Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks

Xinzhou Wu, *Member, IEEE,* R. Srikant, *Fellow, IEEE,* and James R. Perkins, *Member, IEEE*

***Abstract*— We consider the problem of distributed scheduling in wireless networks subject to simple collision constraints. We define the efficiency of a distributed scheduling algorithm to be the largest number (fraction) such that the throughput under the distributed scheduling policy is at least equal to the efficiency multiplied by the maximum throughput achievable under a centralized policy. For a general interference model, we prove a lower bound on the efficiency of a distributed scheduling algorithm by first assuming that all the traffic only uses one-hop of the network. We also prove that the lower bound is tight in the sense that for any fraction larger than the lower bound, we can find a topology and an arrival rate vector within the fraction of the capacity region, such that the network is unstable under a greedy scheduling policy. We then extend our results to a more general multi-hop traffic scenario and show that similar scheduling efficiency results can be established by introducing prioritization or regulators to the basic greedy scheduling algorithm.**

***Index Terms*— Multi-hop wireless networks, scheduling, greedy algorithms, resource allocation**

I. Introduction and network model

Designing efficient distributed scheduling algorithms for wireless networks has attracted much attention recently. As compared to distributed scheduling algorithms, centralized scheduling schemes usually lead to a better performance at the cost of requiring a central authority to allocate the network resources. In a large-scale multi-hop wireless network, such a central authority does not always exist. Thus, a distributed scheduling algorithm would be very appealing even if it can only obtain a certain *fraction* of the *capacity region* of a centralized scheduling scheme. Throughout this paper, we refer to this *fraction* as the *scheduling efficiency* of a distributed scheduling scheme. We will define this in a more precise manner in the next section.

In this paper, we study the scheduling efficiency of a certain type of distributed scheduling policies based on the notion of *greedy scheduling* in wireless networks. By a greedy scheduling policy, we mean the following scheduling rule: *Each node attempts to independently schedule transmission over one of its backlogged links. If there is contention on the link, then it randomly picks another link and attempts to schedule a transmission on this link, and so on until it either finds a free link or has failed all possible attempts. Under such a policy, once all nodes have exhausted their attempts at finding a free link, a maximal number of links in the network will be scheduled.* We will describe the algorithm more precisely later. At this point, we note that such greedy schedules are a natural wireless counterpart of maximal matching scheduling for high-speed switches, which were considered in [21], [5]. It was shown there that such a scheduling policy has a stability region which is at least as large as half the capacity region achievable by any other scheduling policy. The main focus of research scheduling in a high-speed switch is the complexity of the scheduling policy. In the multi-hop wireless networks considered in this paper, both low complexity and distributed implementability are important, and the greedy scheduling scheme has both properties which makes it attractive.

In a recent work, it has been shown that maximal matching can also be used to achieve at least half the *capacity region* in wireless networks where the only constraint is that a node cannot simultaneously transmit or receive [12]. In other words, for such an interference model, a greedy scheduling scheme has a scheduling efficiency of at least $\frac{1}{2}$ as compared to the centralized counterpart. Such networks arise in scenarios where the spectrum is divided such that no collisions occur among neighboring nodes, such as in Bluetooth networks [15] or as an approximation to FH-CDMA networks used extensively in military applications for their anti-jamming and low-intercept properties [7]. Further, it is assumed in [12] that the load on a user is directly imposed on all links along its path, rather than packets travelling hop-by-hop through the network. It is well-known in queueing theory that the stability of such a model does not guarantee the stability of the corresponding model where packets traverse the network one link at a time [11], [13], [16]. In an earlier work [22], we proposed a scheduling policy called the *regulated* greedy scheduling policy which achieves a scheduling efficiency of $1/2$ in a multi-hop wireless network, where the assumption in [12] is removed at the cost of having to know the arrival rates for every source-destination pair. More recently, we have also shown that, if the capacity region is reduced to $1/3$ of the maximum capacity region, then even this knowledge is not required and that one achieves any point within $1/3$ of the capacity region by defining appropriate notions of fairness [2].

As explained in the previous paragraph, the interference model studied in [12], [22], [2] is only applicable when different channels (in frequency, time or code) are available for transmissions to different neighbors. When this condition is not satisfied, the interference model and the results derived are not valid anymore. One example is the 802.11-type of interference model, where a two-hop interference avoidance is enforced by communicating RTS/CTS messages within the network.

In this paper, we consider networks with arbitrary topology and a large class of interference models. Specifically, we consider a time-slotted wireless network which can be modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ where $\mathcal{V}$ is the vertex set and $\mathcal{L}$ is the set of all links. Further, we assume when link l is scheduled, it can transfer up to c_l packets in one time-slot. In other words, the capacity of

Xinzhou Wu is with Qualcomm Flarion Technologies, R. Srikant is with the Department of Electrical and Computer Engineering and Coordinated Science Lab, University of Illinois at Urbana-Champaign and James R. Perkins is with the Department of Manufacturing Engineering, Boston University. Emails: xin2xin@gmail.com, rsrikant@uiuc.edu, perkins@bu.edu. This work was done when Xinzhou Wu was a graduate student at the University of Illinois at Urbana-Champaign.

This work was supported by NSF grants ECS 04-01125 and CNS 05-19535.

link l is c_l. For the interference model, we still assume a simple collision model, i.e., a transmission over link l is successful if and only if there is no other transmission for any links within its *interference set*. The only assumption we make is that the interference relationship is symmetric, i.e., if link l interferes with link k, then k also interferes with l. It can be easily seen that both the model considered in [12] and the 802.11-type of model considered in [23], [3] are a special case of this more general model.

The main results of the paper are summarized below:

(1) We first assume that all the traffic traverse the network in only one-hop. We provide a sufficient condition on the arrival rates for the stability of the network under a distributed greedy scheduling algorithm. Specifically, we show that if the sum of rates (normalized by the capacity) over all links in any interference set is less than 1, the network is stable. This result can be also interpreted as a lower bound of $\frac{1}{\kappa_{max}}$ on the scheduling efficiency of the maximal-matching-based distributed greedy scheduling algorithm, where κ_{max} is the maximum number of non-interfering links in any interference set. This quantity is also called the *interference degree* of the graph in [3]. This lower bound is consistent with results in [12], [23] where specific interference models were studied. This was first observed in [3] where rate stability was proved. In this paper, we strengthen this result by establishing queue-length stability. Specifically, in the node-exclusive model [12], $\kappa_{max} = 2$ and thus greedy scheduling efficiency was shown to be $\frac{1}{2}$. In the 802.11-type interference model, κ_{max} can be further bounded by the maximum number of links (denoted by $N_{\mathcal{E}}$) in the first-hop neighborhood of any link and thus the scheduling efficiency was shown to be $\frac{1}{N_{\mathcal{E}}}$.

(2) We show that the lower bound $\frac{1}{\kappa_{max}}$ on the scheduling efficiency is tight in the sense that, given κ_{max} and $\epsilon > 0$, we can find a network topology which has at least one interference set with κ_{max} non-interfering links, a distributed greedy algorithm, and an arrival rate vector that lies inside $(1+\epsilon)/\kappa_{max}$ of the capacity region, such that the network is unstable under this set of arrival rates.

(3) We extend the above results to a general multi-hop setting by introducing some modifications to the greedy scheduling algorithm. Specifically, we propose a prioritized greedy scheduling algorithm and a regulated greedy scheduling algorithm and show that the network is stable under both policies for any arrival rate within $\frac{1}{\kappa_{max}}$ of the capacity region.

The above results improve upon the results presented in the original version of the paper [23]: A more general interference model and capacity model are used here, the lower bound on the scheduling efficiency has been improved, the new lower bound is shown to be tight in an appropriate sense and the stability of greedy scheduling in multi-hop networks is proved.

II. Definition of Scheduling Efficiency

In this section, we give a precise definition of the scheduling efficiency of distributed greedy scheduling policies. To do that, we first review the definition of capacity region under perfect scheduling. Assume that the route between each source-destination pair fixed. Let S denote the number of users (source-destination pairs) in the network. The routing information is captured through a *routing matrix* $\mathbf{H} = [H_l^s, l \in \mathcal{L}; 1 \le s \le S]$, where H_l^s is an indicator function and is determined by

$$H_l^s = \begin{cases} 1 & \text{if } l \in \mathcal{L} \text{ is on the path of user } s; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Each user s generates packets according to a random arrival process with mean arrival rate λ_s. The arrival processes of different users are assumed to be independent. A special case of this traffic model is the *single-hop traffic model* where all routes consist of only one link.

A scheduling policy is a rule to identify a set of links to be scheduled in each time slot which does not violate the interference constraints described earlier in this section. We let $\pi_l(n)$ denote the indicator function of the event that a transmission is scheduled over link $l \in \mathcal{L}$ at time n.

We define the capacity region to be the set of vectors of arrival rates $[\lambda_s]$ for which the queueing system is stable under some scheduling rule. Note that we use the notation $[x_l]$ to denote a vector whose elements are obtained by varying the index l over its possible set of values, i.e., $[x_l] = (x_1, x_2, \cdots)$. The stability region of a general wireless network has been well characterized in [20], and for the network model we described above, the stability region is given by

$$\Lambda = \{[\lambda_s, s = 1, 2, \cdots, S] : [\sum_{s=1}^{S} H_l^s \lambda_s, l \in \mathcal{L}] \in \mathcal{C}o(\mathcal{R})\}, \quad (2)$$

where $\mathcal{C}o(\mathcal{R})$ is the convex hull of the set of all feasible link schedules that are consistent with the interference constraint.

The performance of a given scheduling rule $\pi = [\pi_l, l \in \mathcal{L}]$ can be measured by the stability region C_π under this scheduling rule. A *throughput-optimal* scheduling rule [20] is a rule which stabilizes the queueing network under any set of user arrival rates that can be stabilized by another scheduling rule. A throughput-optimal scheduling rule is usually a *centralized* rule and requires a central authority to determine the scheduling for all links. The computational complexity of a throughput-optimal scheduler is also non-negligible. Thus it is difficult to implement it in multi-hop wireless networks.

For scheduling rules that are not throughput optimal, there exists a set of points inside the stability region (under a throughput-optimal scheduling) that cannot be supported by the scheduling rule. Thus, it is important to lower bound the achievable stability region of such schemes. For example, as mentioned in the introduction, greedy scheduling in a node exclusive model can achieve at least half the capacity region for any network topology. In view of this, we define the *scheduling efficiency* of a given scheduling policy as follows:

Definition 1: For a given scheduling rule π, the *scheduling efficiency* is the largest number $\alpha_\pi \in [0, 1]$ such that, for all rate vectors that lie strictly inside $\alpha_\pi \Lambda$, the system can be stabilized under π. Further, for any given $\epsilon > 0$, there exists an arrival vector inside $(\alpha_\pi + \epsilon)\Lambda$ such that the system cannot be stabilized by the scheduling rule. $\diamond$

By this definition, throughput-optimal schedules have a scheduling efficiency of 1. In the rest of this paper, we study the scheduling efficiency of greedy scheduling policies.

III. A LOWER BOUND ON THE SCHEDULING EFFICIENCY OF DISTRIBUTED GREEDY SCHEDULING WITH SINGLE-HOP TRAFFIC

In this section, we first establish a *lower bound* on the efficiency of greedy scheduling in a wireless network where only single-hop traffic exists. The single-hop assumption will be removed in Section V. There, some modifications to the simple greedy scheduling algorithm are introduced to accommodate the multi-hop nature of the traffic.

Now we focus on the case where only single-hop traffic exists in the network. Let Q_l be the queue length on link l, i.e., the number of packets backlogged on link l. We also use the notation $\mathbf{Q}(n)$ to denote the set of all queues in the system at time slot n.

Then, the greedy scheduling policy is defined as follows:

Algorithm 1: For any link $l \in \mathcal{L}$ with a non-empty queue exceeding the link capacity, i.e., $Q_l > c_l$, add the link to the schedule, if no links in its interference set $\mathcal{E}_l$ are scheduled. In other words, a schedule is said to be *greedy* if the following condition holds: if $\frac{Q_l}{c_l} \geq 1$, then

$$\sum_{k \in \mathcal{E}_l} \pi_k \geq 1. \tag{3}$$

One of the main results of this paper is stated in the next theorem.

Theorem 1: Let $A_l(n)$ and $D_l(n)$ be the number of packet arrivals and departures in slot n to link l. We assume that the arrival processes are stationary and let $\lambda_l = E(A_l(n))$. Further, we assume that $Cov(A_k(n), A_l(n)) = \sigma_{kl}^2 < \infty, \forall l, k \in \mathcal{L}$. For any set of distributions $P(\{D_l(n)\}|\mathbf{Q}(n))$, the Markov chain $\mathbf{Q}(n)$ is stable-in-the-mean, i.e.,

$$\limsup_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{L} E(Q_l(n)) < \infty,$$

if

$$\sum_{k \in \mathcal{E}_l} \frac{\lambda_k}{c_k} < 1, \qquad \forall l. \tag{4}$$

Further, if $P(\{D_l(n)\}|\mathbf{Q}(n))$ is independent of n, then $\mathbf{Q}(n)$ is a time-homogenous Markov chain and it is positive recurrent. $\diamond$

Remarks: This theorem directly leads to a lower bound on the capacity region under greedy scheduling policies. This is because the sufficient condition (4) for network stability puts a constraint on the total traffic load on any interference set. However, as observed in [3], the network cannot be stable in the arrival rate λ_k exceeds the capacity c_k for any link $k \in \mathcal{L}$. Thus, a necessary condition for network stability under *any* scheduling policy is

$$\sum_{k \in \mathcal{E}_l} \frac{\lambda_k}{c_k} < \kappa_{max}, \qquad \forall l,$$

where κ_{max} indicates the maximum number of non-interfering links in the interference set of any link in the network. The following corollary is thus immediate from Theorem 1.

Corollary 1: Consider a wireless network model with single-hop traffic. Under greedy scheduling as defined by (3), the system is stable for any set of arrival rates $[\lambda_s]$ that lies strictly inside $\frac{\Lambda}{\kappa_{max}}$, where Λ is the capacity region of the wireless network under any scheduling policy, as defined in (2). In other words, a greedy scheduling policy can achieve a stability region of $\frac{\Lambda}{\kappa_{max}}$. $\diamond$

Proof of Theorem 1: The proof below is presented only for the case where

$$A(n) := \{A_1(n), A_2(n), \ldots, A_L(n)\}$$

are i.i.d. across n. However, the arrival process may be dependent across links. The extension to more general Markovian arrival processes is straightforward. In the following Lyapunov analysis, one has to then consider the drift over multiple time slots; otherwise, the proof is similar.

Define the state of the system to be

$$\mathbf{Q}(n) := (Q_1(n), Q_2(n), \ldots, Q_L(n)),$$

where the dynamics of $Q_l(n)$ are given by

$$Q_l(n+1) = Q_l(n) + A_l(n) - D_l(n).$$

Assume that $\{D_l(n)\}$ is chosen according to some probability distribution given $Q(n)$, i.e., $P(\{D_l(n)\}|Q(n))$ is given. Thus, $Q(n)$ is a countable-state-space Markov chain. Note that $P(\{D_l(n)\}|Q(n))$ can be arbitrary. In particular, $\{D_l(n)\}$ could be any sequence of schedules consistent with the interference constraints. In a real-life network, the maximal schedule may by selected by a random access protocol. In this case, it may be reasonable to assume that the all the feasible maximal schedules are equally likely at each time instant. However, our analysis applies to more general models as well; in fact, it holds for any rule used to choose the set of active links, as long as the resulting schedule is a maximal schedule.

Define

$$V(n) = \sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{Q_k(n)}{c_k} \right). \tag{5}$$

The above Lyapunov function is very similar to the Lyapunov function considered in [5] to study maximal matching in high-speed switches. The Lyapunov function in [5] is of the form

$$\sum_l \frac{Q_l(n)}{c_l} \left(\frac{Q_l(n)}{c_l} + \sum_{k \in \mathcal{E}_l} \frac{Q_k(n)}{c_k} \right).$$

It has an additional Q_l/c_l term. This would result in a slightly weaker condition than the one we prove. Now

$$\begin{aligned}
&V(n+1) - V(n) = \\
&\textstyle\sum_l \frac{(Q_l(n+1)-Q_l(n))}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{Q_k(n)}{c_k} \right) \\
&+ \textstyle\sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(Q_k(n+1)-Q_k(n))}{c_k} \right) \\
&+ \textstyle\sum_l \frac{(Q_l(n+1)-Q_l(n))}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(Q_k(n+1)-Q_k(n))}{c_k} \right) \\
&= 2 \textstyle\sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(Q_k(n+1)-Q_k(n))}{c_k} \right) \\
&+ \textstyle\sum_l \frac{(Q_l(n+1)-Q_l(n))}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(Q_k(n+1)-Q_k(n))}{c_k} \right) \\
&= 2 \textstyle\sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} (a_k(n) - \pi_k(n)) \right) \\
&+ \textstyle\sum_l (a_l(n) - \pi_l(n)) \left(\sum_{k \in \mathcal{E}_l} (a_k(n) - \pi_k(n)) \right),
\end{aligned}$$

where $a_k(n) := A_k(n)/c_k$. In the above derivation, we have used the fact that

$$\sum_l \frac{(Q_l(n+1) - Q_l(n))}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{Q_k(n)}{c_k} \right) = \sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{(Q_k(n+1) - Q_k(n))}{c_k} \right).$$

Define $E_X(\cdot) = E(\cdot|X)$. Using the bounded second-moment assumption and the fact that the number of departures from each interference set is bounded, we get

$$\begin{aligned} & E_{Q(n)}\left(V(n+1) - V(n)\right) \\ \le & 2\sum_l \frac{Q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - \sum_{k\in\mathcal{E}_l} \pi_k(n) \right) + B \\ = & 2 \sum_{l:Q_l(n)>0} \frac{Q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - \sum_{k\in\mathcal{E}_l} \pi_k(n) \right) + B \\ \le & 2 \sum_{l:Q_l(n)>0} \frac{Q_l(n)}{c_l} \left(\sum_{k\in\mathcal{E}_l} \frac{\lambda_k(n)}{c_k} - 1 \right) + B_1 \\ \le & -2\epsilon \sum_{l:Q_l(n)>0} \frac{Q_l(n)}{c_l} + B_1, \end{aligned}$$

where $B, B_1 > 0$ are some constants and $\epsilon = 1 - \max_l \sum_{k\in\mathcal{E}_l} \frac{\lambda_k}{c_k}$.

The result above establishes that the drift of the Lyapunov function, i.e., the expected change in the value of the Lyapunov function, decreases when the queue lengths are sufficient far from the origin. From here, stability-in-the-mean follows from [10] and positive recurrence follows from Foster's theorem [1]. ⋄

Next, we comment on prior work on this problem. The policy that we consider in this paper is a natural extension of the maximal schedules considered for high-speed switches in [21], [5] and, for Bluetooth-like wireless networks, in [12]. Assuming $c_l = 1$ for all l, the following results were obtained in [23] and [3]:

- Define the usual graph describing the wireless network as follows: Each mobile is a node in the graph and there is an edge between two nodes if the nodes are within communication range of each other. A special case of the interference model described earlier whereby a link cannot transmit if another link its two-hop neighborhood transmits was considered in [23]. This model is motivated by the 802.11-type protocol where all nodes that hear a RTS or CTS remain silent to enable transmission between the nodes which generated the RTS or CTS. Given any link l, let $N_\mathcal{E}$ be the number of links within its first-hop neighborhood. In [23], it was shown that, under any maximal greedy policy, the queueing system is stable if the arrival rates lie inside $\frac{1}{N_\mathcal{E}}\Lambda$. In other words, the stability region under maximal greedy policies is at least $\frac{1}{N_\mathcal{E}}$ of the stability region under perfect scheduling policies.
- In [3], it was shown that if $\sum_{k\in\mathcal{E}_l} \lambda_k \le 1$ for all $l \in \mathcal{L}$, then the network is rate stable under any maximal greedy scheduling policy, i.e., the departure rate from each link is equal to the arrival rate into the link. If only rate stability is required, the result in [3] is a stronger result than the one in [23]. Note that rate stability does not imply queue length stability. To see this, suppose that the number of arrivals exceeds the number of departures in a time interval of duration T by an amount proportional to $\sqrt{T}$, then the difference between the arrival rate and the departure rate is proportional to $1/\sqrt{T}$ which goes to zero as $T \to \infty$. However, the total queue length at time T in the network is proportional to $\sqrt{T}$ which grows unbounded as $T \to \infty$. Thus, it is important to establish queue length stability. In this section, for the case where the link capacities are in general greater than or equal to 1, we proved that if $\sum_{k\in\mathcal{E}_l} \lambda_k/c_k < 1$, then the network is queue length stable. As in [23], we used a Lyapunov technique to prove this result. However, instead of using the Lyapunov function used in [23], we used a different function motivated by [5] to establish the result. This result was first reported in [24] and it was also obtained independently and simultaneously in [4].

IV. Tightness of the lower bound

In the previous section, we showed that the network is stable if the total arrival rate normalized by the link capacity in each interference set is less than one. This result can also be interpreted as a lower bound on the scheduling efficiency of greedy scheduling policies. In this section, we examine how tight this bound is. In particular, we show the following:

Theorem 2: Given a κ_{max}, which is defined to be the maximum number of non-interfering links of the interference set of any link in the network, and any $\epsilon > 0$, there exists a network $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ with the given κ_{max}, a link $l \in \mathcal{L}$ in the network such that

$$\sum_{k\in\mathcal{E}_l} \frac{\lambda_k}{c_k} = 1 + \epsilon \tag{6}$$

and a greedy maximal schedule such that the system is unstable under this scheduling policy. Further, if $(1+\epsilon)/\kappa_{max} \le 1$, then the set of arrival rates $\{\lambda_l\}$ lies in the capacity region Λ.

Proof: If $(1+\epsilon)/\kappa_{max} > 1$, then the result is trivial since this means that the arrival rates can be chosen to lie outside the stability region of even centralized schedules. Thus, we assume $(1+\epsilon)/\kappa_{max} \le 1$. Consider a network consisting of $N+1$ links with capacity 1 (i.e., $c_k = 1$ for all k in (6)): 1 through $N+1$ such that $\mathcal{E}_1 = \{1, 2, 3, \ldots, N+1\}$, $\mathcal{E}_i = \{1, i\}$ $\forall i \ne 1$. Thus, it is assumed that only link 1 interferes with link i, for any $i \ne 1$, while all other links interfere with link 1. One can visualize this $(N+1)$-link network as being arranged such that link 1 is in the middle of the network and all the other links as surrounding it in such a manner that they do not interfere with each other but interfere with link 1. According to the results presented earlier, this network is stable under maximal greedy scheduling if $\sum_{l\in\mathcal{L}} \lambda_l < 1$. Note the capacity region of this network is given

$$\lambda_1 + \lambda_i < 1, \forall i \ne 1.$$

In other words, there exists a schedule that stabilizes the system if the arrival rates lie within the capacity region.

Suppose that the arrival rates are such that

$$\sum_{l\in\mathcal{L}} \lambda_l = 1 + \epsilon.$$

Clearly this violates the sufficient condition for stability of greedy maximal schedules. For ease of exposition, we will first show that, if ϵ is sufficiently small, then there exist arrival rates that lie within the capacity region, but violate the stability condition for greedy maximal schedules, and provide a greedy schedule such that network is unstable. Then, we will establish a similar result for general values of ϵ. Suppose that the arrival rate to link 1 is $1-\epsilon$ and the arrival rate to each of the other links is $2\epsilon/N$. Then, this set of arrival rates lies in the capacity region since $1-\epsilon+2\epsilon/N<1$. Assume that the link arrival processes are Bernoulli processes, independent of each other. Due to the priority given to links 2 through N, the probability that one of these links is free is given by $1-2\epsilon/N$. Thus, the fraction of time available to serve link 1 is $(1-2\epsilon/N)^N$. Thus, the system will be unstable if

$$1-\epsilon > \left(1-\frac{2\epsilon}{N}\right)^N,$$

since $1-\epsilon$ is the arrival rate to link 1. We note that the above inequality is satisfied for sufficiently small $\epsilon>0$. To see this define

$$f(\epsilon) = 1-\epsilon-\left(1-\frac{2\epsilon}{N}\right)^N.$$

It is easy to verify that $f(0)=0$ and $f'(\epsilon)>0$ for sufficiently small ϵ. Thus, we have demonstrated a set of arrival rates which lie within the capacity region, but the system is unstable under these arrival rates for a particular choice of a greedy schedule.

Next, we extend the above argument for any feasible ϵ. Note that $\epsilon \le N-1$ due to the fact that $\lambda_1+\lambda_i<1\ \forall i\neq 1$. Let $\lambda_i = x/N\ \forall i \neq 1$ for some $x\in(0,N)$. To ensure that the arrival rates lie in the capacity region and to ensure instability in the queue at link 1, we require λ_1 to satisfy

$$\left(1-\frac{x}{N}\right)^N < \lambda_1 < 1-\frac{x}{N}.$$

Using the fact

$$\lambda_1 + N\lambda_i = 1+\epsilon,$$

it follows that

$$\left(1-\frac{x}{N}\right)^N < 1+\epsilon-x < 1-\frac{x}{N}.$$

The above inequalities can be rewritten as

$$g(x) := x+\left(1-\frac{x}{N}\right)^N - 1-\epsilon < 0; \qquad \text{and} \qquad x > \frac{N\epsilon}{N-1}.$$

If we find an $x\in(0,N)$ satisfying the above inequalities, then we will have established the tightness of the capacity loss bound. We first note that

$$g\left(\frac{\epsilon N}{N-1}\right) = \left(1-\frac{\epsilon}{N-1}\right)^N - \left(1-\frac{\epsilon}{N-1}\right) < 0.$$

Due to the continuity of $g(x)$, we can choose x larger than, but arbitrarily close to, $\epsilon N/(N-1)$ such that $g(x)<0$, thus proving the desired result. ◇

Remark: In [3], it has been shown that the above bound for throughput loss is tight for certain types of arrival processes and topologies. However, the argument requires that the arrival processes to the various links in the network are correlated. We briefly present the argument in [3] and contrast it with the argument in our paper. Consider the same $N+1$ network. In [3], it has been shown that one can find arrival rates that satisfy $\sum_l \lambda_l \ge 1$ and lie within the capacity region such that there exists a greedy schedule that renders the system unstable. Suppose that one packet arrives once every time slot to link i, $i\neq 1$ such that no two links $i\neq 1$, $j\neq 1$ have a packet arrival at the same time instant. For example, a packet could arrive to link $i\neq 1$ at time slots $Nk+(i-1)$, $k=0,1,2,\ldots$. Further, suppose that a packet arrives in each time slot with probability ϵ (where ϵ can be an arbitrarily small number) to link 1. Consider a greedy maximal schedule which gives priority to links 2 through $N+1$ whenever they have a packet to transmit. This greedy schedule would then alternate between all the links other than link 1 and link 1 would be starved. Thus, the queue length at link 1 would grow unbounded, leading to instability. On the other hand this arrival pattern is clearly stabilizable: Consider a policy which gives priority to link 1 whenever it has a packet to transmit. This policy stabilizes the system since one has a fraction $(1-\epsilon)$ of the time-slots during which 1 is not scheduled. During these time-slots, the other links can be scheduled simultaneously since they do not interfere with each other. Since their arrival rates are $1/N$, the queues at these links are stable. Thus, we have demonstrated a set of arrival rates for which the greedy scheduling policy results in a throughput reduction of $1/\kappa_{max}=1/N$, for any N. This demonstrates the tightness of the capacity loss bound if the arrivals to the links are allowed to be correlated.

The key idea in the correlated arrival case is to saturate the network by picking a schedule and a set of arrival rates such that the network is continuously working on draining the packet backlog in links 2 through $N+1$ so that there is no time left over to work on link 1. Thus, even a small arrival rate on link 1 makes the system unstable. On the other hand, we have seen that in the case of independent link arrival processes, instability is established by making the arrival rate at link 1 sufficiently large so that the time left over after serving links 2 through $N+1$ is insufficient to serve link 1.

V. Extension to multi-hop wireless networks

In this section, we extend the results in the previous sections to wireless networks shared by many *users*, each of whom has a *traffic flow* traversing the network through possibly multiple hops. Specifically, we assume there are S users each of whose traffic takes a fixed path through the network. The path information is contained by the routing matrix $\mathbf{H}=[H_l^s, 1\le l\le L; 1\le s\le S]$, where H_l^s is an indicator function and is determined as follows

$$H_l^s = \begin{cases} 1 & \textit{if } l\in\mathcal{L} \textit{ is on the path of user } s; \\ 0 & \textit{otherwise}, \end{cases} \tag{7}$$

where $\mathcal{L}=\{1,2,\ldots,L\}$. Further to indicate the fact that a path is a directed route, we define two functions $P^s(l)$ and $N^s(l)$ to be the previous and next link, respectively, from link l on the route of user s. Of course, the previous link and the next link are undefined for the first and last link, respectively, on a route. The arrival process $A_s(n)$ of each user s is a random process with mean arrival rate λ_s. Similar to the previous section, we assume that $A_s(n)$ is i.i.d. across time slots, but we also make the additional assumption that the arrival processes of different users are independent. Let L_m denote the maximum number of hops traversed by any user in the network. We also define q_l^s to be the queue length of user s on link l, and q_l without the superscript s denotes the total queue length at l, i.e., $q_l=\sum_s q_l^s H_l^s$.

In the following sections, we will introduce different modifications to the basic greedy scheduling algorithms to extend the single hop stability results to the multi-hop scenario.

A. Prioritized greedy scheduling

One way to ensure the stability of greedy scheduling in a multi hop wireless network is to introduce a priority structure to the scheduling algorithm. Specifically, we give higher priority to the traffic which has travelled fewer hops at each node. We will refer to such scheduling schemes as *prioritized greedy scheduling* schemes. The construction of this scheduling scheme is motivated by the proof techniques in [13].

To describe the extensions to multihop traffic, we introduce the following notation. For any link $l \in \mathcal{L}$, we let $Q_l^{(h)}$ be the queue length at link l contributed by those users for whom l is the h^{th} hop (in either direction) on their path. Similarly, $Q_l^{(\leq h)}$ is the queue length at link l of the users such that l is within the first h hops of their path, i.e.,

$$Q_l^{(\leq h)} = \sum_{h'=1}^{h} Q_l^{(h')}.$$

Now we propose the prioritized greedy scheduling algorithm.

Algorithm 2: (Prioritized greedy scheduling) At each time slot, we perform L_m rounds of greedy schedulings. In the first round, the greedy scheduling is based on the queues $\{Q_l^{(1)}\}$. More specifically, only nodes with first hop traffic which exceeds the corresponding channel capacity, i.e., source nodes of all traffic, are eligible to be a part of the greedy scheduling. For the second round of greedy scheduling, if a node in the first round has any incident links that are scheduled, then all of its incident links are removed from the graph. The second round of greedy scheduling is then implemented on the rest of the graph based on queues $\{Q_l^{(\leq 2)}\}$. Similarly, prior to the k^{th} round, if a node has any incident link that has been used in a greedy scheduling in one of the prior rounds, then all of its incident links are removed from the graph. Then, in the k^{th} ($k \leq L_m$) round, the greedy scheduling is performed on the remaining graph based on $\{Q_l^{(\leq k)}\}$. In all, up to L_m rounds of scheduling may have to be performed to implement prioritized greedy scheduling. ◇

We prove the following theorem in the appendix. It uses ideas from [12]; however, the traffic model in [12] is quite different, since they consider stability of file arrivals (user arrivals) and departures as opposed to packet arrivals and departures for a fixed number of users. Further, the idea of using multiple rounds of matchings in prioritized greedy scheduling is not explicit in [12] and our proof clarifies the role of these multiple rounds in ensuring stability.

Theorem 3: For a multi-hop wireless network, if the rate vector $\{\lambda_s; 1 \leq s \leq S\}$ satisfies

$$\sum_{k \in \mathcal{E}_l} \frac{\sum_s \lambda_s H_k^s}{c_k} < 1 \tag{8}$$

for every link l in the network, the network is queue length stable under prioritized greedy scheduling. In other words, a prioritized greedy scheduling policy can achieve a stability region of Λ/κ_{max}.

Proof: See Appendix. ◇

Remark: To prove this theorem, we perform an induction on the number of hops in the network. Such an induction was originally introduced in [13] and is used in [12] to prove the stability of the so-called imperfect scheduling algorithms. The most important property of this prioritized construction of greedy scheduling is that the traffic due to packets which have travelled a larger number of hops are transparent to the lower-hop traffic. Hence, we can prove the stability of the lower-hop traffic queueing system first and then prove the stability of higher-hop traffic queues using induction.

There are significant difficulties in implementing the prioritized greedy scheduling algorithm. In a large random ad hoc network, the number of hops on a typical user's route scales approximately as $O(\log N)$, where N denotes the number of nodes in the network [6]. (In the worst-case, it can be $O(N)$ since every node in the network can be on the same route between a source-destination pair.) Hence, as the network size increases, the overhead of this scheduling algorithm also increases. In addition, network-wide coordination is required to ensure that nodes whose backlogged packets have all traversed at least one hop do not request transmissions on their links before a greedy scheduling is found among those nodes which have backlogged packets which are on their first hop. Similarly, nodes with backlogged packets which have already traversed two or more hops have to wait for nodes which have packets that have traversed fewer hops to complete their maximal schedules. Thus, global synchronization is required to implement this algorithm, which is not desirable in a fully distributed network.

B. Regulated greedy scheduling

Another way to achieve queue stability in a multi-hop network system is to introduce *regulators* in the system. We have investigated such problems in our previous work [22], [2] for networks with a simple interference constraint. In this section, we show that the same idea is also applicable to the general model considered in this paper, and prove that the sufficient condition for single-hop stability case essentially remains the same even with multi-hop routes.

A regulator is introduced, for each flow using a link, such that the burstiness of the packets belonging to each user is regulated before entry into the node. A λ-regulator associated with link l is a logical device with a maximum service rate λ, i.e., it generates packets for the node at its output at a maximum rate of λ. Specifically, assuming c_l to be the capacity of link l, at each time slot, a λ-regulator associated with link l checks its buffer size and if it exceeds link capacity c_l, it transfers c_l bits to the user's queue with probability $\frac{\lambda}{c_l}$. Otherwise, it transfers nothing. We use the notation $R_l^s(n)$ to denote the number of departing packets from the regulator of user s on link l. The idea of a regulator was originally suggested in the context of re-entrant manufacturing lines in [8].

Denote the arrival rate vector consisting of the arrival rates of all the users by $[\lambda_s]_{1 \leq s \leq S}$. We choose the regulators of each user s according to the following rule: for the first hop (node) along the path of user s, we use a λ_s-regulator at its input queue for user s; for k^{th} hop ($k \geq 2$), we use a $(\lambda_s + (k-1)\epsilon)$-regulator.

We define the combination of the regulators and maximal greedy scheduling to be a regulated greedy scheduling algorithm. The main result of this section is given below:

Theorem 4: For a multi-hop wireless network, if the rate vector $\{\lambda_s; 1 \leq s \leq S\}$ satisfies

$$\sum_{k \in \mathcal{E}_l} \frac{\sum_s \lambda_s H_k^s}{c_k} < 1 \tag{9}$$

for any link l in the network, the network is queue length stable under regulated greedy scheduling.

Proof: For clarity of the presentation of the proof, first we introduce some notation. As before, let $D_l(n)$ be the number of departures from link $1 \le l \le L$ in time slot n. Further, $D_l^s(n)$ is the number of departures of user s packets over link l in time slot n, i.e.,

$$D_l(n) = \sum_s H_l^s D_l^s(n).$$

Let p_l^s denote the length of the regulator queue on link l for user s. It is easy to see that the whole system $(\mathbf{q}(n), \mathbf{p}(n))$ is a Markov Chain. We define the following Lyapunov function for the system:

$$V(\mathbf{q}, \mathbf{p}) = V_1(\mathbf{q}) + \xi V_2(\mathbf{p}, \mathbf{q}), \tag{10}$$

where $V_1(\mathbf{q})$ is a natural modification of the Lyapunov function in (5). Specifically, due to fact that we are considering multiple flows in each link, we define

$$V_1(\mathbf{q}) = \sum_l \frac{\sum_s q_l^s H_l^s}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{\sum_s q_k^s H_k^s}{c_k} \right).$$

On the other hand, V_2 is defined as follows

$$V_2(\mathbf{p}, \mathbf{q}) = \frac{1}{2} \sum_{l \in \mathcal{L}} \sum_s (p_{N^s(l)}^s + q_l^s)^2.$$

ξ is a positive parameter.

The queue update equations for this system are

$$\begin{aligned} q_l^s(n+1) &= q_l^s(n) - D_l^s(n) + R_l^s(n) & (11)\\ p_l^s(n+1) &= (p_l^s(n) - R_l^s(n))^+ + D_{P^s(l)}^s(n), & (12) \end{aligned}$$

where $R_l^s(n)$ is the output of the regulator that immediately precedes link l on the path of user s. When link l is the first link on the path of user s, then $D_{P^s(l)} = A_s(n)$. We also use the notation $r_l^s(n)$ to denote the departure normalized by the associated link capacity c_l, i.e.,

$$r_l^s(n) = \frac{R_l^s(n)}{c_l}.$$

Thus, $r_l^s(n)$ is either 0 or 1. Due to our definition of the regulator, R_l^s can only be non-zero when p_l^s exceeds c_l. Hence, we can remove the projection in (12) as well and we have

$$p_l^s(n+1) = p_l^s(n) - R_l^s(n) + D_{P^s(l)}^s(n).$$

We have to upper bound

$$E\left[V(\mathbf{q}(n+1), \mathbf{p}(n+1)) - V(\mathbf{q}(n), \mathbf{p}(n)) | \mathbf{q}(n), \mathbf{p}(n)\right]$$

by a negative number for all states $(\mathbf{q}, \mathbf{p})(n)$ except possibly in a bounded region, where the drift should simply be finite.

We consider the contribution of $V_1(\mathbf{q})$ and $V_2(\mathbf{p}, \mathbf{q})$ separately for now. Thus, we first look at

$$\begin{aligned} \Delta V_1(\mathbf{q}) &= V_1(\mathbf{q}(n+1)) - V_1(\mathbf{q}(n)) \\ &= 2\sum_l \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(q_k(n+1) - q_k(n))}{c_k} \right) \\ &+ \sum_l \frac{(q_l(n+1) - q_l(n))}{c_l} \left(\sum_{k \in \mathcal{E}_l} \frac{(q_k(n+1) - q_k(n))}{c_k} \right) \\ &= 2\sum_l \frac{q_l(n)}{c_l} \left(\sum_{k \in \mathcal{E}_l} \sum_s (r_k^s(n) - d_k^s(n)) \right) \\ &+ \sum_l \sum_s (r_l^s(n) - d_l^s(n)) \left(\sum_{k \in \mathcal{E}_l} \sum_s (r_k^s(n) - d_k^s(n)) \right). \end{aligned}$$

Similar to the proof in Theorem 1, the second term can be bounded by a constant. Further, given the fact that $E_{Q(n),P(n)}[r_k^s(n)] \le \frac{\lambda_s + L_m \epsilon}{c_k}$, (due to the definition of the regulators), we have

$$E_{Q(n),P(n)}\left(V_1(n+1) - V_1(n)\right) \le -2\eta \sum_{1 \le l \le L} \frac{q_l(n)}{c_l} + B,$$

Next, consider the contribution of V_2 to the drift. Note that

$$\begin{aligned} q_l^s(n+1) + p_{N^s(l)}^s(n+1) &= q_l^s(n) + p_{N^s(l)}^s(n) \\ &\quad + R_l^s(n) - R_{N^s(l)}^s(n). \end{aligned} \tag{13}$$

We can bound the drift due to V_2 as follows:

$$\begin{aligned} \Delta V_2(\mathbf{p}, \mathbf{q}) &= E\left[V_2(\mathbf{p}(n+1), \mathbf{q}(n+1))\right. \\ &\quad \left. - V_2(\mathbf{p}(n), \mathbf{q}(n)) | \mathbf{q}(n), \mathbf{p}(n)\right] \\ &= E\left[\sum_{1 \le l \le L} \sum_s (p_{N^s(l)}^s(n+1) + q_l^s(n+1))^2 \right. \\ &\quad \left. - \sum_{1 \le l \le L} \sum_s (p_{N^s(l)}^s(n) + q_l^s(n))^2 | \mathbf{q}(n), \mathbf{p}(n) \right] \\ &= E\left[\sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n) + q_l^s(n)\right) \left(p_{N^s(l)}^s(n+1) \right.\right. \\ &\quad \left.\left. + q_l^s(n+1) - p_{N^s(l)}^s(n) - q_l^s(n)\right) | \mathbf{q}(n), \mathbf{p}(n) \right] \\ &\quad + \frac{1}{2} E\left[\sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n+1) + q_l^s(n+1)\right.\right. \\ &\quad \left.\left. - p_{N^s(l)}^s(n) - q_l^s(n)\right)^2 | \mathbf{q}(n), \mathbf{p}(n) \right] \end{aligned}$$

The second term above can be bounded by a constant C_3 independent of $\mathbf{p}(n)$ and $\mathbf{q}(n)$ as follows:

$$\begin{aligned} & E\left[\sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n+1) + q_l^s(n+1)\right.\right. \\ &\quad \left.\left. - p_{N^s(l)}^s(n) - q_l^s(n)\ \right)^2 \Big| \mathbf{q}(n), \mathbf{p}(n) \right] \\ &\le 2 \sum_{1 \le l \le L} \sum_s ((\lambda_s + L_m \epsilon))^2 + ((\lambda_s + L_m \epsilon))^2 = C_3. \end{aligned} \tag{14}$$

The first term can be bounded by

$$\begin{aligned} & E\left[\sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n) + q_l^s(n)\right) \left(p_{N^s(l)}^s(n+1)\right.\right. \\ &\quad \left.\left. + q_l^s(n+1) - p_{N^s(l)}^s(n) - q_l^s(n)\right) | \mathbf{q}(n), \mathbf{p}(n) \right] \\ =& \sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n) + q_l^s(n)\right) \\ &\times E\left[R_l^s(n)\ - R_{N^s(l)}^s(n) | \mathbf{q}(n), \mathbf{p}(n)\right] \\ =& \sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n) + q_l^s(n)\right) \\ &\times \left[\bar{R}_l^s I_{p_l^s(n) \ge c_l}\ - \bar{R}_{N^s(l)}^s I_{p_{N^s(l)}^s(n) \ge c_{N^s(l)}}\right] \\ \le& \sum_{1 \le l \le L} \sum_s \left(p_{N^s(l)}^s(n) + q_l^s(n)\right) \\ &\times \left[\bar{R}_l^s\ - \bar{R}_{N^s(l)}^s I_{p_{N^s(l)}^s(n) \ge c_{N^s(l)}}\right], \end{aligned}$$

where $\bar{R}_l^s$ is the average departure rate of regulator R_l^s, if p_l^s exceeds c_l. From our design of the regulators, we know that

$$\bar{R}^s_{N^s(l)} = \bar{R}^s_l + \epsilon,$$

for any l on the path of user s. From this, we have

$$\Delta V_2(\mathbf{p},\mathbf{q}) \leq \sum_{1\leq l\leq L}\sum_{s}\left(p^s_{N^s(l)}(n)+q^s_l(n)\right) \times \left(\bar{R}^s_l - \bar{R}^s_{N^s(l)} I_{p^s_{N^s(l)}(n)\geq c_{N^s(l)}}\right) + C_2. \qquad (15)$$

By combining (14) and (15), we have

$$\begin{aligned}
&E\left[V(\mathbf{q}(n+1),\mathbf{p}(n+1)) - V(\mathbf{q}(n),\mathbf{p}(n))|\mathbf{q}(n),\mathbf{p}(n)\right]\\
\leq\ & -\eta\sum_{1\leq l\leq L}\frac{q_l(n)}{c_l} + \xi\sum_{1\leq l\leq L}\sum_s\left(p^s_{N^s(l)}(n)+q^s_l(n)\right)\\
&\times\left(\bar{R}^s_l - \bar{R}^s_{N^s(l)} I_{p^s_{N^s(l)}(n)\geq c_{N^s(l)}}\right) + C_3\\
=&-\sum_{1\leq l\leq L}\sum_s\left[\frac{\eta}{c_l} - \xi\left(\bar{R}^s_l - \bar{R}^s_{N^s(l)} I_{p^s_{N^s(l)}(n)\geq c_{N^s(l)}}\right)\right]q^s_l(n)\\
&+\xi\sum_{1\leq l\leq L}\sum_s p^s_{N^s(l)}(n)\left(\bar{R}^s_l - \right.\\
&\left.\bar{R}^s_{N^s(l)} I_{p^s_{N^s(l)}(n)\geq c_{N^s(l)}}\right) + C_3\\
\leq\ & -\sum_{1\leq l\leq L}\sum_s\left[\frac{\eta}{c_l}-\xi c_l\right]q^s_l(n)\\
&+\xi\sum_{1\leq l\leq L}\sum_s p^s_{N^s(l)}(n)\left(\bar{R}^s_l - \bar{R}^s_{N^s(l)} I_{p^s_{N^s(l)}(n)\geq c_{N^s(l)}}\right)\\
&+C_3\\
=\ & -\sum_{1\leq l\leq L}\sum_s\left[\frac{\eta}{c_l}-\xi c_l\right]q^s_l(n) - \epsilon\xi\sum_{1\leq l\leq L}\sum_s p^s_{N^s(l)}(n)\\
&+C_4,
\end{aligned}$$

where C_4 is another constant. We can easily choose ξ (independent of $\mathbf{p}(n)$ and $\mathbf{q}(n)$, of course) here such that

$$\frac{\eta}{c_l} - \xi c_l \geq C_0 > 0$$

for any l (we assume $c_l > 0$ for any l; links with zero capacity can be removed from the network without affecting the capacity region), and thus

$$\begin{aligned}
&E\left[V(\mathbf{q}(n+1),\mathbf{p}(n+1)) - V(\mathbf{q}(n),\mathbf{p}(n))|\mathbf{q}(n),\mathbf{p}(n)\right]\\
\leq\ & -\sum_{1\leq l\leq L}\sum_s C_0 q^s_l(n) - \epsilon\xi\sum_{1\leq l\leq L}\sum_s p^s_l(n) + C_4.
\end{aligned}$$

This concludes the proof of this theorem. ◇

VI. Resource Allocation

In the previous section, we assumed that the arrival rate satisfies the sufficient condition for stability given by (8). In this section, we discuss how to ensure that the arrival rates can be controlled to satisfy the sufficient condition. Associate a utility function $U_s(\lambda_s)$ with each user s, where $U_s(\cdot)$ is an increasing, twice differentiable concave function. Suppose that the $\{\lambda_s\}$ are chosen to solve the following optimization problem:

$$\max_{\{\lambda_s\geq 0\}}\sum_s U_s(\lambda_s) \qquad (16)$$

subject to

$$\sum_{k\in\mathcal{E}_l}\sum_{s:H^s_k=1}\frac{\lambda_s}{c_k}\leq 1-(L_m+1)\epsilon, \qquad \forall l,$$

where ϵ is chosen to be sufficiently small. This problem can be solved in a distributed manner to obtain the $\{\lambda_s\}$; see [19] for a general overview of congestion control and [17], [25], [2] for an application to wireless networks with a special interference constraint. Here we will briefly present how such a congestion-control algorithm can be implemented for our model. Each source chooses its arrival rate at each time instant according to the following algorithm which depends on the congestion price $\tilde{p}_l(n)$ generated by each link:

$$U'(\lambda_s(n)) = \sum_{l:H^s_l=1}\tilde{p}_l(n), \qquad (17)$$

and each link implements the following algorithm to compute the congestion price $\tilde{p}_l(n)$:

$$\tilde{p}_l(n+1)=\left[\tilde{p}_l(n)+\delta\left(\sum_{k\in\mathcal{E}_l}\sum_{s:H^s_k=1}\frac{\lambda_s}{c_k}-1-(L_m+1)\epsilon\right)\right]^+.$$

Using techniques in [9], [14], [26], [19], this algorithm can be shown to converge to the optimal solution of (16) if δ is chosen to be sufficiently small. We assume here that the congestion information is available instantaneously to each source; algorithms and analysis for more general delay models can be obtained by appropriately modifying the results in [2]. Now, these arrival rates, can be used to design the regulator parameters, which along with greedy maximal scheduling ensures the stability of the network. For an alternative solution to the resource allocation problem for the special case of max-min fairness, we refer the reader to [18].

VII. Conclusions

In this paper, we consider the capacity region achievable under greedy scheduling in a general interference model. We define the ratio between the achievable capacity region under greedy scheduling and throughput optimal scheduling to be the greedy scheduling efficiency. A lower bound on the greedy scheduling efficiency is established, which is inversely proportional to the maximum number of simultaneously schedulable links within the interference set of any link in the network. We further show that the lower bound is tight.

Although the above results are shown in a single-hop traffic scenario, where all users only choose one of their direct neighbors as the destination of the traffic, we have extended the results to the more general multi-hop traffic scenario by introducing more sophisticated greedy scheduling policies, specifically, prioritized greedy scheduling and regulated greedy scheduling. These can be implemented in a distributed fashion; however, regulated greedy scheduling has less overhead associated with it.

Appendix

As in [12], we prove Theorem 3 by using an induction on the longest path L for any user in the graph $\mathcal{G}$. When $L=1$, all the users have one-hop traffic. In this case, we can apply Theorem 1 directly and the stability of $\mathbf{Q}$ under greedy scheduling follows.

Assume Theorem 3 is true when $L=k$. To complete the induction, we need to show that for $L=k+1$, the system is still stable if the arrival rate vector satisfies (8).

We separate the queue $Q_l(n)$ into two terms:

$$Q_l(n) = Q_l^{(\leq k)}(n) + Q_l^{(k+1)}(n). \quad (18)$$

Further, define $x_s^{(l)}(n)$ $(l = 1, 2, \dots, k+1)$ to be the number of packets normalized by the link capacity c_l arriving the lth hop on user s's path. It can be easily seen that

$$x_s^{(1)}(n) = \frac{A_s(n)}{c_1}.$$

The queue update rule can be rewritten as

$$\frac{Q_l(n+1)}{c_l} = \frac{Q_l(n)}{c_l} - \frac{D_l(n)}{c_l} + \sum_{h=1}^{k+1}\sum_s x_s^{(l)}(n) H_l^{sh}, \quad (19)$$

where H_l^{sn} is the indicator function that link l is the nth hop of user s.

Due to the structure of prioritized greedy scheduling, it can be easily seen that $[Q_l^{(\leq k)}]$ is equivalent to a queueing system with traffic that traverses at most k hops. Actually, if we cut off the last hop of all $k+1$-hop users in our network (19) and make the kth hop as the destination for the corresponding user, we obtain a k-hop network whose queueing system evolves exactly in the same manner as $[Q_l^{(\leq k)}]$, under the greedy scheduling rule we defined in this paper.

Since, for any link l, the arrival rate vector $[\lambda_s]$ satisfies

$$\sum_{m\in\mathcal{E}_l} \frac{\sum_s H_m^s \lambda_s}{c_m} = \sum_{m\in\mathcal{E}_l} \frac{\sum_s \sum_{h=1}^{k+1} H_m^{sh}\lambda_s}{c_m} < 1,$$

we must have

$$\sum_{m\in\mathcal{E}_l} \frac{\sum_s \sum_{h=1}^{k} H_m^{sh}\lambda_s}{c_m} < 1.$$

In other words, $[\lambda_s]$ is also within the stability region of the k-hop network $\mathbf{Q}^{(\leq k)}$ under distributed greedy scheduling. Thus, from the induction procedure, we know that $[Q_l^{(\leq k)}]$ must be stable, or equivalently, the Markov chain $\mathbf{Q}^{(\leq k)}$ is positive recurrent. Hence, to show the stability of the original queueing system $\mathbf{Q}(n)$, it suffices to show the stability of $\mathbf{Q}^{(k+1)}(n)$.

To show this, we apply Foster's theorem [1, Proposition 5.3] to $\mathbf{Q}^{(k+1)}(n)$. We define the following Lyapunov function:

$$\tilde{V}(\mathbf{Q}^{(k+1)}) = V(\mathbf{Q}), \quad (20)$$

where $V(\mathbf{Q})$ is defined as below:

$$V(\mathbf{Q}) = \sum_l \frac{\sum_s Q_l^s H_l^s}{c_l}\left(\sum_{k\in\mathcal{E}_l} \frac{\sum_s Q_k^s H_k^s}{c_k}\right). \quad (21)$$

We can use techniques similar to those used in the proof of Theorem 1 here to bound the drift of $\tilde{V}(\mathbf{Q}^{(k+1)})$ conditioned on $Q^{(k+1)}(n)$ as follows

$$\begin{aligned} & E\left[\tilde{V}(\mathbf{Q}^{(k+1)}(n+1)) - \tilde{V}(\mathbf{Q}^{(k+1)}(n))\middle| \mathbf{Q}^{(k+1)}(n)\right] \\ = \;& E\left[V(\mathbf{Q}(n+1)) - V(\mathbf{Q}(n))\middle| \mathbf{Q}^{(k+1)}(n)\right] \\ = \;& 2E_{\mathbf{Q}^{(k+1)}(n)}\left[\sum_{l\in\mathcal{L}} \frac{Q_l(n)}{c_l}\right. \\ & \left.\times \sum_{m\in\mathcal{E}_l}\left(\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_m^{sh} - \pi_m(n)\right)\right] + B. \end{aligned}$$

Notice that $x_s^{(h)}(n)$ for all $l = 1, 2, \dots, k+1$ is only determined by the arrival process and the departure process of $\mathbf{q}^{(\leq k)}(n)$, both of which are independent of $\mathbf{q}^{(k+1)}(n)$. It is easy to see that

$$\begin{aligned} & E_{\mathbf{Q}^{(k+1)}(n)}\left[\sum_{l\in\mathcal{L}} \frac{Q_l(n)}{c_l}\sum_{m\in\mathcal{E}_l}\left(\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_m^{sh} - \pi_m(n)\right)\right] \\ \leq\; & E_{\mathbf{Q}^{(k+1)}(n)}\left[\sum_{l\in\mathcal{L}} \frac{Q_l(n)}{c_l}\left(\sum_{m\in\mathcal{E}_l}\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_m^{sh} - 1\right)\right] \\ =\; & E\left[\sum_{l\in\mathcal{L}} \frac{Q_{ij}^{(\leq k)}(n)}{c_{ij}}\left(\sum_{l\in\mathcal{E}_l}\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_l^{sh} - 1\right)\right] \\ & + \sum_{l\in\mathcal{L}} \frac{Q_{ij}^{(k+1)}(n)}{c_{ij}}(E\left[\sum_{l\in\mathcal{E}_l}\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_l^{sh}\right] - 1) \end{aligned}$$

Since $\mathbf{Q}^{(\leq k)}(n)$ is an ergodic process and $x_s^{(h)}$ is simply some deterministic function of $\mathbf{Q}^{(\leq k)}(n)$, we know that for any $\epsilon > 0$, there exists T_ϵ, such that for all $n \geq T_\epsilon$ and $h = 1, 2, \dots, k+1$,

$$|E[x_s^{(h)}(n)] - \lambda_s| \leq \epsilon$$

and there exists a constant C_2 such that

$$E\left[\sum_{l\in\mathcal{L}} \frac{Q_l^{(\leq k)}(n)}{c_l}\left(\sum_{m\in\mathcal{E}_l}\sum_{h=1}^{k+1}\sum_s x_s^{(h)}(n) H_m^{sh} - 1\right)\right] \leq C_2.$$

Finally, we have for $n \geq T_\epsilon$,

$$\begin{aligned} & E\left[\tilde{V}(\mathbf{Q}^{(k+1)}(n+1)) - \tilde{V}(\mathbf{Q}^{(k+1)}(n))\middle| \mathbf{Q}^{(k+1)}(n)\right] \\ \leq\; & 2\sum_{l\in\mathcal{L}} \frac{Q_l^{(k+1)}(n)}{c_l}\left\{\sum_{m\in\mathcal{E}_l}(\sum_s(\lambda_s+\epsilon)H_m^s) - 1\right\} + C_2 + \epsilon \\ \leq\; & -2\sum_{l\in\mathcal{L}} \frac{Q_l^{(k+1)}(n)}{c_l}\left\{(-\epsilon+\eta)\sum_{m\in\mathcal{E}_l}(\sum_s H_m^s)\right\} + C_2 + \epsilon, \end{aligned}$$

where η is defined as follows: if we assume $\{\pi^{(k)}: \; 1 \leq k \leq K\}$ to be the set of all possible matchings, then for any rate vector $[\lambda_s]$ satisfying (8), there exists a probability vector $[\alpha_i, 1 \leq i \leq K](\sum_i \alpha_i = 1)$ and a positive number $\eta > 0$, such that for all l, $\lambda_l \leq \lambda_l^* - \eta$ and all $l \in \mathcal{L}$, $[\lambda_l^*]$ satisfies

$$\kappa_{max}\lambda_l^* = \sum_{k=1}^{K} \alpha_k \pi_l^{(k)}. \quad (22)$$

Note here we use the fact that κ_{max} is the maximum number of simultaneously schedulable links in any interference set.

By choosing $\epsilon = \eta/2$, we find a bounded region of $\mathbf{Q}^{(k+1)}$

$$\chi = \left\{\mathbf{Q}^{(k+1)} : \sum_{l\in\mathcal{L}} \frac{Q_l^{(k+1)}}{c_l}\sum_{m\in\mathcal{E}_l}(\sum_s H_m^s) \leq \frac{C_2+\eta/2}{\eta}\right\},$$

such that, when $n \geq T_{\eta/2}$, the expected drift is always negative if $\mathbf{Q}^{(k+1)}(n)$ is outside this region. The stability of the system follows. ◇

REFERENCES

[1] S. Asmussen. *Applied Probability and Queues*. Springer-Verlag, New York, 2003.

[2] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu. Joint congestion control and distributed scheduling in multihop wireless networks with a node-exclusive interference model. In *Proceedings of IEEE INFOCOM*, 2006.

[3] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees in maximal scheduling in allerton networks. In *Proceedings of 43rd Annual Allerton Conference on Communication, Control and Computing*, 2005.

[4] P. Chaporkar, K. Kar, and S. Sarkar. Achieving Queue Length Stability Through Maximal Scheduling in Wireless Networks. *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[5] J.G. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proceedings of IEEE INFOCOM*, pages 556–564, 2000.

[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.

[7] B. Hajek and G. H. Sasaki. Link scheduling in polynomial time. *IEEE Transactions on Information Theory*, 34(5):910-917, 1988.

[8] C. Humes, Jr. A regulator stabilization technique: Kumar-Seidman revisited. *IEEE Transactions on Automatic Control*, pages 191–196, 1994.

[9] F. P. Kelly, A. Maulloo, and D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.

[10] P. R. Kumar and S. P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Transactions on Automatic Control*, 40:251–260, February 1995.

[11] 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*, pages 289–298, March 1990.

[12] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer control in multihop wireless networks. In *Proceedings of IEEE INFOCOM*, 2005.

[13] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, pages 1406–1416, December 1991.

[14] S. H. Low and D. E. Lapsley. Optimization flow control, I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, pages 861–875, December 1999.

[15] B. Miller and C. Bisdikian. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication*. Prentice Hall, 2000.

[16] A.N. Rybko and A.L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problems of Information Transmission*, 28, 1992. Translated from Problemy Peredachi Informatsii, vol. 28, no. 3, pp. 3-26, 1992.

[17] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Ad-hoc Networks," in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003.

[18] S. Sarkar, P. Chaporkar, and K. Kar. Fairness and throughput guarantees with maximal scheduling in wireless networks. In *Proceedings of the 4th Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, Boston, MA, 2006.

[19] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.

[20] 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*, pages 1936–1948, December 1992.

[21] T. Weller and B. Hajek. Scheduling non-uniform traffic in a packet-switching system with small propagation delay. *IEEE/ACM Transactions on Networking*, pages 813–823, December 1997.

[22] X. Wu and R. Srikant. Regulated maximal matching: A distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing. *Proceedings of the IEEE Conference on Decision and Control*, December 2005.

[23] X. Wu and R. Srikant. Scheduling efficiency of greedy distributed algorithms in wireless networks. In *Proceedings of INFOCOM*, 2006.

[24] X. Wu and R. Srikant and J. R. Perkins. Queue-Length Stability of Maximal Greedy Schedules in Wireless Networks. *Proceedings of Information Theory and Applications Inaugural Workshop*, University of California, San Diego, February 2006.

[25] Y. Yi and S. Shakkottai, "Hop-by-hop Congestion Control over a Wireless Multi-hop Network," in *Proceedings of IEEE INFOCOM*, Hong Kong, March 2004.

[26] H. Yaiche, R. Mazumdar, and C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, Oct. 2000.

PLACE PHOTO HERE

Xinzhou Wu (S '01-M'06) received his B.E. from Tsinghua University, Beijing, China in 1998, his M.S. and Ph.D. from University of Illinois at Urbana-Champaign in 2000 and 2004. He was a member of technical staff at Flarion Technologies from 2005 to 2006. He is currently a staff engineer at Qualcomm Flarion Technologies. His current research interests are in wireless networks, multiple antenna systems, resource allocation in cellular communication systems, and information theory.

PLACE PHOTO HERE

R. Srikant (S '90-M '91-SM '01-F '06) received his B.Tech. from the Indian Institute of Technology, Madras in 1985, his M.S. and Ph.D. from the University of Illinois in 1988 and 1991, respectively, all in Electrical Engineering. He was a Member of Technical Staff at AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois at Urbana-Champaign, where he is a Professor in the Department of Electrical and Computer Engineering, and a Research Professor in the Coordinated Science Lab.

He was an associate editor of Automatica and the IEEE Transactions on Automatic Control, and is currently an associate editor of the IEEE/ACM Transactions on Networking. He has also served on the editorial boards of special issues of the IEEE Journal on Selected Areas in Communications and IEEE Transactions on Information Theory. He was the chair of the 2002 IEEE Computer Communications Workshop in Santa Fe, NM and will be a program co-chair of IEEE INFOCOM, 2007. His research interests include communication networks, stochastic processes, queueing theory, information theory, and game theory.

PLACE PHOTO HERE

James R. Perkins received the A.B. degree in engineering science from Harvard University, Cambridge, MA, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana-Champaign. Since 2002, he has been an Associate Professor of Manufacturing Engineering at Boston University, Boston, MA. His current research interests include dynamical systems, real-time deterministic and stochastic control of manufacturing systems, human resource scheduling and allocation, queueing theory, and scheduling and congestion control in communication networks.