

Appropriate Control of Wireless Networks with Flow Level Dynamics

Long Le and Ravi R. Mazumdar

Abstract— We consider the network control problem for wireless networks with flow level dynamics under the general k -hop interference model. In particular, we investigate the control problem in low load and high load regimes. In the low load regime, we show that the network can be stabilized by a regulated maximal scheduling policy considering flow level dynamics if the offered load satisfies a constraining bound condition. Because maximal matching is a general scheduling rule whose implementation is not specified, we propose a constant-time and distributed scheduling algorithm for a general k -hop interference model which can approximate the maximal scheduling policy within an arbitrarily small error. Under the stability condition, we show how to calculate transmission rates for different user classes such that the long-term (time average) network utility is maximized. Our results imply that congestion control is unnecessary when the offered load is low and optimal user rates can be determined to maximize users' long-term satisfaction. In the high load regime where the network can be unstable under the regulated maximal scheduling policy, we propose the cross-layer congestion control and scheduling algorithm which can stabilize the network under arbitrary network load. Through numerical analysis for some typical networks, we show that the proposed scheduling algorithm has much lower overhead than other existing queue-length-based constant-time scheduling schemes in the literature, and it achieves performance much better than the guaranteed bound. In addition, using congestion control in the low load condition results in much lower average utility compared to that due to the optimal transmission rate derived in the paper.

I. INTRODUCTION

Resource allocation in communication networks has been an active research topic for the last several years. While optimal rate control in wired networks can be achieved by a distributed algorithm, solving this problem in wireless networks is much more challenging. In fact, one of the most critical tasks in designing and engineering wireless networks is to schedule simultaneous transmissions of different wireless links in the network. The difficulty of the scheduling task comes from the combinatorial nature of the problem which renders it a very complex and challenging problem [1]. The scheduling problem is also complicated by the fact that interference relationship in the network varies with the physical layer technologies and it depends on numerous parameters of the wireless channel. One of the general interference models which is determined by a single parameter is the k -hop interference model. For this interference model, wireless links k or more hops away from one another can be scheduled to transmit data at the same time. In this paper, we develop a control framework for wireless networks under the general k -hop interference model.

There are several optimal and suboptimal scheduling schemes proposed for wireless networks in the literature. In

a seminal paper [2], Tassiulas and Ephremides proposed an optimal back-pressure policy which achieves the maximum network throughput. This scheduling policy is, however, centralized and computationally expensive. In [3], a randomized linear-complexity scheduling algorithm was proposed where a transmission schedule in time slot t was constructed by choosing the one with larger total weight between the schedule in time slot $t - 1$ and a newly-generated schedule in time slot t . This idea was used to develop distributed throughput-optimal scheduling policies in [4]–[6] for one-hop and two-hop interference models. Note that these scheduling algorithms achieve full utilization of wireless networks with respect to what remains in the data transmission phase only. Specifically, a large amount of bandwidth has been wasted to exchange control information in the schedule construction phase which would otherwise be used for data transmission. Recently, constant-time queue length-based scheduling policies were proposed for one and two hop interference models [9]–[11]. These scheduling algorithms only achieve a guaranteed fraction of the capacity region but they have constant amount of overhead. A more general maximal scheduling policy was considered in [11], [12] where several throughput performance bounds were investigated.

In practice, it is desired that each node only communicates with its immediate neighbors (e.g., those whose transmissions interfere with that of the underlying node) to construct a transmission schedule in each time slot. Also, scheduling algorithms should be able to work for a general class of interference models (e.g., k -hop interference model [1]). Another aspect which was ignored by most existing works in the literature is that no conflict-free schedule is available to exchange control information at the beginning of each time slot. Therefore, control information can only be exchanged by using contention-based transmissions which renders information exchange more than one hop away a time-consuming operation. Also, it is important to quantify amount of time/overhead used to construct the schedule and to develop explicit procedure to exchange control information in each time slot.

In this paper, we show the performance guarantee of the *regulated maximal scheduling* policy in wireless networks considering flow level dynamics. In fact, *regulated maximal scheduling* is simply the combination of the maximal scheduling policy [11], [12] and a traffic regulator implemented at each wireless link. Because regulated maximal scheduling is a general rule, we propose a constant-time and distributed algorithm to implement it in each time slot. We show that the proposed scheduling algorithm can approximate the regulated maximal scheduling policy within an arbitrarily small error. The proposed scheduling algorithm works for a general k -hop interference model and does not require queue length information. Moreover, we explicitly describe how wireless links coordinate their contentions to construct the schedule in each time slot.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. E-mail: longle@engmail.uwaterloo.ca, mazum@ece.uwaterloo.ca.

The results presented in this paper implies that we do not need to perform congestion control even with flow level dynamics if the traffic load lies within a region which can be stabilized by the underlying scheduling algorithm. This is an interesting finding given the fact that there are existing works which employ congestion control algorithms to stabilize the network [7], [8]. Given the fact that the scheduling algorithm is a randomized one and the number of users are dynamic, we are interested in finding transmission rate for each user class which achieves maximum long-term utility. In particular, we show that there exists optimal transmission rates for all user classes to achieve such maximum long-term utility.

The results in this paper have several important implications for system implementation. First, we do not need to perform congestion control in low network load even with flow level dynamics. Hence, communication overhead as well as implementation issues such as noisy [15] and asynchronous feedback [16] due to congestion control operations can be completely avoided. Second, the problem of network utility maximization can be decoupled from that of stabilizing the network. Specifically, the network can be stabilized by implementing traffic regulators at wireless links together with suitable scheduling mechanism. In fact, we show via numerical examples that using congestion control algorithm to stabilize the network in low load actually degrades the long-term network utility considerably.

The remaining of this paper is organized as follows. We describe the system models and performance bound in section II. Performance guarantee of the regulated maximal scheduling policy is presented in section III. In section IV, we present the distributed scheduling algorithm to approximate the maximal scheduling policy. We derive the optimal transmission rates to achieve long-term utility maximization in section V. Some numerical results are presented in section VI and section VII states the conclusion. For notational convenience, we will put elements of different measures into the corresponding vector. For example, the vector of transmission rates will be denoted by \vec{x} where x_s is its s -th element which is the transmission rate of class- s users.

II. SYSTEM MODELS AND PERFORMANCE BOUND

We model a wireless network as a directed graph $G = (V, E)$ where V is the set of wireless nodes and E is the set of wireless links. A wireless link from node i to node j exists if node j can correctly receive information transmitted by node i . In practice, existence of such a link depends on transmission power, path loss, fading, interference, desired bit error rate and other factors.

We assume that there are S classes of users each of which associates with a fixed routing path from a source node to a destination node. The user routes are stored in an incidence matrix $[H_s^k]$ where $H_s^k = 1$ if link k is on the route of class- s users and $H_s^k = 0$ otherwise. Users of class s arrive to the network with rate λ_s and each brings a file for transfer whose size is exponentially distributed with mean $1/\mu_s$. The offered load by class- k users is, therefore, $\rho_s = \lambda_s/\mu_s$. The vector of offered load will be denoted as $\vec{\rho} = [\rho_1, \rho_2, \dots, \rho_S]$. We assume that users of each class transmit at the same rate.

Interference constraints are modeled using a contention matrix $[C_{ij}]_{i,j \in E}$. Specifically, link i is said to interfere with link j if $C_{ij} = 1$ and $C_{ij} = 0$ otherwise. This general interference

relationship is only used to show the performance guarantee of the regulated maximal scheduling policy in sections II, III while the k -hop interference model will be assumed in all other subsequent sections. Time is divided into slots of unit duration. Link l can transmit at rate R_l if its interfering links are not scheduled to transmit in a same time slot. Unless otherwise stated, we will assume that interference relation is symmetric (i.e., link i interferes with link j if and only if link j interferes with link i). In the following, we provide some important definitions which will be used in the sequel (similar definitions can be found in [8], [12]).

Definition I: Interference set I_l of link l is the set of links which interfere with link l , i.e.,

$$I_l = \{k \in E : C_{kl} = 1\}. \quad (1)$$

Definition II: Interference degree $d_I(l)$ of link l is the maximum number of links in its interference set which do not interfere with each other.

Definition III: Interference degree $d_I(G)$ of graph G is the maximum interference degree of its links, i.e., $d_I(G) = \max_{l \in E} d_I(l)$.

Capacity region is defined to be the set of traffic load such that the network can be stabilized by some scheduling policy. In [2], capacity region for wireless networks is well characterized. In particular, capacity region is given by the set

$$\Omega = \left\{ \vec{\rho} : \left[\sum_{s=1}^S \frac{H_s^l \rho_s}{R_l} \right]_{l \in E} \in \text{Co}(\mathcal{R}) \right\} \quad (2)$$

where $\text{Co}(\mathcal{R})$ is the convex hull of all link schedules \mathcal{R} that satisfy the constraints imposed by our interference model. A scheduling policy is said to be throughput optimal if it stabilizes the network for all offered load within the capacity region Ω .

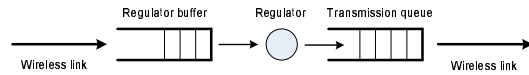


Fig. 1. Regulator implementation at each wireless link.

A new schedule is constructed in the first phase of each time slot which is used to transmit data in the second phase of the time slot. We further assume that traffic of each user class is regulated before entering a transmission queue for transfer over each wireless link. The employment of regulator was previously proposed by Humes to stabilize manufacturing systems [13] which have been shown to be unstable in some cases due to cycles of material flow [14]. Regulators were recently used in wireless networks [16], [11]. A λ -regulator associated with link l generates packets to transmission queue of link l with a maximum rate of λ . A regulator can be implemented as follows. In each time slot, a λ -regulator associated with link l checks the corresponding regulator queue. If the queue length is greater than link capacity R_l then it transfers R_l units of data to the transmission queue with probability λ/R_l . Otherwise, it transfers nothing. The regulator implementation is illustrated in Fig. 1.

In this paper, we assume a maximal scheduling policy which was proposed in [11], [12]. The maximal scheduling works as

follows. For any link $l \in E$ with transmission queue length larger than the link capacity in any time slot, we only require at least one link in its interference set I_l be scheduled. Specifically, if $Q_l/R_l \geq 1$ (where Q_l is the queue length of transmission queue for link l), we require

$$\sum_{k \in I_l} \pi_k \geq 1 \quad (3)$$

where $\pi_k = 1$ if link k is scheduled and $\pi_k = 0$ otherwise. Due to the combination of maximal scheduling and regulator implementation, the resulting scheduling will be called *regulated maximal scheduling* in the sequel. Note that maximal scheduling is a general scheduling rule without specific implementation. We will present a constant time and distributed scheduling algorithm which approximates the maximal scheduling in section IV. The following performance bound of the maximal scheduling policy was proved in [8], [12], it is restated here for completeness.

Lemma 2.1: For all traffic load $\vec{\rho}$ within the capacity region defined in (2), we have

$$\sum_{k \in I_l} \sum_{s=1}^S \frac{H_s^k \rho_s}{R_k} \leq d_l(l), \quad \forall l \in E. \quad (4)$$

This upper bound will be used to quantify the throughput guarantee of the regulated maximal scheduling policy in the next section.

III. PERFORMANCE OF THE REGULATED MAXIMAL MATCHING SCHEME

In this section, we show that the network is stable under the regulated maximal scheduling when the offered load satisfies a specific condition. We assume that a $(\rho_s + k\epsilon)$ -regulator is employed in k -th hop on the route of class- s users. It is worth to mention that the following stability result is similar in spirit to that in [11], although there is an important difference here. In fact, we capture flow dynamics in this paper while the authors in [11] only considered dynamics at the packet level. In [7] and [8], the authors captured flow dynamics but their stability results were achieved by a cross-layer congestion control algorithm. In this paper, network stability is achieved by a simple regulator implementation so communication overhead involved in the congestion control operation can be avoided. The stability result is stated in the following proposition.

Proposition 1: If the traffic load satisfies

$$\sum_{k \in I_l} \sum_{s=1}^S \frac{H_s^k \rho_s}{R_k} < 1, \quad \forall l \in E \quad (5)$$

then the network is stable under the regulated maximal scheduling policy. This condition will be called a constraining bound in the sequel.

Proof: We only provide the sketch for the proof here. Let $Q_l^s(t)$ and $Q_l(t)$ be transmission queue length for class s and total queue length for all user classes at link l in time slot t , respectively. Similarly, let $P_l^s(t)$ and $P_l(t)$ be regulator queue lengths for class s and for all user classes at link l in time slot

t , respectively. The proof is done by showing that an appropriately chosen Lyapunov function has negative drift when backlogs of regulator and transmission buffers become large enough. Specifically, we will choose the following Lyapunov function

$$V(\vec{P}, \vec{Q}) = V_1(\vec{Q}) + \xi V_2(\vec{P}, \vec{Q}) \quad (6)$$

where

$$\begin{aligned} V_1(\vec{Q}) &= \sum_l \frac{Q_l(t)}{R_l} \sum_{k \in I_l} \frac{Q_k(t)}{R_k} \\ V_2(\vec{P}, \vec{Q}) &= \sum_l \sum_s \left(P_{a_l^s}^s + Q_l^s \right)^2 \end{aligned}$$

Details of the drift analysis is available in [18]. ■

By comparing the upper and constraining bounds on capacity region in (4) and (5), respectively, and using the definition of $d_l(G)$, we can easily see that the regulated maximal scheduling policy achieves $1/d_l(G)$ capacity region.

IV. DISTRIBUTED SCHEDULING ALGORITHM

As mentioned before, maximal scheduling is a general rule whose implementation is not specified. In this section, we present a distributed scheduling algorithm which approximates the maximal scheduling policy in each time slot within an arbitrarily small error. In fact, the proposed algorithm will include the BP-SIM scheduling algorithm [10] proposed for the node exclusive (i.e., one-hop) interference model as a special case. Our proposed algorithm works with the general k -hop interference model. Also, in contrast to the existing queue-length-based scheduling algorithms [9], [10], in our algorithm each node with incident backlogged links does not require queue length information of other links in its neighborhood to construct the transmission schedule. In addition, the proposed algorithm is fully distributed and it has constant time overhead which does not grow with the network size. Our proposed algorithm is, therefore, much more flexible and general than existing ones in the literature. For ease of reference, we will refer to our scheduling algorithm as random approximate maximal matching (RAMM) scheduling in the sequel.

A. Algorithm Description

The RAMM algorithm is run in the first phase of each time slot. Specifically, we divide each time slot into two phases: a scheduling phase and a data transmission phase. The transmission schedule is constructed in the scheduling phase, and it is used to transmit data in the data transmission phase. The scheduling phase is further divided into K rounds each of which contains B mini-slots. In each round, new links are added to the current transmission schedule. The transmission schedule obtained at the end of the K -th round will be used to transmit data in the data transmission phase. In addition, only wireless links whose queue lengths are larger than the link capacity are scheduled by the algorithm in each time slot. The time slot structure of the RAMM algorithm is illustrated in Fig. 2.

Links are added to the schedule in each round through a matching request and matching acknowledgment message exchange as follows. At the beginning of each round, each *active* node (the notion of *active/inactive* nodes will be clarified shortly) decides to be *left* or *right* with probability $1/2$. Nodes

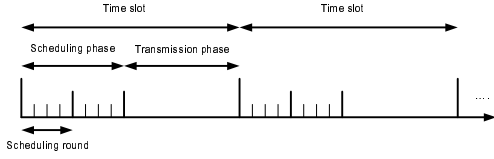


Fig. 2. Timing diagram of the RAMM scheduling algorithm.

becoming right wait to receive matching requests from their neighboring nodes. Backlogged links are added to the schedule in each round as follows. Each left node with at least one backlogged outgoing link (i.e., a link from this node to one of its neighbors) will choose a random backoff in $[1, B]$. When the backoff expires, a left node will choose one of its backlogged neighbors randomly to send a matching request if it has not heard any matching requests transmitted by other nodes so far in the round. A right node which receives a matching request will reply with a matching acknowledgment message and the corresponding link is added to the schedule. We assume that if two or more matching requests are transmitted in one mini-slot in the same neighborhood, collision occurs and no matching acknowledgment message is transmitted.

In each round, we require that if a link is added to the transmission schedule, all wireless links in its interference set be not added to the transmission schedule in subsequent rounds. This requirement guarantees that we will obtain a conflict-free transmission schedule at the end of the scheduling phase. It is observed that this requirement can be easily achieved by one-hop and two-hop interference models. Specifically, for the one-hop interference model after a link is added to the schedule, both its transmitting and receiving nodes will not transmit and reply any matching requests. For the two-hop interference model after a link is added to the schedule, all their one-hop neighboring nodes (i.e., one hop away) of both transmitting and receiving nodes will be aware of this (through hearing the matching request or matching acknowledgment) and will not transmit or reply any matching requests until the end of the scheduling phase.

For a general k -hop interference model with $k \geq 3$, we assume that a large enough power level is used to transmit matching request/acknowledgment messages in the scheduling phase so that if a link is added to the schedule in one round, all nodes within $k - 1$ hops from both the transmitting and receiving nodes of the link are aware of this so they will not transmit or reply any matching requests in subsequent rounds. Nodes within $k - 1$ hops from the transmitting and receiving nodes of any links in the schedule are called *inactive* nodes. All other nodes are *active* ones. Note that any *inactive* node will remain inactive until the end of the scheduling phase. In general, the number of nodes participating in the schedule construction process reduces rapidly over consecutive rounds. Because new links are kept added to the existing schedule in each round, the transmission schedule in the last round would approximate well the maximal schedule if B and K are large enough. We will show the performance guarantee of the proposed scheduling algorithm in the next subsection.

B. Analysis

Now, let degree d_i of node i be the number of nodes having links directly connecting to node i (i.e., one-hop neighbors of

node i). Let d^* be the maximum of d_i for all nodes in the network (i.e., $d^* = \max_{i \in V} d_i$). In addition, a matching request transmitted by one node may collide with those transmitted by other nodes. Let I_i be the number of nodes whose transmitted matching requests may collide with that of node i if node i and one or more of these nodes transmit simultaneously under the corresponding power level used in the scheduling phase. Let I be the maximum of I_i (i.e., $I = \max_{i \in V} I_i$). Also, let I_0^* be the maximum number of nodes which are at most $k - 1$ hops away from either A or B including A and B for any link AB in the network. We have the following result.

Proposition 2: For any $\mu \in (0, 1)$, we can choose the number of scheduling rounds K which depends only on B , d^* , I , I_0^* , and μ but independent of network size such that for any backlogged link l , the probability that at least one backlogged link in its interference set I_l is scheduled after K rounds is larger than or equal to μ .

Proof: The proof is available in [18]. ■

Proposition 3 has some similarities with lemma 3 of [10]. However, the proof is more demanding because the proposed algorithm works with a much more general interference model than that in [10] (i.e., node exclusive interference model was assumed in [10]). Now, using RAMM together with regulator implementation as described in section II, we have the following stability result.

Proposition 3: If the traffic load satisfies

$$\sum_{k \in I_l} \sum_{s=1}^S \frac{H_s^k \rho_s}{R_k} < \mu, \quad \forall l \in E \quad (7)$$

and under the condition stated in proposition 2, the network will be stable when RAMM algorithm is used together with the regulator implementation as described in section II.

Proof: The proof follows the same line with that of Proposition 1. However, the right hand side of the constraining bound becomes μ instead of one due to the performance bound achieved by RAMM scheduling scheme. ■

V. LONG-TERM UTILITY MAXIMIZATION UNDER LOW LOAD CONDITION

In section III, we have shown that the network is stable if the proposed scheduling algorithm is used and the traffic load satisfies the reduced constraining bound in (7). As a consequence of this result, it is clear that we do not need any congestion control algorithm as long as the traffic load in the network is low. Hence, communication overhead due to message exchange of the congestion control algorithm can be avoided if regulators are implemented in the network.

Proposition 3 means that when regulators are implemented and traffic load satisfies the condition stated in (7), the network is stable as long as user rates are bounded away from zero. Under this stability condition, it is natural to ask: how to choose user rates such that optimal long-term network utility can be achieved? Also, performing congestion control in this case is unnecessary because it may degrade the network performance in terms of long-term utility. Now, our objective is to maximize the long-term network utility which can be explicitly stated as

$$\max_{\vec{x}(t)} \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_{t=0}^{\tau} \left[\sum_{s=1}^S n_s^t(t) U_s(x_s(t)) \right] dt \quad (8)$$

where $n_s^t(t)$ and $x_s(t)$ are the number of class- s users transmitting in time slot t and their transmission rate, respectively; $U_s(x_s)$ is the utility function, which can, for example, reflect the level of satisfaction for class- s users. We assume that users arriving during time slot t can only transmit from time slot $t+1$ onward. Suppose that the queueing process at each source node is ergodic (this fact was justified in [17]). Let $f(\vec{n}^t, \vec{x})$ denote the joint probability density function of \vec{n}^t and \vec{x} in equilibrium. Because elements of \vec{n}^t are pairwise independent, we have $f(\vec{n}^t, \vec{x}) = \left[\prod_{s=1}^S f(n_s^t | \vec{x}) \right] f(\vec{x})$. Thus, we can rewrite (8) as

$$\max_{\vec{x}} \int_X \sum_{s=1}^S \left[\sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) f(n_s^t | \vec{x}) \right] f(\vec{x}) d\vec{x} \quad (9)$$

Let us define

$$\begin{aligned} g(\vec{x}) &= \sum_{s=1}^S \sum_{n_s^t=0}^{\infty} n_s^t U_s(x_s) f(n_s^t | \vec{x}) \\ &= \sum_{s=1}^S U_s(x_s) \sum_{n_s^t=0}^{\infty} n_s^t f(n_s^t | \vec{x}) \\ &= \sum_{s=1}^S U_s(x_s) E[N_s^t | \vec{x}] \\ &= \sum_{s=1}^S \frac{\rho_s}{x_s} U_s(x_s) \end{aligned}$$

where we have used Little's law in deriving $E[N_s^t | \vec{x}]$ in the above equation. Specifically, the expected waiting time for a class- s user is $1/(\mu_s x_s)$, using Little's law we have $E[N_s^t | \vec{x}] = \lambda_s / (\mu_s x_s) = \rho_s / x_s$. Thus, we can rewrite (9) as

$$\max_{\vec{x}} \int_X g(\vec{x}) f(\vec{x}) d\vec{x} \quad (10)$$

Now, suppose we wish to find optimal user rate $x_s \in [0, M_s]$. Let x_s^* be the maximum of $g_s(x_s) = \rho_s / x_s U_s(x_s)$ in $[0, M_s]$ and the corresponding optimum rate vector is \vec{x}^* . Then, it is easy to see that choosing $f(\vec{x}) = \delta(\vec{x} - \vec{x}^*)$ will maximize (10) where $\delta(\cdot)$ is the delta function. Thus, the long-term utility maximization can be achieved by allowing users of class s to transmit at the optimal rate x_s^* . In summary, when the traffic load satisfies (7), the network is stable under the proposed scheduling policy and no congestion control is needed. In addition, we can decouple the long-term utility maximization from stability under this stability condition.

Example: When the utility function is $U_s(x_s) = \ln(x_s)$ which corresponds to proportional fair rate allocation among users, we have $g_s(x_s) = \rho_s / x_s \ln(x_s)$. The global maximum of $g_s(x_s)$ is $x_s^* = e$. Thus, if $M_s > e$, the optimal transmission rate to achieve maximum long-term utility is $x_s^* = e$. We will compare long-term utility under this solution and for the case where cross-layer congestion control algorithm is used as in [8].

VI. NUMERICAL RESULTS

In this section, we show some illustrative numerical results for the proposed scheduling algorithm and the optimal transmission rate for long-term utility maximization. We consider

grid and random networks and a two-hop interference model. We assume that transmission rate on each wireless link equals $R_l = 10$ units/time slot, average length of each file brought by any user class is $1/\mu_s = 10$ units. Users of each class arrive according to Poisson process with arrival rate λ_s . We vary arrival rate to adjust the traffic load $\rho_s = \lambda_s / \mu_s$. Here, a unit of data is a block of information bits of suitable size. We assume all flows have the same load ρ in all the results.

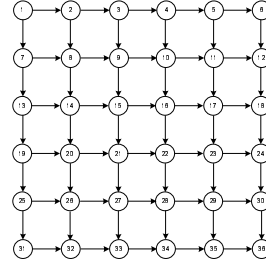


Fig. 3. Grid network of 36 nodes with 60/120 one-hop flows.

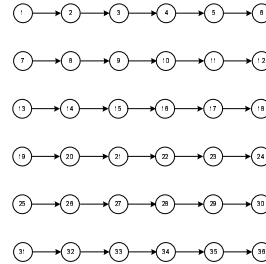


Fig. 4. Grid network of 36 nodes with 30 one-hop flows or 6 multihop flows.

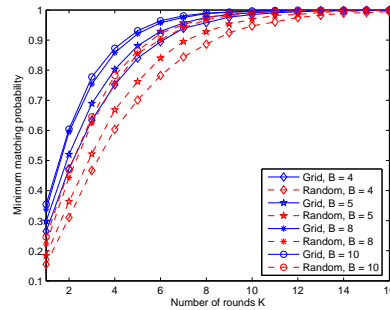


Fig. 5. Minimum probability of achieving the maximal scheduling versus number of rounds for a grid topology in Fig. 3 with 120 one-hop flows and a random topology with $d^* = 9$, $l = 24$, 198 one-hop flows under two-hop interference model.

A. Performance of Scheduling Algorithms

In Fig. 5, we show the minimum probability of achieving a maximal schedule due to the RAMM algorithm versus the number of rounds K under different maximum backoff values B for the grid topology (in Fig. 3 where there are two flows in two opposite directions on each unidirectional link) and the random topology. We assume that all the flows always have backlogs which is the worst case scenario. The probability of

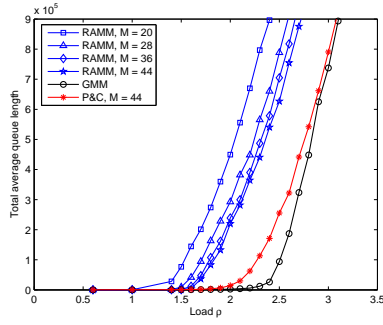


Fig. 6. Performance of RAMM scheduling scheme (for maximum backoff value $B = 4$, grid network with 36 nodes and 30 one-hop flows).

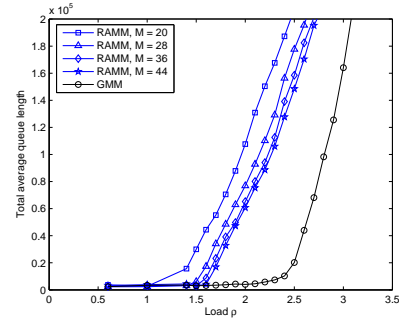


Fig. 8. Performance of different scheduling schemes (for $B = 4$, grid network with 36 nodes and 6 multihop flows).

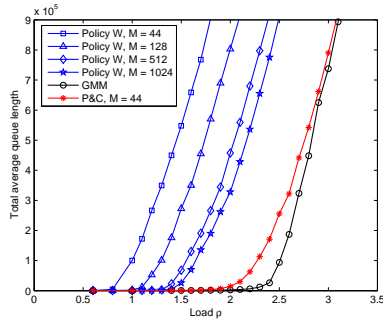


Fig. 7. Performance of policy W (for $M = 44, 128, 512, 1024$, grid network with 36 nodes and 30 one-hop flows).

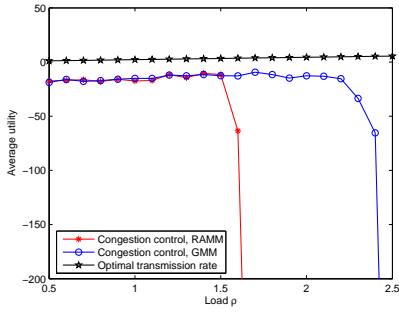


Fig. 9. Average utility of the proposed optimal transmission rate and with congestion control algorithm under two-hop interference model (for grid network with 36 nodes and 6 multihop flows, GMM and RAMM scheduling with $B = 4$, $K = 11$ in Fig. 4).

achieving a maximal schedule is the probability that a backlogged link is scheduled or at least one link in its interference set is scheduled. The minimum probability is the smallest probability among those of all the links. We have obtained this probability by averaging over 10^4 time slots. This figure shows that with $B = 4$, we only need $K = 11$ for the grid topology and $K = 13$ for the random topology to achieve minimum matching probability very close to 1. It is also shown that even the size of the node interference set is very large ($I = 22$ for the grid topology and $I = 24$ for the random topology), the improvement of minimum matching probability is very marginal when the maximum backoff B is larger than 8.

In Fig. 6, we compare performance of different scheduling schemes. Here, a grid network with 36 nodes and 30 one-hop flows as in Fig. 4 is considered. For the RAMM algorithm, we show the performance with different transmission rounds K while we fix the maximum backoff value of $B = 4$. We also present performance of Pick-and-Compare (P&C) [3], [5] and greedy maximal matching (GMM) [7] scheduling schemes. For the P&C scheme, a new schedule is generated in each time slot by using RAMM algorithm. Then the total weights of the old schedule and the newly generated schedule are compared where a link weight of a link is total size of regulator and transmission buffers of that link and the schedule with larger weight is chosen for transmission in the time slot. In fact, most of the complexity of the P&C scheme incurs in the “compare” step. For the GMM scheme, the schedule is constructed by adding one link with the largest weight to the schedule and removing all conflicting links with the added link in each step. This procedure is repeated

until no further link can be added to the schedule.

Note that both P&C and GMM scheduling schemes are either centralized or require huge overhead to implement in a distributed manner. Also, it is known that these two scheduling schemes achieve almost the capacity region. It is confirmed in this figure that P&C and GMM schemes achieve similar performance although GMM has a bit smaller total queue length for traffic load close to the boundary of the capacity region. It is evident that when we increase the number of rounds for the RAMM algorithm, we achieve better performance. Moreover, in the considered network $d_I(G) = 4$, so the performance guarantee stated in proposition 1 is just $1/4$ capacity region. However, the actual performance achieved by the proposed scheduling algorithm is much better than the performance bound as can be seen in Fig. 6.

In Fig. 7, we present the performance of the queue length based and constant time scheduling policy W for two hop interference model proposed in [9]. We show the performance of this scheduling scheme for different values of mini-slots used in the scheduling phase (the first phase of this scheduling policy corresponding to the first phase of our RAMM scheduling algorithm). As is evident, with $M = 44$ mini-slots, policy W achieves much lower performance than the RAMM algorithm. Even with $M = 1024$ mini-slots, performance of policy W is still below that of RAMM algorithm with $M = 44$ mini-slots. Note that due to practical implementation constraints, the time slot duration is usually limited to few milliseconds as in most wireless systems. If the duration of a minislot is $20 \mu s$ as in

the WLAN standard, $M = 100$ corresponds to 2 ms which is already quite large. Our scheduling algorithm, therefore, presents significant improvements compared to policy W because we cannot make the time slot interval arbitrarily large in practice.

In Fig. 8, we show performance of the RAMM scheduling algorithm for this grid network with 6 multihop flows. Similar performance to the single-hop case presented in Fig. 6 is observed for this setting. It is evident that although multihop flows contribute more traffic to the network because each flow traverses multiple links, scheduling algorithm is still the key to determine performance of the network.

B. Long-term Utility

We will compare the long-term utility under optimal transmission rates derived in section V and under the case where cross-layer congestion control is used. Specifically, we will consider the cross-layer congestion control algorithm proposed in [8] which is the extension of that for one-hop interference model in [7]. The cross-layer congestion control algorithm works as follows:

- Congestion price for each link l is updated as

$$q_l(t+1) = [q_l(t) + \alpha \Delta q_l(t)]^+ \quad (11)$$

where α is the step size and

$$\Delta q_l(t) = \sum_{k \in I(l)} \left[\sum_{s=1}^S H_s^k \int_t^{t+1} \frac{n_s(t) x_s(t)}{R_k} - 1_{k \in S(t)} \right]$$

where $S(t)$ denotes the sets of links belonging the schedule in time slot t , $1_{(\cdot)}$ is the indicator function.

- Transmission rate of class- k users is updated as

$$x_s(t+1) = \min \left\{ \frac{1}{\sum_{l \in E} q_l(t+1) \sum_{k \in I(l)} \frac{H_s^k}{R_k}}, M_s \right\}. \quad (12)$$

- Transmission scheduling: The network is scheduled in each time slot by the corresponding scheduling algorithms (greedy maximal matching or RAMM algorithms).

We consider the utility function $U_s(x_s) = \ln(x_s)$. Hence, for the proposed approach the optimal transmission rate for each user is $x_s^* = e$ (i.e., we assume that $M_s > e$). We illustrate performance of the cross-layer congestion control algorithm under both GMM and RAMM scheduling schemes. Long-term average utility is obtained by averaging the utility over 10^5 time slots. For the cross-layer congestion control algorithm, we fixed the transmission rate at $x_s^* = e$ for the first 10^3 time slots while still updating the congestion prices and generating a new schedule in each time slot. This initial period provides time for the congestion prices to converge to “better” values. The step size is initialized as $\alpha = 0.1$ and it is updated as $\alpha = \max \{ \alpha/t, 10^{-3} \}$.

It is evident that the proposed approach achieves higher average utility than those using congestion control algorithms for all traffic load. In fact, the average utility with congestion control decreases significantly when the traffic load is close to the boundary of the regions which can be stabilized by the underlying scheduling schemes. This observation confirms the argument that performing congestion control is unnecessary if the

network can be stabilized by the underlying regulated scheduling algorithms.

VII. CONCLUSIONS

We have presented a constant-time and distributed scheduling algorithm for a general k -hop interference model in this paper. The scheduling algorithm does not require queue length information and has overhead not growing with network size. With flow dynamics consideration, we have shown that the network can be stabilized by using a regulated maximal scheduling policy if the offered load satisfies the constraining bound. Our proposed scheduling algorithm achieves performance arbitrarily close to that of the regulated maximal scheduling. Under the stability condition, we have derived optimal transmission rates which achieve maximum long-term utility. Numerical results have shown that the proposed scheduling algorithm achieves much better performance than the existing constant-time scheduling algorithm and has much better performance than its performance guarantee. Also, performing congestion control under low load condition actually degrades performance in terms of long-term utility significantly compared to the optimal rate derived in the paper.

REFERENCES

- [1] G. Sharma, R. Mazumdar, and N. Shroff, “On the complexity of scheduling in multihop wireless systems,” *IEEE MOBICOM 2006*.
- [2] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
- [3] L. Tassiulas, “Linear complexity algorithms for maximum throughput in radio networks and input queued switches,” *IEEE INFOCOM 1998*.
- [4] E. Modiano, D. Shah, and G. Zussman, “Maximizing throughput in wireless networks via gossiping,” *ACM SIGMETRICS 2006*.
- [5] A. Eryilmaz, A. Ozdaglar, and E. Modiano, “Polynomial complexity algorithms for full utilization of multi-hop wireless networks,” *IEEE INFOCOM 2007*.
- [6] S. Sanghavi, L. Bui and R. Srikant, “Distributed link scheduling with constant overhead,” *ACM SIGMETRICS 2007*.
- [7] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on cross-layer congestion control in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 302–315, April 2006.
- [8] G. Sharma, R. Mazumdar, and N. Shroff, “Joint congestion control and distributed scheduling for throughput guarantees in wireless networks,” *IEEE INFOCOM 2007*.
- [9] C. Joo and N. B. Shroff, “Performance of random access scheduling schemes in multi-hop wireless networks,” *IEEE INFOCOM 2007*.
- [10] A. Gupta, X. Lin and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” *IEEE INFOCOM 2007*.
- [11] X. Wu, R. Srikant, and J. R. Perkins, “Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, June 2007.
- [12] P. Chaporkar, K. Kar, S. Sarkar, “Throughput guarantees through maximal scheduling in wireless networks,” *Allerton Conference on Communication, Control and Computing*, Sept. 2005.
- [13] C. Humes, Jr., “A regulator stabilization technique: Kumar-Seidman revisited,” *IEEE Transactions on Automatic Control*, vol. 39, no. 1, pp. 191–196, Jan. 1994.
- [14] P. R. Kumar and T. I. Seidman, “Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems,” *IEEE Transactions on Automatic Control*, vol. 35, no. 3, pp. 289–298, Mar. 1990.
- [15] J. Zhang, D. Zheng, and M. Chiang, “The impact of stochastic noisy feedback on distributed network utility maximization,” *IEEE INFOCOM 2007*.
- [16] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, “Joint asynchronous congestion control and distributed scheduling for multi-hop wireless networks,” *IEEE INFOCOM 2006*.
- [17] K. Ma, R. R. Mazumdar, and J. Luo, “On the performance of primal/dual schemes for congestion control in networks with dynamic flows,” *IEEE INFOCOM 2008*.
- [18] L. Le and R. R. Mazumdar, “Control of wireless networks with flow level dynamics under constant time scheduling,” under submission.