Throughput of Random Access without Message Passing

Alexandre Proutière, Yung Yi, and Mung Chiang

***Abstract*— We develop distributed scheduling schemes that are based on simple random access algorithms and that have no message passing. In spite of their simplicity, these schemes are shown to provide high throughput performance: they achieve the same performance as that of some maximal scheduling algorithms, e.g. Maximum Size scheduling algorithms.**

I. Introduction

A. Motivation and Summary

The quest for throughput optimal (distributed) scheduling schemes in wireless networks has attracted a lot of attention during the past ten years. We can categorize such distributed scheduling schemes into two broad categories. A first class of schemes aims at proposing distributed implementations (e.g., see [1]–[4] and the references therein) of the centralized Max-Weight scheduler originally proposed by Tassiulas and Ephremides [5]. To achieve throughput optimality, the proposed algorithms rely on (sometimes heavy) signaling procedures, whose impact on the actual performance is unclear and has not been yet quantified. A second class of algorithms is based on random access protocols (e.g., see [6], [7]). The idea there is that each transmitter tunes its transmission probability depending on information in its local neighborhood (typically the set of neighbors and the states of their buffers). Then these algorithms also rely on message passing among neighboring links, and yet their throughput performance remains unclear.

In this paper, we propose a different approach for the design of efficient scheduling schemes. The aim is to design schemes based on simple random access algorithms (basically slight extensions of ALOHA-like algorithms) that operate *without* any information exchange among links. This is in sharp contrast to the collision-free algorithms or random access algorithm requiring queue-length information exchange mentioned above, making it the most practical MAC protocol. Actually in many practical scenarios, e.g. in case of networks with hidden terminals, it proves difficult to exchange information among neighbors. The main contribution of the paper is to show that such simple schemes without message passing may provide at least the same throughput performance guarantees as some particular maximal scheduling algorithms, e.g. Maximum Size scheduling algorithms.

The efficiency of simple random access algorithms without message passing has been recently investigated by Durvy-Thiran [8]. They proposed to model the behavior of such algorithms using loss networks. A similar model has been suggested by Kelly [9]. In [8], the authors studied the spatial reuse of a saturated network under very simplistic multi-access schemes where all the transmitters attempt to use the radio resource with fixed probability. The present paper extends Durvy-Thiran's model to propose simple schemes whose throughput performances are no less than maximal scheduling. Unfortunately, maximal scheduling is known to be unable to provide throughput-optimal [10], [11]. However, simulations indicate that its throughput performance is generally good. We provide further theoretical evidences of this throughput efficiency.

We believe that the absence of information exchange makes it impossible in most cases to achieve throughput optimality. We explain this observation based on the models proposed of this paper. However we also believe that further simple modifications of random access schemes could greatly improve the throughput performance, and make it very close to optimality.

A. Proutière is with Microsoft Research, Cambridge, UK (e-mail: Alexandre.Proutiere@microsoft.com), and Y. Yi and M. Chiang are with the Department of Electrical Engineering, Princeton University (e-mail: {yyi,chiangm}@princeton.edu).

B. Model

Network and Traffic Model. Consider a wireless multi-hop network with a set $\mathcal{L}$ of L links. We model interference by a boolean matrix $A \in \{0,1\}^{|\mathcal{L}|\times|\mathcal{L}|}$, where $A_{lk} = 1$ if and only if the transmission on link l interferes that on link k, and 0 otherwise. For simplicity we assume that A is a symmetric matrix[1]. The transmitters are assumed to transmit at a fixed rate, say 1, when active. The results can be readily extended to different transmission rates over links. Packets are assumed to have a fixed size, so that the packet transmission on any link has a fixed duration that we take equal to 1. Packets, stored in the infinite buffers, are generated according to a stationary ergodic process of intensity λ_l on link l. Unless otherwise mentioned, we further assume that the numbers of packets arriving at a transmitter per unit of time are i.i.d. across time units, and do not exceed one packet per time unit.

Maximum throughput region. Let Γ be the set of vectors $\gamma = (\gamma_l, l \in \mathcal{L})$ representing feasible link throughputs. Let $\mathcal{P}_{\mathcal{L}}$ be the set of subsets of $\mathcal{L}$ (including $\emptyset$). Further define $\Upsilon = \{\tau = (\tau_m, m \in \mathcal{P}_{\mathcal{L}}), \forall m, \tau_m \geq 0, \sum_{m\in\mathcal{P}_{\mathcal{L}}} \tau_m \leq 1\}$. Then, we have:

$$\Gamma = \left\{ \gamma : \exists \tau \in \Upsilon, \forall l \in \mathcal{L}, \gamma_l = \sum_{m\in\mathcal{P}_{\mathcal{L}}:l\in m} \tau_m \right\}.$$

[1] Interference is not necessarily symmetric, but practically acknowledgement at the MAC layer induces symmetry.

Γ is referred to as the maximum throughput region, which is a convex, coordinate convex set[2]. The boundary of Γ, $\partial\Gamma$, can be represented using the set $\mathcal{M}$ of maximal schedules. A maximal schedule is a set of non-interfering links such that it is impossible to add a new link to this set without creating interference.We have:

$$\partial\Gamma = \left\{ \gamma : \exists \tau \in \Upsilon, \sum_{m\in\mathcal{M}} \tau_m = 1, \forall l, \gamma_l = \sum_{m\in\mathcal{M}: l\in m} \tau_m \right\}.$$

We know that Γ is the set of intensity vector $\lambda = (\lambda_1, \ldots, \lambda_L)$ for which there exists a scheduling algorithm stabilizing the network [5]. The throughput performance of a given scheduling algorithm Π is measured in terms of its throughput region Γ_Π, i.e., the set of vectors λ for which the algorithm Π stabilizes the network. An algorithm Π is said to be θ-throughput optimal if θ is the maximum positive number α such that $\alpha\Gamma \subset \Gamma_\Pi$.

II. Random access algorithms and their throughput regions

In this section, we introduce a model to characterize the throughput performance of simple random access algorithms. We consider a slotted system where the slot duration is denoted by β. Each transmitter runs a non-adaptive CSMA (Carrier-Sensing Multiple-Access) protocol, i.e., after observing a idle slot, the transmitter of link l attempts to use the channel with probability p_l if its buffer is not empty. When it decides to transmit, it sends one packet before releasing the channel. Non-adaptiveness means that we do no change transmission probability over time.

Characterizing the throughput region of such algorithms is notoriously extremely difficult. However, in recent work [12], an approximation of the throughput region has been proposed and evaluated. It has be shown that the approximation is exact for some large networks, but is also very tight in small networks. However, the approximation is quite complex, and hard to manipulate and exploit: it involves the stationary distribution $\pi(p, \rho)$ of a certain discrete-time loss network parameterized by the vector $p = (p_1, \ldots, p_L)$, and by some other vector $\rho = (\rho_1, \ldots, \rho_L) \in [0 : 1]^L$. Then there are some well-defined functionals F_l such that for any point λ on the Pareto-boundary of the approximated throughput region, there exists a parameter vector ρ such that: for all l, $\lambda_l = F_l(\pi(p, \rho), p, \rho)$.

In Figure 1, we present a 6-link grid network and its throughput region, when links 1, 2, 5, and 6 (resp. 3 and 4) are equally loaded, i.e., $\lambda_1 = \lambda_2 = \lambda_5 = \lambda_6$ (resp. $\lambda_3 = \lambda_4$). The throughput region is shown for $\beta = 1/10$ and $\beta = 1/100$. All the transmitters transmit with probability 1/10.

Figure 1 suggests that when β is small enough (e.g., $\beta = 1/100$ roughly corresponds to today's 802.11g-based networks), even non-adaptive random access algorithms achieve very good throughput performance. In our example, for $\beta = 1/100$, transmitting with fixed probability 0.1 is almost 0.9-throughput optimal. Note that in the model of our example, there might be collisions when two interfering links start transmitting simultaneously, where the collision durations are set to be equal to those of successful packet transmissions. In the following sections, we will consider the case where the collisions last a *single slot* only. Then, we expect even higher throughput efficiency.

[2]A set $\mathcal{Y} \subset \mathbb{R}_+^L$ is coordinate-convex if $x \in \mathcal{Y}$ then for all $y \in \mathbb{R}_+^L$ with $y \leq x$, $y \in \mathcal{Y}$.

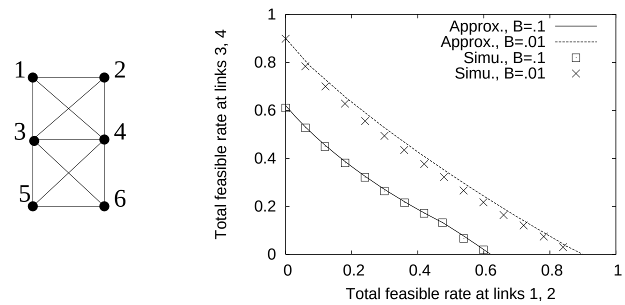

Fig. 1. A 6-link grid network and its throughput region. The left figure represents the network interference graph, e.g., an edge between links 1 and 4 indicates that these links interfere.

III. Weighted Fair Maximal scheduling via non-adaptive Random MAC algorithms

In this section, we show that by letting the contention period, i.e., β, be very small, non adaptive MAC algorithms approximately realize particular maximal scheduling schemes that we call *Weighted Fair Maximal* (WFM) scheduling algorithms. With WFM scheduling, at any time, a maximal schedule is chosen according to some distribution that depends on the set of active links (the links having packets in the corresponding buffers).

As mentioned earlier, we henceforth assume that collisions last for one slot only: if two transmitters of interfering links try to transmit simultaneously, they figure out that there is a collision at the end of the slot. We analyze the simple random access algorithms by considering the following two types of systems:

1) **Synchronous systems.** In these systems, time is divided into frames. At the beginning of each frame, each transmitter with non-empty buffer will attempt to use the channel with fixed probability at each slot until it can actually start transmitting, or until it senses activity of its neighbors succeeding to start transmitting, in which case it will wait for the next frame for further transmission attempts.
2) **Asynchronous systems.** In these systems, transmitters always sense the channel, and when their buffers are not empty, they start transmission of data just after an idle slot. As a consequence, the channel busy periods in different areas of the network may be not synchronized.

A. Synchronous systems

In synchronous systems, each frame is divided into two parts: (i) a first part devoted to contention resolution, and (ii) a second part corresponding to the transmission of actual data packets. To formally let β tends to 0, we consider a sequence of systems indexed by an integer N. In the N-th system, the duration of a slot is β^N and the contention resolution part of each frame is composed by M^N slots. The choices of β^N and M^N are made such that

$$\lim_{N\to\infty} \beta^N = 0, \quad \text{and } \beta^N \times M^N = \epsilon, \tag{1}$$

where ϵ is a very small constant. The second equation in (1) guarantees that the contention time is kept constant at ϵ irrespective of the sequence of systems. In what follows, for any $\mathcal{A} \subset \mathcal{L}$, we denote by $\mathcal{M}_\mathcal{A}$ the set of maximal schedules when the set of links is reduced to $\mathcal{A}$.

Lemma 1: Denote by $m_p^N(t, \mathcal{A})$ the schedule used in frame t in the N-th system when the transmission probabilities of the various links are p and when the set of active links is $\mathcal{A}$, and define:

$$\forall m \subset \mathcal{A}, \quad \tau_p^N(m, \mathcal{A}) \triangleq \mathrm{P}[m_p^N(t, \mathcal{A}) = m].$$

Then for all $m \subset \mathcal{A}$, $\tau_p^N(m, \mathcal{A})$ converges to $\tau_p(m, \mathcal{A})$ when N tends to ∞, and $\tau_p(m, \mathcal{A}) > 0$ if and only if $m \in \mathcal{M}_\mathcal{A}$.

The proof of the lemma relies on the fact that during the contention period, a schedule is constructed through a random packing process. This process ends if the contention period expires or if a maximal schedule has been constructed. Now it is easy to observe that the probability that a maximal schedule is constructed after M slots tends to 1 when M grows large, and the result follows.

Note that the random variables $m_p^N(t, \mathcal{A}), t = 0, 1, \dots,$ are i.i.d., which justifies the definition of $\tau_p^N(m, \mathcal{A})$. Lemma 1 says that, depending on the set of active links and on the transmission probabilities, at each frame, a schedule is chosen with some probability. With probability approaching 1, this schedule is maximal. In other words we obtain a scheduling algorithm that probabilistically chooses a maximal schedule in each frame. Usually, the limiting distributions $\tau_p(\cdot, \cdot)$ is difficult to compute since the analysis of the random packing process is not trivial [13]. This process is different than those usually investigated in the literature, because here, several links can be activated simultaneously. This property ensures a better spatial reuse in the network.

As an example, consider a simple 3-link network where links 1 and 3 interfere link 2, but do not interfere with each other. Then, the limiting distribution of maximal schedules when all links are backlogged (i.e., $\mathcal{A} = \mathcal{L}$) is:

$$\tau_p(\{2\}, \mathcal{L}) = \frac{p_2(1-p_1)(1-p_3)}{\prod_{i=1}^3 (1-p_i) + p_2(p_1+p_3) - \prod_{i=1}^3 p_i},$$

$$\tau_p(\{1,3\}, \mathcal{L}) = \frac{(1-p_2)(1-(1-p_1)(1-p_3))}{\prod_{i=1}^3 (1-p_i) + p_2(p_1+p_3) - \prod_{i=1}^3 p_i}.$$

When $p_1 = p_2 = p_3 = 1/2$, the links 1 and 3 are scheduled 3/4 of the frames. For more general networks, it becomes non-trivial. However there are some cases where we are able to identify the mapping $p \mapsto \tau_p(\cdot, \cdot)$. We propose here two ways of controlling the weights of the WFM scheme obtained as a limit of random access algorithms. One, where the transmission probabilities are made very small, and the other where they are made very close to 1.

1) Small transmission probabilities: In conjunction with the assumption (1), the sequence of transmission probabilities p^N are such that:

$$\forall l, \lim_{N\to\infty} p_l^N = 0, \quad \lim_{N\to\infty} \frac{\beta^N}{\min_{l\in\mathcal{L}} p_l^N} = 0. \tag{2}$$

We further assume that all the following limits exist:

$$\forall l, k \in \mathcal{L}, \lim_{N\to\infty} \frac{p_k^N}{p_l^N} = a_{kl} \in \mathbb{R}^+ \cup \{\infty\}. \tag{3}$$

Fix the set of active links $\mathcal{A}$, and define the following randomized algorithm (R1) to build a maximal schedule $\mathcal{S}$:

Step 1 $\mathcal{S} = \emptyset, \mathcal{R} = \mathcal{A}$

Step 2 With probability $(\sum_{k\in\mathcal{R}} a_{kl})^{-1}$ do $\mathcal{S} = \mathcal{S} \cup \{l\}$ and $\mathcal{R} = \mathcal{R} \setminus \{k : A_{kl} = 1\}$,

Step 3 Apply Step 2 until $\mathcal{R} = \emptyset$

We denote by $\tau_1(m, \mathcal{A})$ the probability that the maximal schedule built by R1 is m.

Lemma 2: Denote by $m_{p^N}^N(t, \mathcal{A})$ the schedule used in frame t in the N-th system when the set of active links is $\mathcal{A}$, and define $\forall m \subset \mathcal{A}$, $\tau_{p^N}^N(m, \mathcal{A})$ the probability that it is equal to m. Then, under Assumptions (1)-(2)-(3), for all $m \subset \mathcal{A}$, $\tau_{p^N}^N(m, \mathcal{A})$ converges to $\tau_1(m, \mathcal{A})$ if $m \in \mathcal{M}_\mathcal{A}$, and to 0 otherwise.

The proof is similar to that of Lemma 1. Note that Assumption (2) ensures that the construction of a maximal schedule finishes before the end of the contention period with a probability that tends to 1 when N grows large. Indeed, if link l is in $\mathcal{R}$, the probability that it will appear before its neighbors in $\mathcal{S}$ during the next s fraction of the contention period (that is during the next $\epsilon s/\beta^N$ slots) is roughly equal to $1 - \exp(-\epsilon s p_l^N/\beta^N)$, which tends to 1 when N tends to ∞.

Suppose that we set the transmission probabilities to be equal to $(c_N)^{n_l(\mathcal{A})}$, where c_N is a sequence tending to 0 when N tends to ∞, and $n_l(\mathcal{A})$ is the number of interfering neighbors of link l in $\mathcal{A}$. Then, the corresponding random access algorithm tends to realize the *minimum-degree greedy* algorithm [13] to build the schedule, i.e., it builds the schedule sequentially choosing the link with the least number of interfering neighbors. However, the transmitters have to know the number of *active* interfering links. We next give an algorithm that realizes minimum-degree greedy algorithm, but does not require any information on active neighbors.

2) Large transmission probabilities: Again consider a sequence of systems, and assume:

$$\forall l, \lim_{N\to\infty} p_l^N = 1, \quad \lim_{N\to\infty} \frac{\beta^N}{\min_{l\in\mathcal{L}}(1-p_l^N)} = 0. \tag{4}$$

We further assume that all the following limits exist:

$$\forall l, k \in \mathcal{L}, \lim_{N\to\infty} \frac{\prod_{j\neq k: A_{jk}=1}(1-p_j^N)}{\prod_{j\neq l: Ajl=1}(1-p_j^N)} = b_{kl} \in \mathbb{R}^+ \cup \{\infty\}. \quad (5)$$

Fix the set of active links $\mathcal{A}$, and define the following randomized algorithm (R2) to build a maximal schedule $\mathcal{S}$:

Step 1 $\mathcal{S} = \emptyset, \mathcal{R} = \mathcal{A}$

Step 2 With probability $(\sum_{k\in\mathcal{R}} b_{kl})^{-1}$ do $\mathcal{S} = \mathcal{S} \cup \{l\}$ and $\mathcal{R} = \mathcal{R} \setminus \{k : A_{kl} = 1\}$,

Step 3 Apply Step 2 until $\mathcal{P} = \emptyset$

We denote by $\tau_2(m, \mathcal{A})$ the probability that the maximal schedule built by Algorithm (R2) is m.

Lemma 3: Denote by $m_p^N(t, \mathcal{A})$ the schedule used in frame t in the N-th system when the set of active links is $\mathcal{A}$ and define $\forall m \subset \mathcal{A}$, $\tau_{p^N}^N(m, \mathcal{A})$ the probability that it is equal to m. Then, under Assumptions (1)-(4)-(5), for all $m \subset \mathcal{A}$, $\tau_p^N(m, \mathcal{A})$ converges to $\tau_2(m, \mathcal{A})$ if $m \in \mathcal{M}_\mathcal{A}$, and to 0 otherwise.

The proof is similar to that of Lemma 2. If we set the transmission probabilities to be equal to $1 - c_N$, where c_N is a sequence tending to 0 when N tends to ∞, then the corresponding random access algorithm realizes the minimum-degree greedy algorithm, where each node does not need to know the active interfering links, different from the algorithm R1. In other words, with the simplest random access algorithm, we can realize the maximal scheduling algorithm that picks the schedule according to the minimum-degree scheme (with probabilistic tie breaking).

B. Asynchronous systems

In case of asynchronous systems, to understand the system behavior, we have to fix the set of active links $\mathcal{A}$. Then, for any fixed β and p, the system can be modeled as a loss network as explained in [12]. Due to collisions, the loss network loses its reversibility and thus its analysis becomes non-trivial in general. However we can build sequences of systems such that the contention period tends to 0, such that the system behavior becomes very close to that of a continuous-time reversible loss network. Such reversible processes have been suggested by Kelly [9] to model Local Area Networks and recently revisited and extended by Durvy and Thiran [8] to understand random access protocols in wireless networks. Next, we construct a sequence of systems where all the transmitters use random access algorithms, such that the system behavior converges to that of a particular loss network in heavy traffic.

Let us first introduce the loss network model corresponding to the system considered. We start from the interfernec (or conflict) graph: for each edge between two (interfering) links k and l in this graph, we build a unit-capacity link kl in the loss network. In this loss network, the routes are $r_l = \{(kl) : A_{kl} = 1\}$ for all $l \in \mathcal{A}$. Clients arrive on route r according to a Poisson process of intensity ν_l and leave the loss network with rate 1. Let $m \subset \mathcal{A}$ such that for all $k, l \in m$, $A_{kl} = 0$ (m is a feasible schedule): the stationary probability that there is a client on route r_l (which corresponds to the fact that there is a successful transmission on link l) for each $l \in m$ is proportional to $\prod_{l\in m} \nu_l$. Now consider the loss network in heavy traffic: for all l, $\nu_l = u \times \eta_l$, where u tends to ∞. Then at the limit when u tends to ∞, the set of active routes $m_\eta(t, \mathcal{A})$ at time t can be seen as the state of a Markov process alternating between maximal sets of routes (corresponding to maximal schedules). This Markov process is ergodic with stationary distribution $(\tau_\eta(m, \mathcal{A}), m \subset \mathcal{A})$ such that $\tau_\eta(m, \mathcal{A}) = 0$ if $m \notin \mathcal{M}_\mathcal{A}$ and if $m, m' \in \mathcal{M}_\mathcal{A}$:

$$\frac{\tau_\eta(m, \mathcal{A})}{\tau_\eta(m', \mathcal{A})} = \frac{\prod_{l\in m} \eta_l}{\prod_{l\in m'} \eta_l}. \quad (6)$$

Note that to have a Markovian setting, we need that the clients use routes for exponentially distributed periods (exponentially distributed packet transmission durations). However, this assumption does not make change the picture here, since the processes considered are reversible, and then insensitive to the distribution of holding times.

Let us go back to wireless systems based on random access algorithms. Consider a sequence of asynchronous systems such that (1)-(2)-(3) are satisfied, with for all $k, l \in \mathcal{L}$, $a_{kl} = \eta_k/\eta_l$, with for all l, $\eta_l > 0$. We have:

Lemma 4: Assume that the set of active link is always $\mathcal{A}$. Denote by $m_{p^N}^N(t, \mathcal{A})$ the schedule used at time t in the N-th system. The process $(m_{p^N}^N(t, \mathcal{A}), t \geq 0)$ converges in law (on all compacts) to the process $(m_\eta(t, \mathcal{A}), t \geq 0)$ representing the heavy-traffic behavior of the corresponding loss network.

Lemma 4 can be proved using classical continuity arguments for finite state-space Markov processes. The lemma states that in asynchronous systems, when the set of active links is fixed, then after a while (after the time it takes for the distribution of the limiting process $(m_\eta(t, \mathcal{A}), t \geq 0)$ to be close to its stationary distribution), random access algorithms tend to realize a Maximum Size scheduling scheme with probabilistic tie breaking. It is important to note the difference with synchronous systems. In synchronous systems, random access algorithms realize a Maximal scheduling scheme, with schedules chosen by the minimum-degree greedy algorithm, and the maximal schedules are found at the *beginning of each frame.*

C. Throughput guarantees

So far, we have shown that one can build sequences of systems based on non-adaptive random access algorithms that approximately realize particular maximal scheduling algorithms. Does it imply that we have found non-adaptive random access algorithms (without any message passing) that achieve almost the same throughput performance as that of these maximal scheduling algorithms? The answer is yes. In particular, for synchronous systems, the proof basically comes from the fact that the choices of the maximal schedule are independent across frames. Denote by Γ_{MD} the throughput region of the scheduling scheme for which the maximal schedule is constructed using the minimum-degree greedy algorithm.

Proposition 1: In synchronous systems, for any $\epsilon > 0$, there exists a scheduling scheme based on a non-adaptive random access algorithm and with throughput region $(1-\epsilon) \times \Gamma_{\rm MD}$.

In asynchronous systems, obtaining the similar result to synchronous systems is less obvious, since identifying a schedule with maximum size requires the underlying loss network to be close to its stationary regime. We now sketch the way one can handle this difficulty. The convergence to the stationary regime is known to be geometric, and we can make the rate of convergence uniform in the set of active links $\mathcal{A}$. To allow time for convergence, we need to ensure that the set of active links does not change very often. To that end, we propose and apply the following rules to the random access algorithms.
Rule 1. Let $T > 0$. The transmitter of a link l behaves as follows: if its queue length becomes smaller than LT, it stops trying to transmit; if its queue length becomes greater than $2LT$ then it will re-start trying to attempt the channel.
Rule 2. Periodic reset. Periodically, links wait after finishing a transmission until all transmitters have also finished transmitting their current packet. The period of resets T' can be made large enough so that this impact on the throughput region of resets is negligible.

Now, Rule 1 and the fact that the arrival and departure rates of packets in any buffer is bounded by 1 (by assumption) allows the interval between two successive changes in the set of active links to be at least T. We consider a sequence of systems based on random access algorithms satisfying Rules 1 and 2, and the assumptions of Lemma 4. Let us compute the ergodic average of the chosen schedules in a period of time where there is no change in the set of active links and no reset. The duration T_0 of this period lies between T and T'. Denote by $r_l^N(T_0)$, $r_l(T_0)$, and $r_l^{\rm MS}(T_0)$ the average number of packets served on link l in the system indexed by N, the limiting system, and the system where the schedule is chosen according to the Maximum Size scheduling scheme. Let us fix $\epsilon > 0$, one can find T large enough such that $|r_l(T_0) - r_l^{\rm MS}(T_0)| < \epsilon/2$, by ergodicity of the loss network process. Then by Lemma 4, we may find N (depending on ϵ, T, and T') such that $|r_l^N(T_0) - r_l(T_0)| < \epsilon/2$. We then conclude that between two resets, the average number of packets served on link l in the N-th system and the system running with a Maximum Size scheduling scheme are equal up to ϵ.

Denote by $\Gamma_{\rm MS}$ the throughput region of the maximal scheduling choosing schedules using Maximum Size algorithm.

Proposition 2: In asynchronous systems, for any $\epsilon > 0$, there exists a scheduling scheme based on a non-adaptive random access algorithm, modified as above, and with throughput region $(1-\epsilon) \times \Gamma_{\rm MS}$.

IV. Stability of Networks under Weighted-fair Maximal scheduling

The stability properties of WFM scheduling schemes are very difficult to analyze, since under these schemes, the service rate on a link depends on the set of active links, i.e., a system of interacting queues. It is a future direction to carry out an exhaustive analysis of the throughput region of general WFM algorithms.

In this section, we analyze, for some specific networks, the throughput performance of the Maximum Size scheduling schemes that can be realized using random access algorithms for large transmission probabilities. Few papers have investigated this issue. In [10], the authors provide a lower bound on the stability region of maximal scheduling in general; they prove that maximal scheduling is $1/\kappa_{\max}$-throughput optimal, where $\kappa_{\max}$ is the maximum number of independent[3] interferers of a link. The authors show that the bound is tight, but to obtain tightness they consider a worst case in terms of (a) the network topology and size, (b) the maximal scheduling algorithm, (c) the relative mean arrival rates at the various links, (d) the correlations among the packet arrival processes at various links. In [11], the authors show that the bound is also tight if we consider worst cases only w.r.t. (a), (b) and (c). The worst network topology is that of a star network as depicted Figure 2. This network has $K+1$ links: link 0 interfered by all other links, and links $1,\dots,K$ that do not interfere with each other. The worst maximal scheduling scheme is then a Maximum Size scheduling scheme that always breaks ties in favor of links $1,\dots,K$ (link 0 then may transmit only when the buffers of all the other links are empty).

Using random access algorithms, we may emulate Maximum Size scheduling scheme with probabilistic tie breaking (the probabilistic tie breaking rule is done according to (6)). For example, when all transmitters access the channel with the same probability, tie is broken uniformly in symmetric network. Note that tie is broken only when the buffers of link 0 and only one of the other links are not empty. Denote by a the probability that link 0 captures the channel is such cases. We show next that the probabilistic tie breaking rule considerably improves the throughput performance (compared to that of the worst maximal scheduling algorithms considered in [10], [11], i.e., $a = 0$). Denote by λ (resp. λ_i) the packet arrival rate at link 0 (resp. i for $i \geq 1$). The following proposition characterizes the throughput region of the Maximum Size scheduling schemes with probabilistic tie breaking. We use the notations:

$$f_i = \prod_{j=1, j\neq i}^{K} (1-\lambda_j), \quad f = \prod_{j=1}^{K} (1-\lambda_j).$$

Proposition 3: The network is unstable if and only if one of the following conditions holds:
(i) $\exists i \in \{1,\dots,K\}$: $\lambda_i + \lambda > 1$,

[3]Two links are independent if they do not interfere.

(ii) $\forall i \in \{1, \ldots, K\}$: $\lambda_i + \lambda \leq 1$ and $af_i \leq 1$, and

$$\lambda > \left(1 + a\sum_{i=1}^{K} \frac{\lambda_i f_i}{1-\lambda_i - af_i}\right)^{-1} \left(f + a\sum_{i=1}^{K} \frac{\lambda_i(1-\lambda_i) f_i}{1-\lambda_i - af_i}\right).$$

The previous proposition is a generalization of a result obtained by Zhang-Shen et al. [14] for 2x2 switches (equivalent to the wireless network considered here with $K = 2$). Let us sketch the proof of the result. Without loss of generality, we consider now a slotted system, where the packet transmission duration is one slot, and where packets are generated at the beginning of time slots (at most one packet per slot per link). Instability under condition (i) is immediate since in this case, the arrival rate vector does not belong to the maximum throughput region Γ. Now assume that the arrival rate vector is in Γ, and assume the system is empty at time 0 (we can do that by irreducibility). Then remark that at any time t, at most one of the links $1, \ldots, K$ is active (because when two buffers from links $1, \ldots, K$ are not empty, they are served in priority). We can show as in [14] that in case of instability, buffer 0 has to be unstable. Hence we may assume that link 0 is always active. With this assumption the system behavior is simple to analyze and we easily deduce that (ii) is a necessary and sufficient condition for instability.

In Figure 2, we give the worst throughput efficiency of the star network with $K + 1$ link depending on the tie breaking parameter a. Note that $a = 0.5$ is simply achieved assuming that transmitters attempt to access the channel with the same probability. As illustrated, the worst efficiency is far from being $1/\kappa_{\max}$. The latter lower bound is only tight when the number of links K is very large.

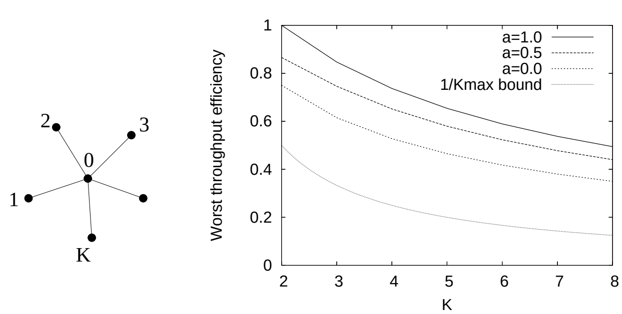

Fig. 2. The worst throughput efficiency in a star network with $K + 1$ link.

V. Beyond non-adaptive random access algorithms

Although non-adaptive random access algorithms can offer good throughput performance, these algorithms may not provide throughput optimality. Actually, one may argue that it is impossible to guarantee throughput optimality without any information exchange among links. An idea to provide throughput optimality in asynchronous systems could be to let the transmitters access the channel with probabilities that depend on the size of the corresponding buffer. In the setting of Section III-B, one may propose that link l transmits with probability $1-\exp(-\alpha Q_l)$, where Q_l is the buffer size of link l and α is large parameter. With such choices, one can show that in the limiting regime characterized by (6), the system would choose the schedule with *maximum weight* with high probability (depending on α). The problem is that we can not bound the time it takes to converge to such favorable configuration uniformly in the buffer sizes: it would take roughly $\exp(\alpha Q_l)$ slots to identify the favorable schedule.

Hence it seems crucial to exchange information among links if we want to design throughput optimal schemes using random access algorithms. First steps in this direction have been provided in [6], [7]: the proposed schemes are certainly very close to optimality. We let for future work a formal proof of this fact. However, as to whether it is better to deploy a very simple MAC protocol that does not have throughput optimality, or increasingly sophisticated ones with such optimality at the expense of communication overhead, the answer will depend on the actual application. This paper serves to provide another step along the way of understanding the really simple ones without any message passing.

References

[1] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proceedings of ACM Sigmetrics*, 2006.

[2] S. Sanghavi, L. Bui, and R. Srikant, "Distributed link scheduling with constant overhead," in *Proceedings of ACM Sigmetrics*, 2007.

[3] Y. Yi and M. Chiang, "Wireless scheduling with o(1) complexity for m-hop interference model," in *Proceedings of IEEE ICC*, 2008, to appear.

[4] Y. Yi, A. Proutiere, and M. Chiang, "Complexity in wireless scheduling: Impact and tradeoffs," in *Proceedings of ACM Mobihoc*, 2008, to appear.

[5] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, December 1992.

[6] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad-hoc wireless networks," in *Proceedings of IEEE Conf. on Decision and Control*, 2006.

[7] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Proceedings of IEEE Infocom*, 2007.

[8] M. Durvy and P. Thiran, "Packing approach to compare slotted and non-slotted medium access control," in *Proceedings of IEEE Infocom*, 2006.

[9] F. Kelly, "One-dimensional circuit-switched networks," *Annals of Probability*, vol. 15, pp. 1166–1179, 1987.

[10] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees in maximal matching in wireless networks," in *Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.

[11] X. Wu, R. Srikant, and J. R. Perkins, "Queue-length stability of maximal greedy schedules in wireless networks," *IEEE Transactions on Mobile Computing*, vol. June, pp. 595–605, 2007.

[12] C. Bordenave, D. McDonald, and A. Proutiere, "Performance of random multi-access algorithms, an asymptotic approach," *Proceedings of ACM Sigmetrics*, 2008.

[13] M. M. Halldorsson and J. Radhakrishnan, "Greedy is good: Approximating independent sets in sparse and bounded-degree graphs," *Algorithmica*, vol. 18, no. 1, pp. 145–163, 1997.

[14] R. Zhang-Shen, I. Keslassy, and N. McKeown, "Maximum size matching is unstable in any packet switch," *Technical report, Stanford Univ., TR03-HPNG-030100*, 2003.