packets)

(b) Average Queue Size (in packets)

(c) Average End-to-End Delay (in time slots)

(d) Average node duty cycle for transmissions (in percentage)

(e) Maximum packet misordering (in packets)

(f) Total number of blocked packets (in packets)

Fig. 2. Performance comparison between the MCRS and the Tassiulas algorithms. In all plots, the horizontal axes is the average rate of packet arrivals per session, during the session's ON state

time if $i = n$, or $(i, n) \in \mathcal{L}$, or $m = k$, or $(m, k) \in \mathcal{L}$. In simulating the MCRS as well as the Tassiulas algorithm, the MWIS problem (12) arising in each time slot is solved exactly and through exhaustive search.

The following 10 sessions are assumed to be present in the network (with $i \rightarrow j$ referring to a session from the origin node i to the destination node j): $4 \rightarrow 2$, $4 \rightarrow 14$, $4 \rightarrow 11$, $4 \rightarrow 7$, $12 \rightarrow 14$, $12 \rightarrow 1$, $12 \rightarrow 6$, $12 \rightarrow 7$, $8 \rightarrow 15$ and $8 \rightarrow 3$. The activity of each session follows an ON-OFF markov process, with the average ON and OFF periods equal to 100 slots. During the ON state of each session, packets arrive according to a poisson process with mean λ packets/slot, which is the same for all sessions. The simulations are repeated for various values of λ, ranging from $\lambda = 0.01$ to $\lambda = 0.37$. Additional simulation tests (not presented here) indicate that the rate $\lambda = 0.37$ almost lies on the boundary of the capacity region for the Tassiulas algorithm. Each simulation test is run for 40000 time slots.

Figure 2 compares various aspects of the performance of the MCRS and the Tassiulas algorithms. The per-node queue sizes are compared in Figures 2(a) and 2(b). In Figure 2(a), we have determined the maximum queue size attained at any network node during the whole interval of simulation and compared this value for the two algorithms. This figure indicates that, in general, more buffer space may be required under the MCRS algorithm, compared to the Tassiulas algorithm.

In Figure 2(b), we compare the queue sizes once averaged over the time and across all nodes. For both algorithms, the average queue size increases with λ; the increase becomes faster as the network load approaches the capacity region boundary around $\lambda \gtrsim 0.37$. Regardless of traffic load, the average queue size is considerably smaller for the MCRS algorithm.

The average of the end-to-end delays experienced by all packets (from all session) under Tassiulas and MCRS algorithms are depicted in Figure 2(c). This figure clearly illustrates that the MCRS algorithm manages to provide considerably smaller end-to-end delays compared to Tassiulas, regardless of the traffic load. For individul sessions too, the delays under MCRS algorithm are consistently better, although per-session results are not illustrated here. The consistent superiority of the MCRS algorithm with respect to end-to-end delay is due to two factors. First, the average per-node queue sizes are more under Tassiulas (Fig. 2(b)), resulting in larger per-hop packet delays. But, more importantly, the number of hops a packet takes to reach the destination can be much bigger for Tassiulas algorithm, under which packet routing is guided only by per-session queue differentials between adjacent nodes. In contrast, the MCRS algorithm maintains a complete multi-hop view of network paths (via the marginal costs $\partial H / \partial \varphi_{ik}^{j}$), in order to come up with the best routes in the network.

As another observation in Figure 2(c), notice that the delays under Tassiulas do not steadily decrease with the reduction of

λ. Instead, the average delay undergoes a sharp increase as λ approaches zero. This is due to the fact that , under very light load, it takes a long time for the queues to reach proper levels for reasonable routing. Until then, packets aimlessly wander in every possible network direction. Notice that, with each change in the traffic pattern, this process (which may be viewd as the learning process of the algorithm) has to be repeated.

Another striking result is illustrated in Figure 2(d) which shows the average (over al nodes) of the percentage of time slots during which each node transmits. We refer to this metric as the *average node duty cycle* (ANDC). Obviously, energy consumption in the wireless network is directly related to its ANDC. Fig. 2(d) shows that, under the MCRS algorithm, ANDC steadily increases with λ, as we would expect. However, under the Tassiulas algorithm, the ANDC is essentially the same for all values of λ, which is highly undesirable. Clearly, energy consumption is lower for the MCRS algorithm, and the difference with the Tassiulas algorithm is particularly pronounced at low traffic loads.

Figure 2(e) compares the maximum packet misordering in the network, under the two algorithms. This metric specifies the largest difference in the sequence numbers of any pair of subsequently received packets at a common destination. Out-of-order packet delivery occurs due to multi-path routing in both algorithms, and it increases as the traffic load goes up. However, as Fig. 2(e) illustrates, it is far more severe under Tassiulas algorithm. The wider range of packet misordering under Tassiulas is also indicative of larger delay jitters resulting from this algorithm. This observation is indeed confirmed by additional simulation results which are not included here.

Finally, Figure 2(f) confirms that under the Tassiulas algorithm some packets will be blocked at node 2 which has no outgoing link, and will never reach their destinations. This undesirable outcome can be explained as follows: The Tassiulas algorithm uses information inherent in the queue build-ups across the network, in order to come up with proper routes to the destinations. Before the per-session queue sizes at node 2 are increased beyond the corresponding queue sizes at the neighboring nodes, the algorithm has no way of stopping the traffic from going to node 2. Since the packets accumulated at node 2 up to this point have no outlet, they will remain there forever and will be accounted as lost packets by the corresponding destinations. Notice that if node 2 discards these blocked packets after a while, new packets will be sent to this node, until proper queue levels are again reached. In the MCRS algorithm, such undesirable behavior is absent, as confirmed by Figure 2(f), because the routes are determined based on a more complete view of the network.

VII. Conclusion

In this paper, we came up with a generalization of the minimum delay routing problem, in order to formulate the network scheduling and routing problems within a unified framework. We then converted the problem to a form amenable to distributed implementation. Optimality conditions for the problem were discussed and a joint routing and scheduling algorithm for the solution was proposed. Using simulation on a network with 15 nodes and 39 links, we compared the performance of the algorithm with Tassiulas algorithm, which is known to be throughput-optimal, and which represents a few other similarly formulated algorithms. While the simulation results do not reflect any noticeable difference in the throughput accommodated by the two algorithms, our algorithm is shown to provide considerable advantage in terms of packet delay and jitter, packet loss and misordering, and energy consumption.

References

[1] M. Alicherry, R. Bhatia and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," In *Proceedings of ACM Mobicom*, pp. 58-72, 2005.

[2] D. Bertsekas, and R. G. Gallager, *Data Networks*, Prentice-Hall, NJ 1987.

[3] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling Distributed Throughput Maximization in Wireless Mesh Networks A Partitioning Approach," In *Proceedings of ACM MobiCom*, 2006.

[4] A. Brzezinski, G. Zussman, and E. Modiano, "Local Pooling Conditions for Joint Routing and Scheduling," *Information Theory and Applications Workshop*, 2008.

[5] L. Chen, S. H. Low, M. Chiang and J. C. Doyle, "Optimal Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," In *proceeding of IEEE INFOCOM*, 2006.

[6] R. L. Cruz and A. V. Santhanam, "Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks," In *proceeding of IEEE INFOCOM*, 2003.

[7] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," *IEEE Transactions on Communications*, Vol. com-25, No. 1, January 1977.

[8] S. J. Golestani, R. Rastogi and M. Smith, "Dynamic Channel Assignment in wireless LAN," 2007. *Communication Systems Software and Middleware*, 2007.

[9] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," In *Proceedings of ACM MobiCom*, 2005.

[10] S. Merlin, N. Vaidya, M. Zorzi, "Resource Allocation in Multi-Radio Multi-Channel Multi-Hop Wireless Networks," In *proceeding of IEEE INFOCOM*, 2008.

[11] M. J. Neely, E. Modiano and C. E. Rohrs, "Dynamic Power Allocation and Routing for Time-Varying Wireless Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 1, January 2005.

[12] G. Sharma, N. B. Shroff and R. R. Mazumdar, "Joint Congestion Control and Distributed Scheduling for Throughput Guarantees in Wireless Networks", In *proceedings of IEEE INFOCOM*, 2007.

[13] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and Scheduling policies for maximum throughput in multi-hop radio networks," *IEEE Trans. Autom. Control*, Vol. 37, No. 12, Dec. 1992.