A Coupled Processors Model for 802.11 Ad Hoc Networks Under Non Saturation

Christian Vitale
Institute IMDEA Networks, Spain
christian.vitale@imdea.org

Gianluca Rizzo
HES SO Valais, Switzerland
gianluca.rizzo@hevs.ch

Vincenzo Mancuso
Institute IMDEA Networks, Spain
vincenzo.mancuso@imdea.org

***Abstract*—In this paper we present an analytic approach to performance analysis of ad hoc networks under non saturation conditions, which does not rely on any assumption on traffic statistics. Our approach assumes traffic to be constrained by leaky bucket arrival curves, and it relies on a *coupled processors* model to capture the dependencies between user achievable rates due to sharing of the wireless transmission medium. We derive sufficient conditions for stability of transmission queues in an ad hoc network, and we describe a method for the determination of a proportionally fair allocation of resources, which allows trading the fairness of the solution for computational complexity. We validate our results through simulations, showing how our approach allows deriving operating points which both increase the fairness of the allocation and the overall average utilization of network resources with respect to saturated models.**

I. Introduction

Wireless mesh networks based on IEEE802.11 are nowadays an inexpensive, well widespread solution to easily, effectively and wirelessly connect entire cities. Thanks to such pervasiveness, they are poised to play a central role in many "Internet of Thing" application scenarios, with very diverse QoS requirements. Such wide deployment makes it crucial to develop models for analytical performance study of such networks, as empirical studies in such a complex environment hardly give clear indications on general properties of such systems. Performance analysis of the 802.11 CSMA/CA mechanism has traditionally focused on saturated traffic assumption [2]. Many of the available results for non-saturated conditions do not capture the effects of traffic dynamics on system performance. [9] for example, assumes that the probability of transmission and the probability of success of the stations in non-saturated conditions is always the same during time, despite traffic dynamics do alter significantly such quantities. [7] expands the work of Bianchi to unsaturated conditions, including transmission errors and capture effects. All of these approaches are based on a Poissonian traffic assumption, and they depend heavily on the assumption of traffic stationarity. [10] analyzes non saturation in heterogeneous traffic conditions, but its results still requires a complete stochastic traffic characterization to be parametrized. This has been done despite the fact that traffic in real networks is well far from being Poissonian (see [11] and related literature). In particular, traffic from live audio/video streaming exhibits a periodic behavior which substantially departs from the Poisson model, and which is characteristic of several known examples of instability [1]. This leaves open the issue of how to derive valid performance guarantees in ad hoc networks in realistic settings, when little is known about traffic statistics.

In the present work we propose a different approach, which assumes traffic to be constrained by *leaky bucket arrival curves* [4], which limit the maximum amount of bits which can arrive in a given time interval. Our analysis is based on a *coupled processors* model [3], which allows capturing the dependencies between user achievable rates due to sharing of the wireless transmission medium (mediated by the CSMA/CA mechanisms) and traffic dynamics, which such coupling entails. For the analysis of such model, we adopt the approach recently proposed in [13], based on Network Calculus.
The main contributions of this paper are as follows.

- We present an analytical approach to the study of ad hoc networks under non saturation conditions, which assumes traffic to be leaky bucket constrained. By applying our approach, we derive sufficient conditions for stability of transmission queues in ad hoc networks.
- We present a computationally feasible method for the determination of a proportionally fair allocation of resources in ad hoc networks, which allows trading the amount of fairness of the solution for computational complexity.
- We validate our results through simulations, assessing the quality of the bounds and of the optimal allocations derived with our approach.

The paper is organized as follows. Section II introduces the system model, and the main assumptions underlying our analysis. In Section III we introduce our method based on the coupled processor model, and we derive sufficient conditions for stability of our network. Section IV presents a heuristic for the computation of a proportionally fair operating point for the system. In Section V we assess numerically our results, and in Section VI we conclude our work.

II. System model

We consider a scenario with N hosts communicating via the IEEE 802.11 protocol in ad hoc mode. We assume all hosts are in range of each other. We assume hosts do not suffer for interference and that they do not move. Moreover, their capacity to each other does not change over time.
We do not make any assumption on the traffic generated by applications at each host. However, we assume that such traffic is passed through a *leaky bucket controller* [4], before being sent through the wireless connection. This device forces its output to be constrained by a *leaky bucket arrival curve*, with

parameters σ (burstiness) and ρ (rate). That is, if $A(t)$ is the cumulative arrival rate at the output of the controller, $\forall t \geq 0$, $\forall t' \leq t$, $A(t) - A(t') \leq \sigma + \rho t$ [4]. At the controller, all arrivals from the application which do not conform to such arrival curve are buffered. Introducing such controller allows to tune the amount of traffic buffered at the transmission queues of the hosts and, possibly, to avoid some pathological conditions for the system. For instance, it may happen that few hosts with a poor channel jeopardize the medium, at the expense of all other hosts. In these cases, constraining the traffic sent to the network by each host might help achieving a better allocation of system resources. For instance, this could be implemented at the application layer through communication among nodes, adjusting the operating point of the system in order to optimize a given utility function (e.g., in order to achieve some form of fairness). Moreover, assuming some form of constraint on sources allows dividing hosts into classes, with different service levels for each class. We assume traffic to be packetized, with a finite number of possible packet sizes.

Finally, let us briefly recall the definition of Generalized Processor Sharing (GPS) node [4]. A GPS node with n queues, total service rate R, and queue weights w_i, $i = 1, ..., n$, serves traffic at the i-th queue at time t at a constant rate $a_i R$ where a_i is the ratio between w_i and the sum of the weigths of all nonempty queues at t.

III. A CPS BASED ANALYSIS OF AD HOC NETWORKS

In this section we present an analytical method for the analysis of ad hoc networks under non saturation. First we introduce the *coupled processor* model, by which we capture the coupling in performance which characterizes ad hoc networks. Then we present our main results, based on a worst case approach to coupled processors systems. Finally, we describe a practical method for deriving performance bounds, based on an approximation technique which adapts to the characteristics of the system under study and of the specific performance problem to solve.

A. A CPS model for ad hoc networks

A crucial aspect of performance analysis in an ad hoc network is to being able to capture the effect of traffic of a given user on the performance experimented by other users. In what follows, we address this issue by modeling these interaction by means of a *coupled processors* model [3]. A CP system (CPS) is a set of parallel queues (i.e., queues which do not exchange traffic among them) served by work conserving schedulers, and whose service rates at any time t is completely determined by the set of active queues at that time. We define as the *state* of the system at a given time t the array $\mathbf{I}(t) = (I_1(t), I_2(t), ..., I_N(t))$ where for each node i, $I_i(t)$ is a binary variable which is equal to 0 if the queue at the $i-$th node is empty at time t, and 1 otherwise. Then at time t the service rate of the i-th queue in $\mathbf{I}(t)$ is $R_i(t) = R_i(\mathbf{I}(t))$, i.e., it is only function of the state of the system at time t.

In what follows, we model our N-node ad hoc network as a N-queue CPS, with one queue per transmitter. In such system, the coupling is in users transmission rates. Such coupling arises from sharing the same transmission medium, and is mediated by the CSMA/CA algorithm. The state $\mathbf{I}(t)$ of such CPS is given by the set of active ad hoc transmitters at time t. In modeling our ad hoc network as a CPS, we assume that for each ad hoc transmitter i at time t, the service rate $R_i(t)$ is completely determined by the state of the system. For each state of the system, we characterize the underlying CPS through the saturation throughput of the subset of active nodes, derived in [2]. That is, $\forall t \geq 0$, if $\mathbf{I}(t)$ is the set of active transmitters at time t, the instantaneous service rate at the i-th active transmitter is given by:

$$R_i(t) = \frac{P_s P_{tr} E(P)}{(1 - P_{tr})\delta + P_{tr} P_s T_s + P_{tr}(1 - P_s) T_c}. \quad (1)$$

Here, T_s is the average time the channel is sensed busy because of a successful transmission. T_c the average time the channel is sensed busy by each station during a collision; P_s is the probability that a transmission occurring on the channel is successful, which is given by the probability that exactly one station transmits on the channel, conditioned on the fact that at least one station transmits. P_{tr} is the probability that there is at least one transmission in the considered slot time. $E(P)$ is the average packet payload size, and δ is the duration of an empty slot time. All these parameters can be computed directly from the parameters of the CSMA/CA protocol, and they refer to the particular set of active transmitters $\mathbf{I}(t)$. For the expressions of each parameter, please refer to [2].

Indeed, in an ad hoc network the instantaneous service rate is determined by the CSMA/CA algorithm, and for a same system state (set of active queues) it generally varies over time. Therefore, as it is common in the study of such systems [2], in adopting a CPS model for such ad hoc network we are assuming that those system dynamics due to the CSMA/CA mechanisms take place on a smaller time scale than traffic dynamics (more specifically, the events of queues getting empty or full), so that they can be adequately modeled through their average effect on the system. Furthermore, while the ad hoc transmissions are scheduled sequentially, via contention, and the change of system state can happen only when a new host is scheduled, we assume the equivalent CPS model works at the "fluid" limit, i.e., each queue serves its traffic as it were infinitely divisible, and it can change state at any time t. In Section V we assess the validity of such assumption, showing numerically that these approximations model accurately the performance of our ad hoc network.

B. Sufficient conditions for stability

In this section we use the CPS model to derive an inner bound to the *stability region* for an ad hoc network, i.e., a set of leaky bucket rates for which traffic can be sustained by the ad hoc network without losing packets (when buffers in the network have a finite size) and with a finite bound on delay.

Our analysis is based on [13]. Differently than other results in the literature, such an approach is purely analytic and, as we show, it enables the derivation of heuristics that suitably trade-off accuracy for computational complexity. The first step consists in the derivation, from our N-queue CPS, of a set of feed-forward networks (which we call *auxiliary networks*) each composed by N GPS nodes, as depicted in Fig. 1, whose stability implies the stability of the CPS.

Let us label the queues of the CPS from 1 to N. With $\mathbf{n} = n_1, ..., n_N$ we indicate one of the possible mappings which associates the j-th stage of the feed-forward network,

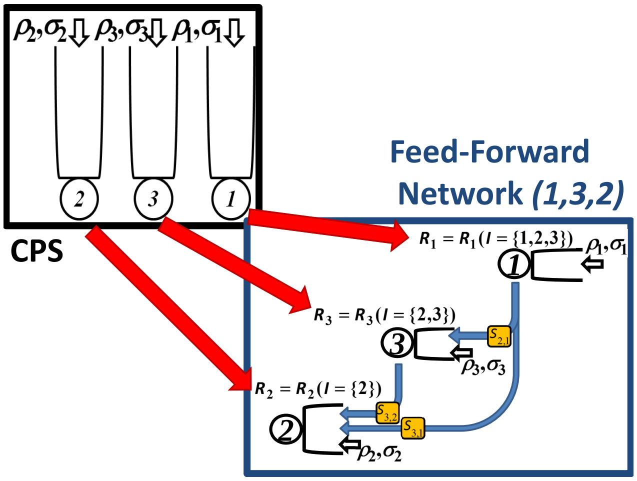


Fig. 1. A three-nodes CPS, and the structure of an auxiliary network, associated to the mapping $\{1, 3, 2\}$.

$j = 1, ..., N$ to the n_j th node of the CPS, j, $n_j = 1, ..., N$. To this mapping it corresponds a specific auxiliary network, composed by N GPS nodes.

In what follows, we briefly describe the structure of such network. We assume arrivals at the CPS queues are exactly the same at their corresponding GPS nodes in the auxiliary network, at any time t. The GPS nodes are connected according to the sorted list $\mathbf{n}$, in a feed-forward configuration: the first node receives the traffic that in the CPS corresponds to the first queue in $\mathbf{n}$, i.e., n_1. The second node receives the traffic corresponding to n_2, plus a *rescaled* version of the output of the first GPS node. In general, the j-th GPS node receives the traffic of the n_j-th CPS queue plus a rescaled version of the traffic served by GPS nodes $\in \{1, ..., j-1\}$. Fig. 1 shows an example with $N = 3$.

Each GPS node is composed by two queues, and it works at the fluid limit. One queue is dedicated to rescaled traffic, while the other is dedicated to fresh arrivals (i.e., arrivals from applications at each host). The total service rate of the j-th GPS node $R^{up}_{n_j}$ is the same as the n_j-th CPS queue when the CPS queues $\{n_j, n_{j+1}, ..., n_N\}$ are all active. The GPS weights are $w = R^{sat}_{n_j}/R^{up}_{n_j}$ for fresh traffic, and $1 - w$ for traffic from the policer, where $R^{sat}_{n_j}$ is the service rate at the n_j-th CPS queue when all queues at the CPS are active. Through the GPS scheduling, rescaled traffic models the effect of coupling between CPS queues on service rates, by decreasing the service rate offered to fresh traffic by an amount determined by the GPS coefficients.

The tuning of the rescaled traffic is achieved through the use of *scalers* [8], system components that reduce or amplify the volume of traffic moving from a queue to another in the network. More specifically, a scaler with scaling factor S, is a device such that if for any $t, \tau \geq 0$, $A(t) - A(t - \tau)$ is the amount of traffic which arrived at its input in the time interval $[t - \tau, t]$, the traffic at its output in the same time interval is $S(A(t) - A(t - \tau))$. In our auxiliary network, we assume the scaling factor $S_{j,k}$ for the traffic going from the k-th stage ($k = 1, ..., j - 1$) to the j-th GPS node is equal to:

$$S_{j,k} = \frac{R^{up}_{n_j} - R^{k,up}_{n_j}}{R^{up}_{n_k}} \tag{2}$$

where $R^{k,up}_{n_j}$ is the service rate at the CPS node n_j when the CPS nodes $n_p, ..., n_N$ are active.

For our ad hoc network, having defined the structure of each of the $N!$ auxiliary networks makes it possible to compute performance bounds, by applying some well known Network Calculus results on each of these networks. Exploiting the properties of the set of auxiliary networks, the following result defines a set of sufficient conditions ensuring that the leaky bucket rates for fresh traffic ρ_j, $j \in 1, ..., N$, yield a stable behavior for our system.

Theorem 3.1: With the given assumptions on the ad hoc network, and given a set of leaky bucket parameters $(\rho_j, \sigma_j), j = 1, .., N$ for fresh traffic, if it exists at least one auxiliary network such that at each stage $j = 1, ..., N$, ρ_j satisfies

$$\rho_j \leq max\left(R^{sat}_{n_j}, R^{up}_{n_j} - \sum_{p=1}^{j-1} S_{j,p}\rho_{n_p}\right) \tag{3}$$

then the ad hoc network is stable.

Proof: (sketch) This result is a direct application of the stability conditions for CPS in [13]. It is derived by first showing that the choice of scaling values in (2) guarantees that if at least one auxiliary network associated to our system is stable for a given set of leaky bucket traffic descriptors, then the ad hoc network is stable for those traffic descriptors. Indeed this choice brings to have at each GPS node a service rate for fresh traffic that, at an time t, is always not greater than the one offered at its corresponding node at the CPS that models our ad hoc network. Then the sufficient conditions are computed from analyzing the feed-forward network by stages, and they derive from imposing node stability at each GPS [13]. ■

In those cases in which it exists at least one auxiliary network for which the sufficient conditions in Theorem 3.1 are satisfied, hard bounds on backlog at the transmitter queues and on packet delay can be easily derived, by application of basic Network Calculus results at each node of the auxiliary network (see, e.g., [4], [13]).

IV. Derivation of the Optimal Operating Point

A. Problem Formulation

As already discussed, choosing the leaky bucket parameters of traffic sources in our networks allows tuning the operating point of the system, possibly in order to maximize some utility function. For instance, in order to guarantee that some form of fairness is maximized. In the present work, the utility function that we choose to optimize is a weighted fairness function, which is one possible way of balancing some notion of fairness among users with, for instance, different classes of service. Its expression is:

$$U = \sum_{i=1}^{N} w_i \log\left(\frac{\rho_i}{\rho_0}\right), \tag{4}$$

where ρ_0 is the minimum bit rate for an acceptable performance for the application. The feasible set of leaky bucket rates over which to optimize such utility is given by the set of inequalities in Theorem 3.1, as they define the set of rates for which the ad hoc network is able to serve the traffic load with

a finite maximum packet delay and backlog at each node.
The feasible operating points which maximize the weighted fairness are therefore the solutions of the following optimization problem, computed over the set of auxiliary networks and of fresh traffic leaky bucket rates:

$$\begin{aligned}
&\underset{\rho \geq \mathbf{0}, \mathbf{n} \in \mathcal{N}}{\text{maximize}} \quad \sum_{i=1}^{N} w_i \log\left(\frac{\rho_{n_i}}{\rho_0}\right); \\
&\text{subject to:} \\
&\forall \mathbf{n},\ \forall j = 1, \ldots, N, \\
&\rho_{n_j} + \min\left(R^{up}_{n_j} - R^{sat}_{n_j}, \sum_{p=1}^{j-1} \frac{R^{up}_{n_j} - R^{p-up}_{n_j}}{R^{up}_{n_p}} \rho_p\right) \leq R^{up}_{n_j},
\end{aligned} \tag{5}$$

where the constraints derive from Theorem 3.1. $\mathcal{N}$ is the set of the $N!$ possible permutations of the labels of ad hoc users (i.e., CPS queues).
Solving this problem is challenging for two main reasons: (i) the presence of the min function in the constraints, which leads to a non-convex feasibility region; and (ii) the complexity of the problem, which scales factorially with N.

B. Heuristic Approach

In order to practically solve the above problem, we propose a heuristic approach that consists in two parts.
The first part aims at reducing the problem to a tractable problem that can be solved with standard tools. Specifically, since the presence of the min function in the constraints of the above problem leads to a non-convex feasibility region, we use the so called *big-M transformation* [12]. In such way, the two terms of the min in (5) are not *active* at the same time. Instead of that constraint, the method builds two constraints in which we add a binary variable multiplying a large constant value M. Whenever the binary variable is equal to one, the large constant makes the constraint useless because all the feasible sets of leaky bucket rates satisfy it, while when the binary variable is zero the constraint is active. By choosing a value for the binary variables we select a part of the feasibility region.
The problem obtained in this way belongs to the mixed-binary programming family. We solve it by means of the *branch-and-bound* method [5]. For each $\mathbf{n}$, we stop the branch-and-bound evaluation when the intermediate solution is at most at ϵ away from the optimum. Tuning ϵ allows achieving different tradeoffs between computational cost and optimality of the solution.
The second part of our heuristic aims at reducing the number of auxiliary networks over which to search for the optimum. It is based on running a set of greedy searches from a set of starting points, each of which has been derived as follows. To each node $j = 1, ..., N$ of the CPS, we associate the quantity $\frac{1}{w_j}$. Then, starting from the first stage of the auxiliary network, we assign a node of the CPS to each stage with a probability proportional to this quantity. The idea underlying such algorithm for the choice of the starting points is that nodes with higher weights w_i in the utility function need to be modeled more accurately than the others. This is achieved by assigning those nodes to the last stages of the auxiliary network. Indeed, due to the structure of the auxiliary network, the lower the stage a node belongs to, the larger the set of nodes whose coupling with the considered one is modeled through accurate rescaled traffic rather than via a conservative penalty on the service rate, holding for any time t, and therefore independent on traffic patterns at interfering nodes.
We describe now the elementary step of the search. From a network $\mathbf{n}$, we consider the set of $N-1$ networks obtained by a swap of two contiguous nodes in $\mathbf{n}$. If the log utility value of $\mathbf{n}$ is lower than the max log utility among all these $N-1$ networks, the network with the highest log-utility value among all the $N-1$ is selected. Otherwise, the search stops. The largest of all local maxima computed from all starting points is the final output of our heuristic. By changing the number of starting points we can achieve different trade-offs between computational cost and optimality of the solution.

TABLE I. SETUP OF THE WIRELESS SCENARIO

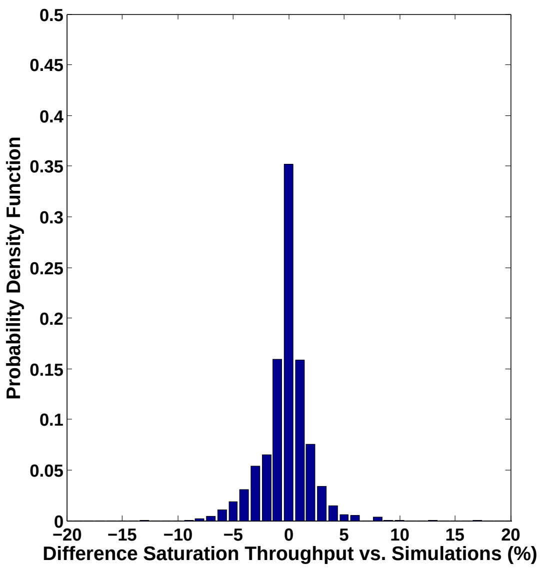
Available Channel Speeds (Mbit/s)	1;2;5.5;6;9;11;12 18;24;36;48;54
CWmin	16
Backoff stages	5
Preamble + PHY header(μs)	20
SIFS (μs)	16
ACK Time (μs)	24
DIFS (μs)	34
Slot time (μs)	9
MAC header(bits)	224
Chunk size (bits)	15000

V. NUMERICAL EVALUATION

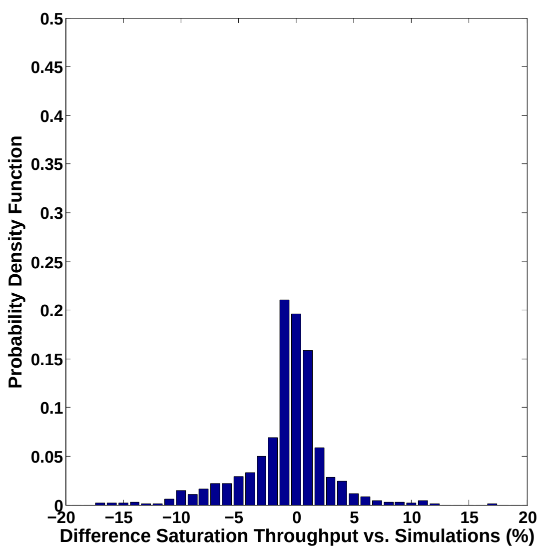
In this section we assess numerically our results. First, we evaluate the fitting of the proposed CPS model for a WiFi network. In particular, we evaluate the impact of assuming the saturation throughputs in (1) as the service rates of the CPS equivalent model. Then, we evaluate the performance of our heuristic, for proportional fairness. The operating points derived with our approach are compared to those obtainable in saturated condition, i.e., assuming all queues are always active, in function of the number of users in the system. Moreover, whenever feasible, i.e., for settings with only a few nodes, less than 7 in our evaluations, we have compared our results with those obtainable by solving the optimization problem (5) through brute force over all the upper bounding networks presented in Section III-A.
The parameters used in the considered scenarios are presented in Table I. We have chosen the 802.11 b/g standard for WiFi communications. The values of the parameters are derived from [6].

A. CPS Model Validation

In this section, we evaluate the fitting of the CPS model for 802.11 communications. In particular, for different scenarios, we pick at random a stable set of arrival rates for the transmitters. Such decision ensures that the set of active transmitters present in the wireless scenario, i.e., the system state changes over time. Then, for each system state and for each transmitter, we evaluate the difference among the average



(a) 3 Nodes.



(b) 7 Nodes.

Fig. 2. Probability density function, difference Simulations vs. CPS service rates.

throughput achieved during simulations and the service rate used in the CPS model, i.e., (1). Fig. 2 presents the probability density function (pdf) of the differences for scenarios having 3 and 7 nodes. Similar results have been obtained also for different set-ups of the wireless scenario.

In both cases, the service rates used in the CPS modelling are close to the ones computed during simulations. In particular, in the 99.73% and the 97.26% of the cases, respectively, the absolute value of the difference among the service rates computed as in (1) and the corresponding ones achieved through simulations is less than the 10% of the ones achieved through simulations. Even if the system assumes a given state just for a short interval of time, i.e., even if the subset of active transmitters does not change just for a limited period, the above result shows that the saturation throughput (1) is reached fairly soon in almost all the cases. Therefore, we consider as negligible the impact of the approximations introduced during the modelling of the system as a CPS.

B. *Proportional Fairness Optimization Results*

In this section we evaluate the results achieved through the optimization problem (5) in a large set of scenarios. In each setting, we solve (5) exploiting our heuristic, computing for a given number of users the set of leaky bucket rates which maximize the log-utility function.

In the following, the weights w_i of the utility function, defined in Section IV-A, are uniformly distributed in $[0, 1]$. Instead of considering a specific propagation model, and modelling its impact on the achievable channel speeds, for each user we assigned channel speed randomly, assuming speeds to be uniformly distributed among the set of available rates in Table I. We have set the parameter ϵ for the branch-and-bound algorithm to 5%. Furthermore, for a given number of users N, we considered a number of starting points for the heuristic, i.e., a number of upper bounding networks, which scales with N. Indeed, as N grows, it increases also the solution space, as well as it does the space of possible values of the weights of the utility function, and of the channel rates. Empirically, and for the number of users considered in our evaluations, we have found that scaling the number of starting points as $\lceil \frac{N}{2} \rceil$ brought acceptable results in terms of output of the optimization and of computational complexity.

Overall, for each value of N we considered a total number of instances of our setting (i.e., a particular choice of starting points, set of weights and set of channel rates) sufficient to get a 95% confidence interval within the 15% of the value of the average utility U achieved. In any case, we never used less than 100 instances.

A first objective of our numerical evaluation has been to assess the performance of the proposed heuristic, which has been introduced as a computationally feasible approach to the problem of maximizing the (weighted) proportional fairness in the allocation of leaky bucket rates among ad hoc users. More specifically, we have tried to give an idea of how far are, on average, the solutions of our heuristic from the optimal values, in order to evaluate the impact of the approximations on which the heuristic is based.

To this end, for scenarios with a small number of nodes (for which an exhaustive search still brings to an acceptable computational complexity), we have compared the average log-utility derived through our heuristic with the one derived through exhaustive search over all the possible upper bounding networks and choosing a $\epsilon = 0\%$. From Table II we observe that, in the considered scenarios, the solutions from our heuristic bring a utility which is on average very close to the optimal values derived through exhaustive search. This suggests that the approximations on which our heuristic is based have an overall low impact on the optimality of the operating point derived.

In Fig. 3 we compare the average log-utility, together with the 95% confidence interval, from our heuristic and the one obtainable in the saturated scenario. Fig. 3 also contains the median of the utility U in the same cases. We can see how in all cases the average log-utility derived by optimizing (through our heuristic) over the set of operating points which are stable according to our method is always at least 18.43% larger than the one derived by assuming the system in saturation. Moreover, we see that the relative improvement brought over

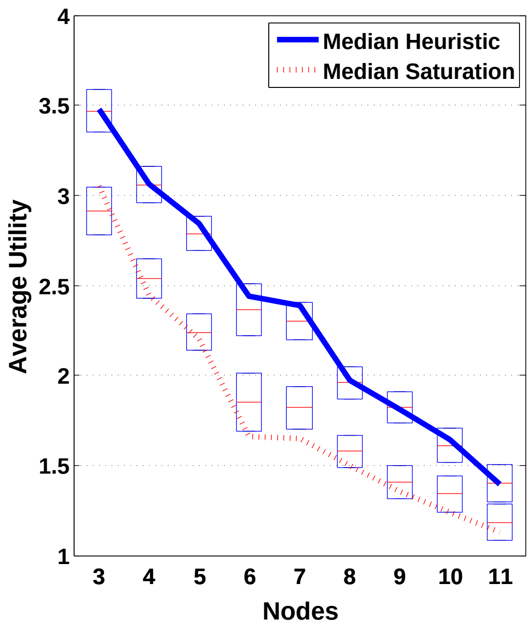


Fig. 3. Average and Median Log-utility. Heuristic vs. Saturation approx.

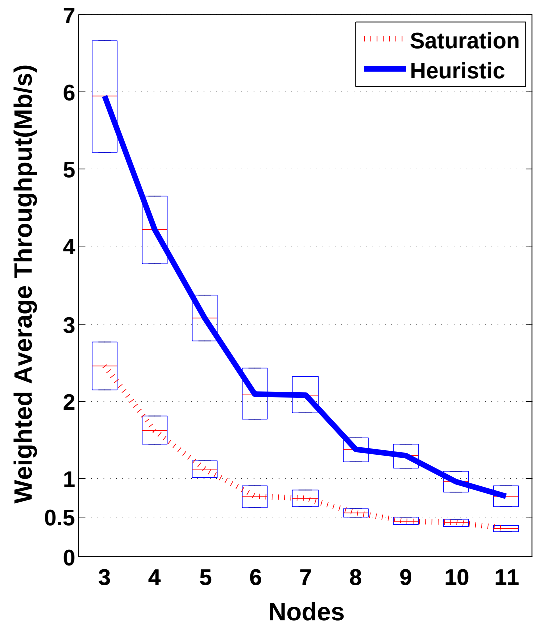


Fig. 4. Weighted Av. Throughput. Heuristic vs. Saturation Condition.

TABLE II. AVERAGE LOG-UTILITY: HEURISTIC VS. EXHAUSTIVE SEARCH

Opt. technique	3 Nodes	4 Nodes	5 Nodes	6 Nodes
Ex. Search	3.4690	3.0688	2.8034	2.3933
Heuristic	3.4690	3.0590	2.7879	2.3630
Difference	0	-0.32%	-0.55%	-1.27%
Max Difference	0	-6.56%	-10.64%	-15.88%

by our heuristic over the utility achieved under saturation assumption grows with the size of the scenario. The larger is the number of the stations in the system, indeed, the higher is the rate of contentions and, consequently, the inefficiency of the MAC under saturation assumptions. Applying leaky bucket controllers at the stations reduces the number of always contending stations and therefore improves, as this result confirms, the efficiency of the wireless channel. Nevertheless, it is interesting to note that for both methods, the optimal value of log-utility decreases when the number of nodes increases. Therefore, also when the heuristic is applied, the devices face an increase of the inefficiency of the MAC, even if in smaller scale.

As we can see from the evaluation of the median in Fig. 3, the difference between the optimal values of the utility function derived with these two methods is relevant also in distribution, and even larger than the one for the average.

In order to have a better idea of the difference between the operating points resulting from the heuristic and from the saturation assumption, we have compared them on the basis of the total average throughput, weighted in order to take into account the relative contribution of each host to the utility of the system. That is, the weights in these sums are the same as those adopted in the utility function.

The results are shown in Fig. 4, where the case under analysis is exactly the same used in Fig. 3. We see how our heuristic brings the system to an operating point for which the total average throughput is at least 114.43% higher than the total average throughput achieved under saturation assumptions. This shows how our heuristic derives system operating points that, besides maximizing the utility function, bring to a much more efficient utilization of network resources.

In order to understand the feasibility of the proposed approach, we analyzed the complexity of the heuristic we propose versus the exhaustive search of Table II. In Table III we present first the average number of networks the two methods analyze in order to get the final log-utility of the system. In case of the heuristic, we count the total number of upper bounding networks analysed, considering all the $\lceil \frac{N}{2} \rceil$ starting points. We also present the number of optimizations problems which the branch-and-bound method solves for each network. It can be easily proven that, in the worst case, the number of branches visited by the branch-and-bound for N nodes is $\sum_{i=1}^{N-1} 2^{N-i}$. Results are shown up to the point at which the comparison is computationally feasible (6 nodes), and for the largest scenario we analyzed through our heuristic (11 nodes). We can see how the heuristic requires a considerably inferior number of evaluations (both in terms of upper bounding networks, both in terms of branches), with a very limited impact on the optimality of the value of the log-utility derived (as seen in Table II). Please note that the different starting points of the heuristic are completely independent from each other. Therefore, the computational time can be reduced sensibly if the heuristic is evaluated in parallel. Finally we evaluate how the results change varying the number of starting points and the value of ϵ. Also here, we evaluate all the scenarios shown antecedently when 4 and 7 nodes were present in the network. The results presented in Table IV use the setting having $\epsilon = 5\%$ and $\lceil \frac{N}{2} \rceil$ starting points as a benchmark.

Even though the utility U remains almost the same in every configuration of the heuristic, the weighted average throughput changes sensibly. The most affecting parameter of the heuristic

TABLE III. COMPLEXITY OF THE HEURISTIC

	Average Number Network - Heuristic	Maximum Number Network ($N!$)
3 Nodes	6	6
4 Nodes	17.15	24
5 Nodes	41.04	120
6 Nodes	59.22	720
11 Nodes	409.05	39.9 10^6
	Av. Per-Network branches - Heu.	Av. Per-Network branches - Exhaustive Sear.
3 Nodes	4	4
4 Nodes	6.49	8.32
5 Nodes	9.47	12.87
6 Nodes	11.89	17.80
11 Nodes	29.11	-

TABLE IV. DEPENDENCE OF RESULTS ON ϵ AND ON NUMBER OF STARTING POINTS (SP) HEURISTIC

	4 Nodes	7 Nodes
U, $\epsilon = 5\%$, $SP = \lceil \frac{N}{4} \rceil$	-0.64%	-0.92%
U, $\epsilon = 0\%$, $SP = \lceil \frac{N}{2} \rceil$	$+0.02\%$	$+0.47\%$
U, $\epsilon = 10\%$, $SP = \lceil \frac{N}{2} \rceil$	-0.02%	0%
U, $\epsilon = 5\%$, $SP = N$	$+0.24\%$	0.74%
Weight. Av. Th., $\epsilon = 5\%$, SP = $\lceil \frac{N}{4} \rceil$	-1.99%	-5.54%
Weight. Av. Th., $\epsilon = 0\%$, SP = $\lceil \frac{N}{2} \rceil$	$+0.11\%$	$+0.44\%$
Weight. Av. Th., $\epsilon = 10\%$, SP = $\lceil \frac{N}{2} \rceil$	-0.08%	0%
Weight. Av. Th., $\epsilon = 5\%$, SP = N	$+1.10\%$	$+3.78\%$

is clearly the number of starting points used. Increasing the number of starting points indeed, U and the weighted throughput increases accordingly. Unfortunately, the number of upper bounding networks increases 67.24% and 106.10%, respectively, leading to a computationally expensive resolution of the proposed optimization. On the other hand, considering as acceptable the complexity of the proposed heuristic, reducing the number of starting points results in an unnecessary lost in performance, that can become tricky when the number of nodes increases. The same reasoning applies when we evaluate the choice of ϵ: sometimes the difference of performance is too small to justify the decreasing of the value of ϵ. For instance, the reduction of branches analysed when $\epsilon = 10\%$ is of at most the 20.03%, while the accuracy of performance estimates remains similar. The presented heuristic represents, therefore, a good trade-off among complexity and performance.

VI. CONCLUSIONS

In the present paper, we have proposed a new analytical method for the analysis of ad hoc networks, valid for any number of nodes. Our method does not require the traffic to be Poissonian, nor to be stationary, but only to be constrained by a deterministic arrival curve. We have described a heuristic for the derivation of a stable and proportionally fair allocation of resources. Our approach can be easily expanded to a vast class of arrival curves. We plan of extending our work by expanding our method to model more complex and more realistic scenarios, to those cases in which the connectivity graph of the ad hoc network is not a full mesh, as well as to settings where coupling is also due to interference.

ACKNOWLEDGMENTS

This work has been partially funded by the EU Seventh Framework Programme under grant agreement n° 318115 (CROWD). The work has also been partially supported by the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

REFERENCES

[1] Matthew Andrews. Instability of FIFO in the permanent sessions model at arbitrarily small network loads. *ACM Transactions on Algorithms*, 5(3), 2009.

[2] Giuseppe Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE JSAC*, 18(3):535–547, 2000.

[3] Sem C. Borst, Matthieu Jonckheere, and Lasse Leskelä. Stability of parallel queueing systems with coupled service rates. *Discrete Event Dynamic Systems*, 18(4):447–472, 2008.

[4] Jean-Yves Le Boudec and Patrick Thiran. *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *LNCS*. Springer, 2001.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, USA, 2004.

[6] IEEE LAN/MAN Standards Committee et al. Ieee 802.11-wireless lan medium access control (mac) and physical layer (phy) specifications. In *IEEE, June*, 2007.

[7] F. Daneshgaran, M. Laddomada, F. Mesiti, and M. Mondin. Unsaturated throughput analysis of ieee 802.11 in presence of non ideal transmission channel and capture effects. *Wireless Communications, IEEE Transactions on*, 7(4):1276–1286, April 2008.

[8] Markus Fidler and Jens B. Schmitt. On the way to a distributed systems calculus: an end-to-end network calculus with data scaling. *SIGMETRICS*, 34(1):287–298, jun 2006.

[9] A. De la Oliva, A. Banchs, P. Serrano, and F. A. Zdarsky. Providing throughput guarantees in heterogeneous wireless mesh networks. *Wirel. Commun. Mob. Comput.*, 2012.

[10] D. Malone, K. Duffy, and D. Leith. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *Networking, IEEE/ACM Transactions on*, 15(1):159–172, Feb 2007.

[11] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, pages 226–244, 1995.

[12] Aldo Vecchietti, Sangbum Lee, and Ignacio E. Grossmann. Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers and Chemical Engineering*, 27(3):433–448, 2003.

[13] Christian Vitale, Gianluca Rizzo, and Balaji Rengarajan. *Performance Bounds in Coupled Processor Systems*, 2013. Available at *http://publications.hevs.ch/index.php/attachments/single/665*.