Performance Bounds in Coupled Processor Systems

Christian Vitale
Institute IMDEA Networks, Madrid, Spain
UC3M, Madrid, Spain
Email: christian.vitale@imdea.org

***Abstract*—We consider queuing systems with coupled processors, where the service rate at each queue depends on the set of active queues in the system. The coupled-processors model arises naturally in the study of systems where a resource is shared by several classes of customers. For instance, its application has been suggested to model the complex interdependence that is present in wireless scenarios.**

In general, the queue lengths of such systems, that we call Coupled Processor Systems (CPSs), are modelled through complex Markov Chains whose steady state distributions are known only for two-queue systems. In this paper we propose a new approach for the study of the performance of the CPSs, based on worst case analysis. Our method exploits Network Calculus results over a set of networks whose stability implies the stability of the CPS, and allows to derive sufficient conditions for the stability of a CPS, as well as to compute bounds to queue size and packet delay. We also apply the CPS model to an ad-hoc $IEEE$ 802.11 scenario, where we present a method to derive practical results, and where we show numerically that considering the characteristics of the incoming flows at the nodes improves substantially the resource allocation in the system.

I. Introduction

We consider a system of parallel queues, for which the service rate that the incoming traffic receives depends on the set of nonempty queues in the system. Such model, known in the literature as "Coupled Processors System" (CPS) [1], arises from several contexts, in which coupling typically derives from sharing of a common resource. In wireless communications, CPS models the effects on performance of the complex interdependence among terminals due to the shared medium, as well as to interference [2]. For instance, the downlink throughput of mobile devices in cellular networks is strictly related to the set of active neighbouring BSs at any given time. Therefore, a CPS would be able to model the effect on performances of base stations service time. Furthermore, to model such type of scenarios is becoming of utmost importance, due to the fact that the increase in capacity demand is bringing up base stations density. In LTE, for instance, the use of a low reuse factor has been proposed in order to increase the capacity for user [3]. Moreover, short-ranged, low-cost, and in-band points of access, referred to as femtocells, are typically deployed alongside the normal coverage of macrocells base stations.

Another application of the coupled processors model is in the analysis of the performance of Generalized Processor Sharing (GPS) schedulers [4]. A GPS node has a dedicated queue for each class of traffic at its input. At every instant, its service is divided among the nonempty queues, according to some fixed weights. Therefore, the service rate each queue receives at the GPS can be univocally gathered from the set of not-empty queues and can be easily modeled by a CPS, with one processor for each GPS queue. In this case, the coupling derives from the dependency of the service offered to each queue on the occupancy of all queues, and not from interference as in the previous case.

In the present work, we consider coupled processors systems where the service rate of each queue is a decreasing function of the number of active queues in the system. This class of CPSs applies to many problems of practical importance, including all of the above mentioned examples. In the downlink of a wireless cellular network, the interference experienced by mobile devices monotonically increases when the set of transmitting base stations enlarges. Similarly, in a GPS node, when the number of empty queues decreases, the service rate offered to each of the previously active queues decreases too. A trivial way of analyzing the system would be to assume a static, worst-case scenario, where service rates are those of a system in saturation, with all queues active. This would lead to heavily pessimistic bounds and results, often of little practical interest. Research efforts have focused therefore on the (very difficult) problem of capturing the effects of system dynamics on its performance. [5; 6] derived closed-form necessary and sufficient conditions for the stability of a CPS composed by two queues with one class of traffic each. Both works assume Poisson arrivals and exponential service times. [7] derives a similar result, assuming long tailed distribution for file sizes. Another work of Borst [8] provides a method to know if a particular configuration of the system is stable or not, based on the use of the steady-state probability of related Markov chains, derived through computationally expensive simulations, and under poissonian traffic assumption. In particular, if the system is composed by N queues, in the worst case $N!$ are the Markov chains to be analyzed to know if a given point belongs to the stability region. [9] applies this method to a scenario with three queues and one single class of traffic for each of them. The authors of [1] focus on the stochastic characterization of valid bounds on the moment generating function for the queue lengths of the CPS and [10] proposes approximation methods for the derivation of the stability conditions, considering all queues but one in worst case coupling. As it is clear from the state of the art, deriving nontrivial performance bounds for CPS is still an open problem. Moreover, all available results depend factorially or exponentially on the number of queues.

In the present paper, we propose a method for the perfor-

mance analysis of a generic N-queues CPS, deriving sufficient conditions for stability, and hard bounds on backlog and delay at each queue. Our method assumes that input traffic can be modeled as leaky bucket constrained, and it is based on the derivation, for a given CPS, of a set of networks which can be analyzed using standard Network Calculus results, and whose performance bounds hold also for the CPS. The main contributions of this paper are:

- We derive a method for computing sufficient stability conditions for a generic N-queues CPS;
- For a simple two queues CPS, we propose a comparison among the sufficient stability conditions we gather from our method and the only other one present in the state of the art, i.e. [8];
- We show how to derive hard bounds on backlog at each queue and, when they are FIFO, on maximum packet delay;
- For a N-queues system, we show on a $IEEE$ 802.11 ad-hoc scenario a method for deriving practical results which are computationally feasible, based on exploiting the characteristics of the problem given. The numerical assessment of our results shows that considering the dynamics of the incoming traffic at a CPS helps improving substantially the resource allocation of the system.

The paper is organized as follows. In Section II we present the system model used in this paper. In Section III we introduce some important Network Calculus concepts. Section IV presents our method for the derivation of performance bounds in coupled processors systems. In Section V we show how to derive practical results on a wireless ad hoc scenario, assessing their performance on some numerical examples. We conclude our paper and discuss future research directions in Section VI.

II. System model

We consider a system of N parallel queues, where each queue receives traffic from one or more fresh sources. We assume such queues are served by work conserving schedulers. We define as the state of the system at a given time t the array $I(t) = (I_1(t), I_2(t), ..., I_N(t))$ where for each queue i, $I_i(t)$ is a binary variable which is equal to 0 if the $i-$th queue is empty at time t, and 1 otherwise. $\forall i$ we assume that the instantaneous service rate at that queue $R_i(t)$ is only function of the state of the system at that time, i.e., $R_i(t) = R_i(I(t))$. Note that, since the system states are finite, the transmission rates assumed by each queue are by definition a finite set too. We call such a system a *Coupled Processor System* (CPS).

Moreover, if I' and I'' are two different states of the system such that $\forall i\ I'_i \leq I''_i$, then $\forall i,\ R_i(I') \geq R_i(I'')$. Without loss of generality, we assume arrivals to be *packetized*, with a finite number of packet sizes. We consider that no losses are present at queues (buffers of infinite capacity). We assume the traffic from sources outside the system to be constrained by a *leaky bucket arrival curve* [11]. This implies that if $A(t)$ is the cumulative arrival function for a given traffic source for the interval $[0, t]$, than for any time interval (t_1, t_2) with $t_1 \leq t_2$, $A(t_2) - A(t_1) \leq \rho(t_2 - t_1) + \sigma$, where ρ, σ are the *leaky bucket parameters* for the considered source [11]. This assumption does not limit the applicability of our analysis. Indeed, in practical settings any source is constrained by some form of leaky bucket arrival curve, possibly by means of some conservative assumptions on the statistics of the traffic (e.g., burstiness of the flow).

An example of our system model for a simple three-queues CPS is given in Fig. 1, where R_1, R_2 and R_3 represent, respectively, the set of possible service rates $Queue\ 1$, $Queue\ 2$ and $Queue\ 3$ receive, depending on the state of the system.

III. Network Calculus Building Blocks

Network Calculus is a min-plus system theory for deterministic performance analysis of a queuing system. It provides tools for the derivation of hard bounds for backlog and for packet delay at each queue. The concept of arrival curve already introduced belongs to the Network Calculus framework. In this section we present some results we use in the paper. For a more complete treatment, refer to [11].

Definition III.1 (Service Curve, Backlog and Virtual Delay)**.** *If $A(t)$ and $A'(t)$ are the cumulative arrival functions respectively, at the input and at the output of a node, any $\beta(t) \in \mathcal{F}$ (the set of increasing functions passing from the origin) which satisfies*

$$A'(t) \geq \inf_{s \in [0,t)} \{A(t-s) + \beta(s)\} = (A \otimes \beta)(t)$$

is said to be a minimum service curve for the node. The operator $\otimes$ is the min-plus convolution operator. The backlog $v(t)$ and the virtual delay $h(t)$ at that node at time t are given by:

$$v(t) = A(t) - A'(t);$$

$$h(t) = \inf_{s \geq 0} \{A(t) \leq A'(t+s)\}.$$

The virtual delay at time t is the delay that would be experienced by a bit arriving at time t if all bits received before it are served before it. If the node is FIFO, virtual delay corresponds to the delay which such bit would actually experience. By exploiting the knowledge of an arrival curve for the whole of the traffic at the input of a node and of the service curve of the node, it is possible to derive upper bounds for backlog and delay at that node, by means of the following result:

Theorem III.1 (Performance Bounds)**.** *Consider a node with service curve $\beta(t)$, and let $\gamma(t)$ be an arrival curve for the aggregate of the input traffic. An arrival curve for the aggregate of the output traffic at the node is given by:*

$$\gamma'(t) = \sup_{s \in [0,t)} \{\gamma(t+s) - \beta(s)\} = (\gamma \oslash \beta)(t).$$

$\oslash$ is the min-plus deconvolution operator. An upper bound for the backlog of the node at any time t is given by $(\gamma \oslash \beta)(0)$,

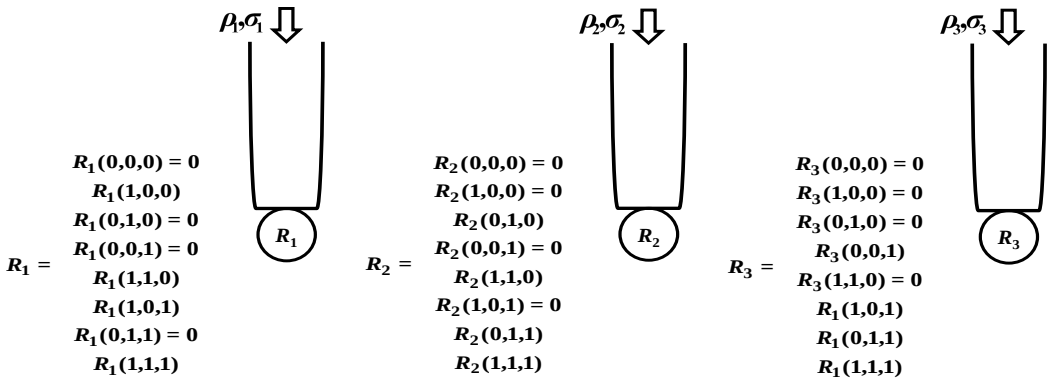


Fig. 1. Three-queues CPS example.

i.e., by the maximum vertical deviation between service curve and arrival curve. Moreover, an upper bound to the virtual delay at any time t is given by $\inf_{t\geq 0}\{\gamma \oslash \beta)(-t) \leq 0\}$, i.e., by the maximum horizontal deviation between service curve and arrival curve.

Due to the fact that the virtual delay bound is a valid bound for each of the bits of the arrivals, when the arrivals are packetized, as we assumed they are in Section II, it holds also for the last bit served of each packet. Therefore, if the node is FIFO and the arrivals are packetized, the virtual delay is a valid packet delay bound.

We also present the *continuous data scaling block*, originally introduced in [12]. We use a slightly different version of it:

Definition III.2 (Continuous Data Scaling Block)**.** *Let $\alpha(t)$ be the instantaneous arrival rate at a node. The node is a Continuous Data Scaling Block with scaling value $S \in \mathbb{R}^+$ if at its output the instantaneous departure rate is $S \cdot \alpha(t)$.*

The data scaling block makes it possible to model transformation processes which alter not only the timing of data arrivals, but also the total amount of data arriving, like data processing, encoding/decoding, and so on. Another concept we use is the one of *policer*:

Definition III.3 (Policer)**.** *A policer with policing function $\gamma(t) \in \mathcal{F}$ is a processing device such that, for any arbitrary input traffic, it forces $\gamma(t)$ as the arrival curve for the output traffic.*

Note that the definition does not specify what happens to the part of the input traffic exceeding $\gamma(t)$. In what follows we consider unbuffered policers, which discard non-conformant traffic.

IV. Performance Bounds for Coupled Processors Systems

A. A method for derivation of performance bounds

In this section we introduce a method for the derivation of performance bounds of a CPS. A primary issue in coupled processors systems is to determine what are the sufficient conditions for the system to be stable, in order to be able to derive hard bounds on packet delay and on backlog. We adopt the following definition of stability:

Definition IV.1 (Stability)**.** *Let us consider a system of N queues, and for every queue $i \in [1, ..., N]$ and any time $t \geq 0$ let $b_i(t)$ be the backlog at queue i at time t. The system is stable if $\forall i \in [1, ..., N]$ it exists a finite $\Gamma_i > 0$ such that for any arrival pattern from traffic sources, $sup_{t\geq 0} b_i(t) \leq \Gamma_i$*

In general, sufficient conditions for stability of such systems imply some constraints on the fresh traffic arrivals (in our case, on the leaky bucket parameters) and on the network (service rates, etc). Given the difficulty in analyzing directly CPSs, our method is based on the derivation, from a given CPS, of an auxiliary system whose stability implies the stability of the original CPS (we say that such a system *upper bounds* the CPS), and such that it can be analyzed using standard Network Calculus techniques. The following theorem defines a sufficient condition for this auxiliary system to upper bound a CPS:

Theorem IV.1 (Sufficient conditions for upper bounding)**.** *Let us consider a CPS with N queues, and another system with $N' \geq N$ queues, such that there is a one-to-one mapping between all queues of the CPS and a subset of N queues of the other system. The mapping is such that each queue of the CPS corresponds to a queue in the second system, with the same arrivals at any time t as the corresponding queue in the CPS. If at any time t the service rate at each queue of the CPS is larger than or equal to the one at the corresponding queue of the other system, the latter upper bounds the CPS.*

Proof. Due to the fact that the service rate of the queues belonging to the CPS is always larger than the service rate of the corresponding queue of the system taken into account, and due to the fact that the arrivals at such queues are exactly the same, it can be easily proved that the queues of the system contains at any time t a larger quantity of traffic than the corresponding queues of the CPS. Then, a hard bound on the backlog of the system introduced implies a hard bound on the queues of the CPS, i.e. its stability. □

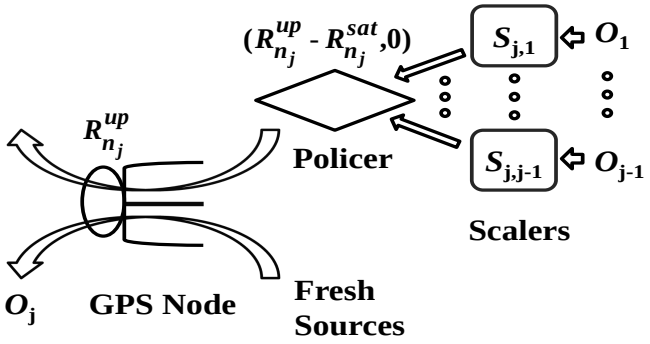


Fig. 2. Structure of the j-th layer of an upper bounding feed-forward system.

B. Derivation of an upper bounding system

In this section we show how to derive, from a generic N-queue CPS, an upper bounding system satisfying the hypothesis of Theorem IV.1. The system we derive is a network of N GPS nodes, each having two different queues. For each of these GPS nodes, one of the two queues is in a one-to-one mapping with one of the N queues of the CPS as described in Theorem IV.1, i.e., the arrivals at the two queues are exactly the same for any given time t. The second queue instead, as we will show shortly, acts in a way it allows to the other queue of the GPS node to respect the condition Theorem IV.1 imposes, i.e., that its service rate is smaller or equal to the service rate of the mapped queue of the CPS, at any given time t. Before going into details, we first describe the intuition behind the proposed derivation. Since we only consider CPSs where the effect of coupling brings to a decrease in the service rate of the affected queues, in the system we model these coupling effects reserving part of the capacity of the GPS node where the queue affected by coupling has been mapped, to its second queue. This is achieved by means of additional traffic coming from other nodes in the system. More specifically, every time a queue affects the service of another one in the given CPS, in the system there is some traffic going out from the node corresponding to the affecting queue in the CPS, and entering the node which corresponds to the affected queue in the CPS. As such system works at the *fluid limit* (i.e., traffic is served as if it was infinitely divisible), by appropriately modulating the additional traffic through shapers and data scaling blocks, we are able to obtain a system which satisfies the hypothesis of Theorem IV.1, and which therefore upperbounds the CPS.

The system we propose has a feed-forward topology and, for the sake of simplicity, we present it split into N "layers". In each of those layers, it is assigned one of the mentioned GPS nodes. Let us now denote each queue of the CPS with index i, and each node of the auxiliary system with index j, with $i, j \in \{1, ..., N\}$. Let $\mathbf{v} = (1, ..., N)$ be the array of all the indexes in increasing order. If $\mathbf{n}$ is one of the $N!$ possible permutations of $\mathbf{v}$, and if $\forall j$, n_j is the $j-th$ element of $\mathbf{n}$, then the n_j-th queue of the CPS is mapped into the $j-th$ layer of the system, $\forall j$. The structure of the generic j-th layer, with $j \in \{1, ..., N\}$, is shown in Fig. 2. It is composed, as we already said, by a work-conserving GPS node, with two queues and working at the fluid limit. One queue is dedicated to fresh arrivals, i.e., traffic coming from external sources, while the other receives only traffic coming from upper layers $(1, ..., j-1)$. Given the particular assignment $\mathbf{n}$ of the queues into the layers we decided to evaluate, we set the total service rate of the GPS node at layer j to $R^{up}_{n_j}$, i.e., the service rate of the n_j-th queue of the CPS when the set of CPS queues with nonempty queues is $(n_j, n_{j+1}, ..., n_N)$. If we indicate with $R^{sat}_{n_j}$ the service rate of queue n_j when all queues are active in the CPS, the GPS weights of the two queues are $w = \frac{R^{sat}_{n_j}}{R^{up}_{n_j}}$ and $w' = 1 - w$, respectively for the queue receiving traffic from fresh sources and from the other layers of the system. Therefore, the minimum service rate the GPS node reserves to the fresh traffic is $R^{sat}_{n_j}$, while its maximum service rate is $R^{up}_{n_j}$. Before entering the GPS node, traffic coming from the k-th upper layer, with $k \in [1, ..., j-1]$, passes through a *data scaling block* (see Section III), whose scaling value is $S_{j,k} \geq 0$. The aggregate of the traffic from upper layers is then fed into a *policer*, with zero burstiness and rate equal to the minimum service rate offered to this traffic at the GPS node, i.e., $R^{up}_{n_j} - R^{sat}_{n_j}$. The final outcome is that such aggregate traffic is served at the GPS node right away and its never queued. Hence, the effect of the coupling is accounted instantaneously and it is not deferred in time. At the output of the GPS node, traffic coming from upper layers exits the system, while the remaining traffic first enters a scaling block that multiplies it by a constant factor C equal to the number of the following layers, i.e., $C = N - j$, and then is split in equal parts and fed to each of the following layers. In this way, each of those layers receives from layer j an amount of traffic equal to its output flow.

As a consequence of the chosen architecture, the first layer of the system we introduced does not receive any additional traffic from the other nodes, and the service rate of its GPS node is the service rate that the queue of the CPS indexed by n_1 has when all the other queues are active. The node that has been assigned to the second layer receives additional traffic just from the GPS node at the first layer, while its service rate is the service rate of the corresponding queue of the CPS n_2 when all the queues of the CPS, except n_1, are active. This structure extends up to layer N, where the assigned GPS node receives additional traffic from all the other nodes in the system, while its service rate is the service rate of queue n_N of the CPS when n_N is the only active queue. Fig. 3 represents an example of the described system for the CPS of Fig. 1, when the assignment of the queues into the layers is $\mathbf{n} = (1, 2, 3)$.

Analogously to the CPS, we now introduce the state vector I_F, of length N, which at any given time t indicates with binary variables the state of the queues in the system which are dedicated to fresh sources. In the following we prove that through I_F it is possible to univocally determine the state of the system introduced.

Theorem IV.2. *It is given a CPS, and a system built as described, with a given set of scaling values* ***S****. If $I_F(t)$ is the*

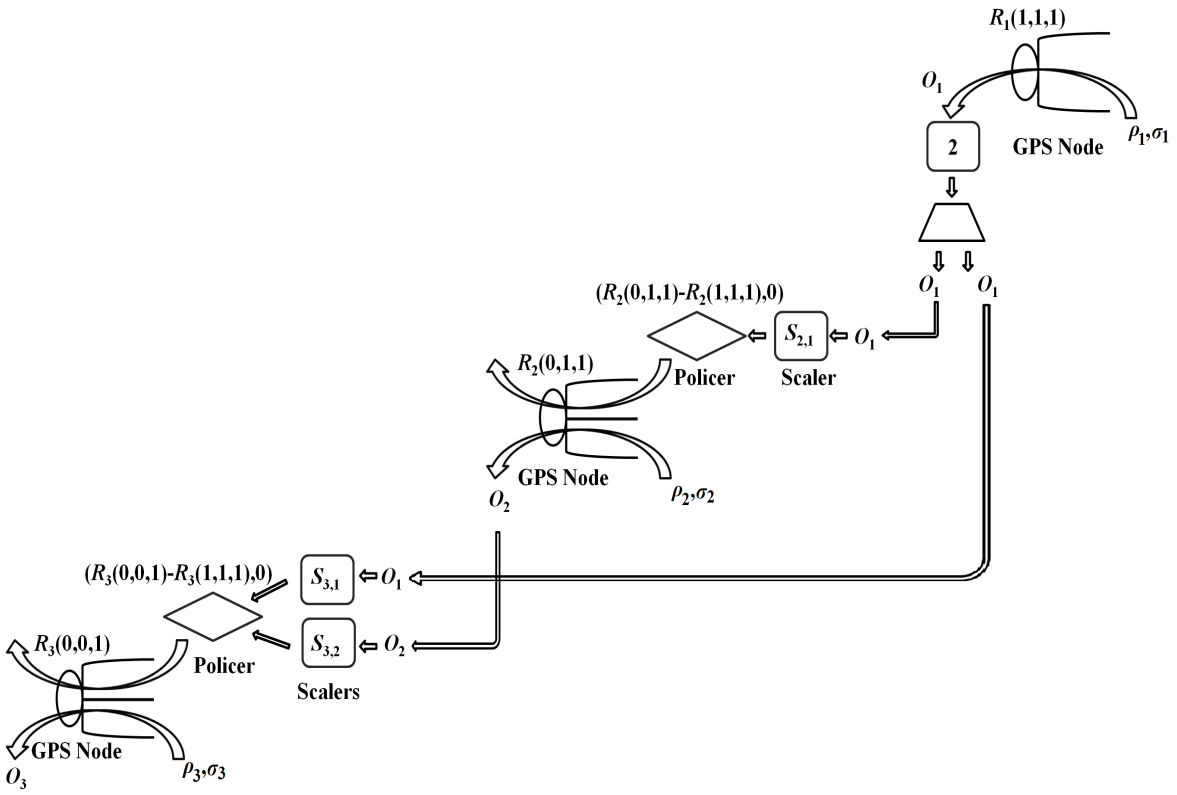


Fig. 3. A realization of the upperbounding system, three-queues case.

state vector of the system at a given time t, then the service rate that each active node j in the system reserves to its fresh traffic at time t, $O_j(t)$ satisfies

$$O_j(t) = R^{up}_{n_j} - \min\left(\sum_{p=1}^{j-1} O_p(t)S_{j,p}\cdot(I_F(t))_p, R^{up}_{n_j} - R^{sat}_{n_j}\right) \tag{1}$$

where $(I_F(t))_p$ is the p-th element of I_F.

For the proof see Section A of the Appendix. This result shows that the instantaneous service rate that each node reserves to its fresh traffic could be gathered recursively as a function of $I_F(t)$. Exploiting the result presented above, the following theorem defines sufficient conditions on the scaling values of the system for upperbounding the given CPS.

Theorem IV.3. *It is given a CPS, and a system, structured as described, whose state vector at time t is $I_F(t)$. Then the system upper bounds the CPS if, $\forall t \geq 0$, and for any node j which is active at time t, the set **S** of scaling values satisfies:*

$$\sum_{p=1}^{j-1} O_p(t)S_{j,p} \geq R^{up}_{n_j} - R^{up}_{n_j}(I_F(t)) \tag{2}$$

where $R^{up}_{n_j}(I_F(t))$ is the service rate for fresh traffic at queue n_j of the CPS, assuming that the queues of the CPS which are active are all those in $I_F(t)$, plus queues $n_{j+1}, ..., n_N$.

Proof. The system introduced and the CPS share the same set of states, i.e. I for the queues of the CPS and I_F for the fresh traffic queues in the system. If (2) is replaced into (1), we get that $O_j(t) \leq R^{up}_{n_j}(I_F(t)) \leq R_{n_j}(I_F(t))$, where $R_{n_j}(I_F(t))$ is the service rate of queue n_j of the CPS when the corresponding nodes active in $I_F(t)$ are active also in the CPS itself. In practice, (2) ensures that the service rate of each fresh source arrivals of the system is always inferior to the one of the corresponding queues in the CPS, when the two systems are analyzed in the same state $I = I_F$. We now prove that such property is sufficient for the introduced system to respect the hypothesis of Theorem IV.1. We prove it by contradiction. We imagine that the system satisfies the presented property, but at time t^* for the first time at least one of the queues dedicated to the fresh sources of the system is served faster than the corresponding queue of the CPS. At t^* the two systems are not in the same state, or due to (2) we contradict the definition of t^*. In order to exist, by construction, at least one of the active queues in the CPS should not be active in the system. If not, the set of active queues for fresh sources in the system is a superset of the queues active in the CPS, therefore their service rate is smaller or at most equal in all queues. Due to the fact that the arrivals at the queues of the CPS are exactly the same of the mapped nodes of the system, the queue that is empty in the system and not in the CPS should have been served faster, at least for an instant, in a previous moment of

t^*, which is a contradiction. □

Let us recall that $O_p(t) = 0$ if the node is not active and that $O_p(t)$ can be recursively computed from (1) for each possible state of the system we introduced. Let us also note that Theorem IV.3 does not require a constraint for all time instants t. The set of states in which the system could be is finite, and is given by all the possible configurations of $I_F(t)$. Therefore, imposing (2) for all possible states I_F is sufficient to respect (2) at any time t, so that the system upper bounds the original CPS.

C. Derivation of sufficient conditions for stability of a CPS

Given a set of descriptors $(\{\rho_1, \sigma_1\}, ..., \{\rho_N, \sigma_N\})$ for the arrivals of the CPS, one per queue, in this section we derive sufficient conditions on these leaky bucket parameters for the stability of the CPS, exploiting the properties of the upper bounding system.

Theorem IV.4. *Let us consider a CPS, with fresh arrivals constrained by leaky bucket arrival curves, with parameters* $(\{\rho_1, \sigma_1\}, ..., \{\rho_N, \sigma_N\})$. *Let us consider an upper bounding system for the CPS, derived as shown in Section IV-B, which satisfies Theorem IV.3 with a given set of scaling values* ***S****. If for any node* $j \in [1, ..., N]$ *of the system the leaky bucket rate satisfies:*

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

then the CPS is stable.

Proof. A GPS Node is stable if its capacity is at least as high as the sum of the long term rates of the arrivals [13]. Thanks to the fact that the system is a feed-forward network, let us assume the first $j-1$ layers of the system are stable. [14] shows that in such a case, valid arrival curves for each of the output flows coming to layer j from the upper layers $(1, ..., j-1)$ is still a leaky bucket curve with the same long term arrival rate ρ_i, with $i = \{1, ..., j-1\}$, of the corresponding incoming traffic. Considering that the sum of the leaky bucket characterizations of the first $j-1$ layers is a valid characterization of the incoming flow at the policer and that $(R^{up}_{n_j} - R^{sat}_{n_j}, 0)$ is its shaping function, (3) comes straightforward. □

Let us apply the results of Theorem IV.4 to the particular upper bounding system presented in Fig. 3, valid for the CPS of Fig. 1, where $S_{2,1}, S_{3,1}, S_{3,2}$ are fixed. In such a case, a set of descriptors for the arrivals $(\{\rho_1, \sigma_1\}, \{\rho_2, \sigma_2\}, (\rho_3, \sigma_3\})$ is stable if the following inequalities are satisfied:

$$\begin{cases} \rho_1 \leq R_1(1,1,1) \\ \rho_2 \leq \max(R_2(1,1,1), R_2(0,1,1) - S_{2,1} \cdot \rho_1) \\ \rho_3 \leq \max(R_3(1,1,1), R_3(0,0,1) - S_{3,1} \cdot \rho_1 - S_{3,2} \cdot \rho_2) \\ S_{2,1} \cdot R_1(1,1,1) \geq R_2(0,1,1) - R_2(1,1,1) \\ S_{3,1} \cdot R_1(1,1,1) \geq R_3(0,0,1) - R_3(1,0,1) \\ S_{3,2} \cdot R_2(0,1,1) \geq R_3(0,0,1) - R_3(0,1,1) \\ S_{3,1} \cdot R_1(1,1,1) + \\ \quad + S_{3,2}(\max(R_2(1,1,1), R_2(0,1,1) - S_{2,1} \cdot R_1(1,1,1))) \\ \quad \geq R_3(0,0,1) - R_3(1,1,1) \end{cases}$$

As we have seen, each upper bounding system is associated to a particular choice for the permutation vector **n** of the indexes of the CPS nodes, and for the array of scaling values **S**. Then a CPS is stable according to our method if among all systems associated to a given **S** and **n**, there exists at least one which satisfies Theorem IV.4. Note that the sufficient conditions for stability only involve the long term rates assumed by the arrivals. Therefore it is possible to draw the region in a N-dimensional space (one per node), of the set of long term rates of the leaky bucket characterizations for which the CPS is stable.

D. Bounds on Backlog and Virtual Delay

Given a CPS, and an upper bounding system characterized by the couple $(\mathbf{n}, \mathbf{S})$ which is stable according to Theorem IV.4, then it is possible to derive upper bounds to packet delay and to backlog at each queue for fresh sources of the upper bounding system. Such results are gathered by means of standard Network Calculus results.

By construction, at any time, any queue for the fresh sources of an upper bounding system built as in Section IV has always a larger backlog and a service rate which is not larger than that of the correspondent queue of the CPS. Therefore, bounds for delay and backlog for those queues in such an upper bounding system hold also for the corresponding queue in the CPS.

Theorem IV.5. *Let us consider a CPS and an upper bounding system which verifies the hypothesis of Theorem IV.4. Then a bound for backlog at each node* n_j *of the CPS is given by:*

$$\sigma^*_{n_j} = \begin{cases} \sigma_{n_j}, & \text{if } \rho_{n_j} \leq R^{sat}_{n_j}, \\ \sigma_{n_j} + \frac{(\rho_{n_j} - R^{sat}_{n_j}) \cdot (\sum_{k=1}^{j-1} S_{j,k} \cdot \sigma^*_{n_k})}{R^{up}_{n_j} - R^{sat}_{n_j} - \sum_{k=1}^{j-1} S_{j,k} \cdot \rho_{n_k}} & \text{otherwise.} \end{cases} \tag{4}$$

Moreover, if the nodes of the CPS are FIFO, a bound to packet delay at the same node is:

$$d_{n_j} = \begin{cases} \frac{\sigma_{n_j}}{R^{sat}_{n_j}}, & \text{if } \rho_{n_j} \leq R^{sat}_{n_j} \text{ and } \sigma_{n_j} \leq b_{n_j} \\ T_{n_j} + \frac{\sigma_{n_j}}{R^{up}_{n_j} - \sum_{k=1}^{j-1} S_{j,k} \cdot \rho_{n_k}}, & \text{if } \sigma_{n_j} > b_{n_j} \\ \frac{b_{n_j}}{R^{sat}_{n_j}} - \frac{b_{n_j} - \sigma_{n_j}}{\rho_{n_j}}, & \text{if } \rho_{n_j} > R^{sat}_{n_j} \text{ and } \sigma_{n_j} \leq b_{n_j}. \end{cases}$$

with:

$$b_{n_j} = \begin{cases} \infty, & \text{if } R_{n_j}^{sat} \geq R_{n_j}^{up} - \sum_{k=1}^{j-1} S_{j,k}\rho_{n_k}, \\ \frac{R_{n_j}^{sat} \sum_{k=1}^{j-1} S_{j,k}\sigma_{n_k}^*}{R_{n_j}^{up} - R_{n_j}^{sat} - \sum_{k=1}^{j-1} S_{j,k}\rho_{n_k}} & \text{otherwise.} \end{cases}$$

and:

$$T_{n_j} = \frac{\sum_{k=1}^{j-1} S_{j,k} \cdot \sigma_{n_k}^*}{R_{n_j}^{up} - \sum_{k=1}^{j-1} S_{j,k} \cdot \rho_{n_j}}$$

Proof. Given the fact that the set $\mathbf{S}$ allows to find a stable upper bounding system since it respects Theorem IV.4, through [14] it is possible to compute an arrival curve for the output traffic of the fresh sources at each of the first $j-1$ layers. At layer j, [14] also shows how to compute minimum service curve for the fresh traffic queue of the GPS node. Then it is easy to obtain the presented bounds by means of the results in Section III. □

For each of the possible assignment of the queues of the CPS into the layers and for each of the possible scaling values, i.e., for each couple $(\mathbf{S}, \mathbf{n})$, which leads to a stable upper bounding system, the bounds we could gather exploiting Theorem IV.5 are valid bounds for the original CPS. Therefore, the minimum bounds on delay and backlog we could gather analysing the whole set of couples $(\mathbf{S}, \mathbf{n})$ are also the best possible bounds achievable by means of our method. By the way, given a particular choice of $(\mathbf{S}, \mathbf{n})$, we do not consider in their computation the fact that the traffic flows coming from the upper layers $j-1$, could have be limited by the link capacity of the GPS nodes [11]. Therefore, please note that the bound introduced could be improved by means of increased complexity.

E. Relationship with [8]

In this section we investigate the relationship between our method and the one described in [8], on a setting with two queues (for larger number of queues, a very high number of simulations would be needed to apply the result in [8]). In particular, we compare our sufficient conditions for stability with the necessary and sufficient conditions in [8], when independent Poisson arrival process with $\{\lambda_1, \lambda_2\}$ as parameters and exponentially distributed lengths for the arrivals with parameter $\{\mu_1, \mu_2\}$ are assumed for the system. As the two methods relay on different assumptions for the input traffic, and on different notions of stability, no rigorous comparison is possible. By the way, it is possible to get an intuition of how they relate to each other by using a stochastic network calculus formulation.

Recalling the notation already used, R_i, with $i \in [1, 2]$, is the service rate of queue i of the CPS if the other queue of the CPS is not active while R_i^{sat} is the service rate node i reserves to its own queue when both the nodes of the CPS are active. Applying the results of [8], necessary and sufficient conditions for stability are:

$$\begin{cases} \frac{\lambda_1}{\mu_1} \leq R_1^{sat} \\ \frac{\lambda_2}{\mu_2} \leq R_2 - \frac{\lambda_1}{\mu_1} \cdot \frac{R_2 - R_2^{sat}}{R_1^{sat}} \end{cases}$$

$$\begin{cases} \frac{\lambda_2}{\mu_2} \leq R_2^{sat} \\ \frac{\lambda_1}{\mu_1} \leq R_1 - \frac{\lambda_2}{\mu_2} \cdot \frac{R_2 - R_1^{sat}}{R_2^{sat}} \end{cases} \quad (5)$$

In order to have an idea of how this result compares with ours, we use the Stochastic Network Calculus characterization of the Poisson inputs from [15]. If $A(t)$ is the cumulative arrival function, then a stochastic arrival curve can be defined for the inputs, with long term rate $\rho_i' = \frac{\lambda_i}{\mu_i - \delta}$, violation probability p, and burstiness $\sigma_i' = \frac{ln(p)}{-\delta}$, such that for any time interval $[s, t]$, with $s, t \geq 0$, it holds:

$$Pr\left(A(t) - A(s) > \frac{\lambda}{\mu - \delta} \cdot (t - s) + \frac{ln(p)}{-\delta}\right) \leq p.$$

with $\delta < \mu_i$. Now let us consider the same 2-queues CPS, whose input are constrained by deterministic leaky bucket, with parameters (σ_i', ρ_i'), and let us consider the two upper bounding systems we can build with our method. By applying Theorem IV.4, our sufficient conditions for stability are:

$$\begin{cases} \frac{\lambda_1}{\mu_1 - \delta} \leq R_1^{sat} \\ \frac{\lambda_2}{\mu_2 - \delta} \leq R_2 - \frac{\lambda_1}{\mu_1 - \delta} \cdot \frac{R_2 - R_2^{sat}}{R_1^{sat}} \end{cases}$$

and

$$\begin{cases} \frac{\lambda_2}{\mu_2 - \delta} \leq R_2^{sat} \\ \frac{\lambda_1}{\mu_1 - \delta} \leq R_1 - \frac{\lambda_2}{\mu_2 - \delta} \cdot \frac{R_2 - R_1^{sat}}{R_2^{sat}}. \end{cases}$$

Therefore, we can see that as the parameters δ and p of the stochastic arrival curve tend to zero, the system with deterministic leaky bucket inputs approximates better and better the one with Poisson arrivals, and our sufficient stability conditions tend towards those obtained with [8].

V. Application on a Wireless Scenario

As we have seen, using our method to find whether an N-queue CPS is stable for a given set of leaky bucket constraints for the inputs implies a search over all the couples $(\mathbf{S}, \mathbf{n})$ which satisfy Theorem IV.4. In the worst case, this means solving for each of the $N!$ upper bounding systems a feasibility problem over the scaling values $\mathbf{S}$, with a number of constraints which is exponential in N. Hence, the direct application of the results in Section IV is computationally infeasible for scenarios with more than a few queues. By the way, its formulation is such that it can be easily simplified for deriving practical results.

In this section we describe a practical method for deriving performance bounds, based on an approximation technique which adapts to the characteristics of the CPS and of the specific performance problem to solve. We present our method by applying it on a classical problem in wireless LANs.

A. Description of the scenario

We consider a scenario with N hosts communicating among them via the $IEEE$ 802.11 protocol in ad hoc mode, and exchanging files through an application. We assume all hosts are in range of each other, i.e., no packet relaying is needed. We assume hosts do not suffer for interference and that they do not move, so that their capacity to each other does not change over time. We also assume that, if no traffic shaping is done

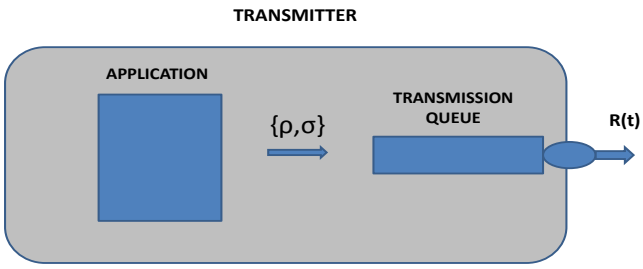


Fig. 4. Sketched architecture of the Transmitter

at the application, each host always has something to send. In such a setting the system is in saturation condition, and the average throughput achieved by each host could be computed applying the results of the well known paper of Bianchi [16]. By the way, giving each host the same probability to access the transmission medium often results in some pathological conditions, where, for instance, few users with a poor channel jeopardize the medium, at the expense of all the other hosts [17]. Moreover, due to the intrinsic packet-fairness of the $IEEE$ 802.11 protocol, it would not even be possible to distinguish hosts into classes, with different service levels. In these cases, a better allocation of system resources could be achieved by letting the application at each host impose some limit to the traffic sent to the network, and tune the operating point of the system in order to optimize a given utility function. In our example, we assume that this limitation on traffic is implemented at each host via a leaky bucket controller, which buffers non conformant traffic produced by the application. Fig. 4 represent the mentioned architecture of the transmitter.

Introducing the leaky bucket controllers to a scenario that could work in saturation, and ensuring stability for the offered traffic to the transmission cards, let the long term rates of the controllers be also the throughput achieved by each couple of transmitter/receiver. Therefore, the presence of leaky bucket controllers, which ensures the stability of the backlog of the transmission queues in the wireless scenario, forces the system to work in non-saturation condition. Many of the available results for non-saturated conditions do not capture the effects of the dynamics of the input traffic on the performance of the system. [18] 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 the dynamics of the traffic do alter significantly such quantities over time.

Here we propose a different approach, which tries to model more accurately the effect of the dynamics of the input traffic over the performance of the system. Indeed, we approximate the introduced wireless scenario by means of a CPS, having a queue for each of the hosts present. We characterize the underlying CPS identifying the state of the wireless system, at each time, by the set of nonempty transmitters. For each possible state of the system we compute the throughput just for the active hosts as in Equation (4) of [18], while we set to 0 the throughput of the rest of the hosts not included in the set under analysis. Please note that the underlying CPS we gather respects the monotonicity property we assumed for its formulation. Indeed, when a superset of the initial group of hosts is analysed, the service rate of each of them, computed as described, decreases.

By means of the approximation of the WiFi scenario through a CPS, we assume that the dynamics of the system due to the CSMA/CA mechanisms take place on a smaller time scale than the dynamics of the traffic from the application, so that they can be adequately modeled through their average effect on the system. By the way, due to the fact that the delay each packet suffers is strictly related to the CSMA/CA mechanism, the same approximation does not hold for the proposed delay/backlog bounds. A validation of the proposed approximation is given in Section V-D.

The utility function that we choose in our example is a weighted fairness function [19], which is one possible way of balancing some notion of fairness among users with, for instance, different classes of service. Its expression, for the case of N hosts, is:

$$U = \sum_{i=1}^{N} w_i \cdot log\left(\frac{\rho_i}{\rho_0}\right). \tag{6}$$

where we assume ρ_0 is the minimum bit rate for an acceptable performance for the application, independently from the traffic class.

B. Setting the scaling values

The problem consists therefore in finding a set of leaky bucket parameters for the WiFi scenario which maximizes our utility function, while ensuring that the system is stable. We now show how our method can be adapted to the particular application we want to analyze.

In this case we present the set of systems that upper bounds the CPS setting the scaling vector $\mathbf{S}$ to some fixed value. We start by proposing a new sufficient condition for the system with the structure described in Section IV-B to upper bound its associated CPS:

Theorem V.1. *It is given a CPS, and an upper bounding system built as in Section IV-B that is associated to a given couple* $(\boldsymbol{S}, \boldsymbol{n})$. *If for any layer* j *and for any* $p \in [1, ..., j-1]$ *the following holds:*

$$S_{j,p} = \frac{R^{up}_{n_j} - R^{p-up}_{n_j}}{R^{up}_{n_p}} \tag{7}$$

where $R^{p-up}_{n_j}$ *is the service rate at the CPS queue* n_j *when the set of CPS queues* $[n_p, ..., n_N]$ *is active, then the system satisfies the hypothesis of Theorem IV.3 and it is therefore an upper bounding system for the CPS.*

Proof. Let us assume a generic system state $I_F(t)$ for the system introduced and k be the first layer whose $(I_F(t))_k = 1$.

Substituting (7) into (2) for a generic layer j we can obtain:

$$\sum_{p=1}^{j-1} O_p(t)S_{j,p} \geq O_k(t)S_{j,k} = R_{n_j}^{up} - R_{n_j}^{k-up}, \qquad (8)$$

that satisfies the upper bounding condition Theorem IV.3 imposes by construction. □

Theorem V.1 allows to greatly reduce the complexity of our method, by assigning a constant value to the scaling vector **S**. Let us show what is the intuition behind the choice of scaling values in this result. Let us consider layer j and a state vector I_F for the system, whose first active queue is at layer $k < j$. Whatever is the subset of active queues in layers $[k, ..., j-1]$, this choice of scaling values makes the traffic coming from layer k alone reduce the service rate at layer j at the value it would have if all the queues at layers $[k, ..., j-1]$ were active. The effect of this approximation is mitigated by the fact that in our method we use different upper bounding systems, associated to different assignments of CPS queues to layers. In case of all the scenarios with three couples transmitter/receiver used in Section V-D, we compared the sufficient stability conditions derived with Theorem V.1 with the one obtained with the original formulation of the method, through Montecarlo simulation. Fig. 5 represent the case where the it has been registered the maximum difference among the volumes of the two regions (9.41%).

C. Optimizing over the set of upper bounding systems

Given a CPS, and the $N!$ upper bounding systems derived with our method, and whose scaling values are given by Theorem V.1, we can formalize our problem as an optimization problem over these $N!$ systems. In this section we first formulate our optimization problem, and then we present a heuristic which searches for the optimum over a suitable subset of the possible upper bounding systems.

1) Formulation of the optimization problem: Our optimization problem can be formulated as follows:

$$\underset{\rho,\mathbf{n}}{\text{maximize}} \quad U = \sum_{i=1}^{N} w_i \cdot log(\frac{\rho_{n_i}}{\rho_0})$$

subject to

$$\forall j = 1, \ldots, N$$

$$\rho_{n_j} + min(R_{n_j}^{up} - R_{n_j}^{sat}, \sum_{p=1}^{j-1} \frac{R_{n_j}^{up} - R_{n_j}^{p-up}}{R_{n_p}^{up}} \cdot \rho_p) \leq R_{n_j}^{up}.$$

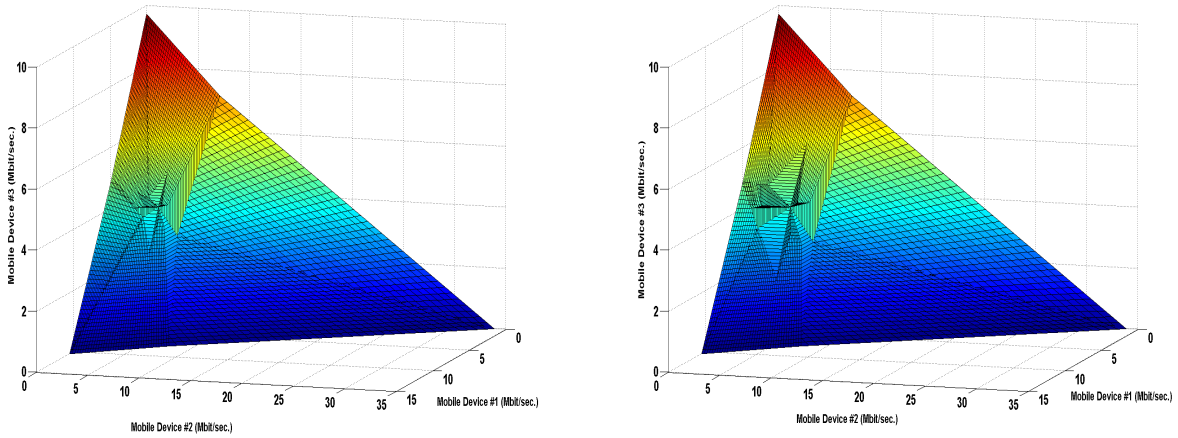
where the constraints derive from Theorem IV.4, and where the scaling values have been assigned according to Theorem V.1. The presence of the min function in the constraints leads to a non-convex feasibility region. In order to make this problem tractable, we use the so called Big-M transformation [20]. The two elements inside the min function are active disjointly. Therefore, we can gather from each of the constraints of the optimization problem two different constraints in which we add a binary variable multiplying a large constant value M. Whenever the binary variable is 1, the large constant makes the constraint useless because all the feasible ρ satisfy it, while when the binary variable is zero the constraint is active. Choosing a value for the binary variables we also choose a part of the solution space where we search for the optimal value. Thus the optimization problem belongs to the mixed-binary programming family. We solve it by means of a *branch-and-bound* method [21].

2) Derivation of an Heuristic: The approach we follow in order to derive a heuristic which is computationally feasible, is the following. First, in order to reduce the computation required by the optimization problem, on each of the systems gathered by the assignment of the queues of the CPS to the layers system we stop the evaluation when the lower bound and the upper bound for the optimum given by the branch-and-bound method are $\epsilon\%$ apart. This parameter can then be tuned according to the desired tradeoff between computational cost and performance of the heuristic.

The second approximation introduced is in the set of systems that we evaluate. Instead of evaluating all of the $N!$ systems, we start by choosing a small set of systems as starting points for our algorithm. For each of these systems, we identify the set of $N-1$ systems each obtained by swapping two queues assigned to two contiguous layers in the original system. Within this set, the system with the higher log-utility value is taken as the new reference system, and the whole process is repeated until a local maximum for the log-utility is found. The system corresponding to the largest local maximum is the solution of the optimization problem proposed. Obviously, the trade-off among computation and accuracy of the solution is given by the number of starting points we choose.

In order to identify the systems we use as starting points, we employ the following algorithm. To each queue j of the CPS, we associate the quantity $\frac{1}{w_j}$, which is the inverse of the weight assigned to the queue in the utility function U. Then, starting from the first layer of the upper bounding system, we assign a queue of the CPS to each layer with a probability proportional to this quantity. In this way, queues with a higher weight, and with a correspondingly higher contribution to the utility function U, are more likely to end up in the lower layers of the upper bounding system we choose to analyse. Indeed, due to the structure of the upper bounding system, the lower the layer a queue belongs to, the largest is the set of queues whose coupling is modeled by incoming traffic rather than via a fixed penalty on its service rate. As a result, we obtain a mapping between queues of the CPS and of the upper bounding system in which those coupling relationships involving queues with higher impact on the utility function tends to be modeled more accurately. Section V-D shows how this method for building the starting points improves considerably the efficiency of the heuristic.

Note that if the weights are all the same, the optimization tends towards a fair allocation of the resources. Due to the nature of WiFi, in that case the saturation throughput is exactly the same for all users (if packet lengths are the same for each host), and the heuristic always converges to such fair allocation of resources for each of the possible starting point.



(a) Original Method

(b) Approximation

Fig. 5. Sufficient stability conditions in the worst case Approximation

D. Numerical Evaluation

1) Validation of the assumptions: Before evaluating how the knowledge of sufficient conditions for stability could improve the assignment of the resources to the hosts of the scenario introduced, we first assess the approximation we propose, i.e., the modeling of a WiFi environment by means of a CPS.

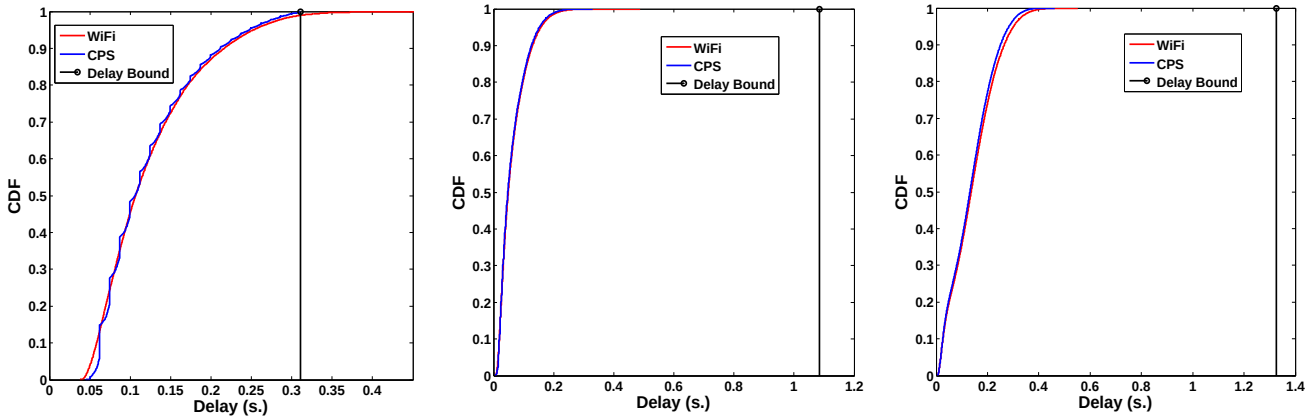
First, we simulated WiFi scenarios with a small number of hosts, and we compared the throughput achieved by the hosts in each state with the one computed as described in Section V-A. The main goal is to show that the dynamics introduced by the non-saturation conditions do not affect the asymptotic service rates that the system should have in each state. In order to perform the validation, the was assigned to system states on a time slot basis. Therefore, if an arrival occurred during a time slot, the system state was updated in the following one. We simulated 10 three-queues cases and on 2 four-queues cases. As transmission parameters we used the one of Table I, while as input traffic we used chunk of data respecting in each case the leaky bucket parameters that, according to our heuristic, optimize the utility function U we introduced in Section V. The possible lengths of those chunks of data were a finite set, since we assumed they could be just a multiple of the packet length. To be more precise, they could be a randomly chosen number of packets among 5 and 25 [22]. In the worst case among the ones evaluated, the average difference between the expected throughput and the throughput experienced by the hosts was of the 4.7%, with no significant difference among the cases with 3 and 4 queues.

Then, we picked the particular case where the difference among the expected behaviour of the WiFi environment and the simulated one was the largest (a 3 queue case). In this scenario, we evaluated the empirical delay distribution of the chunks for the WiFi environment and, by means of simulations, the empirical delay distribution of the underlying CPS we used to evaluate his sufficient conditions for stability. Fig. 6 shows how close the two distributions are for each queue analysed. Fig. 6(a) clearly shows that the probability of having small chunks arrivals at $Queue\ 1$ of the CPS when $Queue\ 1$ is the only empty queue in the system, and leaving the system while the other two queues are still active, is larger than zero.

In Fig. 6, we also show the delay bound computed as reported in Section IV-D. In order to perform the computation, the whole set of upper bounding systems and of the feasible values for the scalers was used, and the minimum of the delay bound achieved was taken as the delay bound of the CPS. Even if we introduced an approximation in the computation of such bounds (see Section IV-D), $Queue\ 2$ and $Queue\ 3$ reach the 35% of the delay bounds, while some of the chunks of $Queue\ 1$ actually experience the expected maximum delay.

2) Heuristic's performance: In this section we assess numerically the performance of our heuristic, for the wireless scenario described. Results are compared to those obtainable via a static model of the system, which assumes all queues are always active, i.e., in saturated condition. We perform such analysis by varying the number of transmitter/receiver pairs in the WiFi scenario and, whenever feasible (i.e., for settings with less than 7 hosts), we compare such results also with the one obtainable by optimizing through exhaustive search over all the set of systems $\mathbf{n}$, fixing the set of scaling values using (7). The parameters used in the considered scenarios are still the one of Table.I.

We randomly assigned the channel speeds to each transmitter/receiver pair, and we evaluated the set of leaky bucket rates which maximizes the log-utility function. Note that the possible configurations of the CPS are finite, and that they can be computed off-line without affecting the time required to find the solution of the optimization problem. We repeated the evaluation at least one hundred times for each scenario. The parameter ϵ for the branch-and-bound algorithm is set to 5%, while the number of starting points is set to $\lceil \frac{N}{2} \rceil$.



(a) Node 1 (b) Node 2 (c) Node 3

Fig. 6. Delay distribution - CPS vs. WiFi

TABLE I
SET-UP OF THE WIRELESS SCENARIO

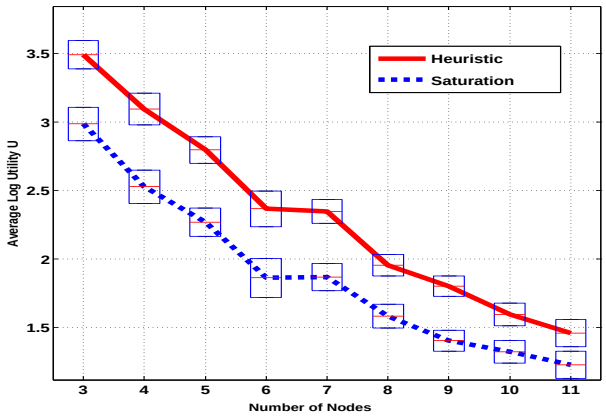
Available Channel Speeds (Mbit/s.)	1;2;5.5;6;9;11;12 18;24;36;48;54
Weights	Uniformly Distributed
CWmin	16
Backoff stages	5
Preamble + PHY header(μs)	20
SIFS (μs)	16
ACK Time (μs)	24
DIFS (μs)	34
Slot time (μs)	9
MAC header(bits)	224
Packet size (bits)	15000

First of all, in Table II we compare, for scenarios with a small number of hosts, the average log-utility derived through the heuristic with the one obtained via exhaustive search over the systems, i.e., using the whole set of $N!$ upper bounding systems and $\epsilon = 0\%$. We observe that in the considered scenarios, the results of the heuristic are very close to the optimal values, and that the approximations on which our heuristic is based has an overall limited impact on the optimality of the operating point derived.

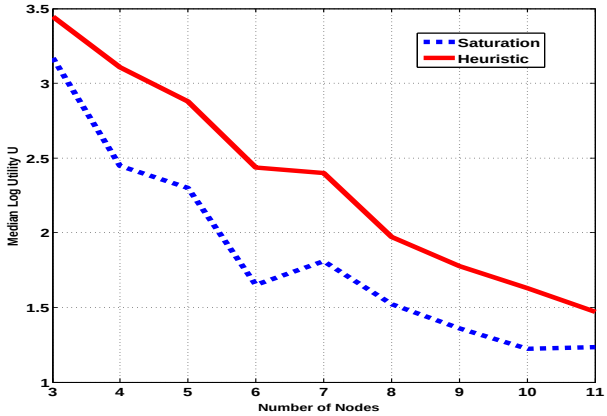
TABLE II
AVERAGE LOG-UTILITY: HEURISTIC VS. EXHAUSTIVE SEARCH

Opt. technique	3 Hosts	4 Hosts	5 Hosts	6 Hosts
Heuristic	3.4932	3.0969	2.7977	2.3660
Ex. Search	3.4932	3.1075	2.8108	2.3991

On Fig. 7(a) we compare the average log-utility, together with the 90% confidence interval, from our heuristic and the one obtainable in the saturated scenario, and in Fig. 7(b) we show the median in the same cases. We can see how in all cases the average log-utility derived by optimizing (through an heuristic) over the set of operating points which are stable according to our method is always at least 20% larger than the one in saturation conditions. As we can see from Fig. 7(b), the difference between the optimal values of the utility function



(a) Average Log-utility U. Heuristic vs. Saturation Condition



(b) Median Log-utility U. Heuristic vs. Saturation Condition

Fig. 7. Log-Utility Results

derived with these two methods is relevant also in distribution, and even larger than the one for the average. It is interesting to note that for both methods, the optimal values of log-utility decrease when the number of hosts increases. This behaviour is due to the increase of the rate of contentions

among hosts, which brings to an increased inefficiency of the MAC. In order to have an intuition of the difference between

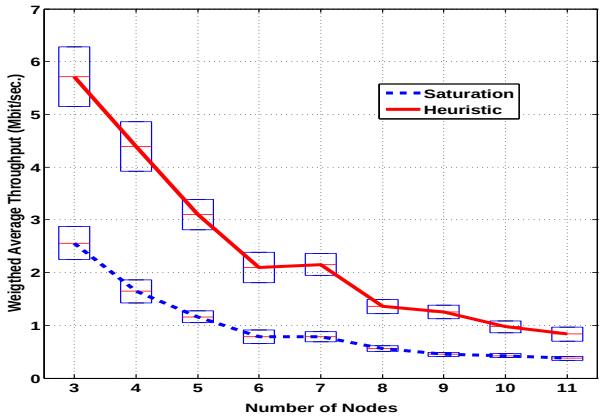


Fig. 8. Weighted Av. Throughput. Heuristic vs. Saturation Condition

the operating points resulting from the heuristic and the ones obtained in saturated conditions, in Fig. 8 we compare the weighted throughput of the two scenarios. We see how our heuristic brings the system to an operating point for which the total average throughput, weighted in order to take into account the relative contribution of each host to the utility of the system, is more than double of the one obtainable through a model based on saturation conditions.

Finally, we analyzed the complexity of the heuristic we propose versus the mentioned exhaustive search approach to solve the optimization problem described. In Table.III we present first the average number of upper bounding systems the two methods analyze in order to get the final Log-Utility. In case of the heuristic, we present per-starting point statistics. Indeed, the execution of the heuristic is independent when different starting point are taken into account, and we imagine the parallel execution of the computation. Then we also present the number of optimizations which the branch-and-bound method performs for each upper bounding system analysed. It can be easily proven that, in the worst case, the number of optimizations the branch-and-bound could perform to reach the optimum for N queues is $\sum_{i=1}^{N-1} 2^{N-i}$. In both cases we show the results up to the point the comparison is computationally feasible (6 hosts), and for the largest scenario we analyzed through our heuristic (11 hosts). We can see how the heuristic requires a considerably inferior number of evaluations, with a very limited impact on the optimality of the value of the log-utility derived (as seen in Table II). It is interesting also to note that, even in the case of the exhaustive search approach, the average number of optimization problems solved with the branch-and-bound method is way far from its upper limit.

TABLE III
COMPLEXITY OF THE HEURISTIC

	Average Per-Start. Point Upp. Systems - Heuristic	Maximum Number Upp. Systems ($N!$)
3 Nodes	5.18	6
4 Nodes	8.72	24
5 Nodes	13.56	120
6 Nodes	19.71	720
11 Nodes	70.03	$39.9 \cdot 10^6$
	Av. Per-Upp.System Optimizations - Heu.	Av. Per-Upp.System Opt. - Exhaust. Search
3 Nodes	4	4
4 Nodes	6.19	8.28
5 Nodes	9.65	12.92
6 Nodes	11.74	17.82
11 Nodes	29.55	-

VI. CONCLUSIONS

In the present paper, we propose a new method for the analysis of coupled processor systems, valid for any number of nodes, and we describe a technique for deriving practical results, which exploits the characteristics of the considered system. Our results show that even on a worst case framework as Network Calculus, taking into account the dynamics of the system through a characterization of the input traffic brings to substantially better resource allocations than those obtainable through a static model of the system. We plan to extend this work in two main directions. On one side, we plan to extend our framework to include CPS where nodes are interconnected, and to consider dependency on queue length rather than only on queue activity. On the other hand, we plan to apply our method to interference-limited wireless scenarios, to multiprocessor systems, data centers, and in general, to all those CPS systems for which simulation has been so far the main method of analysis, in order to get some insight on their performance.

REFERENCES

[1] B. Rengarajan, C. Caramanis, and G. de Veciana, "Analyzing queuing systems with coupled processors through semidefinite programming," 2008. [Online]. Available: http://users.ece.utexas.edu/ gustavo/papers/BCD08.pdf

[2] S. Borst, N. Hegde, and A. Proutiere, "Interacting queues with server selection and coordinated scheduling: application to cellular data networks," *Annals of Operations Research*, vol. 170, no. 1, pp. 59–78, 2009.

[3] R. Madan, J. Borran, A. Sampath, N. Bhushan, A. Khandekar, and T. Ji, "Cell association and interference coordination in heterogeneous lte-a cellular networks." *IEEE JSAC*, vol. 28, no. 9, pp. 1479–1489, 2010.

[4] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[5] G. Fayolle and R. Iasnogorodski, "Solutions of functional equations arising in the analysis of two server queueing models," in *Performance*, 1979, pp. 289–303.

[6] F. Guillemin and D. Pinchon, "Analysis of the weighted fair queuing system with two classes of customers with exponential service times," in *Journal of Applied Probability*, 2004.

[7] S. C. Borst, O. J. Boxma, and P. R. Jelenkovic, "Coupled processors with regularly varying service times," in *INFOCOM*, 2000, pp. 157–164.
[8] S. C. Borst, M. Jonckheere, and L. Leskelä, "Stability of parallel queueing systems with coupled service rates," *Discrete Event Dynamic Systems*, vol. 18, no. 4, pp. 447–472, 2008.
[9] M. Jonckheere and S. C. Borst, "Stability of multi-class queueing systems with state-dependent service rates," in *VALUETOOLS*, 2006, p. 15.
[10] T. Bonald, S. C. Borst, N. Hegde, and A. Proutiére, "Wireless data performance in multi-cell scenarios," in *SIGMETRICS*, 2004, pp. 378–380.
[11] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, ser. LNCS. Springer, 2001, vol. 2050.
[12] M. Fidler and J. B. Schmitt, "On the way to a distributed systems calculus: an end-to-end network calculus with data scaling," *SIGMETRICS*, vol. 34, no. 1, pp. 287–298, jun 2006.
[13] Z.-L. Zhang, D. F. Towsley, and J. F. Kurose, "Statistical analysis of generalized processor sharing scheduling discipline," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1071–1080, 1995.
[14] M. Fidler, "Survey of deterministic and stochastic service curve models in the network calculus," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, pp. 59–86, 2010.
[15] F. Ciucu, "Scaling properties in the stochastic network calculus," Ph.D. dissertation, University of Virginia, Charlottesville, VA, USA, 2007.
[16] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE JSAC*, vol. 18, no. 3, pp. 535–547, 2000.
[17] G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda, "Performance anomaly of 802.11b," in *INFOCOM*, 2003.
[18] A. D. la Oliva, A. Banchs, P. Serrano, and F. A. Zdarsky, "Providing throughput guarantees in heterogeneous wireless mesh networks." *Wirel. Commun. Mob. Comput.*, 2012.
[19] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, 1998.
[20] A. Vecchietti, S. Lee, and I. E. Grossmann, "Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 433–448, 2003.
[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, USA: Cambridge University Press, 2004.
[22] Bit torrent specification. [Online]. Available: https://wiki.theory.org/BitTorrentSpecification/

APPENDIX

A. Proof of Theorem IV.2

Proof. (1) can be proved by induction. Let us imagine that the particular set of active fresh queues at time t is in the set of layers $A = [a_1, ..., a_{n'}]$. Let us start with the service rate of the *first* active node at layer a_1. Thanks to the policer, there is no backlog at its queue reserved for the traffic coming from upper layers and, obviously, at time t no traffic is neither arriving from those layers. Then $O_{a_1}(t) = R^{up}_{n_{a_1}}$, respecting (1). Thanks to the fluid limit assumption we do on the GPS nodes, at layer a_2, the only flow the queue for the upper layers is receiving comes from layer a_1. Then $O_{a_2}(t) = R^{up}_{n_{a_2}} - min(R^{up}_{n_{a_1}} S_{a_2,a_1}, R^{up}_{n_{a_2}} - R^{sat}_{n_{a_2}})$, respecting again (1). By induction, such expression can be easily extended to all the set A, proving the theorem. □