Prioritized Maximal Scheduling in Wireless Networks

Qiao Li
qiaoli@cmu.edu
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

Rohit Negi
negi@ece.cmu.edu
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

***Abstract*— This paper considers the scheduling problem in wireless networks. We focus on prioritized maximal scheduling, where a maximal scheduler chooses the links in an order specified by certain priority. We first analyze the capacity region of a maximal scheduler with fixed priority, where a lower bound is formulated and shown to be tight. Next we propose both centralized and distributed algorithms to search for the optimal priority for a given arrival rate. The algorithms are optimal in the sense that, if the arrival rate is in the lower bound region of *any* prioritized maximal scheduling, it is stable under the result of our algorithms. Finally, by combining the priority assignment and maximal scheduling, we prove that we can achieve a certain fraction of the optimal capacity region, which is bounded below by a constant for most networks.**

Keywords: Maximal scheduling, wireless networks, capacity region, priority, MAC.

I. Introduction

Scheduling in wireless networks is, in general, an NP-hard problem. In the simplest setting, where the external sources behave like constant rate fluids, scheduling is achieved by channel assignment, which associates each link with some periodic time slots and is further reduced to the well known NP-complete graph coloring problem. The problem is more intricate when we consider the queueing dynamics under random packet arrivals and departures. In such stochastic networks, the optimal scheduling can be reduced to solving another NP-complete problem, the maximum weighted independent set (MWIS) problem [1], *in every time slot*. In fact, this optimal scheduling algorithm is rarely implemented in realistic networks due to the high resource consumption. As an alternative, researchers are seeking suboptimal scheduling algorithms, where a certain algorithm with low complexity is chosen under the assumption that it is "efficient". However, this notion of efficiency is hard to justify in most cases, since it is difficult to analyze the performance of a scheduling algorithm in the first place.

In past research, maximal scheduling is used as a model to analyze the performance of suboptimal scheduling [2]. In a maximal schedule, no link can be added without violating the interference constraint. In [2], a lower bound on the capacity region of the maximal scheduling is proposed and shown to achieve a certain fraction of the optimal region, where the fraction depends on the maximum "interference degree" of the network. This model seems to be sound since an efficient scheduler must be maximal. But the notion of maximal scheduling does not fully capture the variety of suboptimal schedulers that are possible. With a performance guarantee applicable to *any* maximal scheduling, their bound is only tight for the worst case. Not surprisingly, it is observed that their lower bound is quite loose when applied to certain examples (e.g. Table 1 in [2]). Therefore, it is reasonable to go beyond maximal scheduling to analyze and design provably efficient schedulers.

Motivated by the above observation, we consider *prioritized* maximal scheduling, where a maximal scheduler schedules the links in an order specified by certain priority. The priorities are pre-computed based on network parameters and change when network topology changes. During the scheduling, in each step the scheduler picks the link with the highest priority among the unprocessed links with nonempty queues, and adds it to the schedule if it does not violate the interference constraint. Note that this can also be implemented locally, where a link with nonempty queue can transmit if and only if all of the neighbors with higher priorities have decided not to transmit. In the analysis, we obtained a tight lower bound of the capacity region for the maximal scheduling with a fixed priority. Also we show that this lower bound region is a superset of the one proposed in [2], which agrees with our observation that [2] considers the worst case maximal schedule.

Next we consider how priorities can be chosen based on network parameters. Note that for a specific priority, the prioritized maximal scheduler may not be efficient. We first propose a centralized algorithm, GLOBAL-PRIORITY, which produces an optimal priority assignment for a given arrival rate. It is optimal in the sense that, if the arrival rate is in the lower bound region of *any* priority assignment, GLOBAL-PRIORITY will output a priority which makes the network stable. Further more, we show that the same performance can be achieved by a distributed algorithm, LOCAL-PRIORITY. Combining the priority assignment and scheduling, we can guarantee that if the arrival rate is inside the union of these lower bound regions, we can achieve stability by a distributed algorithm. Note that the information about the arrival rate can be obtained through estimation, and the result converges under the assumption that the arrival process obeys the strong

law of large numbers (S.L.L.N). Since the priority assignment must be run only when the network changes (rather than in every time slot), it is suitable for slowly-changing network topologies.

Finally we analyze the approximation ratio of our prioritized maximal scheduling algorithms. We show that if an arrival rate a is stable under the optimal scheduling, a/δ is stable under our algorithms, where δ is related to the interference degree in the contention graph and is a function of the network topology and the physical layer parameters. For example, δ is approximately 7 if the receiver requires a sufficiently high signal to interference plus noise (SINR) threshold [5]. Also we show the tightness of our guaranteed capacity region by proving that, there exists a network and an arrival process whose rate is stable under the optimal scheduling and is arbitrarily close to our guaranteed region, such that the network is not stable under any prioritized maximal scheduling.

The organization of this paper is as follows: in Section II we describe the system model. In Section III we analyze the capacity region of maximal scheduling with a fixed priority, and in Section IV we formulate the algorithms for priority assignment. Section V shows the performance guarantee of our algorithms, Section VI gives simulation results, and Section VII concludes this paper.

II. System Model

We consider the scheduling problem at the MAC layer of a wireless network. The topology is described by a directed graph $G = (V, E)$, where V is the set of user nodes and E is the set of communication links. A node pair $(u, v) \in E$ only if node v is in the transmission range of node u, which is determined by the transmission power and the channel propagation. The interference model is a contention graph $G_c = (V_c, E_c)$, where V_c is the set of links, and E_c is the set of conflicts, i.e., $(i, j) \in E_c$ if and only if they are not allowed to transmit together. Define a link i's neighbor set as $N_i = \{j : (i, j) \in E_c\}$. A link i is allowed to transmit only if no link in N_i is transmitting. The interference degree of link i, denoted as Δ_i, is defined as the cardinality of the maximum independent set in the subgraph formed by $N_i \cup \{i\}$. Denote a link removal sequence as $r = (i_1, i_2, \ldots, i_n)$, which is a permutation of $(1, 2, \ldots, n)$. We define $\delta = \min_{r\in\Pi} \max_{i\in V_c} \delta_i^{(r)}$, where $\delta_i^{(r)}$ is the interference degree of link i in the subgraph when it gets removed according to the order specified by r, and Π is the set of all permutations.

We assume the time is slotted. The arrival process at link i is $A_i(n)$, which is the cumulative number of packets arrived from an external source during the first n time slots. The only requirement on the arrival process is the S.L.L.N, i.e., for any link i we have $\lim_{n\to\infty} A_i(n)/n = a_i$ with probability 1 (w.p.1). We assume the packets arrive at the transmitters only at the beginning of a time slot. In each time slot, a scheduler chooses an independent set $I \subseteq V_c$ with nonempty queues for transmission. The queue length dynamics at link i can be described as:

$$Q_i(n) = Q_i(0) + A_i(n) - D_i(n) \quad (1)$$

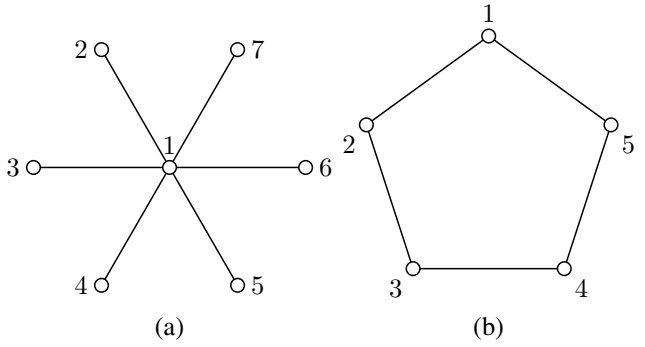


Fig. 1. (a) is a star-shaped network, and (b) is a pentagon-shaped network. Each node corresponds a link in the network, and each edge corresponds to an interference conflict.

where $Q_i(n)$ is the queue length of link i at the end of slot n, $Q_i(0)$ is the initial queue length of link i, and $D_i(n)$ is the cumulative departures during the first n slots. We assume that if a link is scheduled, it can transmit 1 packet in that time slot. The network is rate stable [2] if for each link i we have $\lim_{n\to\infty} D_i(n)/n = a_i$, w.p. 1, i.e., the departure rate equals the arrival rate.

Denote the maximal scheduler guided by priority p as π_p, where $p = (p_1, p_2, \ldots, p_n)$ and p_i is the priority of link i. Thus, if $p_i > p_j$, we say link i has lower priority than link j. During scheduling, for a link i with nonempty queue, either it is scheduled by π_p, or there is a link in N_i with higher priority scheduled.

Example: Consider the star-shaped network in Fig. 1(a), we have $\Delta_i = 1, 2 \le i \le 7$ since $\{1\}$ is the maximum independent set in the subgraph formed by $N_i \cup \{i\}$. Similarly we have $\Delta_1 = 6$. We have $\delta = 1$ since $\delta^{(r)} = 1$ when $r = (2, 3, 4, 5, 6, 7, 1)$. Suppose the priority is $p = (7, 4, 3, 5, 1, 2, 6)$. Thus, link 1 has the lowest priority (7). Under maximal scheduler π_p, link 1 can transmit if and only if none of the 6 links in N_1 is scheduled. However, any other link can transmit whenever it has packets since there is no neighbor with higher priority.

We evaluate the performance of a maximal scheduler π_p by analyzing its capacity region $\mathcal{A}_p$, which is the set of stable arrival rates under π_p. The capacity region of our prioritized maximal scheduler is $\mathcal{A} = \cup_p \mathcal{A}_p$, i.e., $a \in \mathcal{A}$ if and only if there exists a priority p such that a is stable under π_p. We will first analyze $\mathcal{A}_p$ for a fixed p in the next section.

III. Capacity Region of Prioritized Scheduling

In this section we analyze the capacity region of a prioritized maximal scheduler π_p with a fixed priority p. In general, characterizing $\mathcal{A}_p$ exactly is a hard problem, since if $a \in \mathcal{A}_p$, we have to show that the network is stable under all types of arrival processes with rate a. Therefore, we try to derive a lower bound in the following theorem.

Theorem 1: The network $G_c = (V_c, E_c)$ with arrival rate a is rate stable under prioritized maximal scheduler π_p if for

each link i, we have

$$a_i + \sum_{j \in N_i} a_j \mathbf{1}(p_i - p_j) < 1 \tag{2}$$

where

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We first discuss some intuition behind the proof. For a fixed priority p, denote the set of link i's neighbors with higher priority as L_i. In a time slot, for any link i with nonempty queue, either it is scheduled, or there is another link $j \in L_i$ is scheduled. In both cases the sum departures of link i and the links in L_i is not less than 1. Thus if we require that the corresponding sum of the arrival rates is less than 1, the network should be stable.

Proof: The stability is proved by the fluid model [3]. Note it can be obtained via minor modification to the proof in [2] (Lemma 2), where the maximal scheduler does not use priority. We omit the details due to space limit. ■

This region, denoted as $\mathcal{A}_p^L$, is a superset of $\mathcal{A}_{\min}^L$, which is the region obtained in [2]. To see this, suppose $a \in \mathcal{A}_{\min}^L$, we have $a_i + \sum_{j \in N_i} a_j < 1$ for any i. Therefore

$$a_i + \sum_{j \in N_i} a_j \mathbf{1}(p_i - p_j) \le a_i + \sum_{j \in N_i} a_j < 1. \tag{4}$$

from which we conclude $a \in \mathcal{A}_p^L$. Since $\mathcal{A}_{\min}^L \subseteq \mathcal{A}_p^L$, the performance guarantees in [2] about $\mathcal{A}_{\min}^L$ apply to $\mathcal{A}_p^L$ for any fixed p. The agrees with our observation that we are improving over the worst case maximal scheduler in [2]. Next we show this lower bound $\mathcal{A}_p^L$ is tight, in the sense that for any network, there exist a point on the boundary of $\mathcal{A}_p^L$ which is also on the boundary of $\mathcal{A}_p$.

Theorem 2: For any network $G_c = (V_c, E_c)$ and fixed priority p, there exists an arrival process with rate a which is arbitrary close to the boundary of $\mathcal{A}_p^L$, such that the network is unstable under the maximal scheduler π_p.

Proof: For any $\epsilon > 0$, we denote the link with the lowest priority as i. If i has no neighbor, the arrival rate $a = (0, \ldots, 0, 1 + \epsilon, 0 \ldots, 0)$ with the nonzero entry at position i will make link i unstable. Suppose i has neighbors $i_1, i_2, \ldots, i_k$. Consider an arrival rate such that $a_{i_l} = 1/k$ for $1 \le l \le k$, $a_i = \epsilon$ at link i, and $a_j = 0$ elsewhere. This a is just outside the boundary specified by (2). The arrival process is as follows: in the lth out of every k time slots, one packet arrives at link i_l, such that it can be immediately transmitted. In every time slot, one packet arrives at link i with probability ϵ. One can see that link i never gets a chance to transmit and hence the network is unstable. ■

The maximal scheduling with a fixed priority may not be better than any arbitrary maximal scheduling. For example, for the star-shaped network in Fig. 1(a), the priority assignment which gives the lowest priority to link 1 is precisely the worst case scheduler. Therefore one natural question is: is there an efficient way to search for the best priority? This question is quite hard since we do not have the exact characterization of $\mathcal{A}_p$, i.e., given an arbitrary a we don't know whether π_p can make it stable or not. We only know that a is stable under π_p if $a \in \mathcal{A}_p^L$, i.e., the lower bound region, and that there exists a priority under which a is stable if $a \in \mathcal{A}^L = \cup_p \mathcal{A}_p^L$. Therefore we modify the question as: suppose $a \in \mathcal{A}^L$, is there an efficient way to search for the best priority? In the next section we show that this problem can be solved, even by a distributed algorithm.

IV. Priority Assignment Algorithm

In this section we find a (optimal) priority that stabilize the network whenever $a \in \mathcal{A}^L$. We first propose a centralized algorithm, as below.

GLOBAL-PRIORITY(G_c, a)
1 **for** $k = 1$ to n
2 **do** $s \leftarrow \arg\min_{i \in V_c} \{a_i + \sum_{j \in N_i} a_j\}$
3 $p(s) \leftarrow n + 1 - k$;
4 Remove s from V_c and incident edges from E_c;
5 **return** p

During the process, GLOBAL-PRIORITY assigns the priority from bottom up. In each step, the link with the minimum sum rate of neighborhood is assigned the lowest priority available and then gets removed. Note this is similar to the coloring algorithm in [4], where in each step the link with the minimum degree is removed, and after all the nodes are removed, the colors are assigned in the reverse order. In fact, if we consider a special case, where the arrival rates are uniform among the links, the priority chosen by GLOBAL-PRIORITY is exactly the same as the one chosen by the coloring algorithm. Intuitively, the coloring order is equivalent to the priority in scheduling, since when a link is being assigned a color, only the neighbors that are already colored matter. Thus, essentially, we have generalized the coloring algorithm in [4] to the stochastic network. Next we prove that GLOBAL-PRIORITY is optimal.

Theorem 3: If $a \in \mathcal{A}^L$, GLOBAL-PRIORITY will return a priority p such that $a \in \mathcal{A}_p^L$.

Proof: Since $a \in \mathcal{A}^L$, there exists a priority p' such that

$$a_i + \sum_{j \in N_i} a_j \mathbf{1}(p'_i - p'_j) < 1 \tag{5}$$

for every link i. We claim that at any step k, the link s chosen by the algorithm satisfies the following:

$$a_s + \sum_{j \in N_s \cap V_c(k)} a_j < 1 \tag{6}$$

where $V_c(k)$ is the remaining vertex set at step k. To see this, consider any link $i \in V_c(k)$, (5) implies

$$a_i + \sum_{j \in N_i \cap V_c(k)} a_j \mathbf{1}(p'_i - p'_j) < 1. \tag{7}$$

Particularly, if we consider the link s' with the lowest priority in $V_c(k)$ according to p', we have

$$a_{s'} + \sum_{j \in N_{s'} \cap V_c(k)} a_j < 1. \tag{8}$$

Thus (6) follows from the fact that

$$a_s + \sum_{j \in N_s \cap V_c(k)} a_j \leq a_{s'} + \sum_{j \in N_{s'} \cap V_c(k)} a_j < 1 \qquad (9)$$

since s was chosen to correspond to the minimum (step 2).

Finally, note that because we are assigning priority from the lowest to the highest, when we remove a link s at step k, the links in $N_s \cap V_c(k)$ are exactly the links with higher priority than s in N_s. Thus (6) is equivalent to (2) for the link s removed at step k. Since each link is eventually removed, (2) holds for all the links. ■

Note the above proof does not need s to be the minimum, instead one can choose any link s with $a_s + \sum_{j \in N_s \cap V_c(k)} a_j < 1$ because of (8). However, this method fails if $a \notin \mathcal{A}^L$. For example, consider a square-shaped graph with node set $V_c = \{1, 2, 3, 4\}$ and edge set $E_c = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$. It can be shown that the network is stable under any π_p as long as $a_i + a_j < 1$ for any $(i, j) \in E_c$. But for a uniform rate of $5/12$, no link satisfies $a_i + \sum_{j \in N_i} a_j < 1$ in the first step. In this case the GLOBAL-PRIORITY can still output a valid priority.

One can also observe that a partial ordering in a link's neighborhood is sufficient for prioritized maximal scheduling, since it only need to know whether a neighbor with higher priority is scheduled or not. Therefore, under the restriction that $a \in \mathcal{A}^L$, we can achieve the priority assignment by a distributed algorithm. In this algorithm, each link i takes the arrival rates in $\{i\} \cup N_i$, and generates L_i, a list of links with higher priority. During the scheduling, link i can be scheduled if and only if none of the links in L_i is scheduled. If a link decides that it has low priority, it sends a message to its neighbors.

LOCAL-PRIORITY($\{a_j : j \in \{i\} \cup N_i\}$)
1 $L_i \leftarrow N_i$;
2 **while** (1)
3 **do** Upon receiving a message from a neighbor j
4 Remove j from L_i;
5 **if** $a_i + \sum_{j \in N_i \cap L_i} a_j < 1$
6 **then** Send a message to each link in L_i;
7 **return** L_i

From the above discussion, we have the following theorem:

Theorem 4: If $a \in \mathcal{A}^L$, LOCAL-PRIORITY will return a partial priority under which a is stable.

V. Performance Analysis

In this section we discuss the performance guarantees of our combined priority assignment and maximal scheduling algorithm. It has been proved in [2] that the worst case maximal scheduling can achieve $1/\Delta$ of the capacity region, where $\Delta = \max_{i \in V_c} \Delta_i$ is the maximum interference degree of the graph. By choosing the priority properly, our algorithm can achieve a better approximation ratio.

Theorem 5: If a is stable under the optimal scheduling, then a/δ is stable under the priority p computed by LOCAL-PRIORITY.

Proof: Assume δ is achieved under a removal sequence $r^\star$, we assign the priority $p^\star$ to the links in the order specified by $r^\star$, such that $p^\star_{r_k} = k$, i.e., a link has higher priority if it appears earlier in r. Now consider link i and L_i, which is the set of i's neighbors with higher priority. We claim that under $\pi_{p\star}$, at most δ links in $\{i\} \cup L_i$ can transmit together. This follows from the fact that due to the removal procedure, the number of independent links in $\{i\} \cup L_i$ is $\delta_i^{(r^\star)}$, which is further bounded by δ.

From the above discussion, one can see that for any a that is stable under the optimal scheduling, we have

$$a_i + \sum_{j \in N_i} a_j \mathbf{1}(p_i^\star - p_j^\star) < \delta \qquad (10)$$

Dividing both sides by δ we have

$$\left(\frac{a_i}{\delta}\right) + \sum_{j \in N_i} \left(\frac{a_j}{\delta}\right) \mathbf{1}(p_i^\star - p_j^\star) < 1 \qquad (11)$$

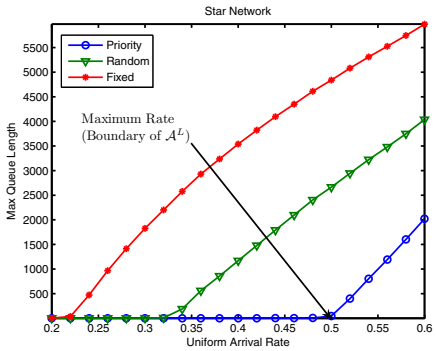
from which we conclude $(a/\delta) \in \mathcal{A}^L$. Thus according to Theorem 4, LOCAL-PRIORITY can produce a stabilizing priority. ■

Note δ can be upper bounded by the max-min interference degree of G_c, i.e., $\delta \leq \max_{H \subseteq G_c} \min_{i \in H} \Delta_i$, which in the worst case can be further bounded by a constant depending on the physical layer model. For example, [5] gives an upper bound on δ, which depends on both γ and α, the SINR threshold at the receiver and the distance loss exponent, respectively. For a fixed α, the upper bound of δ is shown to be a decreasing function of γ. Particularly, for sufficiently large γ, it is upper bounded by 7.

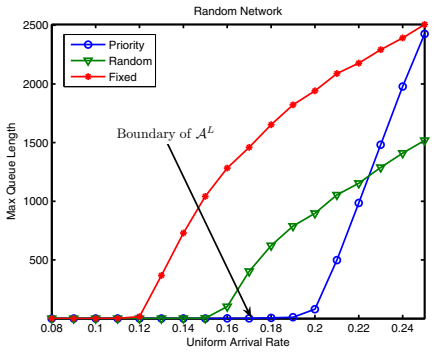
Since our algorithm can guarantee stability if $a \in \mathcal{A}^L$, one concern is: is it possible that this region is much smaller than $\mathcal{A}$? We try to answer this question by showing that there exist some network instances where some points just outside the boundary of $\mathcal{A}^L$ are also outside the boundary of $\mathcal{A}$.

Theorem 6: There exists a network and an arrival rate a which is arbitrarily close to (but outside) the boundary of $\mathcal{A}^L$, such that for any p, π_p can not stabilize an arrival process with rate a, i.e., a is outside $\mathcal{A}$.

Proof: Consider the pentagon-shaped graph in Fig. 1(b). Note the maximum uniform rate for this network is $2/5$, because at most two links can transmit together in any time slot, and it can be stabilized by the optimal scheduling [1]. But $\mathcal{A}^L$ only achieves a uniform rate of $1/3$ because of (2). Consider the uniform arrival rate of $\frac{1}{3} + \frac{1}{M}$, where M is a large integer. Suppose it is stable under π_p, in which link 1 has the lowest priority. Now consider the following arrival process with period $3M$: in each of the first $M + 3$ slots, one packet arrives at link 2 and 4, respectively, and in each of the second $M + 3$ slots, one packet arrives at link 3 and 5, respectively, such that they are all transmitted immediately. Further, assume that in each time slot one packet arrives at link 1 with probability $\frac{M+3}{3M}$. Therefore link 1 only gets a transmission rate of $\frac{M-6}{3M}$ and is not stable. ■



(a)



(b)

Fig. 2. The stability results of prioritized scheduling, random scheduling and fixed priority scheduling in (a) a star-shaped network and (b) a random network with 20 links.

VI. Simulation Results

In this section we compare the performance of our algorithm to other algorithms by a simulation in some sample networks. We assume the arrival processes are i.i.d Bernoulli processes. In our algorithm, we first compute a priority using GLOBAL-PRIORITY and then schedule the links according to that priority. For comparison we consider maximal scheduling with a fixed priority $p = (n, n-1, \ldots, 1)$, and use it as an upper bound (in capacity) of the worst case maximal scheduling in [2]. We also consider a non-prioritized scheduling, the random maximal scheduling, where in each time slot the scheduler chooses the links in a random order. Note the only performance guarantee of the random maximal scheduler that we are aware of is the worst case bound in [2]. The stability result is illustrated by the maximum queue length in the network after 10^4 time slots. All results are obtained by taking the average of 30 independent simulations.

A. Star Network

We first consider the star-shaped network in Fig. 1(a). Note the maximum uniform rate in the star network is 0.5. Fig. 2(a) shows the stability result under different arrival rates. One can observe that the prioritized scheduling achieves the maximum rate 0.5, but the random scheduling only achieves a rate of 0.32, and the fixed priority only has 0.22. Theoretically, the worst case maximal scheduling can only guarantee a rate of $1/7 \approx 0.14$. From the simulation result, our algorithm has a throughput gain of at least 2.3 compared with the worst case maximal scheduling. In fact, because the max-min interference degree in this network is 1, Theorem 5 shows that our prioritized maximal scheduler is optimal.

B. Random Network

We generate a random network with 20 links in a squared region of area 1. The links have the same normalized length of 0.3, and the contention graph is generated using the "disk" model, i.e., two links form an edge if and only if the distance between one link's receiver and the other's transmitter is less than a certain threshold. In this simulation we choose the distance threshold as 0.33. Fig. 2(b) shows the stability results under different scheduling algorithms. One can see that the prioritized scheduling has the maximum rate of 0.19, which is better than that of the fixed priority, 0.12. Thus the prioritized scheduling has a gain of at least 1.6 compared to the worst case scheduling. A random scheduling achieves the maximum rate of 0.15. Also note that the theoretically guaranteed stable rate (i.e., the boundary of $\mathcal{A}^L$) for this network is about 0.17, which is greater than the simulation result of the random scheduler, whose maximum rate 0.15 is not guaranteed to be stable under all types of arrival processes.

VII. Conclusion

In this paper we consider prioritized maximal scheduling in wireless networks. We first propose a tight lower bound of the capacity region in the case of fixed priority. We then formulate two algorithms, one centralized and one distributed, to search for the optimal priority, for the case where the arrival rate is in the union of the lower bound regions. It is proved that these algorithms can achieve a certain fraction of the optimal region, where the fraction is related to the interference degree of the network and can be bounded below by a constant for most networks. Finally the tightness of the region guaranteed by our algorithm is discussed.

Acknowledgement

This work was supported in part by the US NSF under awards CNS-0347455 and CNS-0520153, and by US ARO award W911NF0710287.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, Vol. 37, No. 12, pp. 1936-1949, December 1992

[2] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks", *IEEE Trans. on Info. Theory*, Vol. 54, No. 2, February 2008, pp. 572-594.

[3] J.G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proc. of the IEEE INFOCOM*, 2:556-564, March 2000.

[4] D. S. Hochbaum, "Efficient bounds for the stable set, vertex cover, and set packing problems," *Discrete Applied Mathematics 6*, pp. 243-254, 1983.

[5] R. Negi and A. Rajeswaran, "Physical layer effect on MAC performance in ad-hoc wireless networks," *Proc. of CIIT*, Phoenix, USA, Nov. 2003.