

Distributed Scheduling in Multihop Wireless Networks with Maxmin Fairness Provisioning

Keivan Ronasi, *Student Member, IEEE*, Vincent W.S. Wong, *Senior Member, IEEE*, and Sathish Gopalakrishnan, *Member, IEEE*

Abstract—Fair allocation of resources is an important consideration in the design of wireless networks. In this paper, we consider the setting of multihop wireless networks with multiple routing paths and develop an online flow control and scheduling algorithm for packet admission and link activation that achieves high aggregate throughput while providing different data flows with a fair share of network capacity. For fairness provisioning, we seek to maximize the minimum throughput provided to flows in the network. To cope with different degrees of data reliability among the different links in the network, we use different channel code rates as appropriate. While we expect performance improvement using channel coding and multipath routing, the main contribution of our work is a joint treatment of network stability, multipath routing and link-level reliability in meeting the overarching goal of maxmin fairness. We develop a decentralized, and hence practical, scheduling policy that addresses various concerns and demonstrate, via simulations, that it is competitive with respect to an optimal centralized rate allocator. We also evaluate the fairness provisioning under the proposed algorithm and show that channel coding improves the performance of the network significantly. Finally, we show through simulations that the proposed algorithm outperforms a class of existing approaches on fairness provisioning, which are developed based on utility maximization.

Index terms: Scheduling, decentralized resource allocation, fairness, multipath routing, channel coding, Lyapunov stability theory.

I. INTRODUCTION

Wireless multihop networks can provide good geographic coverage at low cost. However, wireless links have limited capacity and may interfere with each other. The variation of the link capacity and network traffic can have an impact on the stability of the network. The network is said to be *stable* if every node only has a finite number of packets queued for transmission. Stability is subject to the condition that the data transmission rates lie within the network capacity region, i.e., it is feasible to transmit all packets with bounded delay. Another aspect is that wireless links are not as reliable as wired connections, and data packets may be corrupted during transmission. Moreover, without careful resource allocation strategy, certain users may be starved for network access whereas others may receive an unfairly large share of the

available network bandwidth (e.g., see [1]). This latter aspect relates to *fairness*. We propose to address such problems through a stable and decentralized scheduling mechanism that allocates resources such that wireless links do not interfere with each other and fairness is provided while maintaining a high network throughput. We shall begin with a discussion of the key ideas and highlight our main contributions.

Most resource allocation problems can be formulated as network utility maximization (NUM) problems. The utility function represents an objective that is to be maximized and the constraints model the different underlying network characteristics. The NUM approach has been applied in different problems, including energy minimization [2], congestion control [3], and cross-layer optimization [4]. Other rate allocation approaches are also considered. In [5], a rate control protocol has been proposed and a control theoretic analysis of the system has been provided. However, these approaches do not provide the scheduling policy in a slotted notion of time.

Using a slotted notion of time, we consider link scheduling to determine the active links in each time slot. Lyapunov techniques have been used to construct stable and optimal decentralized scheduling policies [6]. They are also applicable to throughput maximization [7] and energy minimization in single hop [8] and multihop [9] networks. A utility optimal algorithm with delay consideration using the shortest paths is also developed in [10]. Throughput optimal scheduling in ad hoc networks, which is an NP-hard problem [11], has often been reduced to a rate allocation problem, which only provides an upper bound on the rates that a network can support. Near optimal scheduling algorithms for mobile ad hoc networks have been proposed in [12]. However, fairness is not considered in the above mentioned work.

Fairness provisioning in wireless networks has been considered [13]–[19]. The impact of imperfect scheduling on network performance is studied in [13]. Proportional fairness is provided in single hop wireless networks using token counter mechanisms [14]. Fairness provisioning is also studied using back-pressure combined with random access algorithms [15]. In [16], fairness is provided in the cellular networks, where there is only one transmitter and all transmissions are one hop. In [17], fairness is provided with maximizing the summation of utility functions corresponding to the individual flows. Our work is different from all of the above in that the minimum throughput of the network is directly maximized to provide maxmin notion of fairness instead of considering utility functions for different users. We achieve this by using Lyapunov stability theory and constructing virtual queues.

Manuscript was received on Mar. 20, 2011, revised on Oct. 29, 2011, and accepted on Jan. 13, 2012. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The review of this paper was coordinated by Dr. Felipe A. Cruz-Perez.

K. Ronasi, V.W.S. Wong, and S. Gopalakrishnan are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, V6T 1Z4, email: {keivanr, vincentw, sathish}@ece.ubc.ca.

In this paper, we take the advantage of two opportunities offered by multihop networks. First, we employ multiple paths for data flows from the source to the destination. Second, we utilize different channel code rates at different links to compensate for variations in link reliability. By distributing the network load over multiple paths, the capacity of the network is better utilized and less data needs to be transmitted over links with lower reliability. Multipath routing has been explored to improve network behavior. The impact of multipath routing on energy consumption is examined in [20]. The effectiveness of multiple paths in meeting delay constraints is also studied in [21].

Channel coding has been used to tolerate link-level errors by adding redundant bits to the data bits in a codeword [22]. By increasing the number of redundant bits in a codeword, one can increase the probability of decoding a codeword correctly at the receiver; the tradeoff is that redundancy increases the network load thereby reducing the effective data throughput. The aspect of improving the reliability of data delivery through channel coding in a wireless network has been considered. Lee *et al.* have examined the rate-reliability tradeoff [23], but they considered a single routing path between each source-destination pair. In previous work, we improved the network throughput with channel coding and multipath routing [24]. Maxmin fairness provisioning is also considered in that context [25]. The impact of channel coding and multipath routing on delay improvement is also studied in [26]. However, we used a centralized NUM approach to determine the average rates but not the exact scheduling policy.

Our work differs from the previous work in the literature in several aspects. While most of the NUM problems determine the average sending rates [3], [4] at which the objective function is maximized, in this paper we propose an efficient scheduling and link activation policy. Our work is also novel because in the related work on fair scheduling (e.g., [13]–[17], [27]), fairness is improved by maximizing the utility function while we directly maximize the minimum throughput in the network. We will show how this improves the network performance. Maxmin fair scheduling is considered in one hop networks with multi-radio receivers [19]. Optimal maxmin transmission and forwarding rates for sensor networks are studied in [28]. A maxmin fair scheduling policy for one hop wireless networks is also considered [29]. Our work is different from the above as we propose a distributed scheduling policy and consider multihop networks with single radio and therefore interference effects are incorporated. In addition, the optimality is analytically proved. While some papers considered channel coding to improve the network throughput [23], no previous work has considered both multipath routing and channel coding in the joint problem of code rate assignment, flow control and decentralized scheduling to achieve maxmin fairness in wireless networks.

The main contributions of this paper are as follows:

- We propose a decentralized online scheduling and flow control algorithm, which we call DisF, that aims to provide fairness for each flow by maximizing the minimum throughput in a multihop wireless network. We integrate the use of multipath routing and link-dependent channel

code rates in our solution approach to utilize resources better and improve reliability. While we do improve link reliability by using appropriate channel coding, we also assume the use of link-level acknowledgments and retransmissions when a node is unable to decode a packet. We demonstrate stability of the proposed algorithm by applying the Lyapunov stability theory.

- We develop an optimal centralized rate allocation method using geometric programming, which provides an upper bound on the performance of any decentralized scheduling policy.
- We study the performance of DisF and optimal centralized algorithms through simulations over multiple random topologies. We show that the DisF algorithm ensures stability, whenever feasible, and that its performance is comparable to that of the centralized optimal solution. We also compare DisF with Lyapunov stability-based algorithms, which do not consider fairness in their design. Next, we show that the use of different channel code rates can improve the performance of the network. We also compare DisF with a class of existing approaches which use utility functions to provide maxmin fairness through simulations [17].

This paper is organized as follows. The system model is described in Section II. The decentralized stable algorithm is developed in Section III. The centralized approach is formulated as a geometric programming problem and it is solved in Section IV. In Section V, the algorithm is evaluated through simulations and the paper is concluded in Section VI.

II. SYSTEM MODEL

We model the wireless network with a graph $G(\mathcal{N}, \mathcal{E})$, where \mathcal{N} represents the set of $N = |\mathcal{N}|$ wireless nodes and \mathcal{E} denotes the set of directed wireless links. Link $e = (m, n) \in \mathcal{E}$ connects two nodes $m, n \in \mathcal{N}$ if and only if node n is in the transmission range of node m . We use the notations e and (m, n) interchangeably. The set of data flows is denoted by \mathcal{F} and the number of data flows is denoted by $F = |\mathcal{F}|$. The set of source nodes is denoted by \mathcal{S} . Data transmission between a source $s_f \in \mathcal{S}$ and the destination d_f of flow $f \in \mathcal{F}$ can be relayed through multiple hops.

We use multipath routing for data transmission. The set \mathcal{K}_f contains $K_f = |\mathcal{K}_f|$ routing paths for flow $f \in \mathcal{F}$. For each link $e \in \mathcal{E}$, path $k \in \mathcal{K}_f$, and flow $f \in \mathcal{F}$, we define $a_e^{fk} = 1$ if link e belongs to the k^{th} routing path for flow f , and $a_e^{fk} = 0$, otherwise. For any node $n \in \mathcal{N}$, each data flow $f \in \mathcal{F}$, and any path $k \in \mathcal{K}_f$, let i_n^{fk} and $o_n^{fk} \in \mathcal{E}$ be the input and output links to and from node n on path k of flow f , respectively. Whenever the context is clear, we remove the indices n, f, k and denote the input and output links with i and o , respectively (see Fig. 1).

A slotted notion of time is used with time slots $t \in \{1, 2, \dots\}$. We denote the value of time-varying parameters at the beginning of each time slot t with the index t . We use the same parameter without the index t to denote its average value over all time slots. At each intermediate node $n \in \mathcal{N}$, we assume a separate queue for any path $k \in \mathcal{K}_f$ of flow

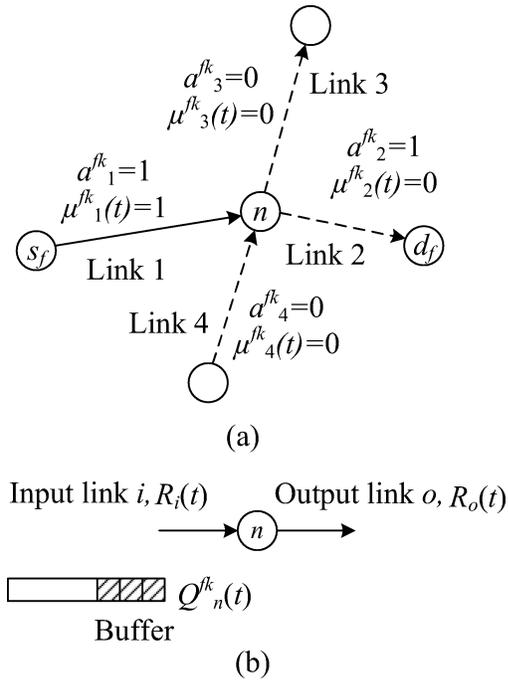


Fig. 1. (a) Path k of flow f from node s_f to node d_f which uses node n as a relay node is shown. It is shown that solid link 1 is active ($\mu_1^{fk}(t) = 1$) while dotted links 2 and 3 are not active ($\mu_2^{fk}(t) = \mu_3^{fk}(t) = 0$) in that particular time slot t . Note that links 1 and 2 belong to the mentioned path ($a_1^{fk} = a_2^{fk} = 1$) but link 3 does not ($a_3^{fk} = 0$). (b) A relay node n is shown with its input link i and its output link o corresponding to the k^{th} path of flow f . The corresponding packets are stored in Q_n^{fk} before they are sent.

$f \in \mathcal{F}$. The number of data bits corresponding to path k for flow f , stored in node n is denoted as $Q_n^{fk}(t)$. We assume $Q_n^{fk}(t) = 0, \forall t, k \in \mathcal{K}_f, f \in \mathcal{F}$ since the received bits are transferred to the upper layers at the destination node d_f . We incorporate all the queue backlogs in the vector $\mathbf{Q}(t) = (Q_n^{fk}(t), \forall n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F})$.

We use link-dependent channel code rates to counter channel variations and improve network reliability. Each source node or intermediate node $n \in \mathcal{N}$ for any flow encodes data bits by adding redundant bits and transmitting the resultant codeword of length g . Hereafter, we assume each packet consists of one codeword and we use the terms packet and codeword interchangeably. We define the code rate $R_e(t)$ as the ratio of the data bits to the total transmitted bits (data plus redundant bits) on link $e \in \mathcal{E}$. We concatenate code rates for all links $e \in \mathcal{E}$ in vector $\mathbf{R}(t) = (R_e(t), \forall e \in \mathcal{E})$. The smaller the code rate $R_e(t)$, the greater number of redundant bits is added, and the higher the reliability is. The reliability is gained at the cost of increased network traffic. When $R_e(t)$ is equal to one, channel coding is not used on link e .

We use $R_{0e} \leq 1$ to denote the *cut-off rate* of wireless link $e \in \mathcal{E}$. The cut-off rate is a channel parameter to which the rate of the adopted coding scheme is always limited [22]. In general, R_{0e} depends on the particular modulation scheme which is being used and also the signal-to-noise ratio (SNR) in the receiver node. For example, for a binary phase shift keying (BPSK) waveform [22], we have $R_{0e} = 1 - \log_2(1 + e^{-\gamma_e})$, where γ_e denotes the SNR at the receiver node of wireless link $e \in \mathcal{E}$. When γ_e is relatively large, R_{0e} is close to 1.

Given $R_e(t) \leq R_{0e}$ for $e \in \mathcal{E}$, we have

$$P_e(t) \geq 1 - 2^{-g(R_{0e} - R_e(t))}, \quad (1)$$

where $P_e(t)$ is the probability that a codeword of length g is received correctly on link e with rate $R_e(t)$ [22, pp. 392-397]. The vector $\mathbf{P}(t) = (P_e(t), \forall e \in \mathcal{E})$ represents the successful probabilities on all links $e \in \mathcal{E}$. For the rest of this paper, we consider the worst case in which inequality (1) is satisfied with equality. For each transmission on link $e \in \mathcal{E}$, we define $\rho_e(t) = 1$ if the packet is transmitted successfully and $\rho_e(t) = 0$ otherwise. We have $\rho_e(t) = 1$ with the probability of $P_e(t)$. We define $\boldsymbol{\rho}(t) = (\rho_e(t), e \in \mathcal{E})$ as the *channel state* at time slot t .

As mentioned above, a codeword may be corrupted with probability $1 - P_e(t)$ through a transmission on link $e \in \mathcal{E}$. The receiver at link e sends a link-level acknowledgement (ACK) to the transmitter if the packet is received correctly. The transmitter retransmits the packet if no ACK is received within a predefined time period. Retransmissions ensure that packets admitted to the network will be received at their corresponding destination nodes. This is at the cost of increased network load.

We denote the number of data bits which are admitted to the path $k \in \mathcal{K}_f$ of flow $f \in \mathcal{F}$ at the beginning of time slot t as $\alpha_k^f(t)$. The vector $\boldsymbol{\alpha}(t) = (\alpha_k^f(t), \forall k \in \mathcal{K}_f, f \in \mathcal{F})$. Suppose all admissions are upper bounded (i.e., $\alpha_k^f(t) \leq \alpha_{max}$). We assume that all source nodes are backlogged (i.e., each source node has at least α_{max} data bits available to send over each of its routing paths at any time slot). We define the capacity region Λ as the closure of the set of all sending rate vectors $\boldsymbol{\alpha}$ (considering all possible routing and scheduling policies), for which the network is stable, that is

$$\Lambda = \left\{ \boldsymbol{\alpha} \mid \boldsymbol{\alpha} \succeq \mathbf{0}, \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{\substack{k \in \mathcal{K}_f, f \in \mathcal{F}, \\ n \in \mathcal{N}}} E\{Q_n^{fk}(\tau)\} < M \right\}, \quad (2)$$

where M is a finite number. Note that $\boldsymbol{\alpha} = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \boldsymbol{\alpha}(\tau)$ is the time average value of $\boldsymbol{\alpha}(t)$.

Two links $e_1, e_2 \in \mathcal{E}$ mutually interfere with each other if and only if the receiver of one link is in the transmission range of the sender of the other. At each time slot t , only one wireless link may be active among those wireless links which are in mutual interference with each other. We define $\mu_e^{fk}(t) = 1$ if link e is active in data transmission for the k^{th} routing path of flow f at time slot t , and $\mu_e^{fk}(t) = 0$ otherwise. We define c_e as the number of bits that can be transmitted by link $e \in \mathcal{E}$ in each time slot t . c_e contains *data* bits as well as *redundant* bits due to channel coding. An example of the modeled network is depicted in Fig. 1.

III. DECENTRALIZED AND STABLE SCHEDULING

In this section, we tackle the problem of online flow control and scheduling for wireless links. Consider the following maxmin fair optimization problem.

$$\begin{aligned} & \text{maximize} && \min_{f \in \mathcal{F}} \alpha_f \\ & \text{subject to} && \boldsymbol{\alpha} \in \Lambda. \end{aligned} \quad (3)$$

The goal in problem (3) is to admit new packets and schedule the transmissions such that the minimum sending rate $\alpha_f = \sum_{k \in \mathcal{K}_f} \alpha_f^k$ over all flows $f \in \mathcal{F}$ is maximized and all queues in the network remain stable, that is the number of bits stored in any queue is bounded. Note that data bits are removed from the queue of the sender node only after it has received an ACK from the receiver. Therefore, if the queues are stable, the sending rate of each flow is the same as its throughput at the corresponding destination.

To enhance the minimum throughput of the network, we need to introduce a decision parameter $\lambda(t)$ and a set of *virtual queues* $Z_f(t), \forall f \in \mathcal{F}$. We denote $\mathbf{Z}(t) = (Z_f(t), \forall f \in \mathcal{F})$. For each virtual queue $Z_f(t)$ for flow f at each time slot t , we set $\sum_{k \in \mathcal{K}_f} \alpha_f^k(t)$ as the service rate and $\lambda(t)$ as the input rate. Then, we have the following update equation:

$$Z_f(t+1) \leq \max \left[Z_f(t) - \sum_{k \in \mathcal{K}_f} \alpha_f^k(t), 0 \right] + \lambda(t). \quad (4)$$

Suppose $\lambda(t)$ is upper bounded (i.e., $\lambda(t) \leq \lambda_{max}$ for any time slot t) and its time average $\lambda = \lim_{t \rightarrow \infty} \frac{\sum_{\tau=0}^{t-1} E\{\lambda(\tau)\}}{t}$ exists. We will show later that burstiness in the network increases when λ_{max} increases. The stability of each virtual queue Z_f implies that the time average of its input rate is less than or equal to that of its service rate. That is

$$\lambda \leq \sum_{k \in \mathcal{K}_f} \alpha_f^k, \quad (5)$$

where $\alpha_f^k = \lim_{t \rightarrow \infty} \frac{\sum_{\tau=0}^{t-1} E\{\alpha_f^k(\tau)\}}{t}$ is the time average value of $\alpha_f^k(t)$. Therefore, if all virtual queues are stable, maximizing the time average value of $\lambda(t)$ is equivalent to maximizing the minimum throughput among all data flows in the network. The goal is to maximize the time average value of $\lambda(t)$ such that both real queues (which store the data bits) and virtual queues remain stable.

We now present some aspects of Lyapunov stability theory [6] that are useful for developing our scheduling algorithm. Let *Lyapunov function* $L(\Theta(t))$ be a non-negative function of any queue vector $\Theta(t)$. We define the *Lyapunov drift* $\Delta(\Theta(t)) \triangleq E\{L(\Theta(t+1)) - L(\Theta(t)) \mid \Theta(t)\}$.

Proposition 1: (Lyapunov Optimization [6]) Let $u(t)$ be a utility function and $B > 0$, $\epsilon > 0$, and $V > 0$ be constants such that for all time slots t and queue vector $\Theta(t) = (\Theta_q(t) \mid q \in \Pi = \{1, 2, \dots, |\Pi|\})$, we have

$$\Delta(\Theta(t)) - VE\{u(t) \mid \Theta(t)\} \leq B - \epsilon \sum_{q \in \Pi} \Theta_q(t) - Vu^*, \quad (6)$$

where u^* is a target value for utility function $u(t)$, then we have

$$u_{inf} \geq u^* - B/V,$$

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{q \in \Pi} E\{\Theta_q(\tau)\} \leq \frac{B + V(u_{sup} - u^*)}{\epsilon},$$

where $u_{inf} = \lim_{t \rightarrow \infty} \inf \frac{1}{t} \sum_{\tau=0}^{t-1} E\{u(\tau)\}$ and $u_{sup} = \lim_{t \rightarrow \infty} \sup \frac{1}{t} \sum_{\tau=0}^{t-1} E\{u(\tau)\}$.

The proof of the proposition can be found in [6, pp. 82-84].

Note that the expectation is over random parameters such as channel states and possibly randomized scheduling policies.

Let $u(t) = \lambda(t)$. We concatenate the backlog queues and virtual queues in the vector $\Theta(t) = (\mathbf{Q}(t), \mathbf{Z}(t))$. Proposition 1 states that if condition (6) holds under a scheduling algorithm, then all the queues in $\Theta(t)$ are stable and λ will be at most B/V away from the target value λ^* . Stability of virtual queues \mathbf{Z} ensures that λ is always less than or equal to the minimum throughput of the network. By increasing V , we can get closer to the target value at the cost of a linear increase in the congestion in the network. Next, we obtain $\Delta(\Theta(t))$ for any time slot t . We define

$$L(\Theta(t)) = \sum_{n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F}} \frac{Q_n^{fk}(t)^2}{2} + \sum_{f \in \mathcal{F}} \frac{Z_f^2(t)}{2}. \quad (7)$$

We assume that scheduled transmissions occur at the beginning of each time slot. For an intermediate relay node $n \in \mathcal{N}$, $n \neq s_f$, any path $k \in \mathcal{K}_f$ and flow $f \in \mathcal{F}$, we have

$$\begin{aligned} Q_n^{fk}(t+1) &\leq Q_n^{fk}(t) - \min[Q_n^{fk}(t), c_o R_o(t)] \mu_o^{fk}(t) \rho_o(t) \\ &\quad + c_i R_i(t) \mu_i^{fk}(t) \rho_i(t) \\ &= \max[Q_n^{fk}(t) - c_o R_o(t) \mu_o^{fk}(t) \rho_o(t), \\ &\quad Q_n^{fk}(t) (1 - \mu_o^{fk}(t) \rho_o(t))] \\ &\quad + c_i R_i(t) \mu_i^{fk}(t) \rho_i(t). \end{aligned} \quad (8)$$

For source node $s_f \in \mathcal{S}$, $f \in \mathcal{F}$ and $k \in \mathcal{K}_f$, we have

$$\begin{aligned} Q_{s_f}^{fk}(t+1) &\leq Q_{s_f}^{fk}(t) - \min[Q_{s_f}^{fk}(t), c_o R_o(t)] \mu_o^{fk}(t) \rho_o(t) \\ &\quad + \alpha_f^k(t) \\ &= \max[Q_{s_f}^{fk}(t) - c_o R_o(t) \mu_o^{fk}(t) \rho_o(t), \\ &\quad Q_{s_f}^{fk}(t) (1 - \mu_o^{fk}(t) \rho_o(t))] + \alpha_f^k(t). \end{aligned} \quad (9)$$

Lemma 1: For any $\rho \in \{0, 1\}$, U, R , and $\mu \in \{0, 1\}$ we have

$$\max[U - R\mu\rho, U(1 - \mu\rho)] \leq \max[U - R\mu, 0] + R\mu(1 - \rho). \quad (10)$$

Proof: Let $\mu = 1$. We verify the inequality in both cases when $U \geq R$ and when $U < R$ separately. If $U \geq R$, then we have $U(1 - \rho) < U - R\rho$ and both sides of (10) are equal to $U - R\rho$. On the other hand, if we have $U < R$, the left hand side of (10) is $U(1 - \rho)$ and the right hand side is $R(1 - \rho)$ and the inequality is verified. In the case where $\mu = 0$, inequality (10) states that $U \leq \max[U, 0]$, which is true. ■

From (8) and Lemma 1, for a relay node $n \in \mathcal{N}$, $n \neq s_f$, $k \in \mathcal{K}_f$, and $f \in \mathcal{F}$, we have

$$\begin{aligned} Q_n^{fk}(t+1) &\leq \max[Q_n^{fk}(t) - c_o R_o(t) \mu_o^{fk}(t), 0] \\ &\quad + c_o R_o(t) \mu_o^{fk}(t) (1 - \rho_o(t)) \\ &\quad + c_i R_i(t) \mu_i^{fk}(t) \rho_i(t). \end{aligned} \quad (11)$$

Considering (9) and Lemma 1, for any source node $s_f \in \mathcal{S}$, $k \in \mathcal{K}_f$, $f \in \mathcal{F}$, we have

$$\begin{aligned} Q_{s_f}^{fk}(t+1) &\leq \max[Q_{s_f}^{fk}(t) - c_o R_o(t) \mu_o^{fk}(t), 0] \\ &\quad + c_o R_o(t) \mu_o^{fk}(t) (1 - \rho_o(t)) + \alpha_f^k(t). \end{aligned} \quad (12)$$

We now introduce two lemmas to simplify (11) and (12).

Lemma 2 ([6]): For any positive U_1, U_2, η , and ν , if we

have $U_1 \leq \max[U_2 - \eta, 0] + \nu$, then

$$U_1^2 \leq U_2^2 + \eta^2 + \nu^2 - 2U_2(\eta - \nu). \quad (13)$$

The proof of Lemma 2 can be found in [6].

Lemma 3: For positive U_1, U_2, O, I, ρ , and $\rho' \leq 1$, if $U_1 \leq \max[U_2 - O, 0] + O(1 - \rho') + I\rho$, then

$$U_1^2 \leq U_2^2 + B - 2U_2(O\rho' - I\rho),$$

where $B = O^2 + \rho^2 I^2 + O^2(1 - \rho')^2 + 2\rho(1 - \rho')OI$.

Proof: From Lemma 2, by substituting $\eta = O$ and $\nu = O(1 - \rho') + I\rho$, Lemma 3 is proven. ■

Using Lemma 3 and inequalities (11) and (12), for any $k \in \mathcal{K}_f$ and $f \in \mathcal{F}$, we have

$$Q_n^{fk}(t+1)^2 \leq Q_n^{fk}(t)^2 + B_n^{fk}(t) - 2Q_n^{fk}(t)(c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - c_i R_i(t)\mu_i^{fk}(t)\rho_i(t)), \quad (14)$$

for any intermediate relay node $n \in \mathcal{N}$ ($n \neq s_f$) where

$$B_n^{fk}(t) = c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 + \rho_i(t)^2 c_i^2 R_i(t)^2 \mu_i^{fk}(t)^2 + c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 (1 - \rho_o(t))^2 + 2\rho_i(t)(1 - \rho_o(t))c_o R_o(t)\mu_o^{fk}(t)c_i R_i(t)\mu_i^{fk}(t).$$

Similarly, for source node s_f , we have

$$Q_{s_f}^{fk}(t+1)^2 \leq Q_{s_f}^{fk}(t)^2 + B_{s_f}^{fk}(t) - 2Q_{s_f}^{fk}(t)(c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - \alpha_f^k(t)), \quad (15)$$

where

$$B_{s_f}^{fk}(t) = c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 + c_o^2 R_o(t)^2 \mu_o^{fk}(t)^2 (1 - \rho_o(t))^2 + 2(1 - \rho_o(t))c_o R_o(t)\mu_o^{fk}(t)\alpha_f^k(t) + \alpha_f^k(t)^2.$$

From Lemma 2 and inequality (4), for each virtual queue Z_f , $f \in \mathcal{F}$, we have

$$Z_f^2(t+1) \leq Z_f^2(t) + \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t)\right)^2 + \lambda^2(t) - 2Z_f(t) \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) - \lambda(t)\right). \quad (16)$$

Now, we can write $\Delta\Theta(t) - VE\{\lambda(t) | \Theta(t)\}$ as

$$\begin{aligned} & E\{L(\Theta(t+1)) - L(\Theta(t)) | \Theta(t)\} - VE\{\lambda(t) | \Theta(t)\} \\ & \leq B - \sum_{k \in \mathcal{K}_f, f \in \mathcal{F}} \left(Q_{s_f}^{fk}(t) E\{c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - \alpha_f^k(t) | \Theta(t)\} \right. \\ & \quad \left. + \sum_{n \neq s_f} Q_n^{fk}(t) E\{c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) - c_i R_i(t)\mu_i^{fk}(t)\rho_i(t) | \Theta(t)\} \right) \\ & \quad - \sum_{f \in \mathcal{F}} Z_f(t) E \left\{ \sum_{k \in \mathcal{K}_f} \alpha_f^k(t) - \lambda(t) | \Theta(t) \right\} \\ & \quad - VE\{\lambda(t) | \Theta(t)\}, \end{aligned} \quad (17)$$

where $B_Q = \frac{5NF}{2}(\max_{f \in \mathcal{F}} K_f) \max[\alpha_{max}^2, \max_{e \in \mathcal{E}} c_e^2]$, $B_Z = F((\alpha_{max} \max_{f \in \mathcal{F}} K_f)^2 + \lambda_{max}^2)/2$, and $B = B_Q + B_Z$. Inequality (17) holds under any adopted scheduling algorithm. Note that the expected values are taken over the random channel state probabilities.

Let λ^* be the optimal value for λ which can be achieved by an algorithm such that all backlog and virtual queues

remain stable. Since the corresponding sending rates are stably supported by the network (they are inside the capacity region), there exist link data rates for all wireless links that support data transmission in the network. These data rates can be achieved with a possibly randomized *channel state-only* algorithm \mathcal{X} . This is proved by projection of link data rates in different time-varying channel states and then expressing each projection as the convex combination of corresponding independent sets. Further details can be found in [7]. Assume that Algorithm \mathcal{X} determines the decision parameters ($\lambda(t), R_e(t), \mu_e^{fk}(t)$, and $\alpha_f^k(t)$ for all $e \in \mathcal{E}, k \in \mathcal{K}_f$, and $f \in \mathcal{F}$) at the beginning of each time slot t such that λ achieves the optimal value λ^* , virtual queues are stable (i.e., $\lambda < \sum_{k \in \mathcal{K}_f} \alpha_f^k$ for any $f \in \mathcal{F}$) and backlog queues are also stable. Note that Algorithm \mathcal{X} is a *channel state-only* algorithm which makes the decisions only based on the observed channel states at each time slot. Therefore, it needs *a priori* knowledge on channel states. The stability of backlog queues implies that under Algorithm \mathcal{X} , for an intermediate relay node $n \in \mathcal{N}$ and $n \neq s_f$, we have $E\{c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) | \Theta(t)\} > E\{c_i R_i(t)\mu_i^{fk}(t)\rho_i(t) | \Theta(t)\}$. For any source node $s_f, f \in \mathcal{F}$, we have

$$E\{c_o R_o(t)\mu_o^{fk}(t)\rho_o(t) | \Theta(t)\} > E\{\alpha_f^k(t) | \Theta(t)\}. \quad (18)$$

The stability of virtual queues implies that

$$E\{\lambda(t) | \Theta(t)\} < E\left\{\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) | \Theta(t)\right\}, \quad (19)$$

for all $f \in \mathcal{F}$. Then, from (17), there exists $\epsilon > 0$ such that

$$\begin{aligned} & \Delta(\Theta(t)) - VE\{\lambda(t) | \Theta(t)\} \leq B \\ & - \epsilon \left(\sum_{n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F}} Q_n^{fk}(t) + \sum_{f \in \mathcal{F}} Z_f(t) \right) - V\lambda^*, \end{aligned} \quad (20)$$

under Algorithm \mathcal{X} . Note that here the expectation is taken over different random channel states and different randomized decisions.

We now present the distributed fair (DisF) algorithm for maximizing the minimum throughput in a multihop network with channel coding and multipath routing. The goal of the algorithm is to select the decision parameters $\lambda(t), R_e(t), \alpha_f^k(t)$ and $\mu_e^{fk}(t)$ for all $e = (n_e, n'_e) \in \mathcal{E}, k \in \mathcal{K}_f$, and $f \in \mathcal{F}$, such that

$$\begin{aligned} & \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \alpha_f^k(t) \left(Z_f(t) - Q_{s_f}^{fk}(t) \right) + \lambda(t) \left(V - \sum_{f \in \mathcal{F}} Z_f(t) \right) \\ & + \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \sum_{e \in \mathcal{E}} a_e^{fk} c_e R_e(t) \mu_e^{fk}(t) \left(Q_{n_e}^{fk}(t) - Q_{n'_e}^{fk}(t) \right) P_e(t) \end{aligned} \quad (21)$$

is maximized over all available decision parameters at each time slot t . Before we present the algorithm in detail, we analyze the performance of the algorithm in Theorem 1.

Theorem 1: Let Algorithm DisF be an algorithm that maximizes (21) over all available decision parameters at each time slot t . Then, it is throughput-optimal. That is, it stabilizes the network if any other algorithm can do so and the minimum throughput in the network is at most B/V away from the optimal value.

Proof: Algorithm DisF maximizes (21), which can be rewritten as

$$\begin{aligned} & \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} \left(Q_{s_f}^{fk}(t) (c_o R_o(t) \mu_o^{fk}(t) P_o(t) - \alpha_f^k(t)) \right. \\ & \left. + \sum_{n \neq s_f} Q_n^{fk}(t) (c_o R_o(t) \mu_o^{fk}(t) P_o(t) - c_i R_i(t) \mu_i^{fk}(t) P_i(t)) \right) \\ & + \sum_{f \in \mathcal{F}} Z_f(t) \left(\sum_{k \in \mathcal{K}_f} \alpha_f^k(t) - \lambda(t) \right) + V\lambda(t). \end{aligned} \quad (22)$$

Recall from Section II that for any node $n \in \mathcal{N}$, each data flow $f \in \mathcal{F}$, and any path $k \in \mathcal{K}_f$, i_n^{fk} and $o_n^{fk} \in \mathcal{E}$ denote the input and output links to and from node n on path k of flow f , respectively. We remove the indices n, f, k and denote the input and output links with i and o in (22), respectively. Recall that inequality (17) holds under any algorithm including the DisF algorithm. By maximizing (22) at each time slot t , we minimize the upper bound on $\Delta(\Theta(t)) - VE\{\lambda(t) \mid \Theta(t)\}$ (right hand side of (17)) over its value under any other algorithm including Algorithm \mathcal{X} , that is on the right hand side of (20). Then, under the DisF algorithm, we have

$$\begin{aligned} & \Delta(\Theta(t)) - VE\{\lambda(t) \mid \Theta(t)\} \\ & \leq B - \epsilon \sum_{n \in \mathcal{N}, k \in \mathcal{K}_f, f \in \mathcal{F}} Q_n^{fk}(t) - \epsilon \sum_{f \in \mathcal{F}} Z_f(t) - V\lambda^*. \end{aligned} \quad (23)$$

This is the condition of Proposition 1. \blacksquare

Note that while maximizing (21) over all possibilities leads to an optimal solution, it is an NP-hard problem due to the interference constraints [30]. In order to solve this problem in a distributed manner and in reasonable time (compared to the introduced delay and channel variation time due to channel fluctuations), we use the greedy maximal scheduling (GMS) policy. GMS is sub-optimal and may be implemented in a distributed manner. It has been shown in [31] that its efficiency ratio is at least 1/2 in the 1-hop interference model. In other words, GMS can achieve at least half of the throughput achieved by the optimal policy. Algorithm 1 shows the DisF algorithm which aims at maximizing (21) at any time slot t in a decentralized manner. This algorithm has several phases that are performed at the beginning of each time slot t .

1) *Flow Control:* Each source node s_f checks the backlog queue $Q_{s_f}^{fk}(t)$ for each path $k \in \mathcal{K}_f$ of flow f . If $Q_{s_f}^{fk}(t) \leq Z_f(t)$, s_f schedules α_{max} new data bits for flow f to be admitted to the corresponding path (i.e., sets $\alpha_f^k(t)$ to be equal to α_{max}) (Lines 2-9).

2) *Scheduling:* The candidate set is initialized with all links that have data to send. Each link $e = (n, n') \in \mathcal{E}$, sets its weight w_e equal to the maximum value of $Q_n^{fk}(t) - Q_{n'}^{fk}(t)$ over all paths $k \in \mathcal{K}_f$ and flows $f \in \mathcal{F}$ that use that link. Then the link with the maximum weight is selected to transmit data for the corresponding path and flow, it is removed from the candidate set and all links which interfere with that link are also removed from the set. This process continues until no link remains in the candidate set (Lines 10-20). The scheduling process (lines 15-20) can be implemented in a distributed manner by making modifications to the medium access control (MAC) parameters. This is done in [30] by varying contention

Algorithm 1 Distributed fair (DisF) algorithm for maximizing the minimum throughput among the entire users in the network, executed at the beginning of each time slot t . The algorithm is initiated with $\mathbf{Z}(0) = \mathbf{0}$ at $t = 0$.

```

1: Initialization: Assign values for  $\alpha_{max}, \lambda_{max}, V$ .
2: Set  $\alpha(t) = \mathbf{0}$ .
3: for each source  $s_f \in \mathcal{S}, f \in \mathcal{F}$  do
4:   for each  $k \in \mathcal{K}_f$  do
5:     if  $Q_{s_f}^{fk}(t) \leq Z_f(t)$  then
6:        $s_f$  sets  $\alpha_f^k(t) = \alpha_{max}$ .
7:     end if
8:   end for
9: end for
10: Initiate candidate set  $C$  of all links that have data to send.
11: for each link  $e = (n, n') \in C$  do
12:   Set  $w_e = \max_{f, k} (Q_n^{fk}(t) - Q_{n'}^{fk}(t))$ .
13:   Set  $(f_e^*, k_e^*) = \arg \max_{f, k} (Q_n^{fk}(t) - Q_{n'}^{fk}(t))$ 
14: end for
15: while  $C$  is not empty do
16:   Set  $e^* = \arg \max_{e \in C} w_e$ .
17:   Set  $\mu_{e^*}^{f_e^* k_e^*}(t) = 1$ .
18:   Find  $R_{e^*}(t)$  by solving  $2^{-g(R_{0e^*} - R_{e^*}(t))} = \frac{1}{1 + R_{e^*}(t)g \ln 2}$ .
19:   Remove  $e^*$  and all links that interfere with  $e^*$  from the candidate set  $C$ .
20: end while
21: for each source  $s_f \in \mathcal{S}, f \in \mathcal{F}$  do
22:   if  $\sum_{f \in \mathcal{F}} Z_f(t) \leq V$  then
23:     Set  $\lambda(t) = \lambda_{max}$ .
24:   else
25:     Set  $\lambda(t) = 0$ .
26:   end if
27: end for
28: for each source  $s_f \in \mathcal{S}, f \in \mathcal{F}$  do
29:    $Z_f(t+1) = \max [Z_f(t) - \sum_{k \in \mathcal{K}_f} \alpha_f^k(t), 0] + \lambda(t)$ .
30:   Update the other sources with  $Z_f(t+1)$  through control messages.
31: end for

```

window parameters CW_{min} and CW_{max} for links to transmit more or less aggressively according to their weights w_e .

3) *Code Rate Allocation:* For each scheduled link e , the optimal code rate that maximizes $R_e(t)(1 - 2^{-g(R_{0e} - R_e(t))})$ is determined by solving $2^{-g(R_{0e} - R_e(t))} = \frac{1}{1 + R_e(t)g \ln 2}$ (Line 18). In case of any change in the wireless environment, $R_e(t)$ is determined according to the new available updates for R_{0e} .

4) *Fairness Provisioning:* The source node of each data flow $f \in \mathcal{F}$ sets $\lambda(t) = \lambda_{max}$ if $\sum_{f \in \mathcal{F}} Z_f(t) \leq V$, and sets $\lambda(t) = 0$ otherwise (Lines 21-27). Virtual queues are updated according to (4) (Line 28-31). The value of virtual queues at each time slot is transmitted between the source nodes through control messages. Next, the scheduled links transmit their packets and new data bits are admitted in the source node queues.

IV. GEOMETRIC PROGRAMMING (GP) FORMULATION

In this section, we provide a benchmark for performance of the decentralized scheduling algorithm for evaluation purposes. We formulate the problem of code rate and sending rate allocation as a NUM problem and describe a centralized

solution approach. We allow multipath routing and channel coding to improve the reliability and use retransmissions in case of any data loss. We do not consider scheduling. The slotted notion of time is not considered and the variables are time average values.

Recall that the sending rate for the k^{th} path of flow $f \in \mathcal{F}$ is α_f^k . Channel coding is performed on link $e \in \mathcal{E}$ at a rate of R_e . Thus, link e must transmit packets for path $k \in \mathcal{K}_f$ of flow $f \in \mathcal{F}$ with a rate of α_f^k/R_e if all transmissions are successful. Since the probability of a successful transmission is P_e , each transmission is completed within $1/P_e$ attempts on average where $P_e = 1 - 2^{-g(R_{0e} - R_e)}$ in the worst case. We have

$$u_e^{fk} = a_e^{fk} \frac{\alpha_f^k}{R_e P_e}, \quad (24)$$

where u_e^{fk} is the data rate at which link e is being used to transmit data for the k^{th} path of flow $f \in \mathcal{F}$. We can express the usage of wireless link $e \in \mathcal{E}$ as

$$u_e = \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} a_e^{fk} \frac{\alpha_f^k}{R_e P_e}. \quad (25)$$

To model the interference, we use a contention graph $G_C(\mathcal{N}_C, \mathcal{E}_C)$. The set of vertices \mathcal{N}_C represents the set of wireless links in graph G . There is a link between each two vertices if and only if the corresponding links in graph G interfere with each other. Each *complete* subgraph in graph G_C is called a clique and a maximal clique ω is one that is not a subset of a larger clique. We define Ω as the set of all maximal cliques in the network. It is *necessary* for successful transmissions that the summation of link usages over all links in each maximal clique be less than the capacity of the clique ζ_ω , which leads to an *upper bound* on the network performance.

Now, we can formulate the problem of fair sending rate and code rate allocation in a multihop wireless network as

$$\begin{aligned} & \underset{\alpha, \mathbf{R}, \sigma}{\text{maximize}} && \sigma \\ & \text{subject to} && \sigma \leq \sum_{k \in \mathcal{K}_f} \alpha_f^k, \quad \forall f \in \mathcal{F}, \\ & && \sum_{e \in \omega} \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} a_e^{fk} \frac{\alpha_f^k}{R_e P_e} \leq \zeta_\omega, \quad \forall \omega \in \Omega, \\ & && P_e \leq 1 - 2^{-g(R_{0e} - R_e)}, \quad \forall e \in \mathcal{E}, \\ & && \alpha \succ \mathbf{0}, \mathbf{0} \prec \mathbf{R} \prec \mathbf{R}_0. \end{aligned} \quad (26)$$

In problem (26), the objective is to maximize the throughput of the flow with the minimum value. The second set of constraints satisfies the *necessary* condition for successful transmissions and leads to an upper bound on the performance. Since retransmissions due to packet loss are taken into account in the second set of constraints, the total number of admitted packets will be received at the destination. Therefore, by maximizing the sending rate of any flow, the received throughput is also maximized.

The objective function in problem (26) and the left hand side of the second set of constraints are posynomials. The first set of constraints has *signomials* on the left hand side. Therefore, we can apply *signomial programming* techniques [32] to solve

this problem. In this regard, we need to approximate the right hand side of the first set of constraints with a monomial around an initial point $\hat{\alpha}$. For a parameter $b > 1$ very close to 1, we have

$$\sum_{k \in \mathcal{K}_f} \alpha_f^k \approx \hat{\Lambda}_f^{-1} \prod_{k \in \mathcal{K}_f} \left(\frac{\alpha_f^k}{\hat{\alpha}_f^k} \right)^{\hat{\alpha}_f^k \hat{\Lambda}_f}, \quad \forall \alpha \in [\hat{\alpha}/b, b\hat{\alpha}], \quad (27)$$

where $\hat{\Lambda}_f^{-1} = \sum_{k \in \mathcal{K}_f} \hat{\alpha}_f^k$. Finally, we tackle the third set of constraints in (26). We can approximate the exponential term using the *Taylor series expansion* and rewrite the constraint as

$$P_e \leq 1 - X_{1e} \sum_{n=0}^{\infty} \frac{(X_2 R_e)^n}{n!}, \quad (28)$$

where $X_{1e} = 2^{-gR_{0e}}$ and $X_2 = g \ln 2$. For large M_e , we have

$$P_e \leq 1 - X_{1e} \sum_{n=0}^{M_e-1} \frac{(X_2 R_e)^n}{n!}. \quad (29)$$

M_e must be large enough such that $(X_2 R_e)^{M_e} \ll M_e!$ and can be found through simulations. Now, we can rewrite problem (26) in the standard form of geometric programming (GP) problem as

$$\begin{aligned} & \underset{\sigma, \alpha, \mathbf{R}, \mathbf{P}}{\text{minimize}} && \sigma^{-1} \\ & \text{subject to} && \hat{\Lambda}_f \sigma \prod_{k \in \mathcal{K}_f} (\hat{\alpha}_f^k \alpha_f^k)^{-1} \hat{\alpha}_f^k \hat{\Lambda}_f \leq 1, \quad \forall f \in \mathcal{F}, \\ & && \sum_{e \in \omega} \sum_{f \in \mathcal{F}} \sum_{k \in \mathcal{K}_f} a_e^{fk} \alpha_f^k R_e^{-1} P_e^{-1} \leq \zeta_\omega, \quad \forall \omega \in \Omega, \\ & && \frac{P_e}{1 - X_{1e}} + \frac{X_{1e}}{1 - X_{1e}} \sum_{n=1}^{M_e-1} \frac{(X_2 R_e)^n}{n!} \leq 1, \quad \forall e \in \mathcal{E}, \\ & && \hat{\alpha}/b \preceq \alpha \preceq b\hat{\alpha}, \\ & && \sigma > 0, \mathbf{0} \prec \mathbf{R} \prec \mathbf{R}_0, \mathbf{P} \succ \mathbf{0}. \end{aligned} \quad (30)$$

The above problem is a geometric programming problem that can be solved iteratively using the interior-point method [32]. In each iteration, we use the solution obtained in the previous iteration as the new initial point $\hat{\alpha}$ and use the approximation in (27) around that point. We use the solution of this problem as a benchmark for evaluating the DisF algorithm.

V. PERFORMANCE EVALUATION

In this section, we evaluate our proposed algorithm through Matlab simulations. First, we compare the performance of the DisF algorithm with the centralized approach obtained through solving geometric programming problem. Then, we study fairness provisioning in the network. We quantitatively measure fairness under the DisF algorithm in several random topologies and compare it with a Lyapunov-based algorithm in which the fairness is not considered. Here, we call that algorithm as DisA algorithm. The goal in DisA algorithm is to maximize the aggregate throughput of the network (i.e., $\sum_{f,k} \alpha_f^k$) while DisF algorithm maximizes the minimum throughput in the network (i.e., $\min_f \sum_k \alpha_f^k$). We also show how fairness is provided at the cost of degrading the aggregate throughput in the network. We show the effect of channel coding on the network performance by comparing with the case when

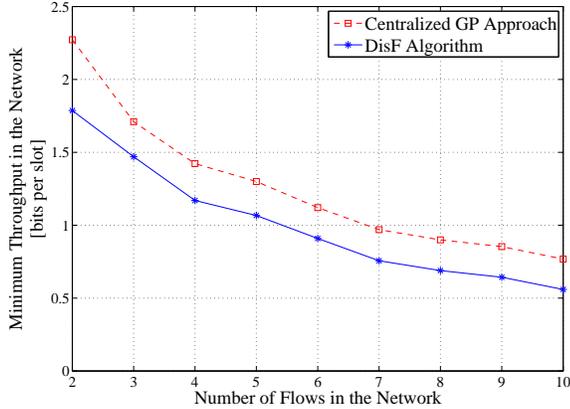


Fig. 2. The performance of the DisF algorithm is compared with the solution of centralized geometric programming problem. Each point represents the average performance obtained over 50 random topologies. The DisF algorithm follows the centralized approach as the number of flows increases.

channel coding is not used. Moreover, we study the effect of algorithm parameter V on the obtained performance of the DisF algorithm. Finally, we compare the proposed approach with the distributed utility-based approach presented in [17].

We compare the minimum throughput under the DisF algorithm with the minimum throughput obtained through solving the geometric programming problem (26). We run the simulations for both approaches in topologies with 30 nodes. The number of flows varies between 2 and 10. In this set of simulations, we set $\alpha_{max} = 2$, $\lambda_{max} = 10$, $V = 50$, and $c_e = 10$ bits for all links $e \in \mathcal{E}$. This is because the MOSEK software [33] that we used to solve the geometric programming problem cannot solve the problem when c_e and consequently M_e (see (29)) grows. We run the simulations on 50 random topologies. The DisF algorithm follows the optimal solution as the number of flows in the network grows (Fig. 2). Increasing the number of flows leads to higher load on the network and causes a degradation in the minimum throughput in the network for both approaches.

Hereafter, we set parameters $\alpha_{max} = 1000$, $\lambda_{max} = 5000$, $V = 250000$, and $c_e = 5000$ bits for all links $e \in \mathcal{E}$. Next, we study the max-min fair (DisF) algorithm in regards of fairness provisioning. We run both DisF and DisA algorithms in a sample network topology (Fig. 3) with 20 nodes and 5 data flows. We observe that the achieved throughput, for the sample topology, is distributed fairly under the DisF algorithm while this is not the case for the DisA algorithm (see Fig. 4).

We also study the fairness provisioning quantitatively on several random topologies. We use the Jain's fairness index [34], to measure the fairness among network users. The fairness index $\psi = \frac{(\sum_{f \in \mathcal{F}} \alpha_f)^2}{|\mathcal{F}| \sum_{f \in \mathcal{F}} \alpha_f^2}$, where $\alpha_f = \sum_{k \in \mathcal{K}_f} \alpha_f^k$ denotes the throughput of flow $f \in \mathcal{F}$. Using the DisF algorithm in several random topologies, the fairness index is always higher than 0.95 while that of DisA algorithm degrades to 0.55 when the number of flows is equal to 10 (see Fig. 5).

Improving the minimum throughput in the network is at the cost of degrading the aggregate throughput which is achievable with the entire users in the network. Fig. 6 shows the tradeoff between the minimum throughput and aggregate throughput

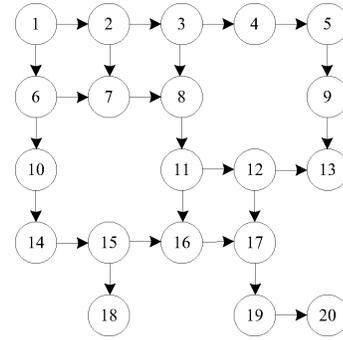


Fig. 3. A sample random topology is shown in this figure. There are five flows between nodes $1 \rightarrow 16$, $3 \rightarrow 13$, $2 \rightarrow 8$, $14 \rightarrow 17$, and $6 \rightarrow 20$. There are 4, 2, 2, 1, and 3 routing paths available for these flows, respectively.

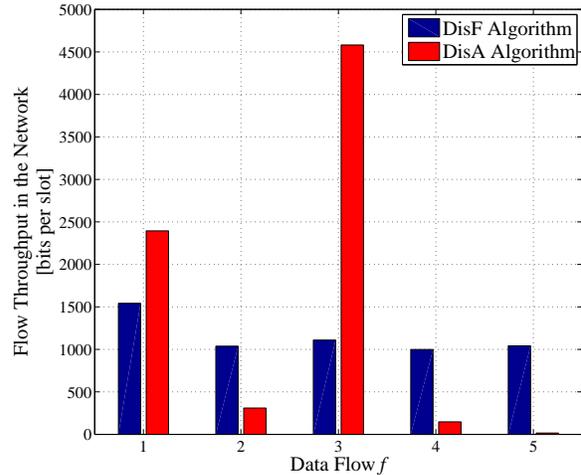


Fig. 4. In a sample topology, fairness is provided under the DisF algorithm while some flows starve under the DisA algorithm that does not consider fairness.

of the network. It is shown that the minimum throughput (and also the fairness) is improved via DisF algorithm (Fig. 6 (a)). However, that is gained at the cost of degrading the aggregate throughput of the network (Fig. 6 (b)).

Fig. 7 shows the effect of channel coding on the performance of the DisF algorithm. Each point in Fig. 7 represents the average value over 50 random topologies. Here, we assume the probability at which a packet is transmitted successfully over a wireless link to be 0.8 if channel coding is not used. We observe that the minimum throughput increases by 28% when the number of flows is 10.

In Fig. 8, we study the effect of varying parameter V on both the minimum throughput and the delay in the network under the DisF algorithm. We used the total backlog in the network as a measure of the delay. We vary V from 0 to 50000 and it is shown that the minimum throughput in the network increases with increasing V at the expense of a linear increase of delay in the network.

Finally, we compare our approach with the one introduced in [17] as an example of a class of approaches that use utility functions to provide different notions of fairness including maxmin fairness. This is comparing to our proposed method which directly pushes the minimum throughput up

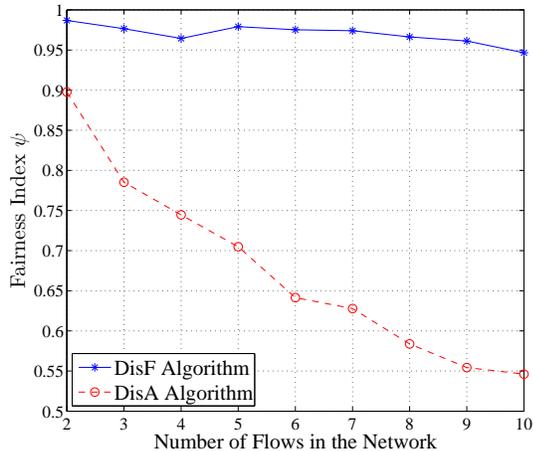


Fig. 5. The performance of DisF and DisA algorithms in regard to fairness provisioning. Each point represents the average value of the fairness index over 50 random topologies.

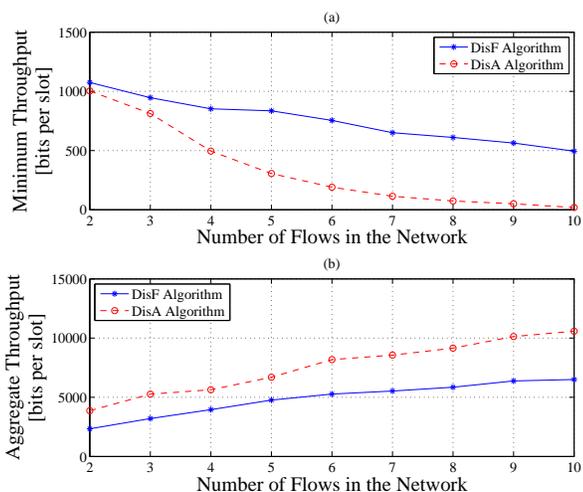


Fig. 6. The tradeoff between minimum throughput in the network (a) and aggregate throughput in the network (b) is shown under both DisF and DisA algorithms.

instead of using utility functions. In [17], the problem of maximizing the aggregate network utility for all flows in the network was considered as maximizing $\sum_f u_f^\beta(\alpha_f)$ subject to $\alpha \in \Lambda$, where $u_f^\beta(x)$ is the utility function (i.e., $u_f^\beta(x) = x^{1-\beta}/(1-\beta)$, $\beta \neq 1$, and $u_f^\beta(x) = \log x$, $\beta = 1$). It was shown that by increasing β in the above utility functions, the performance of the network converges to maxmin fairness provisioning. The two algorithms are compared in Figs. 9 and 10 regarding the minimum throughput they provide for network users and fairness provisioning. Each point represents the average of simulation results in 50 random topologies. To make the comparison fair, we assume wireless links are completely reliable. We do not employ multipath routing and channel coding in the network. Parameter β for the utility function is chosen to be 0, 1, and 100 which is for aggregate throughput maximization, proportional fairness, and maxmin fairness, respectively. It is verified (Fig. 9) that the fairness index is improved under DisF algorithm compared with the

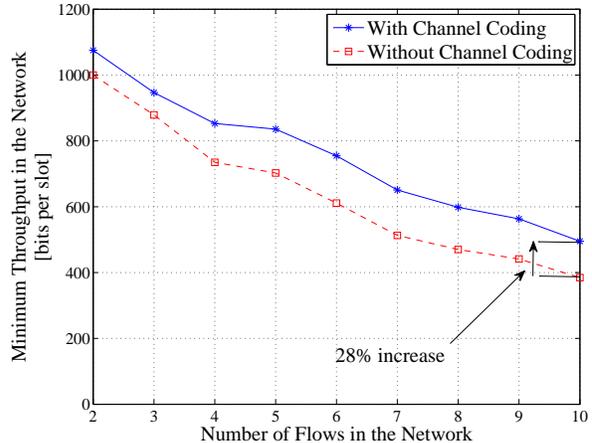


Fig. 7. Performance of the network with channel coding is compared with the case in which channel coding is not being used.

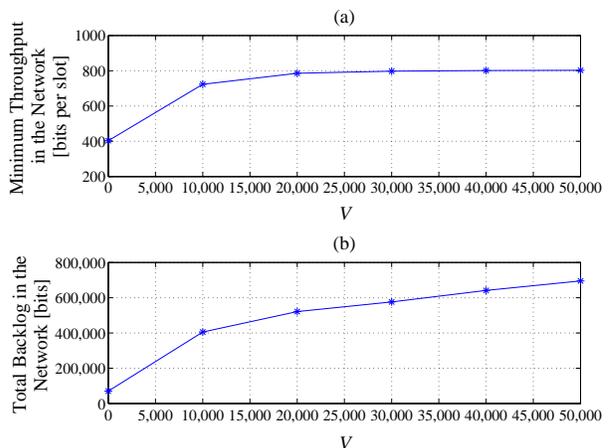


Fig. 8. The effect of increasing parameter V is shown on the minimum throughput (a). This is at the expense of increasing the congestion in the network (b).

utility-based algorithm when $\beta = 0$ and 1. The fairness index of the utility-based algorithm improves when β increases but at the cost of the dramatic decrease in the minimum achieved throughput (Fig. 10).

VI. CONCLUSION

In this paper, we studied fairness provisioning in multihop wireless networks. We developed an online decentralized algorithm to schedule new data packet admission and packet transmissions such that the minimum throughput of the network is maximized. We considered networks with multipath routing and channel coding. We proved the convergence of the algorithm analytically. Through simulations, we showed that the proposed algorithm followed the optimal centralized approach with under control degree of sub-optimality. We also showed that the proposed algorithm improves the performance of the network regarding fairness comparing to the other approaches which ignore fairness provisioning. Moreover, we showed the effectiveness of channel coding on the performance of the network. Finally, we showed through simulations that

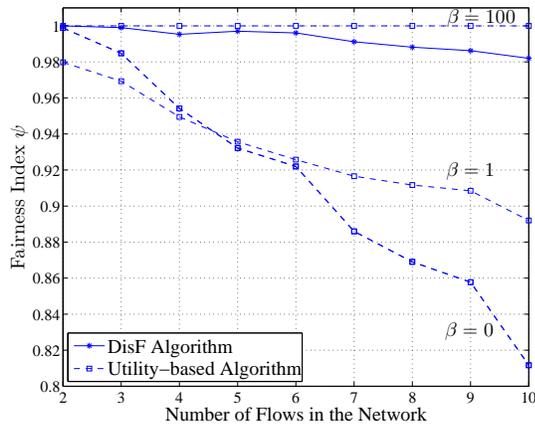


Fig. 9. DisF algorithm is compared with the utility-based approach in terms of fairness provisioning using Jain's fairness index. It is shown that for large values of β , the utility-based approach has a better performance comparing with DisF algorithm. This is at the cost of a decrease in the minimum throughput. (Fig. 10)

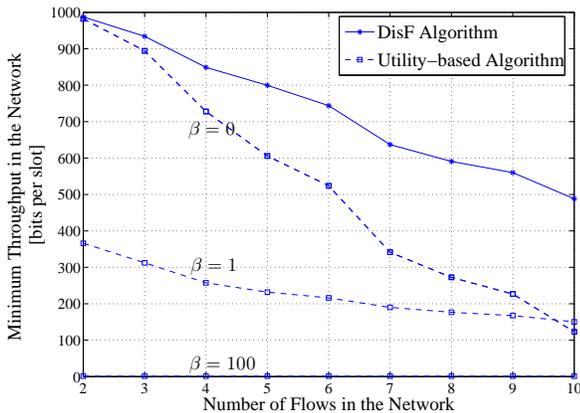


Fig. 10. DisF algorithm is compared with the utility-based approach when different number of flows are using the network. It is shown that increasing β is at the cost of decreasing the minimum throughput.

our proposed approach has a better performance in terms of fairness provisioning comparing to the class of utility-based approaches. Since the proposed algorithm determines the scheduling at each time slot, it can adapt to the dynamic changes of the wireless environment. In the future, we will use Lyapunov stability theory to determine the scheduling policy in wireless networks with network coding with the goal of fairness provisioning.

REFERENCES

- [1] J. Shi, O. Gurewitz, V. Mancuso, J. Camp, and E. W. Knightly, "Measurement and modeling of the origins of starvation in congestion controlled mesh networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [2] M. Andrews, A. F. Anta, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *Proc. of IEEE Infocom*, San Diego, CA, Mar. 2010.
- [3] Y. Li, M. Chiang, and A. R. Calderbank, "Congestion control in networks with delay sensitive traffic," in *Proc. of IEEE Globecom*, Washington, DC, Nov. 2007.
- [4] A. H. Mohsenian-Rad and V. W. S. Wong, "Cross-layer fair bandwidth sharing for multi-channel wireless mesh networks," *IEEE Trans. on Wireless Communications*, vol. 7, no. 9, pp. 3436–3445, Sept. 2008.

- [5] T. Voice and G. Raina, "Stability analysis of a max-min fair rate control protocol (RCP) in a small buffer regime," *IEEE Trans. on Automatic Control*, vol. 54, no. 8, pp. 1908–1913, Aug. 2009.
- [6] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross Layer Control in Wireless Networks*. Foundations and Trends in Networking, 2006.
- [7] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
- [8] C. P. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Trans. on Mobile Computing*, vol. 9, no. 4, pp. 527–539, Apr. 2010.
- [9] L. Lin, X. Lin, and N. B. Shroff, "Low-complexity and distributed energy minimization in multihop wireless networks," *IEEE/ACM Trans. on Networking*, vol. 18, no. 2, pp. 501–514, Apr. 2010.
- [10] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *Proc. of IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.
- [11] L. Chen, S. H. Low, J. C. Doyle, and M. Chiang, "Cross layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. of IEEE Infocom*, Barcelona, Spain, Apr. 2006.
- [12] L. Ying and S. Shakkottai, "Scheduling in mobile ad hoc networks with topology and channel-state uncertainty," in *Proc. of IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.
- [13] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. on Networking*, vol. 14, no. 2, pp. 302–315, Apr. 2006.
- [14] M. Andrews, L. Qian, and A. Stolyar, "Optimal utility based multi-user throughput allocation subject to throughput constraints," in *Proc. of IEEE Infocom*, Miami, FL, Mar. 2005.
- [15] J. Liu, A. Stolyar, M. Chiang, and H. V. Poor, "Queue back-pressure random access in multihop wireless networks: Optimality and stability," *IEEE Trans. on Information Theory*, vol. 55, no. 9, pp. 4087–4098, Sept. 2009.
- [16] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Trans. on Networking*, vol. 15, no. 6, pp. 1333–1344, Dec. 2007.
- [17] M. J. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. on Networking*, vol. 16, no. 2, pp. 396–409, Apr. 2008.
- [18] S. Sarkar and L. Tassiulas, "Fair distributed congestion control in multirate multicast networks," *IEEE/ACM Trans. on Networking*, vol. 13, no. 1, pp. 121–133, Feb. 2005.
- [19] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 163–173, Jan. 2005.
- [20] D. Quintas and V. Friderikos, "Energy efficient relaying within a cell: Multi path versus shortest path routing," in *Proc. of IEEE Infocom*, San Diego, CA, Mar. 2010.
- [21] S. Misra, G. Xue, and D. Yang, "Polynomial time approximations for multi-path routing with bandwidth and delay constraints," in *Proc. of IEEE Infocom*, Rio De Janeiro, Brazil, Apr. 2009.
- [22] J. G. Proakis, *Digital Communications*. 4th edition, New York: McGraw-Hill, 2001.
- [23] J. W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithms for rate-reliability based tradeoff in network utility maximization," *IEEE J. Select. Areas Commun.*, vol. 24, pp. 962–976, May 2006.
- [24] K. Ronasi, A. H. Mohsenian-Rad, V. W. S. Wong, S. Gopalakrishnan, and R. Schober, "Optimal data and channel code rate allocation in wireless networks with multi-path routing," in *Proc. of IEEE ICC*, Kyoto, Japan, June 2011.
- [25] —, "Optimal data transmission and channel code rate allocation in multipath wireless networks," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 8, pp. 3963–3974, Oct. 2011.
- [26] —, "Delay-throughput enhancement in wireless networks with multi-path routing and channel coding," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 3, pp. 1116–1123, Mar. 2011.
- [27] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.
- [28] S. Chen, Y. Fang, and Y. Xia, "Lexicographic maxmin fairness for data collection in wireless sensor networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 7, pp. 762–776, July 2007.

- [29] H. Shirani-Mehr, G. Caire, and M. J. Neely, "MIMO downlink scheduling with non-perfect channel state knowledge," *IEEE Trans. on Communications*, vol. 58, no. 7, pp. 2055–2066, July 2010.
- [30] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, "Back-pressure routing and rate control for ICNs," in *Proc. of ACM MobiCom*, Chicago, IL, Sept. 2010.
- [31] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks," in *Proc. of IEEE Infocom*, Phoenix, AZ, Apr. 2008.
- [32] S. Boyd, S. J. Kim, L. Vandenberghe, and A. Hassibi, *A Tutorial on Geometric Programming*. Springer Science and Business Media, 2007.
- [33] "MOSEK software." [Online]. Available: <http://www.mosek.com>
- [34] R. Jain, W. Hawe, and D. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report DEC-TR-301, Sept. 1984.



Keivan Ronasi (S05) was born in Isfahan, Iran in 1981. He received B.Sc. degree in electrical and computer engineering from Isfahan University of Technology, Isfahan, Iran in 2004. He received M.Sc. degree from University of Tehran, Tehran, Iran in 2007. Currently, he is a Ph.D. candidate in University of British Columbia, Vancouver, Canada. His research interests include modeling and design of reliable wireless communication networks, network coding, network optimization, and stochastic network optimization.



Vincent W.S. Wong (SM07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently an Associate Professor. His research areas include protocol design,

optimization, and resource management of communication networks, with applications to the Internet, wireless networks, smart grid, RFID systems, and intelligent transportation systems. Dr. Wong is an Associate Editor of the *IEEE Transactions on Communications* and *IEEE Transactions on Vehicular Technology*, and an Editor of *KICS/IEEE Journal of Communications and Networks*. He has served as the Symposium Co-chair of *IEEE Globecom'11, Wireless Communications Symposium*.



Sathish Gopalakrishnan is an Assistant Professor of Electrical and Computer Engineering at the University of British Columbia. His research interests center around resource allocation problems in several contexts including real-time, embedded systems and wireless networks. Prior to joining UBC in 2007, he obtained a PhD in Computer Science and an MS in Applied Mathematics from the University of Illinois at Urbana-Champaign. He has received awards for his work from the IEEE Industrial Electronics Society (Best Paper in the IEEE Transactions

on Industrial Informatics in 2008) and at the IEEE Real-Time Systems Symposium (in 2004).