To be Presented at ICNP '09

A Unified Theory of Scheduling, Routing, and Flow Control in Wireless Networks

Naghmeh S. Moayedian
Electrical and Computer Eng. Dept.
Isfahan University of Technology
Isfahan, Iran
Email: moayedian@ieee.org

S. Jamaloddin Golestani
Electrical and Computer Eng. Dept.
Isfahan University of Technology
Isfahan, Iran
Email: golestani@ieee.org

***Abstract*—A new approach to joint scheduling, flow control and routing in wireless networks, based on formulation as a convex optimization problem is presented. This approach is novel in that it integrates optimal scheduling and flow control with a modified version of minimum delay routing, resulting in significant performance advantage over alternative approaches. We come up with a distributed algorithm for implementation of the proposed scheme and show, by analysis and simulation, that the algorithm achieves fairness and/or priorities among users in accordance with pre-assigned user parameters. In sharp contrast to alternative algorithms that perform scheduling and packet routing based on per-session queue differential between adjacent nodes, our algorithm uses a complete multi-hop view of network conditions for packet routing and retains the desirable properties of minimum delay routing. As the result, it achieves queue sizes and end-to-end delays that are several times smaller, without compromising the throughput.**

I. Introduction

In this paper, we propose and study a joint scheduling, routing and flow control algorithm for multi-hop wireless networks, based on a single convex optimization problem. The novel aspect of our formulation is that it integrates optimal scheduling and flow control with a modified version of minimum delay routing, resulting in considerable performance advantage over the alternative approaches discussed in [3], [4], [5], [6].

The problems of routing, flow control, scheduling, and power control all deal, in one form or another, with the sharing of common resources among different users, in a setting where both the resources and the users are distributed. Therefore, these problems are fundamentally interrelated and any scheme used to address one of them, will impact the others. There have been many research efforts in the past, trying to address these problems within a unified theoretical framework and/or to provide coherent and well-coordinated solutions to them. These research efforts fall in two categories. In one category, each problem is tackled, more or less separately, but efforts are made to integrate the resulting algorithms, in a coherent manner [7], [8], [9]. The second category of research, to which the present paper belongs, are those that try at the outset to capture multiple problems in one theoretical formulation, thereby exploiting their relationship and finding an integrated solution to them, in a natural way.

An early example of this latter approach is *the unified theory of flow control and routing*, developed in the context of wireline networks, nearly 30 years ago [16], [18]. Another prime example is the optimal scheduling developed by Tassiulas et al. in [3] for wireless networks witch, in effect, accomplishes both routing and scheduling. Neely et al. built upon this work to come up with a theory for power control, scheduling and routing [4]. The capacity regions provided by the algorithms in [3] and [4] are proved to be optimal, in the sense that no other algorithm can provide service to a set of input demands, if these algorithms fail to do so. Another approach is that of Chen et al [5], which is closely related to [3] and [4] in terms of the end result, but not the underlying theory. However, the algorithms resulting from these alternative formulations are surprisingly similar, as far as routing and scheduling are concerned. Another related work is [6] which extends the work of Chen [5] to a multichannel scenario.

Unfortunately, the routing aspect of the algorithms in [3], [4], [5], and [6] is not desirable since they do not use any multi-hop view of the network map and traffic conditions. Instead, at each hop, packets are forwarded based on per-session queue differentials between the current node and the adjacent nodes. Until these queues are built up to appropriate levels that enable sound routing, packets will wander in the network in every possible direction. The result, as shown in [14], is long routing paths, frequent out of order packets, large delays, and high energy consumption, even when the network is lightly loaded. Interestingly, these undesirable features exist, even though the algorithms in [3], and [4] are *throughput optimal*, as noted above. The reason behind this apparent inconsistency is that, under these algorithms, it takes a long time until queues are built up to the high levels that can lead to proper queue differentials needed for effective routing.

In this paper, we circumvent the above problems by taking a different approach to the joint formulation of scheduling, routing and flow control. We start with the optimization formulation for joint routing and flow control in [16] and [18], and find a way of incorporating scheduling into this framework that retains its desired minimum cost routing property. We discuss the optimality conditions, establish properties with respect to the provision of fairness and priorities, and develop an algorithm for the solution which is distributed as far as its routing and flow control is concerned. The scheduling aspect of the algorithm boils down to solving a variation of the maximum weighted independent set(MWIS) problem on the conflict graph of the network, as is the case in all the above-mentioned algorithms. We do not attempt to find approximate solutions to this problem, which has been extensively studies elsewhere [10], [11], [12], [13]. Numerical and simulation studies on two networks with 22 and 39 links show that our algorithm, when compared to [5], essentially achieves the same throughput, is somewhat better in terms of fairness/priority properties, and is far superior with respect to queue sizes and end-to-end delays.

The rest of the paper is organized as follows. In Sections II, the system model is presented. Formulation of the problem as convex optimization and its basic properties are discussed in Section III. In Section IV, the problem is reformulated to facilitate distributed implementation, and the optimality conditions are discussed. A distributed algorithm for the solution is presented in Section V, while Section VI deals with the numerical study and simulation results. 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 one-way transmission links. We denote a link going from node $i \in \mathcal{N}$ to node $k \in \mathcal{N}$ by (i,k). Let $\mathcal{S}$ denote the set of all sessions in the network, and $\mathcal{S}_i^j,\ i \neq j$ stand for the set of sessions originating at node i, and destined for node j. Denote by $r_s,\ s \in \mathcal{S}$, the average rate of data from session s that enters the network.

Let f_{ik} denote 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. Thus,

$$f_{ik} = \sum_{j \neq i} f_{ik}^j. \tag{1}$$

Clearly, at any node i and with respect to any destination j, the following equality must hold:

$$\sum_{v:(v,i)\in\mathcal{L}} f_{vi}^j + \sum_{s \in \mathcal{S}_i^j} r_s = \sum_{k:(i,k)\in\mathcal{L}} f_{ik}^j\ ,\ i,j \in \mathcal{N},\ j \neq i. \tag{2}$$

Assume that there are B transmission channels, denoted by $b = 1, ..., B$, shared by the network links. We use a conflict graph to indicate the presence of harmful interference between two links when they use the same channel, simultaneously. Let $\mu_{ik}(b)$ denote the transmission rate afforded to link (i,k), when it uses channel b in the absence of harmful interference. Our analysis here assumes a static network topology where the set of links and the set of link transmission rates, $\mu_{ik}(b)$, do not change with time. From a practical viewpoint, this assumption is satisfied as long as topology changes are slow compared to the convergence time of the algorithms discussed here. We allow for multiple radios per node, with R_i denoting the number of radios of node i. Each link (i,k), however, may only use one radio (i.e., one channel) at a time. Extension of results to a more general scenario is readily possible.

We divide time into equal slots, and during each slot use a fixed assignment of channels to the links, across the network. We use the term *schedule* to refer to a given set of channel assignments to the links. More specifically, a schedule m corresponds to a vector $\vec{\eta}^m = [\eta_{ik}^m]$, with the entries η_{ik}^m, $(i,k) \in \mathcal{L}$, defined as:

$$\eta_{ik}^m \triangleq \begin{cases} 0 & \text{no channel is assigned to } (i,k), \\ b & \text{channel } b, b = 1, ...,\ B, \text{ is assigned to } (i,k). \end{cases} \tag{3}$$

A link (i,k) for which $\eta_{ik}^m \neq 0$, will be referred to as *active*, according to schedule m. Define $\mathcal{M}$ to be the set of schedules m that are feasible, i.e., satisfy the following restrictions:

i) At each node i, the number of outgoing and incoming links that are active, does not exceed R_i.
ii) Any pair of links (i,k) and (v,w) for which $\eta_{ik}^m = \eta_{vw}^m \neq 0$, are noninterfering, i.e., the corresponding vertices in the conflict graph are not connected.

Clearly, $|\mathcal{M}|$ is finite, although it can be very large. Given the conflict graph, B, and R_i's, the set $\mathcal{M}$ can be constructed, in principle. Adopting a probabilistic view with respect to channel scheduling, let ψ_m denote the probability that schedule m is used during an arbitrary time slot. Obviously, $\sum_{m\in\mathcal{M}} \psi_m = 1$. Define the *scheduling probability vector* $\Psi = [\psi_m]$, to be composed of one component ψ_m per schedule $m \in \mathcal{M}$.

Finally, for each link (i,k), we define the capacity C_{ik} as the expected rate of service afforded to link (i,k), given a scheduling probability vector Ψ. C_{ik} may be expressed as,

$$C_{ik} = \sum_{m\in\mathcal{M}} \psi_m\ \mu_{ik}^m, \tag{4}$$

where we use μ_{ik}^m to denote the transmission rate of link (i,k) under schedule m, i.e., $\mu_{ik}^m \triangleq \mu_{ik}(\eta_{ik}^m)$. For consistency, we let $\mu_{ik}^m(0) \triangleq 0$. Notice that, in comparison to the capacity of a wired link, here the transmission rate C_{ik} is not provided to link (i,k) on a steady basis; the actual rate of transmission fluctuates, with its average C_{ik} depending on the scheduling probabilities ψ_m.

Eq. (4) may be expressed in vector form as:

$$\vec{C} = \sum_{m\in\mathcal{M}} \psi_m\ \vec{\mu}^m, \tag{5}$$

where the $|\mathcal{L}|$-dimensional vectors $\vec{C} \triangleq [C_{ik}]$, and $\vec{\mu}^m \triangleq [\mu_{ik}^m],\ m \in \mathcal{M}$, are comprised of one component per link (i,k). We see from Eq. (5) that, under any ergodic scheduling policy $\mathcal{P}$, the capacity vector $\vec{C}$ can be expressed as a convex combination of $|\mathcal{M}|$ vectors $\vec{\mu}^m$, with the scheduling probabilities ψ_m serving as the corresponding coefficients. Notice that the vectors $\vec{\mu}^m$ themselves are independent from the scheduling policy $\mathcal{P}$; each $\vec{\mu}^m$ is the capacity vector under the scheduling policy that applies schedule m all the time, i.e., with probability 1. Let us denote by $\mathfrak{C}$, the set of capacity vectors $\vec{C}$ that can be obtained under some feasible scheduling policy. It follows from (5) that $\mathfrak{C}$ is a convex set.

III. Problem Formulation

The joint formulation of scheduling, flow control and routing proposed here is closely related to the *unified theory of flow control and routing* developed in late 1970's [16], [18]. We first provide a brief overview of that work.

A. Joint Routing and Flow Control

We temporarily remove the scheduling aspect from the network model of section II, by assuming that all links (i,k) are permanently active and have a fixed service rate $C_{ik},\ (i,k) \in \mathcal{L}$. This would be the case, for example, if we deal with a wireline network, or if there are enough channels in the wireless network to allow permanent activation of all links by using fixed channels, without facing undue interference. Joint flow control and routing in such a network can be practiced by solving the following convex optimization problem, across the network:

$$\min_{r,f}\ J(r,f) = \sum_{s\in\mathcal{S}} e_s(r_s) + \sum_{(i,k)\in\mathcal{L}} h_{ik}(f_{ik}), \tag{6}$$

$$\text{subject to } r_s \geq 0, \tag{6a}$$

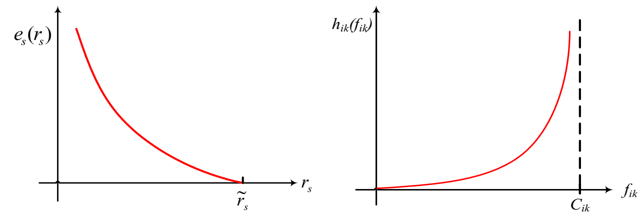
$$f^j_{ik} \geq 0, \tag{6b}$$

and constraints (1) and (2). Here $e_s(r_s)$ is a decreasing convex cost function assigned to each session s, to reflect user dissatisfaction when a rate r_s smaller than some desired rate $\tilde{r}_s$ is assigned to the user (Fig. 1(a)). We refer to $e_s(r_s)$ as the cost function of session s. Similarly, $h_{ik}(f_{ik})$ is an increasing convex cost function assigned to each link $(i,k) \in \mathcal{L}$, to account for the cost of congestion on the link. As the link flow, f_{ik}, approaches the link capacity, the average number of packets waiting on the link increases, and so does the cost $h_{ik}(f_{ik})$ (Fig. 1(b)). A natural choice for the link cost functions is the average queue size on each link, i.e., to set $h_{ik}(f_{ik}) = D_{ik}(f_{ik})$, where D_{ik} denotes the average number of packets waiting to be sent on (i,k). A more elaborate choice is to set

$$h_{ik}(f_{ik}) = g(D_{ik}(f_{ik})), \tag{7}$$

where $g(D)$ is a convex increasing function of D, that grows rapidly as D approaches some critical threshold D_{max}. For more details, see [16].

The optimization in (6) is performed in terms of flow control control variables r_s, collectively refered to as r, and per-destination link flows f^j_{ik}, that specify the routing and are collectively refered to as f. It is intuitively clear that at a solution point (r^*, f^*) of (6), the optimal input rates r^* strike a balance between the cost of congestion at the links and the cost of restricting user rates. The optimal link flows f^* specify a routing that distributes the input traffic (given by r^*) among various network paths so as to minimize the total network congestion.



(a) Cost of limiting the rate of session s to r_s. (b) Congestion cost of the flow f_{ik} on link (i,k).

Fig. 1. Cost functions used in optimization for joint flow control and routing.

From Figures 1(a) and 1(b), it might visually appear that the optimization (6) does not lead to complete or near-complete utilization of link capacities, even where a link is in demand. This impression is not necessarily valid. As will be demonstrated later, with proper scaling of the link and user cost functions, the optimal point (r^*, f^*) of (6) will result in full utilization of network resources. Still, one might argue that it is preferable to use the following optimization problem, instead of (6):

$$\min\ J(r,f) = \sum_{s\in\mathcal{S}} e_s(r_s), \tag{8}$$

subject to the previous constraints and the new constraint

$$f_{ik} \leq C_{ik}. \tag{8a}$$

Here, instead of using link cost functions, one avoids congestion by incorporating constraint (8a) that allows utilization up to the full capacity of each link. Indeed, (8) is an alternative way of formulating network flow control as convex optimization. This approach was proposed in late 1990's [17].

In comparing (6) and (8), it is possible to interpret (6) as an approximation to solving (8), using the well-known barrier method, with $h_{ik}(f_{ik}),\ (i,k) \in \mathcal{L}$, serving as barrier functions. However, this is not a comprehensive characterization of problem (6), since the capacity constraint (8a) is not the only type of constraint that one faces in a network with random traffic. We also need to maintain manageable queue sizes, and this requirement is best captured and most directly expressed by problem (6), using link cost functions defined in (7).

The flow control problem (8) was later adopted in [5] as the basis for developing a joint scheduling, routing and flow control algorithm. In this paper, we also come up with a unified formulation of scheduling, routing and flow control; however, our formulation is based on (6) and uses an entirely different approach. The simulation and numerical results presented in Section VI confirm that the throughputs afforded by these two algorithms are virtually identical. However, the queue sizes and the end-to-end delays resulting from our approach are far superior to that of [5]. The reason behind this performance difference is that we incorporate optimal scheduling into (6) in a manner that preserves its minimum cost routing property, thus leading to a routing policy that guides packets through the network along the least congested paths. In [5], incorporation of scheduling and routing into (8) is done differently, and leads to a routing policy that, like algorithms in [3], [4], and [6], is based on a view depth of only one hop, regarding network traffic conditions.

B. Joint Scheduling, Routing and Flow Control

When scheduling is incorporated in the network model, the link capacities (i.e., average service rates) C_{ik} are adjustable variables, and not given constants. Furthermore, the build up of congestion on a link (i,k) depends on both the link flow f_{ik} and the capacity C_{ik}. We are thus led to extend the optimization problem (6) as follows, in order to account for scheduling:

$$\min_{r,f,\vec{C}} \; J(r,f,\vec{C}) = \sum_{s\in\mathcal{S}} e_s(r_s) + \sum_{(i,k)\in\mathcal{L}} h_{ik}(C_{ik}, f_{ik}), \tag{9}$$

$$\text{s. t.} \; \sum_{(v,i)\in\mathcal{L}} f^j_{vi} + \sum_{s\in\mathcal{S}^j_i} r_s = \sum_{(i,k)\in\mathcal{L}} f^j_{ik}\;,\; i,j\in\mathcal{N},\; j\neq i, \tag{9a}$$

$$f_{ik} = \sum_{j\neq i} f^j_{ik}, \tag{9b}$$

$$0 \le r_s \le \tilde{r}_s, \tag{9c}$$

$$f^j_{ik} \ge 0, \tag{9d}$$

$$\vec{C} \in \mathfrak{C}. \tag{9e}$$

Here, the optimization variables include r and f, as well as the capacity vector $\vec{C}$. Notice that any feasible point $(r,f,\vec{C})$ of (9) corresponds to a set of admissible session rates r_s, a routing policy in accordance with link flows f^j_{ik}, and a scheduling policy with scheduling probabilities ψ_m derived from (5). Again, for the problem to be meaningful, the link cost functions $h_{ik}(C_{ik}, f_{ik})$ should provide a good indication of congestion on link (i,k). Since $\mathfrak{C}$ is a convex set, convexity of the problem is preserved, provided that h_{ik} is convex with respect to the joint variables (f_{ik}, C_{ik}). More specifically, assume that the session and link cost functions in (9) satisfy the following:

- A1: $e_s(r_s),\; s\in\mathcal{S}$, is a twice continuously differentiable, decreasing and strictly convex function of r_s over the region $0 < r_s \le \tilde{r}_s$.
- A2: $h_{ik}(C_{ik}, f_{ik}),\; (i,k)\in\mathcal{L}$, is a twice continuously differentiable and strictly convex function of the joint variables (C_{ik}, f_{ik}) over the region $0 \le f_{ik} < C_{ik}$.
- A3: $h_{ik}(C_{ik}, f_{ik})$ is increasing in f_{ik}, and decreasing in C_{ik}, for $0 \le f_{ik} < C_{ik}$.
- A4: $h_{ik}(C_{ik}, f_{ik}) \to \infty$ as $f_{ik} \to C^-_{ik}$, and $h_{ik}(C_{ik}, f_{ik}) = \infty$ for $f_{ik} \ge C_{ik}$.

Furthermore, we make the following natural assumption with respect to service rates $\mu_{ik}(b)$:

- A5: $\mu_{ik}(b) > 0,\; (i,k)\in\mathcal{L},\; b = 1, ..., B.$

Corollary 1: Given A1-A5, the feasibility region of (9) is nonempty. Furthermore, at any feasible point $(r,f,\vec{c})$, we have $f_{ik} \le C_{ik}$, $\forall (i,k)\in\mathcal{L}$.

This corollary which is easy to prove, verifies that, at any feasible point of (6), the average link flows are kept below the link capacities.

Theorem 1: Define the length of each link $(i,k)\in\mathcal{L}$ as it's marginal cost $\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}}$, and denote the shortest distance from i to j, for any node pair (i,j), by λ^j_i. At any optimal point $(r^*, f^*, \vec{C}^*)$ of (9), the following are satisfied:

i: For any session $s\in\mathcal{S}^j_i$,

$$-e'_s(r^*_s) \begin{cases} \ge {\lambda^j_i}^* & r^*_s = \tilde{r}_s \\ = {\lambda^j_i}^* & 0 < r^*_s < \tilde{r}_s \\ \le {\lambda^j_i}^* & r^*_s = 0 \end{cases} \tag{10}$$

where ${\lambda^j_i}^*$ denote the value of λ^j_i at the optimal point.

ii: Any active path between i and j is a shortest path. In other words, if there is a path π from i to j such that for some node l and for all links (v,w) that lie on π, $f^{l*}_{vw} > 0$, then π must be a shortest path between i and j.

The above theorem can be proved by using (5) to replace variables $\vec{C}$ with Ψ, then applying Karush-Kuhn-Tucker Theorem [15] to to the problem. Details are omitted due to lack of space.

${\lambda^j_i}^*$ is a measure of the incremental cost of congestion for any traffic going from i to j. The above theorem states that, at the optimal point, the traffic is only moved along paths that pose the smallest incremental cost of congestion. Furthermore, each optimal rate r^*_s is chosen such that the incremental reward to session s and the incremental cost of congestion to the network are equal—see(10). Motivated by this result, we introduce the following definitions, the significant of which will be shortly explained:

Definition 1: The priority function of each session $s\in\mathcal{S}$, is minus derivative of its cost function, i.e., $-e'_s(r_s)$.

Definition 2: The congestion measure of each session $s\in\mathcal{S}^j_i$, denoted as γ_s, is equal to the shortest distance of i and j at the optimal point of (9), i.e.,

$$\gamma_s = {\lambda^j_i}^* \tag{11}$$

C. Multipath Routing with Shortest Distance

The second property stated in Theorem 1, summarizes the important routing features of the proposed scheme that sets it apart from the alternative schemes proposed in [3], [4], [5], and [6]. According to this property, at the optimal point of (9), all traffic is sent to the destination over the least congested paths, i.e., paths with the shortest distance. This does not imply, however, that all traffic going from an origin node i to a destination node j moves along a single path. In fact, the opposite is typically true since the act of load balancing that is inherent to optimization (9) often gives rise to multiple equidistance paths between different origin-destination pairs.

D. Provision of Fairness and Priorities

Consider the following class of priority functions:

$$-e'(r) = \big(\frac{\alpha}{r}\big)^{\vartheta}. \tag{12}$$

In view of (11), the following observations can be made:

1- Let two sessions s and u have priority functions of form (12) with coefficients α_s, α_u, and $\vartheta_s = \vartheta_u$. Furthermore, assume that $\gamma_s = \gamma_u$. This could be the case, for example, when s and u share the same origin and destination. It follows from (10), (11) and (12) that

$$\frac{r_s^*}{r_u^*} = \frac{\alpha_s}{\alpha_u}, \tag{13}$$

meaning that, under similar conditions expressed by the congestion measure, rates are assigned in proportion to the coefficient α.

2- In the above scenario, let $\gamma_s = \kappa.\gamma_u$, and $\alpha_s = \alpha_u$. This could be the case, for example, if the number of hops included in the routing path of s is κ times the routing path of u, and all hops have the same distance (i.e., experience the same congestion). In this case, it follows that

$$\frac{r_s^*}{r_u^*} = \frac{1}{\sqrt[\vartheta]{\kappa}}. \tag{14}$$

For $\vartheta = 1$, r_s^* is less than r_u^* by a factor of κ. As ϑ increases, penalization of session s due to its higher congestion measure (e.g., its longer path) reduces. In the limit, when ϑ becomes very large, r_s^* and r_u^* are almost equal. This limiting case, is actually equivalent to using a maximin criterion [1] in assigning rates to sessions.

3- Finally, for a single session s, we can conclude from (11) and (12) that

$$\frac{dr_s^*}{r_s^*} = -\frac{1}{\vartheta_s}\frac{d\gamma_s}{\gamma_s}. \tag{15}$$

This indicates that, as the congestion measure γ_s increases, the *relative* decrease in r_s^* equals the *relative* increase in γ_s, times $\frac{1}{\vartheta_s}$. For example, for $\vartheta_s = 2$, a 10% increase in γ_s results in a 5% decrease in r_s^*. In other words, as ϑ_s becomes larger, the sensitivity of the allocated rate to the congestion measure goes down.

In conclusion, the above observations indicate that the rates assigned to sessions at the optimal point of (9) are not arbitrary and are based on predefined rules of competition. Specifically, observation 1 amounts to a fairness property, whereas observations 2 and 3 suggest a mechanism for giving priorities to sessions.

E. Choice of Link Cost Functions

The optimization problem (9) 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 (9)), we may use a cost function of the form

$$h_{ik}(C_{ik}, f_{ik}) = C_{ik}^{\nu}\int_0^{\rho_{ik}} D_{ik}(\rho_{ik})d\rho_{ik}, \tag{16}$$

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 the required properties A2-A4, for $\nu > 1$. While the theoretical results of this section and Section IV are valid for any choice of link cost functions that conform to A2-A4, the simulation results in Section VI are based on the choice in (16).

IV. Formulation for Distributed Implementation

In [2], Gallager develops a distributed algorithm for solving the minimum delay routing problem. In [16], distributed algorithm for joint flow control and routing based on (6) are developed. The key to these approaches 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 the same approach here. Let r_i^j denote the total rate of traffic entering the network at node i and destined for j. Clearly,

$$r_i^j = \sum_{s\in\mathcal{S}_i^j} r_s. \tag{17}$$

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 $\varphi_{ik}^j \triangleq f_{ik}^j/t_i^j$. Denote by Φ and t the set of fractions φ_{ik}^j, and the set of node flows t_i^j, respectively. 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:

$$t_i^j = r_i^j + \sum_{k:(k,i)\in\mathcal{L}} t_k^j\ \varphi_{ki}^j \quad , \tag{18}$$

$$f_{ik}^j = t_i^j \varphi_{ik}^j. \tag{19}$$

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 [2]:

Theorem 2: Let a network have the input set r and the routing parameter set Φ. Then the set of equations (18) and (19) have a unique solution for t and f. Each component t_i^j or f_{ik}^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. (5), we are able to restate the optimization problem (9) in terms of the new parameter sets Φ and Ψ, as follows:

$$\min_{r,\Phi,\Psi} \; J(r,\Phi,\Psi) = E(r) + H(r,\Phi,\Psi), \tag{20}$$

$$\text{s.t.} \quad \psi_m \geq 0, \qquad m \in \mathcal{M}, \tag{20a}$$

$$\sum_{m\in\mathcal{M}} \psi_m = 1, \tag{20b}$$

$$\varphi_{ik}^j \geq 0, \qquad (i,k) \in \mathcal{L}, j \neq i, i, j \in \mathcal{N}, \tag{20c}$$

$$\sum_{k:(i,k)\in\mathcal{L}} \varphi_{ik}^j = 1, \qquad j \neq i, i, j \in \mathcal{N}, \tag{20d}$$

$$0 \leq r_s \leq \tilde{r}_s, \qquad s \in \mathcal{S}, \tag{20e}$$

where

$$E(r) = \sum_{s\in\mathcal{S}} e_s(r_s), \tag{21}$$

$$H(r,\Phi,\Psi) = \sum_{(i,k)\in\mathcal{L}} h_{ik}\big(C_{ik} \;, f_{ik}\big). \tag{22}$$

C_{ik} is given in (4) in term of Ψ. f_{ik} is given in (1), where per-session link flows f_{ik}^j are implicitly given by (18) and (19) in terms of r and Φ, as established in Theorem 2.

A. Optimality Conditions

The objective function in (20) is not convex in terms of the variable set (r,Φ,Ψ), due to the nonlinear relationship between Φ and f. We have established the following optimality conditions for (20), which are stated here without proof due to lack of space:

Theorem 3: Sufficient conditions for optimality of a feasible point (r^*,Φ^*,Ψ^*) of (20) is that there exist numbers ζ_m, $m \in \mathcal{M}$, such that:

$$\frac{\partial H(r^*,\Phi^*,\Psi^*)}{\partial r_i^j} \begin{cases} = -e_s'\big(r_s^*\big) & 0 < r_s^* < \tilde{r}_s \;, \\ \geq -e_s'\big(r_s^*\big) & r_s^* = 0 \;, \\ \leq -e_s'\big(r_s^*\big) & r_s^* = \tilde{r}_s \;, \end{cases} \tag{23}$$
$$i,j \in \mathcal{N}, i \neq j, s \in \mathcal{S}_i^j \;,$$

$$\frac{\partial h_{ik}(C_{ik}^*, f_{ik}^*)}{\partial f_{ik}} + \frac{\partial H(r^*,\Phi^*,\Psi^*)}{\partial r_k^j} \geq \frac{\partial H(r^*,\Phi^*,\Psi^*)}{\partial r_i^j},$$
$$(i,k) \in \mathcal{L}, i,j \in \mathcal{N}, i \neq j, \tag{24}$$

$$\sum_{(i,k)\in\mathcal{L}} - \mu_{ik}^m \, \frac{\partial h_{ik}(C_{ik}^*, f_{ik}^*)}{\partial C_{ik}} \leq \zeta, \quad m \in \mathcal{M}, \tag{25}$$

with equality in (24) and (25), if $\varphi_{ik}^{j*} > 0$, and $\psi_m^* > 0$, respectively. In (24) and (25), C_{ik}^* and f_{ik}^* are the link capacities and flows corresponding to Ψ^*, r^* and Φ^*, and $\partial H/\partial r_i^j$ satisfy the set of equations

$$\frac{\partial H(r,\Phi,\Psi)}{\partial r_i^j} = \sum_{k:(i,k)\in\mathcal{L}} \varphi_{ik}^j \left(\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}} + \frac{\partial H(r,\Phi,\Psi)}{\partial r_k^j} \right)$$
$$i,j \in \mathcal{N}, i \neq j. \tag{26}$$

Theorem 4: For any optimal point $(r^*, f^*, \vec{C}^*)$ of problem (9), there exists some feasible Ψ^* and Φ^* such that the optimality conditions (24) and (25) hold true for (r^*,Φ^*,Ψ^*).

V. A Minimum Cost Joint Routing, Flow Control, and Scheduling (MCRFS) Algorithm

In this section, we present a distributed algorithm for solving (20), to which we shall refer as the minimum cost joint routing, flow control, and scheduling (MCRFS) algorithm. In this algorithm, the routing, scheduling and flow control parameters are updated as follows:

A. Routing

Each time a packet for destination j arrives at a node i, it is added to the queue of the outgoing link (i,k^*), that provides the shortest path to the destination. In other words, the outgoing link (i,k^*) is determined as:

$$k^* \in \arg \min_{k:\; (i,k)\in\mathcal{L},\; k\notin B_i^j} \partial H(r,\Psi,\Phi)/\partial\varphi_{ik}^j, \tag{27}$$

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

The routing parameters Φ are updated continually by the corresponding nodes, based on the packet routing decisions made so far, and using the exponentially weighted averaging rule. Then, φ_{ik}^j, $k \in B_i^j$, are set to zero and the remaining φ_{ik}^j, $k \notin B_i^j$, are normalized, so that they add up to one. It can be shown that the above routing decision, in effect, corresponds to slightly increasing $\varphi_{ik^*}^j$, and making up for this increase by proportionately decreasing φ_{ik}^j, for other outgoing links (i,k). Notice that the routing algorithms proposed in [16] and [2] differ from the approach here, in that they first update Φ and then use 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 [2]. The marginal costs $\partial H/\partial r_k^j$ are regularly updated by each node k, using (26), and the results are sent to the neighboring nodes [2].

B. Flow control

For each session s, the rate is periodically updated based on the difference between the sessions's priority function and its incremental cost of congestion. More precisely, the new rate $\hat{r}_s$ is determined as

$$\hat{r}_s = \begin{cases} r_s - \alpha \; \delta_s & 0 \leq r_s - \alpha \; \delta_s \leq \tilde{r}_s \\ 0 & r_i^j - \alpha \; \delta_s \leq 0 \\ \tilde{r}_s & r_s - \alpha \; \delta_s \geq \tilde{r}_s \end{cases} \tag{28}$$

where r_s is the current rate of s, α is a positive scale factor, and

$$\delta_s = \frac{\partial J(r,\Psi,\Phi)}{\partial r_s} = e_s^{'}(r_s) + \frac{\partial H(r,\Psi,\Phi)}{\partial r_i^j}. \tag{29}$$

Here, i and j are the origin and destination of session s.

C. Scheduling

For each time slot, the optimal schedule $m*$ is chosen as:

$$m^* = \arg \max_{m \in \mathcal{M}} \sum_{(i,k) \in \mathcal{L}} \mu_{ik}^m \, w_{ik}, \qquad (30)$$

where the link weights w_{ik} are defined as

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

For the case of $B = 1$ channel, the optimization in (30) boils down to the well-known maximum weighted independent set (MWIS) problem over the conflict graph of the network. For $B > 1$, (30) defines a more general problem on the conflict graph, to which we may refer as the *multichannel* MWIS problem.

It turns out that the scheduling parameters Ψ need not be determined explicitly in order to run this algorithm. However, as we mentioned in connection with routing updates, the above scheduling decision amounts to slightly increasing ψ_{m^*}, and making up for this increase by proportionately decreasing other scheduling probabilities ψ_m.

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

D. Choice of Cost Functions $h_{ik}(C_{ik}, f_{ik})$

While the methodology, optimality conditions, the algorithm proposed in this paper and its convergence properties are valid for any choice of the cost functions $h_{ik}(C_{ik}, f_{ik})$ that comply with A2-A4, we have used the cost functions (16) with $\nu = 1$ for the simulations discussed in Section VI. With this choice, the marginal cost functions needed to compute (27) - (30) in each step of the algorithm, can be expressed as:

$$\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial f_{ik}} = D_{ik}(\rho_{ik}), \qquad (31)$$

$$\frac{\partial h_{ik}(C_{ik}, f_{ik})}{\partial C_{ik}} = \int_0^{\rho_{ik}} D_{ik}(\rho_{ik}) d\rho_{ik} - \rho_{ik} D_{ik}(\rho_{ik}). \qquad (32)$$

It can be shown that, on the RHS of (32), 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.

E. Chen's Algorithm (for comparison)

As explained in the introduction, the algorithms by Tassiulas [3], Neely [4], Merlin [6], and Chen [5] are essentially identical, as far as routing and scheduling are concerned. Since Chen's algorithm also includes flow control, it provides the best match for comparison with our algorithm. This algorithm, maintains separate queues at each node for the traffic going to each destination. The algorithm, when adapted to our network model and notation, may be stated as follows:

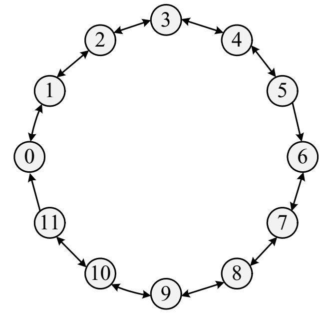


Fig. 2. Topology of networks N1 and N2, consisting of 12 nodes and 22 one-way links. Notice that there are no links from node 6 to node 5, and from node 0 to node 11.

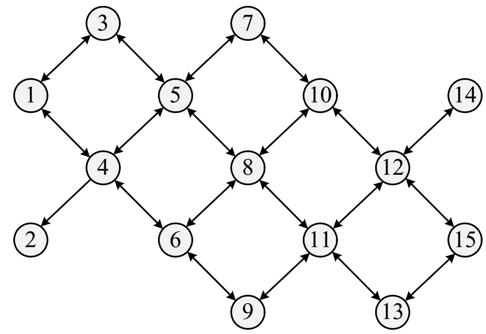


Fig. 3. Topology of network N3, consisting of 15 nodes and 39 one-way links. Node 2 does not have any outgoing link.

For scheduling during each time slot t, first the best destination j_{ik} to be served on each link $(i, k) \in \mathcal{L}$, is determined as:

$$j_{ik} = \arg \max_j \; [Q_i^j - Q_k^j], \qquad (33)$$

where Q_i^j is the total amount of data stored, at the beginning of t, at node i and destined for node j. Then, the optimal schedule m^* is determined from (30), with the link weights w_{ik} defined as:

$$w_{ik} = Q_i^{j_{ik}} - Q_k^{j_{ik}}. \qquad (34)$$

For flow control, the rate r_s of each session $s \in \mathcal{S}_i^j$, is frequently updated as the solution to the following equation:

$$-e_s'(r_s) = Q_i^j. \qquad (35)$$

VI. Simulation Results

In this section, we use simulation to study the behavior of the proposed MCRFS algorithm as outlined in (27)-(30), and to compare its performance with Chen's algorithm as stated in (33)-(35).

A. Simulation Experiments

We present the results of four simulation experiments on three networks, as explained below.

Common Features: The following features are commonly assumed in all cases. All packets are of equal length. There is only $B = 1$ channel available. A one-hop interference model is assumed, i.e., we assume that two links (i, k) and (v, w) cannot be activated at the same time if $i = v$, or $(i, w) \in \mathcal{L}$, or $w = k$, or $(v, k) \in \mathcal{L}$. The transmission rates of activated links are independent of the receiving nodes, i.e., $\mu_{ik} = \mu_i, \ \forall (i, k) \in \mathcal{L}$. All active sessions are greedy, i.e., $\tilde{r}_s = \infty, \ \forall s \in \mathcal{S}$. For an active session with the assigned rate r_s, packets arrive with constant interarrival times $\frac{1}{r_s}$. In the case of both algorithms, the MWIS problem (30) arising in each time slot is solved exactly and through exhaustive search.

Network N1: This network is based on the loop topology of Fig. 2, consisting of 12 nodes $i = 0, 1, ..., 11$, all with transmission rate $\mu_i = 5$ packets/slot, and 22 (one-way) links. Each pair of adjacent nodes are connected in both directions, except for the node pairs $(5, 6)$ and $(11, 0)$ which are only connected in the clockwise direction. This could happen, for example, when nodes 0 and 6 are able to communicate with one neighbor which is closer, but not the other neighbor. There are 12 1-hop sessions, going from each node i to node $i \oplus 1$, where $\oplus$ denotes modulo 12 addition. Similarly, there are 12 6-hop sessions, going from each node i to node $i \oplus 6$. All 24 sessions are permanently active, after $t = 0$.

Network N2: This network is identical to network N1, except that there are 2 additional 6-hop sessions, from node 0 to node 6, and from node 6 to node 0. These two sessions are uncontrolled, i.e., they are not subject to flow control. They periodically, and synchronous with each other, spend 5,000 slots in the OFF state and 5,000 slots in the On state. In the On state, each one transmits at the rate of 1.67 packets/slot.

Network N3: This network has 15 wireless nodes and 39 one-way links, as shown in Fig. 3. 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 Chen's 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. There are 4 metasessions (to be shortly defined) originating from each network node. The activity of each metasession follows an ON-OFF markov process, with the average ON and OFF periods equal to 5,000 slots. They are called metasessions because each of them, at the beginning of every new ON state, randomly picks up a new destination for itself, in effect becoming a new session. Therefore, at any time, we have an average of 30 active sessions in the network, and an average of 2 (and a maximum of 4) active sessions per originating node. These sessions are randomly replaced by new sessions with new destinations, as explained.

Sessions priority Functions: The following two priority functions are used in the experiments:

P1: $-e'(r) = \frac{10}{r}$

P2: $-e'(r) = \frac{10}{r^2}$

Experiment 1: The MCRFS algorithm was tested on network N1, using both priority functions P1 and P2, with a runtime of 40,000 slots. Simulation results are illustrated in Fig. 4.

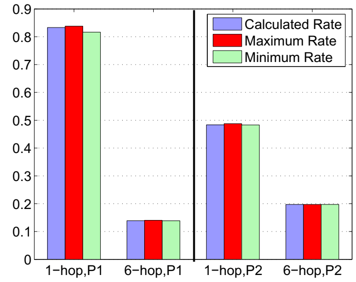


Fig. 4. Assigned rates under the MCRFS algorithm for Network N1. The results on the left hand and right hand sides are obtained under P1 and P2, respectively.

Experiment 2: Both algorithms were tested on network N1, using priority function P1, with a runtime of 40,000 slots. For simulation results see Fig. 5.

Experiment 3: Both algorithms were tested on network N2, using priority function P1, with a runtime of 40,000 slots. Simulation results are shown in Fig. 6.

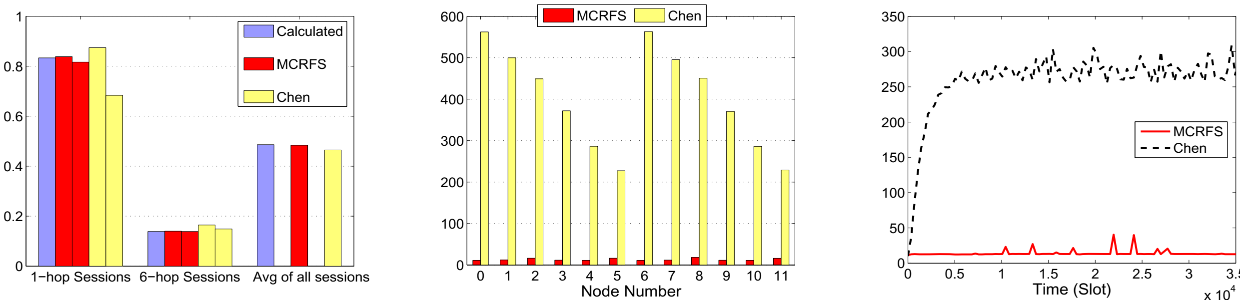
Experiment 4: Both algorithms were tested on network N3, using priority function P1, with a runtime of 70,000 slots. Fig. 7 illustrates the simulation results.

B. Analysis of Results

We now study the results of simulation from several perspectives.

1) Fairness among Sessions: Due to the symmetry in Network 1, it can be shown that, under perfect fairness, all 1-hop sessions must have the same rate r_{1-hop}. Similarly, all 6-hop session should have equal rates r_{6-hop}. Moreover, under priority functions P1 and P2, we must have $r_{1-hop} = 6 \ r_{6-hop}$, and $r_{1-hop} = \sqrt{6} \ r_{6-hop}$, respectively. With the existing symmetry and the one-hop interference model assumed, one can easily conclude that each clockwise link should be activated during one third of time slots, provided that routing is loop-free. Therefore, for clockwise links, $C_{ik} = \mu_i/3$. Assuming further that clockwise links are fully utilized, we get $r_{1-hop} + 6 \ r_{6-hop} = \mu_i/3 = 5/3$ packet/slot. The ideal rates calculated from these results, under P1 and P2, are illustrated in Figures 4 and 5(a), along with simulation results. In Fig. 4, we note a perfect match between calculated results and the highest and the lowest rates assigned to sessions with equal hop count, under both priority functions P1 and P2. We also notice that under P2, assigned rates are less impacted by the number of hops, compared to P2. We conclude that algorithm MCRFS, at least for Network N1, perfectly matches expectations with respect to fairness.

In Fig. 5(a) we note that the highest and the lowest rates assigned under Chen's algorithm to sessions with equal hop count are different by about 25%, in the case of 1-hop sessions, and by about 10%, for 6-hop sessions. We may conclude that Chen's algorithm also achieves some fairness, although it is very coarse.

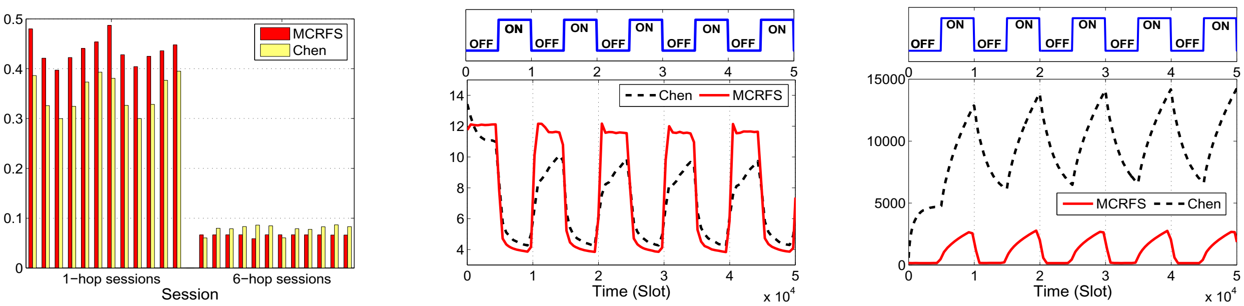


(a) The highest and lowest rates of the two algorithms and the calculated rates for 1-hop and 6-hop sessions (in packets/slot). The average rates for all sessions are also illustrated.

(b) Node queue sizes, averaged over time (in packets)

(c) End-to-end delay, averaged over all received packets (in time slots)

Fig. 5. Performance comparison between the MCRFS and Chen algorithms for Network N1.



(a) Average assigned rates to each controlled session (in packets/slot)

(b) Aggregate throughput of the network (in packets/slot), including uncontrolled sessions

(c) The average of node queue sizes (in packets)

Fig. 6. Performance comparison between the MCRFS and Chen algorithms for Network N2.

2) Throughput: We have already noticed in Fig. 4 that maximum throughput is achieved under the MCRFS algorithm, with both functions P1 and P2. Fig. 5(a) shows that the throughput under Chen's algorithm (with P1) is slightly smaller, compared to MCRFS algorithm. Fig. 6(b) indicates more difference in the throughput under the two algorithms. This is because Chen's algorithm shows slowness (Fig. 6(b)) in adopting to the ON-OFF behavior of the uncontrolled sessions. Finally, we notice in Fig. 7(a) that the throughput of Network 3 under Chen's algorithm is slightly better than MCRFS algorithm. Exact numerical evaluation of simulation results puts the difference at 1.8%.

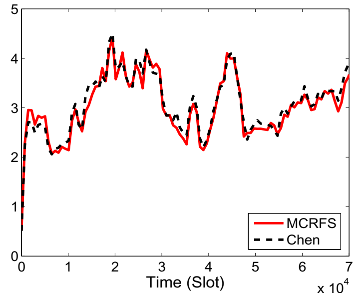
3) Queue Sizes and End-to-End Delays: Here is where the most significant difference between the two algorithms is noticed. Figures 5(b) and 7(b) compare the node queue sizes under the two algorithms, once they are averaged over time. Figures 6(c) and 7(d), compare the time variations of node queue sizes under the two algorithms, after the queue sizes have been averaged across network nodes. Notice that the term *node queue*, in the case of MCRFS algorithm, refers to the aggregated queue of all outgoing links at the node. For Chen's algorithm, the aggregated queue of all sessions at a node is referred to as node queue. Overall, we notice that the queues sizes under Chen's algorithm are much larger than MCRFS algorithm, often by a factor exceeding 10. The difference in queue sizes translates into a huge difference in end-to-end delays, as witnessed by Figures 5(c) and 7(f). Finally, Figure 7(c) compares the maximum queue size attained by each node during the course of simulation, under the two algorithms. According to this figure, buffer requirements under the MCRFS algorithm, are substantially less than Chen's algorithm.

The reason behind these significant differences in delays and queue sizes of the two algorithms are already discussed in the Introduction and in Section III.

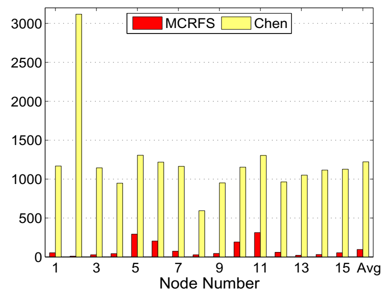
Lastly, we notice from Fig. 7(e) that the queue of node 2, which has no outgoing link, grows indefinitely under Chen's algorithm. These packets are either admitted to the network at node 2 or forwarded from neighboring nodes, due to the algorithm's peculiar routing mechanism. Notice that, although the queue of each session at node 2 must reach some equilibrium, the total number of packets trapped there continues to increase as new sessions are activated.

VII. Conclusion

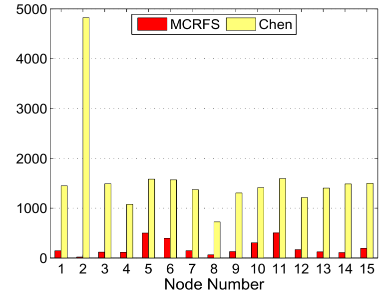
We used a unified convex optimization framework to come up with a joint scheduling, flow control and routing algorithm. Our approach is novel in that it can be viewed as a generalization of minimum delay routing, and guides traffic along path or paths that have the shortest distance (with



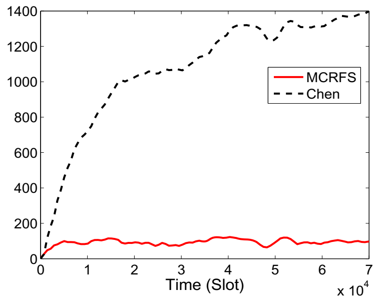
(a) Aggregate throughput of the network as measured at the destinations (in packets/slot)



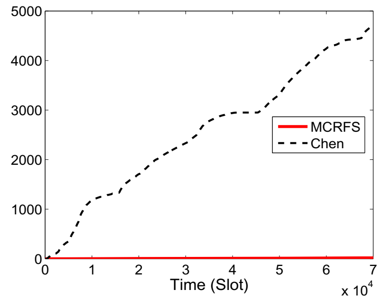
(b) Node queue sizes averaged over time (in packets)



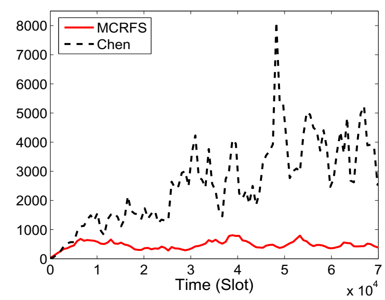
(c) Maximum queue sizes at each node (in packets)



(d) The average of node queue sizes (in packets)



(e) Number of Blocked Packet in Node 2 (in packets)



(f) End-to-end delay averaged over all received packets (in time slots)

Fig. 7. Performance comparison between the MCRFS and Chen algorithms for Network N3.

the length of each link properly defined). We noticed that this property gives rise to substantial performance advantage over alternative algorithms in terms of queue size, power consumption, and end-to-end delays, without compromising the throughput. We also established, through both simulation and analysis, that the algorithm possesses desirable properties with respect to the provision of fairness and priorities among the users.

References

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

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

[3] 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.

[4] 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.

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

[7] 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.

[8] 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.

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

[11] 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.

[12] S. J. Golestani, R. Rastogi and M. Smith, "Dynamic Channel Assignment in wireless LAN," In *proceedings of IEEE COMSWARE*, 2007.

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

[14] N. Moayedian and S.J. Golestani, "Optimal Scheduling and Routing in Wireless Networks: A New approach," *WCNC*, 2009.

[15] D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, Springer, third edition, 2008.

[16] S.J. Golestaani, *A Unified Theory of Flow Control and Routing in Data Communication Networks*, Ph.D Dissertation, MIT, Jan. 1980. A copy may be downloaded from http://ece.iut.ac.ir/faculty/phds/moayedian/golestani_dissertation.pdf.

[17] S. H. Low and D. E. Lapsley, "Optimization Flow Control-I: Basic Algorithm and Convergence," *IEEE /ACM Transactions on Networking*, VOL. 7, NO. 6, DECEMBER 1999.

[18] R. G. Gallager and S.J. Golestaani, Flow Control and Routing Algorithms for Data Networks," *Proc. 5th Internat. Conf. Comput. Comm.*, 1980. A copy may be downloaded from http://ece.iut.ac.ir/faculty/phds/moayedian/golestani80.pdf.